DİJİTAL KÜTÜPHANE KANALI

Keyifli Okumalar Dileriz

Binlerce kategoride milyonlarca PDF e-kitap, ders kitapları, dergiler ve romanlara platformumuz üzerinden ücretsiz ulaşabilirsiniz.

RESMİ WEB SİTEMİZ

https://rantey.org

Yedek ve Duyuru Kanallarımız
Sitemize erişim engeli gelmesi veya adres değişikliği durumunda aşağıdaki kanallardan güncel giriş adresine erişebilirsiniz:

• Telegram: https://t.me/birkitapbir

• WhatsApp Kanalımız : https://whatsapp.com/channel/0029VbDHv8uE50Us4IvMoc0Y

BİLGİSAYAR

ÇAĞDAŞ TEKNOLOJİ

ANSİKLOPEDİSİ

1

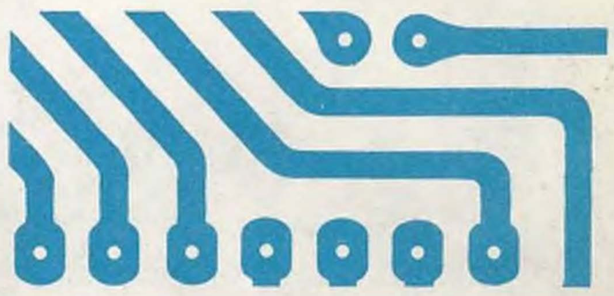
İletişim Yayınları

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYAR

ÇAĞDAŞ TEKNOLOJİ ANSİKLOPEDİSİ

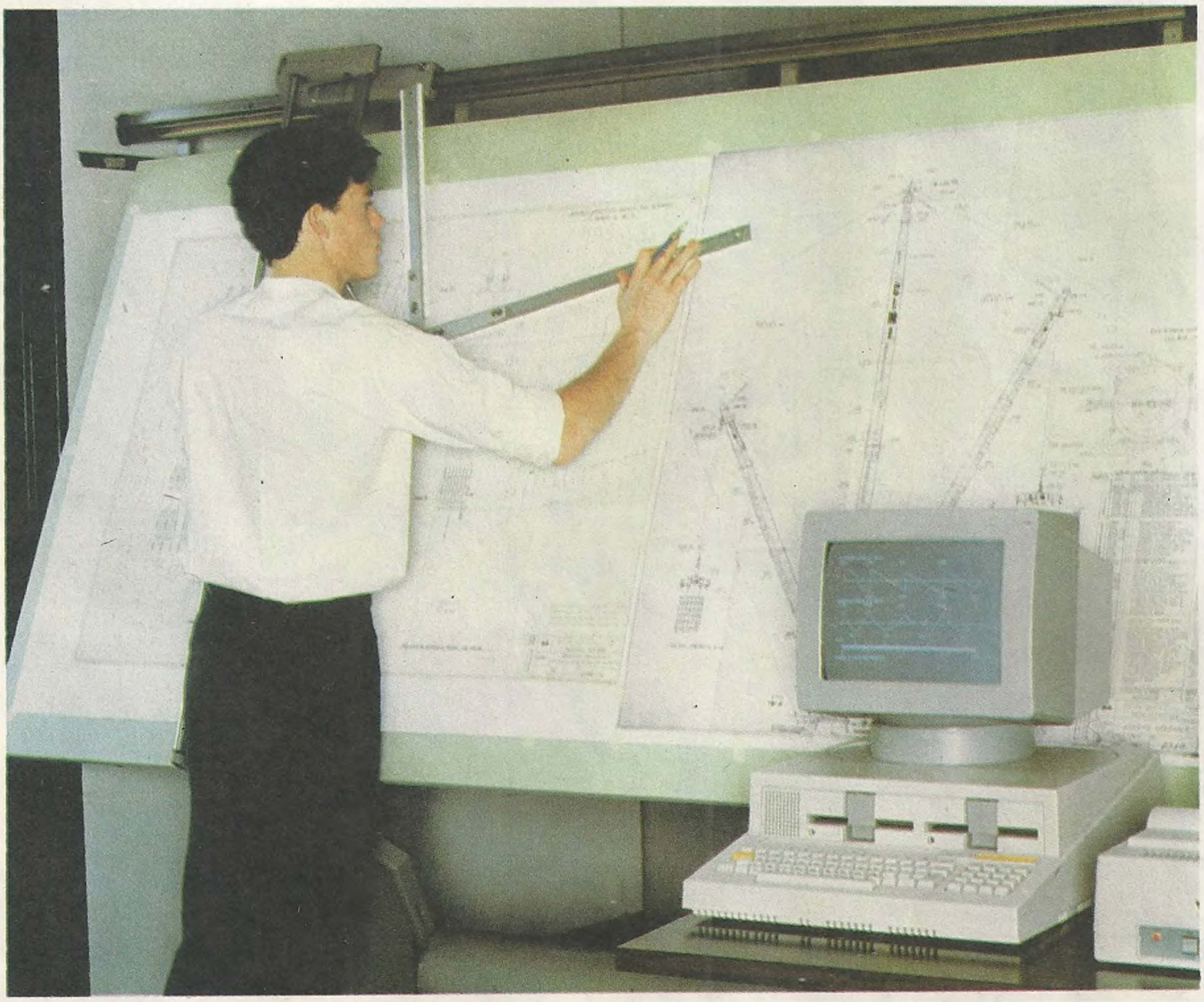


Cilt 1

Ücretsiz PDF Kitap Arsivi - https://rantey.org

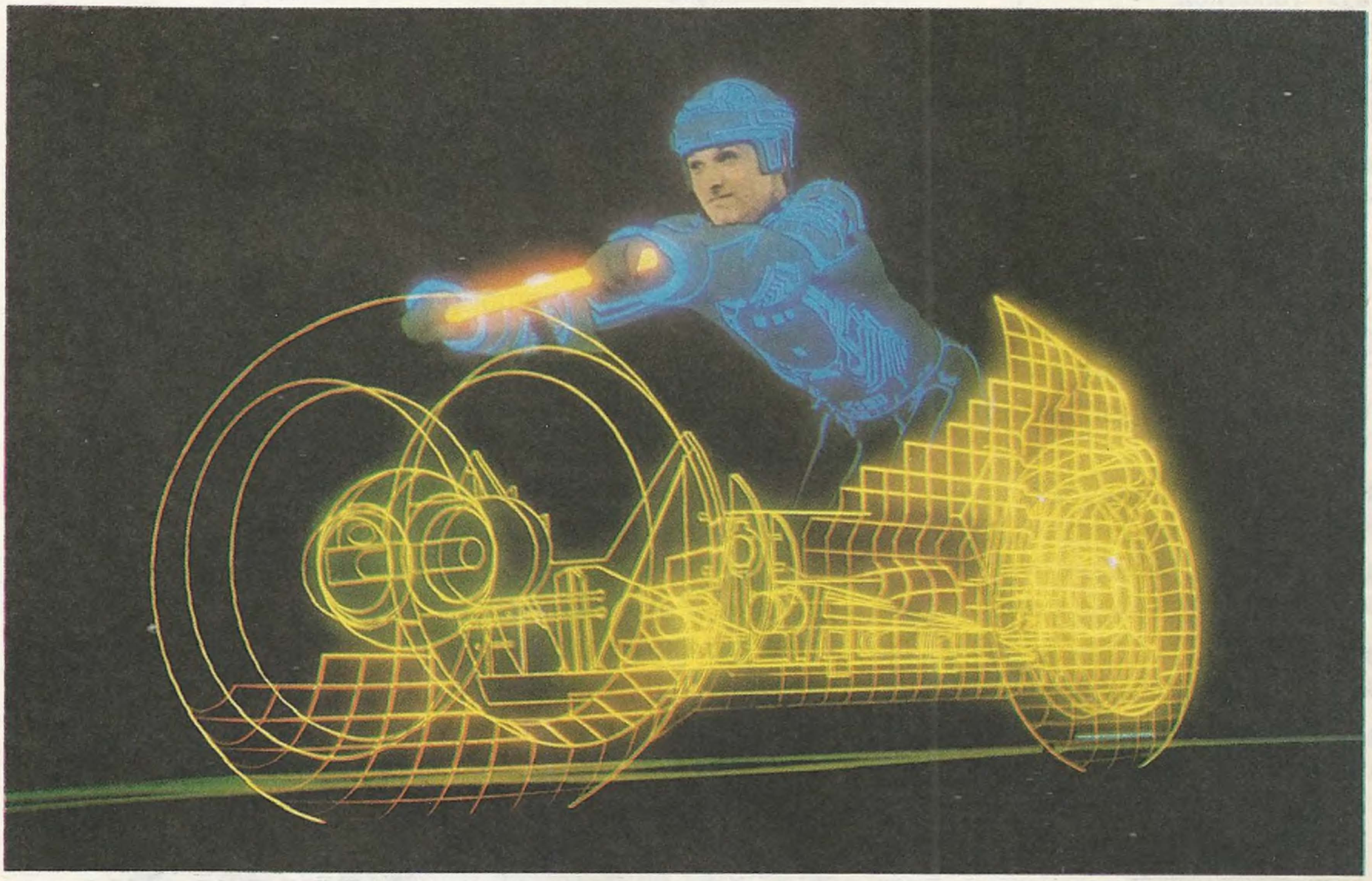
ÇAĞDAŞ TEKNOLOJİ ANSİKLOPEDİSİ

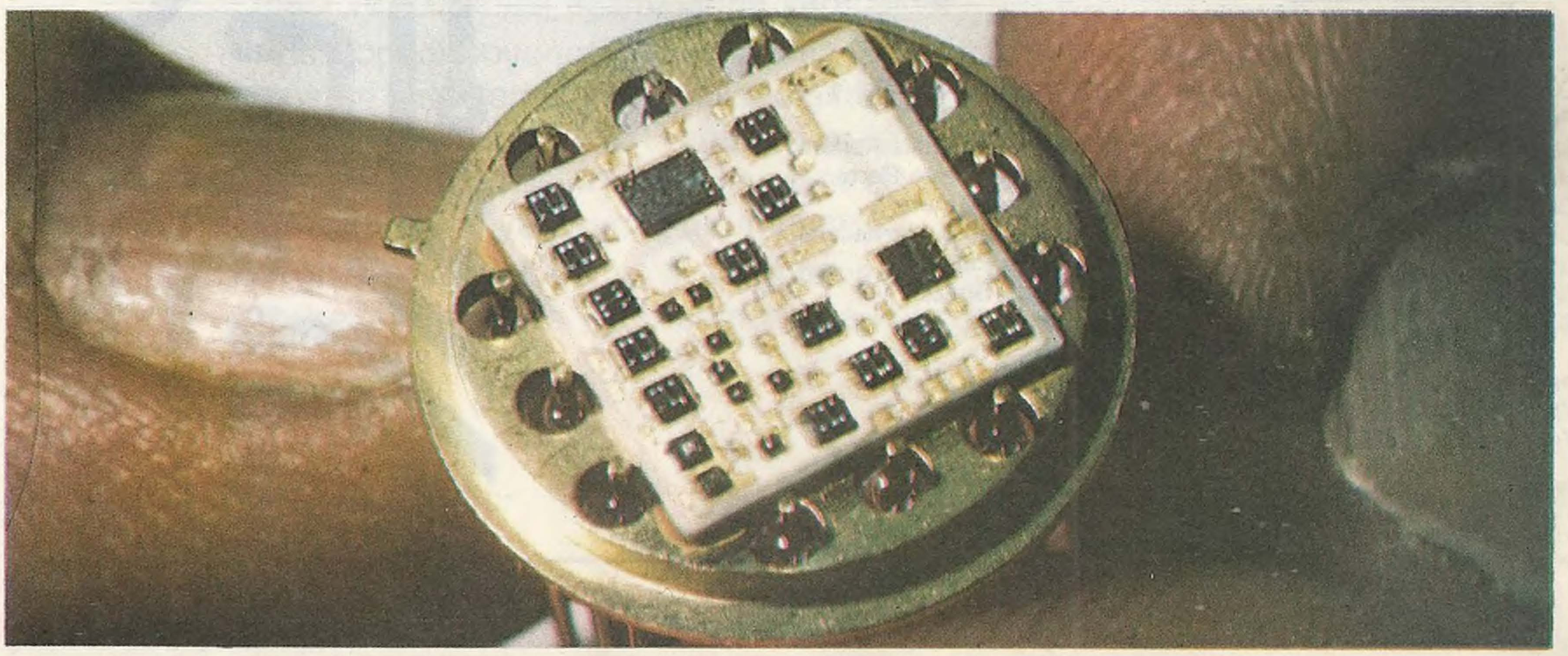
İletişim Yayınları



Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYAR 1





Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYAR

ÇAĞDAŞ TEKNOLOJİ ANSİKLOPEDİSİ

İletişim Yayıncılık A.Ş. adına sahibi
Vedat Çakmak

Genel Yönetmen
Murat Belge

Yayın Yönetmeni
İskender Savaşır

Sorumlu Yazı İşleri Müdürü
Ahmet Ziya Bayman

Danışma Kurulu
Kemal İnan, Ayşe Şenatalar, Taha Parla

Yayın Kurulu
Vedat Çakmak, Bilge Karasu, İskender Savaşır

Çeviri ve Yayına Hazırlayanlar
Cengiz Çetiner, Cengiz Onural, Ahmet Ziya Bayman

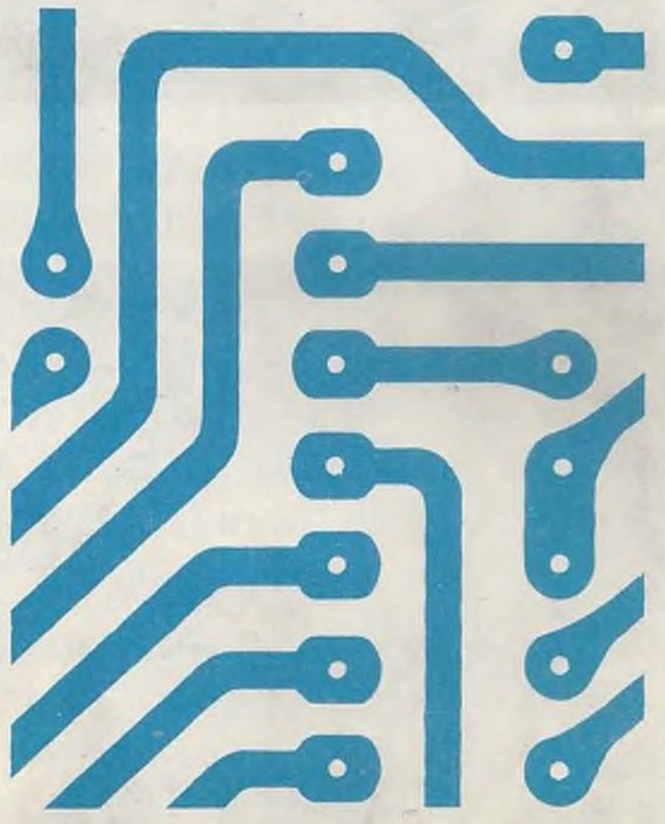
Teknik Yönetmen
Edip Alşar

Reklam Müdürü
Aslı Yada

Grafik/Pikaj
Saadettin Oktay

Dizgi
Serpil Şahin, Hüsnü Abbas

Baskı
Alaş Matbaası



© *Armando Curcio Editore, 1984*
Özgün adı: **''Basic, Enciclopedia dell'Informatica, dei Mini e Personal Computer''**
Sante Senni yönetiminde, E.G.S. Şirketi tarafından Giulia Abriani, Maria Albergo, Giovanni Aloisi, Vittoria Antinori, Alberto Berni, Carla Bertini, Ricardo Cantani, Maria Rita Ciucci, Gabriella Costarelli, Patrizia Fazzio, Renato Lazzarini, Marcella Mareaccini, Gianni Mazzoleni, Rolando Mazzoni, Piergiorgio Palma, Rosella Pozza, Laura Salvini, Roberto Sed, Ugo Spezice ve Graziella Tassi'nin katkılarıyla hazırlanmıştır.

İletişim Yayınları,
Klodfarer Caddesi, İletişim Han, Cağaloğlu, İstanbul
Telefon: 520 14 53/54/55

© Bütün hakları İletişim Yayıncılık A.Ş.'ne aittir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

ÇIKARKEN

Çok değil, daha beş altı yıl önce çoğumuz için bilgisayar, ya da o zamanlar anıldığı adla ''elektronik beyin'', daha çok bilim-kurgu filmlerinden tanıdığımız bir fanteziydi. Oysa şimdilerde, boyutları ve önemi bakımından Sanayi Devrimi ile karşılaştırılabilecek bir Bilişim Devrimi gündelik yaşamımızı dönüştürmeye başladı.

Bilgisayarı diğer teknolojik buluşlardan ayıran en önemli özellik, yalnızca yeni olanaklar sunmakla kalmayıp, aynı zamanda yerleşik alışkanlıklarımızı da yeniden düzenleyecekmiş gibi görünmesi. Örneğin, bugünün çocukları da oyun oynamayı sürdürüyor; ancak karşılarında yepyeni bir oyun arkadaşı var. Daha şimdiden, Batı'da, bilgisayar, eğitim sisteminin vazgeçilmez bir parçası haline geldiği gibi, aile bütçelerinin hazırlanması ve banka hesaplarının tutulmasından tutun da, alışveriş listelerinin hazırlanmasına kadar çeşitli gündelik işler bilgisayarların yardımıyla yürütülüyor.

Bugün ülkemizde ''bilgisayar kullanmayı bilmek'' daha çok bir uzmanlık, bilgisayar teknolojisi ise yeni iş olanakları sunan bir işkolu olarak düşünülüyor. Bir yandan da teknolojideki gelişmeler bilgisayarların fiyatlarının hızla düşmesine yol açıyor. Çok geçmeden ekranı ve bilgiyazarıyla birlikte kişisel bir bilgisayar, bir video cihazı fiyatına satın alınabilecek ve giderek daha çok sayıda işkolu bilgisayarlardan yararlanmaya başlayacak. Bu gelişmelere bağlı olarak da, bilgisayar kullanmak bir uzmanlık alanı olmaktan çıkıp, birçok değişik işin gerektirdiği bir beceri olacak.

Elinizdeki ansiklopedi bu gerekçelerle, size bilgisayar dünyasını tanıtmak ve bilgisayar kullanma becerisini kazandırmak üzere hazırlandı. Bu yüzden, hem okura piyasadaki aygıtların çeşitliliği ve herhangi bir aygıtı seçerken başvurması gereken ölçütler hakkında temel bilgileri sağlayan bir el kitabı hem de bir ders kitabı olma niteliği taşıyor.

Unutulmamalı ki, bilgisayar kullanmayı öğrenmek her şeyden önce bir dil öğrenmektir. Bu yüzden bilgisayar ansiklopedisi, sayısız şema ve resimleriyle, bilgisayarların dillerini, simgelerini ve uygulama olanaklarını öğreten bir kurs şeklinde hazırlandı. Okuyucular bilgisayarların kısa tarihçesi, yapıları ve sınıflandırılmaları hakkında bilgi edindikten sonra, giderek bir programın tasarlanması ve yazılması aşamalarına varacaklar.

Ansiklopedi boyunca bilişim alanında yerleşmiş olan terimler kullanıldı. Aynı kavram farklı terimlerle karşılandığında seçenekler belirtildi. Ayrıca kullandığımız terimlerin İngilizce, Fransızca ve Almanca karşılıklarını içeren terimler kılavuzu ekimizin, bilişim konusunu daha da derinlemesine incelemek isteyen okurlarımızın işine yarayacağını umuyoruz.

Bilgisayar Ansiklopedisi'ni yayımlamakla okurlarımızın çağdaş teknolojiden en verimli bir biçimde yararlanmalarına katkıda bulunduğumuza inanıyoruz.

İletişim Yayınları

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Ücretsiz PDF Kitap Arsivi - https://rantey.org

İÇİNDEKİLER

Bilgisayarın Yapısı

Durmadan genişleyen bilgisayar pazarında herhangi bir sistemin özelliklerini ve yeteneklerini değerlendirebilmek için mantıksal yapısını, yaptığı işlemleri, bu işlemlerin nasıl kullanıldığını ve verilerin alışveriş yöntemlerini anlamak gerekir.

Bilgisayar:
Büyük Bilinmeyen

Program nedir, programlanabilen bir makine ne işe yarar? Veriler; tanımı ve uygulama alanları. Otomatik hesaplayıcının geçmişi ve geleceği: Abaküsden ilk bilgisayar denemelerine. Hesap makineleri. ''Programlama'' fikri; ilk elektronik hesaplayıcı doğuyor. Bilgisayarın gelişimi: Mikroelektronik ''düşünen'' makineleri mümkün kılacak mı? İş dünyasında bilgisayarın en önemli uygulama alanları.

Bilgisayarın Mantığı

Bir bilgisayar nasıl ''düşünür''? Benzeştirmeli ve sayısal bilgisayarlar. Elektronik terazi: İlk uygulama örneği. Bilgisayarların sınıflandırılması. Veri taşıma ağları. Bilgisayarların birbiriyle bağlanması. Bir bilgi-işlem merkezinin yapısı.

Sayı Sistemleri

Darbeler: Tanımlama ve örnekler. Sayı sistemleri. Değişik tabanlı sayıları çevirme. İki tabanlı aritmetik işlemler. İki, sekiz ve onaltı tabanlı sayıların çeviri çizelgeleri.

Mantık Ögeleri

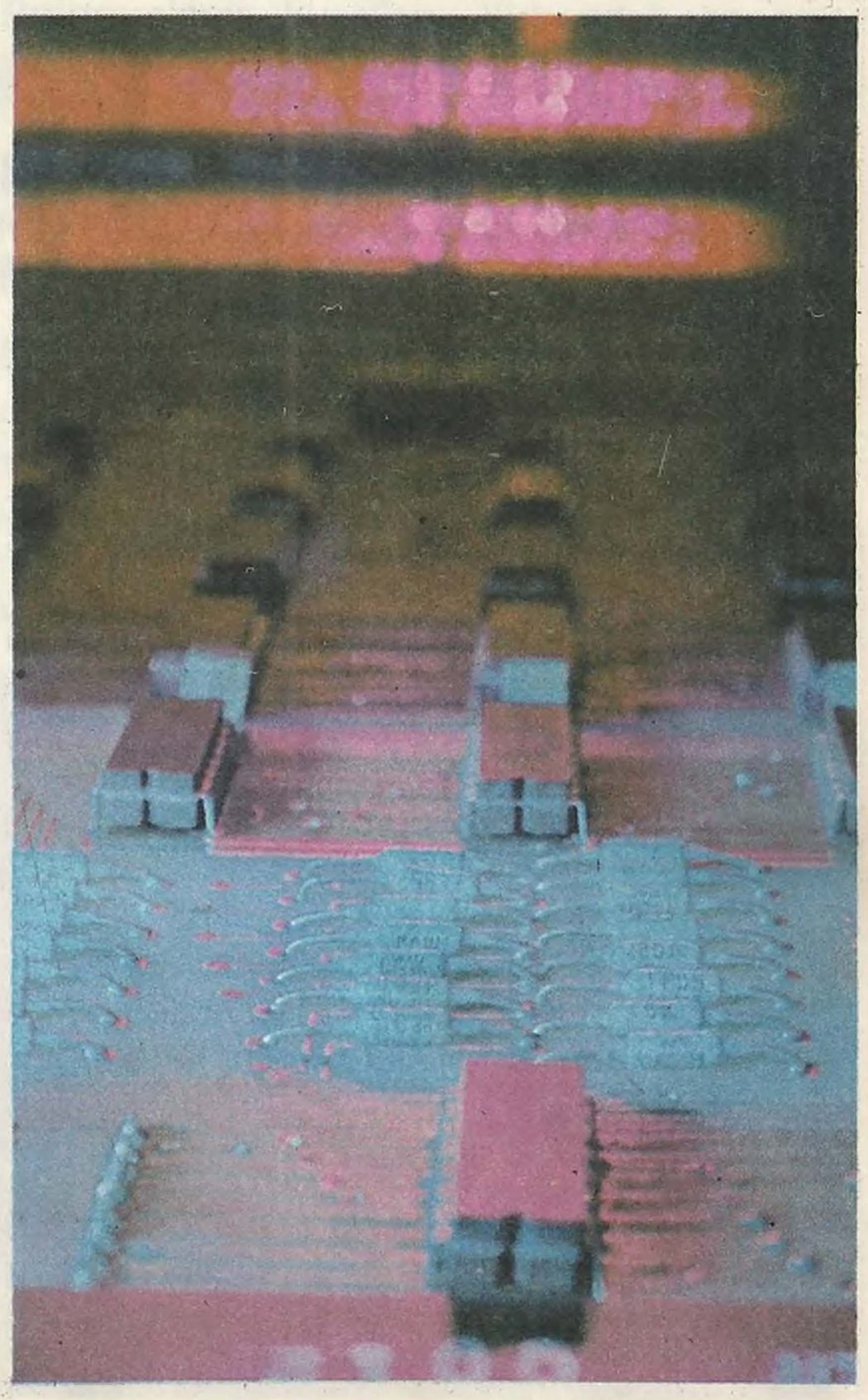
Boole cebiri işlevleri: Nedir ve nasıl uygulanırlar? Ortak özellikler. Boole cebirinin uygulanması: Bellekte arama ve seçim. Tümleşik devrelerin uygulanması. Mantıksal işlev gören tümleşik devrelere örnekler. Verilerin korunması: Boole işlevi kodlaması yapan devreler.

Veri Taşıma Kodları

Kodlar: Çizimlerin makine düzenine çevrilmesi. ASCII kodunun açıklanması: Güdüm işaretleri, harflerin ve sayıların kodlanması. Güvenlik kodlarının hazırlanması. Veri taşıma sistemleri. Eşzamanlı ve eşzamansız taşıma.

Kişisel Bilgisayarlar

Kişisel bir bilgisayarın yapısı. En önemli birimler. Taşıt devreleri. Çevresel birimler, veri taşıma yöntemleri ve kuralları. Merkez işlem biriminin (MİB) işlevi ve işlemler. Bir minikişlemcinin çalışma şekli: Çevrimler ve komutlar. Program sayıcı, sicil, **deste**-bellek. Belleğin kullanımı. Komut çevrimine örnekler. Program içi atlamalar. Dış dünyayla ilişki: Araya girme **(Interrupt)**. Ekrandaki kaymanın kesilmesi **(Hazır-Hold)**. Verilerin taşınmasında doğrudan bellek erişimi.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Pratik Bir Sorunun Mantıksal Bir Şemaya Dönüşümü

Program nedir? Pratik bir sorunun mantıksal bir düzene dönüştürülüp çözülmesinde kullanılan simgeler ve yöntemler. Başlıca simgesel dillere giriş. Veri saklama yöntemleri ve belgelemede izlenen mantık. Veri yapıları.

Programlama

Programlama dilleri. Derleyici ve yorumlayıcı. Derleme. Başlıca yüksek düzeyli diller. Diğer programlama dilleri. Özellikleri, uygulama alanları ve değerlendirme ölçütleri.

Çözümleme ve Akış Şemaları

Programların hazırlanışı: Sorunun ve verilenlerin çözümlenmesi. Sorunun şemaya aktarılması ve sorunu bölümlere ayırma yöntemleri. Uygulama alanları: Belleği kullanma yollarının tanımları. Akış şemaları: Tanımlar ve simgeler. Simgelerin kullanımı. Dizinler ve çentikler. Döngüler ve sayıcılar: Kullanım olanakları, uygulama örnekleri. İki önemli uygulama alanı: Ticaret ve teknik. Programların yapısında başlıca farklar. Geometri formülleri. Ticaret formülleri.

Veri Saklama

Yığın belleklerinin başlıca çeşitleri. Kütükler. Minik ve küçük bilgisayarlarda kütük yapısı. Kütükler ve kayıtlar **(Records)**. Kütüklerin farklı biçimleri: Başlıca özellikleri ve uygulama alanları. Verilerin yapısı ve düzenlenmesi: Açıklayıcı birkaç örnek. Kütüklerin açıklanması. Karmaşık kütükler: Başlıca yöntemlere ilişkin bilgiler. Veri bankaları. Veri bankalarının hazırlanışına birkaç örnek.

Programların Hazırlanması

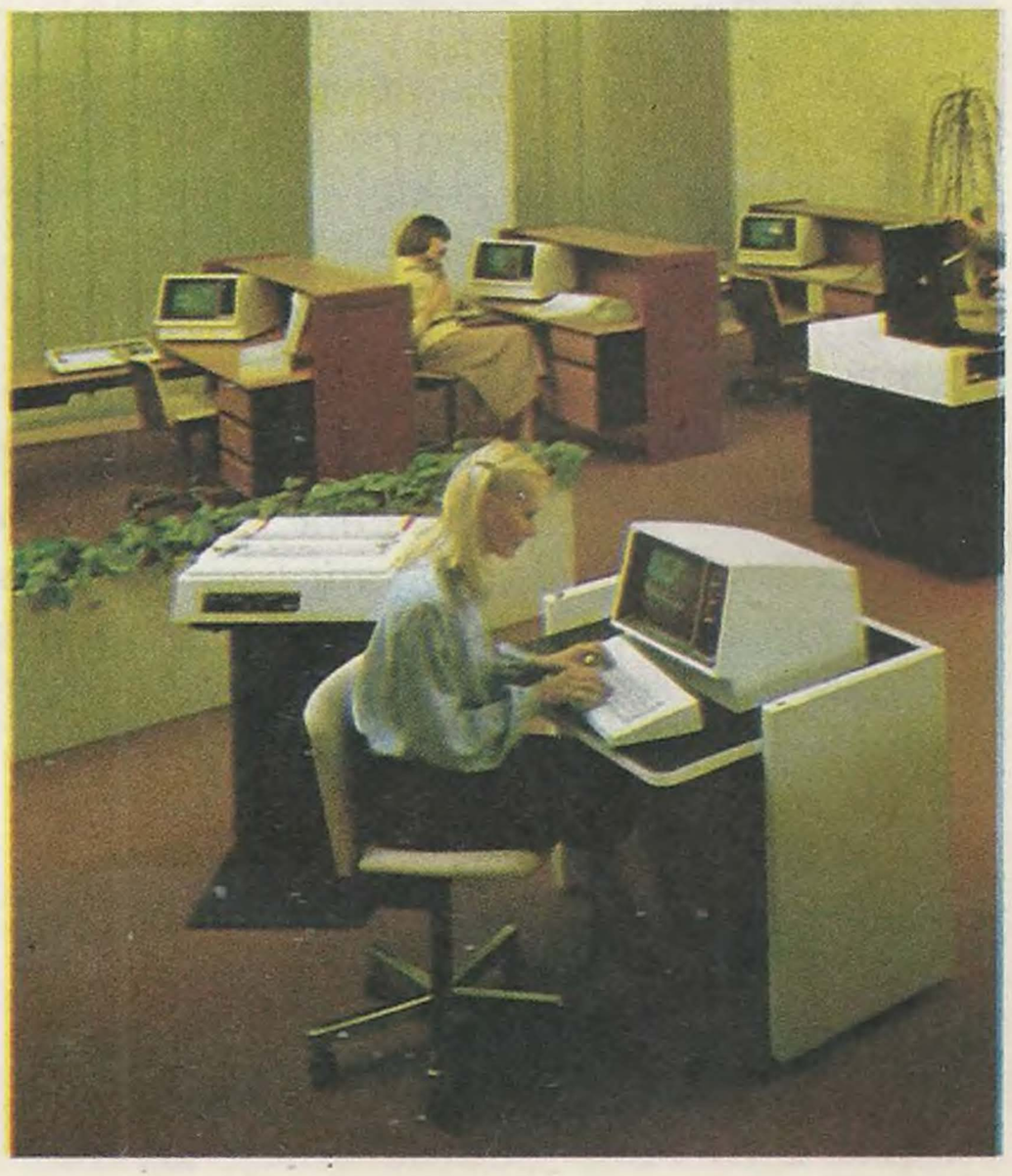
Başlıca kişisel ve küçük bilgisayarların tanıtımı. Karşılaştırma çizelgeleri ve değerlendirme ölçütleri. İşletme sistemlerinin değerlendirilmesi. Karmaşık programların bölümlenmesi. Aygıt seçiminde ölçütler.

Programlama Dili Olarak BASIC

BASIC ile programlamada kuramlar ve yöntemler: ''Etkileşimli Programlama Yöntemi''nin nasıl çalıştığı. Matematiksel işlevler BASIC'te nasıl yazılır? BASIC komutlarının doğru kullanımı için yöntemler. Bir BASIC programının bölümlenmesi.

Genel Bilgiler

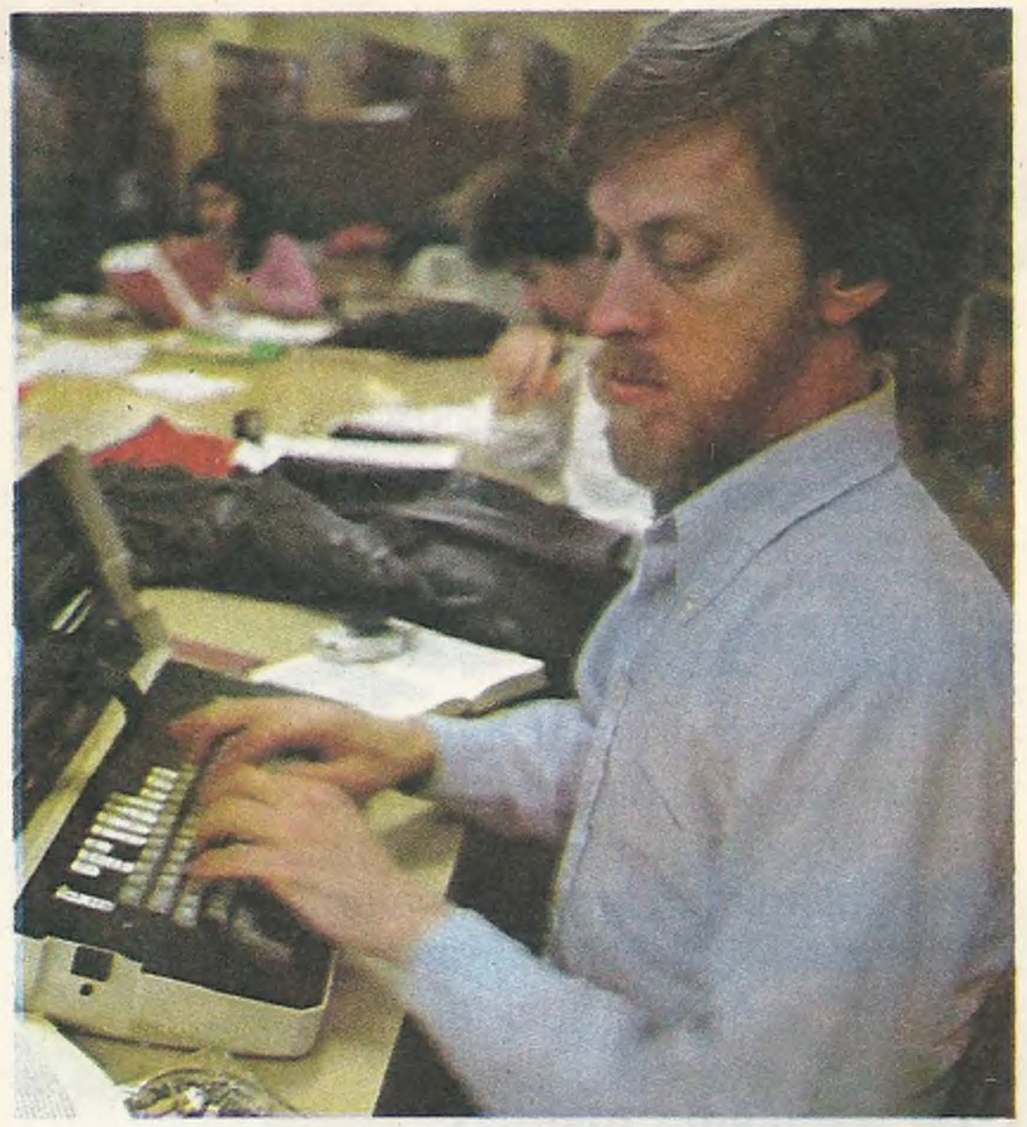
İşlemlerin yürütülmesinde yorumlanmış BASIC kullanımı: ''Etkileşimli Programlama Yöntemi''nde BASIC. Sabitler ve değişkenler, tanımlar, hesaplamada dakiklik. Kayan noktalı gösterilişte gerçel sayılar. Tam ve gerçel sayıların makine içinde betimlenmesi. Bellek bölümleri: Program ve veri belleği. Aritmetik işlemler: Hesap işleminde öncelik, dakiklik ve yuvarlama. Katar değişkenleri **(string variables)**: Ne oldukları ve nasıl uygulandıkları. Katar değişkenleriyle işlem. Kişisel bilgisayarların değişik türleri arasında, erişilebilecek dakiklik açısından farklılıklar. Kısmi katarlar.



Güdüm

BASIC komutları. Programların yüklenmesi ve bellekte saklanması. Çıkış hazırlığı. TRON ve TROFF komutları; arıza aramada kullanılışları. BASIC komutlarının yazı düzeni.

Ücretsiz PDF Kitap Arsivi - https://rantey.org



Komutlar

Sayısal değişkenlere ve katar değişkenlerine değer aktarımı. Döngüler: Uygulama ve sözdizimi. Döngülerde başlıca arızalar. Atlama: Komutla atlama, altprogram çağırma. Boole işlevlerinin programlamada kullanımı. Koşullu komutlar. Arıza aramada koşullu komutların kullanımı. Uygulama alanları: Bilanço hesabı programının formüle edilmesi ve geliştirilmesi.

İşlevler

İşlev kavramı. Çizisel gösterim. BASIC'de sayısal işlevler: Uygulamaları, sınırları ve dakiklikleri. Kullanıcının tanımladığı sayısal işlevler.

BASIC'de katar işlevleri. Kullanıcının tanımladığı katar işlevleri. İstatistikle ilgili açıklamalar: Raslansal sayıların üretilmesi. Uygulama alanları: Denklem çözümü için bir programın hazırlanması, katarları işlemek için program geliştirme. Matematiksel formüller: İşlevler.

Dizeyler (Arrays)

Veri saklama. Tek ve çok boyutlu dizeylerin tanımı. Sayısal dizeyler ve katar dizeyleri. Dizeylerde değer aktarma. Uygulama alanları: İstenilen uzunlukta sayı haneleriyle işlem yapmak için bir programın hazırlanışı; arıza bildirmede dizeylerin kullanımı; alan hesabı için bir programın geliştirilmesi.

Veri Giriş-Çıkış İşlevleri

Veri okuma ve yazma komutları. Düzenleme komutları. İmleç: Ekrana yazma komutları. Bilgisayarın kullanımı, işlevlerin kodları. Programlanabilen tuşlar: Programlama komutları ve kullanımı. Uygulama alanları: Verilerin yerleştirilmesi için bir programın yazılması. MENÜ'ler, program seçimi ve tasarımı için sistemler. BİT'lerle oynama komutları. Doğrudan bellek erişimi için komutlar.

Yığın Belleklerinin Kullanımı

Diskte veri saklama düzeni. Verilerin kodlanması ve çevirisi. Diskler için okuma-yazma kuralları. Diskle ilgili sayısal ve katar komutlar.

Çeşitli belleklerde kütüklerin kullanımı. Karmaşık kütüklerin kullanımı: Veri kullanımı için birkaç program. Kütüklerin yerleştirilmesi. Bu işleve bağlı sorunlar. Kütük yerleşimi için hizmet programları. Verilerin aranması ve seçimi: Karşılaştırıcı altprogramlar. Uygulama alanları: Bir adres dizin programının yazılması; başlıklar ve ajanda; stok durumu programı; sayıları yuvarlamada kullanılan bir altprogramın hazırlanışı.

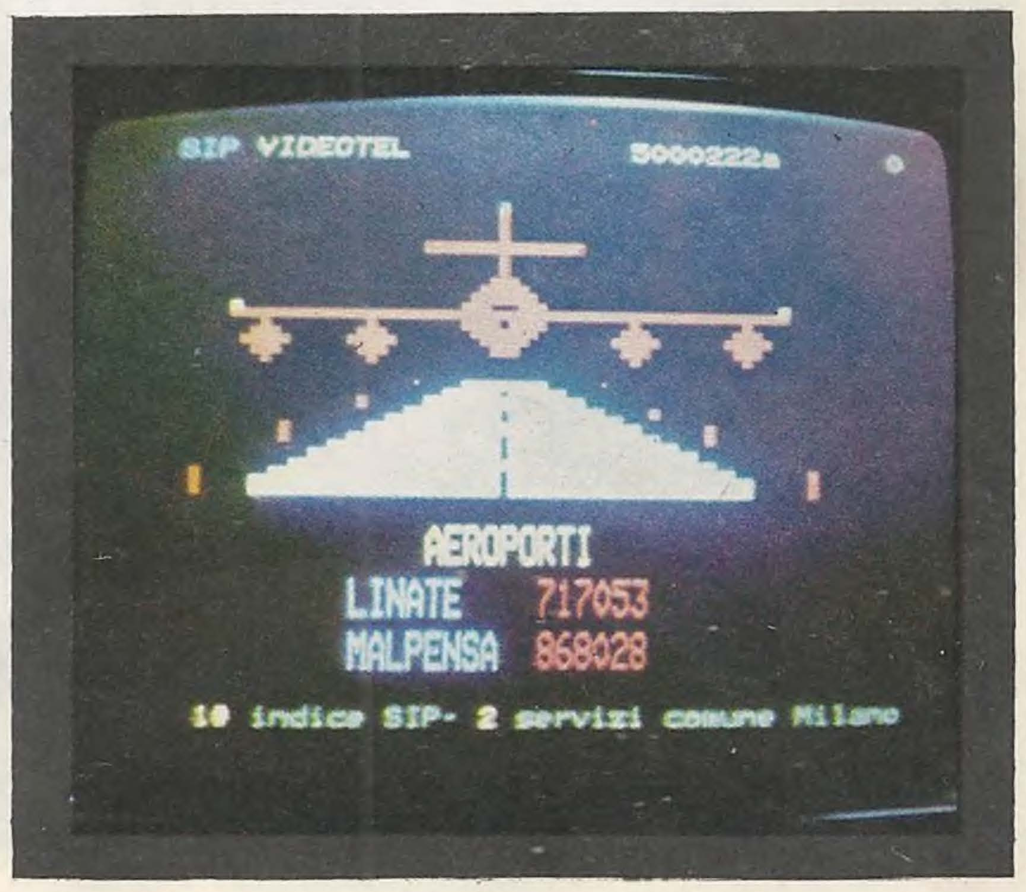
Program Yapısı

Programlama kuralları. Yukarıdan aşağıya yaklaşım yöntemi. Altprogramların genelleştirilmesi. Programların kısımlara ayrılması.

Programlar ya da program parçaları arasında veri iletişimi; yardımcı bellek olarak diskin kullanılması. Ekran örtüleri, uygulama alanları.

Sonuçlar

Program inceleme yöntemleri. Arıza arama. Toplu halde BASIC komutları. Uygulama alanları: Aynı programda birçok kütüğün kullanılması. Bir cari hesap programı için akış şemalarının hazırlanması.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Yüksek Düzeyde Programlama

Üç önemli bilgisayar dili: ASSEMBLER, COBOL, FORTRAN. FORTRAN komutlarının BASIC'inkilerle karşılaştırmalı olarak incelenmesi ve bilimde kullanım alanları. Karışık dillerle programlama yöntemleri. Çizimlerde bilgisayarın kullanımı. Karmaşık aygıtların otomatik güdümü konusu ağırlıklı olmak üzere kullanım alanları.

Assembler

Assembler dil: Yapısı ve başlıca özellikleri. Çeviricide başlıca komutların düzeni. Birkaç uygulama örneği. Büyük çeviricinin kullanımı: Ne zaman ve nasıl faydalı olduğu. Uygulama alanları: Cari hesap işlemi, tam akış şemasının hazırlanması.

COBOL

COBOL programlama dili. Yapısı ve başlıca özellikleri. Birkaç komut örneği.

FORTRAN

FORTRAN'ın yapısı. Uygulama alanı. FORTRAN komutları. FORTRAN ve BASIC'deki komutların karşılaştırılması. Toplu halde FORTRAN komutları. Uygulama alanları: Cari hesaplar; programların yazılması; birkaç BASİC programının FORTRAN'a çevirisi.

Karışık Programlar

Değişik dillerde programlama sistemleri. Karışık programların kullanımı. Derlenmiş BASIC: Yorumlanmış BASIC'e göre farkları ve üstünlükleri. İşletme sistemleri, yapı ve işleyişleri; işletme sistemlerinde başlıca programlar. Çoklu program düzeninde işletme sistemi. Uygulama: Bordro programı.

Bilgisayar Çizimi

Çizimler, histogramlar, tasarım ölçekleri. Histogram ve çizim hazırlayan altprogramlar. Ekran çizimi. Çizici. Histogram çıkışında bilgiyazarın kullanımı. Uygulama alanları: Çizim altprogramlarının geliştirilmesi; histogram çıkışı altprogramların hazırlanması.

Özel Uygulama Alanları

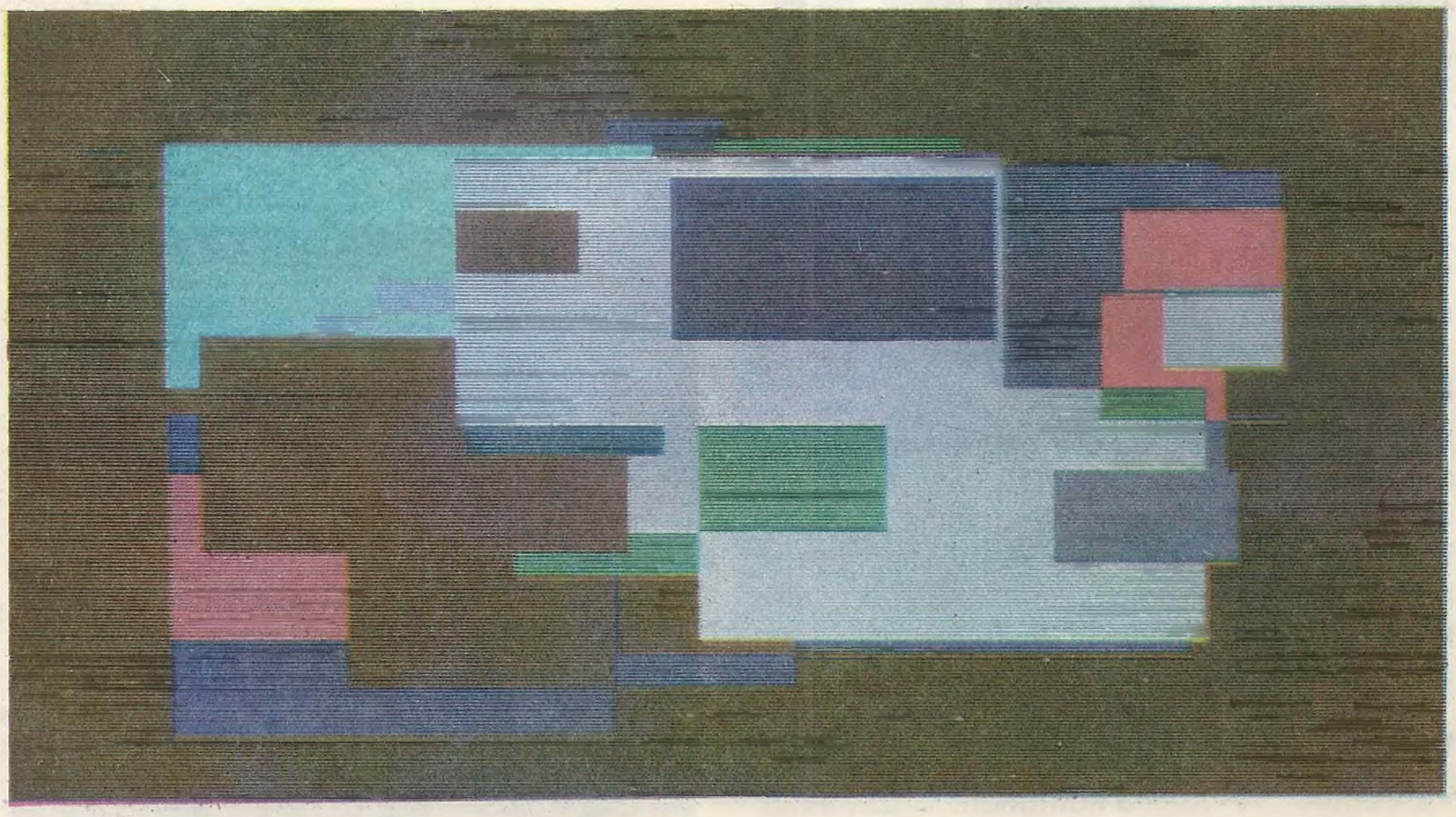
Geleceğe bir bakış: Minik işlemci sistemlerin gelişimi. Otomatik güdüm sistemlerinde minik bilgisayar kullanımı.

Temel Kavramlar

Temel kavramların listesi.

Kavram Açıklamaları

Ansiklopedi'nin sonunda ciltlerden yararlanmayı kolaylaştıracak bir açıklamalı dizin bulunacaktır. Bu bölüm ayrıca temel kavramların dört dilde karşılıklarını veren bir ''Terimler Kılavuzu''nu da içerecektir.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

KÜÇÜK SÖZLÜK

Adres: Bellekteki herhangi bir yeri gösteren sayı. Bellekten veri okumak için ya da belleğe veri yerleştirmek için, komutun belleğin hangi kesimiyle ilgili olduğunun belirtilmiş olması gerekir. Bunu belirten sayıya adres denir. Orta büyüklükteki bilgisayarlarda belleğin hacmine bağlı olarak adresler 0'dan 64.000'e kadar çıkabilirler. Daha büyük bilgisayarlarda bu sayı 500.000 ya da 1.000.000'a ulaşabilir.

Akış Şeması: Bir programın yapması istenen işlerin çizgiyle adım adım gösterilmesi. ''Akış'', sorunu çözmek için makinenin izlemesi gereken yolu belirler.

Alan: Herhangi bir kaydın ''yapısal'' bölümlenmeleri. Alanlar, yalnızca sayıları içerebiliyorlarsa ''sayısal'', aynı zamanda harf de içerebiliyorlarsa ''harfsayısal'' alanlar diye anılırlar.

Altprogramlar: Özel bir işlev görmek üzere hazırlanmış program parçaları. Ana programın herhangi bir noktasından çağrılıp orada kullanılabilirler.

Anahtar: Herhangi bir kaydın yerinin bulunmasını sağlayan bölümü. Örneğin bir telefon rehberinde soyadları, kayıtların anahtar bölümüdür.

Araya Girme: Dışardan gelen bir talebe bağlı olarak programın çalışmasına ara verilmesi.

ASCII: Veri taşımada kullanılan en önemli kodlama sistemi.

Atlama: Programın belli bir bölümünün atlanarak işlenmemesini sağlayan komut.

Bayt (Byte): Komutların tanıdığı temel bilgi birimi; genellikle 8 ''BİT''ten oluşur.

Birleştir: Program veya verileri birleştirme komutu. Bu komutlar, çeşitli kütükler ya da program bölümlerini tek bir bütün haline getirirler.

BİT: 0 ya da 1 değerini alabilen en küçük bilgi birimi.

Çentik: Bir durumun varlığına ya da yokluğuna bağlı olarak 0 ya da 1 değerini alan değişkenler. Söz konusu durumun oluşup oluşmadığı hakkındaki bilgiyi saklamak için kullanılırlar.

Çevresel Gereçler: Bilgisayara bağlı klavye, bilgiyazar gibi aygıtlar.

DBE: Doğrudan bellek erişimiyle veri taşıma yöntemi.

Darbe: Kısa elektrik sinyali. Her darbe bir ''BİT''e karşılık gelir (tekabül eder).

Derlenmiş Program: Derleyici tarafından makine diline çevrilmiş program.

Derleyici: Bilgisayarın genel işletim sisteminin bir parçası olan bir program. Kullanıcıların yazdığı kişisel programları makinenin anlayabileceği makine diline çevirir.

Deste-Bellek: Belleğin yalnızca ''Geriye atlama'' komutlarında kullanılan bölümü. Atlama sırasında ''Atla'' komutunun adresini saklar, böylelikle, istendiği takdirde, atlama noktasına geri dönülerek, atlanmış olan veriler de yeniden işlenebilir.

Disket: Küçük, değiştirilebilir, manyetik disk. Kayıt kapasitesi, türüne göre, 160.000 ile 1.200.000 arasında değişir.

Döngü: Tekrar tekrar işleme konması gereken program bölümleri. Bu bölümlerin kaç kere tekrarlanacağı ya önceden belirlenir ya da bir koşulun gerçekleşmesine bağlanır. Örneğin bir döngü, klavyede bir tuşa basılıncaya kadar tekrarlanabilir.

Giriş/Çıkış (G/Ç: Input/Output, I/O): Bilgisayar ile çevresel aygıtları arasında bilgi alışverişini sağlayan genel giriş ve çıkış işlevleri.

Güdüm İşaretleri: Bir harfi ya da sayıyı çevirmek için değil, bir komut göndermek ya da bir komutu yanıtlamak için kullanılan ASCII kodları..

Ham Program: Kullanıcının simgesel dilde yazdığı program.

Hazır (Hold): MİB'nin bir işlemi yürütmeye hazır olması durumu.

İmleç: Ekranda genellikle küçük bir dikdörtgen şeklinde görünen işaret. Giriş ya da çıkış verilerinin yazılmaya başlandığı noktayı belirtir. Yani, genel olarak, ekranda görünecek işaretin yerini gösterir.

Kayıt: Herhangi bir veri kütüğünü oluşturan birimler; örneğin bir adresler kütüğünde, adreslerden her biri bir kayıt oluşturur.

Kütük: Belleğin aynı türden bilgileri saklamaya ayrılmış bir bölümü.

Merkezi İşlem Birimi (MİB): Kişisel bilgisayarların bütün işlemlerini denetleyen devre.

Sıralayıcı: Verilerin (örneğin alfabetik sıraya göre) sıralanmasını sağlayan program.

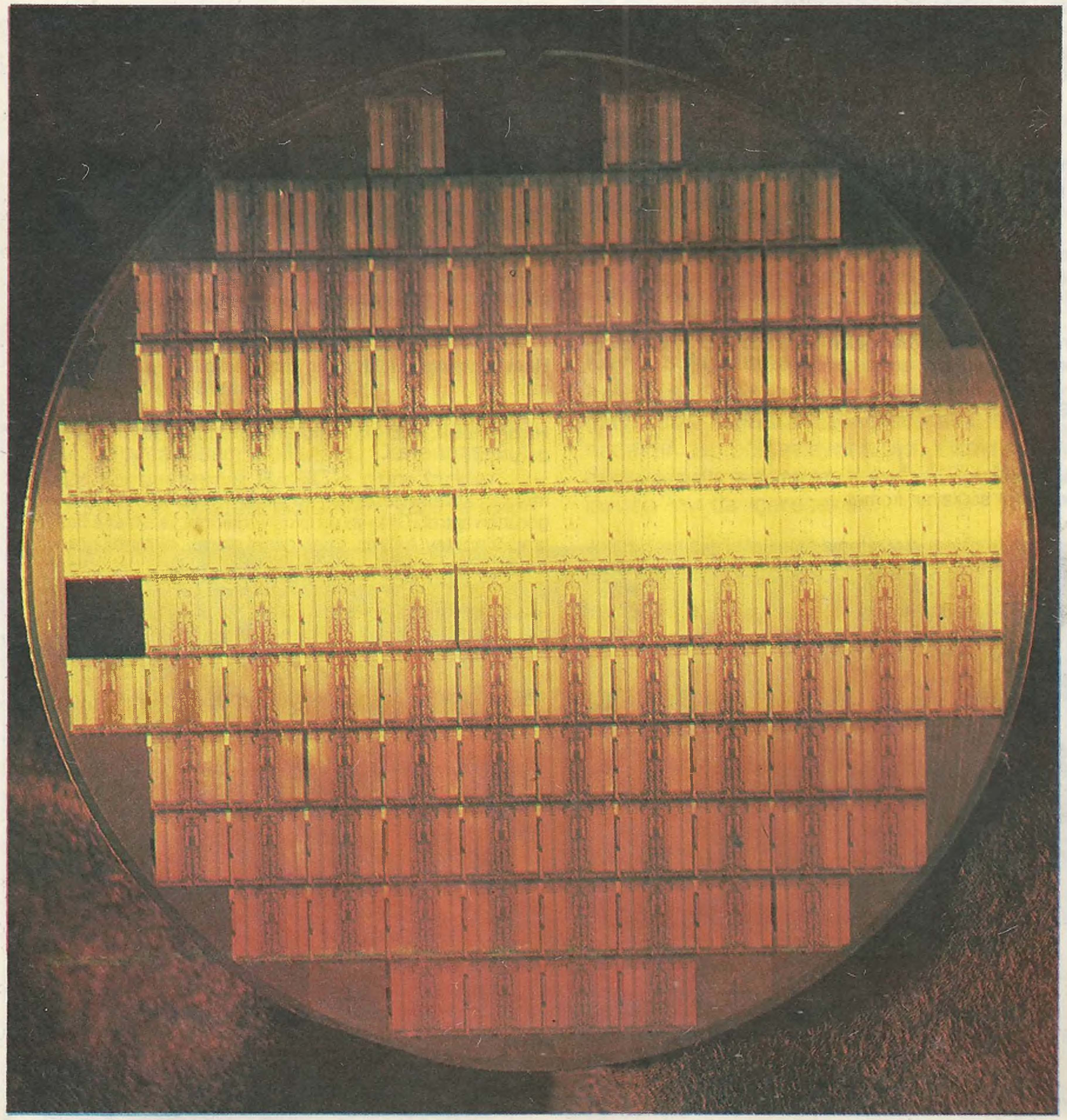
Taşıt: İşaretleri (verileri, komutları vb.) ileten elektrik devresi birimi.

Tümleşik Devre: Özel bir elektrik devresi. Hacminin çok küçük olmasına karşın çok sayıda bileşen içerir ve çok sayıda karmaşık ve güç işlemi kısa zamanda yapabilir.

Yukarıdan-Aşağı: Sorunların çözümlenmesinde ve programlamada kullanılan bir yöntem. Sorunu birçok altsoruna indirgeyecek şekilde bir model oluşturmayı içerir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYAR



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bilgisayar: Büyük Bilinmeyen

Ortaya çıkmalarından bu yana çok kısa bir süre geçmiş olmasına rağmen, bilgisayarlar artık iyiden iyiye tanıdık oldular. Bu aşamada geri dönüp, ilk bilgisayarların doğuşunu hatırlatmakta yarar var. Modern bilgisayarların ataları, dört işlemli hesap makinelerinin yardımıyla, çok sayıda hesabı kısa sürede yapabiliyorlardı. Ancak yalnızca araştırmacılar tarafından kullanıldıklarından halkın çoğunluğu varlıklarından bile habersizdi.

Gerek elektronik teknolojideki ilerlemeler, gerek endüstrinin çok sayıda hesabın üst üste ve güvenilir bir biçimde yapılmasına duyduğu ihtiyaç, bilgisayarların gelişip yaygınlaşmalarına yol açtı.

İlk makinelerin başlıca işlevi, zeka gerektirmemekle birlikte, yine de hesap yapmaktan daha karmaşık olan birtakım işleri yürütmekti. Bu makineler *bilgi işleyebiliyorlardı;* bu yüzden *bilgi-işlem makineleri* diye anılmaya başlandılar. Bugün her iki deyim de, yani, gerek ''bilgisayar'' gerekse ''bilgi-işlem makinesi'', eşanlamlı olarak kullanılıyor.

Bilgisayarlarla hesap makineleri arasındaki temel fark, bilgisayarların ''öğrenme'' yetisine sahip olmalarıdır. Bilgisayar, birtakım işlemlerin nasıl yapılacağını ''öğrenebilir'', yani ''programlanabilir''.

Ancak ilk bilgisayarların öğrenme yetileri tamamen edilgindi; makine öğrenmeye kendinden hiçbir şey katmıyordu. Bu tip öğrenmeyi, (yani programlanmayı) koşullu reflekslere benzetebiliriz.

Bir kütüphanede konusu ya da yazarı bilinen bir kitap, bilgisayar yardımıyla, çok kısa bir sürede bulunabilir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Örneğin, köpeğimizi her gün gazete kulübesine gidip, bir gazete alıp getirmesi için eğittiğimizde, köpek bütün bu işleri zekanın hiç katkısı olmaksızın, koşullu refleksleri sayesinde yapacaktır. Bu yüzden gazetecinin kapalı olması ya da gazetenin henüz dağıtılmamış olması gibi önceden koşullandırılmamış olduğu durumlarda, köpek ne yapacağını bilemeyecektir.

Aynı şey bilgisayarların eğitilmeleri (yani programlanmaları) için de geçerlidir. Bilgisayarlar programlandıkları her işlevi yerine getirebilirler; ancak program tarafından öngörülmemiş bir durumla karşılaştıklarında, işe insanın karışması gerekir.

Ancak kimi araştırmacılar, gelecekte bir tür ''bilinç''e sahip makinelerin yapılabileceğini iddia ediyorlar. Bugün için bu olasılık daha çok ''bilim-kurgusal'' bir düşten ibarettir. Yine de bu yolda, kendi bozuk devrelerini saptayıp yalıtarak yedek devreleriyle işlerini sürdürebilen ''deneme-yanılmalı-bilgisayarlar''ın yapılmasıyla ilk adım atılmıştır.

''Elektronik kalem'' adlı gereç. Bununla bilgiyi doğrudan doğruya ekran üzerinden okumak mümkündür.

Bilgisayar teknolojisindeki inanılmaz gelişmeler, bazen bu çok karmaşık makineleri ancak ''mürit''lerinin anlayabileceği izlenimini doğuruyor. Oysa bilgisayarlar, özellikle de kişisel bilgisayarlar gibi küçük sistemler, herkes tarafından kullanılabilir. Bu bakımdan bilgisayarların gündelik yaşamımızda hesap makinelerinin doğal bir uzantısı olduğu düşünülebilir. Nasıl ki öğrenciler artık hesaplarını bir hesap makinesiyle yapıp, zamanlarının daha büyük bir bölümünü yaratıcı faaliyetlere ayırıyorlarsa, bundan böyle, sıkıcı ve tekdüze işlerimizin çoğunu bilgisayarlara devrederek düşünmeye daha çok vakit ayırabiliriz.

Örneğin, matematik ödevi yapan bir öğrencinin önce problemi kavraması, sonra da hesaplara koyulması gerekir. İşin bu son kısmının, problemin anlaşılması ve sonucuyla pek bir ilgisi yoktur. Dolayısıyla bir makineye bırakılabilir.

Diyelim ki öğrenci iki kenarını bildiği bir dik açılı üçgenin hipotenüsünü bulmaya çalışıyor. Aşağıdaki bağıntıyı veren Pitagor teoremini kullanmak zorundadır:

$$\text{a) Hipotenüs} = \sqrt{(1.\ \text{kenar})^2 + (2.\ \text{kenar})^2}$$

Eğer kenarlar 8 (birinci kenar) ve 4 (ikinci kenar) değerindeyse, sayıları yerlerine koyarak elde edeceğimiz değer:

$$\text{b) Hipotenüs} = \sqrt{8^2+4^2} = \sqrt{64+16}$$

$$= \sqrt{80} \cong 8.9$$

olur. Hipotenüsü bulup soruyu cevaplamak için öğrenci bir sayının karekökünü bulmak zorundadır. Bundan önceki kuşak bu sonucu tablolardan okuyordu; bugünün öğrencisi hesap makinesiyle elde ediyor. Yarının öğrencisi ise sonucu bilgisayara soracak. Neden böyle olacağına bir bakalım:

Problemin çözümü (a) formülündedir; bunu izleyen (b) adımı ise sayısal değerleri bulmak için atılır. Birinci adımı atabilen öğrenci problemi doğru olarak kavradığını ve sonuçlandırabileceğini kanıtlamıştır. Gerisi sıkıcı ve pek

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Kimi yüksek okullarda ''öğretmen'' bilgisayar öğrencilere teker teker kişisel ders veriyor.

fazla da anlamı olmayan bir iştir, dolayısıyla makineye devredilebilir.

Şu beş komutun verilebìleceği bir aygıt olsaydı:

''ilk kenarın değerini al ve karesini hesapla (8^2)'';
''ikinci kenarın değerini al ve hesapla (4^2)'';
''bu iki kareyi topla (64 + 16)'';
''çıkan sonucun karekökünü al ($\sqrt{80}$)'';
''bana cevabı ver (8.9)'';

bu durumda öğrenci, birinci aşamadan sonra ödevini yapmış olmakla kalmayacak, Pitagor teoremini gerektiren bütün problemler için de bir çözüm hazırlamış olacaktı.

Veriler

İşte bilgisayar tam da böyle komutların verilebileceği bir aygıttır. Programlayarak, ona Pitagor teoreminin nasıl çözüleceğini ''öğretebiliriz''. Bizim görevimiz, yalnızca ilk defaya mahsus olmak üzere, ona hangi hesapların nasıl yapılacağı hakkında bütün bilgileri sağlamaktır. Ondan sonra, artık makine yapacağı işlemleri biliyor olacağı için, yalnızca kenarların değerlerini vermemiz, sonucu hızla elde etmek için yeterli olacaktır.

Herhangi bir işlemi (örneğin toplama işlemini) yaparken, iki değeri bilir, sonucu öğrenmek isteriz. Toplanacak sayılar, işlemin verileridir.

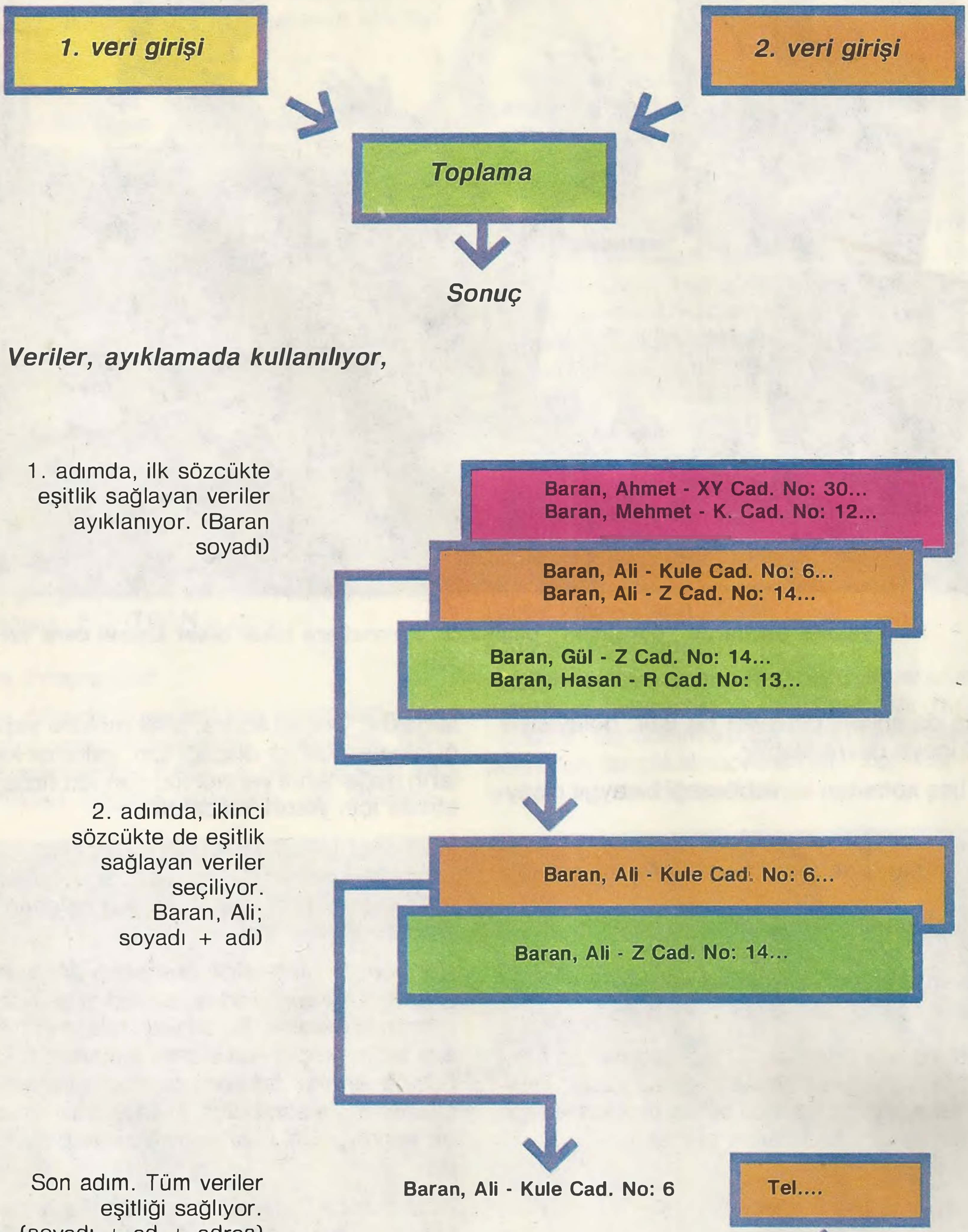
Rehberden bir telefon numarası ararken ise, aradığımız kişinin adını, soyadını ve adresini bilmemiz gerekir. Bu bilgileri telefon numarasını bulmak için veri olarak kullanırız. Bu durumda arama faaliyeti telefon numarasının okunmasıyla son bulur. Aslında bu numara da bir veridir; çünkü zaten rehberde bulunuyordu.

İlk durumda (toplama işlemi), daha önce varolmayan bir değeri (sonucu) elde ediyoruz; ikincide ise (arama), rehberde zaten varolan bir değeri (telefon numarasını) bulup çıkarıyoruz. Verilerin bu iki kullanılış biçimi s.16'da şema olarak verilmiştir. Bilgisayarlarla da aynı mantık kullanılabilir. Bir işlemin çözülebilmesi için problemin tüm sayısal değerlerinin kay-

Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYAR VERİLERİNİN KULLANIMI

Veriler bir sonuç elde etmede kullanılıyor,



Veriler, ayıklamada kullanılıyor,

1. adımda, ilk sözcükte eşitlik sağlayan veriler ayıklanıyor. (Baran soyadı)

2. adımda, ikinci sözcükte de eşitlik sağlayan veriler seçiliyor. Baran, Ali; soyadı + adı)

Son adım. Tüm veriler eşitliği sağlıyor. (soyadı + ad + adres)

Son bilgiden aranan veri parçası elde ediliyor. (Telefon numarası)

Ücretsiz PDF Kitap Arsivi - https://rantey.org

dedilmesi gerekir; bir aramanın sonuçlandırılması için ise, sonucun bulunmasını sağlayacak tüm verilerin bilinmesi gerekir.
Birisiyle konuşurken yaptığımız, bir dizi veri aracılığı ile bir bilgi değiş tokuşunda bulunmaktan ibarettir. Eğer bir olayı anlatıyorsak, aktardığımız veriler belleğimizden kaynaklanıyor demektir; yok eğer bir fikrimizi açıklıyorsak, o zaman belli bir akıl yürütmenin sonucunda ulaştığımız verileri aktarıyoruz demektir.
Ancak, her iki durumda da ayrıntıların iyice bilinmesi gerekir. *Aynı şekilde bilgisayarın da bir cevap verebilmesi için tüm verilere sahip olması gerekir.*
Kaynayan bir su kabına baktığımızı düşünelim; eğer dokunursak bizi yakacağı sonucunu çıkarıyoruz. Bu mantıksal çıkarımı mümkün kılan, daha önce bize bu tehlikeden bahsedilmiş ve bizim bu uyarıyı depolamış olmamız olabilir. Bunun üzerine aynı durumla yeniden karşılaştığımızda bu uyarıyı anımsıyoruz. İzlenen mantık, iki veri (kaynayan su ve yanma tehlikesi) arasında bir ilişki kurmaktan ibarettir.
Aynı mantık bilgisayara da uygulanabilir. Önce tüm veriler ve ilişki kurma kuralları bilgisayara verilir. Daha sonra, yeni durum ortaya çıktığında (örneğin kullanıcı bir soru yönelttiğinde), makine, tüm olası yanıtlar arasından doğrusunu ayıklar ve bize iletir.
Ancak, beynimizin veri edinme olanakları, bize söylenenleri depolamakla sınırlı değildir; deneyimlerimizle de öğreniriz. Kaynayan su örneğine dönelim. Bize bir uyarıda bulunulmamış da olsa, bu bilgiyi yaşayarak edinebilirdik. Su kabına ilk dokunduğumuzda alıcılarımız (sinir uçları) beynimize acı duygusunu iletir, o da tehlikeyi kaydederdi. Gelecekte benzeri bir durumla karşılaştığımızda da, ilk olayın anımsanması davranışımızı belirlerdi.
Yeni veriler geldikçe, durmaksızın bizi zenginleştiren bu öğrenme tarzı, bilgisayarlara da uygulanmaktadır. Bu uygulama henüz başlangıç aşamasında olmakla birlikte, gelecek için umut veren sonuçlara ulaşıyor.
Gözden geçirdiğimiz örneklerde iki düşünce biçimi ile karşılaştık:

- gerekirci yaklaşım
- olasılık yaklaşımı

Gerekirci yaklaşımla çalışan bir bilgisayar kendisine verilmiş çok sayıda veri arasında, yine kendisine verilmiş olan kurallara göre ilişkiler kurar. Bu yüzden, bu yaklaşım, bütün durumların öngörülebilmesini gerektirir.

Ergonomik (yani iş koşulları biliminin elde ettiği sonuçlara uygun olarak tasarlanmış) klavyeli bir kişisel bilgisayar.

Olasılık yaklaşımında ise mantık daha karmaşıktır. Bu yüzden, bir örnekle açıklamak yerinde olur.
Diyelim ki bir bilgisayara satranç oynamasını öğretmek istiyoruz. Gerekirci yaklaşımdan yararlandığımız takdirde, mümkün her durum için belleğe bir hamle depolamamız gerekirdi. Bu iş çok zaman alacağı gibi, mutlaka eksikliklere de yol açardı.
Diğer olanak ise olasılık yaklaşımını kullanarak, makineye taşların hareketini ve uyulması gereken kuralları verdikten sonra bol bol satranç oynatmaktır.
Başlangıçta rasgele yapılmış bir hamle başarılı olduğunda, bilgisayar o andaki durumu kaydeder ve o duruma yol açmış olan hamleye olumlu bir puan verir. Sonuç başarısız ise, durumu gelecekte kaçınılması gereken durumlar sınıfına sokar ve hamleye olumsuz bir puan verir.
Başlangıçta makine bütün oyunları kaybedecek, ancak karşılaştığı durumların sayısı ile birlikte deneyimi de artacaktır. Bu durumda onu yenmek giderek zorlaşacaktır. İşte bir bilgisayar böyle öğrenir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bilgisayarlar Düşünebilir mi?

Amerikalı bilim adamı Robert Jastrow ''The Enchanted Loom'' (Tılsımlı Tezgah) adlı kitabında bilgisayarın ''düşünme'' yeteneği ile ilgili bazı kuramlar geliştirmiş ve geleceğe yönelik ilginç kestirimlerde bulunmuştu. Yazar insan beyninin özellikleriyle bilgisayarı karşılaştırıyordu:

Pek çoğumuz aritmetik ve genelde matematik hesaplar konusunda bilgisayarın üstünlüğünü kabul etmekle birlikte, bilgisayarın bundan fazlasını da yapabileceğini kabule pek yanaşmayız. Nasıl olur da bir makine gerçekten düşünür? Sağduyuya aykırı bir şey bu! IBM için çalışan A.L.Samuel'in raporu bunun pekala olabileceğini düşündürüyor. Bilgisayara dama öğretmeyi üstlenen Samuel, böylelikle doğru bir biçimde hazırlanan bilgisayarın, insanınkine şaşırtıcı biçimde benzeyen bir düşünce tarzına sahip olabileceğini kanıtlamaya çalışmış.

A.L.Samuel, önce bilgisayara dama oyununun kurallarını vermiş. Böylece daha ilk adımda bilgisayar bir üstünlük sağlıyor, çünkü insanlar kuralları doğru olarak uygulamayı ancak deneyimle öğrenebilirler. Ayrıca Samuel bilgisayara bir de, oyun tahtasında kendi durumunu rakibininki ile karşılaştırarak değerlendirmesini sağlayan bir formül vermiş. Bu değerlendirme nasıl yapılıyordu?

Makine, önce tek tek her taş için, o taşın rakip bir taşı alabilip alamayacağını, damaya çıkabilip çıkamayacağını ve o taşın yitirilme olasılığını hesaplıyordu. Formül, makineye taş alma ve damaya çıkma olanağı yaratan hamleleri tercih etmesini, buna karşılık kendi taşlarını tehlikeye sokacak durumlardan kaçınmasını salık veriyordu. Dahası, formül makineye, üstün olduğu durumlarda taş değiştirmeye gitmesini, aksi takdirde bundan kaçınması gerektiğini de öğretiyordu; normal olarak insanların da yapacağı budur. Son olarak, makine oyunun akışı içindeki genel durumunu da bir sayıyla belirtiyordu; sayı daha yüksekse kendi durumu da daha güçlü demekti. Makinede bu formül, insanda tecrübenin oynadığı rolü oynuyordu. İster maliyeci olsun, ister bir komutan, ister işletme yöneticisi, çeşitli seçenekler arasında bir karar vermek durumunda olan herkes, sorununu zaten makinenin dama problemine yaklaştığı biçimde ele alır.

Samuel, oyunun temel stratejisini öğrettikten sonra, makineyi, oynadıkça taktiğini geliştirecek biçimde programladı. Bu, makinenin formülünün de deneyime bağlı olarak değiştirilmesini içeriyordu. Böylelikle makinenin, ustasını, yani kendisine ilk programı ve formülü vermiş olan programcıyı da yenebilmesi sağlanacaktı.

Makine, hamlesinin yararını, mevcut durumu, rakibinin muhtemel hamlesinin ortaya çıkaracağı durumla karşılaştırarak değerlendiriyordu. Tabiî rakip, makinenin öngördüğü hamleyi yapmayabilirdi. Böyle bir durumda, rakibin hamlesinden sonra bir karşılaştırma daha yapılıyor ve makine, formülünün hangi ögelerine daha fazla (ya da daha az) ağırlık vermesi gerektiğine karar veriyordu. Formüldeki ağırlıklar her hamleden sonra yenileniyor ve formülün değişmesiyle makinenin de oyun stratejisi gelişmiş oluyordu. Makine de oynadıkça ustalaşıyordu, tıpkı insanlar gibi.

Bir bilgisayarla nasıl oynanır? Taşlar oyun tahtasına yerleştirilir ve yapılan her hamleyle yerleri değiştirilir. Ya da oyun tahtası ve taşları bir ekranda görünür. Bütün bunlar bilgisayara karşı oynayan oyuncu için gereklidir; makine gerekli bütün bilgiyi elektronik belleğinde saklar.

Oyuncular hamlelerine başlarlar. İnsan, hamlelerini bir klavye kullanarak makineye bildirir. Makine ise kendi hamlesini ya hamlenin kısa simgelerini televizyon ekranına yansıtarak ya da kâğıt üzerine yazarak bildirir. Sürekli olarak formülüne başvurduğundan, makine de her hamleyi aynen bir insan gibi hazırlar. Oyunun gidişini ve her hamlenin yararlarını mümkün olduğunca önceden görmeye çalışır. Bu sırada, yine aynen bir insanın yaptığı gibi, rakibinin hamlesinin de olası hamleler içinde en iyisi olacağını varsayar.

Bilgisayarın oyun tarzı, tek hamlenin bütün mümkün sonuçlarını tasarlamaya çalışan bir aceminin oyununa çok benzer: ''Ben şu taşı oynarsam, o şunu yapacaktır. Diğerini oynarsam, o...''

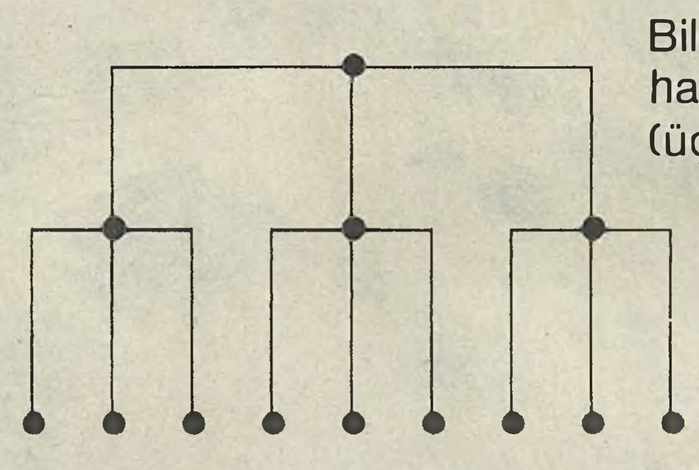
'Deneyimli oyuncu ile deneyimsiz oyuncu arasındaki fark, birincisinin oyundaki birçok durum ve rakibinin olası hamlelerini zaten bilmesidir. Bu yüzden belli bir hamlenin olası sonuçlarını, ara hamleleri hesaplamaya gerek duymaksızın görebilir.

Bu yüzden oyunu dışardan izleyen biri, deneyimli oyuncunun, oyun hakkında neredeyse bir içgüdüsü olduğunu düşünecektir. Ama as-

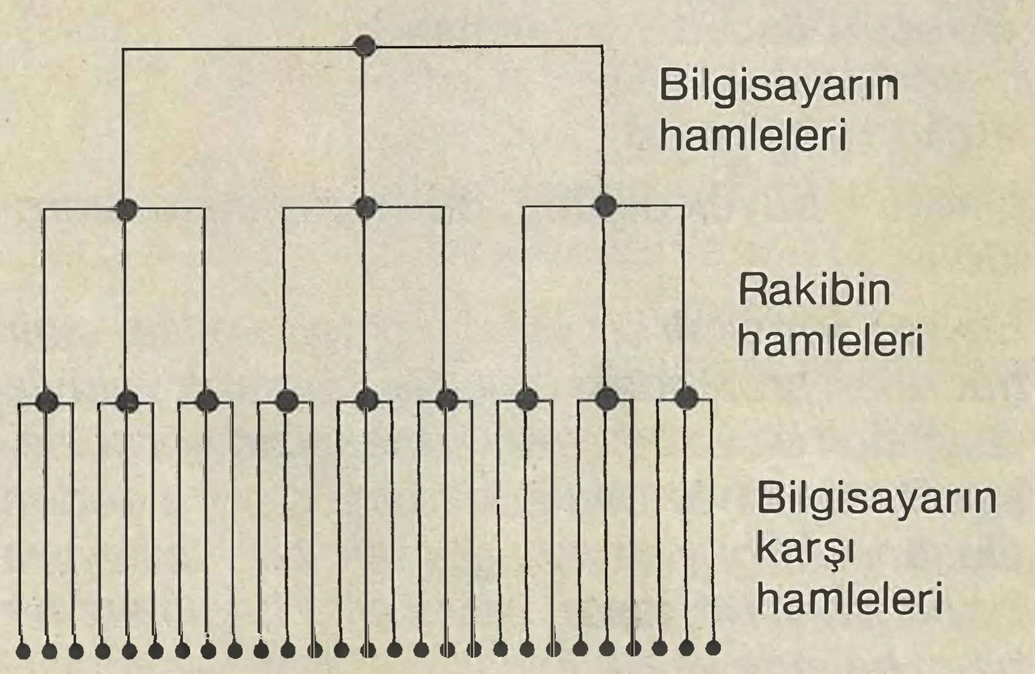
Ücretsiz PDF Kitap Arsivi - https://rantey.org

lında o da hamlelerini tıpkı bir bilgisayarın yaptığı gibi hazırlamaktadır, ama hesapları bilinçsiz bir düzeyde gerçekleşir.

Bilgisayar nasıl ''düşünür''? Diyelim ki, oyunun herhangi bir aşamasında bilgisayarın önünde üç seçenek olsun. Makine bu seçenekleri önce keşfeder, her birine bir puan verir, sonra da rakibinin karşı hamlelerini bulmaya çalışır. Diyelim ki bilgisayarın her hamlesi için rakibin de üç karşı hamlesi olsun. Bilgisayar bu durumda dokuz hamleyi de değerlendirecektir. Aşağıdaki şekil, oyunun bu aşamasında bilgisayarın değerlendireceği seçenekleri göstermektedir. Her dallanma bir seçeneğe tekabül eder. Oluşan çizelgeye ''karar ağacı'' denir.



Bilgisayarın önünde şimdi dokuz olasılık durmaktadır. Elektronik ''beyni'' sayesinde hepsini birden gözden geçirir ve her biri için karşı hamle tasarlamaya başlar. Bu, ağaca yeni dallanmaların eklenmesi demektir. Şimdi dokuz olasılığın her biri için üç tane daha olasılığın eklenmesiyle ''dal''ların adedi 27'ye çıkmış olur.



Bu işlemin bundan sonraki aşaması 81 olasılığı içerecek; bu da makinenin aynı anda 81 dama tahtasını birden gözden geçirmesi anlamına geliyor. Çağdaş bir bilgisayar için bile zor olmaya başlayan bu durum, hele Samuel'in 1950'lerde kullandığı eski modeller için büsbütün olanaksızdı. Bu yüzden Samuel'in programı bu noktada duruyordu.

Bugünkü makineler bin kat daha hızlı çalışıyor, daha çok sayıda hamleyi hesaplıyor ve çok daha iyi dama oynayabiliyorlar. Aynı ilkeler kullanılarak bilgisayarlara, çok daha karmaşık bir oyun olan satranç öğretilmiş ve oldukça iyi sonuçlar alınmıştır.

Satrancın ince hesap gerektiren yapısı, bir kez oyunu anladıktan sonra, bilgisayarın zekasına neredeyse insanca bir nitelik kazandırıyor. Dünyanın en iyi 500 satranç ustası arasında sayılan İskoçyalı David Levy, dünyanın en iyi ''yapay'' oyuncusu kabul edilen bir bilgisayarla karşılaşma yaptıktan sonra, ''insan bilgisayarı insan gibi görmeye başlıyor, hele hele ne yapmaya çalıştığınızı anlayıp, çok ustaca bir hamleyle önlediği zaman'' dedi. Kimi uzmanlar 1990 yılının satranç şampiyonunun bir insan olmayacağı görüşündeler.

Tabiî satranç çok soyut bir oyun. Satranca uygun bir mantığı olan bilgisayar, yine de insan düşüncesinin en yüksek aşaması olan yaratıcılıktan yoksun. Genç bir mezun gibi, üstün yeteneklere sahip olmasına karşın, akıllı ya da bilge olduğu söylenemiyor. Ama insan beyninin bilgisayarlar karşısındaki bu üstünlüğü de son teknik gelişmeler tarafından sorgulanmaya başlandı.

Bu yeni gelişmeler tümleşik devrelerce içerilebilen bileşenlerin sayısındaki inanılmaz artışa bağlı. Tümleşik devreler insan beynindeki nöronları andırıyor. Çok sayıda tümleşik devre birbirlerine bağlanarak modern ''elektronik beyni'' oluşturuyor. 1970'lerin başında, bir yonga çok az sayıda transistör ve başka elektronik bileşen içeriyordu. Bu yüzden ya ''düşünüyor'' ya ''anımsıyor'', ama ikisini birden yapamıyordu. Bugün önde gelen Amerikan ve Japon üreticileri elektronik bileşenleri bir tümleşik devrede öyle biraraya getirebiliyorlar ki, ilk defasında, aynı yonga hem bilgi işleyen bir devre hem de bellek birimleri içerebiliyor.

Bu yüzden artık tek bir devre aynı anda hem ''anımsayabilir'' hem de mantık yürütüp karar verebilir, yani ''düşünebilir''.

Aynı devrenin hem veri depolayabilmesi hem de depoladığı bu veriler üzerinde işlem yapabilmesi, bilgisayarın kapasitesi bakımından çok önemlidir. Çünkü bu sayede bilgisayar içersindeki bir bellek devresi, hem çevresindeki diğer devrelere komut gönderecek, hem de kendisine gönderilen bilgiyi saklayacak biçimde bağlanabilir. Böylelikle, herhangi bir bilgisayar bellek birimine, yalnızca kendisinde de-

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Kişisel bilgisayar satranç oynuyor. Oyunun zorluk derecesi önceden saptanabilir.

polanmış bilgileri değil, başka herhangi bir devrede depolanmış bilgileri sağlaması için de komut gönderilebilir. Diyelim ki bilgisayardan birinin adresini vermesini istedik. Komutun doğrudan doğruya uyardığı bellek birimleri, talebi, bağlı bulundukları ve o kişi hakkında herhangi bir bilgi içeren bütün devrelere iletеceklerdir. Yani bellek birimleri, kullanıcıya, kendi depoladıklarından daha fazla bilgi sunabilirler.

Böylelikle, insan zekasının en önemli bileşenlerinden biri olan ve çağrışımsal düşünce niteliği taşıyan bellek türüne yaklaşmış oluyoruz. Beyin herhangi bir bilgi talebine olağanüstü bir malzeme yığını ile desteklenen bir yanıt verir. Bu malzeme, geçmiş deneyimler sonucunda birbirleriyle ilişkilendirilmiş sayısız unsurdan oluşur. İnsanoğlu, yaşamının zenginliğini, büyük ölçüde, bu çağrışımlara borçludur.

Böyle çağrışımlı bir yanıtın oluşumu ise, sinir hücreleri arasındaki binlerce bağlantı yoluyla ve bilinçli bir çaba gerektirmeksizin gerçekleşir. Birbirleriyle bağlantılı binlerce devreden oluşan bir bilgisayar, gerçek bir ''silisyum beyin'' olur ve insan beyni gibi bağımsız bir birim halinde ve ''bilinçsiz'' olarak işleyebilirdi.

Tümleşik devreler teknolojisindeki son ilerlemeler, yaratıcı düşüncenin bir başka türünü daha gerçekleştirilebilir şeyler kapsamına soktu. Bilgisayarın ''düşüncesi'', hayal gücüne yer vermeyecek ölçüde mekaniktir. Bilgisayar devreleri arasındaki bağlantılar, insan beynindekilerle kıyaslandığında çok daha basittir. Bilgisayarın her bir devresi, makinenin diğer par-

Ücretsiz PDF Kitap Arsivi - https://rantey.org

çalarına iki ya da üç telle bağlıdır. Oysa mikroskopik bir beyin hücresi, onbinlerce ''tel'' ile, yani sinir lifi ile beynin diğer bölümlerine bağlıdır. Bir hücreden diğerine olan sayısız bağlantı, beynin diğer yapısal özellikleriyle birlikte ele alındığında organımızın olağanüstü gücünü açıklıyor.

Kuramsal olarak, bu tür bağlantıları içeren bir bilgisayarın yıllar önce yapılmış olması gerekirdi. Ancak böyle bir bilgisayar için yüzmilyarlarca bağlantı teli gerekir. Dolayısıyla pratik ve teknolojik kısıtlar böyle bir bilgisayarın yapılmasını olanaksız kılıyordu.

Ne var ki, tümleşik devrelerle birlikte durum değişti. Mikroskopik bağlantılar zaten yapının içinde yer aldığından artık kablolara gerek duyulmuyor. Basit bir teknolojik ilerleme gibi görünen bu buluş, tersine, bilgisayarların evriminde temelli bir aşamadır. Çünkü bu buluş, işleyişi insan beyninkini andıran makinelerin yapımını mümkün kılmıştır. Bugünden kestirilebildiği kadarıyla böylesi makinelerin 1990'larda gerçekleşeceği düşünülüyor. Birçok bakımdan insan zekasıyla yarışacak, aklın birçok özelliğine sahip olacaktır; dış dünyaya karşı bir duyarlık, deneyimle öğrenme yetisi, ilk karşılaşılan fikirleri kavrama gibi. Öyleyse artık karşımızda canlı yaratıklar mı var?

Çoğumuz, bir bilgisayarın duygu ve heyecanları olamayacağını, bu yüzden de canlı bir varlık sayılamayacağını söyleriz. Bilgisayar yemez, içmez, hareket etmez, büyümez et ve kemikten değil metal ve plastikten yapılmıştır. Ama bu özelliklerin çoğu bilgisayara kazandırılabilir. Örneğin pilleri boşaldığında bir gerilimölçer bilgisayara açlığını bildirebilir ve makinenin bir priz yönünde hareket ederek, bir lokma elektrikle kahvaltı etmesi programlanabilir. Gerçi birilerinin elektriği onun için üretmesi gerekecek; ama insanların büyük bir çoğunluğunun da besinini başka birileri sağlamıyor mu? Zaten gerektiğinde kendi elektriğini arayıp bulan bir bilgisayar yapmak da o kadar zor olmayacaktır.

Hatta, insan türünün yaşamasını sağlamak için, bir zamanlar Tabiat Ana'nın insan beynine eklediği gibi, duygu ve heyecanlar da, istendiği takdirde bilgisayara eklenebilir. Bu konu, bilgisayarın daha iyi bir öğrenci olup, daha hızlı öğrenmesini isteyen Samuel'in de aklına gelmişti. Makinesi başlangıç için fena oynamamakla birlikte, fazla dayanamıyordu, yarışmacı değildi. Üstün olduğu durumlarda, insanlar gibi, en kestirme yoldan zafere ulaşmaya çalışmıyor, tüm olası seçenekleri değerlendirmek uğruna zaman harcıyordu. Samuel onun psikolojik yapısını değiştirmeye karar verdi. Programı yeniden öyle düzenledi ki, makine üstünlüğün kendisinde olduğu durumlarda hırçın davranarak, kendisini zafere bir an önce ulaştıracak hamleleri seçmeye başladı. Üstünlük rakibe geçtiğinde ise oyalama taktikleri benimsiyor, oyunu uzatmaya çalışıyordu.

(Robert Jastrow'un THE ENCHANTED LOOM, ''Tılsımlı Tezgah'' adlı kitabından, 1981, alınmıştır.)

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Hesap Makinesinden Bilgisayara

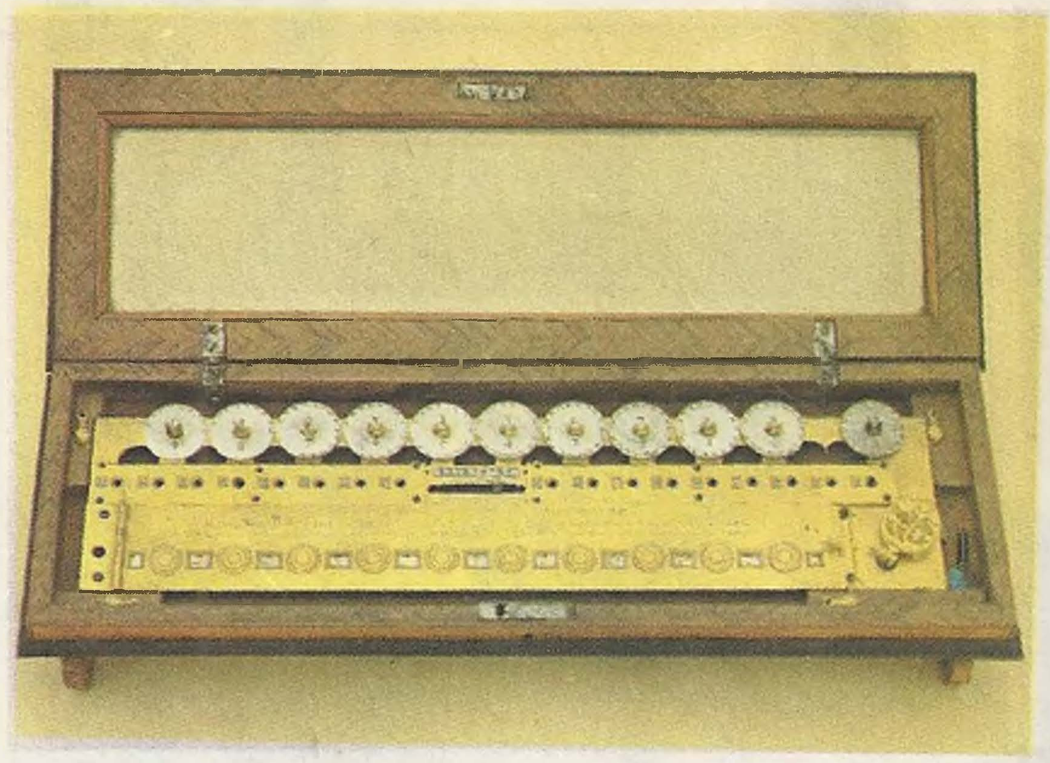
Hesap makinesi hesap işlemleri yapabilen ilk makine oldu. İşleyişi mekanikti ve yalnızca toplama ve çıkarma yapabiliyordu. Gelişimini sürdürdü, elektromanyetik oldu; artık bütün işlemleri yapabiliyordu.

Bu makinenin temel sınırlılıkları, her defasında yapması gereken işlemin yeni baştan bildirilmesi gereği ve bir iş yaparken ona komut verilememesiydi. Oysa bilgisayarlar ''biçimlendirilebilirler'', yani belli işleri yerine getirmek üzere programlanabilirler.

Kolayca anlaşılabilecek bir programlama örneğini çamaşır makinesinde görebiliriz. Bu aygıta, süreye ve suyun sıcaklığına bağlı olarak yıkama evrelerini ayarlayan bir mekanizma eklenebilir. Bu sayede, çeşitli yıkama biçimleri, yani çeşitli programlar seçilebilir (kısa ya da uzun süre; alçak ya da yüksek sıcaklık).

Bu aygıtta veriler, süre ve su sıcaklığıdır. Veriler seçilen programa göre işlenir ve sonuç olarak belli bir yıkama biçimi elde edilir.

Genel olarak programlanabilir bir makinenin işleyişi üç adım olarak betimlenebilir:



Aritmetik makinesi, 1666'da Samuel Morland tarafından yapıldı.

Geçmiş ve Şimdiki Durum

Zamanın büyük bir kısmını düşünmeye ayırabilmek için, uzun ve sıkıcı hesaplardan kurtulma isteği, Antik Çağlar'dan beri insanoğlunun zihnini kurcalamaktadır. Hesaplamayı kolaylaştırıcı ilk alet, boncuklu işlem tahtası (Abakus), M.Ö. VIII. yüzyılda Çin'de ortaya çıktı. Romalılar, içinde yüzüklerin hareket ettirilebildiği yatay oluklar döşenmiş işlem tahtaları kullanıyorlardı. Birinci oluk birler basamağını, ikincisi onlar basamağını vb. gösteriyordu.

İşlem tahtaları bugün hâlâ kullanılıyor, hatta bazılarında karekök bile almak mümkün. İşlem tahtaları uzun süre bilinen tek işlem yapma aracı olarak kaldılar. 1642'de Pascal dişli çarklardan oluşan bir hesap makinesi yaptı. Bu makinenin geliştirilmesiyle çarpma yapabilen (sayıları katlayabilen) ilk makine de mümkün oldu; bu makinede çarpma işlemi, ardarda yapılan toplama işlemlerinden oluşuyordu. Yaklaşık 1850'ye kadar bu makineler fazla bir pratik uygulaması olmayan cesur adımlar olarak kaldı. Bu tarihlerde basit ve güvenilir toplama makineleri geliştirilerek uygulamada kullanılmaya başlandı.

19. yüzyılın sonuna doğru bölme yapabilen ilk makine icad edildi. Pitagor teoremine dayanıyor ve sonucu çizelgelerden arıyordu. Bu makinenin de pratik bir uygulaması olmadı.

Bütün bu makineler mekanik hesaplamanın temelini oluşturdukları halde aslında pratik olarak tamamen yararsızdılar; yalnızca tek bir işlem yapabiliyorlardı ve bellek gibi bir birimden de yoksundular.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Birçok işlem yapma olanağına sahip ve bir bellekle donatılmış ilk makine İngiltere'de Charles Babbage adlı bir matematikçi tarafından yapıldı. Babbage'in getirdiği çalışma ilkeleri doğrultusunda daha sonra daha gelişmiş makineler yapıldı.

1929'da, ABD'de bilimsel hesaplar yapmaya uygun ilk aygıt gerçekleştirildi. Ancak işleyişi hâlâ mekanikti.

Gelişimin bundan sonraki aşaması hareketlerin elektromanyetik enerji ile gerçekleştirilmesi oldu. Bu yenilik 1944'de MARK I'in yapımına yol açtı. Saniyede 10 işlem gibi o zaman için olağanüstü bir hızı vardı. MARK I'in boyutları da olağanüstüydü; 18 metre uzunluk, 2.5 metre yükseklik! Bundan hepsi hepsi üç yıl sonra 1947 yılında ilk elektronik tüplü makine yapıldı. Gerçi işlemler artık elektronik devrelerle yapılıyordu ama ilkeler hâlâ Babbage'in ilkeleriydi.

İlk program fikri 1950'lere doğru ortaya atıldı; makineye hesaplarla ilgili tüm komutlar veriliyor ve bundan sonra makine, kullanıcının müdahalesi olmaksızın işlemleri yapıyordu. İlk bilgisayar doğmuştu.

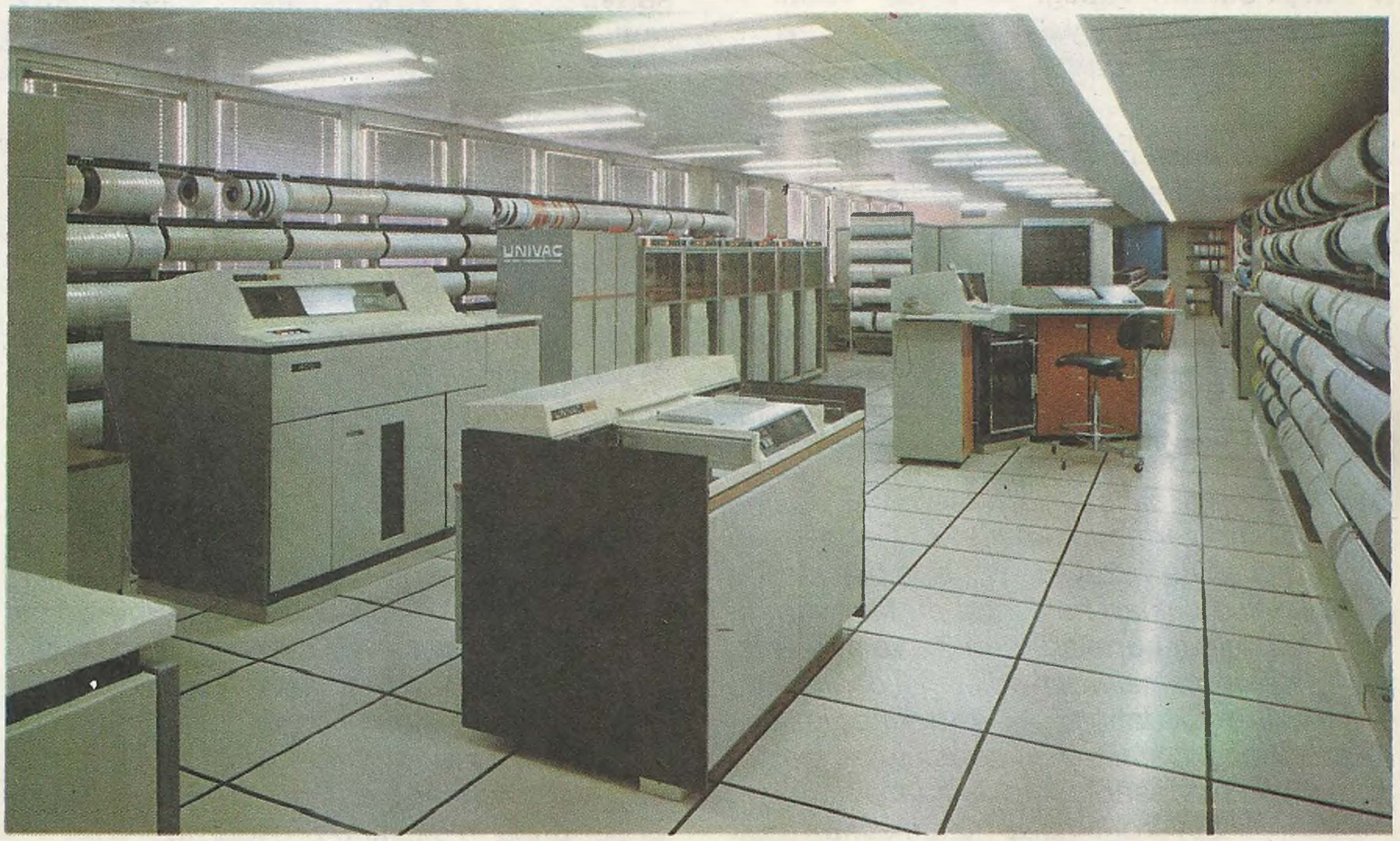
Bunu izleyen yıllarda bu sektörde başka hiçbir bilim dalında raslanmayan başdöndürücü bir gelişme yaşandı.

Elektronik devreler yerlerini daha az yer kaplayan, daha az gerilim gerektiren ve daha az ısı yayan transistörlere bıraktı. Bundan sonraki aşama ise tümleşik devreler oldu (1965-1970 arası). Bilgisayarların boyutları küçüldü, hızları olağanüstü boyutlara ulaştı. MARK I'in saniyede 10 işlemine karşılık, bugünkü bilgisayarlar aynı sürede 1.000.000 işlem yapıyorlar.

1970'den bu yana devrelerin ve bileşenlerinin boyutlarının küçültülmesinde sayısız ilerlemeler kaydedildi. Bu da kişisel bilgisayarların doğmasına yol açtı. Aşağıdaki tablo bu evrimin temel aşamalarını özetlemektedir:

M.Ö. VIII. YY.	İşlem tahtası
1642	İlk toplama makinesi
1850	Bellekli toplama makinelerinin ilk kullanışı
1889	Artık bölme yapılabiliyor.
1929	Bilimsel uygulamalar; klavye
1944	MARK I. Elektromanyetik mekanizmalar
1950	Programlama
1955	Transistör
1970'ler	Tümleşik Devreler
1980'ler	Bilgisayarların yaygın kullanımı

Büyük kapasiteli bilgisayarlardan oluşan modern bir bilgi işlem merkezi.



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Sayı Dokuyucu: Bir Buluşun Hikayesi

18. yüzyıl sona ermek üzereydi. Gerçi ekonomik ve politik karışıklıklar açısından bereketli bir yüzyıl olmuştu ama otomatik hesap makineleri alanında bir hareket yoktu. Ta ki yüzyılın son on yılı içinde Charles Babbage İngiltere'de doğuncaya kadar. Gerçi genellikle yeni bir bilimsel icadın ilk kez ne zaman, kimin tarafından gerçekleştirildiğini saptamak zordur, ama, Babage bugün herkes tarafından hesap makinesinin mucidi kabul ediliyor.

Çağının birçok bilgini gibi ona da büyük bir miras kalmış ve bu mirası yaşamının yapıtı olarak gördüğü bir tasarı için harcamıştı.

19. yüzyılın başında insanlar yeni işkolları ve meslekler yaratıyorlardı. Geçimlerini hesapkitaptan ve bilgi kaydetmekten sağlayanların sayısı artıyordu. En çetin ve hatalara yol açan işlerden biri de Logaritma tablolarının kullanılmasıydı. Babbage bu tablolarda sürekli olarak basit birtakım yanlışlıklar buluyordu. Bu tekdüze ve yıpratıcı işlemlerle uğraşırken düşündüğü şeylerden biri de zaman israfıydı. Babbage bu işi bir makinenin yapması gerektiğini düşünerek ilk taslağını hazırladı.

1821'de çalışmasından emin olarak Krallık Astronomi Derneği'ne başvurdu ve yaptığı prototipi sundu. Yaptığı gösteride makinesinin ''farklar (diferansiyel) yöntemi''ne göre çalıştığını anlatarak inandırıcı açıklamalarda bulundu.

Makinenin nasıl çalıştığını açıklamayacağız; ardışık farkları hesaplayarak yüksek dereceden denklemleri çözdüğünü söylemek yeter. Bunu derneğin 1822'deki bir toplantısında gösterdi; çalışmaları o kadar ilgi uyandırmıştı ki, dernek ona ''Matematik Tablolarının Hazırlanmasında Hesap Makinesinin Kullanımı Üzerine Sir H.Davy'ye Mektup'' *adlı yazısından ötürü bir altın madalya verdi.*

Babbage yeni bir şevkle hesap makinesine son şeklini vermeye koyuldu. Babbage'in planlarına şöyle yüzeysel bir bakış bile, ne kadar cüretli oldukları hakkında bir fikir vermeye yeter. İşlem kuralları, çarklar ve mekanizma, önceki makinelerden çok farklı değildi, ancak makine çok daha büyük ve karmaşıktı. Dahası Babbage, sınırsız iyimserliğiyle çizelgelerin yalnızca hesaplanmakla kalmayıp basılmalarını da istiyordu. İngiliz hükümetine 1.500 sterlinlik bir yardım için başvurdu ve bu parayla kendi toprakları üzerinde bir atölye kurdu, becerikli işçiler buldu ve çalışmaya başladı.

En önemli iş, özel torna tezgâhlarında yüzlerce mil, çark ve vidanın özenle hazırlanmasıydı. Bunlar, farklar makinesinin iskeletini oluşturacaktı ama ilk büyük aksaklık da bu aşamada çıktı. Astronomi Derneği için yapılan küçük model yalnızca bir prototipti. Parçalardaki bir yapım hatası sistemde dengesizliğe yol açıyor, ancak sistemi durduracak kadar olumsuz bir etki yaratmıyordu. Ama asıl farklar makinesinde, en küçük sapmalar birbirlerine eklenerek hatalara ve tıkanıklıklara yol açıyordu.

Babbage, üretimin daha hatasız olması için işçilerine daha dikkatli çalışmalarını söyledi; biraz düzelme görüldü, ancak sistemin bütününe yansımadı. İlk başarısızlıklarına aldırmayan Babbage, çağının makinelerinden verebileceklerinden fazlasını istediğinin farkına varmayarak inat etti.

Hükümetten gitgide daha fazla para istedi; 17.000 sterline kadar isteği karşılandı, ancak bu meblağdan ötesi için ''Bu kadarı da fazla'' dendi ve tasarı 1833 yılında askıya alındı.

Eğer Babbage makul biri olsaydı, ki değildi, bu noktada dururdu. Çünkü makineyi mümkün kılacak teknolojik olanaklar bakımından çağın bir yüzyıl kadar önündeydi. Ama yorulmak bilmez zekası onu daha da büyük bir tasarıya götürdü. Ve işte o anda bilgisayar kavramı doğdu. Farklar makinesinin kendinden önce gelenlere karşı bariz üstünlükleri olsa da, yine de sadece tek bir görevi yerine getirebiliyordu: yüksek dereceden denklemleri çözmek. Sonuçlar, sırası önceden belirlenmiş bir dizi işlemle elde ediliyordu. Bir grup çark bir diğerini harekete geçiriyor, bir kol bir ikincisini çekiyor ve bu böyle sürüp gidiyordu.

Tüm bunlar önceden belirlenmişti. Başka bir deyişle, bu makine bugün ''adanmış makine'' dediğimiz türden bir makineydi. Bir işleve adanmış, o işlev için üretilmişti.

İyice düşününce, Babbage, yanlış yolda olduğunu anladı. Bir tür işlemi gerçekleştirebilen bir makine diğer bütün işlemleri de yapabilir, diye düşündü. Bu fikrin meyvelerini, yüzyıl sonra, bu fikri matematiksel olarak kanıtlayan, bir başka İngiliz deha, Alan Turing topladı. Tek bir görevle sınırlandırılmamış bir makine yapmak niçin mümkün olmasındı? Böyle bir makine, kendisinden istenen bir dizi görevi istendiği anda ve biçimde yapabilirdi.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Kuşkusuz her işlevi görebilen bir makine, tek bir işleve adanmış makineye kıyasla hem daha karmaşık hem de daha pahalı olacaktı. Ama o nisbette de daha yararlı olacaktı.

Yeni makine, aynı giriş mekanizmasını ve aynı ortak yazıcıyı (yani çıkışı) paylaşan birçok makinenin biraraya getirilmesi gibi düşünülebilirdi.

Babbage ise, herkesin aklına gelen ilk fikri reddedip, olağanüstü içgüdüsüne uydu. Tasarısı, makinenin aynı birimden değişik biçimlerde yararlanması olanağını sağlıyordu. Belirli bir işle karşılaşıldığında, tek bir sıra izlenecek ve her değişik iş, makinenin belirli bir parçasına değil, belirli bir işleyiş modeline bağlı olacaktı.

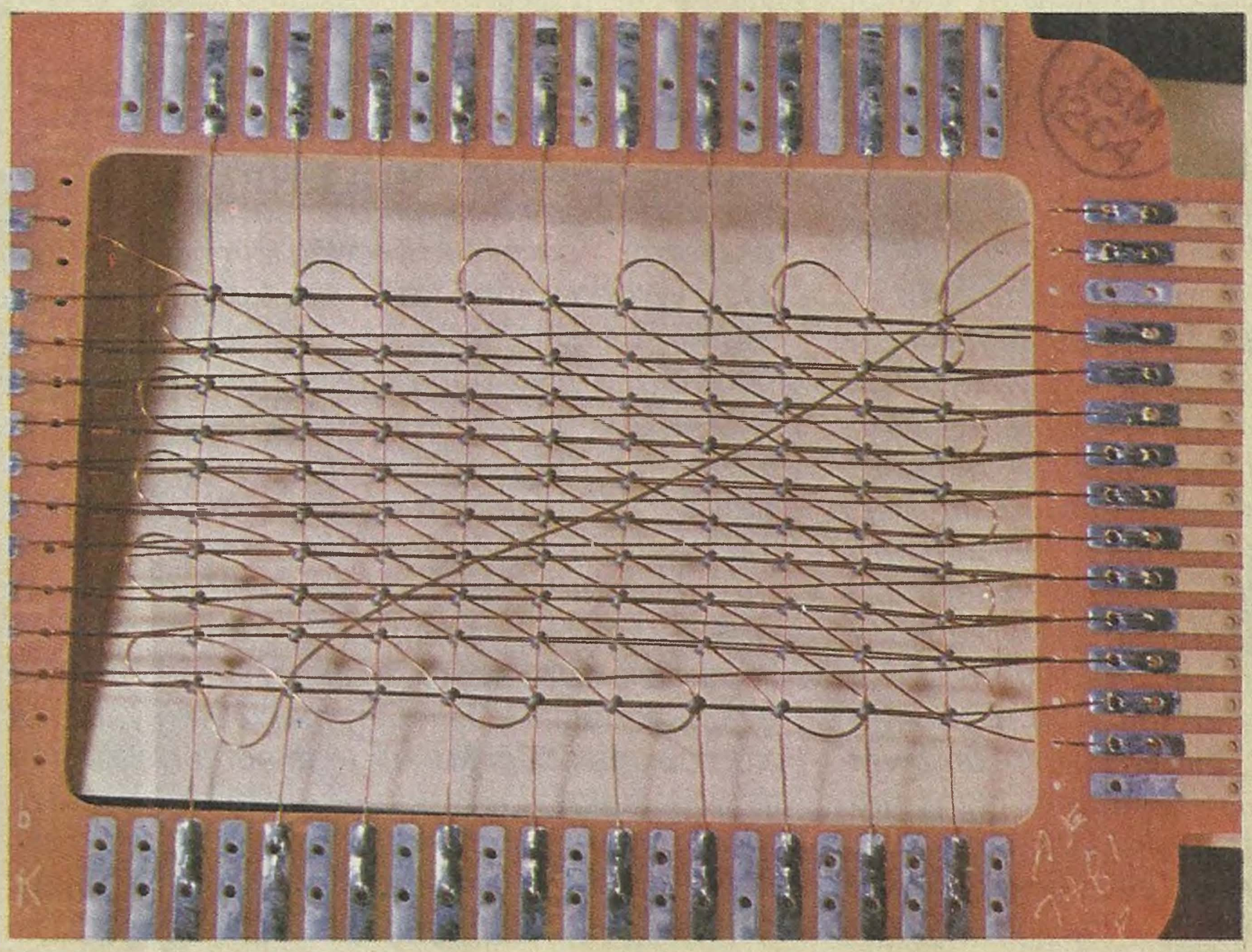
Bu aşamada makineye bütün bu ''faaliyetler kümesi''nden belirli bir anda hangisini seçeceğinin nasıl haber verileceği (ya da ''söyleneceği'') henüz belirlenememişti.

Babbage bu aygıta ''çözümleyici makine'' adını verdi. Burada, çoktan beridir ''programlanabilir hesap makinesi''nden söz ettiğini de belirtelim. Makineye sayı ve komut iletimini mümkün kılan bir giriş mekanizması vardı. Ayrıca bir aritmetik işlem birimi gereken hesapları yapıyordu. Babbage bu birime ''değirmen'' adını verdi. Üçüncü olarak da makinenin belirli bir anda tek bir iş yapmasını ve bütün hesapların gereken sırada yapılmasını sağlayan bir denetim birimi vardı. Sayıları gerektikleri anda bulup çıkaran bir depo, ya da bellek, dördüncü ana birimi oluşturuyordu. Son olarak da, bir çıkış mekanizması vardı.

Saydığımız bu beş bölüm, eski ya da yeni bütün bilgisayarların ana birimlerini oluşturur. Babbage'in hesap ya da bellek aygıtları, inanılmaz bir kol ve eklem ormanı ile birbirine bağlı on silindirik çarktan oluşuyordu.

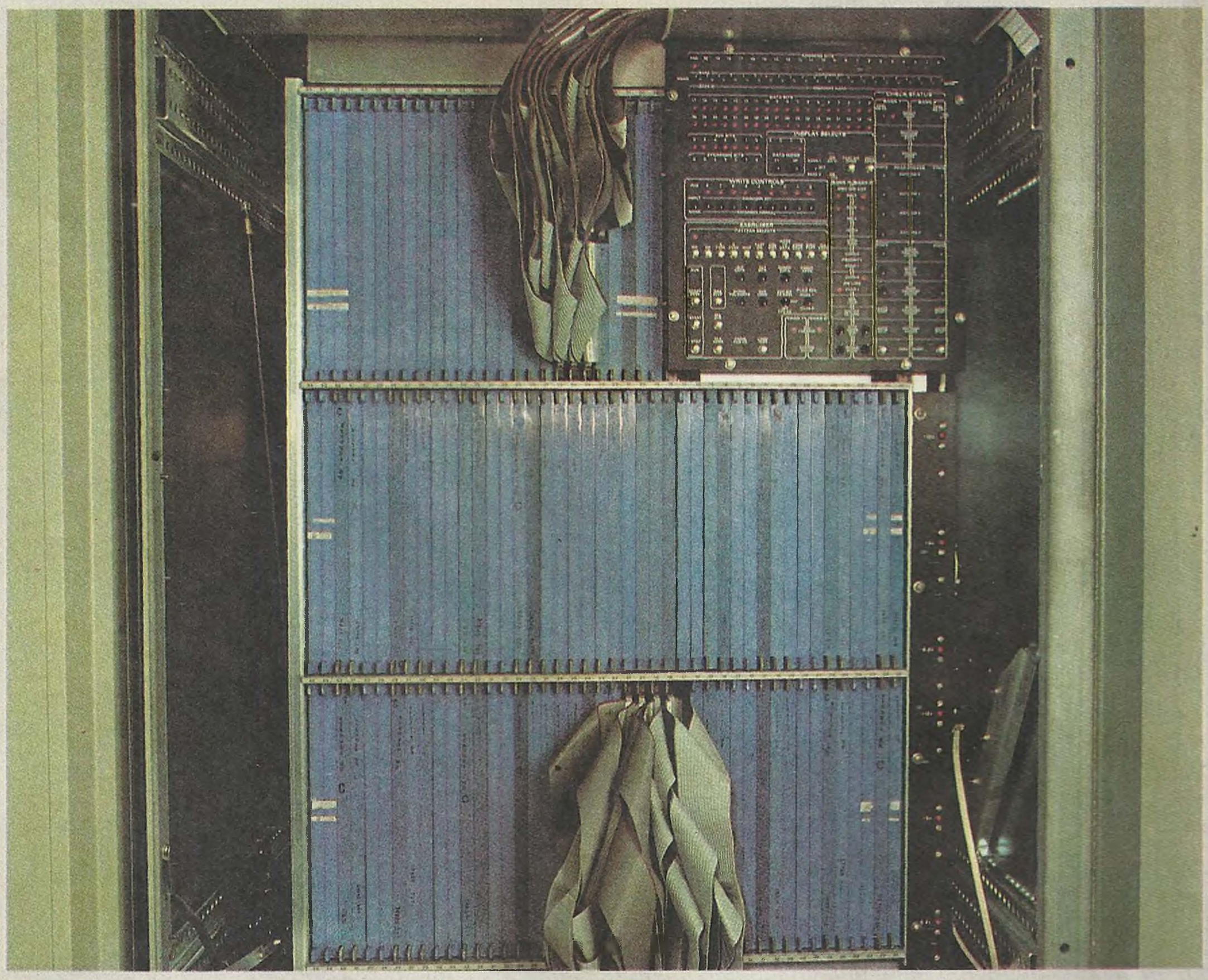
Farklar makinesinin bu prototipi, elle idare ediliyordu. Dişli çark düzeni bir kolu harekete geçiriyor, istenen hesaplar sonuçlandırıldığında ise bir zil çalıyordu. Bunun üzerine yeni bir şey iletmek mümkün oluyordu. Zil yeniden çalıncaya kadar kol yine hareket ettiriliyor ve bu böyle sürüp gidiyordu. Bu işlemlerin saçmalığının farkında olan Babbage, buharlı makine ilkelerini kullanarak çözümleyici makineye son biçimini verdi.

Denetim birimlerinin komutlarını yüklemek ve programlamak için Babbage, Joseph Jacquard adlı bir Fransız'ın buluşundan esinlendi. Jacquard, dokumacıların ince iş gerektirmekle birlikte sürekli tekrarlanan bir iş yaptıklarını, dolayısıyla bu işin otomatikleştirilebileceğini düşünmüştü. Bunun üzerine üzerinde delikler bulunan sert bir karton icad etmişti. Dokuma sırasında birtakım şişler yün zincirine rehberlik ediyordu. Delikli kartonun işlevi de, dokuma sırasında bu şişlerden bazılarını engellerken, bazılarının da deliklerinden geçmesine izin vermekti. Mekiğin her geçişinde,



İlk elektronik bellekler, manyetik çekirdeklerden oluşuyordu. Burada bellek birimi ferrit bir yüzüktü. Verilerin yazımı ve adreslerin seçimi kablolar tarafından iletilen darbelerle gerçekleştiriliyordu.

Ücretsiz PDF Kitap Arsivi - https://rantey.org



Bilgisayarların bellekleri gelişiyor: Modern bir bilgisayarda bellek birbirlerine yassı kabiolarla bağlı bellek kartlarından oluşur.

belli ölçülere göre delinmiş karton şişlerin yoluna çıkıyor ve bu şekilde kumaşın desenini yönlendiriyordu. Babbage, makinesindeki işlemlerin sırasını denetlemek için bu sistemden başka bir biçimde yararlanabileceğini anladı.

Dokuma benzetmesi, Lovelace kontesi Ada'nın da gözüne çarpmıştı; Babbage'in makinesi hakkında bir yorumunda şunları yazıyordu: ''Çözümleyici makine, cebir işlemlerini, aynı Jacquard'ın makinesinin yapraklar ve çiçekleri dokuduğu gibi dokuyor.'' Bu kadın, olağanüstü bir matematikçi olmanın yanı sıra, eşsiz bir güzelliğe sahipti. O zamandan kalan bir resim, onun büyük ölçüde Elizabeth Taylor'a benzediğini ve babası Lord Byron'un güzel hatlarını taşıdığını gösteriyor. Ada, Babbage ile tanıştığında, onun girişimlerinin önemini kavradı ve çözümleyici makinenin planlarını yakından incelemeye başladı. Babbage'a duyduğu yakınlık, Ada'yı onun açıklamalarını dinlemeye ve onun öğrencisi olmaya sevketti. Zengindi, heyecanlıydı ama her şeyi kavrayabilmek için zamana ihtiyacı vardı.

Neyse ki kendisi gibi zengin ve anlayışlı bir insan olan kocası Ada'nın bilimsel ilgilerini destekliyordu. Ada'nın üç çocuğu için bakıcılar tutuldu ve genç anne kendisini tamamen Babbage'ın çalışmalarına hasretti.

O kadar ki, 28 yaşına vardığında Ada, ''Bay Babbage'in Çözümleyici Makinesi Üzerine Gözlemler'' başlıklı uzun bir yazı dizisi yayımladı. Bu gözlemler, Babbage'in girişimini ayrıntılarıyla bilmek isteyenler için son derece yararlıdır. Lady Lovelace, bu yazıda, bu tip bir makinenin yapılmasının ortaya çıkaracağı ve bugün bile çözülmemiş olan bazı felsefi sorunlara değinmiştir. Örneğin, makinenin yaratıcı olup olamayacağı konusunda şunları yazmıştır: ''Çözümleyici makine, ne kadar becerikli olursa olsun, bir şey yaratmak iddiasında değildir. Bizim programladığımız her şeyi yapabilir, ama hiçbir gerçeği ya da çözümsel iliş-

Ücretsiz PDF Kitap Arsivi - https://rantey.org

kiyi keşfedemez. Amacı, bizim zaten bildiğimiz bir şeyi ulaşılır kılmaktır.'' Bu zekice yorum, bir bilgisayarın düşünce kapasitesi hakkındaki ilk açıklama gibi gözükmektedir. Bir bilgisayar neye programlanmışsa, yalnızca onu yapabilir. Sonradan görüleceği gibi, bu soru çok ilginç olmakla birlikte, bir ölçüde ikinci derecede kalmaktadır. Ancak, bunu ortaya koyma onuru da, tamamen Lady Lovelace'a aittir.

Lady Lovelace'in yazılarının en önemli yanı, Babbage üzerindeki psikolojik etkileridir. O zamana kadar, ''Astronomi Derneği''ndeki başarısından bu yana, ne kimse tarafından takdir görmüş, ne de anlaşılmıştı. Oysa şimdi, bir matematik dehası sayılan Kontes, Babbage'ın kuramlarını matematiksel yönden incelemiş ve onlarda hiçbir kusur bulamamıştı. Babbage çok mutluydu. Gerçi çözümleyici makinenin çalışacağına kendisi zaten inanıyordu ama işte şimdi yanında buna inanan bir de kadın vardı. Şimdi bütün iş, söz konusu aygıtı yapmaya kalıyordu.

Bu arada İngiltere'de hükümet değişmişti. Yeni Maliye Bakanı, gerçekleşmesi artık olanaksız görünen bu tasarıyı finanse etmekten vazgeçti. Zaman geçtiği halde, farklar makinesi hâlâ eksik bir tekerlek ve kollar yığını; çözümleyici makine ise Lady Lovelace'ın bir dizi taslak, fikir ve notlarından ibaretti. İşler de kötü gitmekteydi. Ada Lovelace otuzaltı yaşında kanserden öldü.

Bu genç kadın kısa yaşamı boyunca yalnız Babbage'ın tasarılarının değerini görüp onu desteklemekle kalmamış, kendisi de çağının düşünsel sınırlarını aşan bilişimsel kavramlar geliştirmişti.

Örneğin, bugün programcılığın temel taşlarından birini oluşturan döngü kavramını ilk kez Ada Lovelace dile getirmişti. Bazen tek bir işlemi yapmak için bir dizi komut gerektiğinin farkına varan Ada, bu komutlar dizisinin bir koşullu önermeyle (ya da ''atla'' komutuyla) birleştirildiği takdirde birçok işlemin çok daha kısa zamanda yapılabileceğini gösterdi.

Ne yazık ki Ada Lovelace'ın düşünceleri de ölümünden ancak yıllar sonra uygulamaya konabildi. Günümüzün bilişim uzmanları Ada Lovelace'a olan şükran borçlarını, son zamanlarda geliştirilmiş en ileri program dillerinden birine Ada adını vererek ödediler.

Ada ölünce tek başına devam etmek zorunda kalan Babbage çok az sonuç elde edebildi. Hükümetler değişiyor, ama hiçbiri fikirlerine yakınlık göstermiyordu. Disraeli, alaycı bir yaklaşımla, makinenin bundan böyle, ancak kendisinin kaça malolduğunu hesaplayabileceğini yazıyordu...

Buna karşılık, bu sıralarda, başka matematikçi ve mühendisler, Babbage'in yayınlarını ve Lady Lovelace'in notlarını büyük bir ilgiyle karıştırıyorlardı. Bunların arasından biri, İsveçli mühendis George Scheutz,farklar makinesini kendine özgü bir biçimde inşa etti. Babbage'in tersine, Scheutz'un başarısı öyle yankılara yol açtı ki, 1855'teki bir Mühendislik Fuarı'nda ''ilk üretim modeli'' olarak sergilenmek üzere bir prototip yaptı. Bu tuhaf aygıtın etrafını alan müthiş kalabalığın içinde, artık yetmişüçüne gelmiş ve iyice yılmış olan Babbage de yer alıyordu. Scheutz'un makinesi hakkında fikri sorulduğunda, çok nazik cevaplar vererek Scheutz'u kutladı. Ancak bu sırada aklından geçenleri tahmin etmek hiç de zor değil.

Babbage 1871'de yetmişyedi yaşında öldü. Bir gün dünyayı değiştirecek böylesine devrimci projeler tasarlamış olduğu halde, derin bir düş kırıklığı içindeydi. Çağdaşları onu, hayalci de olsa, ayakları yere basmasa da, bir deha olarak kabul ettiler. Zekasının üstünlüğü o derece kabul görmüştü ki, ölümünden sonra, normal insan beyninden fiziksel olarak farklılıklar gösterip göstermediğini saptamak için beyni uzun bir incelemeye tabi tutuldu. Çağının en seçkin cerrahlarından Victor Horsely incelemeyi üstlendi. Artık susmuş olan milyonlarca nöronu didikledikten sonra, bu beynin o zamana kadar gördüğü pek çok beyinden bir farkı olmadığını söyledi. Buna rağmen, hiç kimse Babbage'in beynini yok etmeye kıyamadığından, her bir yarıküresini bir kavanozda sakladılar. Bugün bile, isterseniz Babbage'ın beynini Hunter Kraliyet Tıp Akademisi'nin müzesinde görebilirsiniz.

(Christopher Evans'ın THE MIGHTY MICRO ''Müthiş Mikro''sundan alınmıştır.)

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bilgisayarların Evrimi: Uygulamalar

1980'lerde, bilgisayar alanındaki ilerlemeler öyle boyutlara vardı ki, artık evrimlerini günden güne izlemek mümkün olmuyor. Bilimsel araştırmalar teknolojik buluşlara yol açarken, yeni teknolojik buluşlar da bilimsel araştırmaları hızlandırıyor. Artık gelişmenin hızı aylarla ve hatta günlerle ölçülüyor.

Bu gelişmelerin en önemli sonucu bilgisayarların boyutlarının küçülmesi ve fiyatlarının düşmesi oldu; bu sayede artık amatörler de bilgisayarlardan yararlanabiliyor.

Buna bağlı olarak da, toplum çok köklü bir değişim geçirmekte.

Sanayileşmiş dünyada, en büyük gereksinim bilgidir. Gerektiği anda yapılan bir bilgi alışverişi, bilimsel bir araştırmanın sonuçlandırılması, kârlı bir ekonomik seçim ya da en azından değerli bir zaman kazancı anlamına gelebilir. Bilgisayar, bu tür uygulamalar için ideal bir alettir. Bellek kapasitesi, kullanıcının kolaylıkla ulaşabileceği büyük miktarlarda bilgi biriktirme olanağı sağlar.

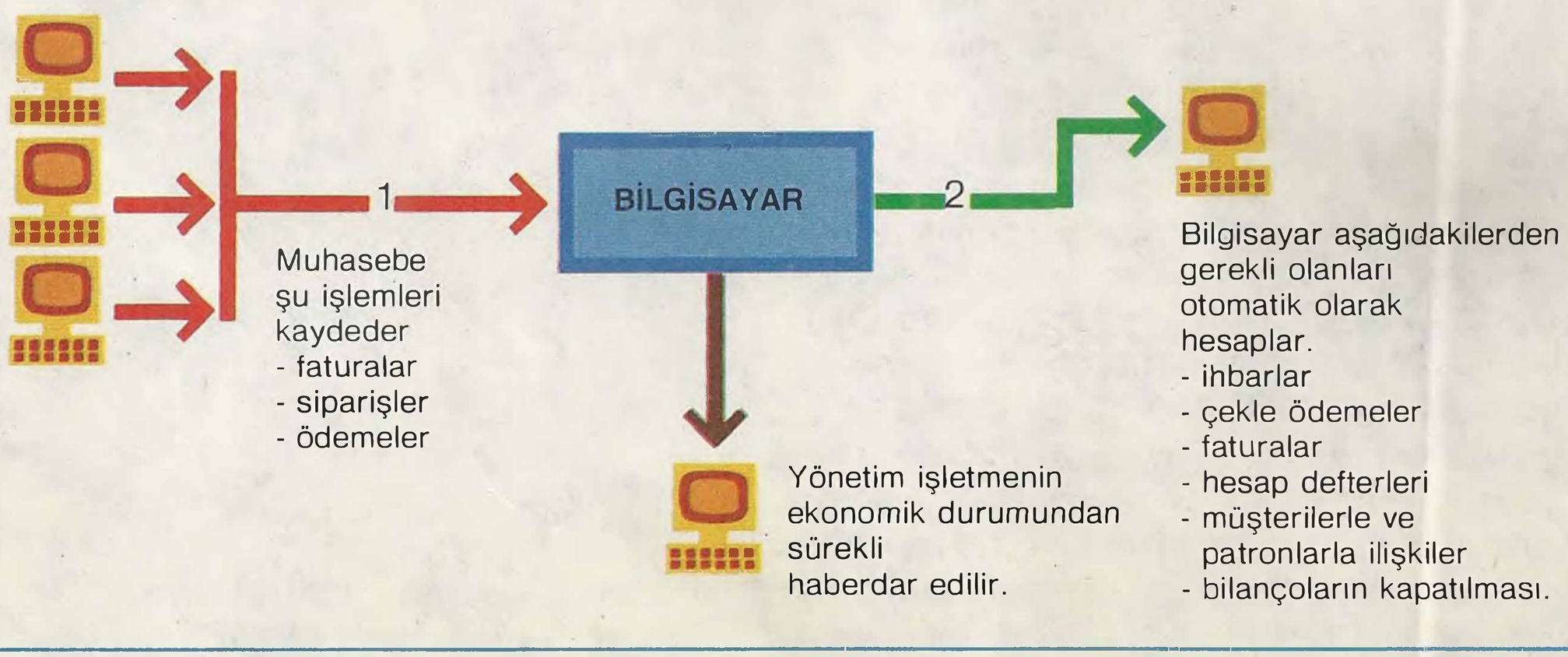
Çok sayıda örnek verilebilir. İlk olarak, bir kütüphanecilik örneği alalım. Elle yapılan bir kaynak taraması uzun ve sıkıcıdır. Oysa, kütüphane bir bilgisayarla donatılmışsa, bedensel bir çalışma gerekmeksizin bilgilere ulaşmak kolaylaşır. Böylelikle kullanıcı da, yaratıcı zamanını kendine ayırabilir.

Başka bir örnek alalım. Bütün işletmelerde, çeşitli bölümler, çalışmalarını yürütmek ve planlayabilmek için, tüm işletmenin durumunu bilmek zorundadırlar. Sözgelimi, ürünlerin stok durumu üretim bölümünü ilgilendirdiği kadar, dağıtımı da ilgilendirir. Bu karmaşık bilgilerin üstesinden bilgisayar kullanmadan gelinemez. Bu sayfalardaki şekiller, bilgisayarın bir işletmede hangi işleri görebileceği konusunda bir fikir vermektedir.

Bilgiyi, kullanmaya elverişli biçimlerde düzenleme tarzları ve yöntemlerini inceleyen bilim dalına, **bilişim** denir.

BİR İŞLETMEDE BİLGİSAYARIN BAŞLICA KULLANIM ALANLARI

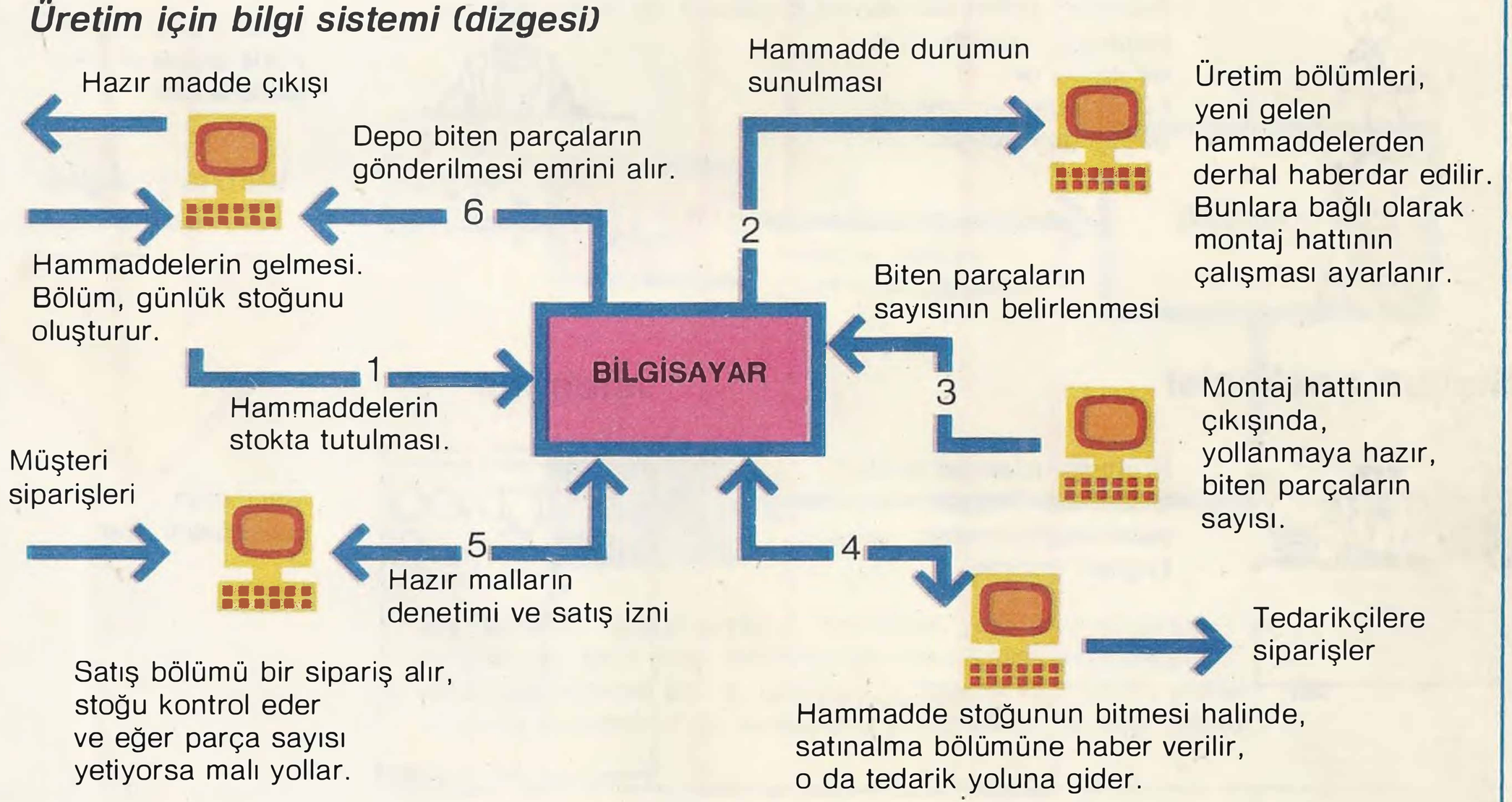
Muhasebe



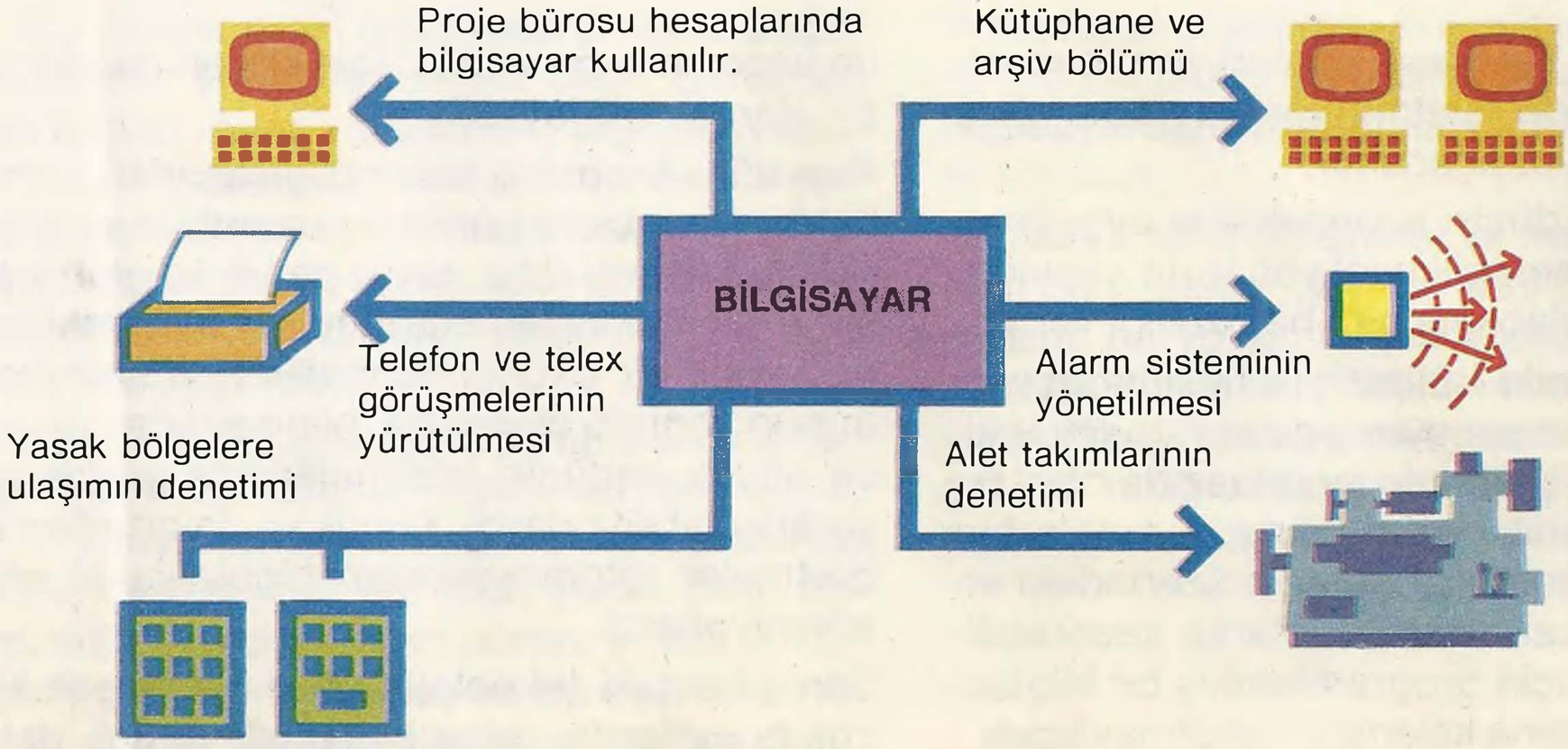
Ücretsiz PDF Kitap Arsivi - https://rantey.org

Personel yönetimi

Üretim için bilgi sistemi (dizgesi)



Özel uygulamalar



Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİR İŞLEM MERKEZİNİN YAPISI

Kullanıcı

Sistem mühendisi

Problemi bütün olarak inceleyip, çözümü için ne tip ve ne büyüklükte bir makine gerektiğini belirtir.

Sistem analizcisi

Problemi işlemsel açıdan çözümler. Programın yazılması için taban bilgileri hazırlar.

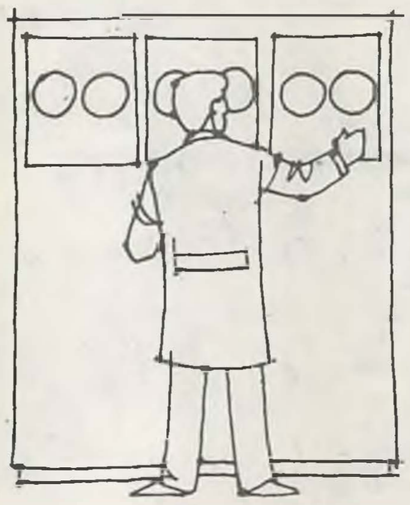
Programcı

Analistin bilgilerini program biçimine sokar.

Veri hazırlayıcı

Verileri bilgisayara yüklenebilecek bir şekle sokar.

İşletmen



Makinenin çalışmasını izler.

Bu bilimin çok karmaşık yönleri vardır ve yakın bir gelecekte işletme yönetiminde köklü değişikliklere yol açacaktır.

Bilişimin bir endüstri kompleksine uygulanabilen bu tamamen bilimsel yönünün yanında, evde de kullanılabilme gibi bir özelliği vardır.

Örneğin aylık aile bütçesinin hazırlanmasını düşünelim. Hepimiz aybaşında unutulmuş kimi harcama kalemlerinin ay ortasında nasıl bir paniğe yol açabileceğini biliriz. Ayrıca artan hayat pahalılığının aile bütçesi üzerindeki etkilerini de, ancak yaklaşık olarak kestirebiliriz. Oysa bu iş için programlanmış bir bilgisayar hiçbir harcama kalemini ''unutmayacağı'' gibi, enflasyon oranına bağlı olarak bütçemizde yapmamız gereken değişiklikleri de otomatik olarak ayarlayabilir.

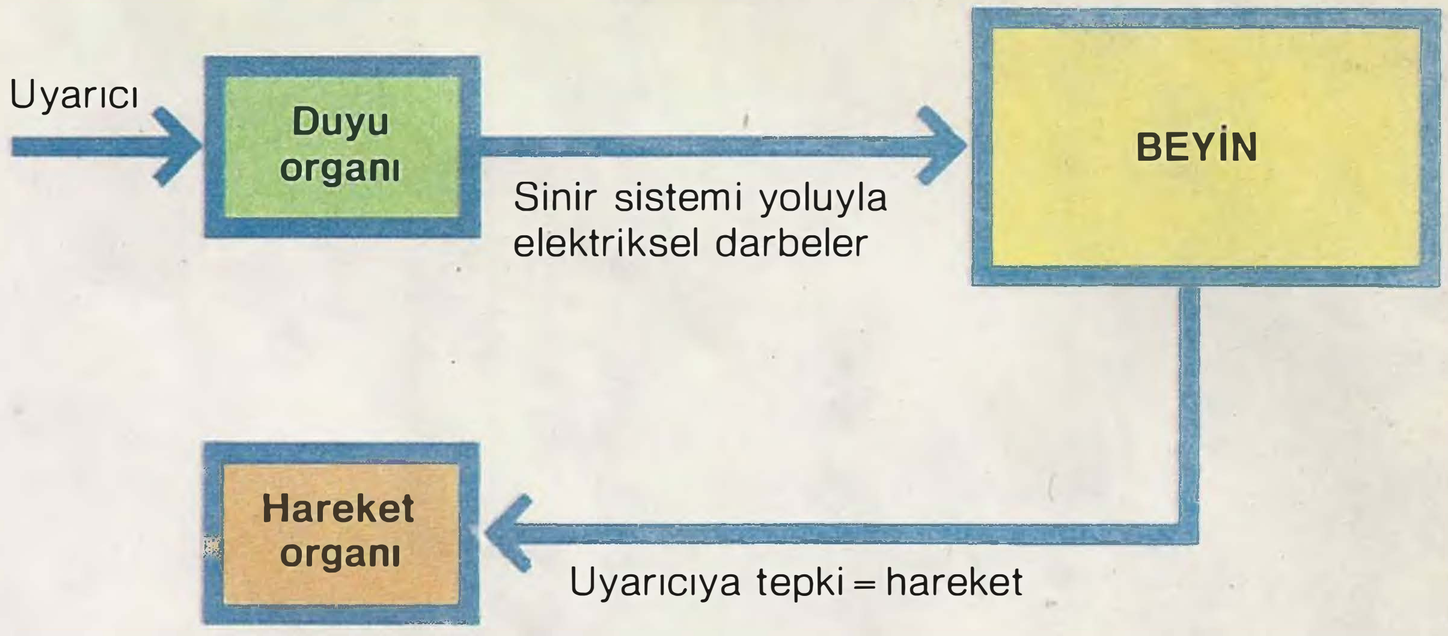
Kısa süre öncesine kadar bilgisayarlar, ancak her biri ayrı bir iş gören bir uzmanlar ekibinin kullanabileceği (bkz. şekil) pahalı ve girift aletlerdi. Bu makinelerin çalışması için gerekli yapı önemli bir ekonomik maliyet oluşturuyor, bunun sonucu olarak da, bilgisayarlar yalnızca büyük endüstri işletmelerinde gerçekten yararlı olabiliyorlardı. Küçük ya da orta ölçekli işletmeler, otomasyondan hiçbir yarar elde edemiyorlardı.

Son yıllardaki teknolojik ilerlemeler, çok küçük boyutlarda, daha kullanışlı ve çok daha ucuz bilgisayarların yapılmasına olanak ver-

Ücretsiz PDF Kitap Arsivi - https://rantey.org

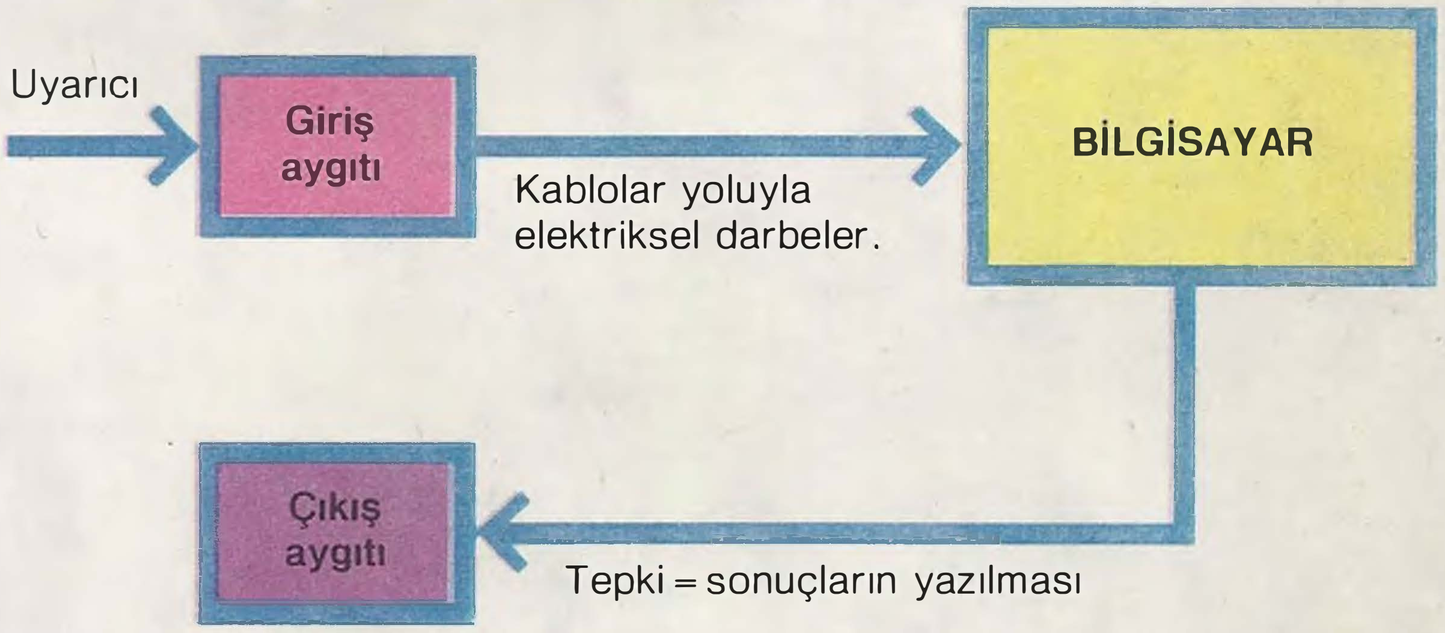
ETKİ-TEPKİ SÜRECİNİN MANTIKSAL ŞEMASI

Beyindeki etki-tepki süreci



Bir uyarı, örneğin bir böceğin sokması, bir dizi elektriksel darbenin iletilmesine yol açar. Beyin değerlendirir ve tepki olarak bir hareket buyurur.

bilgisayardaki etki-tepki süreci



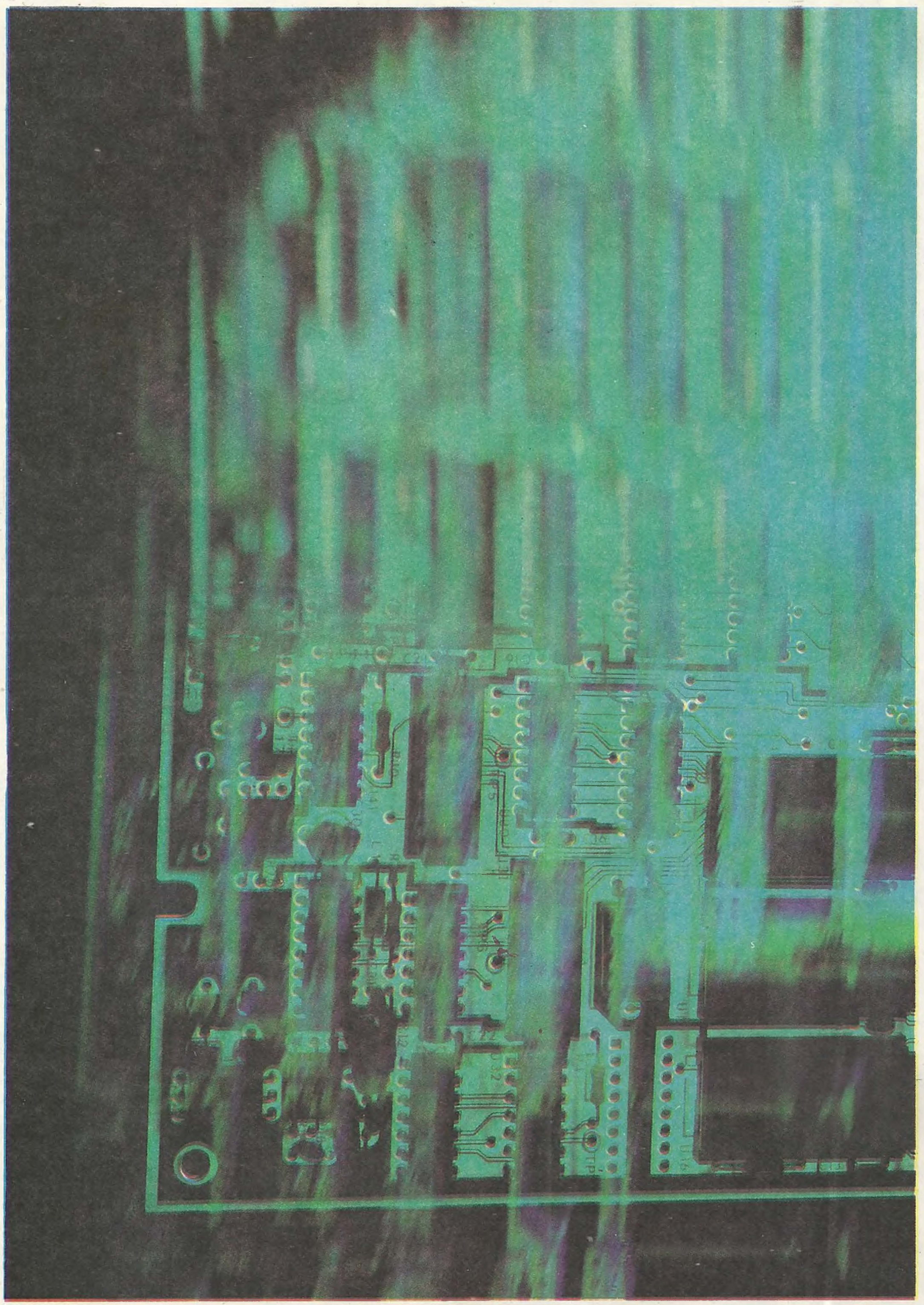
Bilgisayarda, uyarıcı birtakım işlemlerin yapılması yönündeki bir emirdir. Bu emir giriş mekanizmasında elektrik sinyallerine dönüştürüldükten sonra, gereken kablolarla bilgisayara yollanır. Bu sinyaller yorumlanır ve sonuçların yazılmasına yol açar (tepki).

di. Bundan sonraki gelişmelerin neler olacağını tahmin etmek hiç de zor değil. Şimdiden bazı bilgisayarlar konuşuyor, yazı okuyabiliyor, şekil çizebiliyor, satranç oynayabiliyor, çeviri yapabiliyor ve müzik çalabiliyorlar. Yakın gelecekte minik-işlemcilerle donatılmış bazı otomobiller sahiplerinin seslerini ayırdedip sözlerine uyabilecekler.

Ama belki de en ilginç gelişme tıp alanında olacak. İnsan vücudunun uyarıcıları almasıyla, bir bilgisayarın verileri kabul etmesi arasında önemli benzerlikler var (bkz. şekil). Sinir sisteminin temel birimi olan nöron, ancak iki durumda olabilir: Uyarılmış ya da uyarılmamış. Bu iki durum, sayısal bilgisayarların da çalışma temelini oluşturur. Dolayısıyla bu benzerlikten hareket ederek, sinir sistemimizin yapısı ve işleyişi hakkındaki bilimsel araştırmalarda bilgisayardan yararlanılabilir (yukarıdaki şekle bakınız).

Bilgisayarların gelişmesi ve insan vücudu ile olan benzerlikleri, ayrıca yeni bir bilimin doğmasına da yol açtı: Robotbilim (robotik). Bu alandaki buluşlar, pratikte insanın yaptığı tüm işleri yapabilen ve insan tepkileri gösterecek şekilde programlanabilen makinelerin yapılmasını olanaklı kılmıştır.

Ücretsiz PDF Kitap Arsivi - https://rantey.org



Baskı devresi, bütün elektronik makinelerin temel unsurlarından biridir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bilgisayar Mantığı

''Bilgisayar'' dediğimizde şu işlemleri belirli bir mantıki düzen içerisinde gerçekleştiren herhangi bir aygıtı kastediyoruz:

— Bilgilerin edinilmesi
— İşlenmesi
— Bellenmesi (biriktirme)
— Yayılması (iletilmesi, yayımlanması, vb...)

Bilgisayarların ille de elektronik olması gerekmez; örneğin mekanik ve hidrolik bilgi-işlem sistemleri olduğu gibi hava akımından bile yararlananlar vardır. Bu tür sistemlerde işlerin yürütülmesi, yani programın akışı, hava akımı ile ya da hidrolik yönlendirme ile sağlanır.

Bütün bu sistemlerde bilgi işlem, merkezi birim tarafından yürütülen programın görevidir. Bu birim de, bir işlem birimi, verilerin kıyaslanmasını sağlayan bir mantık birimi ve programın yerleştirildiği bir merkez bellekten oluşur.

Bir programın akışı, dizisel ya da mantıksal olabilir. Dizisel bir akışta, her işlev bir öncekinden sonra, bir sonrakinden de önce gerçekleştirilir.

Mantıksal bir akışta ise, normal sıra, belirli olayların gerçekleşmesine bağlı olarak değiştirilebilir. Bilgisayar dışı bazı örneklere bakalım: Üç renkli trafik lambaları dizisel bir biçimde çalışır; renkler hep aynı sırada birbirini izler.

Arabada vites değiştirmek mantıksal bir akıştır, çünkü bir durumdan (hızdan) diğerine geçmek zamana bağlı olmayan nedenlerle belirlenir (bkz. şekil).

Bilgisayarın yerine getirmesi gereken tüm işlevler, önceden tanımlanmış ve programda belirtilmiş, mümkün bütün çıkarımlar ve mantıksal seçenekler program tarafından öngörülmüş olmalıdır.

Bilgisayarlar iki temel grupta sınıflandırılabilirler:

- **Benzeştirmeli (analog) bilgisayarlar**
- **Sayısal (digital) bilgisayarlar**

Benzeştirmeli Bilgisayarlar

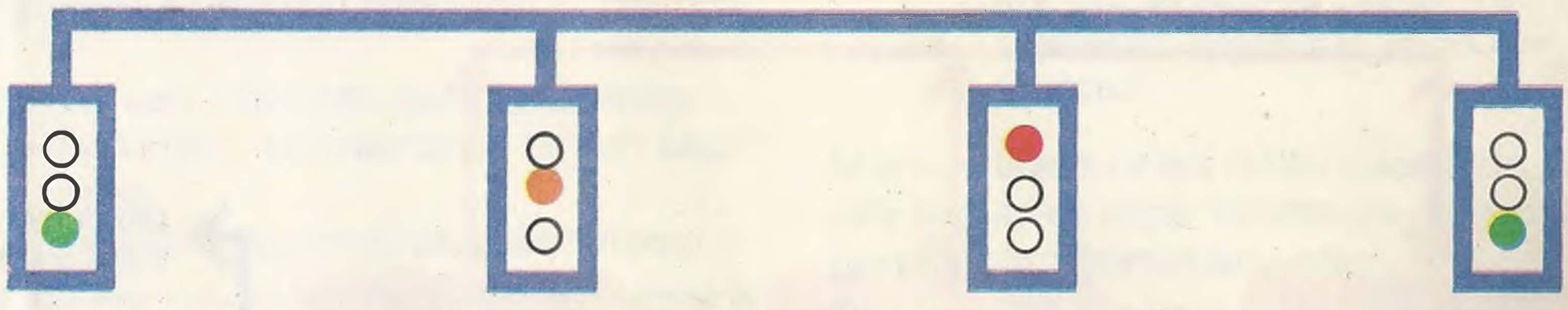
Benzeştirmeli terimi, bu tür bilgisayarların çalışma ilkesinden türetilmiştir. Yapılacak matematik hesaplar, fiziksel büyüklüklerle olan benzerlikler yoluyla çözülür. Örneğin, hesap cetveli benzeştirmeli bir aygıttır, çünkü işlemlerin sonucunu uzunluklarla benzeştirerek bulur (bkz. şekil, s.34). Bu çeşit makinelerde, bir işlemin sonucu, ilgili fizik yasalarından yararlanarak hesaplanır. Bu yüzden, her işlem türü kendi gereklerine uygun bir devre gerektirir. Bu makinelerin kullanımı, tümüyle bilimsel çalışmalarla sınırlıdır.

Sayısal Bilgisayarlar

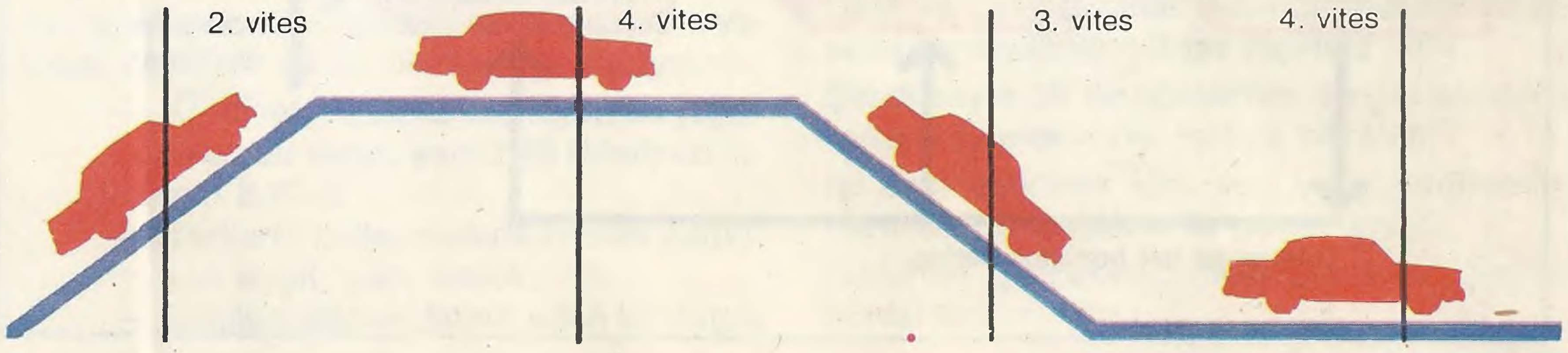
Biz, özellikle günümüzün elektronik bilgisayarları üzerinde duracağız.

Söyleyeceklerimizin çoğu, eski mekanik ve hidrolik bilgisayarlar için de geçerlidir. Elek-

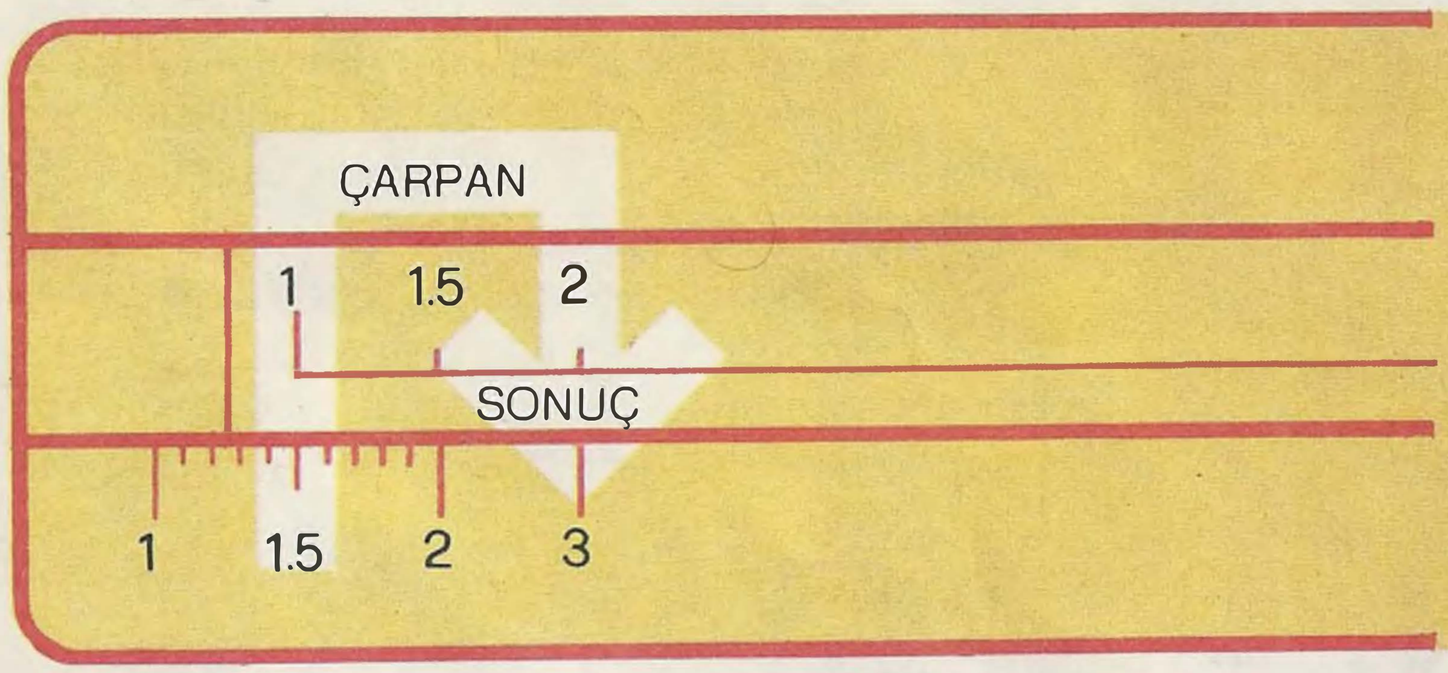
Dizisel sıra: *trafik lambaları*



Mantıksal sıra: *vites kullanımı*



Ücretsiz PDF Kitap Arsivi - https://rantey.org



HESAP CETVELİ:

Hesap cetveli, toplama ve çıkarma dışında karmaşık bazı işlemlerin yapılması için herkes tarafından kullanılır. Bu bir benzeştirmeli aygıt örneğidir: aslında hesaplar, cetvelin üzerindeki uzunluk derecelemesiyle benzeştirilerek yapılır.

tronik bilgisayar, bütün işlemlerini elektronik devreler sayesinde gerçekleştiren bir bilgisayardır.

Bilgisayarın elektronik devreleri iki çeşit olabilir:

- **Sayısal devreler**
- **Benzeştirmeli devreler**

Sayısal bir devre yalnızca iki durum tanır:

Sinyalin varolması = VAR (ON)

Sinyalin olmaması = YOK (OFF)

Oysa mantıksal devre, sinyalin varlığı ve yokluğunun yanında, değerini de ölçebilir.

Fotoğraf filmi, sinyalle (ışıkla) doğru orantılı bir ürün (fotoğraf) ürettiğine göre, benzeştirmeli bir aygıttır.

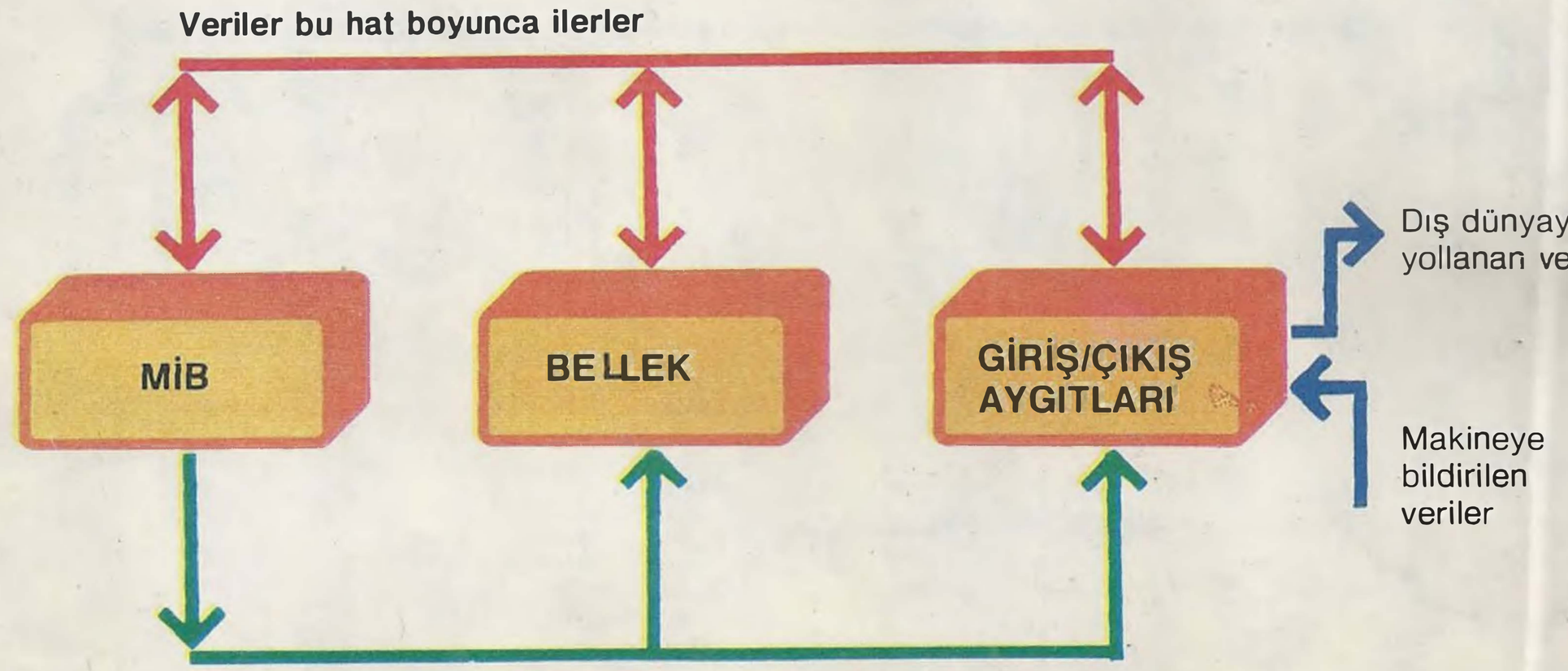
Bir düğme ise sayısal bir aygıttır, çünkü yalnızca açık ve kapalı olma olasılığı vardır. Bir radyo, bir televizyon, bir müzik seti, hep ben-

Sayısal bilgisayar, mantıksal işlemleri devrelerin bir dizi

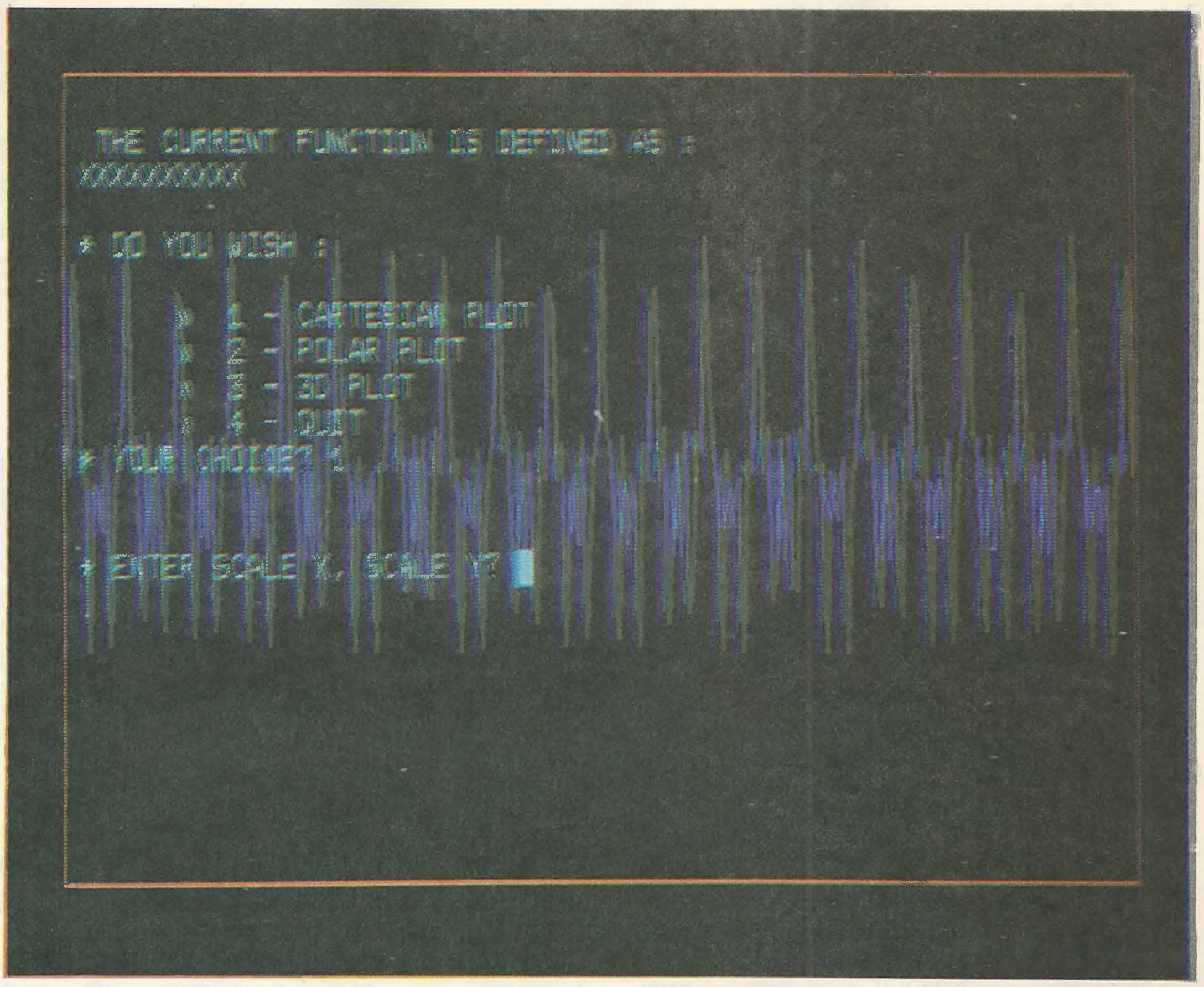
VAR/YOK
(sinyal mevcut/değil)

durumu aracılığıyla yapan bir aygıttır.

KİŞİSEL BİR BİLGİSAYARIN TEMEL YAPISI



Ücretsiz PDF Kitap Arsivi - https://rantey.org



Bilgisayar, faaliyetlerini ekrana yansıtarak da gösterebilir. Resimde görülen çok gelişmiş bir aygıttır. Kimi aygıtlar yalnızca tek çizgili çizimler yapabilir.

zeştirmeli aygıtlardır, çünkü aldıkları sinyalle orantılı olarak görüntü ve ses üretirler.

Bir bilgisayar aşağıdaki işleri yerine getirebilir:

— İki veri üzerinde işlem yapılması
— İki verinin kıyaslanarak birinin seçilmesi
— Verilerin taşınma ve bellenmeleri

Temel işlevlerinin sayısı bu kadar az olmakla birlikte, çeşitli birleşmeler yardımlarıyla, bilgisayar mümkün bütün işlemleri yapabilir.

Sayfa 94'deki şekil (alttaki) bir bilgisayarın yapısını özetliyor ve şu birimleri gösteriyor:

— Aritmetik hesapları ve kıyasları yapabilen bir aygıt, yani MİB (Merkezi İşlem Birimi)
— Verilerin bellenmesine olanak sağlayan aygıt, yani bellek.
— Bilgileri alan ve kabul eden bir aygıt, yani giriş; çıktıları insana ileten bir aygıt, yani çıkış. İkisi birlikte girdi-çıktı ya da giriş-çıkış anlamına, G/Ç ile gösterilir (İngilizce'de I/O = Input - Output).

Merkezi işlem birimi (MİB) işlemlerin yapılmasını sağlar ve diğer birimlerin çalışmaları için gerekli olan komutları yollar.

Bellek, verileri almakla ve biriktirmekle ve istenildiği zaman geri sağlamakla yükümlüdür.

Giriş ve çıkış aygıtları, verileri makine ve insanın anlayabileceği bir biçimde iletir.

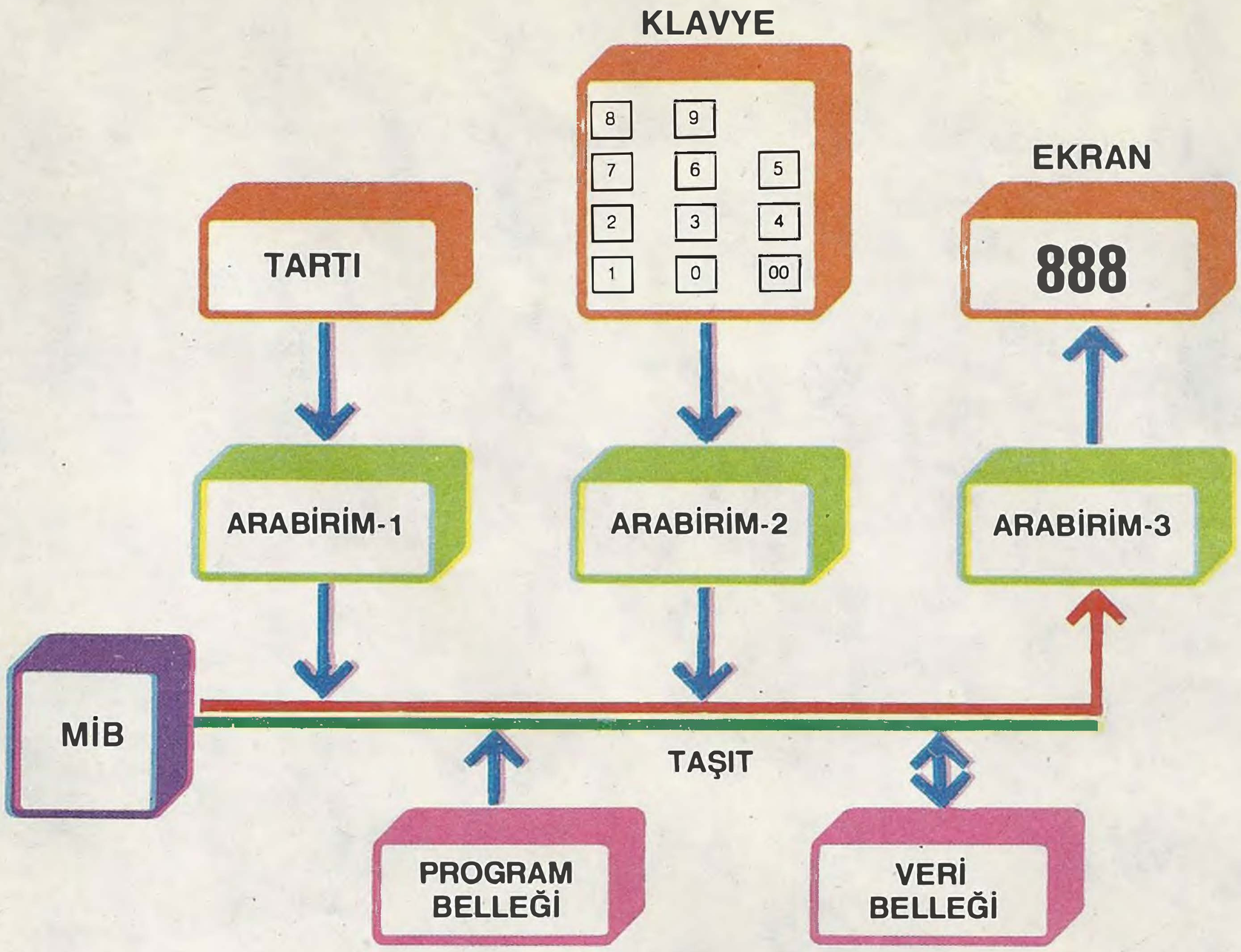
Şimdi sayfa 36'da gösterilen, birçok satıcının kullandığı elektronik tartıya bakalım.

İşi basitleştirmek için, veri ve güdüm iletim hattı tek bir bağlantı ile gösterilmiştir.

Sistemin gerçekleştirmesi gereken işlevler şunlardır:

Ücretsiz PDF Kitap Arsivi - https://rantey.org

BİLGİSAYARLI BİR SİSTEM: ELEKTRONİK TERAZİ



A / Bir cismin tartılması ve bu ağırlığın değerinin makine tarafından algılanması.
Tartma, tartıyla sağlanır. Bu veri, arabirim 1'de sayısal bilgiye çevrilir. Arabirim 1 bir giriş aygıtıdır (Input).

B / Cismin (birim) fiyatının bilinmesi.
Bu işlev, klavye ve arabirim 2 ile yapılır. Arabirim 2 de bir giriş aygıtıdır.

C / Toplam fiyatın hesaplanması.
Merkezi İşlem Birimi'nde (MİB) yapılır.

D / Sonuçların sunulması.
Bunlar arabirim 3 ve göstergeyle(*) sağlanır. Arabirim 3 bir çıkış aygıtıdır (Output).

Şemada iki kutu daha gözükmektedir:

- **Program Belleği**
- **Veri Belleği**

Şimdi, neden iki bellekten yararlanmamız gerektiğini görelim.

Program Belleği'nin işlevi ortadadır. Bu, doğru olarak yorumlandıklarında MİB'e işleme konacak işlevlerin sırasını (A-B-C-D) bildiren (yani programı oluşturan) bir dizi bilgiyi içerir. Veri Belleği ise verilerin yüklendikleri anda kaydedilmelerini ve MİB'in istediği anda onları kullanabilmesini sağlar. İşlemlerin sırası aşağıdaki şekildedir:

1 / Tartı, kefesine bir cismin konduğu işaretini verir.
2 / MİB, Program Belleği'ne yapılması gereken işlevi sorar.
3 / Program Belleği, MİB'e veriyi (ağırlığı) almasını ve Veri Belleği'ne ulaştırmasını söyler.
4 / Bu ulaşım gerçekleştiğinde, MİB gene Program Belleğine danışır.
5 / Program Belleği, MİB'e klavyeyi çalıştırmasını ve yazılan bilgiyi (fiyatı) almasını söyler.
6 / MİB, arabirim 2'yi harekete geçirerek, klavyenin verisini alır.
7 / MİB, gene Program Belleği'ne danışır.
8 / Program Belleği, MİB'e 3. durumda derlediği veriyi alıp, 5. durumda derlediği ile çarpmasını söyler.
9 / Son olarak, MİB'in Program Belleği'ne bir kez daha danışmasıyla, 8. adımın sonucu arabirim 3'e yollanır ve terazi göstergesinde gözükmesi sağlanır.

(*) Ekran ya da gösterge deyimleri ile bilgisayarın çıktılarını görünür kılan bir aygıt kasdediliyor. Bu konuya ileride daha ayrıntılı olarak değineceğiz.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Yukarıda yazılı olan işlemler, bir kez makineye ulaşabilecek bir dile çevrildi mi, artık program oluşmuş demektir.

Bir Bilgisayardan Nerede ve Nasıl Yararlanılmalıdır?

Bir bilgisayarın kullanımı şu ana koşula bağlıdır:

GEREKEN TÜM VERİLER BİLGİSAYARA VERİLMELİDİR

Bu, görüldüğü kadar beylik bir kural değildir. Bilgisayarı yeterince tanımayan ve ona çok abartılmış yetenekler yakıştıran kişiler bazen çok tuhaf sorular sormaktadırlar.

Oysa, her şeyden önce şu hatırlanmalıdır ki, bilgisayar ancak kendisine verilen verileri işleyebilir.

İncelenecek ikinci nokta şudur:

BİLGİSAYAR NE ZAMAN İŞE YARAR?

Bir bilgisayar satın almanın, mali açıdan her zaman iyi bir yatırım olacağı söylenemez. Genelde, böyle bir yatırımın kârlı olabilmesi için, aşağıdaki koşulların hiç olmazsa birkaçı biraraya gelmelidir:

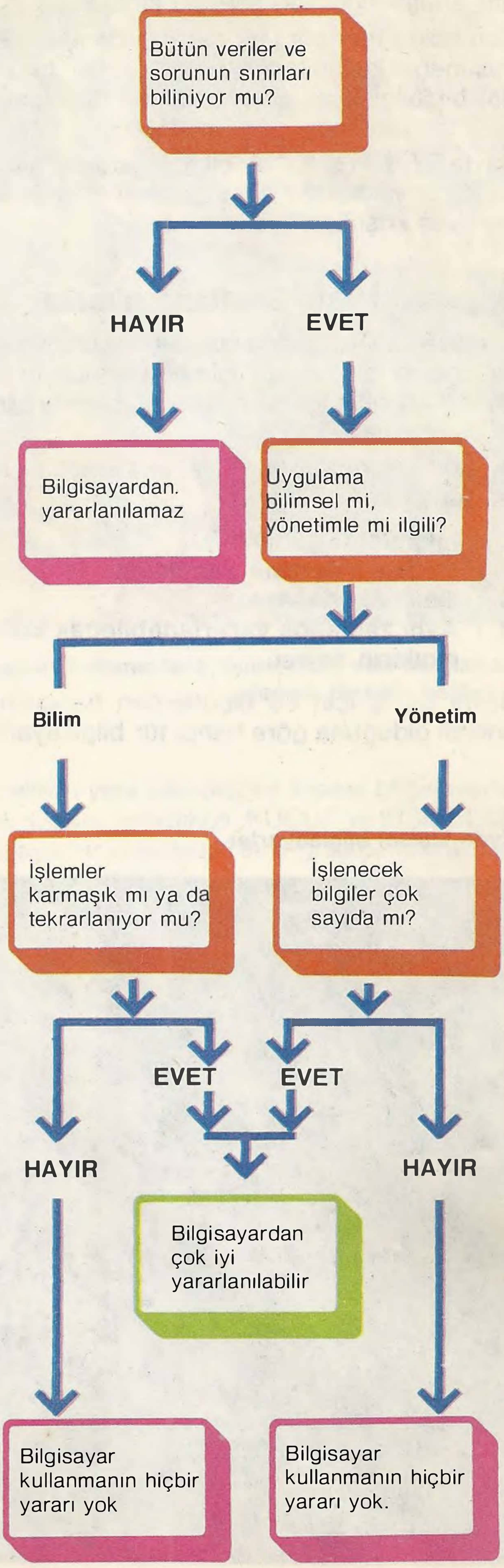
1 / **Tüm veriler kullanıma hazır durumdaysa, yani yeni bir araştırma-inceleme gerekmiyorsa.**
2 / **Çok sayıda veri, tekrar tekrar incelenecekse.**
3 / **Yapılacak çok işlem varsa.**
4 / **Belirli bir iş uzun bir süre için denetlenecekse.**

Birinci nokta temel koşuldur.

Bilgisayar ona sağlanandan daha fazlasını veremez. Başka bir deyişle, programcı makineye gerekli ön bilgileri yüklemediği takdirde, makine sonuç veremez.

İkinci nokta, bir bilgisayarın hangi koşullara yararlı olduğunu belirtmektedir. Burada şunu hatırlatmakta yarar vardır ki, bu tür bir gerecin alınıp yerleştirilmesi, yatırım maliyetleri ile kalmayacaktır. Programların geliştirilmesi (yazılım) yeni ve başka maliyetler getirecek, bu da önemli bir fonun buraya akmasına yol açacaktır. Bu maliyetler, ancak insan çalışmasının büyük ölçüde hafifletilmesiyle karşılanabilir.

SORUNUN KARŞILANABİLİR ANALİZİ



Ücretsiz PDF Kitap Arsivi - https://rantey.org

Üçüncü nokta daha çok bilimsel sorunlarla ilgilidir. Pek sık karşılaşılan bir durum, bilimsel bir tasarının, en iyi sonuç elde edinceye kadar ardışık işlemleri zorunlu kılmasıdır.

Son nokta herhangi bir sürecin denetiminde makinenin kullanılmasıyla ilgilidir. Bu durumda, bir bilgisayar almak, hemen her zaman kârlı bir yatırımdır.

Sayfa 37'deki şekil, bir bilgisayar satın almaya karar vermeden önce gözden geçirilmesi gereken koşulları özetlemektedir.

Bilgisayarların Sınıflandırılması

Şimdiye kadar, aslında çok değişik olabilecek makineleri tanımlamak için hep bilgisayar terimini kullandık. Şimdi onları verimlerine göre sınıflandırmaya çalışalım.

Bu sınıflandırmada kullanılacak başlıca ölçütler şunlardır:

A / İşlemlerin çabukluğu.
B / Verilerin iletilme çabukluğu.
C / Bellek kapasitesi.
D / Aynı zamanda yararlanabilecek kullanıcıların sayısı.

Belirli bir iş için bu ölçütlerden hangisinin önemli olduğuna göre hangi tür bilgisayarların daha yararlı olacağına karar verilebilir. Başlıca bilgisayar türleri şunlardır:

- **BÜYÜK BİLGİSAYARLAR**
- **KÜÇÜK BİLGİSAYARLAR**
- **KİŞİSEL BİLGİSAYARLAR**

BÜYÜK BİLGİSAYAR, adından da anlaşıldığı gibi, büyük boyutlara sahip ve birçok kişi tarafından aynı anda kullanılabilen bir bilgisayardır.

Bunları çalıştırabilmek için bir bilişimci ekibi gerekir. Çok büyük bellek kapasitesi ve çok yüksek bir çalışma hızı gerektirir. Bu tip bir makine, büyük ölçekli endüstri kuruluşlarının işletme ve yönetiminde ya da önemli bilimsel merkezlerdeki araştırmalarda kullanılır. Örneğin, büyük bilgisayarlar uyduların yörüngelerini ve füzelerin rotalarını hesaplar; özellikle büyük inşaatların elektronik devreleri için başvurulan karmaşık hesaplar ve testler böyle bilgisayarlarla yapılır.

Mahkemelerdeki gibi büyük ölçekli arşivlerde kullanılır. Devlet dairelerine, vergi dairesine, mal stoklarına ya da üretim hatlarına da uyarlanabilirler.

Bunların yanında, örneğin bir savaşın temsili (simulation) gibi çok özgül kullanımlardan da

Evde kişisel bilgisayarlar.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Büyük bir bilgisayara bağlı terminaller, kullanıcılara, işletmenin veri bankasına ulaşma olanağı sağlıyor.

söz etmek gerekir. Bu bilgisayarlar hayali bir savaşın tüm evrelerini inceleyebilecek kapasitede olduklarından, birkaç stratejinin aynı anda incelenmesine olanak verirler.

KÜÇÜK BİLGİSAYAR, tam tersine, daha düşük fiyatlı, kullanılması daha kolay ve genelde aynı anda birçok kişiye birden hizmet verebilen bir makinedir. Yeni birimler eklenip kapasiteleri artırılabildiğinden, küçük bilgisayarların yapıları esnektir.

Üreticilerin eğilimi, müşterinin ihtiyaçlarına derhal cevap verecek sınırlı olanaklara sahip makineleri hemen sağlayıp, gerektikçe eklemeleri sonradan sağlamak yönündedir. Bir sistemin (bellek kapasitesi, giriş ve çıkış aygıtları cinsinden) gücüne sistem yapısı adı verilir.

Bu makinelerin kullanım alanları çok değişiktir ve sistem yapılarına bağlıdır. Küçük bilgisayar, büyük bir bilgisayar gerektirecek kadar büyük olmayan, ama kişisel bilgisayarla karşılanabilecek kadar küçük olmayan bir ihtiyaca cevap verir.

KİŞİSEL BİLGİSAYAR, ilkede tek bir kullanıcı için yapılmış ve kullanımı çok kolay olan makinelerdir. İlerde daha iyi göreceğimiz gibi, özellikle yeni teknolojiler kişisel bilgisayarların gücünü artırdıkça, KÜÇÜK ve KİŞİSEL bilgisayarlar arasındaki sınır gittikçe belirsizleşmeye başlamıştır.

Gerçi bazen MİKRO ve kişisel bilgisayarlar arasında bir farklılaştırmaya gidilmektedir, ama bu iki makine türü arasındaki tek fark veri bellek kapasitesindedir; diğer bütün işlevleri benzerdir. Bu yüzden, kullanımlarındaki basitlik ve fiyatlarındaki büyük fark, değişik kullanımlar için kişisel bilgisayarların tercih edilmesine yol açmaktadır.

Aşağıda, kişisel bilgisayarların temel kullanımlarının bir listesi vardır:

- Küçük işletmelerde bordroların hazırlanması
- Küçük ya da büyük mağazalarda muhasebe
- Faturalama
- Emlak yönetimi
- İnşaat mühendisliği hesapları
- Küçük kütüphanelerin işletilmesi
- Adres kütüklerinin yönetilmesi
- Mali çözümlemeler
- Stok kontrolu

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Gündelik Yaşamımızda Bilgisayarlar

Kişisel bilgisayarların yaygınlaşmasıyla birlikte, nüfusun büyük bir bölümünün bilgisayar sahibi olacağı günler iyice yaklaştı. Bu gelişmenin sonuçları için şimdiden bazı kestirimlerde bulunmak mümkün.

Gündelik yaşamımızda her gün tekrarlanması gereken bir sürü zevksiz iş, rahatlıkla bilgisayara terkedilebilecek. Ancak, bilgisayar devralınca, bu işlerin niteliği de kaçınılmaz olarak değişecektir.

Örnek olarak, günlük alışveriş listesinin hazırlanması gibi, ilk bakışta önemsiz görünebilecek bir işi ele alalım. Bir ev hanımı almaya karar vereceği malları birçok yönden birden değerlendirmek zorundadır. Pazarda bulunan malların (örneğin, mevsimlik sebzelerin) hangileri olduğunu, bu malların fiyatlarını, aile bütçesini ve ailenin damak zevki vb. İdeal olarak, piyasada bulunup da aile bütçesinin elverdiği mallar arasından, ailenin damak zevkini en çok tatmin edenleri seçecektir. Eğitim düzeyi de biraz yüksekse, bir yandan da alacağı gıda malzemesinin besin değerini gözönünde bulunduracak, saydığımız koşullara uyan malları bir de ''dengeli beslenme sağlama'' yönünden değerlendirecektir.

Yani ev hanımları, hiç düşünmeden de olsa, çok ince birtakım dengelerin kurulmasını gerektiren bir dizi hesap yapmak zorundadırlar. Tabiî, gündelik sıkıntılar, tedirginlikler, zaman darlıkları kimi zaman bu boyutların hepsinin birden gözönünde bulundurulmasını olanaksız kılacaktır. Buna bağlı olarak da çeşitli fedakârlıklar edilecektir.

Oysa bir kişisel bilgisayar, bütün bu işleri yürütmek üzere programlanabilir. Böylelikle, hem ev hanımı alışveriş listesi için harcadığı zamanı daha yaratıcı faaliyetlere ayırabilecek, hem ailenin mali kaynakları daha verimli bir biçimde kullanılabilecek, hem de aile, gerek sağlık gerek damak zevki bakımından daha doyurucu bir gıda rejimine kavuşacaktır.

Kişisel bilgisayarların yaygınlaşmasının şimdiden önemli değişikliklere yolaçmış olduğu bir başka alan ise eğitim alanı. Bu alandan hareketle kişisel bilgisayarların gündelik yaşam üzerindeki etkilerinin genel bir özelliğine, kişilere tanıdıkları bireyselleşme olanağına, değinebiliriz.

İlgileri ve yetenek düzeyleri farklı bütün öğrencileri tek bir müfredata ve aynı öğrenme düzenine zorlamanın sakıncalarını hep biliriz. Böylesi bir yöntem yalnızca öğrencilerin başarılarını

Evde, kişisel bir bilgisayar günlük alışveriş listenizi hazırlayabilir, hatta size sağlıkla ilgili öğütler verebilir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Odaların sıcaklık ayarı.
Ev hizmetlerinin
örgütlenmesi
ve yönetim.

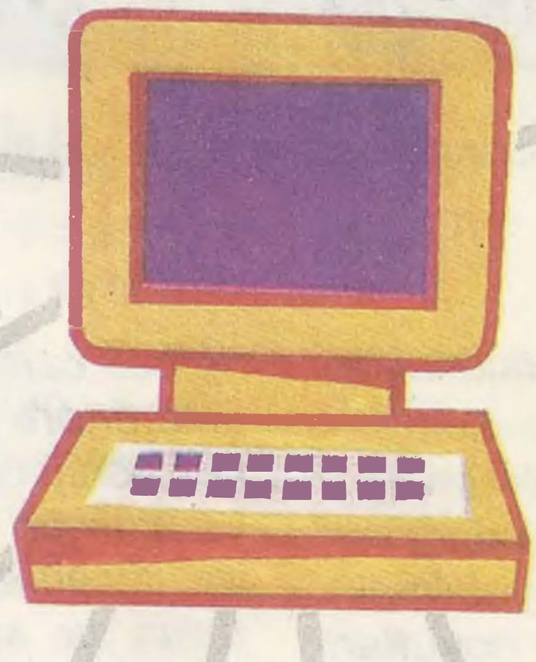
Veri bankalarıyla
bağlantı

Belgelerin
ulaşım ve
elde edilmesi

Kapı ve pencerelerin
kapanıp
kapanmadığının
kontrolü,
yangın ve gaz
kaçağı alarmı.

Ajanda.
Aile arşivi;
muhasebe ve
yazışmalar

Ödeme ve
mevduat
transferi



Eğitim
programları

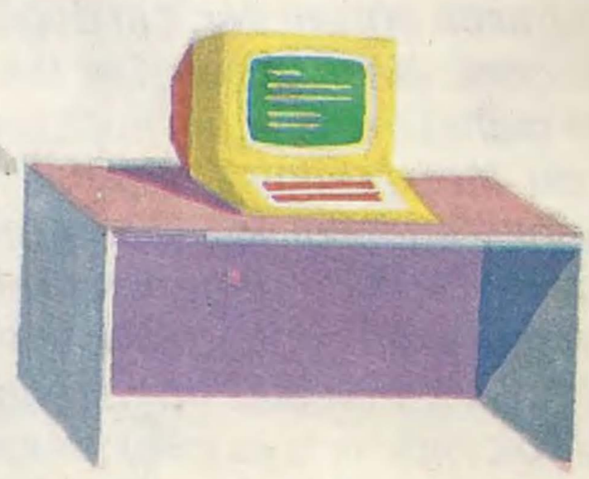
Süpermarketten
alışveriş

Oyunlar

Telefon
rehberleri

Müzik icrası
ve öğrenimi

Kütüphaneden
ve kültürel
kaynaklardan
yararlanmak



Büroyla
irtibatlı
evde çalışma
bölümü

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Evdeki bilgisayar, ailenizin tüm bireylerine yaşamı daha yakından izleme olanağı verir.

olumsuz yönde etkilemekle kalmaz, şevklerini de kırar.

Oysa Batı'da artık yaygın olarak kullanılan eğitim programları sayesinde, öğrenciler kendi yeteneklerine göre öğrenme hızını ayarlayabilmektedir. Yalnızca hız değil, eğitimin diğer boyutları da öğrencilerin niteliklerine göre biçimlendirilebilir. Söz gelimi 50 dakikalık bir ders süresi kimi öğrenciler için dikkatlerini yoracak ölçüde uzunken, kimi öğrenciler için 50 dakikanın sonundaki teneffüs, ilgi dağıtan elverişsiz bir ara olabilir. Ya da kimi öğrenciler, dikkatlerini matematik üzerinde ancak yarım saatliğine toplayabiliyorken, iki saat yorulmadan tarih okuyor olabilirler.

İşte bir bilgisayar, istendiği anda konuyu değiştirecek biçimde programlanabilir. Böylelikle matematik çalışırken, belki kendisi farkına varmadığı halde yorulan öğrenci, bilgisayarın teşvikiyle yeni bir konuya geçecektir.

Bunlar yalnızca öğrenciler için değil, yetişkinler için de geçerli. Nasıl öğrenciler tek bir öğretim sistemine mahkûm olmaktan kurtuluyorlarsa, yetişkinler de, kişisel bilgisayarlar sayesinde, boş zamanlarını, kendi ilgi ve yeteneklerine çok daha uygun bir biçimde değerlendirebilirler. Kişisel bilisayarların bu yönü, özellikle televizyon gibi diğer teknolojik buluşlar ile karşılaştırıldığında çarpıcı bir biçimde ortaya çıkar. Bugüne dek teknoloji, genellikle insan yaşamını birörnekleştirdiği için eleştirilmiştir; televizyon bu bakımdan iyi bir örnektir. Oysa kişisel bilgisayarlar, bugüne dek uzmanlık alanı olagelmiş çeşitli alanları, amatörlere de açarak, insanların ilgilerini daha çok yönlü bir biçimde geliştirebilme olanaklarını yaratmıştır. Örneğin kişisel bilgisayarlar sayesinde, ilgilenilen konularda arşivler düzenlemekten tutun da, satranç problemleri çözmeye, müzik bestelemekten tutun da, çeviri yapmaya varıncaya kadar birçok uğraş, geçmişle kıyaslanamayacak kadar kolay bir hale gelmiştir.

Bunlar, kişisel bilgisayarların öngörülen etkilerinden yalnız bir kısmı. Ancak bir yandan da hiç beklenmedik, öngörülmemiş birtakım gelişmelere tanık oluyoruz. Kişisel bilgisayarların sayısı arttıkça, bilgisayar iletişim ağları kurmak da daha kolay ve elverişli bir hal alıyor. Bu ise yepyeni ilişki türlerinin doğmasına yol açıyor. Bilgisayar ağlarının yaygınlaşmasının mümkün kıldığı belki de en çarpıcı gelişme örneği, Kaliforniya'da iki bini aşkın kişinin, böyle bir ağ sayesinde, bir ''takas'' ekonomisine geçerek neredeyse tamamen parasız bir yaşam sürmeye başlamaları. Geleceğin, benzeri nice ilginç gelişmeye gebe olduğunu kestirebilmek için kâhin olmaya gerek yok.

(İskender Savaşır)

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Bu modellerin küçükleri giderek evlere girmeye başladı. Adres defteri olarak, ders çalışırken, ya da aile bütçesini hazırlarken kullanılıyorlar. Bir televizyondan bile ucuz olan bazı modeller, artık vazgeçilmez ev aletlerinden biri haline gelmeye başladılar.

Bilgisayar Ağları

Bütün bilgisayarlar kullanıcılarıyla bağlantı kurmak zorundadır. Bunun için verileri alan ve ileten devreleri vardır. Genellikle iletim bir klavyeden sağlanıp, bir bilgiyazara ya da bir ekrana gönderilir. Ancak bunlara bir ikinci bilgisayar eklemeyi engelleyen hiçbir şey yoktur.

Böylelikle bir bilgisayar ağı kurulmuş olur. Bilgisayar ağları daha büyük sayıda bilginin iletim ve kullanımını sağlar. Daha önce de gördüğümüz gibi, bir bilgisayarın başlıca işlevi büyük miktarlardaki verilerin işlenmesidir. Ancak bellek birimi çok büyük olduğu ve yeni veri geldikçe sürekli olarak güncelleştirilmesi gerektiği için çok pahalıdır. Bu yüzden aynı belleği birçok kullanıcıya açmak iyi olurdu. İşte bir bilgisayar ağı çok sayıda veriyi birçok kullanıcının ulaşabileceği şekilde tek bir merkezde toplar.

Tipik bir bilgisayar ağının yapısı aşağıdaki şekilde gösterilmektedir.

Bu şekil, şu temel işlevleri göstermektedir:

- 1. Terminal(*) doğrudan ana bilgisayara bağlıdır. Bu bağlantı tarzına ''yıldız bağlantı'' adı verilir.
- 2 ve 3 sayılı yine yıldız bağlantılı terminallerin bir Paylaştırıcısı vardır.

Bu sistem verilerin uzun mesafelere iletilmesi için düşünülmüştür.

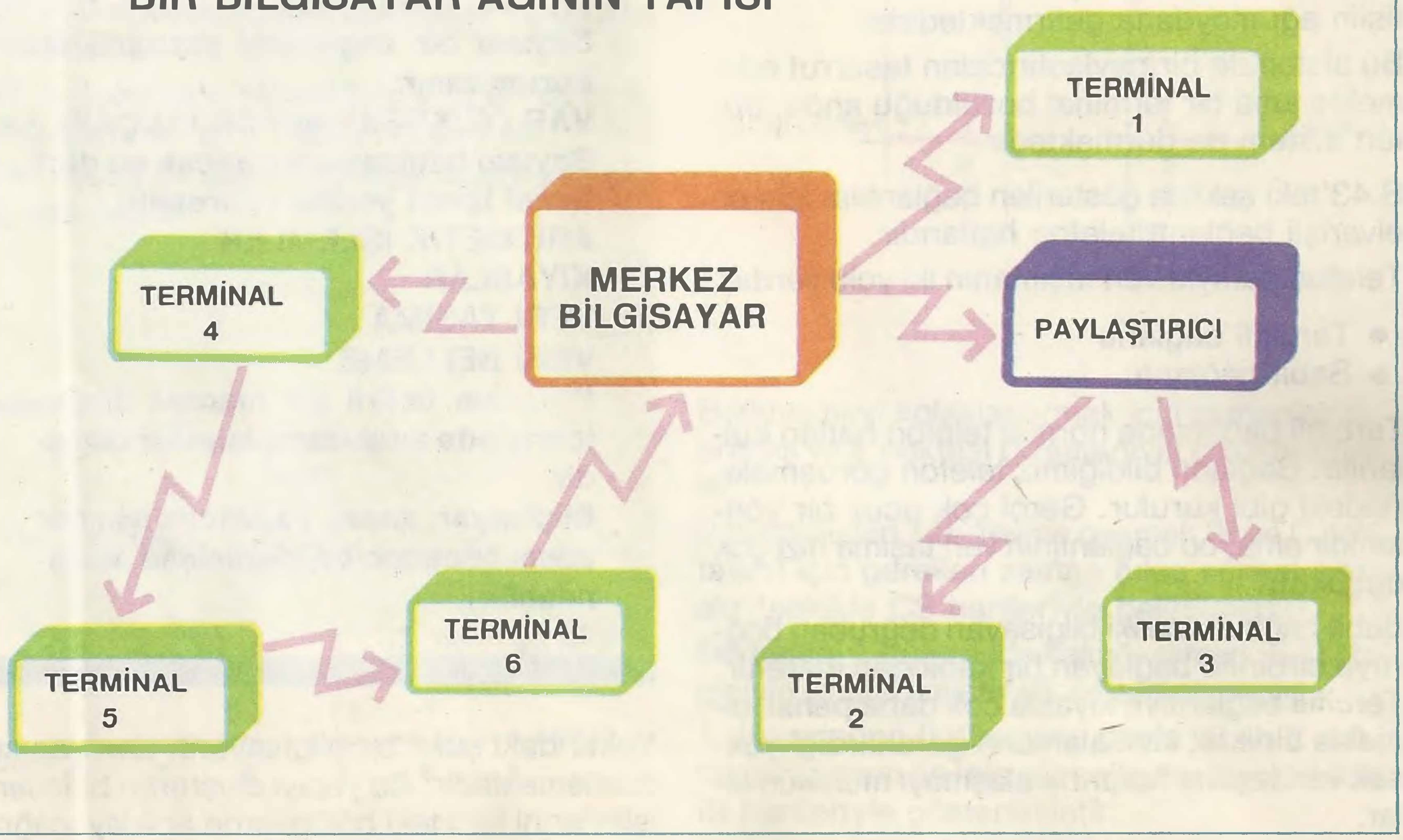
Örneğimizde, iki terminal (2 ve 3) ana bilgisayardan çok uzak bir konumda bulunmaktadır. Bu terminallerin bu uzak mesafeden bilgisayara erişebilmesini mümkün kılan Paylaştırıcıdır.

Paylaştırıcı sayesinde birkaç yerel hattı birleştirir, tek ve uzun bir hat haline getirebiliriz.

Paylaştırıcının görevi, verileri, ana hat üzerinde ve her iki terminal arasında dağıtılacak ya da toplanacak biçimde yönetmektir. Böylelikle çok uzun olan tek bir bağlantı hattı, daha sonra birçok yerel hatta (her biri bir terminale gitmek üzere) bölünür.

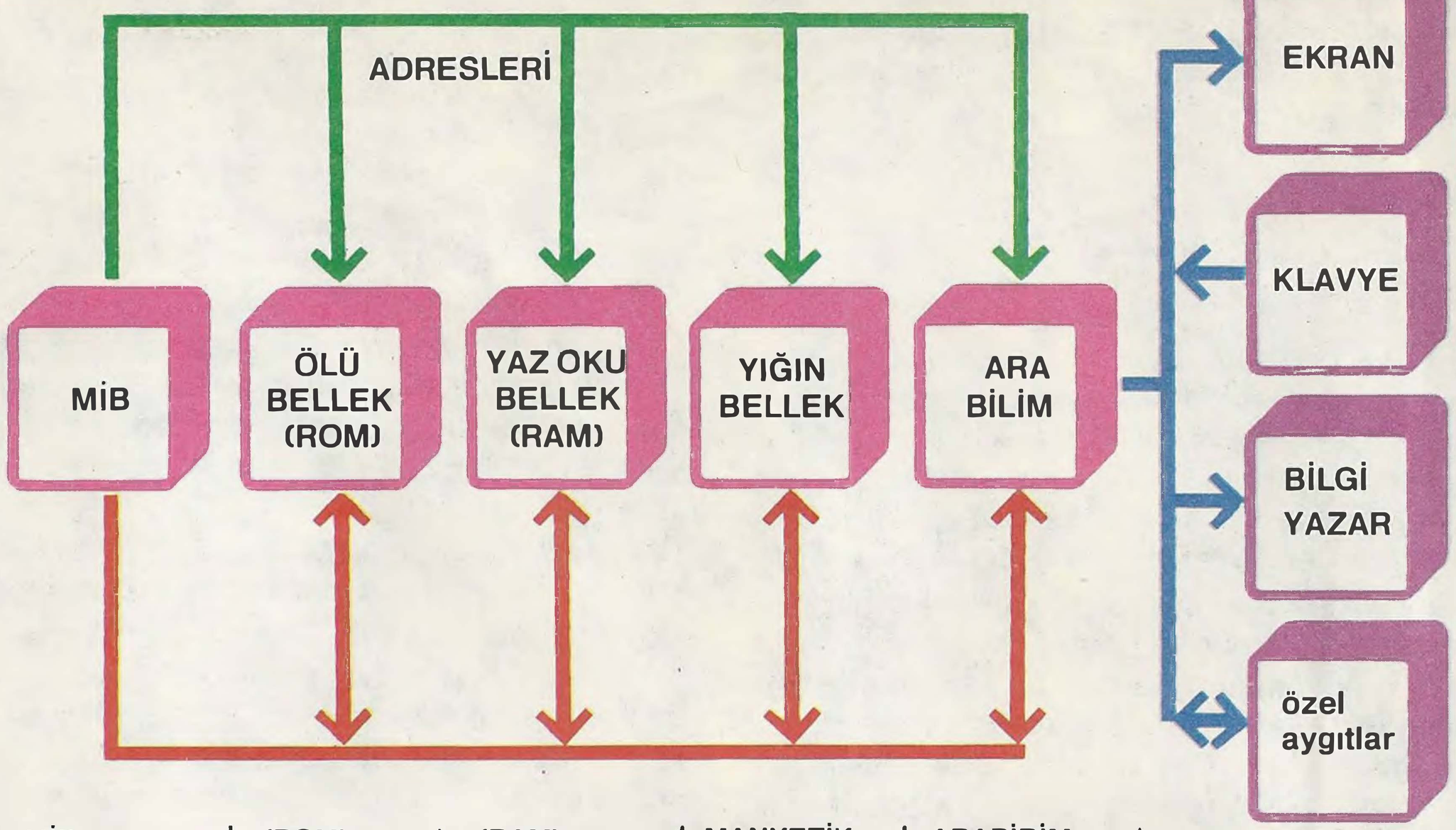
(*) Terminal demekle burada bilgisayar ya da başka bir makine tarafından yönetilebilen herhangi bir aygıt kastedilmektedir.

BİR BİLGİSAYAR AĞININ YAPISI



Ücretsiz PDF Kitap Arsivi - https://rantey.org

TAM TEŞKİLATLI BİR BİLGİSAYAR SİSTEMİNİN YAPISI



MİB
İşlemleri izler, diğer birimleri yönetir.

(ROM)
Okunan verileri içerir; ölü bellek.

(RAM)
Okunan ve yazılan verileri içerir; canlı bellek.

MANYETİK DESTEK
Büyük miktarlardaki verilerin bellenmesine olanak sağlar.

ARABİRİM
Makine ve kullanıcı arasında kodları uyumlar.

Verilerin giriş ve çıkış aygıtlarıdır.

Şekildeki 4, 5 ve 6 sayılı terminaller ise bir iletişim ağı meydana getirmektedirler.

Bu sistemde bir paylaştırıcıdan tasarruf edilmekte ama bir terminal bozulduğu anda, bütün sistem de durmaktadır.

S.43'teki şekilde gösterilen bağlantılar için en elverişli bağlantı telefon hatlarıdır.

Telefon hattıyla veri taşımanın iki yolu vardır:

- Tercihli bağlantı
- Sabit bağlantı

Tercihli bağlantıda normal telefon hatları kullanılır. Bağlantı bildiğimiz telefon görüşmelerindeki gibi kurulur. Gerçi çok ucuz bir yöntemdir ama, bu bağlantının veri taşıma hızı çok düşüktür.

Sabit bağlantı ise iki bilgisayarı doğrudan doğruya birbirine bağlayan bir kablodan ibarettir. Tercihli bağlantıya kıyasla çok daha pahalı olmakla birlikte, kimi alanların gerektirdiği yüksek veri taşıma hızlarına ulaşmayı mümkün kılar.

Sayısal bir bilgisayar yalnızca iki durum tanır:
VAR (YÜKSEK) ve YOK (ALÇAK).
Sayısal bilgisayarlar ancak şu dört temel işlevi yerine getirebilir:
ARİTMETİK İŞLEMLER
KIYASLAR
VERİ TAŞIMA
VERİ BELLEME
Program belirli bir mantıkî düzen içerisinde sıralanmış işlevler dizisidir.
Bilgisayar, ancak yapılacak işin her adımı önceden belirlenmişse, kullanılabilir.

Yukarıdaki şekil bir bilgisayarın tüm yapısını özetlemektedir. Bu yapıyı oluşturan birimlerin işlevlerini ilerideki bölümlerde açıklayacağız.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Sayılama Sistemleri

Bu bölümde simgelerimize (yani rakam ya da harflere) bilgisayarın kullanabileceği bir biçim verme yollarını göstereceğiz.

Bölümün sonunda, bu simgelerden hesap yaparken yararlanma yöntemlerini vereceğiz. Bizim mantığımız, çağrışımlarımız ve içgüdülerimizle beslenen çok ince ve zengin bir mantıktır. Ama bu bölüm okunurken tamamen unutulması gerekir, çünkü bilgisayarın düz mantığı okurun mantığının inceliklerini ve zenginliklerini yorumlayamaz.

Bu yüzden bu bölüm ilk bakışta sevimsiz ve hatta zor gelebilir. Dikkatinizi sorunun kavramsal yönüne yoğunlaştırmanızı salık veririz. Bilgisayarla bir problem çözmek ilk bakışla zor görünse de, aslında çok basittir. Vereceğimiz örnekler bilgisayarın basit dilini öğrenmenize yardımcı olacaktır.

Darbeler

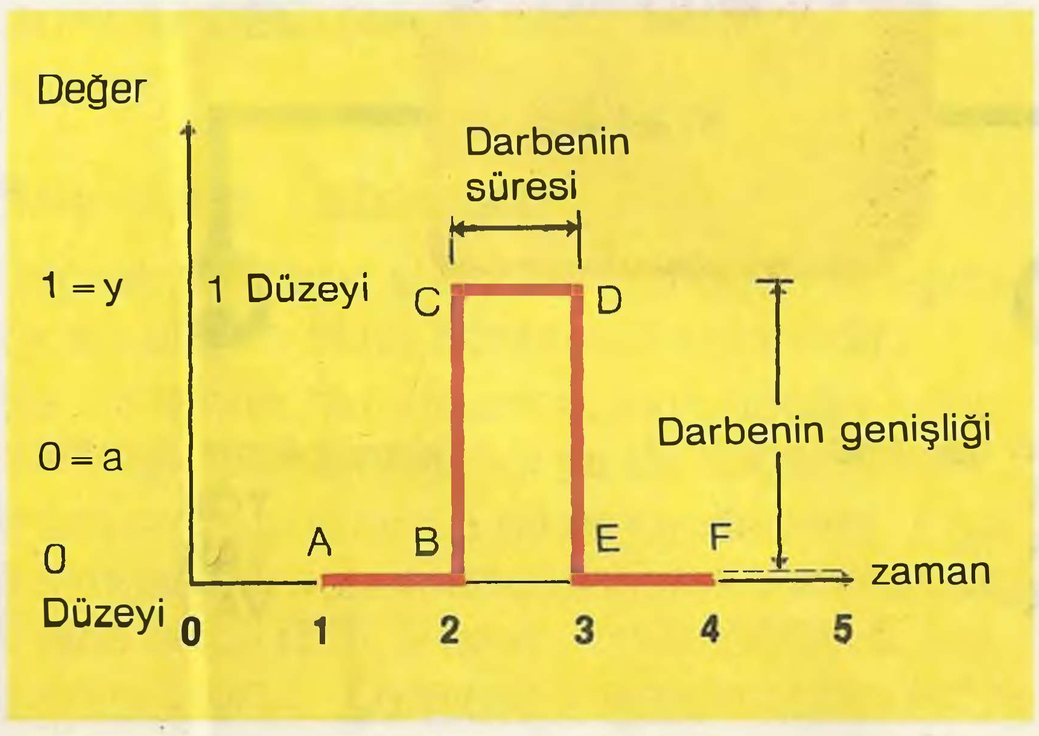
Hangi sayısal aygıta bakacak olursak olalım, yalnızca iki durum tanıdığını görürüz:

VAR (ON) = Sinyal var
YOK (OFF) = Sinyal yok

YOK = 0 durumu işaret noktasıdır ve 0 DÜZEYİ adını alır.

VAR = 1 durumu sinyalin en yüksek düzeyi olarak sunulur ve 1 DÜZEYİ adını alır.

Bu iki koşul, bir çizim üzerinde, aşağıdaki gibi gösterilebilir:



Olayı gözlemeye A anında başlıyoruz. A ile B arasında (diyelim ki bir saniye boyunca) düzey tabanda (yani 0 değerinde) kalır (0 düzeyi, **a** harfiyle de belirtilebilir).

B ve C arasında (çok kısa bir sürede) 0 düzeyinden 1 düzeyine geçilir (1 düzeyi genelde **y** harfiyle gösterilir).

y düzeyi C-D boyunca sürer. D-E aralığında (yine çok kısa bir sürede) **y** düzeyinden **a** düzeyine geçilir.

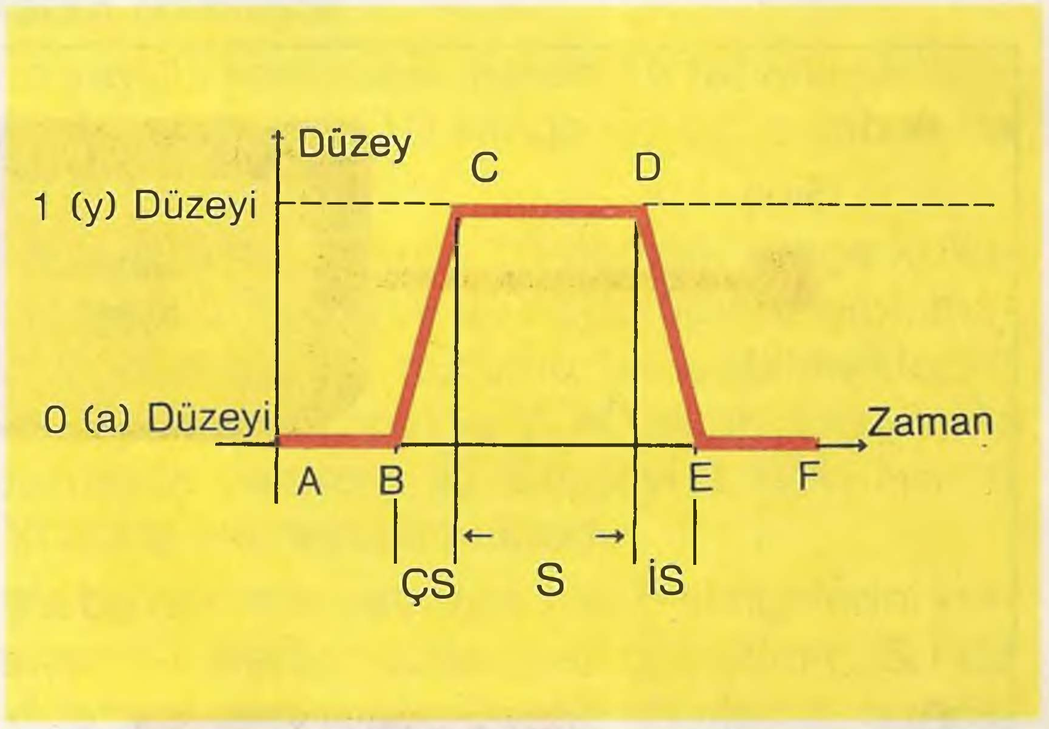
0 düzeyi durumu, sonsuza dek sürebilir. Şu çevrim:

- **a** düzeyi
- **y** düzeyi
- **a** düzeyi

''darbe'' adını alır.

Bir darbe, aslında bizim verdiğimiz biçimde olamaz.

Şekilde 0 (**a**) düzeyinden 1 (**y**) düzeyine geçmek için gerekli olan zaman dikkate alınmamaktadır. Aynı durum, ters duruma (D ve E noktaları) geçerken de söz konusudur. Çünkü hiçbir elektronik devre, bir durumdan diğerine, aradan bir süre geçmeden ulaşamaz. Öyleyse bir darbenin gerçek seyri şudur:



Betimlemeyi kolaylaştırmak için zamanlar artırılmış ve A noktası 0 zamanıyla çakıştırılmıştır.

0 düzeyinden 1 düzeyine geçmek (B ve C noktaları) için gereken zaman **çıkış süresi** adını alır (şekilde ÇS harfleriyle belirtilmiştir).

Darbenin, 1 düzeyinde kaldığı zaman, C-D çizgisiyle gösterilmiştir (S ile belirtilmiştir).

1 düzeyinden 0 düzeyine geçiş (D ve E noktaları) zamanı da **iniş süresi** adını alır; şekilde İS harfleriyle gösterilmiştir.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Çıkış süresini (ÇS) somutlaştıran B-C çizgisi **çıkış yamacı**'dır.

İniş süresini simgeleyen D-E çizgisi ise **iniş yamacı**'dır.

Özet olarak bir darbenin ana parametreleri şunlardır:

- **Süre = S zamanı (C-D parçası)**
- **Çıkış süresi = ÇS zamanı (B-C parçası = çıkış yamacı)**
- **İniş süresi = İS zamanı (D-E parçası = iniş yamacı)**

Anahtarlama

Bir bilgisayar, işlemleri için yalnızca VAR ve YOK simgelerinden yararlanır.

VAR durumunu 1 ile (ya da **y** harfiyle), YOK durumunu da 0 ile (ya da **a** harfiyle) gösteririz.

Şimdi, A şeklinde olduğu gibi bir giriş ile donatılmış ve (örneğin ışıklı bir göstergeyle) durumunu bize gösterebilen bir sayısal aygıtımız olduğunu varsayalım. Başlangıçta, girişin de, çıkışın da 0 durumunda olduğunu (göstergenin sönük olduğunu) düşünelim. Girişe 1 verek, sayısal aygıtın durumunu değiştirecek ve çıkışta 1 gözükmesini (ışığın yanmasını) sağlayacağız.

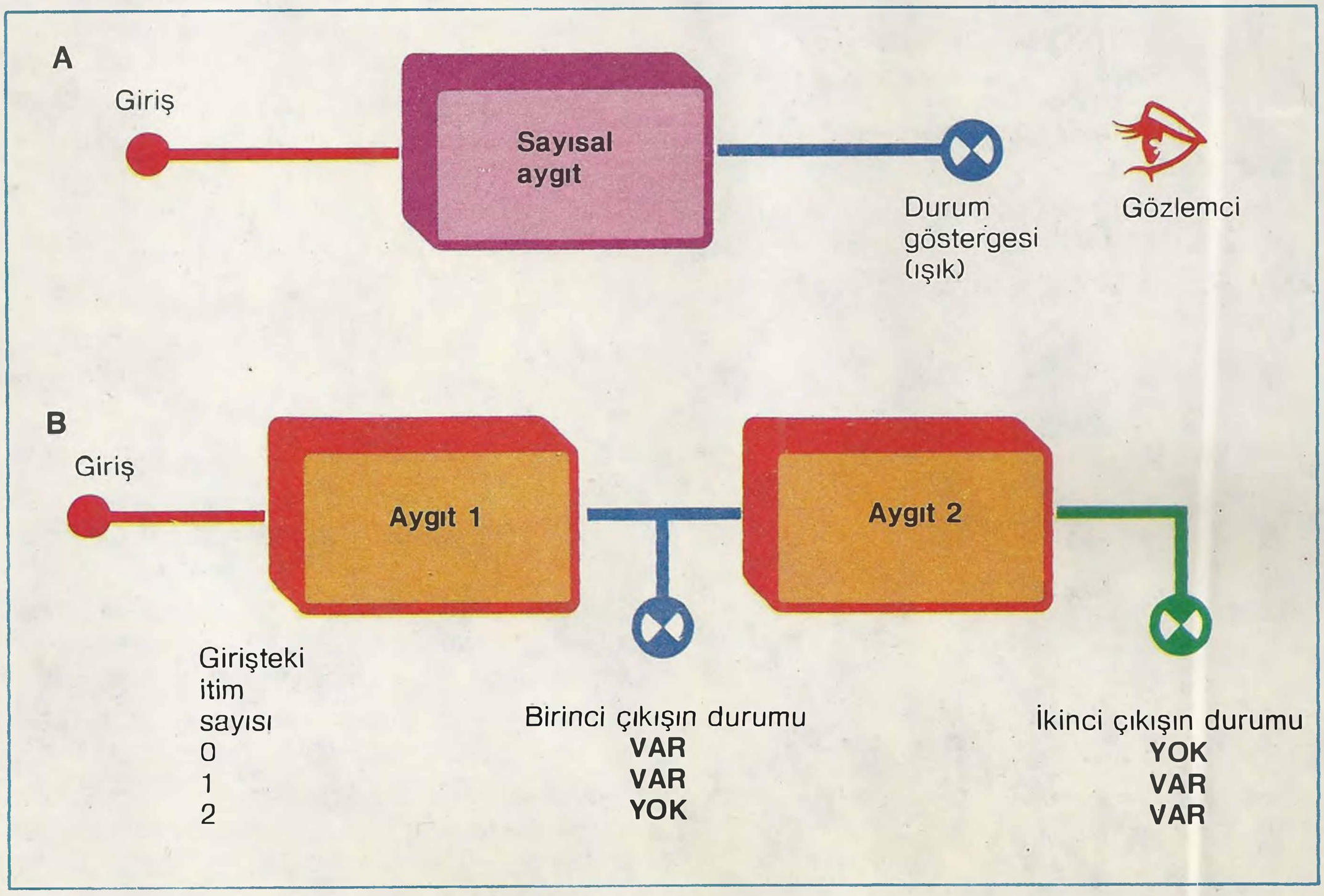
Gözlemci, bir darbenin üretildiğini söyleyecektir. Bu durum değişmesine **anahtarlama** adı verilir.

Aygıt bir kez anahtarlanmıştır.

Girişi 0'a çevirip, sonra yeniden 1'e çevirirsek, aygıt yeniden anahtarlanacak ve çıkıştaki 1, 0'a dönecektir. Gözlemci, başlangıçta 1 durumunu gördüğü için, 0 durumunu görünce yeni bir anahtarlamanın yapıldığını söyleyecektir. Öyleyse bu aygıt iki kez anahtarlanmış, yani 2 saymıştır.

Şimdi de, gözlemcinin gözleme girişteki ikinci darbeden başladığını varsayalım. Bu durumda, çıkışın 1'den 0'a dönüştüğünü görecek ve bir mi, iki mi anahtarlama olduğunu bilemeyecektir. Eğer sistemin çıkışına ışıklı göstergenin yanı sıra (B şeklinde görüldüğü gibi) ikinci bir eş-aygıt daha eklenirse, bu belirsizlik ortadan kalkar.

1'inci aygıtın çıkışı, ikinci aygıtın anahtarlanmasına yol açmaktadır; ikinci aygıt, birincinin geçirdiği değişiklikleri belleyip ortaya koyduğuna göre, girişteki darbe sayısının kaç olduğu (0, 1, 2) belirlememiz mümkün olmaktadır.

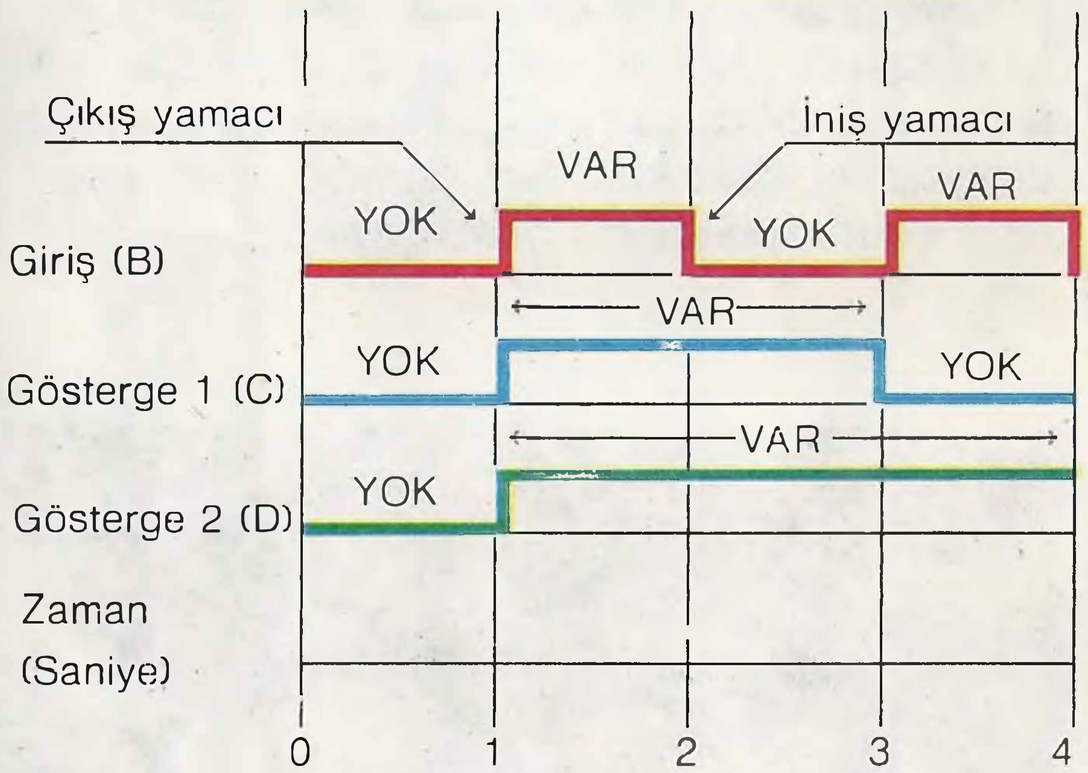


Ücretsiz PDF Kitap Arsivi - https://rantey.org

Girişin, birinci ve ikinci gösterge durumları bir tabloda özetlenebilir.

A	B	C	D
Darbe Sayısı	Giriş	1.Gösterge	2.Gösterge
0	0	0	0
1	1	1	1
2	1	0	1

Sistemin işleyişini mantıksal açıdan gösteren bu tabloya **doğruluk değerleri tablosu** denir. Her sayısal devrenin, yürütmesi gereken işlevlere göre az ya da çok karmaşık bir tablosu vardır.



Girişteki darbelerin iki-saniyede-bir ritminde geldiğini varsayıp, aşağıdaki çizimi hazırlayalım.

- 2. gösterge (D), 1. göstergenin (C) ikinci yanışında sönmektedir; öyleyse ikişer ikişer sayabilen bir sayacımız var demektir.
- Zaman içinde eşit aralıklı darbeler sağlayan aygıta SAAT adını veriyoruz.

Şunu belirtelim ki, elektronik bir sayısal devrede, **GİRİŞTE YOK-VAR GEÇİŞİ OLURSA DURUM DEĞİŞMESİ MEYDANA GELİR.**

Sayılama Tabanları

Değişik sayılama sistemleri içinde, mantığımıza en uygun olanı konumsal sistemdir.

Bu sistemde her rakamın, bulunduğu konuma bağlı olan bir ağırlığı ya da değeri vardır.

5 sayısını yazarken 5 birim kasdederiz. Eğer 5 simgesinin konumunu değiştirecek olursak, 5 tane onluk (50), 5 tane yüzlük (500) vb. kasdetmiş oluruz. Öyleyse 5 simgesi, bulunduğu konuma göre bir değer almaktadır. Konum ile değer arasındaki bu ilişkiyi bir tabloda görelim:

	Yüzler	Onlar	Birler
Ağırlık	100	10	1
Üs	5×10^2	5×10^1	5×10^0
Konum-değeri	5×100	5×10	5×1
Değer	500	50	5

Konumunun her değişmesinde, sayı 10 ile çarpılır. 10 değeri sayılama sistemimizin tabanını teşkil etmektedir, bu yüzden de *10 tabanlı konumsal sistem* adını alır (başka bir deyişle, Arap sayılama sistemi).

Romen sayılaması, konumsal olmayan bir sistem örneğidir. Bu sistemde, 5 değeri V simgesiyle, 50 değeri de L simgesiyle gösterilir. 5'den 50'ye geçmek için, 5 (V) simgesinin konumunu değiştirmek yetmemekte, bir yenisinin (L) işin içine sokulması gerekmektedir.

Yorumlanması pratik olmayan ve hesaplara da yatkın olmayan bu sayılama sistemi bırakılmıştır. Yerine, evrensel olarak konumsal sistem kabul edilmiştir.

En yaygın konumsal sistem 10 tabanlı sistemdir; bu sistemde 10 simge vardır ve taban da 10'dur.

Fakat bilgisayarlarda 10 değişik simge kullanılamaz. Daha önce de söylediğimiz gibi, makine yalnızca iki durumu tanıyabilmektedir: VAR (yüksek düzey) ve YOK (alçak düzey). Bu durumda yalnızca iki simgeyi **1** (VAR) ve **0** (YOK)'u kavrayabilmektedir.

İşte bu nedenle yalnızca 1 ve 0 simgelerini kullanan bir sistem kullanmak gereklidir. Bu da iki tabanlı sistemdir. Bu tip sayılama, sanılanın tersine, bilgisayarlarla birlikte doğmamıştır. XII. yüzyıla doğru Çin'de çoktan kullanılıyordu. İki tabanlı sistem de konumsal bir sistemdir, çünkü mevcut iki simgenin değeri konumlarına bağlıdır.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Okulda Bilgisayar

Bilgisayar eğitimde kullanılmaya çok yatkındır. Yalnızca bilişim, matematik, kimya ve biyoloji gibi bilimsel dallar değil, geleneksel olarak bilimsel kümeye katılmayan müzik ve dil eğitimi için de bilgisayarlara başvurulabilir. Çocuklar, kişisel bilgisayar kullanmaya yönelik doğal bir ön yapıya sahip gibiler. İlkokul öğrencileri için hazırlanmış bir programı örnek olarak alalım. ''Çıkartma işlemini öğreten program.'' Aşağıdaki karşılıklı konuşmada, öğrenciler sorularına derhal cevap almaktadırlar.

Bilgisayar: 3—2 = ?
Çocuk: 1
Bilgisayar: Çok iyi. 7—2 = ?
Çocuk: 5
Bilgisayar: Mükemmel. Peki, 23—16 = ?
Çocuk: 6
Bilgisayar: Aman! Bir kez daha denemek ister misin?
Çocuk: 7
Bilgisayar: Bu sefer oldu. Peki 44—38 = ?

Her doğru cevapta öğrenci kutlanmakta, yanlış cevapta ise yeniden denemeye teşvik edilmektedir. Çıkarma örneğinde kullanılan ''çok iyi'', ''mükemmel'' vs. deyimleri yalnızca şematik cümlelerdir. Doğru cevaplar için gülümseyen, yanlışlar için ise asık bir surat da aynı işlevi görebilir. Ya da, öğrencinin moralini büsbütün bozmamak kaydıyla, yanlış cevaplarda bir fren cayırtısı ya da dehşetli bir gürültü, doğrularda ise etkileyici ve coşkun bir trompet sesi kullanılabilir. Hatta kullanılacak görüntü ve ses efektleri öğrencinin beğenisine göre değiştirilebilir. Yani, program için seçilecek gösterim biçimi sonsuz seçeneklidir.

Örneğimizde program, öğrenci kolaylarını çözebildikçe daha zor sorular sormaktadır. İşte bu yüzden 3—2, 7—2 gibi çözülmesi kolay sorulardan hemen birden fazla basamaklı sayıların çıkarılmasına geçilmektedir. Ayrıca bilgisayar yalnızca öğrencinin ardarda vereceği iki doğru cevabı kaydetmekle yetinmek zorunda değildir. Örneğin bilgisayarda bulunan bir saat, iki cevabın verilmesi için gereken zamanı ölçebilir. Bundan başka, bir önceki dersin verileri, öğrenciye bundan kolay problemleri çözebildiğini göstermek amacıyla saklanabilir.

Sonuç olarak, daha zor olan iki basamaklı sayıların çıkarılmasına (23—16) geçilmiştir. Hesaplarda zorluğu gittikçe artırmaktan başka, program öğrenciye konu değiştirme (örneğin masal okumaya ya da bir rock parçası bestelemeye geçme) olanağını verir. Hatta, bir cevaptan diğerine süre uzuyorsa, bilgisayar, öğrencinin yorulduğu ya da sıkıldığına karar verip derse tamamen son verebilir.

Kısacası önemli olan dersin akış hızını öğrencinin ilgisinin derecesine ve bilgi düzeyine göre ayarlamaktır. Sorular çok zor ya da fazlasıyla kolay olduğu takdirde, öğrencinin dersten alacağı sınırlı olacaktır.

Ayrıca öğrencinin kendi kendini denetleyebilmesi, neyi bilip neyi bilmediğini kavrayabilmesi için cevapların anında verilmesi elzemdir.

Bütün bu gerekçelerden ötürü bilgisayarlar, öğretim yöntemi ve başarısı bakımından geleneksel ders kitaplarına kıyasla, çok daha ileri bir aşama oluştururlar. Çünkü değişmez bir sıra izleyen ders kitapları, öğrenciye aynı sıklıkta kendini deneme olanağı sağlayamaz.

Ücretsiz PDF Kitap Arsivi - https://rantey.org

Oysa bilgisayarla yürütülen dersler, öğrencinin yeteneklerini sonuna kadar geliştirmesini sağlamakla kalmayacak, aynı zamanda da eğitimi öğrenci için bir zevk haline getirecektir.

Buraya kadar söylediklerimiz yalnızca çıkarmayı öğrenme gibi ilkokullara özgü faaliyetler için geçerli değildir.

Gözümüzün önüne liselerin kalabalık sınıflarından birini getirelim. Bu ortamda zaman zaman öğrencilerin tek bir sorusunu olsun cevaplamaksızın, öğretmenin belirli bir konu üzerinde konuşmasıyla bütün bir dersin geçtiğini hepimiz biliriz.

Herhangi bir şey ilgilerini çekse bile, bu kalabalık sınıfta doyurucu bir cevap alamayacaklarını bilen öğrenciler, öğretmenin anlattıklarını kelimesi kelimesine not etmekle yetinirler.

Ayrıca yaptıkları ödevlerin sonuçlarını da bazen haftalar sonra alacaklarından neyi ne kadar öğrenmiş oldukları hakkında da açık bir fikirleri olmayacaktır.

İşte böylesi bir ortamda bilgisayarın öğretmenle öğrenci arasında çok önemli bir bütünleştirici işlevi olabilir. Bilgisayar, müfredatı aktarma işini devraldığı takdirde, öğretmenle öğrenci arasındaki ilişki de çok daha insanî bir nitelik kazanacaktır.

(Joseph Deken'in THE ELECTRONIC COTTAGE, "Elektronik Kulübe", 1981, kitabından alınmıştır.)

Bir Amerikan okulunda bilgisayarlı bir dershane.

Şimdi konumsal bir sayı sisteminin ayrıntılarını daha da yakından inceleyelim. Bildiğimiz 10 tabanlı sayı sisteminde her sayının 0,1,...9 gibi rakamların yanyana sıralanmasıyla ifade edildiğini görmüştük. Bu sıra içerisindeki herhangi bir rakamın, diyelim 5'in değeri, bulunduğu konuma ya da başka bir deyişle, o rakamı izleyen basamakların sayısına bağlı olarak değişir. Örneğin 50 sayısı 500'den böyle ayrılır.

Temel alışkanlıklarımızdan biri olduğu için, 10 tabanlı bir sayı yazmak için özel bir akıl yürütme gerekmez; yine de, basit görünen bu işlem bile aslında soyut bir işlemdir.

Bu sayı sisteminde her rakamı 10 tabanının ardışık kuvvetlerinin çarpanı olarak kullanırız. ''Kuvvet'' kavramının, sayının sağ üstüne yazılan **üst** tarafından gösterildiğini ve kendi kendisiyle çarpılması demek olduğunu anımsayalım. Örneğin;

$10^0 = 1$	sıfır kuvveti *
$10^1 = 10$	birinci kuvvet (onlar basamağı)
$10^2 = 10 \times 10$	ikinci kuvvet (yüzler basamağı)
$10^3 = 10 \times 10 \times 10$	üçüncü kuvvet (binler basamağı)

On tabanlı sayılamada her rakam, sağdan başlayıp sola doğru 10 tabanının bir üst kuvvetinin çarpanıdır. (Bu yüzden sık sık kullandığımız birler, onlar, yüzler vb. basamağı deyimleri yalnızca 10 tabanlı sayı sistemi içerisinde bir anlam ifade ederler. Yoksa, her sayı sistemine uygulanacak bir adlandırma için, birinci kuvvet, ikinci kuvvet vb. basamaklarından söz etmemiz gerekirdi.)

10 tabanlı sayı sisteminde bir sayı nasıl yorumlanır ve değeri nasıl elde edilir? Bu soruları 1024 tam sayısını örnek alarak yanıtlayalım:

Basamak	d	c	b	a
Sayı (rakam)	1	0	2	4
Taban kuvveti	10^3	10^2	10^1	10^0
Konum ağırlıkları	1000	100	10	1
Rakamların değeri (rakam x konum ağırlığı)	1000	0	20	4
Sayının değeri	$1000 + 0 + 20 + 4 = 1024$			

* Her sayının sıfırıncı kuvveti 1'e eşittir. Buna göre 2'nin de sıfırıncı kuvveti 1'dir; yani $2^0 = 1$.

Kimilerine göre 10 tabanının bunca yaygın olmasının nedeni 10 parmağımızın olmasıdır. Bu görüşe göre parmaklarımız ''Sayı'' kavramını kolaylıkla somutlaştırmamızı sağlıyorlar. Aristo da halkın çoğunluğunun onlarla saydığını belirtmişti.

Ancak burada, başka herhangi bir sayının da taban olarak kabul edilebileceğini söylemeliyiz. Örneğin, birçok dilde, hatta Yunancada bile, 5 tabanlı sayı sisteminin, Fransızcada ise 20 tabanlı sayı sisteminin kalıntılarına rastlamak mümkün.

Başka bir sayının taban kabul edilmiş olduğu bir sistemde yazılmış bir rakamı yorumlamak için başvurmak zorunda olduğumuz adımlar, 10 tabanlı örnekte kullandığımız adımların aynıdır. Her sayı sisteminde kullanılacak simgelerin sayısını seçilen taban belirler.

10 Tabanı için 10 simge:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

2 Tabanı için 2 simge:
0, 1 (ya da a, y)

8 Tabanı için 8 simge:
0, 1, 2, 3, 4, 5, 6, 7

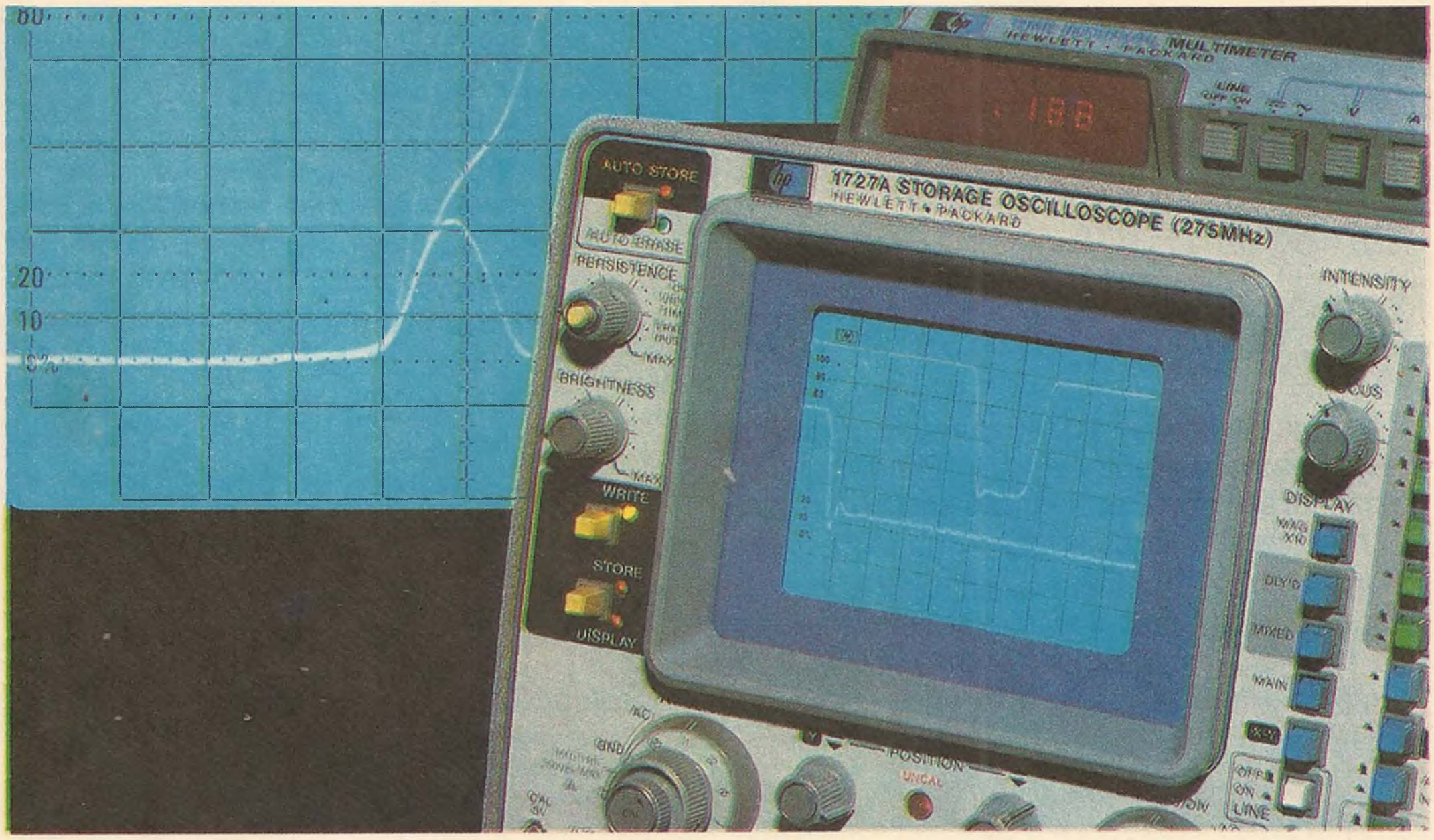
16 Tabanı için 16 simge:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

8 ve 16 tabanlı sayı sistemlerini daha sonra ayrıntılı bir biçimde inceleyeceğiz. Şimdi 2 tabanlı sayı sistemini gözden geçirelim.

2 Tabanlı Sayı Sistemi

Tıpkı 10 tabanlı sayı sistemi gibi 2 tabanlı sistem de konumsaldır. 10 tabanlı sayı sisteminde herhangi bir sayıyı yazmak ve yorumlamak için yapılması gereken işlemleri hiç düşünmeden, otomatik olarak yaparız. Oysa 2 tabanlı sistemde teker teker her adımı düşünmek zorundayız. Ancak kurallar her iki sistemde de aynıdır. Yine de okura aşinalık kazandırmak için, bu bölümde bol bol 2 tabanlı sistemde yazılmış sayı ve yapılmış işlem örnekleri vereceğiz.

Mümkün her sayı 2 tabanlı sistemde de yazılabilir. Şöyle ki, her rakama karşılık olmak üzere yalnızca iki simgeden biri kullanılır: ya 0 ya 1. Değişik konumlarda bulunan rakamlar, 2 tabanının ardışık kuvvetleriyle çarpılır. Bir örnek olarak herhangi bir sayının, örneğin



Hewlett Packard marka bir osiloskop. Bu aygıt elektrik darbelerinin biçimini gösterir.

1011'in yorumlanma biçimini, 10 tabanlı sistem için verdiğimiz örneğe benzeterek gösterelim:

Basamak	d	c	b	a
Sayı (rakam)	1	0	1	1
Taban kuvveti	2^3	2^2	2^1	2^0
Konum ağırlıkları	8	4	2	1
Rakamların değeri (rakam × konum ağırlığı)	8	0	2	1

Sayının değeri $8 + 0 + 2 + 1 = 11$ (10 tabanlı)

Şimdi ikinci bir örnek olarak 2 tabanlı, ama bu kez sekiz rakamlı bir sayının, örneğin **11111111** yorumlanmasını görelim:

Bu sayı, yani en yüksek 10 tabanlı değeri 255 olabilen 2 tabanlı sekiz rakam kümesi bazı bilgisayarların temel veri birimidir ve **sekizli** ya da **oktet** adlarıyla da anılır.

İki tabanlı sistemde yalnızca 0 ve 1 işaretleri kullanıldığından, herhangi bir sayının değerini belirtmek için çok sayıda simgenin kullanılması gerekir. Bu yüzden de uygulamada pek kullanışlı değildir. Bu nedenle sık sık başka iki konumsal sisteme de başvurulur: 8 tabanlı ve 16 tabanlı sistemler.

Bu sistemler tıkız olduklarından, yani daha az sayıda simge kullanılmasını gerektirdiklerinden, 2 tabanlı sisteme kıyasla daha kullanışlıdırlar. Bizi uzun bir 0'lar ve 1'ler dizisine başvurmaktan kurtarırlar. 2, 8 ve 16 tabanlı sistemler arasındaki farkı, zamanın saat, dakika ve saniye olarak gösterilmesine benzetebiliriz: 8 saat ile 480 dakika aynı şeylerdir; aradaki fark az ya da çok tıkız bir sistemin kullanılma-

---

Şimdi de 8 basamaklı 2 tabanlı bir sayının, örneğin 1 1 1 1 1 1 1 1 sayısının, 10 tabanına çevrilmesini görelim:

Basamak	h	g	f	e	d	c	b	a
Sayı (rakam)	1	1	1	1	1	1	1	1
Taban kuvveti	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Konum ağırlıkları	128	64	32	16	8	4	2	1
Rakamların değeri (rakam × konum ağırlığı)	128	64	32	16	8	4	2	1

Sayının değeri $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$ (10 tabanlı)

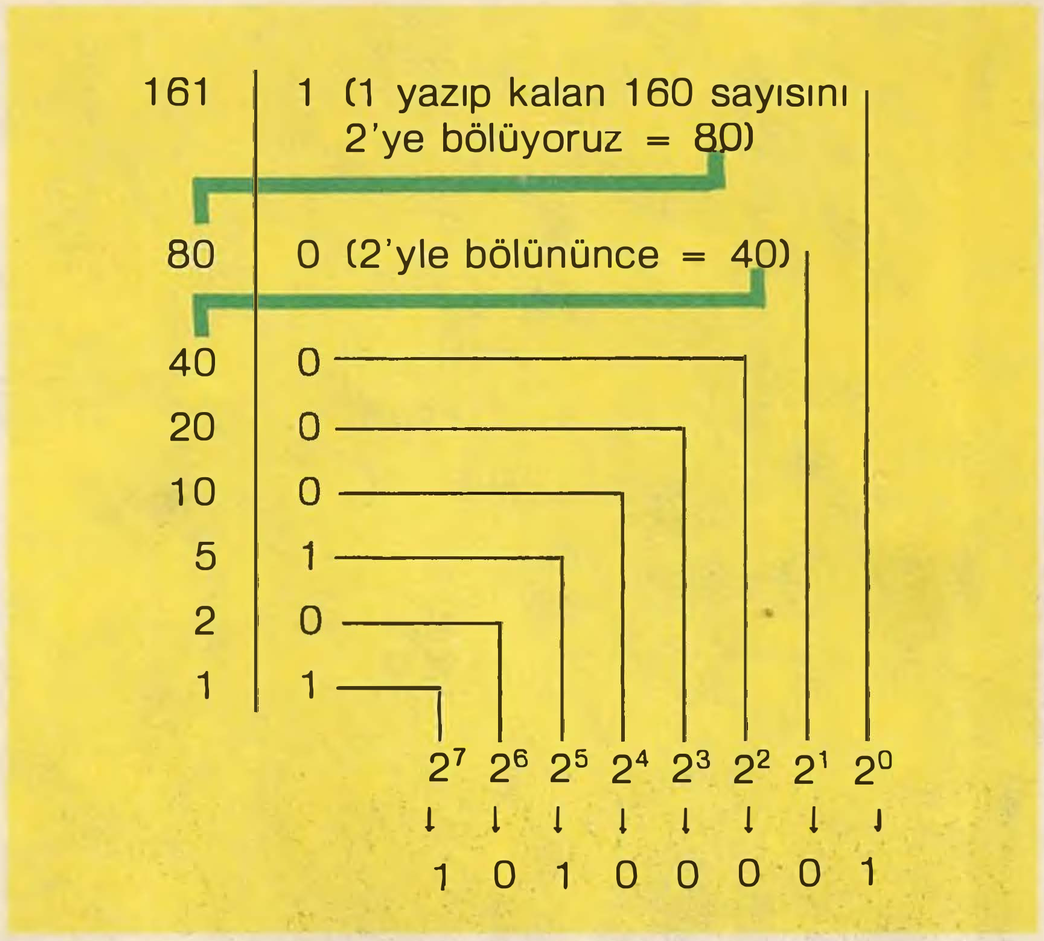
sıdır. Aynı şekilde, bir sayıyı 2, 8, ya da 16 tabanlı sistemlerde yazmak o sayının değerini değiştirmez.

Bilgisayara gelince, o hep ve yalnızca, ''makine dili'' diye de anılan 2 tabanlı sistemi kullanır. Bu nokta unutulmamalı:

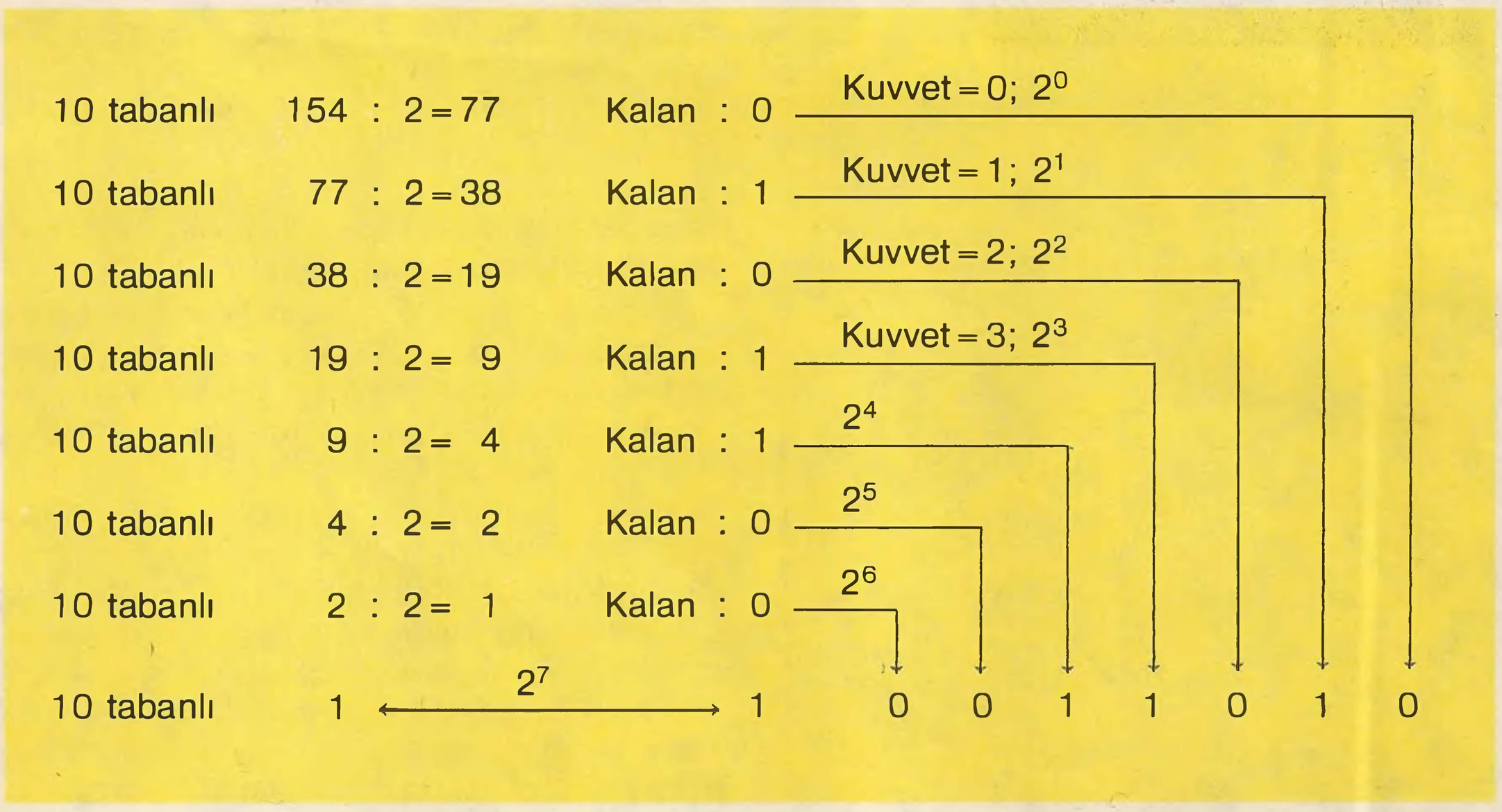
SAYISAL BİR AYGIT, İŞLEMLERİNİ HEP 2 TABANINDA YÜRÜTÜR.

2 tabanlı bir sayının 10 tabanındaki değerinin nasıl bulunabileceğini gördük. Şimdi de bu işlemin tersini görelim: 10 tabanlı bir sayıyı 2 tabanına nasıl çevirebiliriz?

Bunun için 10 tabanlı sayı, 1 elde edinceye kadar 2 ile bölünür. Her bölmede, kalan varsa 1, yoksa 0 kaydedilir. Örnek olarak 10 tabanlı 154 sayısını ele alalım:



görüldüğü gibi 10100001 değerini buluruz.



Kalanları yerlerine koyarak, 10 tabanındaki 154'ün karşılığını 2 tabanında 1 0 0 1 1 0 1 0 olarak buluruz.

Aşağıdaki adımlar bu karmaşık ve uzun işlemi hızlandırır:

A / Sayı çiftse, 0 yazın.
B / Sayı tekse, 1 yazın ve sayıdan da 1 çıkarın.
C / Kalan sayıyı 2'ye bölün.
D / Sonuç 1 değilse, elde edilen değerle A'dan başlayın.

Örneğin 10 tabanlı 161 sayısını ele alırsak:

Sağlama yapmak için bu sayıyı tekrar 10 tabanına çevirelim:

1	0	1	0	0	0	0	1
2^7	+	2^5			+		2^0
128	+	32			+		1

Görüldüğü gibi toplam 10 tabanlı 161 sayısını veriyor.

16 Tabanlı Sayı Sistemi

Yukarıda bilgisayarın, genelde 8 tabanlı sayı sistemini ya da 8'in katları (16, 32) kadar

2 tabanlı sistemin birleşmesinden oluşan sayı sistemlerini kullandığını söylemiştik. Bu sistemler için her biri 1 ya da 0 değerini alabilen 8 (ya da 16, ya da 32) tane simge gerekir. Bu da sayıların yazılmasını çok uzatır.

Bu nedenle daha tıkız bir sistem olan 16 tabanlı sayı sistemine başvurulur. *Ancak makinenin içinde hâlâ iki tabanlı sayı sistemi kullanılmaktadır: 16 tabanlı sistem yalnızca bir gösterim biçimi olarak, bizim için daha elverişlidir.*

Bu sistemde birler basamağı 0'dan 15'e kadar olan değerleri alır (10 tabanlı sistemde birler basamağının 0'dan 9'a kadar olan değerleri alması gibi). Bir üst basamak 16'dan, yani taban değerinden başlar (10 tabanlı sistemde onlar basamağının değerinin 10'dan başlaması gibi). Örneğin 2 tabanlı 1 0 1 0 0 0 0 1 sayısını ele alalım (bu sayının 10 tabanında 161 ettiğini görmüştük); simgelerden her birinin 2'nin bir kuvvetine tekabül ettiğini biliyoruz.

Şimdi bu sayıyı 2'nin tekabül eden kuvvetlerinin gösterildiği tabloya yerleştirelim:

	7	6	5	4	3	2	1	0
Üst								
Kuvvetin değeri	128	64	32	16	8	4	2	1
2 tabanlı sayı	1	0	1	0	0	0	0	1
		X				Y		

Dört basamaklı 2 tabanlı bir sayı ile ifade edilebilecek en yüksek değer 15'tir. 16 değerini yazabilmek için ikinci dört basamaklı kümeye geçilmesi gerekir (yani yukarıdaki tabloda X ile gösterilen kümeye).

Buradan şunu çıkarabiliriz: 2 tabanlı sistemde basamakları dörder dörder kümelersek, 16 tabanlı sistemi elde ederiz. Demek ki;

- **16 tabanlı sistemin birler basamağı olan (Y) kümesi 1'den 15'e kadarki değerlere yazma olanağı sağlar (tıpkı 10 tabanlı sistemde birler basamağının taban değerinden 1 küçük olan 9'a kadar yazma olanağı verdiği gibi).**
- **Bir üstteki (X) kümesi taban değeri olan 16 sayısı ile başlar. Bu basamakta yazılabilecek en küçük değer hep taban değerine tekabül eder.**

Önce 1 0 1 0 0 0 0 1 sayısının sağ yarısını, yani 0 0 0 1'i ele alalım. Görüldüğü gibi bu sayıya 3, 2, 1, 0 üstleri ve 8, 4, 2, 1 konum ağırlıkları tekabül etmektedir.

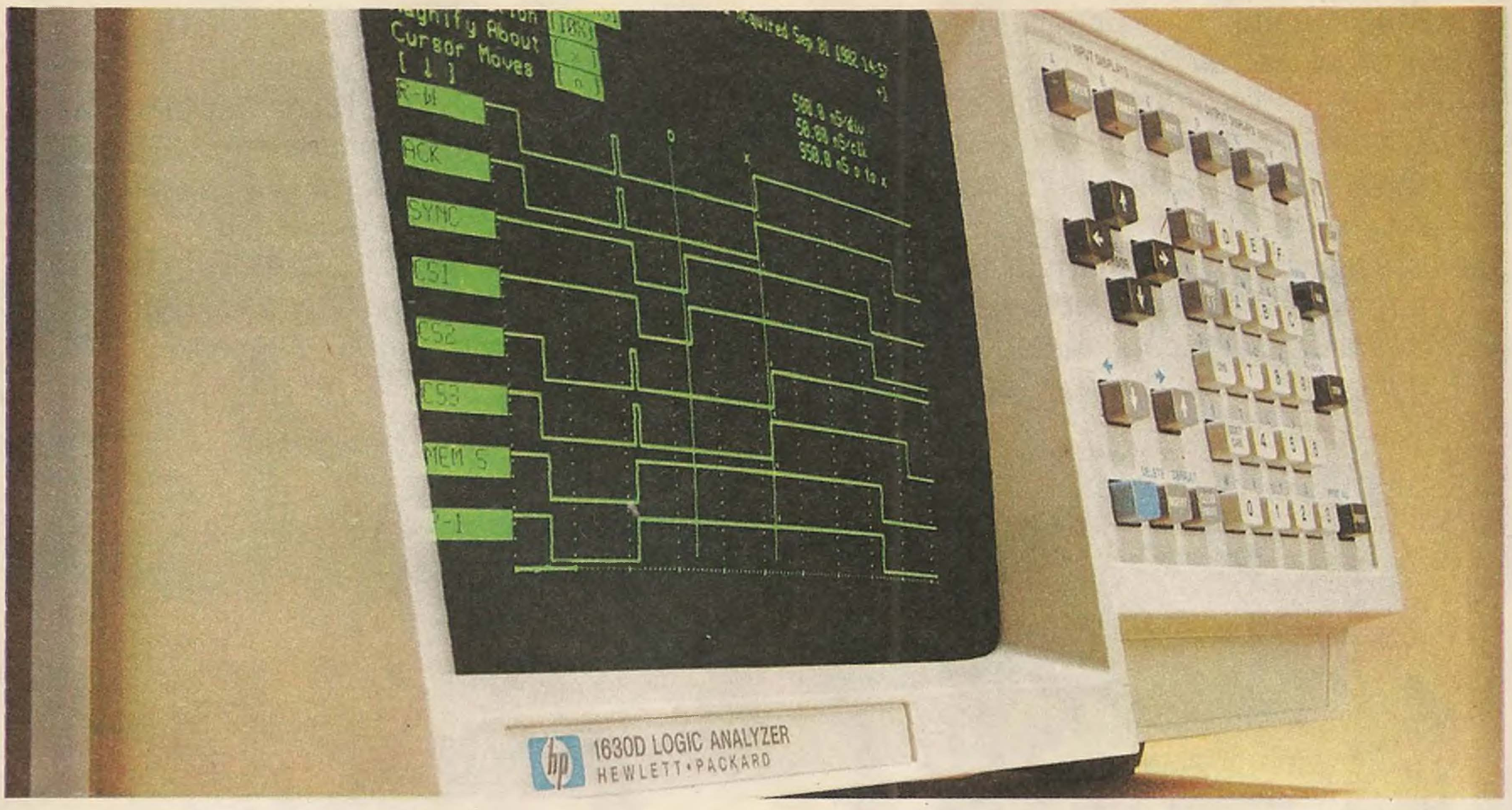
Şimdi bu sayının her basamağına 1 değeri vererek yazabileceğimiz en yüksek sayıyı elde edelim:

2 tabanlı 1 1 1 1 = 8 + 4 + 2 + 1 = 15 (10 tabanlı). 10 tabanlı 16 sayısını yazabilmek için bir üst basamağa, yani sayının sol yarısına geçmek gerekir.

2 tabanlı sayı sisteminin 4 basamağından oluşan her küme, 16 tabanlı sistemin bir basamağına tekabül eder, yani 0 ilâ 15 arasındaki değerlerden birini alır.

Buna göre 2 tabanlı 1 0 1 0 0 0 0 1 sayısını yeniden yazalım:

	İkinci 16 tabanlı basamak				Birinci 16 tabanlı basamak			
Üst	3	2	1	0	3	2	1	0
Kuvvetin değeri	8	4	2	1	8	4	2	1
2 tabanlı sayı	1	0	1	0	0	0	0	1
	X				Y			



HP 1630D modeli bir mantık çözümcüsü; sayısal devrelerin mantıksal durumunu gösteriyor.

Artık X ve Y'nin 16 tabanındaki değerleri bir bakışta hesaplanabilir:

Y = 1 çünkü bu küme yalnızca 2^0 kuvvetini içermektedir ki, bu da 1'a eşittir.

X = 10 çünkü 2^3 ve 2^1 kuvvetlerini içerir ki, bu da 8 + 2 = 10 eder.

Öyleyse toplam sayı şöyle yazılmalıdır:

	X	Y
XY	10	1

Ancak, bu yazılış biçimi yeterince açık değildir, çünkü bu yazılış biçiminden 10 sayısının tamamen X basamağına mı ait olduğunu, yoksa X basamağının 0 değerini taşıyıp, bir de 1 değerine sahip bir Z basamağının mı bulunduğunu kestiremeyiz.

Bu elverişsiz durumu ortadan kaldırmak için, 16 tabanında değeri 9'dan büyük olan rakamların harflerle simgelenmesine karar verilmiştir. Şöyle:

Simge 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 tabanındaki değeri
0 1 2 3 4 5 6 7 8 9 A B C D E F

Öyleyse 10 değeri A simgesi ile gösterilecektir. 2 tabanlı 1 0 1 0 0 0 0 1 sayısı ise şöyle yazılmalıdır:

1 0 1 0 = **A**　　0 0 0 1 = **1**

Bir kez daha kısaca gözden geçirelim:

10 tabanlı sayı	**161**
2 tabanlı sayı	**10100001**
16 tabanlı sayı	**A1**

Görüldüğü gibi 16 tabanlı sayı sisteminin yazılış biçimi daha tıkız ve kullanışlıdır.

Son olarak bir de 215 sayısını ele alalım. Önce 2 tabanına çevirelim:

215	1
107	1
53	1
26	0
13	1
6	0
3	1
1	1

215 = 1 1 0 1 0 1 1 1

Şimdi bu sayıları 16 tabanına çevirme tablomuza yerleştiriyoruz.

16 tabanlı sistemde her basamağın, 2 tabanlı sistemdeki dört basamağı kapsadığını bildiğimizden, 1 1 0 1 0 1 1 1 sayısını iki tane dört basamaklı kümeye ayırmamız gerektiğine karar veriyoruz.

Üst Kuvvetin değeri	İkinci 16 tabanlı basamak				Birinci 16 tabanlı basamak			
	8	4	2	1	8	4	2	1
	1	1	0	1	0	1	1	1
	8 + 4 + 1 = 13, yani = **D**				4 + 2 + 1 = 7, yani = **7**			

O halde sonuç olarak şunları elde ettik:

- 10 tabanlı : 215
- 16 tabanlı : D7
- 2 tabanlı : 1 1 0 1 0 1 1 1

Sağlama olarak bu işlemi ters yönde yapalım; 16 tabanlı D7 sayısından, 10 tabanlı eşdeğerini çıkaracağız.

- 7 değeri birler basamağını göstermektedir, dolayısıyla değişmez.
- D = 13 değeri tabanın birinci katıdır; dolayısıyla 16 ile çarpılması gerekmektedir.

Demek ki;
(13 × 16) + 7 = 208 + 7 = 215 (10 tabanlı).

8 Tabanlı Sayı Sistemi

Bu sistem 16 tabanlıya çok benzeyen bir sayı sistemidir. Bu kez 2 tabanlı sayıları üçer basamaklı kümelere bölüyoruz. Üstleri 2, 1, 0 olan bu basamakların konum ağırlıkları ise 4, 2, 1. Bu yolla;

"(2 tabanlı) 1 0 1 0 0 0 0 1 = 161"

sayısı için şu tabloyu elde ederiz:

Üst Kuvvetin değeri 2 tabanlı sayı	Üçüncü 8 tabanlı basamak			İkinci 8 tabanlı basamak			Birinci 8 tabanlı basamak		
	2	1	0	2	1	0	2	1	0
	4	2	1	4	2	1	4	2	1
		1	0	1	0	0	0	0	1
8 tabanlı değer		**2**			**4**			**1**	

Böylelikle şu karşılıkları bulmuş oluyoruz:
10 tabanlı : 161
2 tabanlı : 1 0 1 0 0 0 0 1
16 tabanlı : A1
8 tabanlı : 241

8 tabanlı 241 sayısını 10 tabanına çevirmek için de, 16 tabanlı sayıları çevirirken uyguladığımız kuralları uygularız:

Birler basamağı değişmez.

Sonraki rakam 8^1 ile çarpılır.

Ondan sonraki 8^2 ile çarpılır ve çarpım, her basamakta bir sonraki kuvvet alınarak bu şekil devam eder.

Bu kuralları uyguladığımızda şu sonuçları elde ediyoruz:

8 tabanlı	Çevirme işlemi	10 tabanlı
1	—	1
4	x 8	32
2	x64	128
	Sonuç	**161**

Buraya kadar söylediklerimizi anımsayalım:

— Sayısal devreler 2 tabanlı sayı sistemine göre çalışır.

— 2 tabanlı bir sayı, bir 0'lar ve 1'ler dizisi olarak yazılır.

— Her 0 ya da 1, 2'nin kuvvetlerinden birinin çarpanı olarak yorumlanır.

— 16 tabanlı sayının 10 tabanlı karşılığı, bu sayının, hangi basamaktaysa 16'nın o basamağa tekabül eden kuvvetiyle çarpımı yoluyla bulunur.

— 8 tabanlı sayı sisteminin tabanı 8'dir.

— 8 tabanlı sayının 10 tabanlı karşılığı, bu sayının, hangi basamaktaysa 8'in o basamağa tekabül eden kuvvetiyle çarpımı yoluyla bulunur.

2 Tabanlı Sayı Sisteminde İşlemler

8 ve 16 tabanlı sayı sistemlerine, kullanılan simge miktarından tasarruf etmek için başvurulduğunu söylemiştik. Ancak bilgisayar doğrudan doğruya bu sistemleri kullanamaz; bütün hesaplarını 2 tabanlı sistemde yapar.

Bir program yazarken, bazen makineye verilen komutların sonuçlarını ''elle'' kontrol etmek gerekebilir. Bu yüzden 2 tabanında toplama ve çıkarmanın nasıl yapıldığını bilmekte yarar vardır.

Ancak ileride de göreceğimiz gibi, bu kontroller, matematiksel işlemlerin doğruluğunu ölçmek için değil (zaten makine bunları bizden çok daha hızlı ve güvenilir bir biçimde yapar), mantıksal işlemlerde özel birtakım sonuçlar elde etmek için yapılır. (Mantıksal işlemlerin ne demek olduğunu ve neye yaradıklarını ileride inceleyeceğiz).

Buraya kadar söylediklerimizden, bizim kullandığımız her sayının makinenin içinde bir dizi 1 ya da 0 ile gösterildiği sonucunu çıkarabiliriz. Makine 1 ve 0 dışında hiçbir simge bilmediği için aynı işlem harflere de uygulanacaktır.

Bizimle konuşabilmek için bilgisayarın, bizim kullandığımız harf ve rakamları kendi kullandığı simgelere çevirmesi gerekir. Başka bir deyişle, bizim kullandığımız tüm çizgisel simgeler bilgisayarda 2 tabanından bir sayıya dönüştürülür.

Örneğin kullanıcı makinenin belleğine A harfini girmek istediği zaman bu harf 2 tabanlı bir sayıya çevrilecektir. B harfinin kodunu elde etmek için ise, A harfinin koduna 1 eklemek yetecektir. Bu işlem tüm alfabeye uygulanabilir. İleride daha etraflı bir biçimde inceleyeceğimiz bu yöntemle, bizim kullandığımız tüm anlatım simgelerinin bilgisayarın içinde temsil edilebilmesi mümkün olmaktadır.

HP 5006 A modeli sınama aygıtı. Bu aygıt sınanan devrelerin doğru çalışıp çalışmadığını gösterir.

TEST 1

Ansiklopedi boyunca işlenen her konuyu bir test izleyecektir. Okur soruları cevaplandırdıktan sonra kendi cevaplarını daha sonra metin içerisinde verilecek doğru cevaplarla karşılaştırabilir. Eğer cevaplarının en az yarısı doğruysa okumaya hiç güçlük çekmeden devam edebilir.

Eğer sonuç daha kötüyse ve soruların metinde verilen doğru cevaplarını anlamadıysanız, size konuyu baştan okumanızı öneririz.

---

1 / Aşağıdaki cümlelerden hangileri doğru, hangileri yanlıştır? Niçin?

☐ a - Bütün bilgisayarlar 2 tabanlı sayı sistemine göre çalışırlar.

☐ b - 8 tabanlı sayı sistemi 2 tabanlı sisteme kıyasla daha tıkızdır.

☐ c - 16 tabanlı sayı sisteminde, herhangi bir basamaktaki bir rakam, en çok 15 değerini ifade edebilir.

☐ d - 8 tabanlı sayı sisteminde 8 rakamı bulunmaz. Bu sistemde 8 sayısını nasıl ifade ederiz?

---

2 / Aşağıdaki 2 tabanlı sayıların 10 tabanındaki değerlerini bulun:

1 0 0 1 - 1 1 - 1 - 1 0 1 0 1 - 1 1 0 0 1 - 1 1 1

---

3 / Aşağıdaki 2 tabanlı sayıların 8 tabanındaki değerlerini bulun:

1 1 0 1 1 1 - 1 1 - 1 0 1 0 1 0 - 1 1 1 1 1 1

---

4 / Aşağıdaki 2 tabanlı sayıların 16 tabanındaki değerlerini bulun:

1 1 0 1 1 1 1 1 - 1 1 0 1 1 0 1 1 - 1 1 1 1 1 1 1 1 - 1 1 1 1 1 1

---

5 / Aşağıdaki 10 tabanlı sayıları 2 tabanlı sayı sistemine çevirin:

1 0 - 1 0 0 - 1 0 0 0

---

6 / Aşağıdaki dört sayı, 1 rakamının her sayıda bir önceki sayıya kıyasla bir basamak sola kaydırılmasıyla elde edilmiştir:

a - 0 0 0 1 b - 0 0 1 0 c - 0 1 0 0 d - 1 0 0 0

☐ a - Bu sayıların 10 tabanındaki değerleri nedir?

☐ b - Bu sayıların 10 tabanındaki değerlerine bakarak, 1 rakamının bir basamak sola kaydırılmasını hangi aritmetik işleme benzetebilirsiniz?

☐ c - 1 rakamının bir basamak sağa kaydırılmasıyla hangi aritmetik işlem yapılabilir?

---

7 / Aşağıdaki cümlelerden hangileri doğru, hangileri yanlıştır? Niçin?

☐ a - 16 tabanlı sayı sisteminde, 9'dan büyük sayılar alfabedeki harflerle gösterilir.

☐ b - Sayısal bir devre, sinyalin var ya da yok olmasından başka durum tanımaz.

☐ c - 16 tabanlı sayı sisteminde 9'dan büyük sayı gösterilemez.

☐ d - Sinyalin alçak düzeyden yüksek düzeye çıktığı süreye çıkış yamacı denir.

---

8 / Aşağıdaki 8 tabanlı sayıları 10 tabanına çeviriniz:

7 - 70 - 700 - 111

---

9 / Aşağıdaki 16 tabanlı sayıları 10 tabanına çeviriniz:

7 - 70 - A - AA - AAA - 3FB

---

10 / 2 tabanlı sayı sisteminde 2 basamaklı, 4 basamaklı ve 8 basamaklı rakamlarla ifade edilebilecek en yüksek değerler hangileridir?

Çözümler, s.76-77'de verilmiştir.

Bilişim Devrimi

Bilgi insan yaşamının temelini oluşturur. Bilgi ve bilginin işlenmesi, yani bilişim, *yakın gelecekte yaşamımızda giderek daha önemli bir yer tutacakmış gibi gözüküyor. Gerçi bu konuda kesin bir şey söylemek güç ama alandaki büyük teknik gelişmelerin hepsi hepsi şu son birkaç yılın araştırmalarının ürünü olduğu düşünülürse, bilişimin giderek hızla artan önemi hakkında bir fikir edinilebilir.*

Bilişim konusunda neredeyse bir buzdağı ile karşı karşıyayız; gelişmenin görünür kısmı, bizi, su altında kalan ve henüz görünmeyen kısmının boyutları hakkında yanıltabiliyor.

Bugün için bilgisayar ve mikroelektronik alanları buzdağının görünür kısmı durumundalar. Önümüzdeki on yıl içinde yaşamımızda göz kamaştırıcı değişikliklere yol açacaklar. Bu değişiklikleri üç başlık altında toplayabiliriz:

Bilgisayarın telekomünikasyon alanındaki uygulamaları (Telematik), *robotlar üzerindeki çalışmalar* (Robotik) *ve büro işlerinin bilgisayarlarca düzenlenmeye başlaması* (Bürotik).

Kısacası bilgisayarlar yaşamın bu üç önemli kesiminde, evde, fabrikada ve büroda gündelik alışkanlıklarımızı değiştirecekler. Kuşkusuz bu değişiklik ansızın değil, belirli bir süreç içerisinde yavaş yavaş gerçekleşecektir.

Telematik iki farklı çalışma alanının işbirliğinden doğmuştur: Telekomünikasyon ve Bilişim (İnformatik). *Demek ki bilişimden yararlanan telekomünikasyon ve genel olarak iletişim teknolojisinden söz edilmektedir.*

Dünyada en yaygın olarak kullanılan aygıt olan telefon (1980'in sonunda dünyada 500 milyon telefon vardı), televizör (yani ekranlı telefon) ve bunları birbirlerine bağlayabilecek kişisel bilgisayar - işte telekomünikasyon alanındaki dönüşümü gerçekleştirecek aygıtlar.

Bu dönüşüm kablo yerine optik liflerin kullanılması, telekomünikasyon alanında giderek daha sık uydulara başvurulması gibi teknolojik gelişmelere bağlıdır. Ancak bu teknolojik gelişmelerin kendileri de makine diline (yani 2 tabanlı sayı sistemine) çevrilmiş bilgilerden büyük ölçüde yararlanıyorlar. Bu tür sayısal bilgi özellikle veri taşıma hızlarını ve veri bankalarının kapasitelerini artırıyor. Bu sayede de çok uzak mesafelerde depolanmış bilgilerden çok kısa bir süre içerisinde yararlanmak mümkün oluyor.

İşte "Telematik" terimiyle geleneksel bağlantılarla ya da sözünü ettiğimiz bu yeni bağlantılarla desteklenen bir dizi hizmeti kasdediyoruz. Telematik sayesinde çok daha fazla sayıda insan, büyük veri bankalarında düzenli bir biçimde saklanmış verilerden yararlanabilecek.

Videoyazım sayesinde evinizdeki televizyon ekranından büyük veri bankalarına ulaşabilirsiniz.

İşte bu nedenlerden ötürüdür ki bütün endüstri dalları telematiğin etkisiyle olağanüstü bir hareketlilik kazanmış durumda; yepyeni dallar ortaya çıkarken geleneksel endüstri alanları da değişmeye zorlanıyor.

Daha şimdiden Telematik yeni tür televizyon ve telefonların yapılmasında ve yeni tür iletişim bağlantılarının kurulmasında kullanılıyor.

Dahası yakın bir gelecekte, telematikten kaynaklanan gelişmeler bilgi sağlayan, bilgi yaratan ve örgütleyen tüm kurumları da etkileyecektir. Örneğin geleneksel olarak bilginin yayılması işlevini gören basın, pazarını genişletmek, kendisine yeni tür iletim olanakları bulmak isteyecektir.

Örgütlenmiş bilgiden yararlanan diğer ekonomik kurumlar (örneğin burs ve kredi kurumları) telematik yoluyla, sahip oldukları bilgiyi daha verimli bir biçimde paylaşma olanaklarını araştıracaklardır.

Telematik, reklamcılara da tüketiciye ulaşma konusunda yepyeni olanaklar sunuyor.

Bütün bu saydıklarımız çok büyük sayıda ve yeni bir dizi makine gerektirmektedir - örneğin, geliştirilmiş televizörler, kişisel bilgisayarlar, telefonlar ve diğer telekomünikasyon aygıtları. Bu gereklilik endüstriye, yeni üretim olanakları sunarak hız kazandıracaktır.

Telematiğin bu alanlardaki ilk uygulamaları şimdiden bütün dünyada yaygınlaşmaya başladı.

* * *

Robotlar, artık yalnızca bilim-kurgu filmlerinde rastladığımız yaratıklar olmaktan çıktı; bugünlerde çeşitli alanlarda, özellikle de endüstri kesimlerinde, araştırmalarda, uzayda ve okyanus diplerinde bir robotla karşılaşmak mümkün.

Buralarda robotlar tekrarlanan, yorucu ve tehlikeli işlerin görülmesinde insanın yerini tutuyorlar.

Robot kullanmak üretim maliyetlerini düşürmekte, şimdiye kadar akla hayale bile gelmeyen işlerin yapılmasını mümkün kılmakta ve rekabeti arttırmaktadır. Yakın bir gelecekte duyu organlarıyla donatılmış, elektronik işlemciler ve minik-işlemcilerle yönlendirilen mekanik robotların çoğaldığını ve çeşitli işlevleri yüklendiğini göreceğiz.

Bugünün robotları genellikle iki parçadan oluşuyor: her yönde hareket edebilen mekanik bir kol ve cisimleri taşımaya ve kavramaya yarayan kullanışlı bir el. Ancak daha şimdiden bu iki temel parçaya, görme ve tad alma organlarından tutun da, konuşmalarını ve laf anlamalarını sağlayan parçalara kadar çeşitli yeni birimler ekleniyor.

Telefon bağlantılarından yararlanan teleks, iletinin isteyen kullanıcının baskı aygıtında basılmasını sağlar.

Robotlar belledikleri işleri yapabildikleri gibi yeni işler de öğrenebiliyorlar; bu ise, mekanik yapıların yanında bir de elektronik yapılarının olduğunu, yani bu makinelerin gerçek birer küçük bilgisayar olduğunu gösteriyor.

Robotların endüstri alanındaki uygulamaları sayısızdır. Özellikle otomobil endüstrisinde zaten yaygın olarak kullanılan robotlar, birçok diğer kesimde de insanın yerini almaya başladı.

* * *

Son olarak bilgisayarların bürolarda gitgide daha fazla kullanılmasıyla doğan yeni bir araştırma alanına değinelim. Bu alan ''bürotik''tir. Bu Fransızca kaynaklı kelime, çeşitli büro hizmetlerinin otomatikleştirilmesine yönelik teknik ve teknolojilere işaret ediyor; büro artı informatik, bugün kullanımı her yerde yaygınlaşan ''bürotik'' terimini vermektedir.

(Egidio Pentiraro'nun A SCUOLA CON IL COMPUTER, ''Bilgisayarlı Okullar'', 1983, adlı kitabından alınmıştır.)

2 Tabanlı Toplama

2 tabanlı aritmetik toplama yapabilmek için, bu sayı sisteminin tabanının 2 olduğunu unutmamak gerekir. Şöyle ki, nasıl 10 tabanlı toplamada sonuç 10'dan büyük çıktığında, sonuçtan 10 çıkarıp kalan değeri yazdıktan sonra elde var 1 diyorsak, 2 tabanlı toplamada da sonuç 2'den büyük olduğu takdirde, sonuçtan 2 çıkarıp elde var 1 demeliyiz.

Örneğin 2 tabanlı 1 rakamını kendisiyle toplayalım:

```
  1
+ 1
----
 10
```

1 + 1'in sonucu 2'dir. Dolayısıyla 2'den 2'yi çıkarır ve bulduğumuz 0 sayısını ilk basamağa yazar ve elde var 1 deriz. İkinci basamakta bir şey olmadığından, eldeki 1'i doğrudan doğruya ikinci basamağa yazarız.

```
  1 (2 Tabanlı) = 1 (10 Tabanlı)
+ 1 (2 Tabanlı) = 1 (10 Tabanlı)
-------------------------------
Sonuç: 10 (2 tabanlı) = 2 (10 Tabanlı)
```

Şimdi aşağıdaki şu iki 2 tabanlı sayıyı toplayalım:

```
1 0 1 1 0 1
  1 0 1 1 0
```

Toplamayı yapmadan önce bu 2 tabanlı sayıları 10 tabanına çevirelim. (Bu değerlerden sağlama yaparken yararlanacağız.)

$1\ 0\ 1\ 1\ 0\ 1 = 2^0 + 2^2 + 2^3 + 2^5 = 1 + 4 + 8 + 32 = 45$ (10 tabanlı)

$1\ 0\ 1\ 1\ 0 = 2^1 + 2^2 + 2^4 = 2 + 4 + 16 = 22$ (10 tabanlı)

$45 + 22 = 67$ (10 tabanlı)

2 tabanlı bu toplamayı incelemek için, iki sayıyı da sağdan başlayarak sütunlar halinde yerleştirelim:

Basamak	g	f	e	d	c	b	a
Birinci toplanan		1	0	1	1	0	1
İkinci toplanan			1	0	1	1	0
	1	0	0	0	0	1	1
Elde		1	1	1	1		

A sütunu ile başlayalım; toplananlar 1 ve 0, dolayısıyla sonuç 1 eder. Öyleyse elde tutulan bir sayı yoktur.

B sütunu: Toplananlar 1 ve 0, sonuç = 1; elde = 0.

C sütunu: Toplananlar 1 ve 1, sonuç = 0, elde = 1.

D sütunu: Toplananlar 1 ve 0, sonuç = 1; geçen basamaktan elde olanı (1'i) eklediğimizde yeni sonuç = 1 + 1 = 0; yeni elde = 1.

E sütunu: Toplananlar 0 ve 1, sonuç = 1; geçen basamaktan elde olan 1'i eklediğimizde yeni sonuç = 1 + 1 = 0; yeni elde = 1.

F sütunu: Toplananlar 0 ve 1, sonuç = 1; geçen basamaktan elde olan 1'i eklediğimizde yeni sonuç = 1 + 1 = 0; yeni elde = 1.

G sütunu: Toplanan yok, dolayısıyla elde son kalan 1'i yazıyoruz. Şimdi artık sonucu şu şekilde yazabiliriz:

$1000011 = 2^0 + 2^1 + 2^6 = 1 + 2 + 64 = 67$ (10 tabanlı).

8 ve 16 tabanlı toplamalar da aynı biçimde yapılabilir.

Bu sayı sistemlerinde, toplamın 8 ya da 16'ya eşit olduğu durumlarda elde 1 tutulacağının unutulmaması gerekir.

Örnek olarak deminki iki sayının 8 tabanlı sistemdeki toplamlarına bakalım. Bunun için önce bu sayıları 8 tabanına çevirmemiz gerekiyor:

Konum ağırlığı	4	2	1	4	2	1
Çevirme işleminde kullanılan 2 tabanlı karşılığı	1	0	1	1	0	1
8 tabanlı değeri	5			5		
	(4 + 1)			(4 + 1)		

Konum ağırlığı	4	2	1	4	2	1
Çevirme işleminde kullanılan 2 tabanlı karşılığı	*	1	0	1	1	0
8 tabanlı değeri	2			6		

$1\ 0\ 1\ 1\ 0\ 1 = 55$ (8 tabanlı)

$1\ 0\ 1\ 1\ 0 = 26$ (8 tabanlı)

* Birinci basamakta herhangi bir simgenin yokluğu 0 anlamına gelir. Yani, 2 tabanlı 10110 sayısı ile 010110 sayısı arasında bir fark yoktur.

Şimdi artık 8 tabanlı toplamayı şöyle yapabiliriz:

```
        5      5 (8 tabanlı)
+       2      6 (8 tabanlı)
-------------------------
1       0      3
```

Şekildeki sonucu (8 tabanlı 103'ü) şöyle elde ettik:

- Sağdan hareketle, 5 + 6 = 11; sonuç tabandan, yani 8'den büyük; dolayısıyla 11'den 8 çıkarıp 3 yazıyoruz; elde var 1. Bu durumda ikinci sütun şöyle olur: 5 + 2 + (eldeki) 1 = 8; tabanı çıkardığımızda 0 kalır; yine elde = 1.

Son sütunda hiçbir sayı olmadığı için yalnızca eldekini yazıyoruz.

Sağlama amacıyla 10 tabanına çevirelim:

8 tabanlı 103 sayısı şu şekilde yazılabilir:

```
8 tabanlı birler basamağı   = 3 × 8^0 =   3 ──┐
                                              +
8 tabanlı onlar basamağı    = 0 × 8   =   0 ──┤
                                              +
8 tabanlı yüzler basamağı   = 1 × 64  =  64 ──┘
                                              ↓
                                  Toplam      67
```

Ya da daha derli toplu bir biçimde:
(8 tabanlı) $103 = 3 \times 8^0 + 1 \times 8^2 = 3 + 64 = 67$.

2 Tabanlı Çıkarma

2 tabanıı çıkarmada da 10 tabanlı çıkarmanın kuralları uygulanır; ama ileride daha iyi anlayacağımız nedenlerden ötürü ''tümleme yöntemi'' adı verilen bir yöntemden yararlanılır.

2 tabanındaki herhangi bir sayının tümleyeni 1'lerin yerine 0, 0'ların yerine ise 1 konarak bulunur. Örneğin bu kurala göre 100101'in tümleyeni 011010'dır.

Herhangi bir sayı ile tümleyenin toplamı yalnızca 1'lerden oluşur. Örnekleyelim:

```
Sayı          1 0 0 1 0 1
Tümleyeni  +  0 1 1 0 1 0
-------------------------
Toplam        1 1 1 1 1 1
```

Bu toplama 1 eklediğimiz takdirde, bu kez 0'larla devam eden bir dizi elde ederiz:
1 1 1 1 1 1 + 1 = 1 0 0 0 0 0 0.

Tümleme yöntemini kullanarak iki tane 2 tabanlı sayı arasındaki fark aşağıdaki şekilde bulunur:

A / Daha küçük olan sayının soluna 0 ekleyerek basamaklarını daha büyük olan birinci sayının basamakları ile eşleştirin ve sonra tümleyenini alın.

B / Birinci sayı ile bulduğunuz tümleyeni toplayın. Eğer toplamın basamakları toplananların basamaklarından fazlaysa, toplamın en sonundaki 1 rakamını atın.

C / Elde ettiğiniz sayıya 1 ekleyin.*

Şimdi bu yöntemi aşağıdaki işleme uygulayalım:

birinci sayı — çıkartılan
1 0 0 0 1 1 — 1 0 1 0 1 (2 tabanlı)

10 tabanında bu fark 35—21 = 14'tür.

A Adımı:
Çıkartılan sayının basamakları birinci sayınınkinden bir eksik; dolayısıyla çıkartılana solda bir sıfır ekledikten sonra tümleyenini alıyoruz:

```
Birinci sayı      1  0  0  0  1  1
Çıkartılan        0  1  0  1  0  1
                  |  |  |  |  |  |
                  ↓  ↓  ↓  ↓  ↓  ↓
Çıkartılanın
tümleyeni     =   1  0  1  0  1  0
```

B Adımı:
Birinci sayı ile ikinci sayının tümleyenini topluyoruz:

```
Birinci sayı                  | 1 0 0 0 1 1
Çıkartılanın tümleyeni        | 1 0 1 0 1 0
                              |-----------
Toplam                  1     | 0 0 1 1 0 1
```

Toplam toplananlardan uzun olduğu için ilk basamağı atıyoruz.

C Adımı:
Bulduğumuz sayıya 1 ekliyoruz:

```
B'nin sonucu      0 0 1 1 0 1
                +           1
                -------------
Sonuç             0 0 1 1 1 0  = 14 (10 tabanlı)
```

---

* Eğer bu toplamada da, toplamın basamakları toplananların basamaklarından fazla ise, yine B adımında yaptığımız gibi en soldaki basamağı atın.

Bu yöntem yukarıda anlatıldığı biçimiyle, olduğundan daha külfetliymiş gibi görünebilir. Yeterince pratik kazandıktan sonra saydığımız adımların çoğu akıldan yapılabilir.

Bir örnek daha alarak yöntemi biraz daha kestirme bir biçimde uygulayalım:

226—174 = 52 (10 tabanlı) =
1 1 1 0 0 0 1 0 — 1 0 1 0 1 1 1 0

Çıkartılanın tümleyeni **(0 1 0 1 0 0 0 1)** ile birinci sayıyı topladığımızda:

```
  | 1 1 1 0 0 0 1 0
  | 0 1 0 1 0 0 0 1
  ---------------------
1 | 0 0 1 1 0 0 1 1
  |               1
  ---------------------
  | 0 0 1 1 0 1 0 0 =
```

= 52 (10 tabanlı)

sonucunu elde ederiz.

Bundan sonraki bölümlerde okura kolaylık sağlamak üzere sayfa 63 ilâ 66 arasındaki tabloda, 1'den 225'e kadar olan sayıların 8 tabanlı ve 16 tabanlı karşılıkları verilmiştir.

66 ve 67. sayfalarda yer alan Tablo 2'de ise 31100 ile 31150 arasındaki sayıların karşılıkları verilmiştir.

Bu veriler bilgisayara verilen tek bir komutla, saniyenin çok küçük bir bölümünde sağlanmıştır. Bu hızı takdir edebilmek için, örneğin 31142 sayısının karşılıklarını elle bulmaya çalışın; ne kadar hızlı ve dikkatli olursanız olun yine de birkaç dakika sarfetmek zorunda kalacaksınız.

HP 9000 modeli bilgisayar. Ekranda bir mantık devresinde sayısal sinyallerin akışı gözüküyor.

TABLO 1

10 TABANLI	2 TABANLI	8 TABANLI	16 TABANLI
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1	1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	2	2
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	3	3
4	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	4	4
5	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	5	5
6	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	6	6
7	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	7	7
8	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	10	8
9	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1	11	9
10	0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0	12	A
11	0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1	13	B
12	0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0	14	C
13	0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1	15	D
14	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0	16	E
15	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	17	F
16	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	20	10
17	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1	21	11
18	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0	22	12
19	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1	23	13
20	0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0	24	14
21	0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1	25	15
22	0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0	26	16
23	0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1	27	17
24	0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0	30	18
25	0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1	31	19
26	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0	32	1A
27	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1	33	1B
28	0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0	34	1C
29	0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1	35	1D
30	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	36	1E
31	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1	37	1F
32	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	40	20
33	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1	41	21
34	0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	42	22

10 TABANLI	2 TABANLI	8 TABANLI	16 TABANLI
35	0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1	43	23
36	0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0	44	24
37	0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1	45	25
38	0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0	46	26
39	0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1	47	27
40	0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0	50	28
41	0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1	51	29
42	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	52	2A
43	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1	53	2B
44	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0	54	2C
45	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1	55	2D
46	0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0	56	2E
47	0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1	57	2F
48	0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0	60	30
49	0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1	61	31
50	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0	62	32
51	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1	63	33
52	0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0	64	34
53	0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1	65	35
54	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	66	36
55	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1	67	37
56	0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0	70	38
57	0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1	71	39
58	0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0	72	3A
59	0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1	73	3B
60	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0	74	3C
61	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1	75	3D
62	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0	76	3E
63	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1	77	3F
64	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	100	40
65	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1	101	41
66	0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0	102	42
67	0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1	103	43
68	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0	104	44
69	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1	105	45

70	0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0	106	46
.....			
215	0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1	327	D7
216	0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0	330	D8
217	0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1	331	D9
218	0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0	332	DA
219	0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1	333	DB
220	0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0	334	DC
221	0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1	335	DD
222	0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0	336	DE
223	0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1	337	DF
224	0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0	340	E0
225	0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1	341	E1
226	0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0	342	E2
227	0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1	343	E3
228	0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0	344	E4
229	0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1	345	E5
230	0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0	346	E6
231	0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1	347	E7
232	0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0	350	E8
233	0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1	351	E9
234	0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0	352	EA
235	0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1	353	EB
236	0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0	354	EC
237	0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1	355	ED
238	0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0	356	EE
239	0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1	357	EF
240	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0	360	F0
241	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1	361	F1
242	0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0	362	F2
243	0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1	363	F3
244	0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0	364	F4
245	0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1	365	F5
246	0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0	366	F6
247	0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1	367	F7

10 TABANLI	2 TABANLI	8 TABANLI	16 TABANLI
248	0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0	370	F8
249	0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1	371	F9
250	0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0	372	FA
251	0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1	373	FB
252	0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0	374	FC
253	0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1	375	FD
254	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0	376	FE
255	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	377	FF

TABLO 2

10 TABANLI	2 TABANLI	8 TABANLI	16 TABANLI
31100	0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0	74574	797C
31101	0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1	74575	797D
31102	0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0	74576	797E
31103	0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1	74577	797F
31104	0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0	74600	7980
31105	0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1	74601	7981
31106	0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0	74602	7982
31107	0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 1	74603	7983
31108	0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 0	74604	7984
31109	0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 1	74605	7985
31110	0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0	74606	7986
31111	0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1	74607	7987
31112	0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 0	74610	7988
31113	0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1	74611	7989
31114	0 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0	74612	798A
31115	0 1 1 1 1 0 0 1 1 0 0 0 1 0 1 1	74613	798B

31116	0 1 1 1 1 0 0 1 1 0 0 0 1 1 0 0	74614	798C
31117	0 1 1 1 1 0 0 1 1 0 0 0 1 1 0 1	74615	798D
31118	0 1 1 1 1 0 0 1 1 0 0 0 1 1 1 0	74616	798E
31119	0 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1	74617	798F
31120	0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0	74620	7990
31121	0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 1	74621	7991
31122	0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 0	74622	7992
31123	0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 1	74623	7993
31124	0 1 1 1 1 0 0 1 1 0 0 1 0 1 0 0	74624	7994
31125	0 1 1 1 1 0 0 1 1 0 0 1 0 1 0 1	74625	7995
31126	0 1 1 1 1 0 0 1 1 0 0 1 0 1 1 0	74626	7996
31127	0 1 1 1 1 0 0 1 1 0 0 1 0 1 1 1	74627	7997
31128	0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0	74630	7998
31129	0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1	74631	7999
31130	0 1 1 1 1 0 0 1 1 0 0 1 1 0 1 0	74632	799A
31131	0 1 1 1 1 0 0 1 1 0 0 1 1 0 1 1	74633	799B
31132	0 1 1 1 1 0 0 1 1 0 0 1 1 1 0 0	74634	799C
31133	0 1 1 1 1 0 0 1 1 0 0 1 1 1 0 1	74635	799D
31134	0 1 1 1 1 0 0 1 1 0 0 1 1 1 1 0	74636	799E
31135	0 1 1 1 1 0 0 1 1 0 0 1 1 1 1 1	74637	799F
31136	0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0	74640	79A0
31137	0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 1	74641	79A1
31138	0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 0	74642	79A2
31139	0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 1	74643	79A3
31140	0 1 1 1 1 0 0 1 1 0 1 0 0 1 0 0	74644	79A4
31141	0 1 1 1 1 0 0 1 1 0 1 0 0 1 0 1	74645	79A5
31142	0 1 1 1 1 0 0 1 1 0 1 0 0 1 1 0	74646	79A6
31143	0 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1	74647	79A7
31144	0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 0	74650	79A8
31145	0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1	74651	79A9
31146	0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 0	74652	79AA
31147	0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 1	74653	79AB
31148	0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 0	74654	79AC
31149	0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 1	74655	79AD
31150	0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0	74656	79AE

Televizyonunuzdaki Terminal

Normal bir televizyon ekranı bugün bir terminal olarak kullanılabilir. Böylelikle bir veri iletim ağına ulaşarak çeşitli veri bankalarında depolanmış bilgilerden yararlanmak mümkün olur.

Batı'da uzun zamandır, bu tür veri iletim ağları yoluyla hava durumu, trafik koşulları, saat ayarı gibi sınırlı birtakım konularda bilgiler içeren veri bankalarına ulaşılabiliyordu. Bunun için yalnızca bir telefon numarası çevirmek yeterli oluyordu. Ancak veri işleme ve bilgi taşıma sistemlerinin birleştirilmesi olanağı doğunca, bu tür hizmetlerde son yıllarda nitel bir sıçrama görüldü. Bugün Avrupa ve Amerika'da birçok televizyon sahibi, televizyonlarının uzaktan kumanda aletleri dışında hiçbir özel araca gerek duymaksızın istedikleri konuda istedikleri ayrıntı düzeyinde çeşitli bilgi kütüklerini çağırabilirler. Tuşlara dokunur dokunmaz istedikleri konu gözlerinin önünde canlanacaktır.

Bugüne kadar bu tür hizmetleri gören iki büyük sistem yaratılmıştır: Teleyazım (teletext) ve videoyazım (videotext). 1970'lerin başında geliştirilen teleyazım normal televizyon ağı ile yayınlanır; videoyazım ise telefon şebekesi aracılığı ile iletilir. Her iki durumda da bilgilerin ekran üzerinde görünebilmesi için özel bir şifre çözücüye gerek vardır.

Teleyazım

Teleyazım sayesinde ekranın yalnızca bir bölümünden de yararlanılabilir; böylelikle aynı anda hem bir televizyon programı izlemek hem

Metinde sözü edilen Prestel sistemi çok çeşitli konulardaki verilere telefonla ulaşılmasını mümkün kılar.

de istenen herhangi bir konudaki bilgileri ekranın bir bölümünde görmek mümkün olur. Kullanıcı uzaktan kumanda aletinin tuşlarına basar basmaz, istediği konudaki bilgiler kütüğünün birinci sayfası ekranda belirir. Bu ilk sayfayı bir ''İçindekiler'' sayfası olarak düşünebiliriz; her konu başlığının yanında bir numara yer almaktadır; artık kullanıcıya kalan, istediği konu başlığının yanında yer alan numarayı uzaktan kumanda aletinin tuşlarına basarak bildirmekten ibarettir. Bunun üzerine istediği konu ekranda belirecek ve yeni bir tuşa basılıncaya kadar da ekranda kalacaktır.

Teleyazımın 'sayfaları' bu konuda uzmanlaşmış bir yazarlar ekibi tarafından hazırlanmaktadır. Bu sayfalar hazırlandıkça ''hizmetçi'' adı verilen merkez bilgisayara yüklenir. Yüklenen her sayfa 40 karakterlik 23 satırdan oluşmak zorundadır.

Yazının dizgisi yedi renkte olabilir - beyaz, kırmızı, mavi, yeşil, pembe, açık mavi ve sarı. Her karakter sayısal bir kodla iletilir. Alıcı cihazda bu kodlar için özel birtakım devreler bulunur. Bu devreler gelen koda göre, mümkün 70 harf ve sayıdan hangisinin gösterilmesi gerektiğine karar verir. Kodlanmış ikinci bir komut ise karakterin hangi renkte gösterileceğini bildirir. Bir teknisyen bu sorunu 70 karakterlik bir ''paket''in özel olarak bölümlenmiş bir ''alan''a yerleştirilmesi sorunu olarak düşünebilir. Tabiî, birtakım özel etkiler yaratmak, örneğin büyük harf göstermek için tek bir karakter birkaç bölüme birden yayılabilir.

Sayfalar, çoğunlukla normal televizyon yayınının olduğu saatlerde ardarda ve aralıksız olarak iletilir. 24 satırlık bir sayfanın iletimi yaklaşık olarak saniyenin dörtte biri kadar bir zaman alır. Bu süre ihmal edilebilir görünse de aslında bir kusurdur. Teleyazımın herhangi bir konu hakkında içerdiği malzeme yüz sayfadan aşağı olamaz. Bütün bu konu satır satır, sayfa sayfa ve sırayla iletilecektir. Bu esnada belli bir sayfayı isteyen okuyucu, bir iki dakika kadar beklemek zorunda kalabilir. Tabiî beklediği bu süre içerisinde ekran boş kalmayacaktır. Bir bellek devresi yeni sayfa gelinceye kadar ekrandaki eski resmi muhafaza eder.

Bu ölü zaman teleyazım sisteminin başlıca aksaklıklarından biridir. Ancak şimdilik deneme aşamasında bulunan çalışmalar bu kusurun kısa zamanda giderilebileceğini düşündürüyorlar.

Teleyazım alanındaki araştırmalar iki ana yönde gelişiyor.

Bunlardan birincisi Teleyazılım diye adlandırılan tasarı üzerindeki çalışmalar. Bu tasarı alıcı cihazlara bir minik işlemcinin eklenmesini öngörüyor. Böylelikle ekrandan doğrudan doğruya merkez bilgisayara ya da bir bilgisayarlar ağına ulaşmak mümkün olacak. Başka bir deyişle, ekranımızı, istediğimiz programları işletebilecek bir ''akıllı terminal'' olarak kullanmak mümkün olacak.

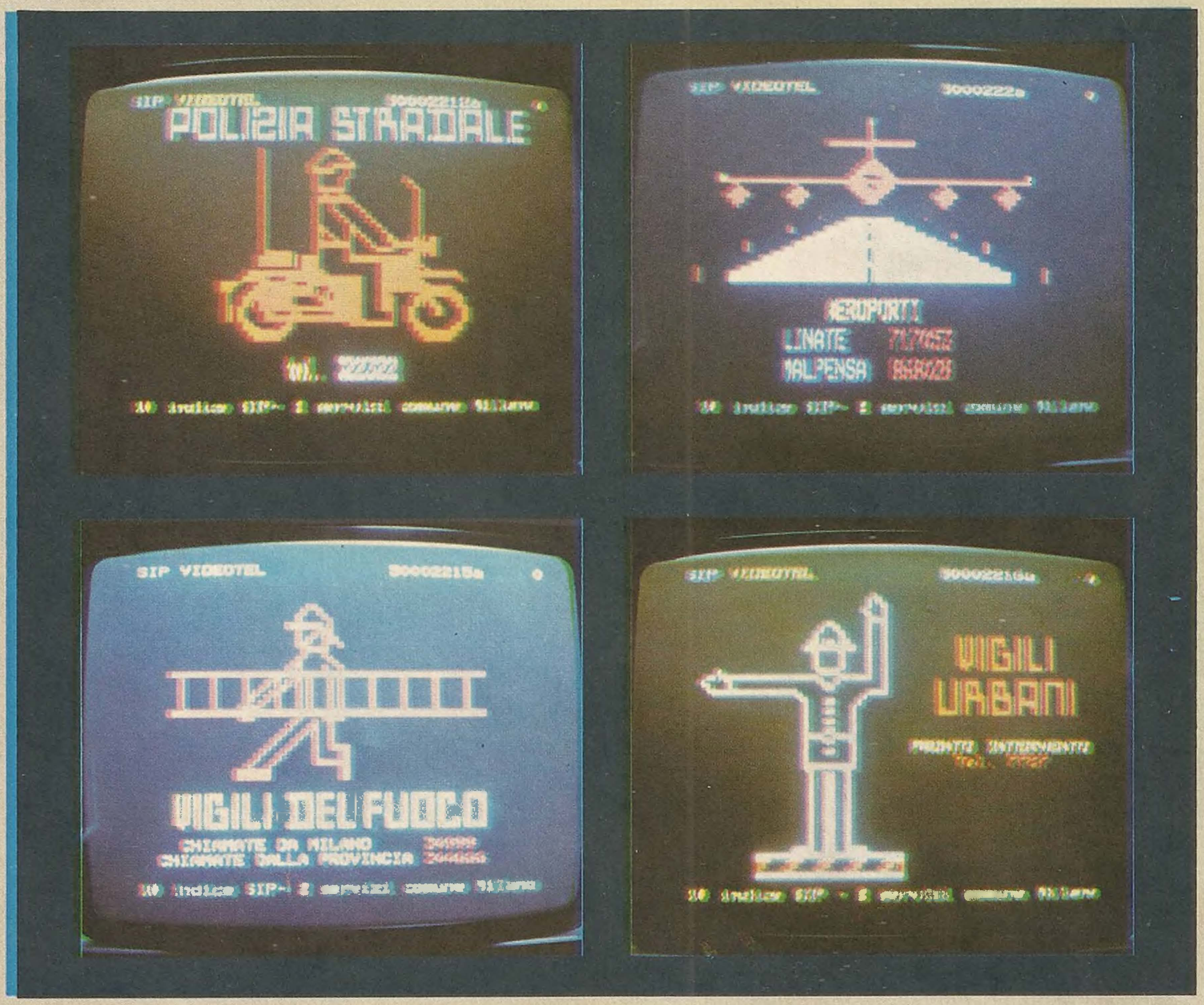
İkinci tasarı ise teleyazımın normal televizyon yayınlarını da kapsayacak biçimde genişletilmesini öngörüyor. Bu sayede, istendiği takdirde, normal televizyon yayınları otomatik olarak altyazılanabilecek. Bu, işitme bozukluğu olanlara büyük yararlar sağlayabileceği gibi, yabancı yayınların izlenmesini de olanaklı kılacaktır.

Videoyazım

Videoyazım ise teleyazımın olanaklarını daha da zenginleştiren bir sistemdir. Bu sistem sayesinde televizyonlu terminaller telefon aracılığıyla ana bilgisayara bağlanabilecek. Örneğin İngiltere'de geliştirilen Prestel videoyazım sistemi merkez bilgisayarla karşılıklı etkileşim olanağı sağlayan ilk sistemdir.

Bu sistemde de, aynı teleyazım sisteminde olduğu gibi, televizyonun önce ayarlanması ve sonra uzaktan kumanda aletinin tuşlarından birine basılması gerekmektedir. Bunun üzerine televizyon ekranı, otomatik olarak telefon bağlantısını kurup merkez bilgisayara ulaşır. Daha az kullanışlı kimi sistemlerde ise kullanıcının kendisi numarayı çevirmek zorundadır. Numarayı çevirdikten sonra santral tarafından yayınlanan bilgilerin genel dizinini elde eder. Bu noktada artık ilgilendiği konu için gereken anahtar kelimeyi uzaktan kumanda aleti ile ekrana bildirebilir. Bunun üzerine ekranda daha ayrıntılı ikinci bir dizin belirir. Böylelikle ilgilendiği konunun ayrıntılarını daha titiz bir biçimde tarar ve bir kez daha gerekli anahtar kelimeyi ekrana bildirir. Bu işlemler kullanıcı istediği ayrıntı düzeyine varana kadar tekrarlanabilir.

Videoyazımın olanakları sonsuz gibi görünmektedir. Teleyazımdan farkı kullanıcının istemediği birtakım sayfaları da izlemek zorunda kalmamasıdır. Çok sayıda ve okurun ilgi-



Videoyazım sistemlerinin sunduğu hizmetlerden birkaç örnek.

sine göre belirlenmiş bilgi derhal çağrılabilir. Bu sistemin sınırları sadece merkez belleğin kapasitesi tarafından belirlenir.

Daha önce sözünü ettiğimiz Prestel sisteminde bilgiler, çeşitli alanlarda yetkili 150 uzman tarafından hazırlanmaktadır. Amaç çok çeşitli konularda kullanıcıların yararlanabileceği bilgileri onların yararlanabileceği bir biçimde derlemektir. Bu konular boş zamanları değerlendirme faaliyetlerinden tutun da ayrıntılı fiyat istatistiklerine kadar çeşitli alanları kapsar.

Bu sistemi kullanan her kullanıcının bir hesabı vardır ve kullanıcı okuduğu sayfa başına ücret öder. Her sayfanın fiyatı ekranda sayfa ile birlikte gösterilir. Kullanıcı da istediği anda hesabının son durumunu ekrandan izleme olanağına sahiptir.

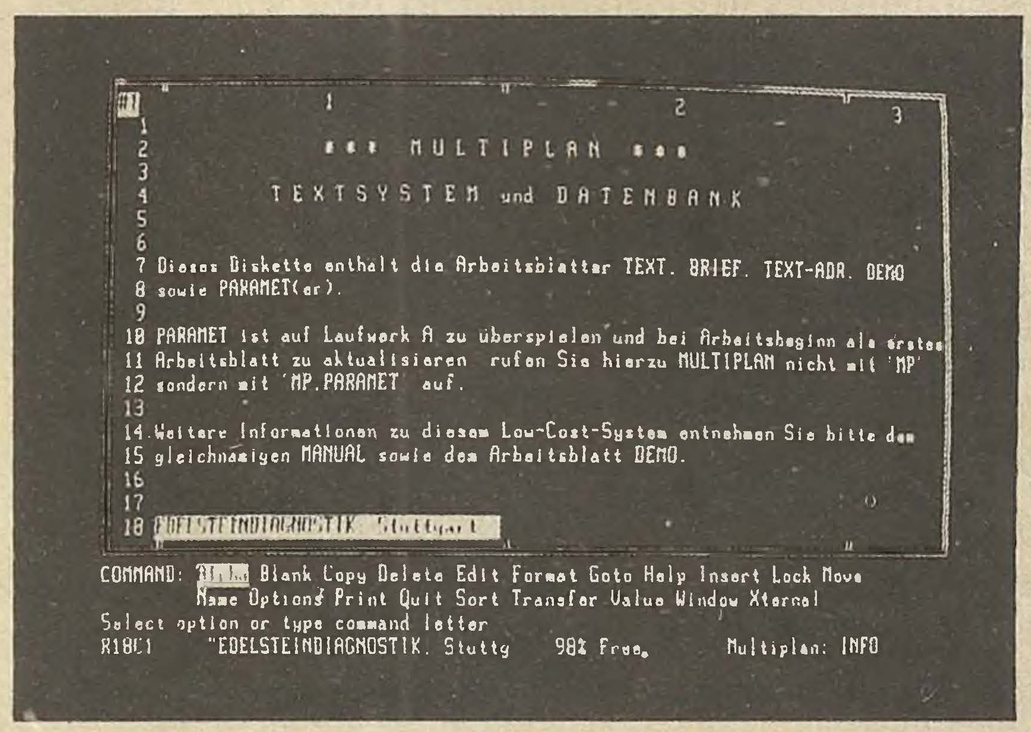
Bunun yanında Fransa'da kurulmuş olan Videotex gibi sistemler, Lockheed gibi büyük firmalar ya da Amerika'da MIT gibi büyük üniversiteler özel sektörün ve şahısların başvurulabileceği büyük birtakım veri bankaları da yönetmektedirler.

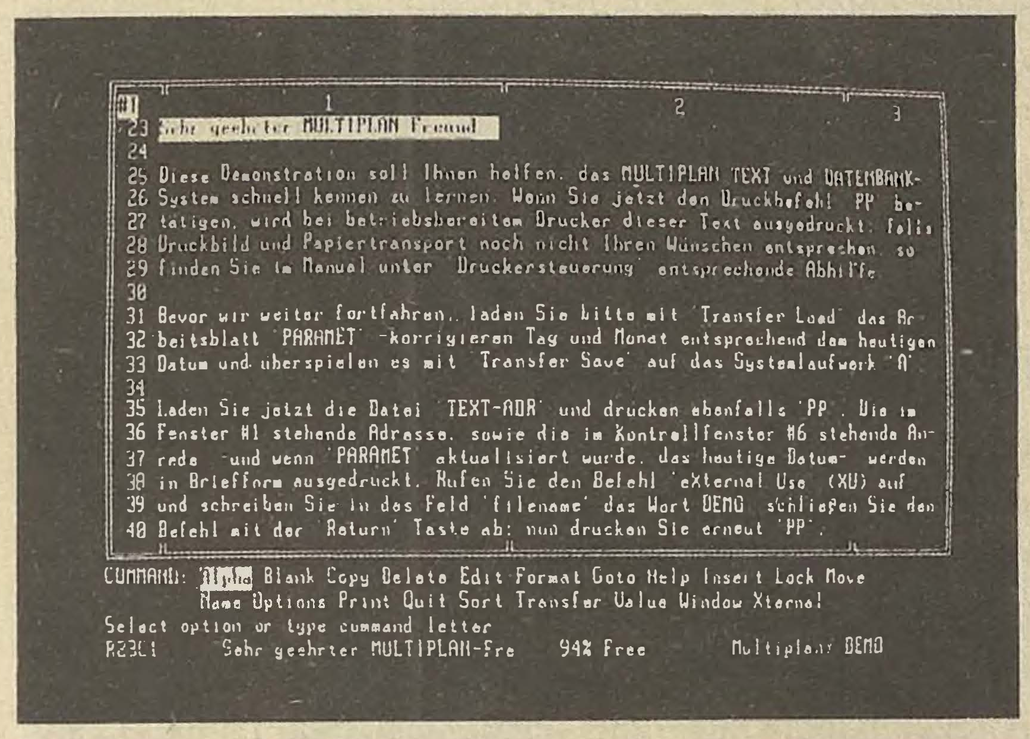
Kuşkusuz böylesi hizmetler daha çok iş hayatına ve ticarete yöneliktir. Başlıca amaçları güvenilir bir biçimde seçilmiş ve iyi derlenmiş verilerin, bu verilere ihtiyacı olan yerlere hızla ulaşmasını sağlamaktır. Ayrıca bu sistemlerde belirli ölçüde bir gizlilik ve güvenlik sağlamak da mümkündür. Kullanıcılara bilgisayar için bir komut işlevi görecek gizli bir şifre verilir. Ayrıca bilgisayara belirli malzemeleri ancak belirli şifrelerin verilmesi halinde açması talimatı verilir. Böylelikle gizlilik sağlanmış olur. Bu şifrelere bilgisayar dilinde ''parola'' deniyor. Ancak parola kullanılacaksa, alıcı cihazın da mutlaka bir minik-işlemcisinin olması gerekir.

Bu sistemlerin bir kısmı, örneğin Fransa'daki Videotex, verileri tarama olanağından başka merkez bilgisayarla etkileşme olanağı da sağlamaktadır.

Böyle bir etkileşim olanağı, örneğin tıbbî teşhis programları için çok önemlidir. Bugün bu programlar Avrupa'daki birçok hastanelerde kullanılıyorlar. Kuşkusuz bu programların işlevi hekimlerin yerini almak değil; teşhis konusundaki olasılıkları sınırlandırarak ve hastayı ne tür bir uzmana gitmesi gerektiği konusunda yönlendirerek hekimlere zaman kazandırmak. Böylelikle hekimler doğrudan doğruya hasta tedavisine çok daha fazla vakit ayırabiliyorlar.

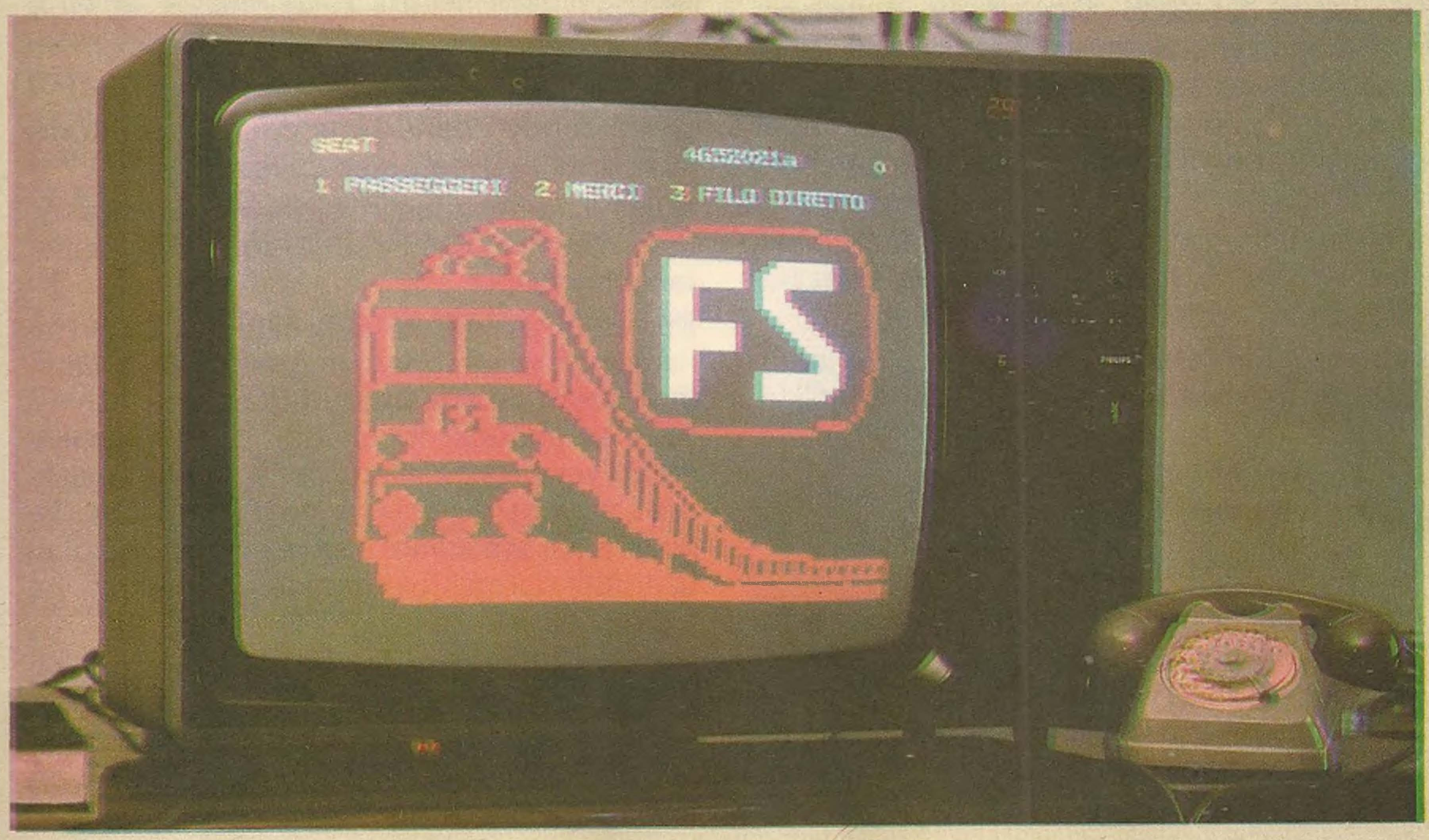
Karşılıklı etkileşim olanağının daha da artması için alıcı cihaza bir minik-işlemci ile bir klavyenin de eklenmesi gerekiyor. Böylesi bir donanımın kişisel bir bilgisayardan bile ucuz olmak gibi bir avantajı var ve çok sınırlı bir ölçüde de olsa programlama olanağı tanıyor. Batı'da artık banka hizmetlerinde sık sık bu makinelerden yararlanıldığını, kişilerin ve firmaların bir hesaptan diğerine mevduat transfer ederken ya da yeni bir hesaba para yatırırken bunlara başvurduğunu söylersek, bu makinelerin kullanılış alanları ve sınırları hakkında bir fikir vermiş oluruz.





Teleyazım ve videoyazım sistemlerinde "sayfa"lar değişik ayrıntı düzeylerinde bilgiler içerir.

Avrupa'da kullanılan videoyazım sistemlerinden biri olan Videotel sistemi tren tarifeleri ve yolculuk hakkında bilgi veriyor.



Mantık Ögeleri

Bilgisayar, işlemlerinden bazılarını şimdiye kadar gözden geçirdiğimiz devrelerden farklı bazı devrelerle yapar. Bu bölümde **kablolu mantık** ya da **bağlantı mantığı** adı verilen bu farklı tekniğin temel özelliklerini inceleyeceğiz.

Bu konu doğrudan doğruya bir bilgisayar konusu olmamakla birlikte, bu makinelerin yapısı ve işleyişlerinin anlaşılması için elzem bir temel oluşturmaktadır.

Konuya bir iki mantık ilkesini gözden geçirerek başlayacağız; aynı ilkeler programlama konusunu incelemeye başladığımızda yeniden karşımıza çıkacak.

İşlemciler

Kişisel bir bilgisayar, ya da daha doğrusu kişisel bilgisayarın bileşenlerinden Merkezi İşlem Birimi (MİB) çeşitli matematiksel hesaplardan başka birtakım mantıksal işlemler de yapabilir.

Bütün mantıksal işlemlerin temelini verilerin kıyaslanması oluşturur. Kıyasın sonucuna bağlı olarak hangi işlerin yapılacağına karar verilir.

Örneğin ödemeleri hesaplayan bir programı ele alalım. Birçok başka program gibi bu programın da ne zaman duracağını bilmesi gerekir. Bunu sağlamak için makineden gelen her kaydı (örneğin ödeme yapılacak kişilerin ismini) ''SON'' adlı bir veriyle karşılaştırması istenebilir. Karşılaştırmanın sonucu olumsuz ise program işlemesini sürdürecek, yok eğer olumlu ise belirli bir sinyal gönderdikten sonra duracaktır.

Kimi bakımlardan mantıksal işlemler aritmetik işlemlere benzetilebilir. Bu benzerliği biraz daha ayrıntılı olarak gözden geçirelim.

Herhangi bir aritmetik işlemde birbirinden farklı üç birimin varlığından söz edebiliriz:

- **İşlenenler**
- **İşlemci**
- **Sonuç**

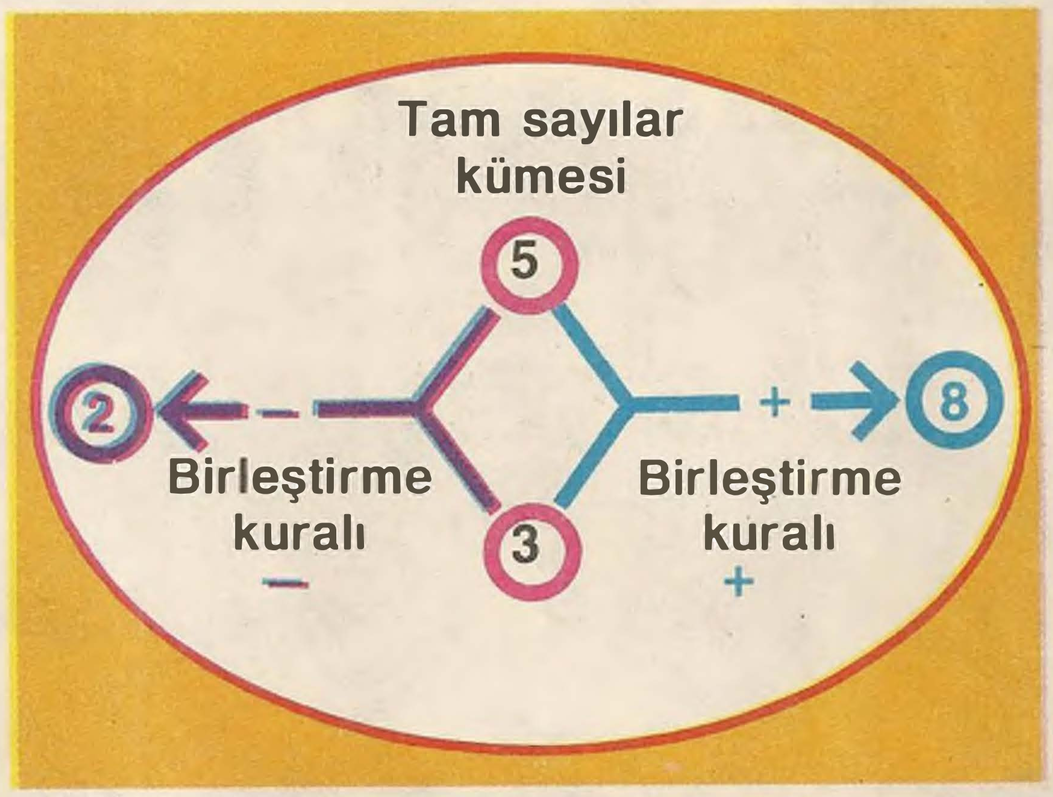
Aritmetik işlemlerde işlenenler, hesaplamada kullandığımız sayılar, işlemciyse, ne tür bir işlemin yapılacağını belirten simgedir. Sonuç ise bu işlemcinin gösterdiği işlem sayesinde işlenenlerden çıkarılan sayıdır.

Örneğin, $5 + 3 = 8$ problemini ele alalım. Sonuç (8), işlenenlere (yani 5 ve 3'e) + (toplama) işleminin uygulanmasıyla bulunmuştur.

İşlemciyi değiştirdiğimiz, örneğin + 'yı — yaptığımız takdirde, aynı işlenenler bu kez başka bir 10 tabanlı sayıyı, 2 sayısını vereceklerdi.

Bu noktadan farklı işlem türlerini (toplama, çıkarma, bölme gibi) farklı **birleştirme kuralları** olarak düşünebiliriz. Çünkü bu işlemler belirli bir sayılar grubuyla, yani işlenenlerle, başka bir sayılar grubunun, yani sonuçların nasıl biraraya getirilebileceğini belirleyen kurallardır. İşte işlemci bu kuralı ifade eder.

Herhangi bir birleştirme kuralı, örneğin toplama ya da çıkarma, ancak türdeş unsurları biraraya getirebilir. Başka bir deyişle, herhangi bir birleştirme kuralının biraraya getirdiği unsurların aynı kümeye ait olması gerekir (bkz. şekil).



Şimdi konuyu daha fazla dağıtmadan küme kavramını aydınlatmaya çalışalım.

Kümeyi, kabaca, belirli birtakım kurallara uymak zorunda olan unsurlar topluluğu olarak tanımlayabiliriz.

Kümelerin uymak zorunda olduğu kuralları tanımlamak ve unsurlarının özelliklerini araştırmak soyut cebirin işidir. Biz burada kümeyi ''türdeş unsurlar topluluğu'' olarak tanımlamakla yetineceğiz. Gerçi bu tanım matematiksel olarak fazla kesin ve doğru bir tanım değildir ama, amacımız bilgisayar uygulamaları üzerinde durmak olduğundan, bizim için yeterlidir.

Bilgisayar uygulamalarında sık sık türdeş olmayan unsurların aynı işlem içerisinde kullanılmasından doğan hatalara rastlamaktayız; bu yüzden bu sorunu ileride ayrıntılı olarak ele alacağız.

Şimdiye kadar yalnızca aritmetik işlemler üzerinde durduk. Buraya kadar söylediklerimizden canımızın çektiği herhangi bir birleştirme kuralını alıp, yeni bir işlem türü olarak tanımlayabileceğimiz izlenimi doğmuş olabilir. Ama işler aslında o kadar basit değil.

Aritmetik işlemler dışındaki birleştirme kurallarını gözden geçirmek için önce **Boole işlemleri** olarak anılan mantıksal işlem türlerini öğrenmemiz gerekir.

Bu işlemlerin, zaman zaman çok karmaşık olabilen, kendine özgü kuralları vardır. Biz burada, yalnızca bilgisayar programlamada kullanılan kurallara değineceğiz.

Boole Cebiri İşlemcileri

Örnek olarak yine trafik ışıklarını ele alalım. Trafik ışıkları şu dört durumdan ancak birinde olabilirler:

1- Çalışmıyor, sönük
2- Kırmızı
3- Sarı
4- Yeşil

''Trafik ışıkları ne durumda?'' sorusuna verebileceğimiz yukarıdaki dört cevabın dışında bir cevap yoktur. Ayrıca herhangi bir anda bu cevaplardan ancak birini verebiliriz. Başka bir deyişle her cevap diğer seçenekleri dışlar.

Dolayısıyla soruya vereceğimiz yanıt, **Çalışmıyor** VEYA **Kırmızı** VEYA **Sarı** VEYA **Yeşil** şeklinde ifade edilebilir.

Burada kullandığımız **VEYA (OR)** kelimesi mantıksal bir işleme işaret eder. Mevcut bütün programlama dillerinde bu işlemin işareti İngilizce **OR** kelimesi olduğundan, biz de bu işlemden söz ettiğimizde bu kelimeyi kullanacağız.

Şimdi yeniden trafik ışıkları örneğimize dönelim:

Trafik işaretinin alabileceği her durumu bir sayı ile belirtelim:

Çalışmıyor = 1; Kırmızı = 2 vb. O zaman şöyle yazabiliriz:

Trafik ışığı = 1 OR 2 OR 3 OR 4.

Soru: ''Bir araba trafik ışıkları hangi durumdayken geçebilir?''

Cevap: ''Işıklar çalışmıyorsa veya yeşil ışıkta.''

Bu cevabı simgesel olarak şöyle yazabiliriz:
GEÇ = 1 OR 4.

Aslında 1 OR 4 ifadesi de yeterli değildir. Çünkü 1 durumunda, yani trafik ışığı çalışmıyorsa, geçmek için önce yolun boş olduğundan emin olmak gerekir. Öyleyse doğru cevabın şöyle yazılması gerekiyor:

GEÇ = Yeşil ışıkta **veya** sinyal çalışmıyorsa **ve** yol serbestse.

Buradaki **ve** kelimesi şu iki durumun her ikisinin birden gerçekleşmesi gerektiğini belirtiyor: Işık sönük, yol serbest. Yani tipik bir **VE (AND)** işlemi; **VE** kelimesi de bu iki koşulun arasındaki ilişkiyi belirtiyor; dolayısıyla bu da bir Boole işlemidir ve programlama dillerinde İngilizce **AND** kelimesi ile gösterilir.

Yolun serbest olma durumunu da 5 rakamı ile gösterelim; o zaman tam cevabı şöyle yazabiliriz:

GEÇ = 4 OR 1 AND 5.

Ancak bu eşitlik iki farklı biçimde yorumlanabilir:

1- GEÇ = (4 OR 1) AND 5.
2- GEÇ = 4 OR (1 AND 5).

Birinci yoruma göre trafik ışıkları ister yeşil olsun ister sönük (4 OR 1), geçebilmek için yolun ille de boş olması gerektiğini (AND 5) söylüyor. Bu yanlış bir yorum; bizim istediğimiz yorum ya trafik sinyalinin yeşil olması (4 OR) ya da hem sönük olması hem de yolun boş olması (1 AND 5).

Bu örnekte de görüldüğü gibi, tıpkı aritmetik işlemlerde olduğu gibi mantıksal Boole işlemlerinde de parantezlerin nasıl yerleştirildiği sonuç bakımından çok önemlidir.

AND ve OR işlemleri herhangi bir kümenin elemanlarına uygulanabilir. Bir örnek olarak 2 tabanlı iki sayı ele alalım ve bu sayıları A ve B ile gösterelim:

A	B
1	1
1	0
0	1
0	0

AND ve OR işlem işaretlerini kullanarak şu sonuçları elde ederiz:

1	2	3	4
A	B	A AND B	A OR B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

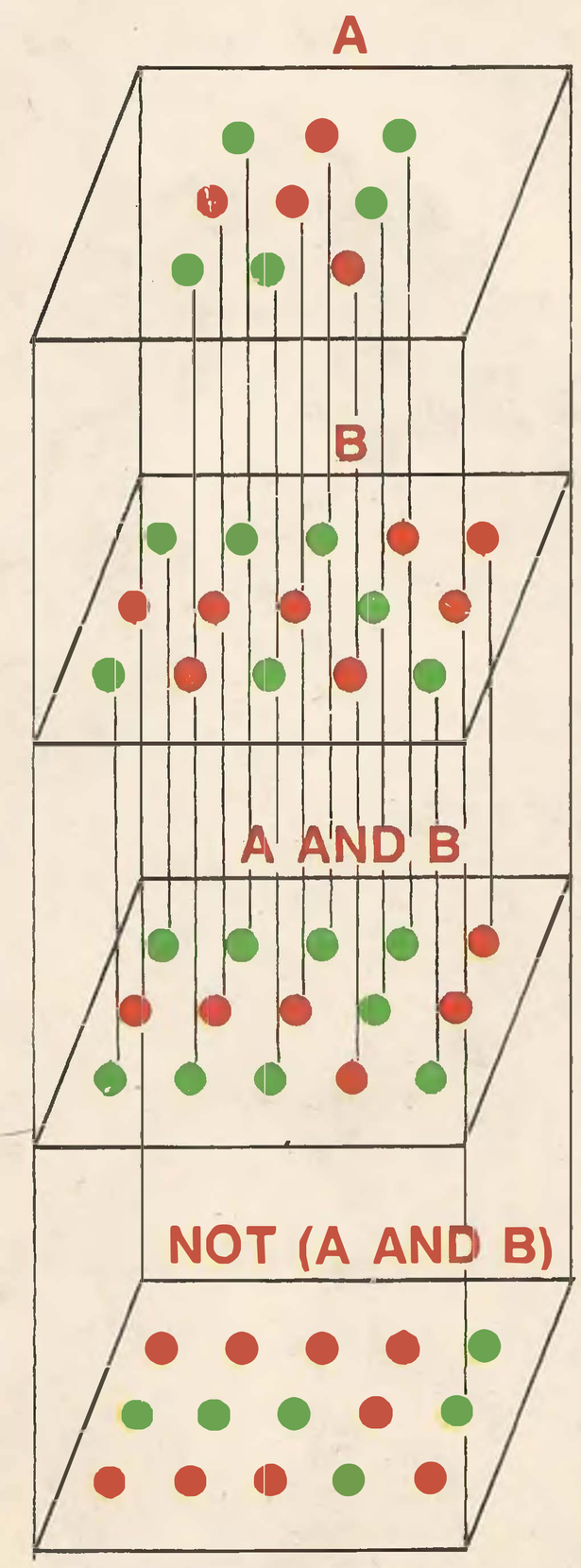
Bu tabloda birinci ve ikinci sütunlar işlenenleri, üçüncü sütun A AND B işleminin sonucunu, dördüncü sütun ise A OR B işleminin sonucunu içermektedir. Görüldüğü gibi A AND B işlemi 1 sonucunu ancak hem A hem B 1 olduğunda alıyor. Buna karşılık işlenenlerden herhangi biri 1 ise, A OR B işleminin de sonucu 1 oluyor. Aşağıdaki tabloda 2 tabanlı kümeye **VE, VEYA, DAR VEYA** ve **DEĞİL** işlemlerinin uygulanmasını ve sonuçlarını görüyorsunuz. Kırmızı noktalar 1 değerini, yeşil noktalar 0 değerini gösteriyor.

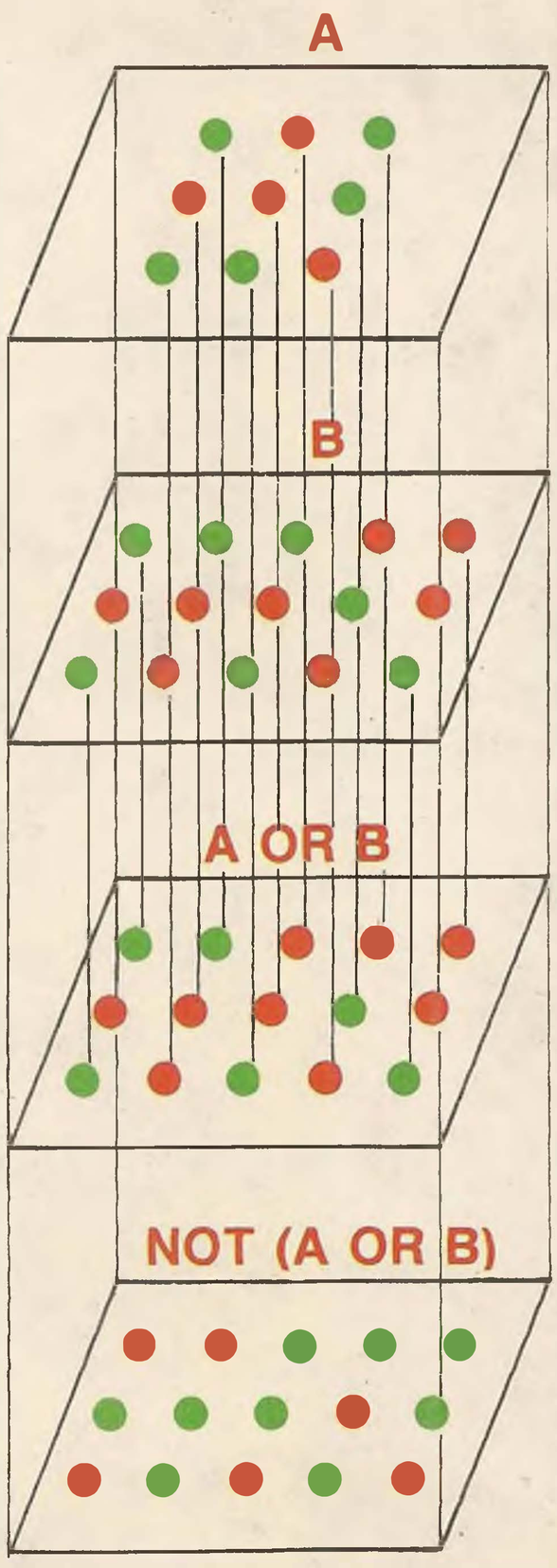
1 AND 1 = 1
1 AND 0 = 0
0 AND 1 = 0
0 AND 0 = 0

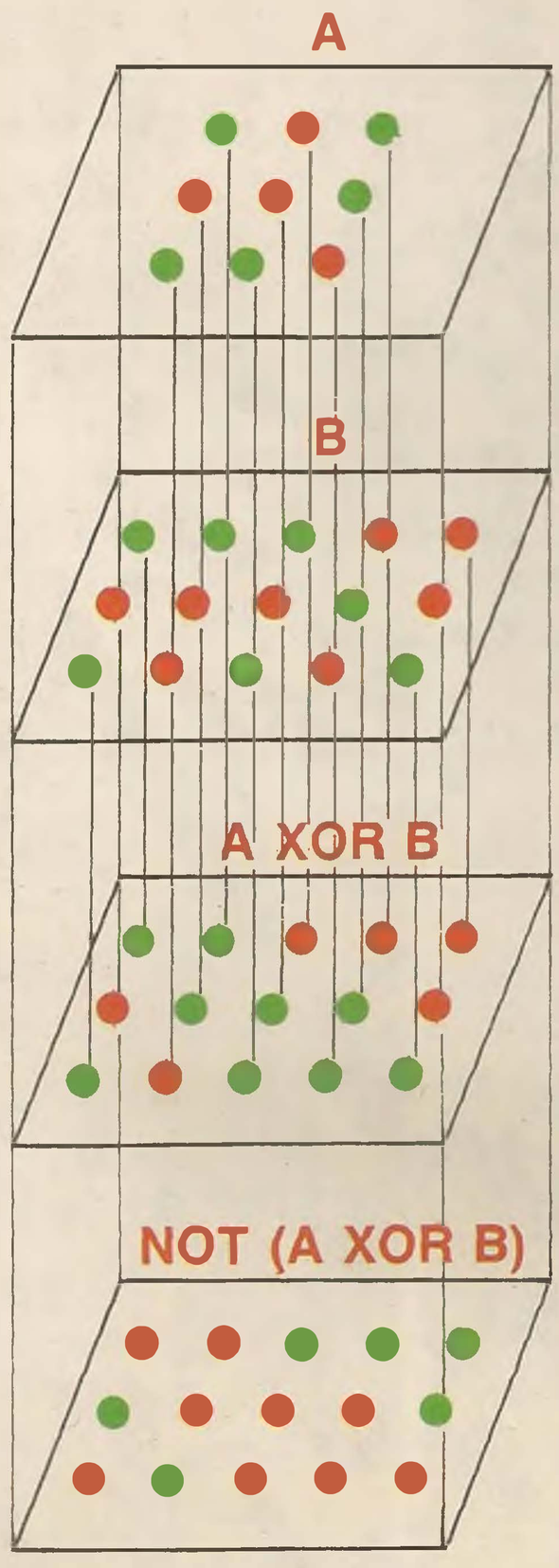
● AND ● = ●
● AND ● = ●
● AND ● = ●
● AND ● = ●

● OR ● = ●
● OR ● = ●
● OR ● = ●
● OR ● = ●

● XOR ● = ●
● XOR ● = ●
● XOR ● = ●
● XOR ● = ●







Şimdi üçüncü bir işlemi daha gözden geçirelim, **DAR VEYA** işlemi; bu işlemin işareti **XOR**'dur. Bu işlemin sonucu A = 1 ve B = 1 olduğu durumlar dışında **VEYA** işleminin sonucuyla aynıdır. Ancak bu durumda, yani hem A hem B 1 olduğunda, **DAR VEYA** işleminin sonucu 0 olur. Başka bir deyişle, **DAR VEYA** 1 sonucunu yalnız işlenenlerden *ancak* biri 1 değerinde ise alır.

Yani, bu işlemin doğruluk değerleri tablosu şöyledir:

A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

İnceleyeceğimiz işlemlerin sonuncusu **DEĞİL** işlemidir (işareti **NOT**). Tek bir işlenene uygulandığında sonuç olarak o işlenenin tümleyenini verir. Yani işleyen 1 ise **DEĞİL** işleminin sonucu 0, 0 ise 1 olur.

A	NOT A
1	0
0	1

Buraya kadar örnek olarak işlenenlerde hep 2 tabanlı tek basamaklı sayıları (A = 1/0, B = 1/0) kullandık; mantıksal işlemleri çok basamaklı sayılara da uygularsak:

A = **0 1 1 0 1**
B = **1 0 0 1 1**

Özet olarak; en önemli mantıksal işlemler şunlardır: VE (AND), VEYA (OR), DAR VEYA (XOR), DEĞİL (NOT). Doğruluk değerleri tabloları şöyledir:

A	B	AND	OR	XOR	NOT A	NOT B
1	1	1	1	0	0	0
1	0	0	1	1	0	1
0	1	0	1	1	1	0
0	0	0	0	0	1	1

Mantıksal işlemler, aritmetik işlemlerle aynı ilkelere göre ele alınırlar.
Bundan başka, mantıksal işlemler yalnızca türdeş elemanlara uygulanırlar.
Şimdiye değin, mantıksal işlemleri hep 2 tabanlı tek rakamlı sayılar olarak ele aldık (A = 1/0, B = 1/0). Halbuki bunlar, iki tabanlı çok rakamlı sayılara da aynen uygulanabilirler.

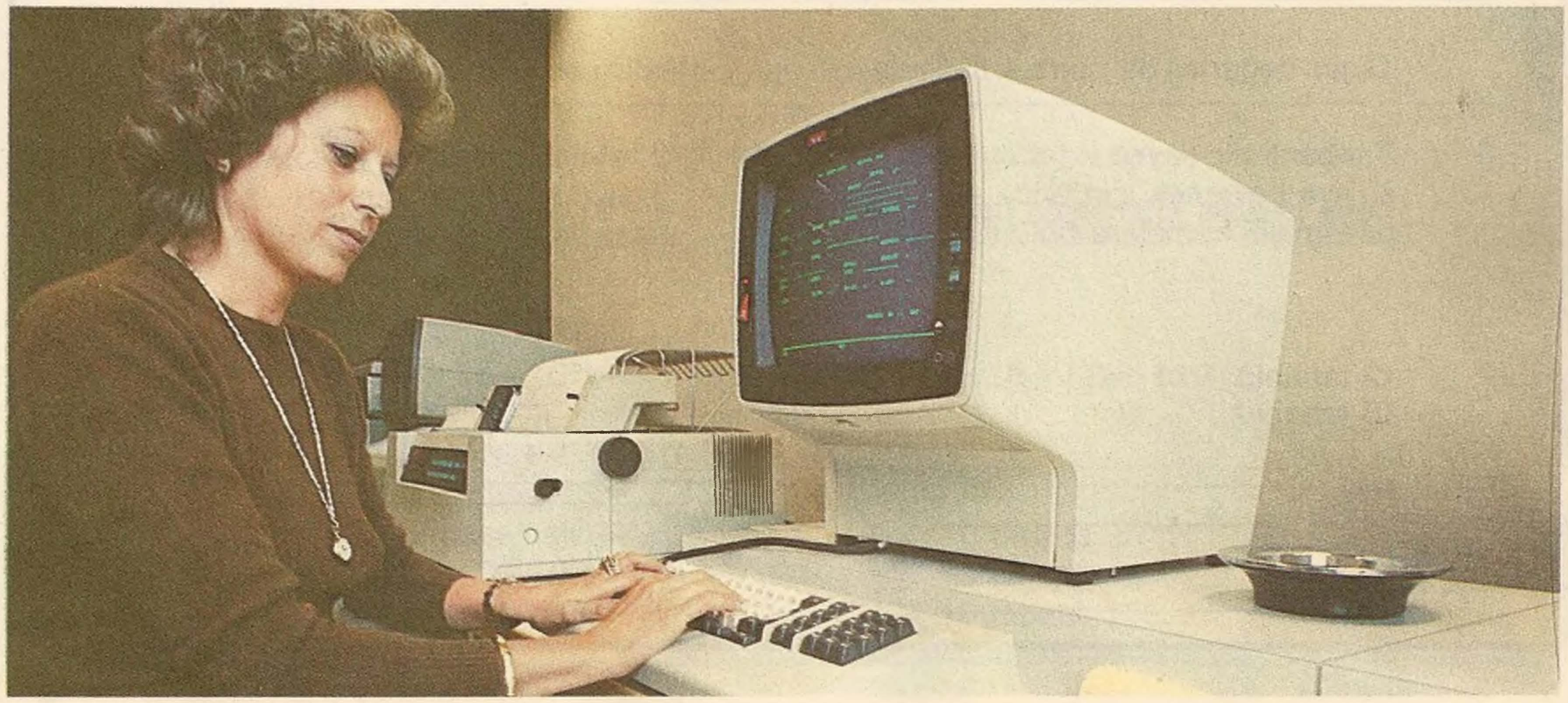
Şu sonuçları elde ederiz:

NOT A = **1 0 0 1 0**
NOT B = **0 1 1 0 0**

Bu sayıların 10 tabanındaki karşılıkları şunlardır:

A = **0 1 1 0 1 = 13** (10 tabanlı) $(2^3 + 2^2 + 2^0)$
B = **1 0 0 1 1 = 19** (10 tabanlı) $(2^4 + 2^1 + 2^0)$

Bir sekreter bilgisayara veri giriyor. Bilgisayara kablo ile bağlı olan klavye, çözülebilir türdendir.



TEST 1'in Çözümleri

1 / a - 2 tabanlı sayı sisteminin temeli sayısal devrelerin işleyişininkiyle aynıdır. Her ikisi de yalnızca iki durum tanır: (sinyal) VAR (ON) ve (sinyal) YOK (OFF). Dolayısıyla 2 tabanlı sistem bilgisayarların kullanabileceği tek sayı sistemidir.

b - 8 tabanlı sayı sistemi, 2 tabanlı üç simgeyi 8 tabanlı tek simgeyle karşılama olanağı sağlar.
Örneğin, 8 tabanlı 7 sayısının, 2 tabanlı 111'e, 57 sayısının 2 tabanlı 101111 'e tekabül etmesi gibi. Bu yüzden bu biçim daha tıkız ve kullanışlıdır.

c - 10 tabanında 1 ilâ 15 arasındaki bütün değerleri ifade edebilmek için 16 tabanında tek basamak yeterlidir (10 tabanlı 15 = 16 tabanlı F). 16 tabanlı sayı sisteminde taban 16 olduğundan, her basamak 2 tabanlı dört rakama tekabül eden bir bilgi içerir:
(2 tabanlı) 1111 = (16 tabanlı) F = (10 tabanlı) 15. 16'dan büyük sayılar için birden fazla basamak gerekir. Örneğin (10 tabanlı) 27 = (2 tabanlı) 11011 = (16 tabanlı) 1B.

d - 8 tabanlı sayı sisteminin tabanı 10 tabanlı sistemde 8 ile ifade edilen değerdir. Dolayısıyla bu değer 8 tabanında 10 rakamı ile ifade edilir.

---

2 / 2 tabanlı bir sayının 10 tabanına çevrilmesi ağırlıklı tablolardan yararlanılarak yapılır. 2 tabanlı bir sayı herhangi bir basamakta 1 rakamını içeriyorsa, 2'nin o basamağa tekabül eden kuvveti alınır. Böylelikle oluşturulan kuvvetler tablosunda, her 1 simgesi içeren basamağa tekabül eden 2'nin kuvvetlerinin toplanmasıyla söz konusu sayının 10 tabanında karşılığı bulunur.

Üst	2^5	2^4	2^3	2^2	2^1	2^0
Değeri	32	16	8	4	2	1
			1	0	0	1
					1	1
						1

1 + 8 = 9 1001 (2 tabanlı) = 9 (10 tabanlı)

1 + 2 = 3 11 (2 tabanlı) = 3 (10 tabanlı)

1 = 1 1 (2 tabanlı) = 1 (10 tabanlı)

Diğer değerler de benzer biçimde tabloya yerleştirerek bulunur.

---

3 / 2 tabanlı bir sayıyı 8 tabanlı bir sayıya çevirmek için önce sayı 3'er basamaklı kümelere bölünür:

(2 tabanlı) 1 1 0 1 1 1
(8 tabanlı) 6 7

(2 tabanlı) → (8 tabanlı)

11 → 3
101010 → 52
111111 → 77

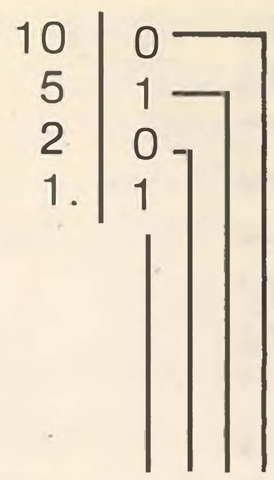
4 / 2 tabanlı bir sayıyı 16 tabanlı bir sayıya çevirmek için önce sayı 4'er basamaklı kümelere bölünür:

(2 tabanlı) (16 tabanlı)

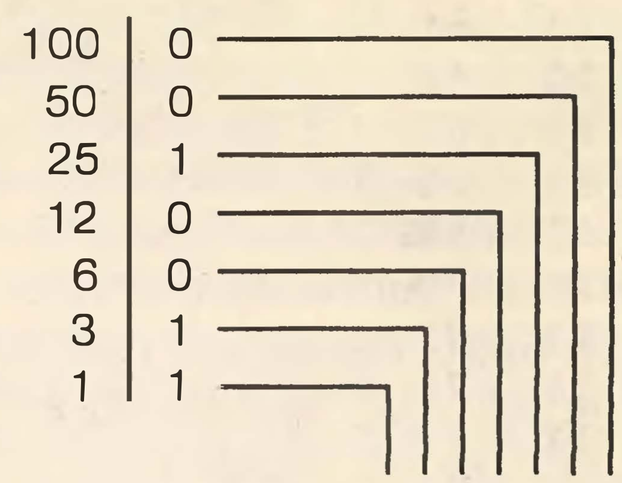
1101 1111 = DF

Diğer değerler sayfa 63-67'deki listelerden elde edilebilir.

5 / Çevrim, kalan değerleri 2'ye bölerek yapılır:



10 (10 tabanlı) = 1 0 1 0



100 (10 tabanlı) = 1 1 0 0 1 0 0

Aynı şekilde 10 tabanlı 1000 = 2 tabanlı 1111101000 = 16 tabanlı 3E8 karşılıkları elde edilir.

---

6 /

2 tabanlı	10 tabanlı
0 0 0 1	1
0 0 1 0	2
0 1 0 0	4
1 0 0 0	8

Görüldüğü gibi 2 tabanlı sayı sisteminde 1 rakamının sola doğru bir basamak kaydırılması, 2 ile çarpılmasına tekabül eder. Sağa doğru kaydırılması ise 2 ile bölünmesine tekabül eder. Aynı sonuca şu gözlemden hareketle de varmak mümkün; sayı sistemi ne olursa olsun, bir sayının bir basamak sola kaydırılması taban değeri ile çarpılmasına tekabül eder. Programlama dilinde bu basamak değiştirmeye ''KAYMA'' adı verilir.

---

7 / a - doğru; b - doğru; c - yanlış; d - doğru.

---

8 / 8 tabanlı bir sayının 10 tabanında karşılığı, her rakamın, 8'in o rakamın bulunduğu basamağa tekabül eden kuvvetinin çarpılması ve bu çarpımların sonuçlarının toplanması ile bulunur. Örneğin, 8 tabanlı 700 sayısının 10 tabanına çevrilmesini ele alalım:

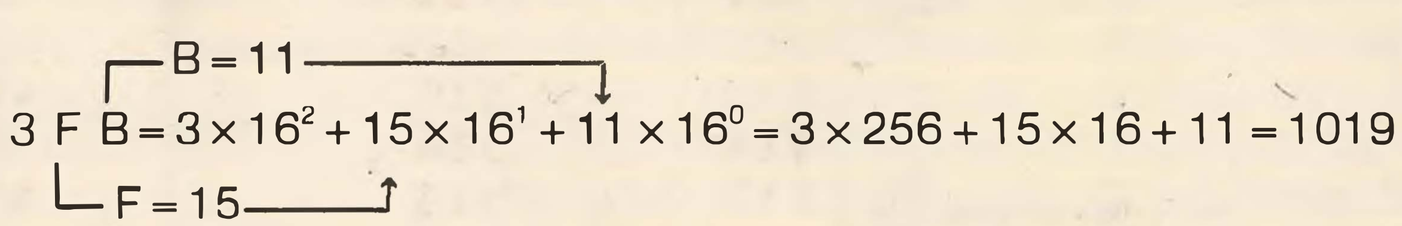
$$(8 \text{ tabanlı}) \; 700 = 7 \times 8^2 + 0 \times 8^1 + 0 \times 8^0 = 7 \times 64 + 0 + 0 = 448$$

---

9 / 16 tabanlı bir sayının başka bir tabanlı sayı sistemine çevrilmesi, 2 tabanlı sayı sistemi yoluyla yapılırsa daha kolay olur. Örneğin, 16 tabanlı 3FB sayısı 2 tabanında şu sayıya karşılık gelir:

3 = 0 1 1
F = 1 1 1 1
B = 1 0 1 1
3FB = 1 1 1 1 1 1 1 0 1 1 = 1019 (10 tabanlı)

Aynı sonucu her rakamı 16'nın o rakamın bulunduğu basamağa tekabül eden kuvvetiyle çarparak da elde edilebilirdi.



---

10 /

2 basamak =	1 1	= 10 tabanlı 3
4 basamak =	1 1 1 1	= 10 tabanlı 15
8 basamak =	1 1 1 1 1 1 1 1	= 10 tabanlı 255

TABLO 3

I	J	AND	OR	XOR	J + (NOT I) + 1
11	65	1	75	74	54
11	66	2	75	73	55
11	67	3	75	72	56
11	68	0	79	79	57
11	69	1	79	78	58
11	70	2	79	77	59
11	71	3	79	76	60
11	72	8	75	67	61
11	73	9	75	66	62
11	74	10	75	65	63
11	75	11	75	64	64
11	76	8	79	71	65
11	77	9	79	70	66
11	78	10	79	69	67
11	79	11	79	68	68
11	80	0	91	91	69
11	81	1	91	90	70
11	82	2	91	89	71
11	83	3	91	88	72
11	84	0	95	95	73
11	85	1	95	94	74
11	86	2	95	93	75
11	87	3	95	92	76
11	88	8	91	83	77
11	89	9	91	82	78
11	90	10	91	81	79

TABLO 4

I	J	AND	OR	XOR	J + (NOT I) + 1
32	65	0	97	97	33
32	66	0	98	98	34
32	67	0	99	99	35
32	68	0	100	100	36
32	69	0	101	101	37
32	70	0	102	102	38
32	71	0	103	103	39
32	72	0	104	104	40
32	73	0	105	105	41
32	74	0	106	106	42
32	75	0	107	107	43
32	76	0	108	108	44
32	77	0	109	109	45
32	78	0	110	110	46
32	79	0	111	111	47
32	80	0	112	112	48
32	81	0	113	113	49
32	82	0	114	114	50
32	83	0	115	115	51
32	-84	0	116	116	52
32	85	0	117	117	53
32	86	0	118	118	54
32	87	0	119	119	55
32	88	0	120	120	56
32	89	0	121	121	57
32	90	0	122	122	58

Bu örnekte A AND B = 1 olduğuna göre, A'nın ve B'nin 10 tabanındaki karşılıkları verildiğinde sonucun yine aynı olması gerekir:

0 1 1 0 1 AND **1 0 0 1 1 = 13** AND **19 = 1**

10 tabanında:
13 OR **19 = 31 (1 1 1 1 1 =**
$2^4 + 2^3 + 2^2 + 2^1 + 2^0 =$ **31)**

10 tabanında:
13 XOR **19 = 30 (1 1 1 1 0 = 30)**

Sayfa 78'deki tablolarda AND, OR, XOR işlemlerinin ve J + (NOT I) + 1 birleşik işleminin J ve I gibi iki sayıya uygulanmasının sonuçlarını görüyorsunuz.

Tablo 3, I'nın 33, J'nin ise 65 ilâ 90 arasında değer aldığı durumlardaki sonuçları, Tablo 4 ise I'nın 33, J'nin yine 65 ilâ 90 arasındaki değerleri aldığı durumlardaki sonuçları veriyor.

Bilgisayarda 65 ilâ 90 arasında kalan sayılar dizisi çok özel bir anlam taşır. Tablolarda gösterilen J + (NOT I) + 1 işlemi ise aslında J ile I arasındaki farkın tümleme yöntemine göre hesaplanmasından ibarettir. (NOT I'nın I'nın tümleyeni olduğunu söylemiştik).

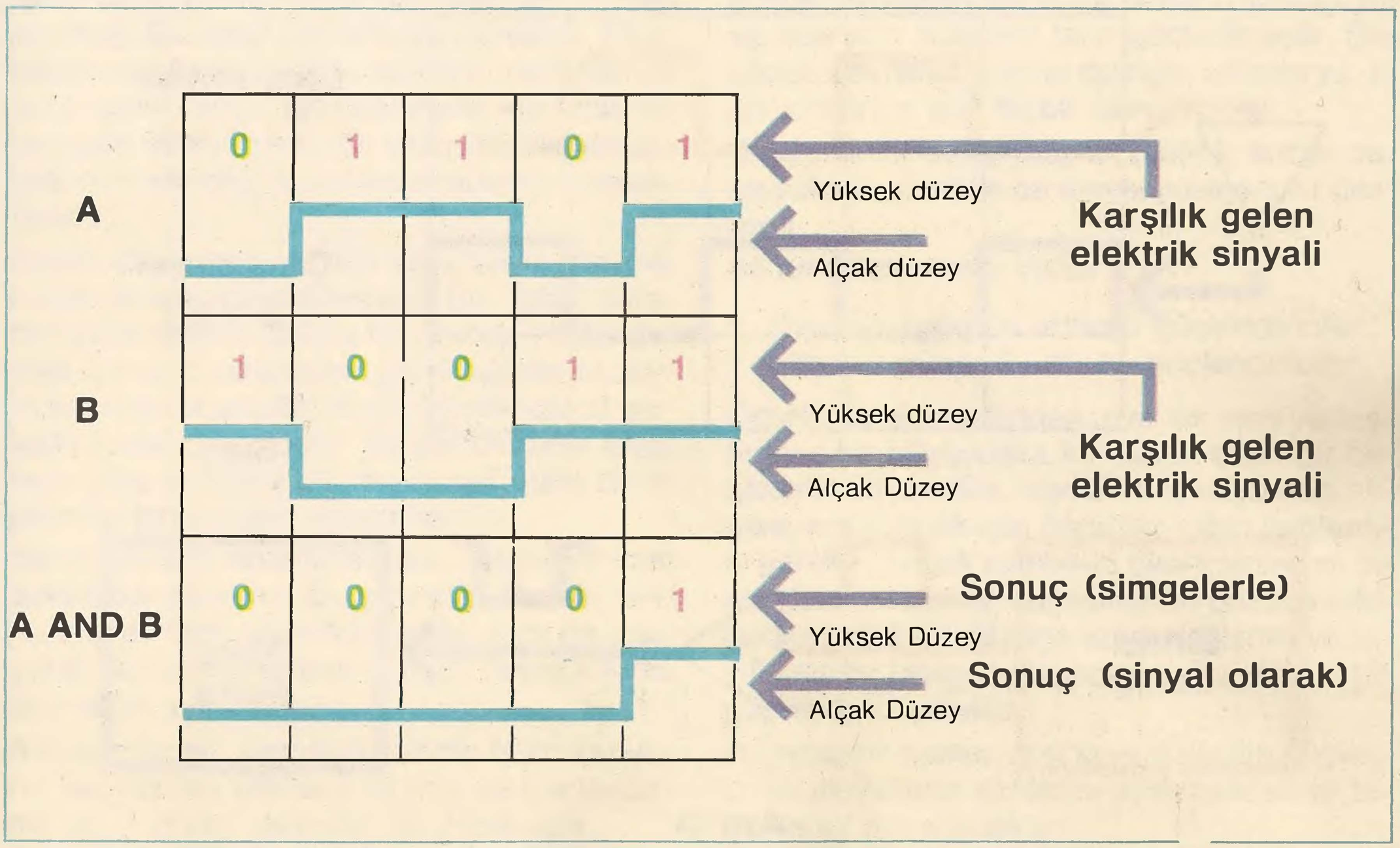
Kablolu Mantık

Mantık işlemleri aynen **1** ve **0** simgelerine uygulandıkları gibi elektrik sinyallerine de (yani s. 45-47 arasında anlattığımız ''darbeler''e) uygulanabilirler. Aslında darbelerin **1** ve **0** simgelerinin fiziksel bir gösterimi olduğunu düşünebiliriz.

Geleneksel olarak,

sinyalin yüksek düzeyi = 1
sinyalin alçak düzeyi = 0

olarak gösterilir. 2 tabanlı A = **0 1 1 0 1** ve B = **1 0 0 1 1** sayılarını iletirken, **1** simgele-



rinde elektriksel sinyal darbenin **yüksek** düzeyine ulaşacak, **0**'larda ise **alçak** düzeyde kalacaktır (bkz. şekil, sayfa 79).

A AND B sinyalini elde etmeyi mümkün kılan işlem ise **mantık devresi** adlı özel birtakım sayısal devreler tarafından gerçekleştirilir.

Her Boole işlemi için belli bir devre vardır. Her devrenin ise açıkça belirlenmiş bazı işlevleri vardır. Demek ki böyle bir devre bize belli bir mantıksal işlemi yapabilecek bir mekanizma sağlamış oluyor. Böyle bir mekanizmanın yapımı, belirli sayıda elektronik bileşenin tümleşik bir devre oluşturacak biçimde, bir devre içerisinde bağlanmasını gerektirir. Bu bağlama sistemine kablo örgüsü de dendiğinden oluşan devreye de **kablolu mantık** adı verilir.

Mantık işlemlerinin böyle yürütülmesinin aksayan yanı apaçık ortadadır. Her işlem için bir devreye gerek vardır. Oysa bilgisayarın **programlanabilir** bir mantığa sahip olduğunu söyledik. Demek ki, bilgisayar mantık işlemlerini kablolu mantık aracılığıyla değil, çok sayıda tümleşik devreden yararlanarak yürütecektir. Yine de bilgisayarlarda da birçok yan işlev, kablolu mantık tarafından yürütülür.

Mantık Devreleri

Temel mantık işlemcilerinin şunlar olduğunu görmüştük:

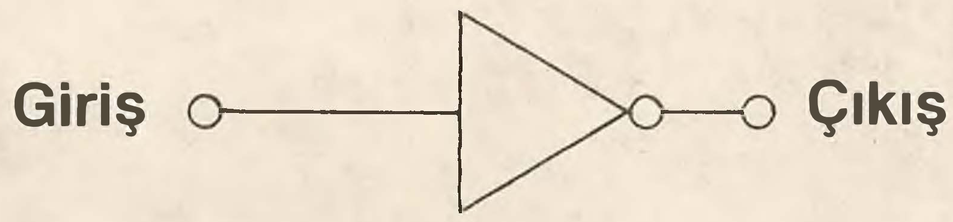
- **NOT (DEĞİL)**
- **AND (VE)**
- **OR (VEYA)**
- **XOR (DAR VEYA)**

Bu işlemler özel devreler tarafından gerçekleştirilir. Yani her işleme tekabül eden belli bir devre vardır.

NOT (DEĞİL) İşlemcisi

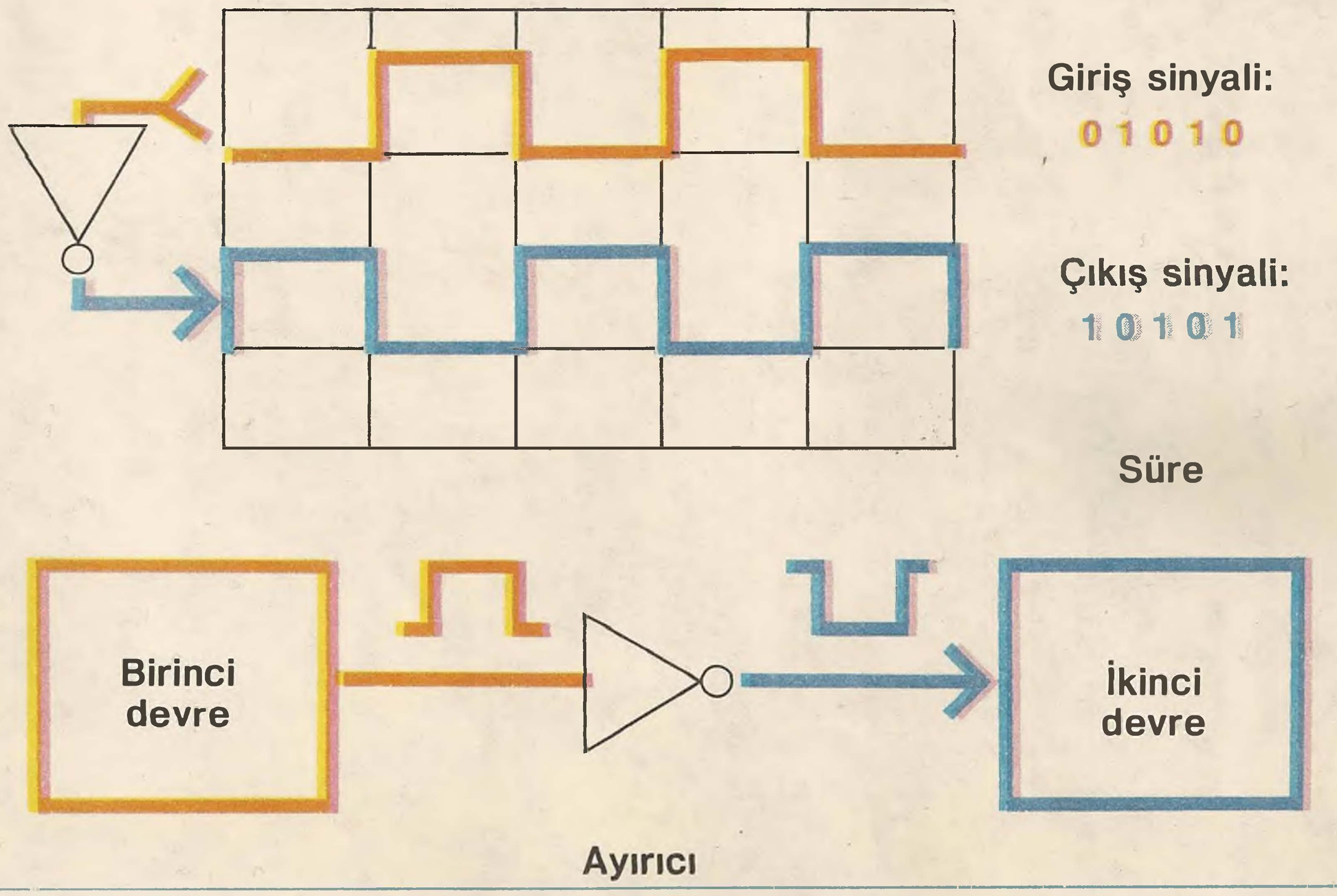
Değil işlemi herhangi bir veriyi tersine çeviren işlemdir. Yani **1** değerini **0, 0** değerini de **1** yapar. Öyleyse bu işlemi yapacak devrenin işlevi de, giren sinyali **1** (ya da **y**) ise **0**'a (**a**'ya), **0** (ya da **a**) ise **1**'e (**y**'ye) çevirmekten ibarettir.

Bu devrenin girdisi **1** (**y**)'den **0** (**a**)'ya değiştiği takdirde, çıktı da tam ters yönde **0** (**a**)'dan **1** (**y**)'ye değişecektir.

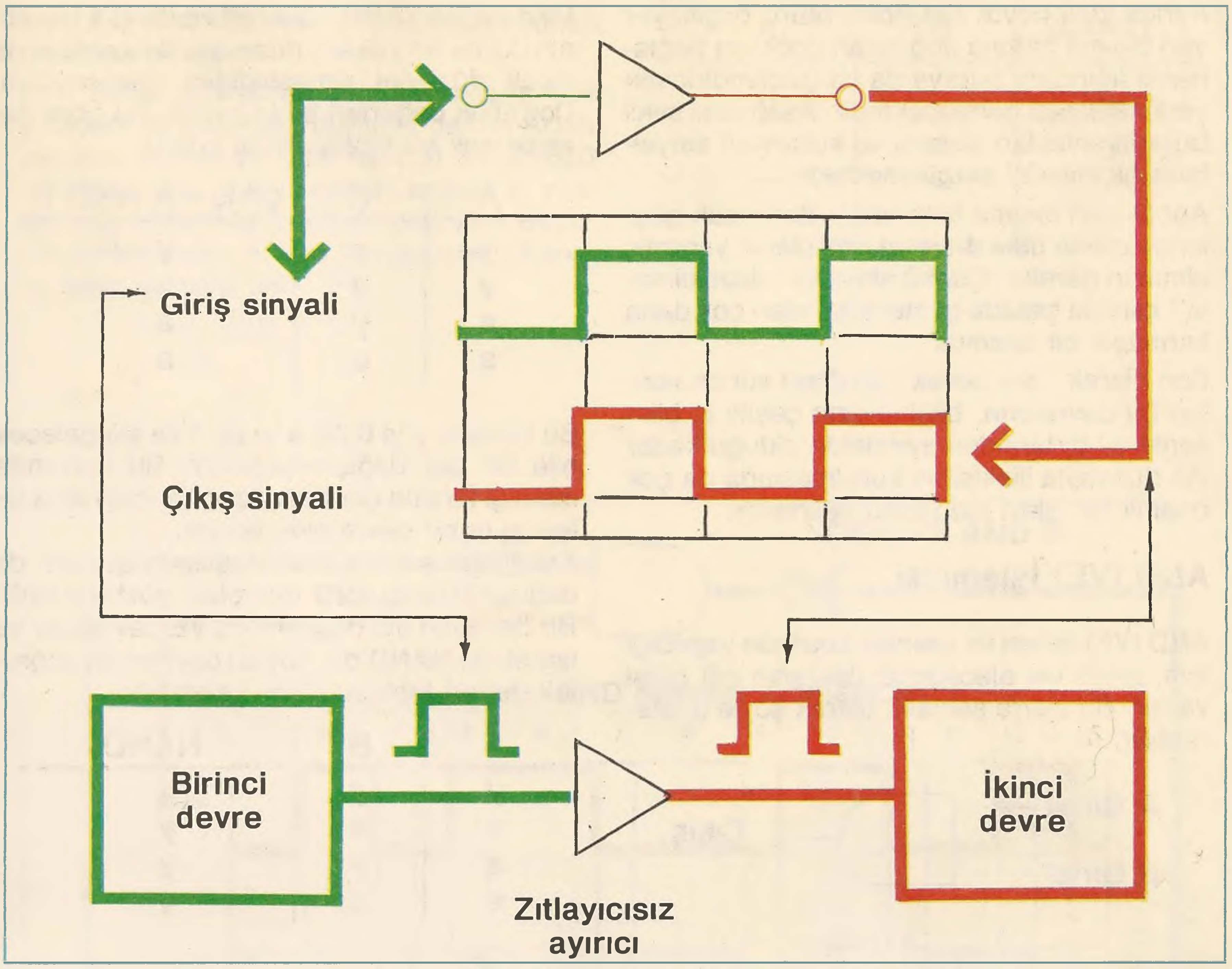


Girişe bir SAAT (bkz. s.47) yerleştirdiğimiz takdirde, çıkış, giriş sinyalinin tam tersini gösterecektir.

Sayfanın altındaki şekil girdi ve çıktı sinyallerinin izlediği yolu gösteriyor.



Ayırıcı



Gelen sinyali zıtlamak dışında bu devrenin bir işlevi daha vardır: Diğer devreleri birbirinden ayırmak. Bu işlevi açıklamaya çalışalım: Bilgisayarı oluşturan çeşitli unsurların karmaşık bir sürü işlevi vardır. Bu unsurların küçücük bir tümleşik devre içerisinde birleştirilebilmesi ancak ileri teknoloji sayesinde mümkün olmaktadır.

Ancak bu yüzden, bu ileri teknolojiyle yapılmış küçük devreler kendilerinden bir ''güç'' yaratamamaktadırlar. Başka bir devreyi yönlendirmek için güç yaratılması gerektiğinde iki devre arasına bir ampflikatörün (güçlendirici) yerleştirilmesi zorunludur. Güçlendiricinin işlevi hem çıkış sinyalini güçlendirmek, hem de iki devreyi birbirinden ayırmaktır.

Şimdi burada anlattıklarımızı, sayfa 80'deki şekille karşılaştırın. Bu çizimde ortadaki devre, sinyali hem güçlendirmekte, hem de zıtlayarak (tersine çevirerek) birinci devreyi ikinci devreden ayırmaktadır.

Ancak zıtlama, kimi durumlarda işleri karıştırır: bu yüzden yalnızca ayırma ve güçlendirme işlevi gören devreler de yapılmıştır.

Yukarıdaki şekilde böyle bir devreyi uyaran bir simge, simgenin yol açtığı sinyalin izlediği yol ve devrenin kullanım tarzı gösterilmiştir. Şekildeki devrenin ayırma dışında, zıtlama ya da güçlendirme gibi hiçbir işlevi yoktur.

Güçlendirici ise bu gelenle orantılı, ancak daha yüksek şiddette bir sinyal yollayan bir devredir.

İki tür güçlendirici vardır:

- Sinyalin değerini arttıran güçlendiriciler;
- Sinyalin gücünü arttıran güçlendiriciler.

Örnek olarak kendinden uzak bir yere yerleştirilmiş bir bilgiyazara kumanda eden bir bilgisayar düşünelim. Verileri bilgisayardan bilgisayara iletmek için önce bir kablo bağlantısı gerekir. Ancak kablonun direncinden ve dışarıdaki parazitten kaynaklanan bozucu etkilerden ötürü sinyal bilgiyazara değişmiş ve zayıflamış bir biçimde ulaşacaktır. Bu yüzden bir güçlendirici gerekir.

Bilgiyazarın hemen önüne yerleştirilen güçlendirici sinyallerin şiddetini ayarlayacak ve biçimlerini düzeltecektir.

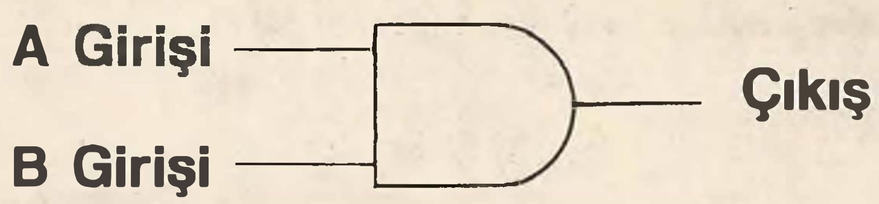
Ayrıca kısa devre riskinden ötürü bilgisayar veri taşıma hattına doğrudan doğruya bağlanamadığından, buraya da bir güçlendiricinin yerleştirilmesi gerekmektedir. Aşağıdaki şekil burada anlatılan sistemi ve kullanılan sinyallerin biçimlerini sergilemektedir.

Ancak veri taşıma hattında kullanılacak güçlendiricilerin **(line drivers)** özel olarak yapılmış olmaları gerekir. Çünkü sinyalin ''düzeltilmesi'' aslında şekilde gösterildiğinden çok daha karmaşık bir işlemdir.

Son olarak ''ara bellek'' **(buffer)** adı da verilen bu devrelerin, bilgisayarın çeşitli iç bileşenlerini birbirinden ayırmakta olduğu kadar dış dünyayla ilişkisinin kurulmasında da çok önemli bir işlevi olduğunu belirtelim.

AND (VE) İşlemcisi

AND (VE) işlemi iki işlenen üzerinde yapıldığı için, şimdi ele alacağımız devrenin çift girişi vardır. Bu devre şematik olarak şöyle gösterilebilir:



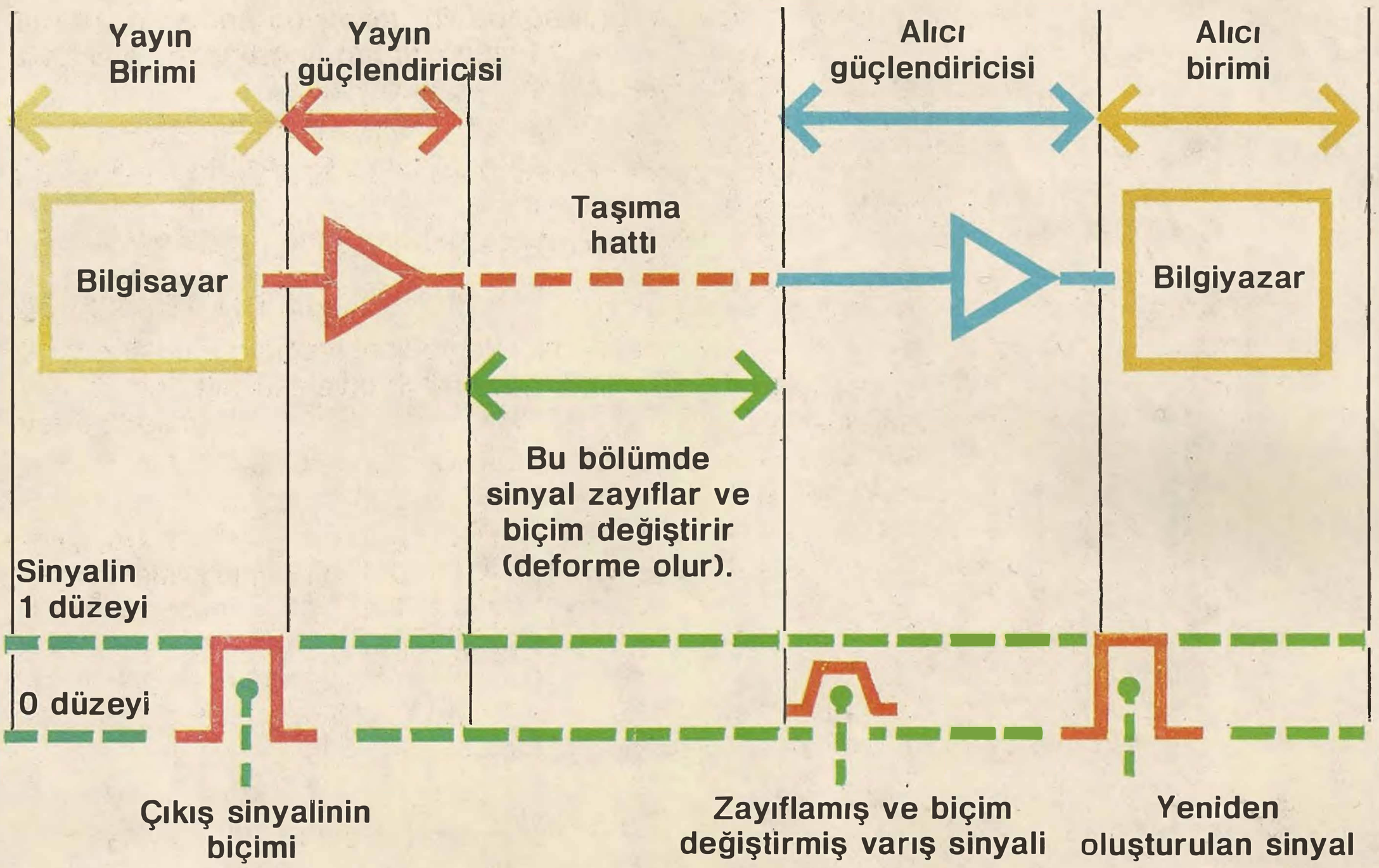
Devre aynen AND işlemi gibi çalışır. **1** işaretinin darbenin yüksek düzeyini, **0** işaretinin de alçak düzeyini simgelediğini anımsayalım. Doğruluk değerleri tablosunu buna göre düzenlersek şu tabloyu elde ederiz:

A	B	çıkış = A AND B
y	**y**	**y**
y	**a**	**a**
a	**y**	**a**
a	**a**	**a**

Bu tabloda **y**'yi **0** ile, **a**'yı da **1** ile simgelesek bile bir şey değişmeyecektir. Bu durumda mantığı tersine çevrilmiş (değillenmiş) ama işlevi aynı bir devre elde ederiz.
Aşağıdaki şekilde hem değillenmiş, hem de değillenmemiş **AND** devreleri gösterilmiştir. Bu devrenin adı değillenmiş **VE** devresidir ve işareti de **NAND**'dır. NAND devresinin doğruluk değeri tablosu şöyle yazılabilir:

A	B	NAND
y	**y**	**a**
y	**a**	**y**
a	**y**	**y**
a	**a**	**y**

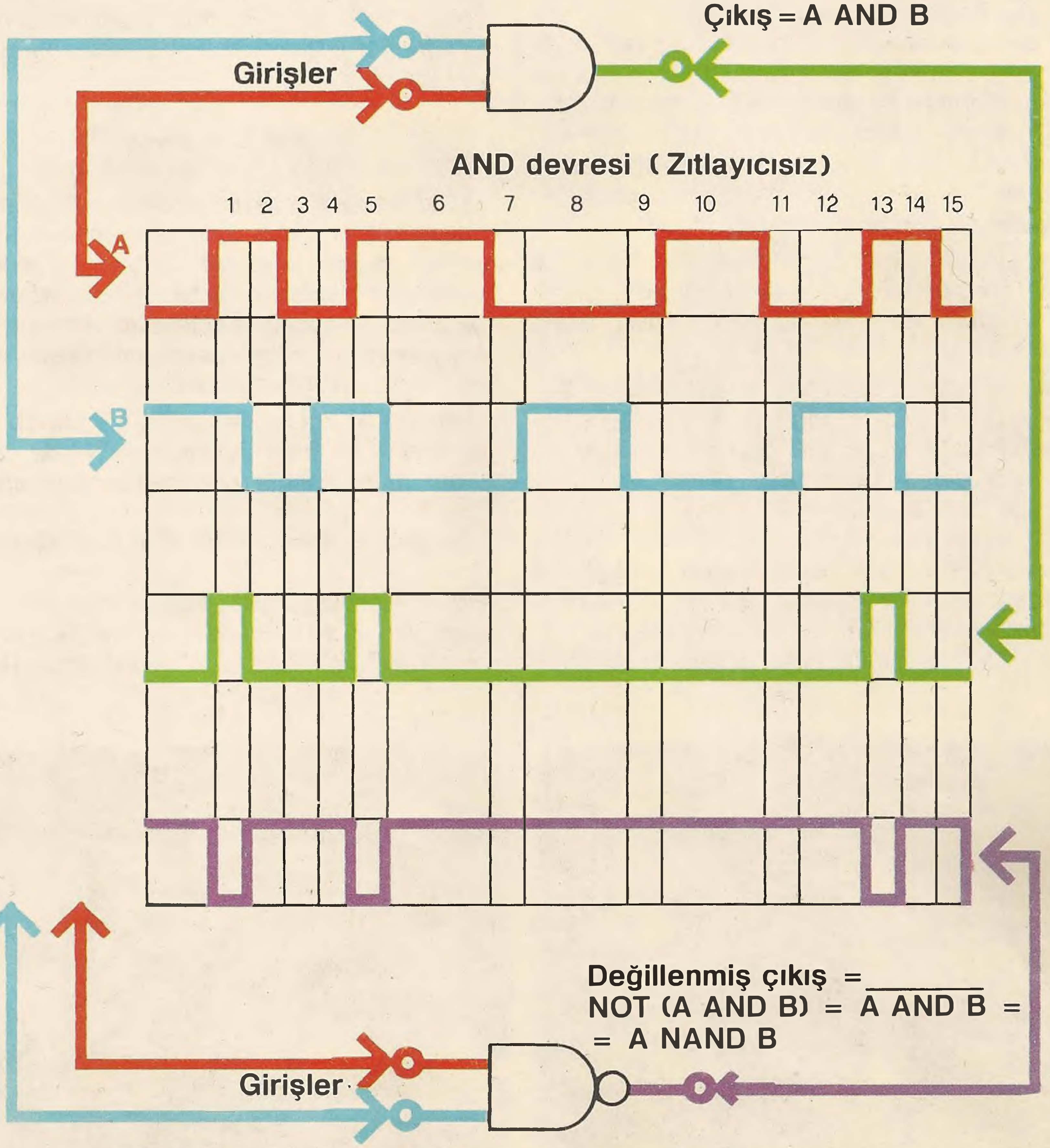
SİNYALLERİN BİR HATTA TAŞINMA ŞEMASI



Bu tablo AND devresi için oluşturduğumuz tablonun tersidir.

Son olarak girdilerden birinin, örneğin A'nın **y** değerinde kilitlendiği durumlarda NAND tablosunun zıtlayıcıya (yani NOT'a) dönüştüğünü söyleyelim. Bunun nedeni açıktır. A, **y** değerinde kilitlendiği takdirde doğruluk değerleri tablosundan A = **a** durumlarını çıkarırız ve tablo şu şekli alır:

A	B	NAND
y	**y**	**a**
y	**a**	**y**
—	—	—
—	—	—



NAND devresi (= değillenmiş AND, yani zıtlayıcı)

Bilgisayara Nasıl Danışılır?

Buraya kadar bilgisayarın çalışma biçimi hakkında biraz olsun bir şeyler öğrendik. Artık bilgisayarın kimi uygulama olanaklarını daha ayrıntılı olarak gözden geçirmenin zamanı geldi. Bu bölümde, gözden geçirdiğimiz mantık işlemlerinden bellekte depolanmış verilere ulaşmada nasıl yararlanılacağını da görmüş olacağız. Bu tür veri aramalara daha çok büyük bilgisayarlarda başvurulmaktadır ama, burada anlatacağımız teknikler en basit ev bilgisayarlarında bile kullanılabilir.

Meslek hayatımızda olsun evimizde olsun sık sık başvurmak zorunda olduğumuz kimi bilgiler vardır; bunları, her ihtiyacımız oluşunda yeniden aramaya başlamak bıktırıcı olabilir. Böylesi işler için istediğimiz anda yeni veriler ekleyebileceğimiz ve istediğimiz verilere kolaylıkla ve hızlı bir biçimde ulaşmamızı sağlayacak bir programa ihtiyacımız vardır.

Gerçi şimdiye dek öğrendiklerimiz böyle bir programı yazmak için yeterli değil ama, programın temelinde yatan mantık ilkelerini gözden geçirebiliriz.

Tipik bir örnek olarak telefon fihristini ele alabiliriz; öyle ki aranan numara hem kolayca bulunabilmeli, hem de istendiği anda listeye yeni numaralar eklenebilmeli. Birinci fasikülde belleğe yerleştirilmiş bir listede yer alan bir telefon numarasını bilgisayarın nasıl arayacağını kısaca özetlemiştik. Bunu çeşitli düzeylerde üstüste seçimler yaparak ilerleyen bir araştırmaya benzetebiliriz. Bu seçimler gitgide ''incelmekte'' ve gitgide daha az sayıda seçenek arasından yapılmaktadır, ta ki tek bir cevap kalıncaya kadar. Şimdi bu arama faaliyetini daha yakından, adım adım izleyelim:

Kuyulubağ Sokağı 30 numarada oturan Mehmet Alım'ın telefonunu bulmak istiyoruz:

Soyadı = Alım

Bilgisayar belleğinde yer alan bütün isimleri tarayacak, hepsinin soyadını ''Alım''la karşılaştıracak ve uyanları seçecektir: Soyadı = Alım.

Bu birinci düzey tamamlandığında cevaba ulaşmaktan henüz çok uzak olacağız, çünkü elimizde kabarık bir ''Alım''lar listesi bulunacaktır. Bu yüzden ikinci bir düzeyde yeni bir seçme faaliyetini başlatmak üzere ikinci bir veriyi daha devreye sokacağız, aradığımız kişinin adını:

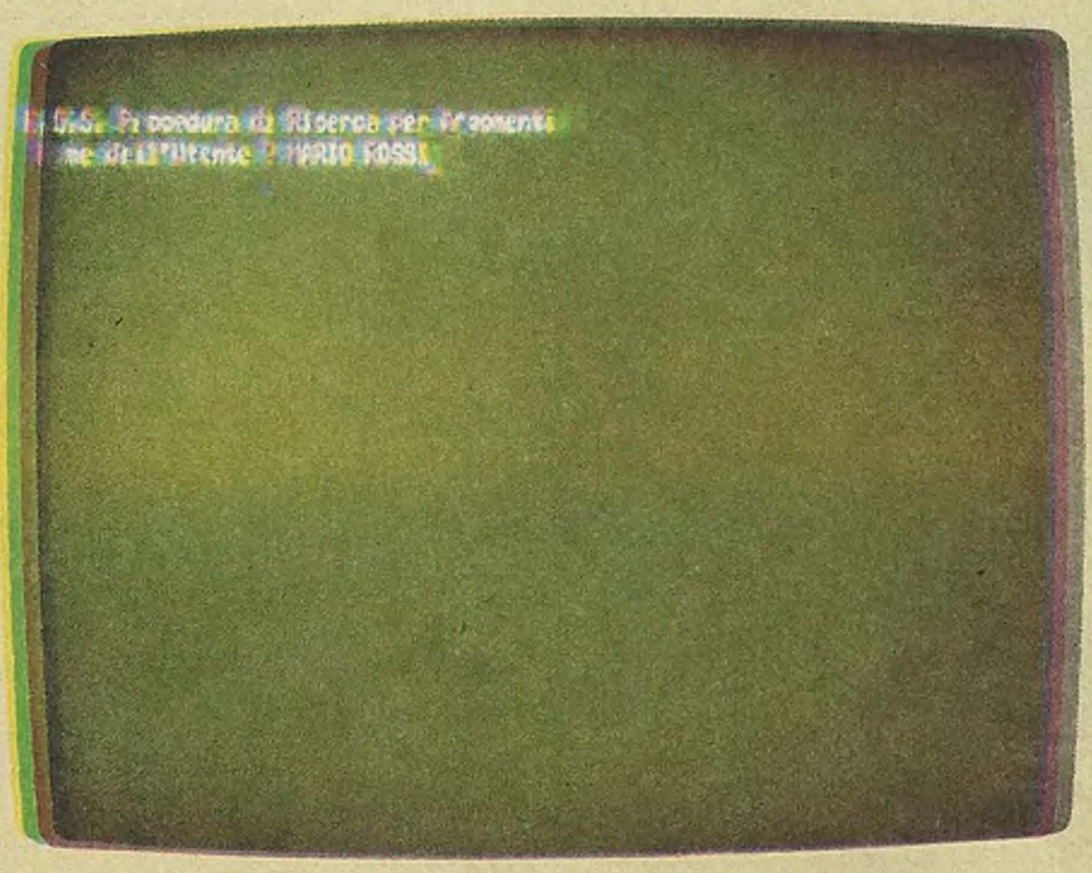
Adı = Mehmet

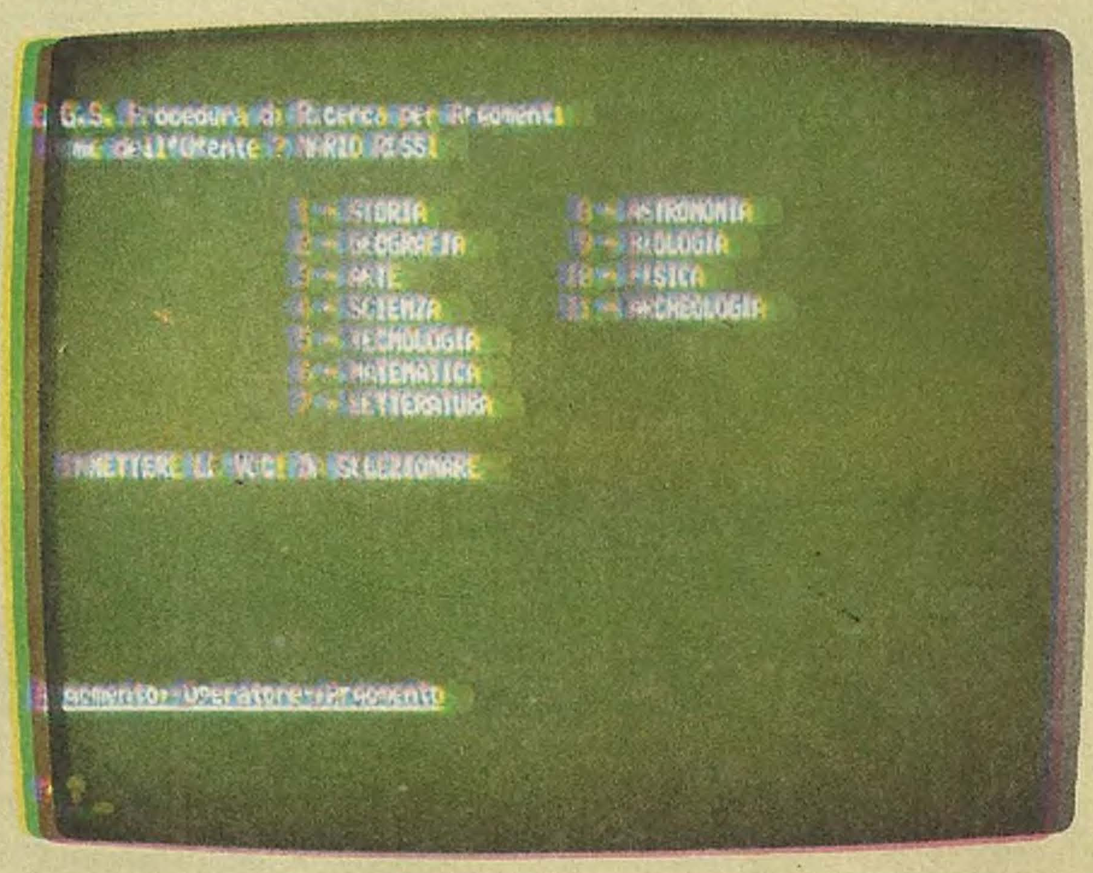
Bu iki komutla makinenin önce, adı ne olursa olsun bütün ''Alım''ları bulmasını, sonra da soyadı ne olursa olsun bütün ''Mehmet''leri bulmasını sağlamış olduk. Yani, birbirinden bağımsız iki ayrı liste elde ettik: Alım soyadını taşıyan telefon abonmanlarının listesi ve Mehmet adlılarınkinin listesini.

Oysa bu iki arama faaliyetini VE **(AND)** *mantık işlemi ile birleştirip cevabın (ya da cevapların) şu tek koşula uymasını sağlayabilirdik.*

Soyadı = Alım — VE — Adı = Mehmet

Böylelikle bilgisayar bize bütün ''Mehmet Alım''ları birden verecek. Ancak bu arama da bize istediğimizden fazlasını verebilir. Dolayı-





sıyla ya üçüncü bir arama düzeyine başvurmamız ya da koşulu üçüncü bir veriyle zenginleştirmemiz gerekir. Üçüncü bir seçim ölçütü olarak adresi katalım; ve diğer ölçüte uyan kayıtların bu ölçütle de karşılaştırılmalarını isteyelim. Bilgisayarın gerçekleştirmesi gereken koşul tümüyle şöyle olacaktır:

Soyadı = Alım
(birinci düzey)
— VE —
Adı = Mehmet
(ikinci düzey)
— VE —
Adresi = Kuyulubağ Sok. 30

Artık bilgisayar bize tam istediğimiz bilgiyi verebilir, yeter ki aynı adreste iki tane Mehmet Alım oturuyor olmasın. Aslında istendiği takdirde bu son durumda bile ayırdedici bir ölçüt bulunabilir.

Şimdi tekrar başa dönelim ve Mehmet Bey'in soyadını hatırlamadığımızı varsayalım Alım mıydı? Balım mı?

Halbuki adresini bilseydik, bu, büyük bir sorun yaratmazdı. Aramamızı bilgisayardan aynı üç koşulu gerçekleştirmesini isteyerek sürdürebilirdik. Ancak bu sefer birinci koşulu şöyle yazmak zorunda kalırdık:

Soyadı = Alım VEYA Balım

O zaman bilgisayarın şu üç koşulu birden gerçekleştirmesi gerekecekti:

1. Soyadı = Alım veya Balım
2. Adı = Mehmet
3. Adresi = Kuyulubağ Sok. 30

Bu üç koşulu birarada şöyle yazabiliriz:

Soyadı = (Alım VEYA Balım) VE
Adı = Mehmet VE
Adresi = Kuyulubağ Sok. 30

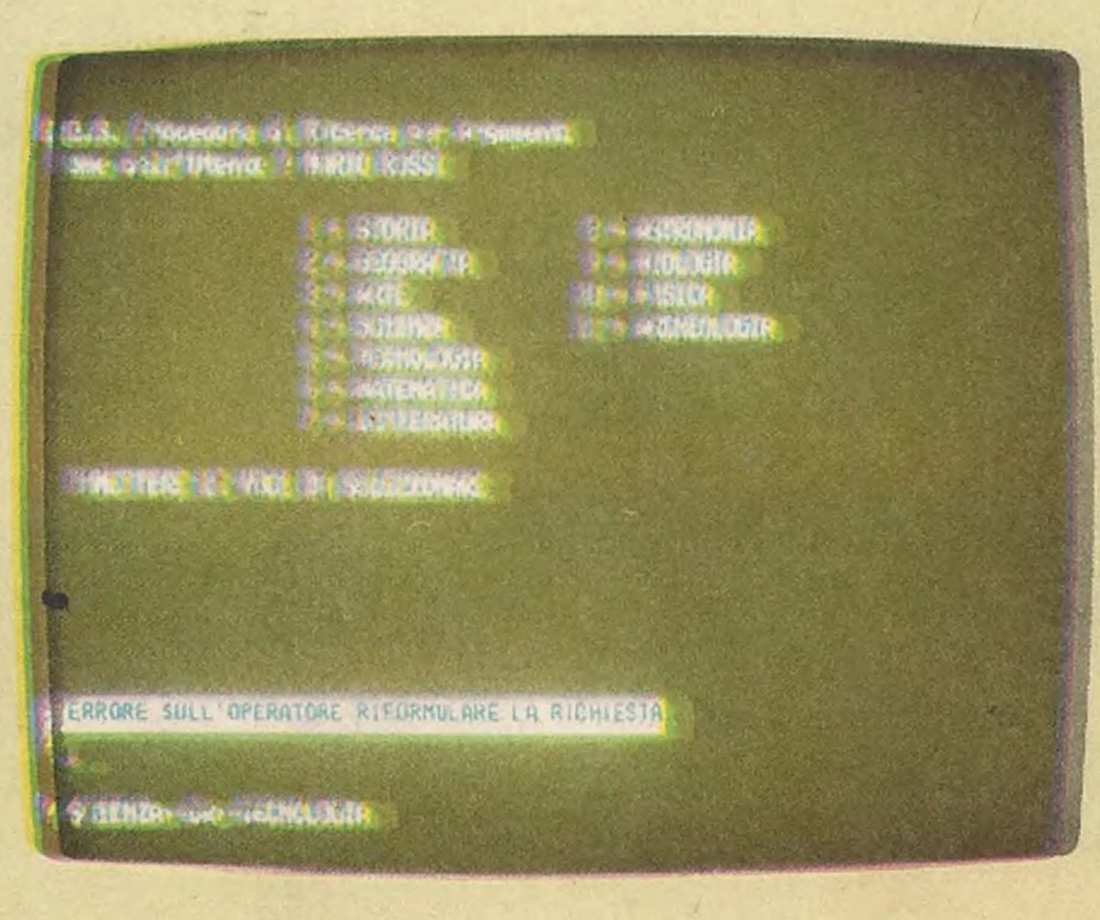
Ad konusundaki tereddüdümüz istediğimiz bilgiyi elde etmemizi engellemiyor çünkü, üçüncü koşul birinci koşulun belirsizliğini telafi edecektir. Bilgisayar Kuyulubağ 30 adresini bulacak, o adreste oturan, Alım da olsa Balım da, telefon numarasını bize verecektir. Çünkü o adreste kayıtlı abonmanın soyadını, bizim soruda verdiğimiz bilgilerin tümü ile, yani hem Alım'la hem Balım'la karşılaştıracaktır.

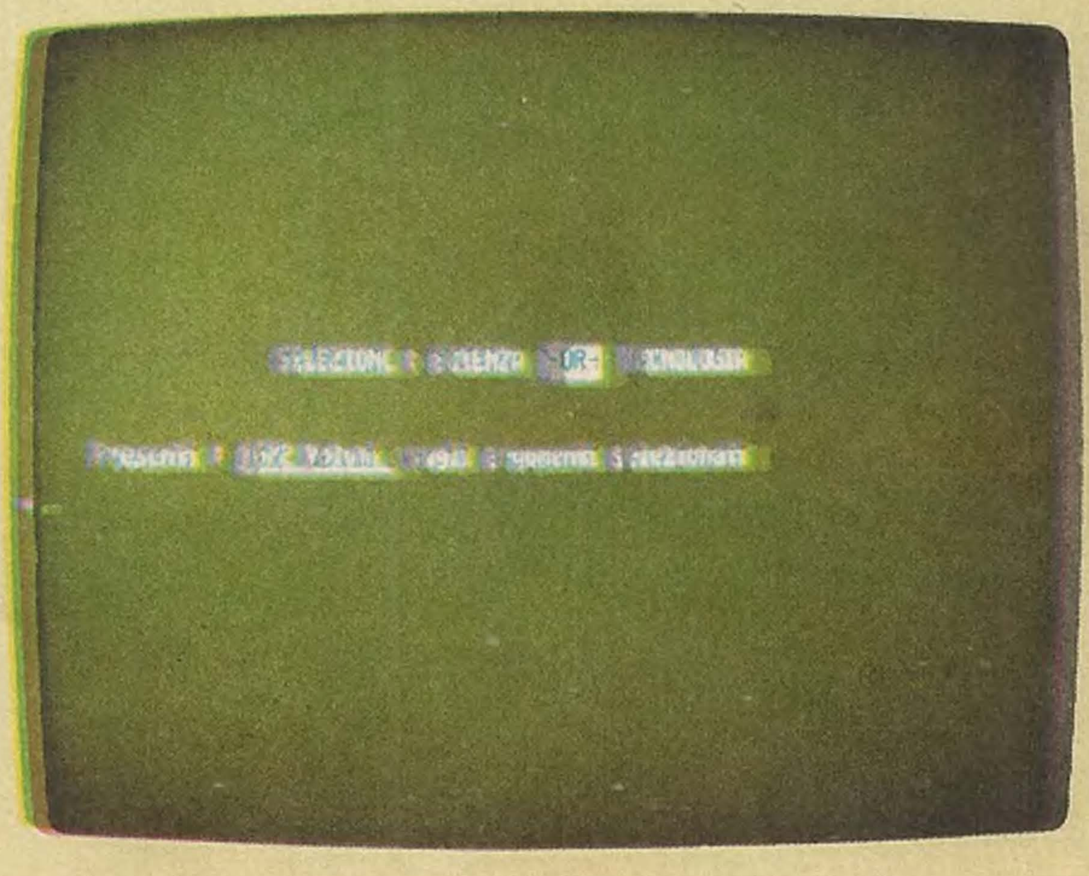
Genel olarak söyleyecek olursak, böyle bir arama faaliyetinde yapılan, soruda verilen koşulların verdiği bilgileri içeren bir veri kaydı aramaktır.

Kuşkusuz Mehmet Bey'in ne soyadını ne de adresini bilseydik durum farklı olurdu. Böyle bir durumda bilgisayar bize, çıktı olarak rehberde yer alan bütün Mehmet'lerin telefon numaralarını sağlardı. Yalnızca tek bir veri kaydını belirleyecek parametreleri sağlayamadığımızdan kuşkusuz liste çok uzun olacaktı.

Bu tür programların sayısız uygulamaları vardır. Örneğin büyük kütüphaneleri ele alalım. Buralarda kütüphane bulunan her kaynak şu verileri içerecek şekilde kaydedilmiştir: Yazarın adı, kitabın (ya da makalenin belgenin vb.) başlığı, türü (roman, inceleme vb.), konusu (ya da konuları), hangi dilde yazıldığı, kütüphanenin hangi rafında bulunduğu, ne kadarlık bir süre için kütüphaneden dışarı çıkarılabileceği vb.

Bu bilgilerin otomatik olarak ulaşılabilir bir hale sokulması dikkatsizlikten ve unutkanlıktan do-



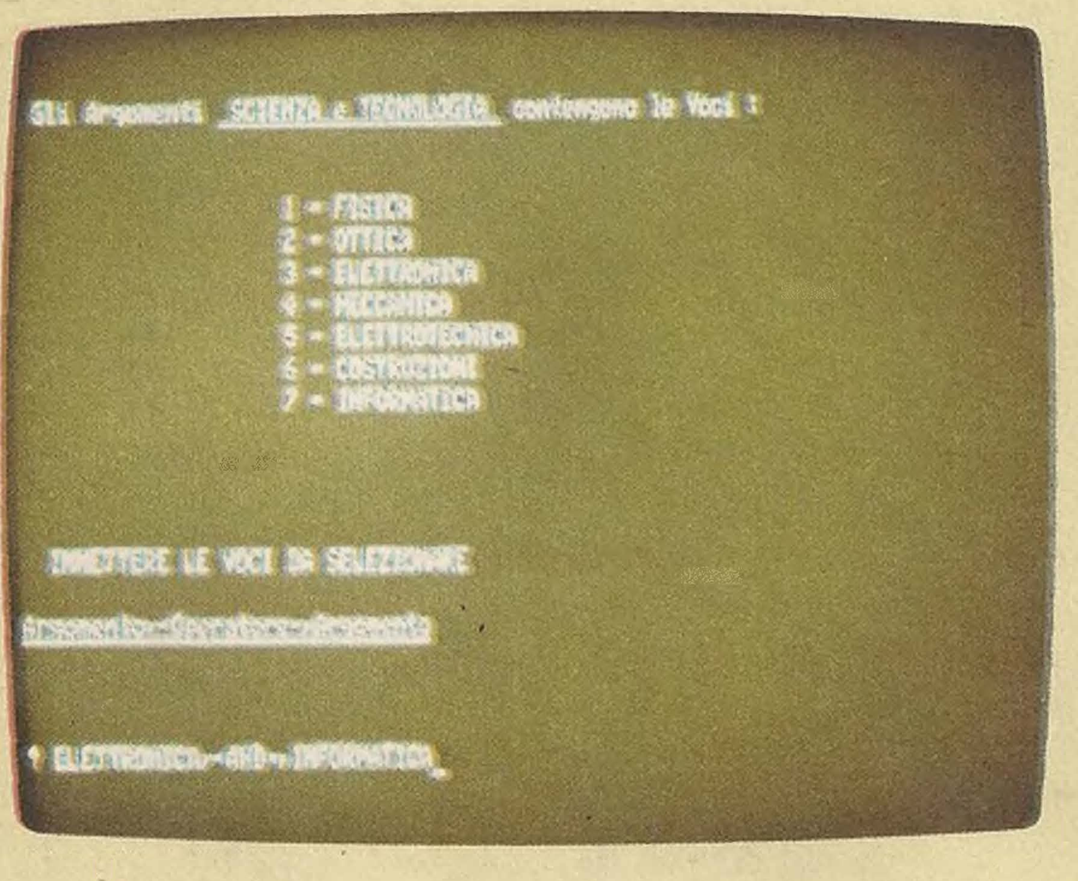


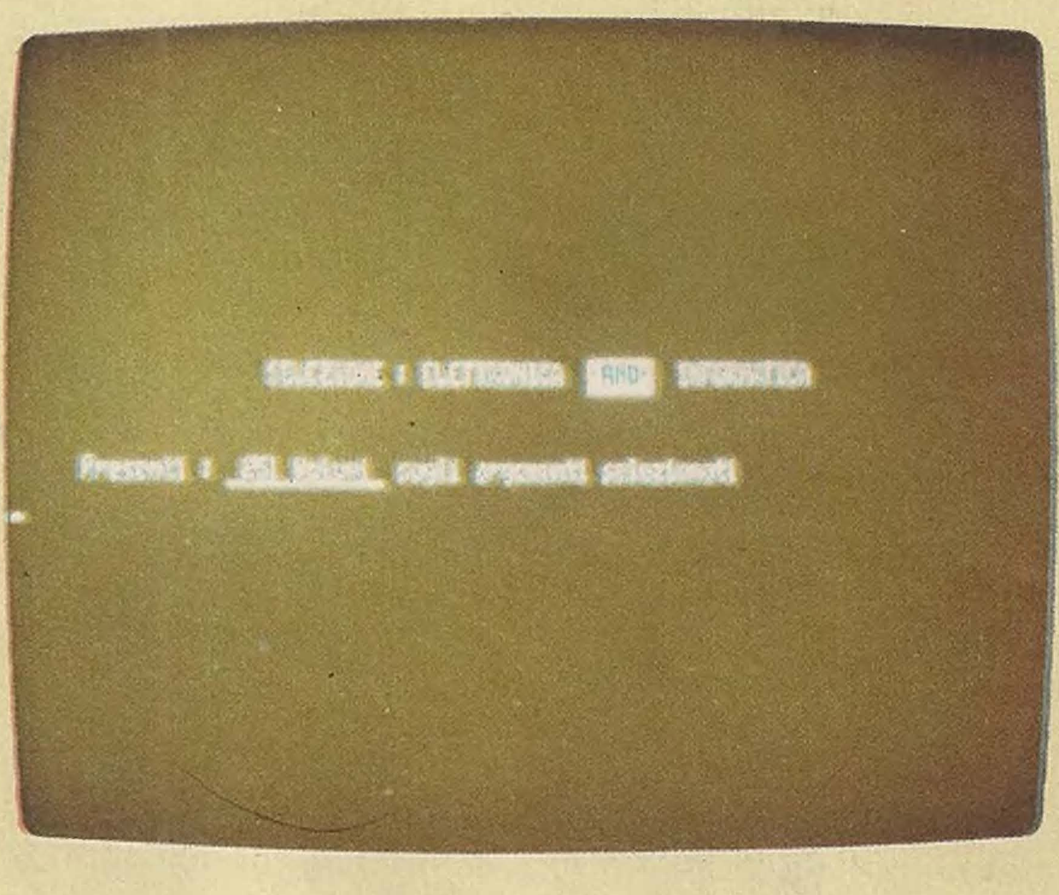
Kaynakçalar bilgisayarda biriktirilerek, kütüphane araştırmaları kolaylaştırılabilir.

ğabilecek çeşitli eksiklikleri, zaman israflarını ortadan kaldıracaktır. Ayrıca böyle bir bilgisayar arşivi olmaksızın belirli bir konuda yayınlanmış bütün *yayınları taramak da zaten hayal olurdu.*

Oysa bugün bilgi işlem kullanan arşivlerde Boole işlemlerini kullanma olanağı, bu tür kaynakça taraması çalışmalarını çocuk oyuncağına çevirdi. İkinci bir örnek olarak belirli bir konuda bir kaynakça hazırlamaya karar verdiğimizi düşünelim ve bu çalışmada atılması gereken adımları gözden geçirelim.

Bir bilgisayarın önüne oturduk, tabiî ilk koşul böyle bir arama programının bilgisayarın program belleğine yüklenmiş olmasıdır. Bu koşulun gerçekleşmiş olduğunu varsayıyoruz. İlk adımda bilgisayar size adınızı sorabilir. Böyle bir sorunun çeşitli nedenleri olabilir; örneğin birtakım istenmeyen kişilerin bilgilere ulaşmasını önlemek, bazı gizli metinlerin gizliliğini sağlamak için ulaşabileceğiniz kaynakların sayısını sınırlandırmak, ya da aynı konularda araştırma yapan araştırmacıları birbirinden haberdar etmek gibi.





Bundan sonra, ekranda kullanabileceğiniz komutların ve verilerin sınıflandırılmasında kullanılmış kodların listesini içeren bir MENÜ belirir.

Diyelim ki bizi ilgilendiren konu atom enerjisinin doğal çevre üzerindeki etkileri; bu konuda yazılmış kitapları bulmak için klavyeyi kullanarak ekrana,

Atom enerjisi VE doğal çevre

yazdık. Ekranda cevaben böyle bir şey belirebilir:

konu -İŞLEMCİ- konu

Bilgisayar bize program dilinin söz dizinine uymamız için uyarıyor. Bu durumda işlemcinin etrafındaki tirelere de dikkat ederek, ve VE işleminin işaretinin AND olduğunu hatırlayarak komutu tekrar şöyle yazabiliriz:

Atom enerjisi -AND- doğal çevre

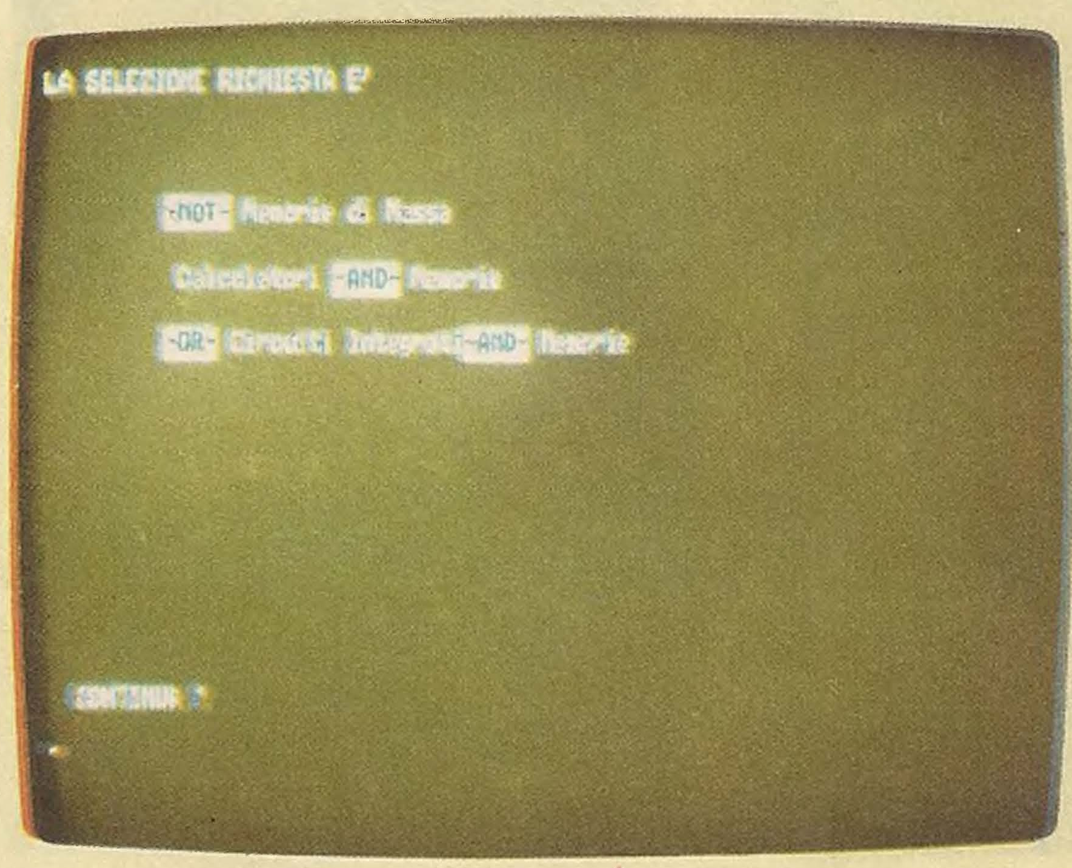
Komut makinenin bildiği söz dizimine uyduğu takdirde makine bellekte konu başlığı olarak bu başlığın kaydedildiği bütün metinlerin sayısını bize sunacaktır. Eğer bu sayı çok fazla değilse, bu metinlerin özelliklerini de içeren bir listeyi görmek isteyebiliriz. Diyelim ki bizim konumuzdaki liste çok uzun, bütün kayıtları tek tek gözden geçirmeye vaktimiz yok. Dolayısıyla tarama alanını daraltmak istiyoruz. Bunu bir komutla bilgisayara bildirmemiz gerekir. Komutu alan bilgisayar, bize atom enerjisi ve doğal çevre konularındaki var olan alt başlıkların bir listesini verecektir. Bizi ilgilendiren alt başlıklardan birinin ihmalkârlık olduğunu varsayalım.

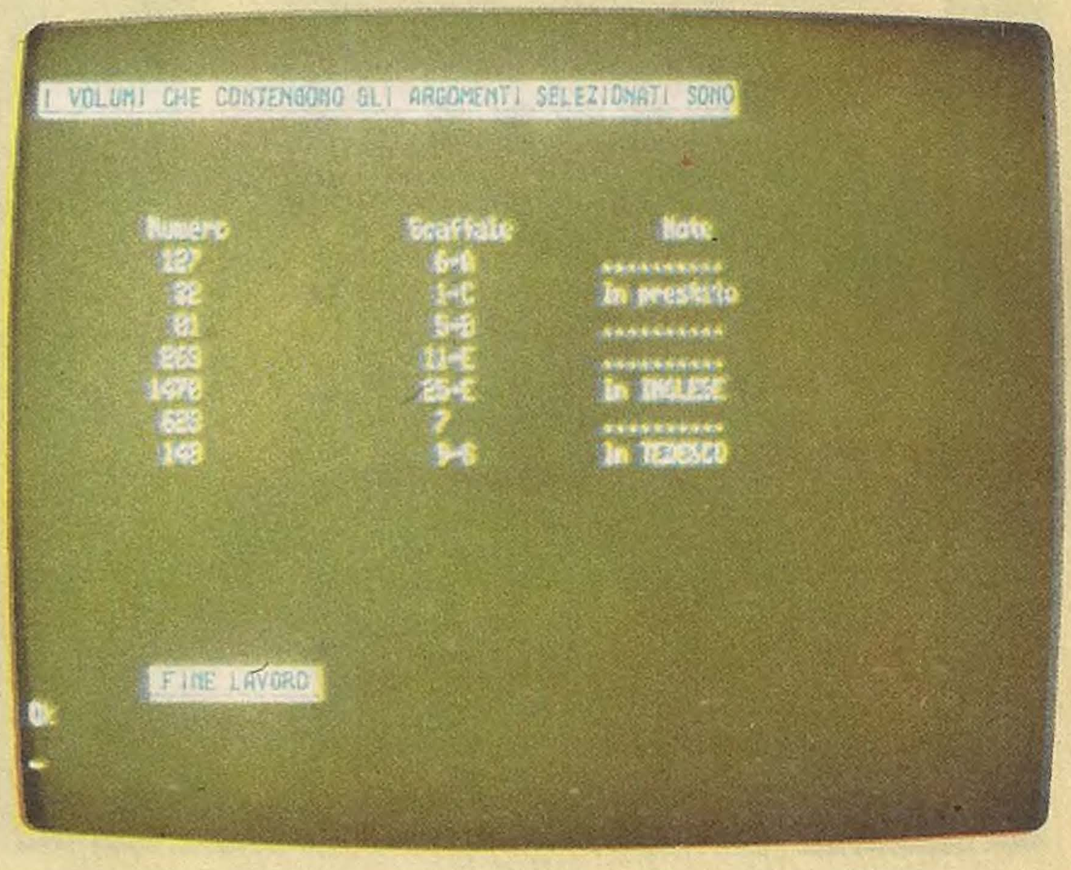
Bunun üzerine komutumuzu tekrar şöyle yazmamız gerekiyor:

Atom enerjisi -AND- (doğal çevre -OR- ihmal)

Eğer bu komuta karşılık bilgisayarın bize bildirdiği kaynak sayısı başa çıkamayacağımız büyüklükte değilse kaynak listesini taramaya, istediğimiz kaynaklar hakkında bilgisayara daha ayrıntılı sorular sormaya başlayabiliriz. Yok, eğer hâlâ istediğimizden daha yüksek bir sayı elde ediyorsak, arama alanını daha da daraltmak, arama koşulumuza yeni ölçütler eklemek söz konusu olurdu. Örneğin, bu kez ilgilendiğimiz kaynak listesini yayın tarihleri bakımından sınırlayabilir ve ilgilendiğimiz konuda son 10 yıl içerisinde yayınlanmış yayınların listesini isteyebilirdik. Veya bilgisayara yalnızca çeviri ya da yalnızca telif yayınlarla ilgilendiğimizi bildirebilirdik.

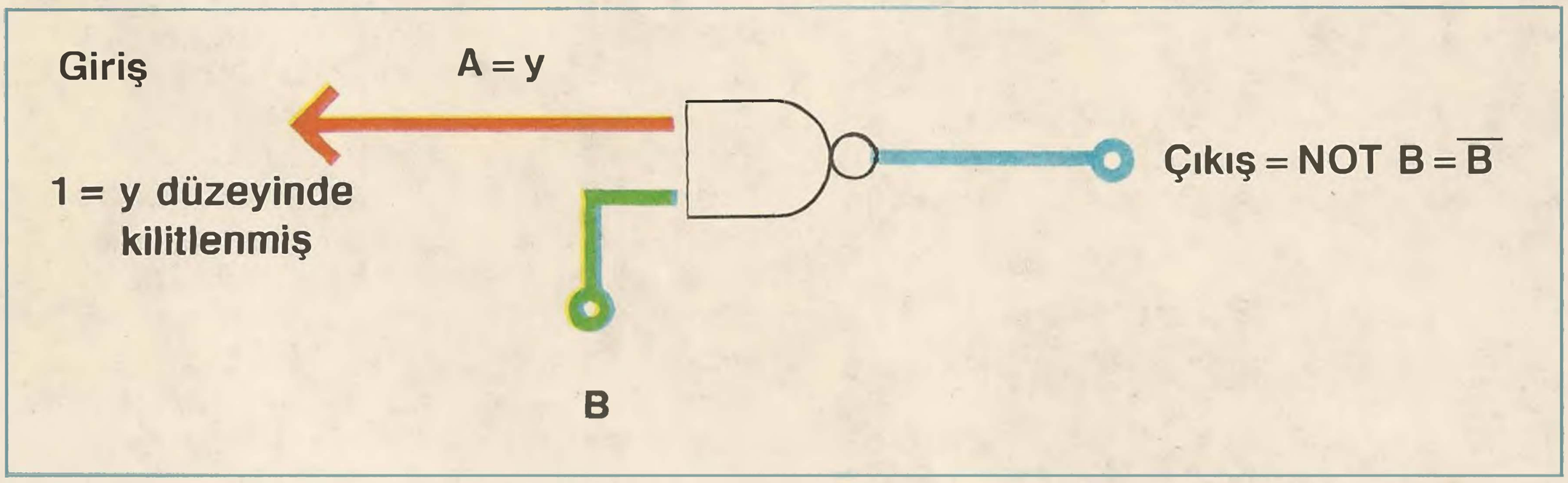
Böylesi olanaklara sahip bir tarama programı ile kişi yalnızca zamandan tasarruf etmekle kalmayacak, kısa zamanda istediği herhangi bir konunun kuşbakışı görünüşünü elde etme olanağını da bulacaktır.





Çıktı, B girdisinin tersidir, ya da başka bir deyişle, çıktı, değillenmiş bir B'dir (NOT B).

Bu durumda girdinin B olduğu durumlarda çıktı da, yani B'nin zıtlanmışı da (değillenmiş B) B ile yazılır. Bu, aşağıdaki şekilde gösterilmektedir:



Şekilde periyodik olmayan dalgalar ileten iki giriş bulunmaktadır.

Çıktıyı elde edebilmek için yine y düzeyini 1 ile a düzeyini de 0 ile gösterelim.

Böylelikle, sayfa 83'teki şekilde de görüldüğü gibi şu sonuçlar elde edilmektedir:

1. Kesim:

A'nın ve B'nin düzeyleri Yüksek, yani **A = 1, B = 1**. Öyleyse **A AND B = 1**, yani Yüksek: **y**.

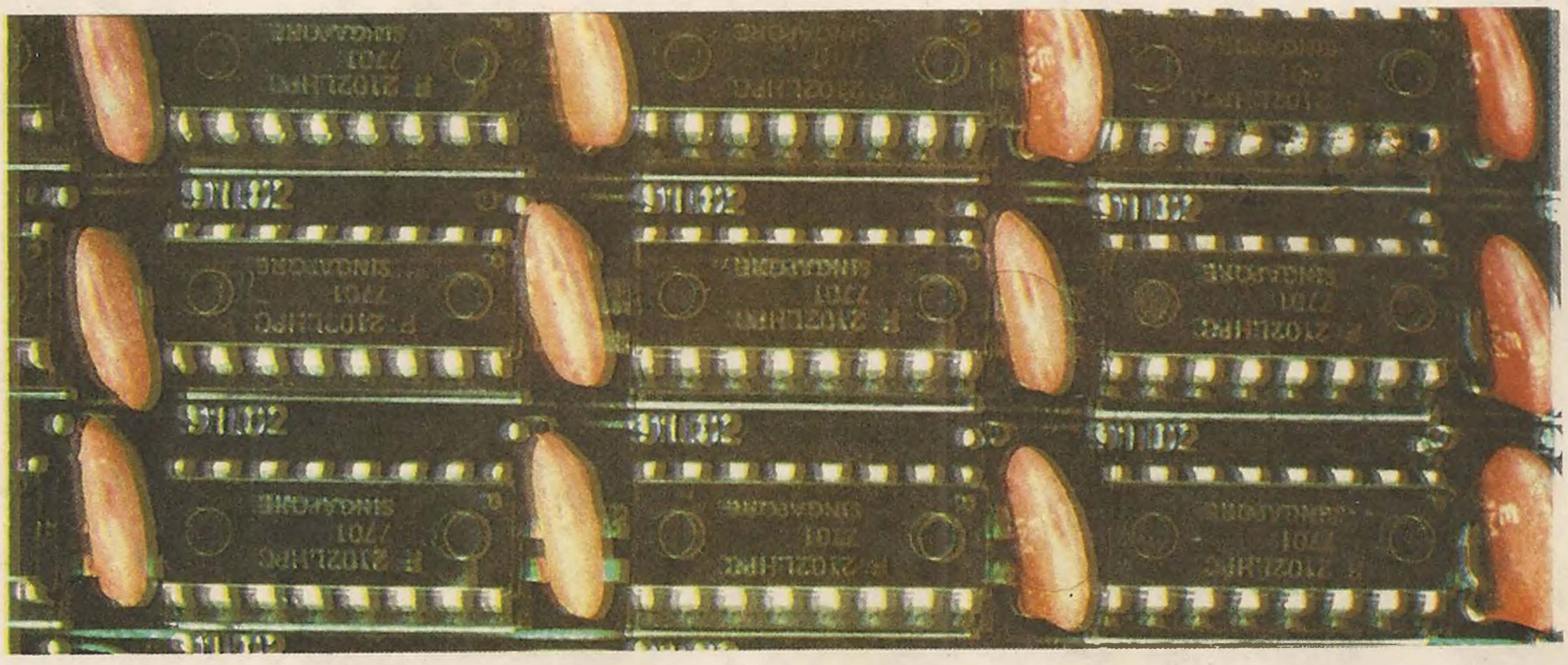
2. Kesim:

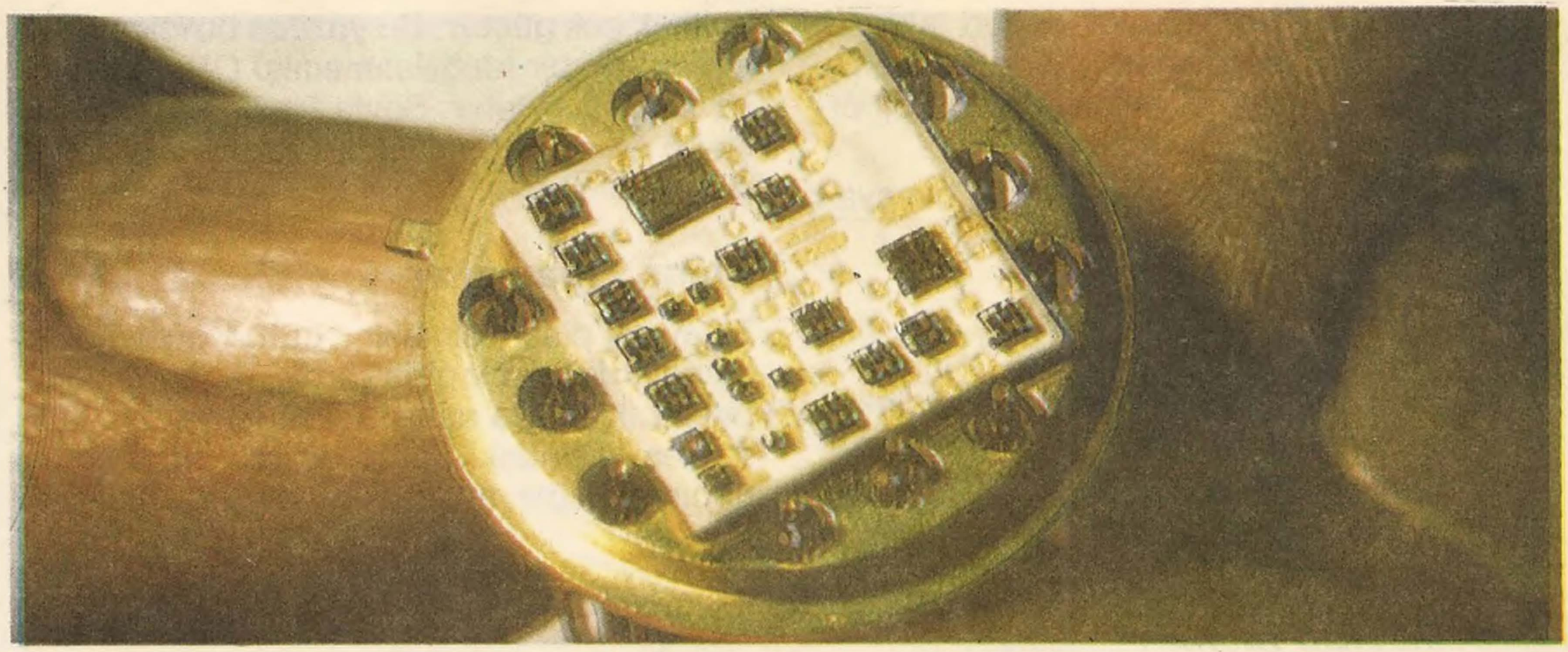
A Yüksek, **A = 1**; B Alçak, **B = 0**; öyleyse **A AND B = 0**, yani Alçak: **a**.

Yandaki tabloda 15 kesimin biraraya getirilmiş halini görüyorsunuz

Kesim	A	B	A AND B	$\overline{\text{A AND B}}$
1	1	1	1	0
2	1	0	0	1
3	0	0	0	1
4	0	1	0	1
5	1	1	1	0
6	1	0	0	1
7	0	0	0	1
8	0	1	0	1
9	0	0	0	1
10	1	0	0	1
11	0	0	0	1
12	0	1	0	1
13	1	1	1	0
14	1	0	0	1
15	0	0	0	1

Bir kart üzerinde kümelendirilmiş tümleşik devreler.





Silisyum safirden yapılma bir yonganın çok büyütülmüş resmi.

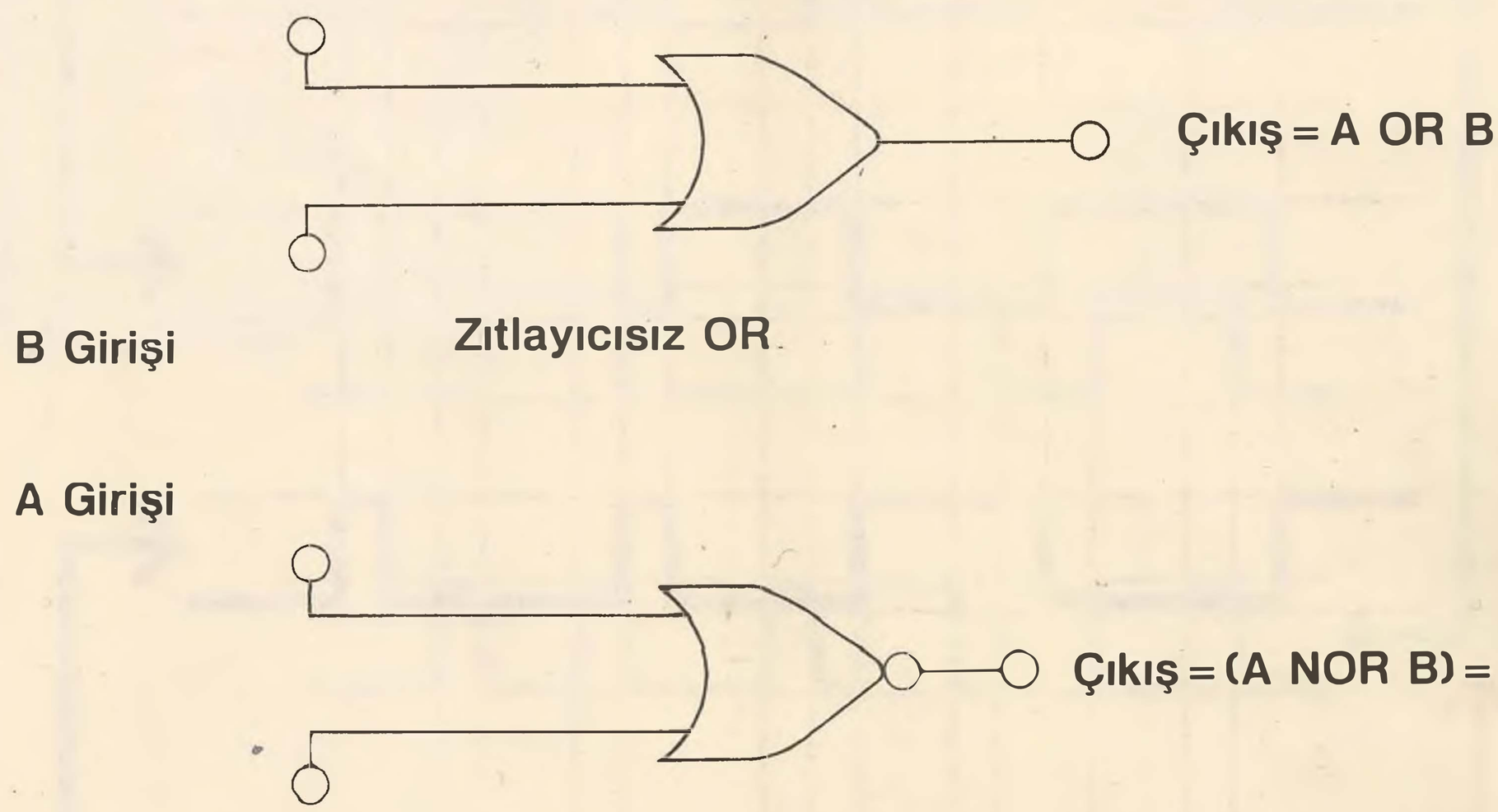
OR (VEYA) İşlemcisi

Artık alıştığımız 1 = **y** ve 0 = **a** eşitliklerinden hareket ederek, OR (VEYA) işlemini gerçekleştiren devrenin doğruluk değeri tablosunun işleminkiyle aynı olduğunu görebiliriz.

Bu devrelerin de iki türü vardır; değillenmiş ve değillenmemiş.

Aşağıdaki şekilde her ikisi de gösterilmiştir.

A	B	OR	$\overline{OR}$
y	**y**	**y**	**a**
y	**a**	**y**	**a**
a	**y**	**y**	**a**
a	**a**	**a**	**y**



Zıtlayıcısız OR

Zıtlayıcılı OR

Değillenmiş OR'un doğruluk değeri tablosu, OR'un tablosu değillenerek elde edilir. Bu sayfadaki şekil, sinyal biçimleri ile ilgili bir örnek daha vermektedir.

Bu örnekte de girdilerin periyodik olmayan dalgalar olduğu varsayılmaktadır.

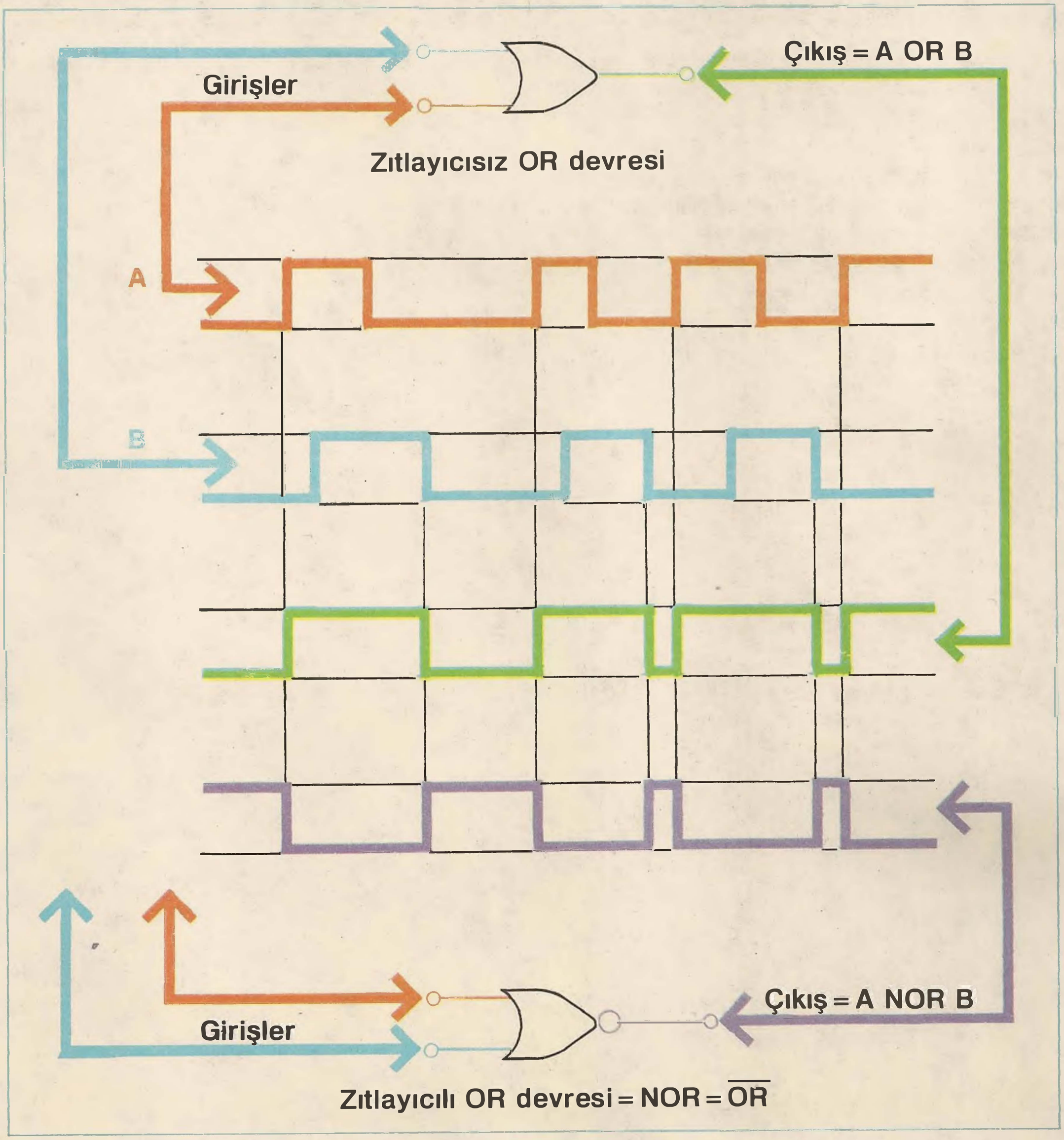
XOR (DAR VEYA) İşlemcisi

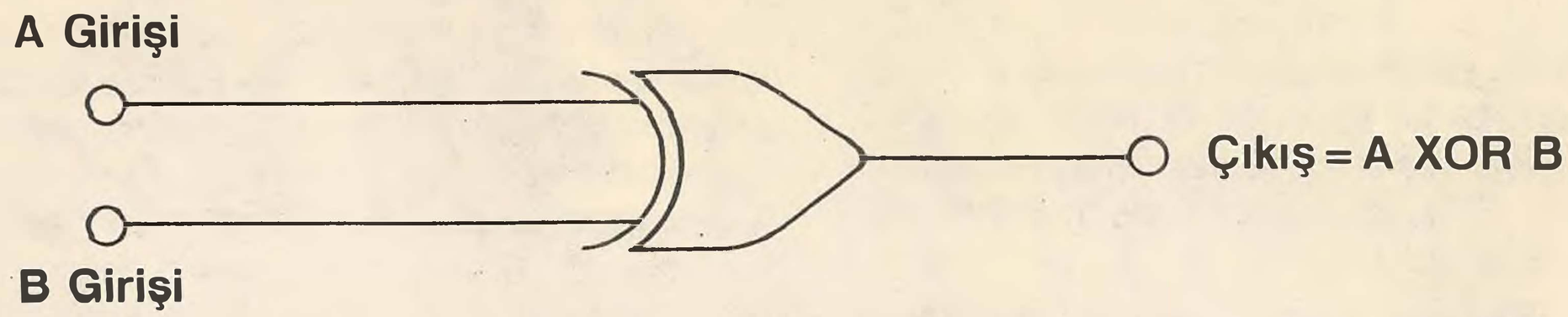
Bu devrenin de doğruluk tablosu, işleminkiyle aynıdır. Devre sayfa 91'de üst taraftaki şekilde çizilmiştir.

Bu devrelerin değillenmişini imal eden firma bulmak çok güçtür. Bu yüzden bu işlemi elde etmek için bir (değillenmemiş) OR devresine bir zıtlayıcı eklenir. Sayfa 91'de alttaki şekilde bu devrenin bağlantıları ve sinyal biçimleri gösterilmiştir.

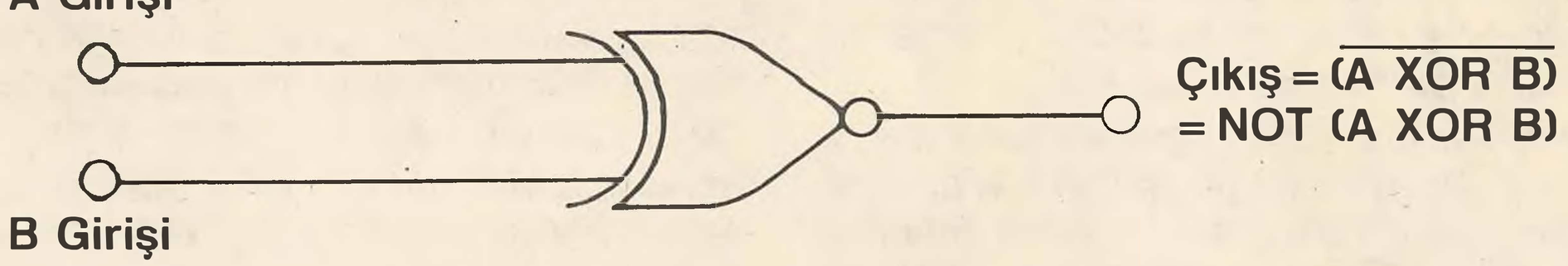
Diğer Zıtlayıcılar

Zıtlayıcılı olsun olmasın, bütün güçlendirici devrelerde tek bir giriş vardır. Diğer tüm devrelerin iki ya da daha çok girişi vardır. XOR devresi dışındaki bütün devrelerde iki girişi birbirine bağlayarak tek bir giriş elde edebiliriz.



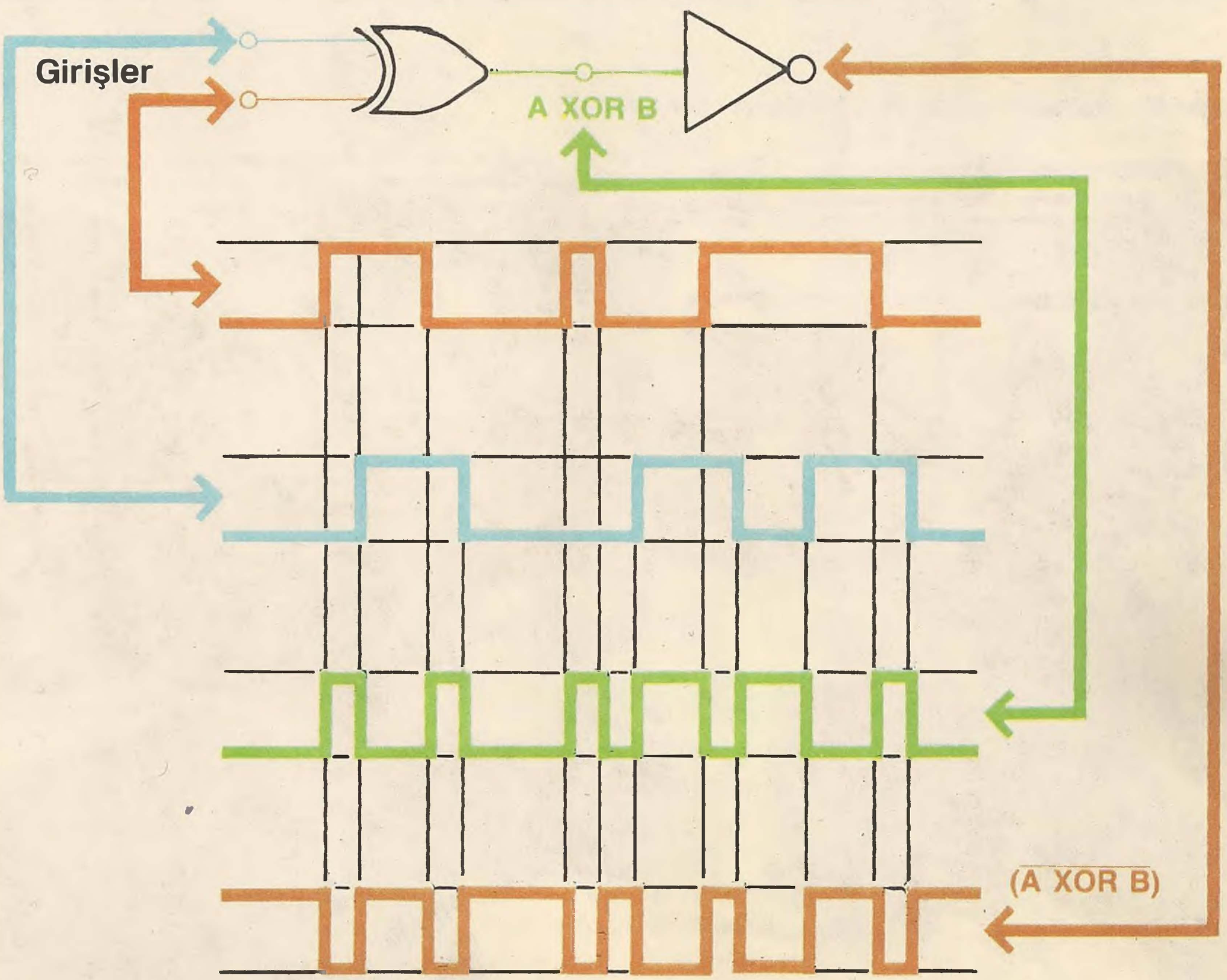


Zıtlayıcısız OR



Zıtlayıcılı OR

Zıtlayıcısız XOR devresi



İletişimde Devrim

Yaklaşık bir yüzyıldır bütün dünyada PTT hizmetleri hantal ve yerel bir iş alanı olarak sürüp gitmekte. Neredeyse her ülkede devlet tekelinde bir PTT, o ülkenin iletişim ihtiyaçlarını karşılamakta.

Ancak, bilgisayar teknolojisindeki gelişmeler beklenmedik bir biçimde, bu alandaki yaygın devlet tekelciliğinin sorgulanmasına yol açıyorlar. Öyle görünüyor ki, ya bilgisayarların da tekelleştirilmesi gerekecek (ki bu tasarlanması bile güç bir girişim) ya da iletişim hizmetleri de rekabet şartlarına açılacak.

Bu alandaki en çarpıcı gelişme Amerika'da görüldü. Bir devlet kuruluşu olmamakla birlikte, Amerikan pazarı üzerinde bir tekel denetimi kurmuş olan ATT, yeni teknoloji sayesinde giderek artan rekabetin sonucu, sonunda geçtiğimiz yıl içerisinde parçalanmak zorunda kaldı. Aynı biçimler almasa da benzer eğilimleri başka ülkelerde de görmek mümkün; İngiliz Hükümeti British Telecom'un % 49'unu satışa çıkarırken, Fransa'nın sosyalist hükümeti PTT hizmetleri konusunda giderek özel firmalarla işbirliğine yöneliyor.

Bu gelişmelerin temelinde dünya iletişim ağlarını temelinden sarsan iki büyük teknolojik sıçrama yatıyor; sayısal bilgisayar ve optik lif. Dünün bakırdan telefon ve telgraf telleri ile mekanik santralleri ancak tek tip mesaj iletimine yarıyorlardı: Telefon görüşmeleri, telgraf ya da teleks mesajları. Oysa bugünün sayısal bilgisayarları sayesinde, görüntü olsun, ses olsun, veri olsun iletilemeyecek bir şey kalmadı. Optik lif ise, bilgisayarın VAR/YOK (ON/OFF) darbelerine çevrilmiş bu iletilerini çok daha hızlı bir biçimde iletiyor. Ayrıca iletişimde uydulardan yararlanmak da deneme aşamasından çıkıp yaygın bir gerçeklik olmak üzere.

Hepimiz telekomünikasyonun tek ve ulusal bir iletişim ağı çerçevesinde gerçekleşmesine öyle alışmışız ki, bu gelişmelerin sağlayacağı olanakları düşlemek bile çoğumuza güç geliyor. Bir kere bilgisayar ağları sayesinde ilgiye ve ihtiyaca göre çeşitli amaçlara hizmet eden değişik büyüklükte iletişim ağları oluşturmak, bir sorun olmaktan çıkıyor. Bu olanağın belki ilk farkına varanlar çokuluslu dev şirketler oldu.

Chase Manhattan Bankası'nda bilgisayar terminalleri.

1 Bilgisayarların yaygınlaşması ve telekomünikasyonu kolaylaştırması, iletişim sektöründeki devlet tekellerinin kalkmasına yol açıyor.
2 Aynı ilgileri paylaşan şahıslar küçük yerel iletişim ağları geliştiriyorlar.
3 Basın-yayın dünyasında köklü değişiklikler: Gazetelerin anında haber toplama olanakları çok arttığı gibi, haberleri abonmanlara bilgisayar yoluyla anında ulaştırabiliyorlar.
4 Bu tür banka ofisleri tarihe karışıyor. Banka hizmetleri doğrudan bankanın merkez bilgisayarı ile ilişki kurularak yapılıyor.

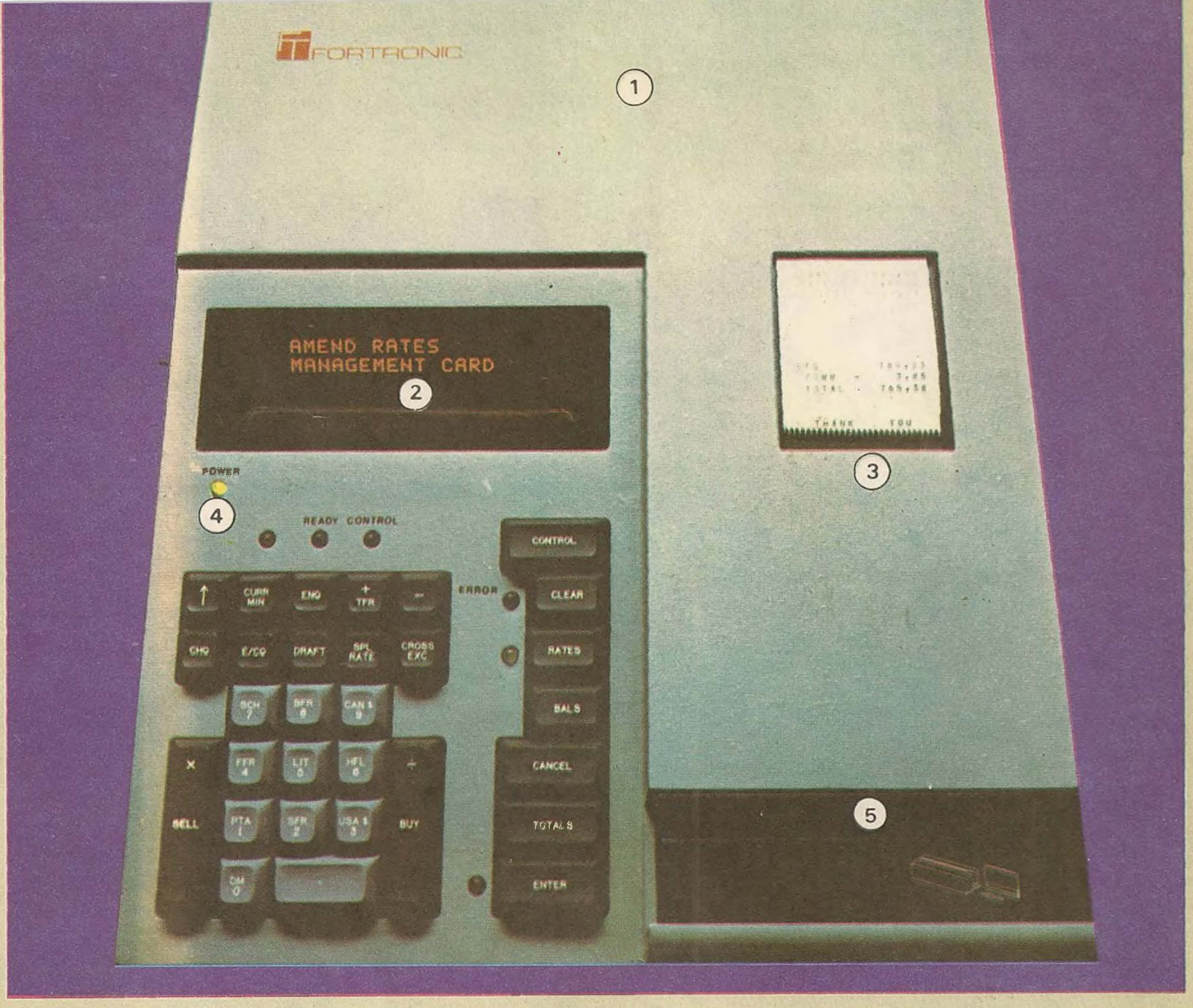
■ Büyük paylaştırıcılar şubeler için bir merkez işlevi görürler. Bu aygıtlar büyük santrallere kıyasla çok daha az yer tutarlar.
● Şubeler merkeze(■) kablolarla bağlıdır.

Örneğin bütün dünyada oto pazarlayan Ford, yakın zamana kadar çeşitli modellerin denetimini Kuzey Amerika, Güney Amerika ve Doğu Asya'daki özerk merkezlere bırakıyordu. Ancak yeni olanaklar sayesinde kendi iletişim ağını örgütleyen Ford firmasında denetim artık tek bir merkezden yapılıyor.

Ancak bilgisayar teknolojisinin ucuzluğu, küçük yerel firmalara, çok-uluslu büyük kuruluşlar karşısında beklenmedik rekabet olanakları yaratıyor. Örneğin son yıllarda dev ITT, yerel ortaklarına giderek daha büyük bir pay bırakmaya zorlanıyor.

Diğer yandan, çok daha küçük yerel gruplar da bu yeni iletişim tarzlarının olanaklarını keşfediyorlar. Örneğin Güneybatı Fransa'da bir çiftçi derneği daha önce sözünü ettiğimiz videoyazımdan yararlanarak kurdukları iletişim ağı sayesinde tarımla ilgili deneyimlerini ve yeni verileri paylaşmakta.

O halde bilgisayarlı iletişimin, ya da Telematik'in, tekelci kuruluşlar üzerindeki etkilerini iki



Ekranlı (2), bilgiyazarlı (3), klavyeli (4) ve manyetik fiş okuyuculu (5) küçük bir banka terminali (1).

boyut üzerinde değerlendirebiliriz. Bu yeni teknolojiler, tekelci kuruluşlara, bir yandan ele geçirdikleri piyasayı çok daha yakından ve günü gününe, hatta dakikası dakikasına denetleme olanağı verirken, diğer yandan tüketicilere kendi ihtiyaçlarını karşılamak üzere alternatif iletişim ağları kurma olanağı tanıyor; böylelikle firmaların tekelci niteliğini tehlikeye sokuyor.

Doğal olarak bu gelişmeler, dünyadaki çeşitli PTT yetkililerince ve hükümet sözcülerince biraz tedirginlikle karşılanıyor.

Örneğin Almanya'da ''Posta Hizmetlerinden Yararlanan Yurttaşlar Derneği''nin Başkanı Wilhelm Hübner, Alman PTT'sinin bir tür ''standartlar enstitüsü'' işlevini üstlenerek telekomünikasyon hizmetlerini serbest piyasaya açmasını isterken, Alman PTT'sinin Müdürü Konrad Schmidt geçmişte pekâla yeterli bir telekomünikasyon hizmeti verdiklerini, bunun gelecekte de aynen devam etmemesi için bir neden olmadığını söylüyor. Dahası, Alman PTT'si, görüşmeleri basına kapalı bir kurum olan Avrupa Posta ve Telekomünikasyon Yetkilileri Konferansı aracılığı ile, telekomünikasyon hizmetlerinin ve ücretlerinin Avrupa çapında ve devlet tekellerince düzenlenmesinde bir eşgüdüm sağlanmasının yollarını araştırıyor.

Bu girişimden tedirgin olan yalnızca tüketici grupları değil; Uluslararası Ticaret Odası da bu gelişmeleri kaygı verici bulduğunu açıkladı.

Hükümetlerin iletişim tekelini ellerinde tutmakta direnmeleri de kolayca anlaşılabilir. Yüzyıllık alışkanlık ve yeni teknolojinin etkilerinin öngörülemezliği bir yana, PTT gelirleri devlet bütçelerinde önemli bir gelir kalemi oluşturuyor.

Örneğin Japonya yakın gelecekte telekomünikasyon hizmetlerini uluslararası rekabete açacağını açıkladı. Ancak daha bu gerçekleşmeden, Japonya'nın dış ülkelerle yürüttüğü telekomünikasyon hizmetlerinin önemli bir bölümünü mevzuattaki boşluklardan yararlanarak devralan Western Union şirketi, Japon

bütçesinin yılda 1 milyon dolarlık geliri kaybetmesine neden oluyor.

Yine de varılan yasal düzenlemeler ne olursa olsun, uzak mesafelerle kurulan iletişimin ancak posta, telgraf ve telefon gibi devlet denetimindeki iletişim ağlarıyla yürütülmesi artık tarihe karışıyor. Bu tartışılmaz görünüyor.

Diğer yandan, bugün yaşanmakta olanın henüz bir geçiş dönemi olduğu da bir gerçek. Nisbeten ucuz olan ve fazla yatırım gerektirmeyen yeni teknolojilerde her yeni ilerleme, her yeni buluş, buluşu yapan firmaya genişleyen piyasada hatırı sayılır bir avantaj sağlıyor. Dolayısıyla teknolojik ilerleme, rekabet şartlarına yeni alanlar katmakla kalmıyor, küçük ve yeni firmaların da bu alanlara girmesine olanak veriyor. Tabii, rekabetin artmasının doğal sonucu olarak, hem iletişim hizmetlerinin fiyatı ucuzluyor, hem de tüketicilerin seçim olanakları artıyor.

Ancak teknolojik ilerlemenin bu hızda süreceğini ummak güç. Bu patlama dönemi bitip, teknolojik ilerleme istikrarlı bir büyüme düzeyine ulaştığı zaman piyasadaki bu çoğulculuk sürecek mi? Yoksa tarihteki diğer teknolojik devrimlerde olduğu gibi, rekabetçi ve çoğulcu bir başlangıç evresini yeni bir tekelleşme mi izleyecek?

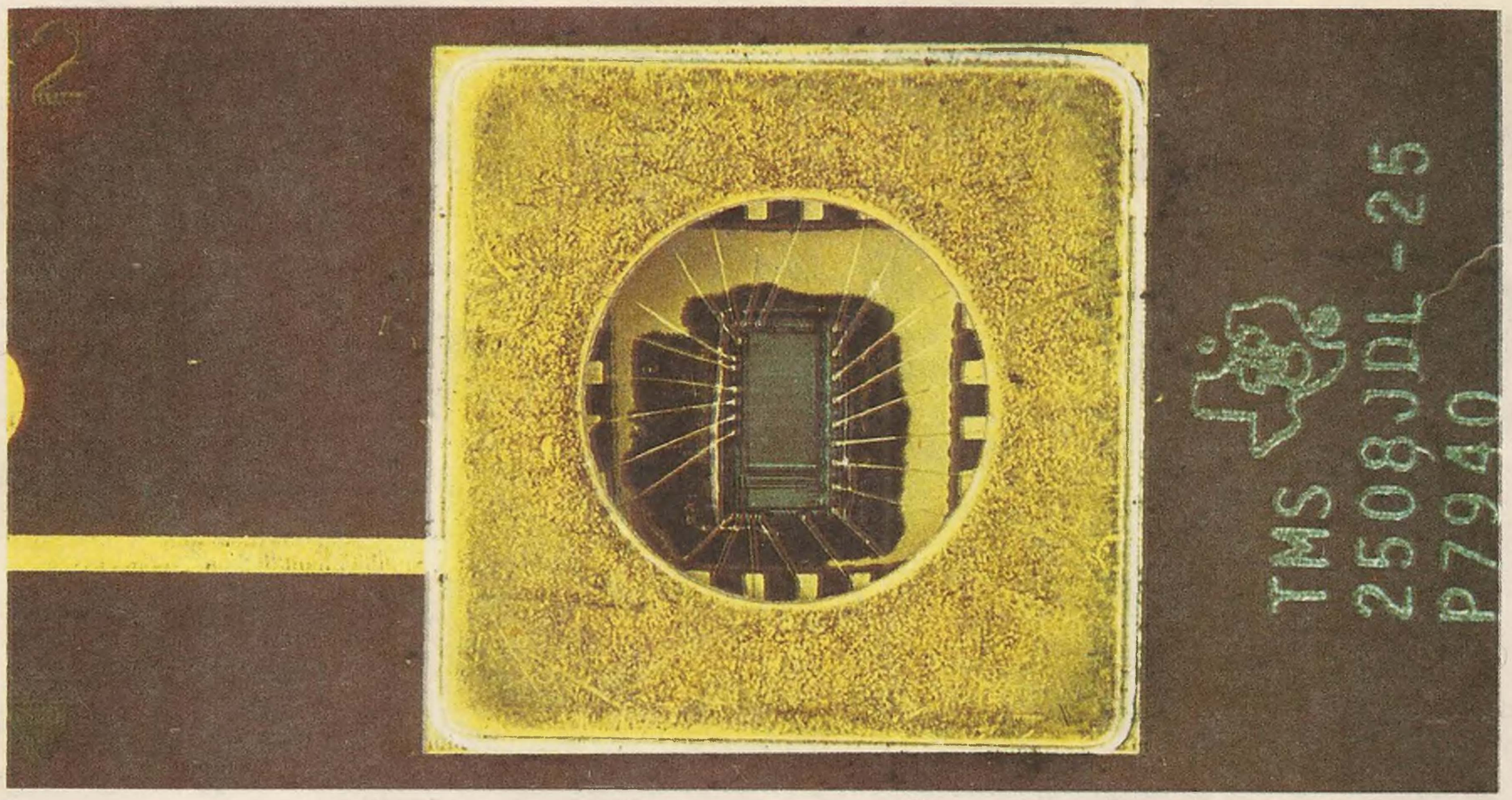
Bugün için bu soruların cevabını vermek güç. Tarihsel örneklerden hareket edersek, bundan sonra da iletişim ağlarının devletten farklı olmakla birlikte, yine de merkeziyetçi ve tekelci niteliğini koruyan birtakım kuruluşların denetiminde olacağı sonucu kaçınılmaz görünüyor. Bu görüşü desteklemek için, Ford firmasının merkezi iletişim ağı, ya da IBM gibi kuruluşların piyasadaki egemenliği gibi örnekler ileri sürülebilir.

Diğer yandan tarihsel örneklerin yanıltıcı olduğu, geçmişteki sanayi devrimlerini mümkün kılmış olan otomotiv endüstrisine ya da buharlı makinelere kıyasla bilgisayar teknolojisinin çok daha esnek bir teknoloji olduğu iddia edilebilir. Başka bir deyişle bilgisayarın, dev bir kurumlaşma olmadan da çok farklı amaçlar için kullanılabileceği, doğası gereği tekelci ve merkeziyetçi kurumlarla bağdaşmayacağı düşünebilir.

Bu gelişmelerden hangisinin gerçekleşeceğini zaman gösterecek.

("TELECOMMUNICATIONS LIBERALIZATION, "Telekomünikasyonda Liberalleşme", Business Week, 24 Ekim 1983 sayısından alınmıştır.)

Kullanılması çok kolay kişisel bilgisayarlar sayesinde, San Fransisko'da yaşayan biri dünyanın öbür ucunda yaşayan biriyle ânında iletişim kurabilir.



Bu silisyum yonga bir elektronik harikasıdır. Resimde ana birimle modüller arasındaki bağlantılar görülüyor.

Daha önce gördüğümüz gibi, iki girişi olan ve girdileri **y** ya da **a** değeri alabilen bir mekanizma için şu seçenekler mevcuttur:

Birinci Girdi	İkinci Girdi
y	**a**
y	**y**
a	**y**
a	**a**

İki giriş birbirine bağlandığı takdirde her iki girdi de aynı değeri almak zorunda kalır; ya ikisi birden **y**, ya ikisi birden **a**.

Seçenekleri yazalım:

A	B
y	**y**
a	**a**

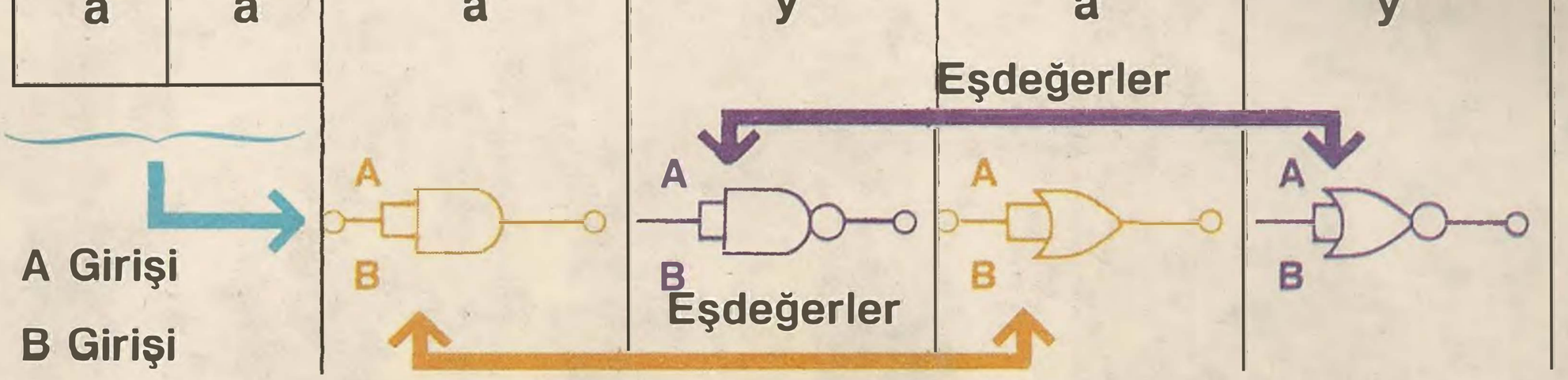
Bu işlemi AND, NAND, OR ve NOR devrelerine uyguladığımızda sayfanın altındaki şekilde resmedilen durumla karşılaşırız. Bu şekle bakınca şu ilişkileri görüyoruz:

AND = OR = zıtlayıcısız güçlendiriciler
NAND = NOR = zıtlayıcı güçlendiriciler

XOR devresinde girdiler bu şekilde bağlanamaz, çünkü:

a (**y** XOR **y** = **a**; **a** XOR **a** = **a**).

A	B	A AND B	A NAND B	A OR B	A NOR B
y	**y**	**y**	**a**	**y**	**a**
a	**a**	**a**	**y**	**a**	**y**



Tümleşik Devreler

Şimdiye dek yalnızca belirli sayıda işlev yapmakla sınırlı devreleri inceledik ve mantıksal açıdan niteliklerini saydık. Ancak bu devrelerin nasıl yapılmış olduklarıyla ilgilenmedik.

Başlangıçta sayısal elektronik devreler, bağımsız olarak üretilmiş ögelerin birleştirilmesinden oluşuyordu. Yani, tek tek ögeler üretiliyor, daha sonra bunlar devreyi oluşturacak biçimde birbirine bağlanıyordu. Yöntemlerin geliştirilmesiyle bir devrenin tüm ögelerini tek yongada birleştirmek mümkün oldu. Artık tek bir birleşen eskiden bir yığın parçanın yaptığı işlevleri yerine getirmeye yetmekte. Bu yeni devrelere fiziksel yapılarından dolayı TÜMLEŞİK DEVRE adı verildi. Tümleşik devrelerin kullanılması çok basittir, giriş ve çıkışları dışında yalnızca besleme gerilimi gereklidir. Gerilim değeri tipe göre değişir.

Fiziksel yapılarından ötürü TTL diye anılan en yaygın tümleşik devreler yalnızca 5 volt gerektirdiklerinden bir kuru pille çalışabilirler. Genellikle bir kılıfta birden fazla devre bulunur. En yaygın tümleşik devrelerden biri VE işlevini gören AND devreleridir. Genellikle dört ya da daha fazla AND devresi birarada bulunur. Bu tümleşik devrenin 14 bağlantı ucu vardır: Her AND devresi için iki giriş ve bir çıkış, iki uç da besleme gerilimi için. Tümleşik devre üreticileri sattıkları her tür için bir de bağlantı şeması verirler. Bu sayfanın sol kenarında 4 tane AND devresi içeren bir tümleşik devrenin bağlantı şemasını görüyoruz.

Gerçi bu devreler küçük olmasına küçüktür ama bilgisayarlar bunlarla yapılsaydı dev boyutlara ulaşılırdı. Günümüz teknolojisi bunların milyonda biri boyutlarında devreler üretebiliyor. Bilgisayarın en önemli işlevlerini gören devreler tek bir yongaya sığdırılabiliyor. Minikişlemci denilen bu devreler bilgisayarın kalbini oluşturuyor.

Birleşik Devreler

Tümleşik devreler aralarında her türlü işlevi yerine getirecek şekilde birbirine bağlanabilirler. Bağlanan devreler aynı gruptan ise bağlantı yaparken özel bir önlem alınması gerekmez.

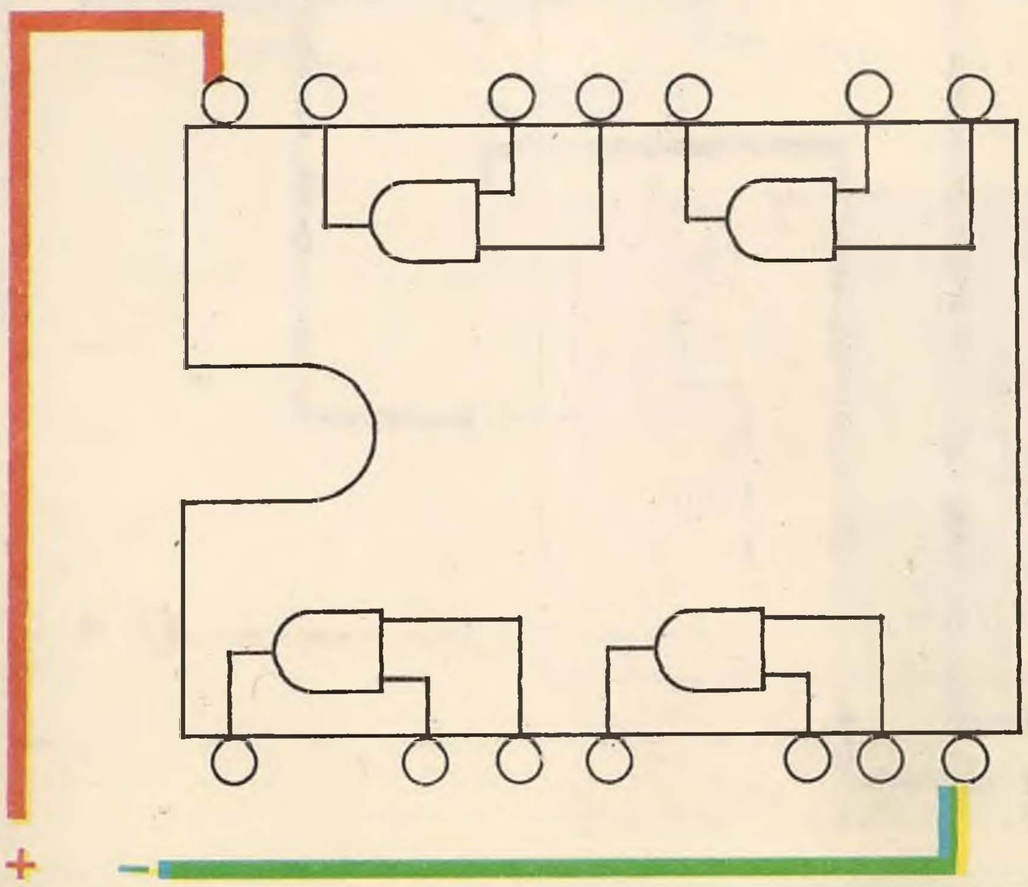
Örnek olarak sayfa 98'de üstteki şekilde AND tümleşik devrelerinden oluşan VEYA işlevi gören bir devre görüyoruz.

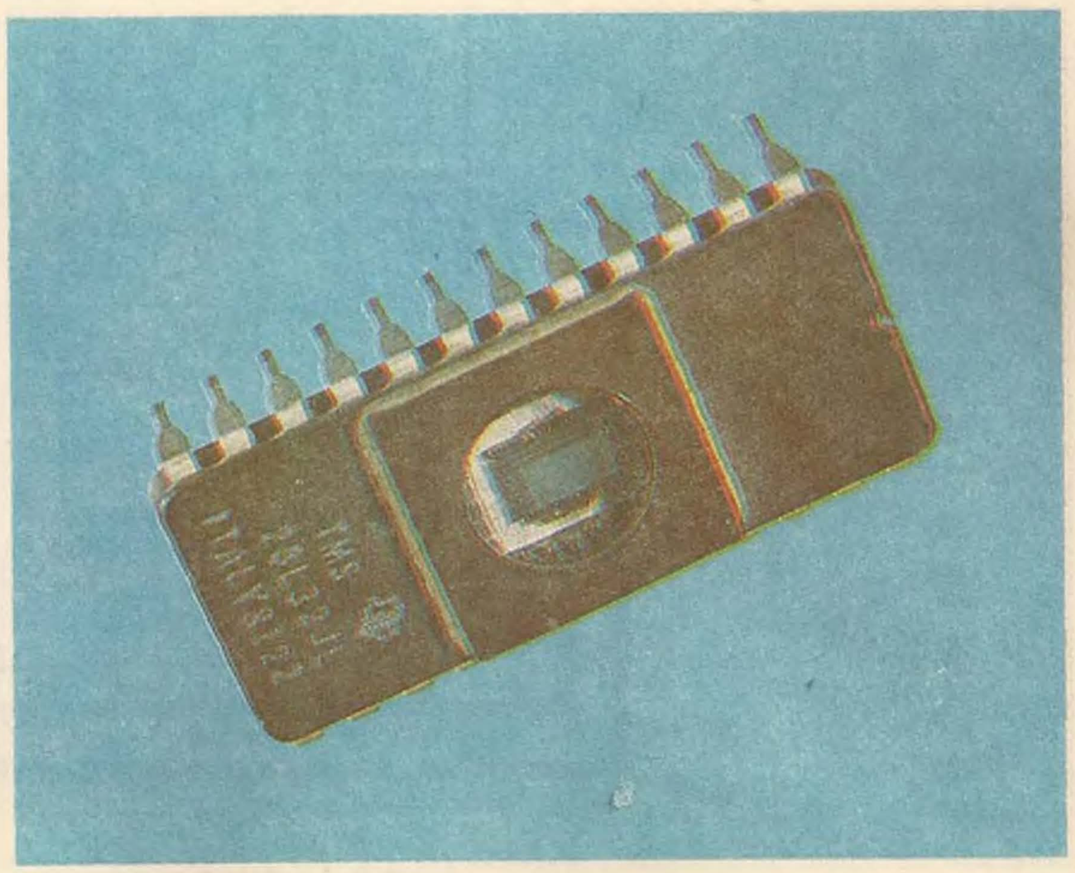
1 ve 2 numaralı devrelerin birer girişinin (B1 ve A2) **y** düzeyinde olduğunu dikkate alarak devrenin bütününün işleyişini, doğruluk değerleri tablosuyla inceleyebiliriz.

Bu devrelerde, **y** düzeyinin besleme geriliminin artı ucu ve **a** düzeyinin de eksi ucu olabileceğini unutmayalım. Şemamızdaki B1 ve A2 girişleri bu nedenle artı uçla bağlanmıştır. Çıkışa bir NAND devresi daha eklendiğinde NOR işlevi elde edilir (Bkz. sayfa 98, ortadaki şekil). OR ve NOR işlevlerinin bağlantı şeması sayfa 98'de alttaki şemada gösterilmiştir.

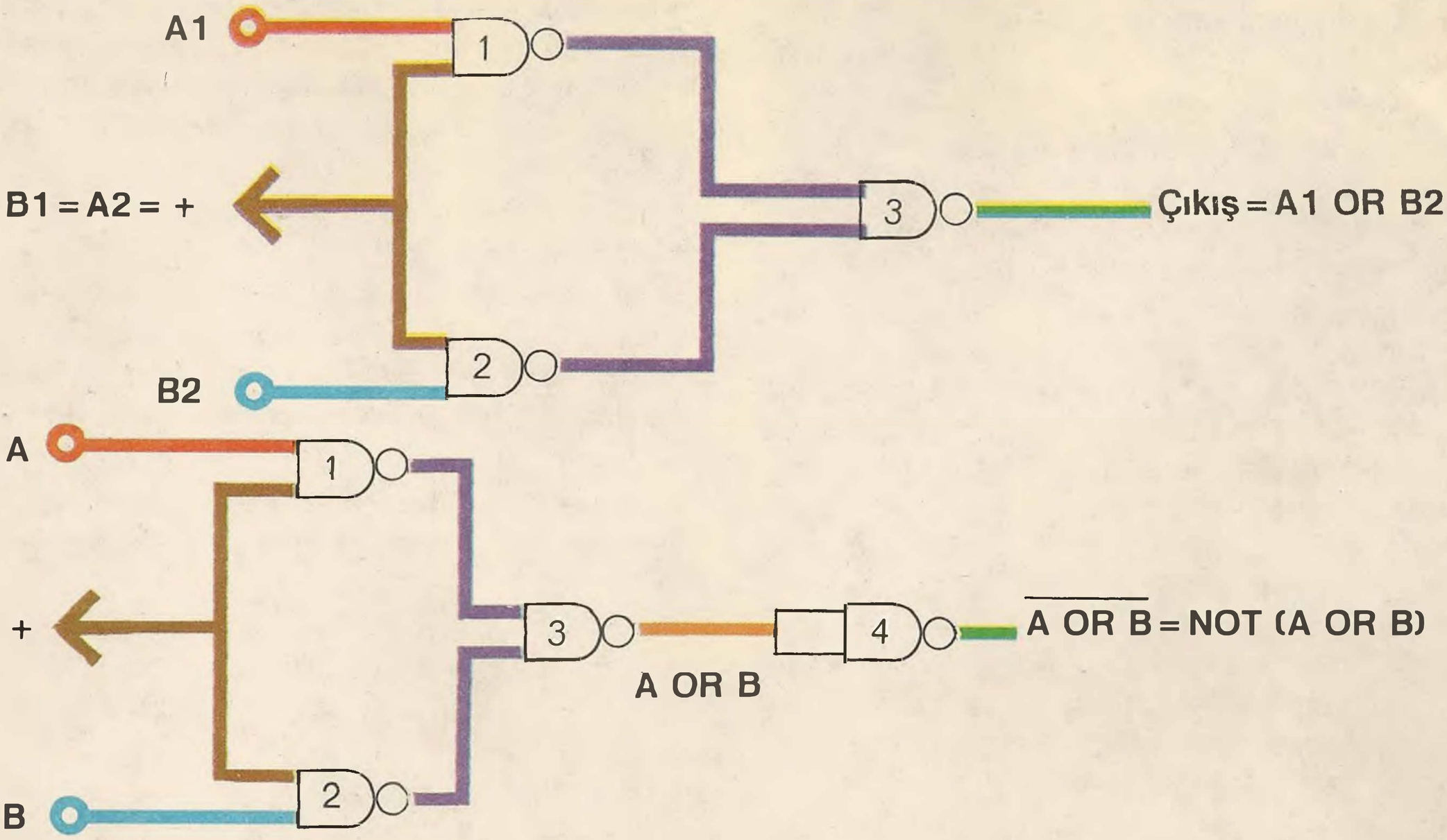
Uygulama Alanları

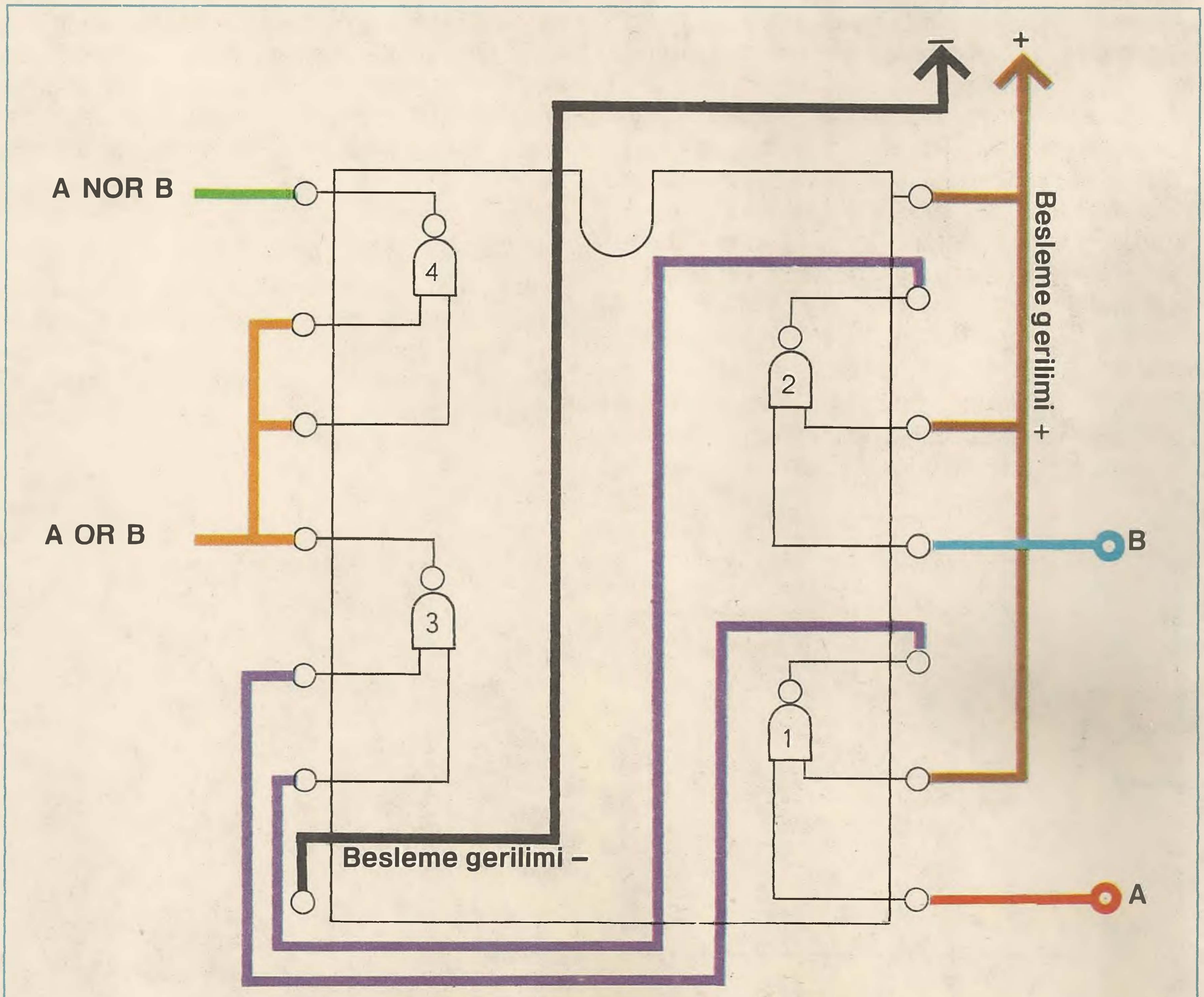
Daha önce gördüğümüz gibi bu devreler bilgisayarlarda ayırıcı işlevi görüyorlar. Şimdi iki uygulama örneğini ele alalım. Birincisi veri taşımada güvenlikle ilgili. Bilgisayar ağlarında büyük uzaklıkları kateden bağlantılar kurula-





Tümleşik devre

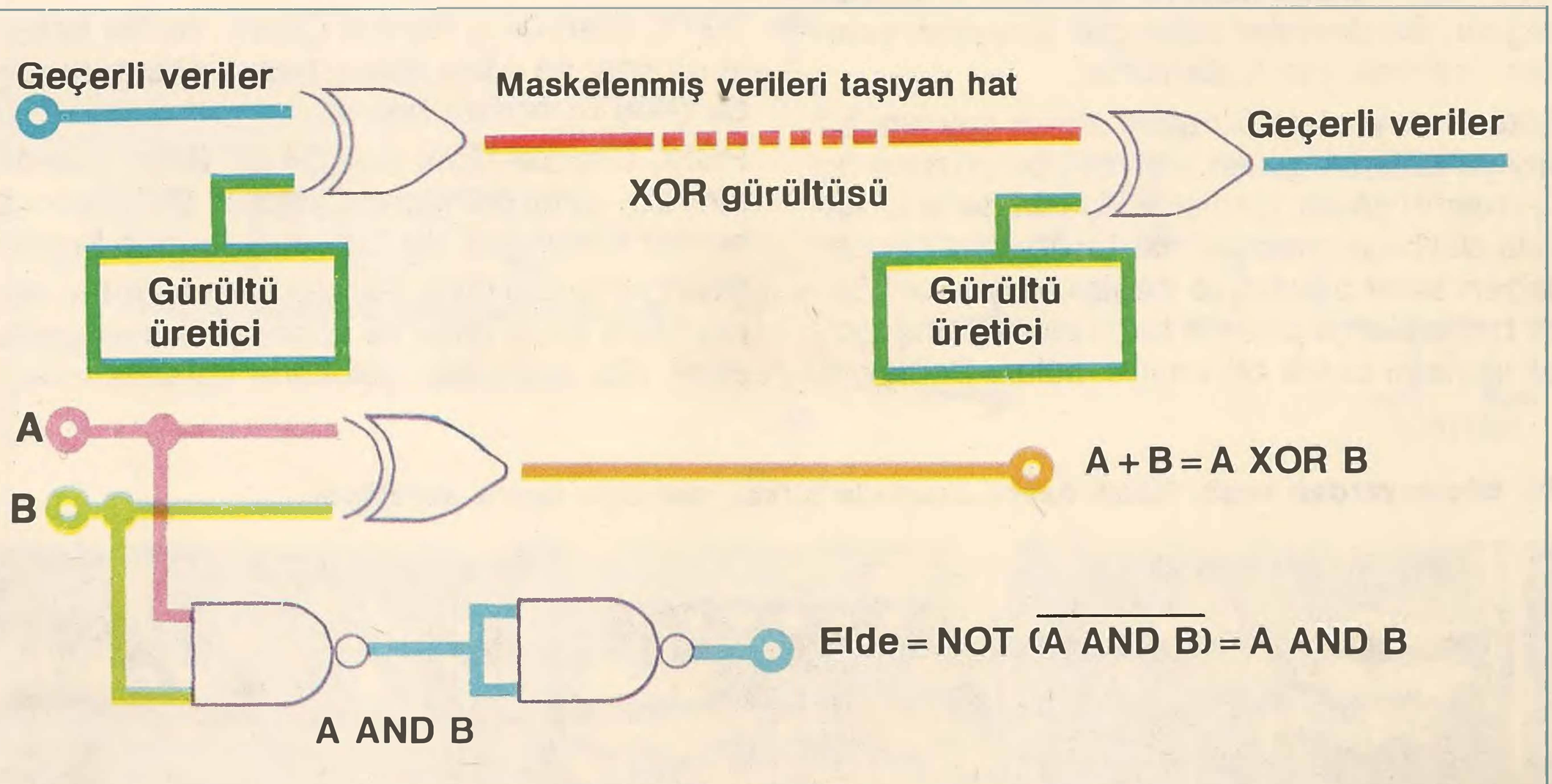

A1
B1 = A2 = +
B2
1
2
3
Çıkış = A1 OR B2
A
+
B
1
2
3
4
A OR B
A OR B = NOT (A OR B)


A NOR B
A OR B
Besleme gerilimi –
Besleme gerilimi +
B
A
1
2
3
4

bilir. Ancak bu şekilde gönderilen veriler kolaylıkla yabancı ellere düşebilir veya değiştirilebilir. Bu nedenle koruyucu önlemler almak gerekir. Örneğin güvenlik sağlayan basit bir koruyucu, şu yöntemi kullanır: Taşıma hattında verilere rasgele darbelerden oluşan bir gürültü karıştırılır ve böylece veriler tanınmaz hale getirilir. Varış yerinde aynı gürültü bir kez daha üretilir ve bu sayede asıl veri elde edilir. Doğaldır ki bunun için aynı gürültüyü iki kez üretebilecek bir üreticiye gerek vardır. Geçerli veriler ancak böyle elde edilebilir. Yöntem basit bir XOR (DARVEYA) devresiyle işaretleri başlangıç noktasında karıştırmak ve varış noktasında ayırmaktan oluşur. Devrenin şemasını aşağıda görüyoruz. Başlangıçta veri girişinin **y** düzeyinde olduğunu varsayalım. Gürültü olarak yine **y** düzeyinde bir işaret karıştırırsak çıkışta **a** elde ederiz (**y** XOR **y** = **a**). Varışta yine **y** düzeyinde gürültü işaretini vermezsek bunun gerçekten bir **a** mı, yoksa gizlenmiş bir **y** mi olduğunu ayırdedemeyiz. Sonuç şöyle bulunur: **y** (gürültü) XOR **a** (varan veri) = **y** (çıkış veri değeri). Bu devrelerin ikinci bir işlevi, iki tabanlı toplama makinelerinde kullanılmalarıdır. Şeklin ikinci bölümünde buna ilişkin şemayı ve doğruluk değerleri tablosunu görüyoruz. Örnek olarak en yaygın kullanılan NAND (değillenmiş ve) devrelerini aldık. Şekildeki ikinci devre zıtlayıcı işlevi görüyor. AND devresi kullansaydık ikinciye gerek kalmaz, birincinin çıkışında A AND B elde ederdik.

Diğer Tümleşik Devre Türleri

Bir kişisel bilgisayar sisteminin üretimi için gerekli tümleşik devreler şimdiye kadar gördüğümüz temel tiplerden ibaret değildir. Basit ya da karmaşık işlevler gören, özel amaçlara hizmet edebilen farklı birçok tür daha vardır. Boole işlevlerini yapan devreler (AND, OR, XOR, vb.), en basit ve kullanım alanı en yaygın olanlarıdır. Herhangi bir devrenin işlevleri



A	B	Toplam	Elde
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

arttıkça karmaşıklığı da artar. Tek bir tümleşik devrede binlerce bileşen içeren minik-işlemci karmaşıklıkta varılan son adımdır. Genel amaçlı diğer bazı karmaşık devreler aşağıda sıralanmıştır.

Zaman Devresi (Timer): Zaman içinde tekrarlanan periyodik sinyalleri elde etmek için kullanılırlar. Genel olarak darbe üretirler. İki sınıfa ayrılırlar: RC ve kuvars tabanlılar. RC devrelerinde darbe süresi bir direnç ve bir kondansatör tarafından belirlenir. Bunlar ucuzdur, ancak darbe süresi dakik bir şekilde ayarlanamaz. Kuvars devrelerinde ise mekanik titreşimler yapan bir kuvars kristali darbe süresini belirler. Bu devreler daha pahalı, ancak daha dakiktirler. Kişisel bilgisayar sistemlerinde sinyal elde etmek için bu tür devreler kullanılır. Bu sistemlerde frekans 2-8 MHz (megahertz) arasındadır, saniyede iki ilâ sekiz milyon titreşime denk düşer.

Tetik: Giriş sinyalinin düzeyine tepki gösterirler. Bu düzey belirli bir değeri aştığında tetikleme yaparlar. Tetikleme anında çıkış durumu Alçak'tan Yüksek düzeye (ya da tersi yönde) değişir. Bu devreler daha çok sinyalleri yeniden üretmek için kullanılırlar.

Bölücü devreler: Girişteki darbe sayısını bir sayıya bölerek çıkışa verirler, bu yüzden bu devrelerin çıktısı, girdisine kıyasla daha az sayıda darbeden oluşur. İki türü vardır: Bölen değeri sabit olanlar ve değişken olanlar. Sabit bölücüler yapılarına bağlı olarak giriş darbe sayısını belirli bir sayıya böler. Değişken bölücülerin bölen sayısı istendiğinde değiştirilebilir.

İkili devreler: Bu devre sinyalin hem VAR olduğu, hem YOK olduğu durumları tanır. Çıktı, girdiyi hemen izlediğinden girdiden ayırdedilmez. İkili devre, durum değişikliklerini girişte kaydeder ve bir zamanlama sinyali gelinceye kadar izlemeyi sürdürür. İlk girdi sinyali ile zamanlama sinyali arasındaki sinyal VAR ve sinyal YOK durumlarını kaydettiğinden, bu devre çift kararlı bir devredir.

Kaydırmalı devreler: Birbiriyle bağlanmış belirli sayıda ikiliden oluşur. Her zamanlama sinyali geldiğinde, veri değerlerinin bir hücreden diğerine kaydırıldığı bir ''hücreler dizisi'' olarak betimlenebilir. Başlangıçta, ilk veri girişi birinci hücreye yapılır. Her zamanlama sinyalinin gelişiyle, veri, yandaki hücreye aktarılır.

Bu devreler işleyiş biçimine göre çeşitli türlere ayrılır. Başlıcaları şunlardır:

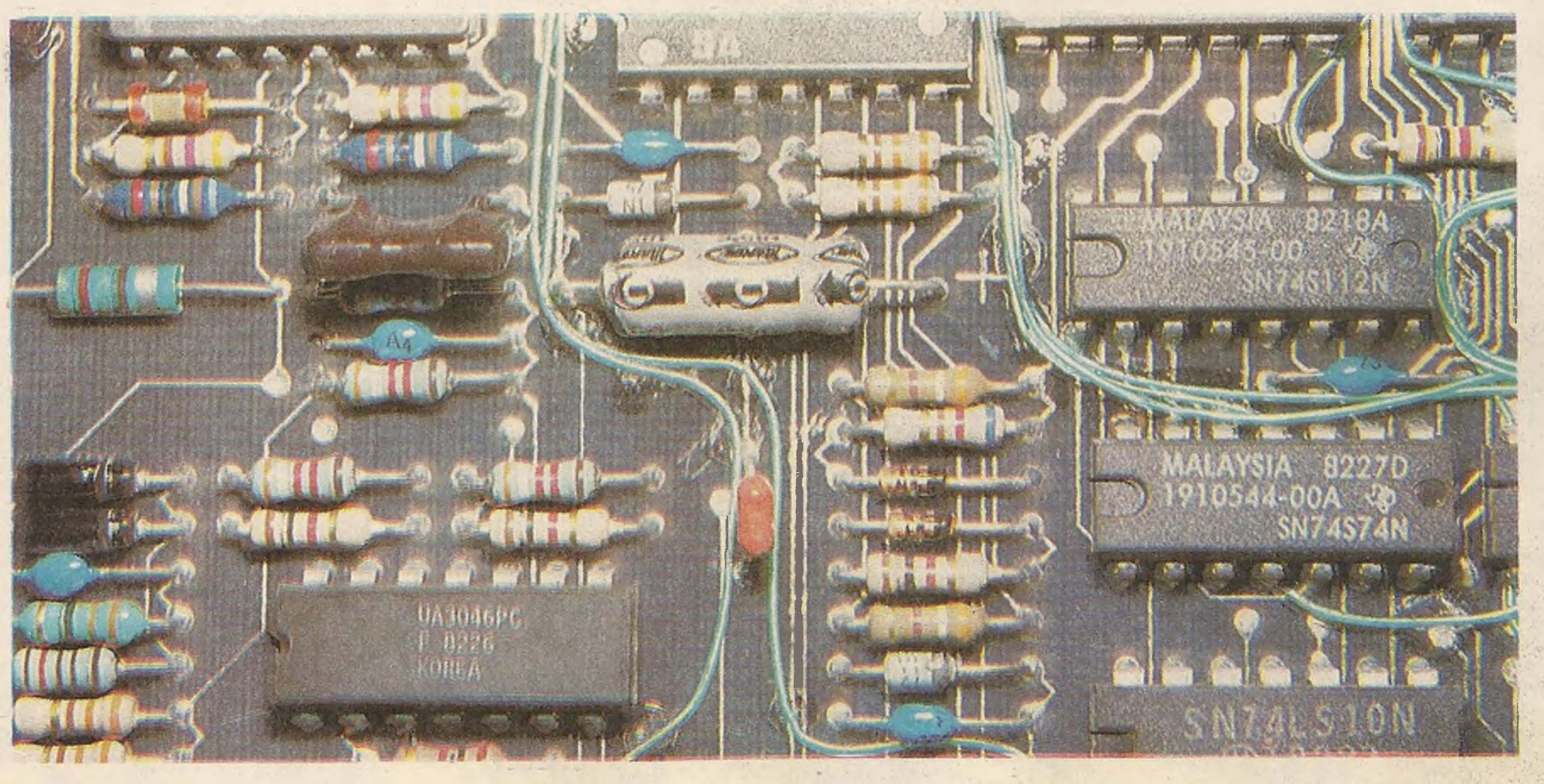
SGSÇ (Seri Giriş Seri Çıkış): Veriler sırayla ve tek tek girerler ve aynı sırayı izleyerek çıkarlar.

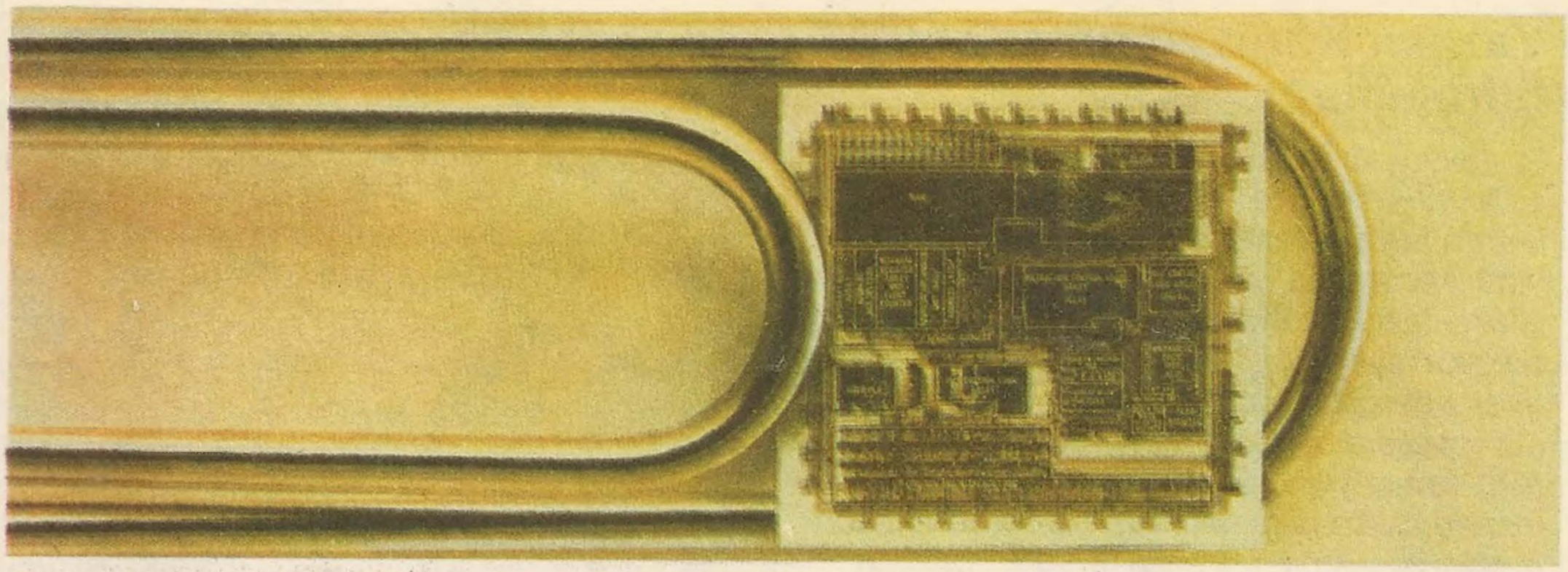
SGPÇ (Seri Giriş Paralel Çıkış): Veriler sırayla girerler ve daha sonra hepsi yine aynı anda çıkış uçlarında belirir.

PGSÇ (Paralel Giriş Seri Çıkış): Belirli sayıda sinyalin girişi aynı anda yapılır. Daha sonra bunlar sırayla tek tek bir çıkış ucunda belirir.

PGPÇ (Paralel Giriş Paralel Çıkış): Bütün veriler aynı anda girer ve sonra yine aynı anda çıkar. (Bu sayfadaki şekillerle karşılaştırınız.)

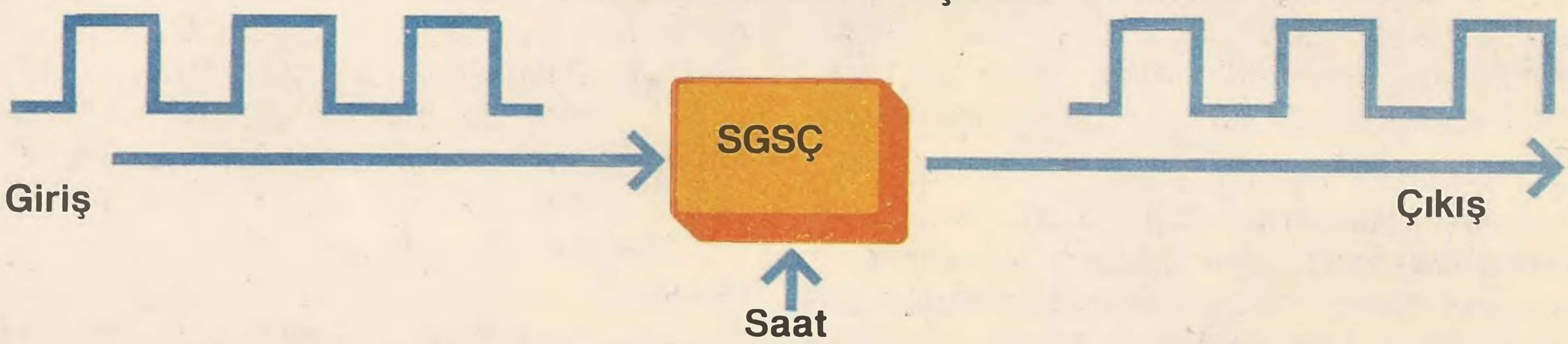
Bir bilgisayardan kesit. Basılı devre üzerinde birkaç tümleşik devre görülüyor.

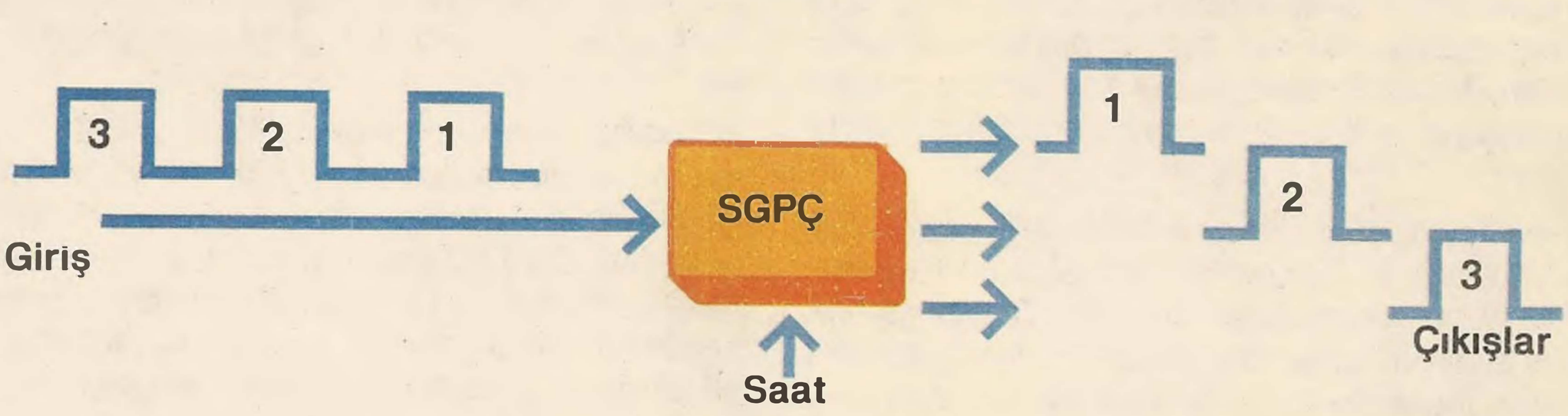


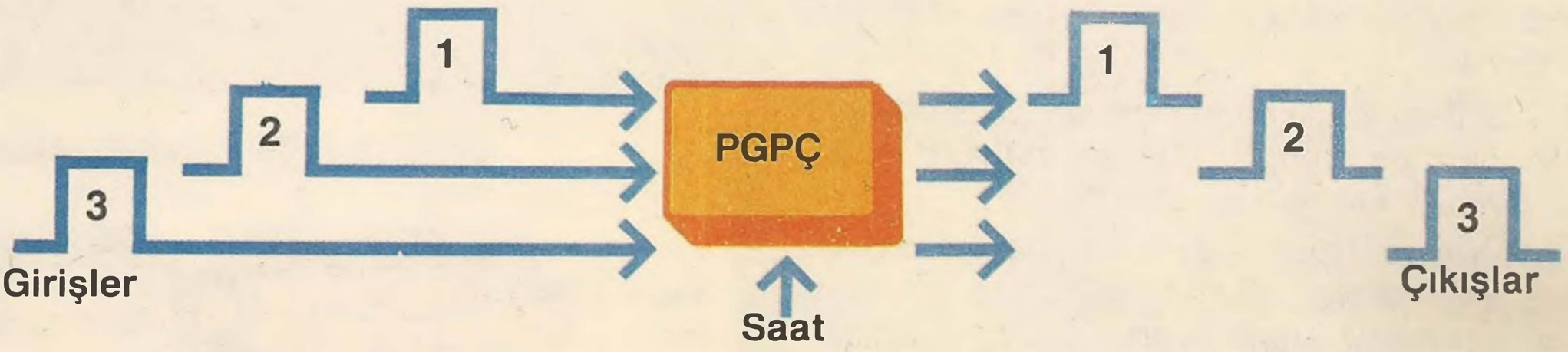


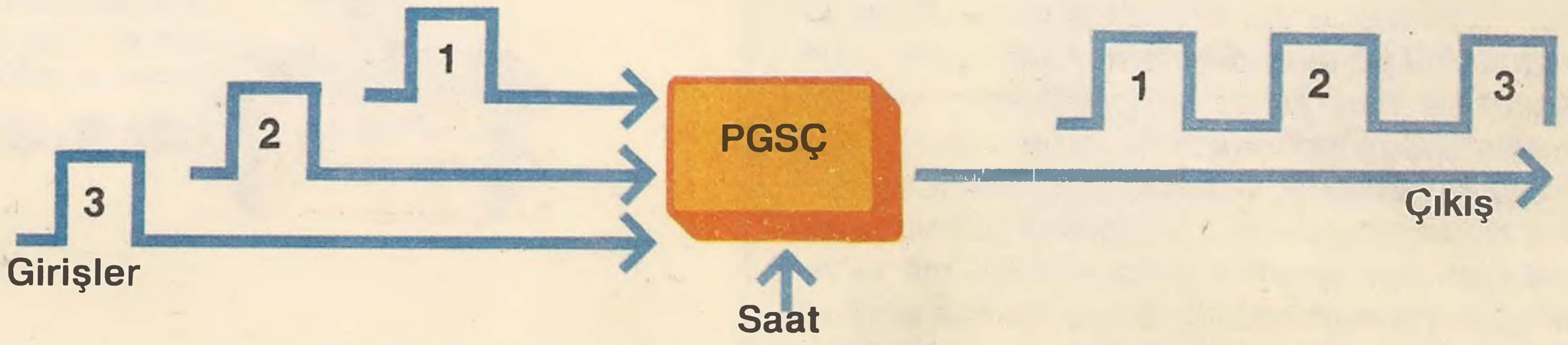
Üzerinde çeşitli bileşenlerinin gösterildiği bir minik-işlemci

KAYDIRMALI SİCİLLERİN BAŞLICA TÜRLERİ









Yazılım Korsanlarına Karşı Şifreleme

Telematik sayesinde hızlı veri taşıma yöntemlerinin yaygınlaşması, herkes için büyük zaman kazancına yol açtı. Binlerce kilometre uzaklıktaki kişiler ve bürolar artık, bilgi, veri ve belge alışverişinde bulunabiliyorlar. Her gün telematik ağlarına, gizli banka hesapları, endüstriyel sırlar gibi gizli bilgiler yükleniyor. İletilen veri sayısı her gün artıyor. Güvenlik önlemleri alınmadığı takdirde bu verilerin çalınmaları ve çarpıtılmaları çok kolay olabilir.

Son zamanlarda dünya basınını en çok ilgilendiren konulardan biri ''yazılım korsanlığı'' *oldu. Bu yolla kimi kurnazlar, büyük uluslararası bankaların müşteri hesaplarına yapılmış ya da yapılacak ödemelerle ilgili günlük haberlerin iletildiği telematik hatları ile korsan bağlantı kurarak büyük kârlar elde etmeyi başardılar.*

Anlaşılır nedenlerden ötürü bankalar bu türden haberlerin fazla yayılmasını engellemeye çalışıyor. Böyle bir haber kampanyasından pek avantajlı çıkmayacakları kesin. Ama bütün çabalarına karşın, bu gibi haberler yine de birkaç kez, günlük gazetelerin birinci sayfalarında yer aldı ve kamuoyunda endişe yarattı.

Ancak günümüz uzmanlarını sürekli olarak yeni ''kamuflaj'' *yöntemleri aramaya iten tek neden yazılım korsanlığı değildir. Dünyada her gün telefon hatlarıyla iletilen gizli bilgiler sayısız konuları kapsıyor: Patent başvuruları, endüstride çeşitli üretim süreçleri, uluslararası siyasal örgütlerin birbirlerine yolladıkları gizli iletiler vb.*

Bu yüzden, bilgisayarın sunduğu olanaklardan yararlanırken, yeni şifreleme yöntemleri de bulmak kaçınılmaz olmuştur.

Şifreleme asıl metnin harf harf başka simgelere çevrilmesidir. Bu çevirinin başlıca amacı çevrilen metni istenmeyen kişilerce tanınmayacak hale getirmektir.

Bu, ya harflerin sırası değiştirilerek ***(konum değişikliği yoluyla şifreleme)****, ya da her harf başka bir harf ya da simgeyle karşılanarak* ***(değiş tokuş yoluyla şifreleme)*** *yapılır. Bu yöntemlerin her ikisi birden aynı metne tekrar tekrar uygulanabilir. Örneğin oldukça uzun metinlerde, kaydırma yoluyla şifreleme etkili bir gizleme yolu olabilir. Bu yöntemde harfler birden çok kez karıştırılır* ***(çift veya üç kat kaydırmalı şifreleme)****. Kaydırma yoluyla şifrelemeye basit bir örnek verelim:*

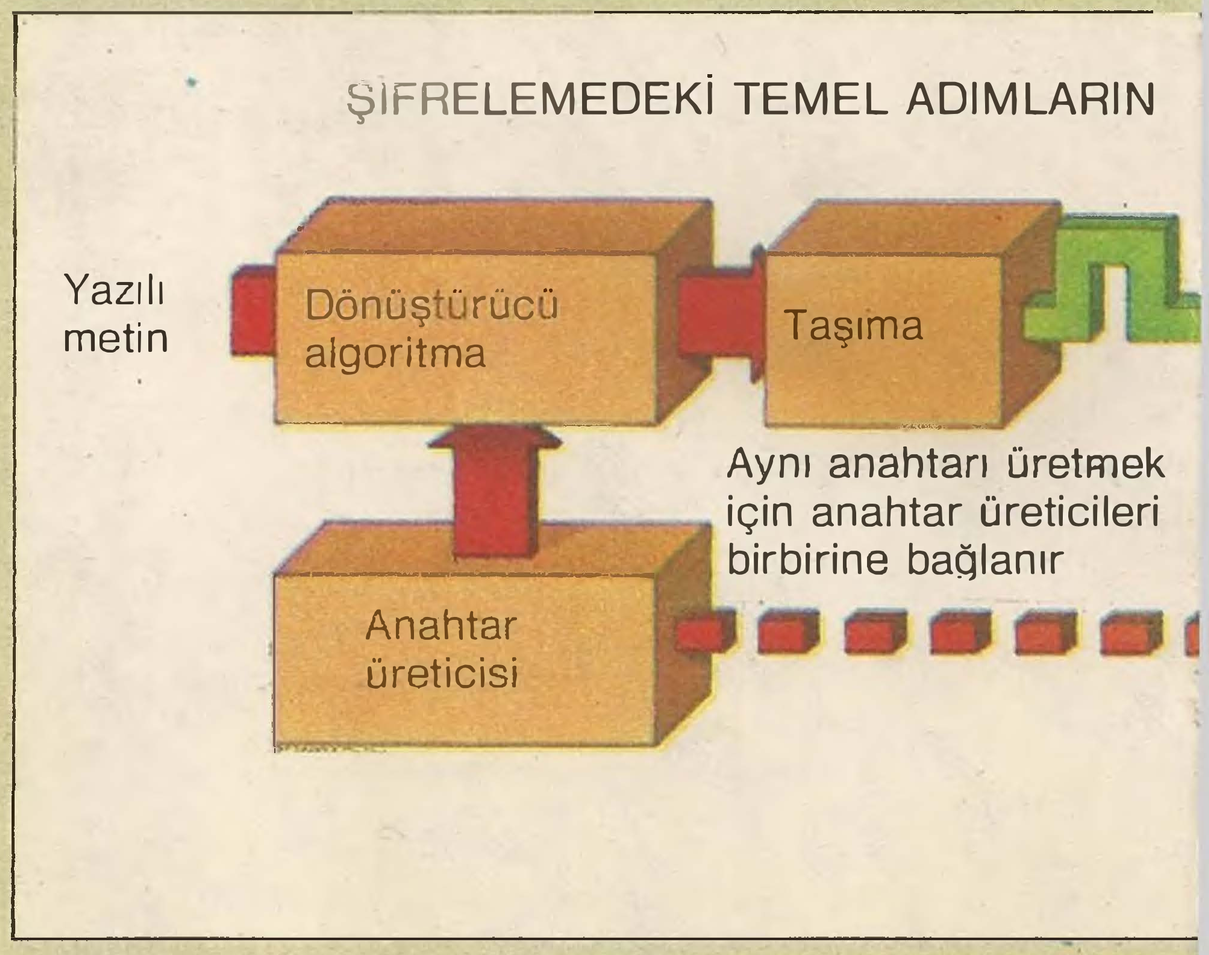
M A L I H
E M E N G
Ö N D E R
İ N

Metni yatay yazarsak MALI HEMEN GÖNDERİN olur. Bu ileti dikey okunarak şifreli yazı haline getirilebilir: MEÖİ AMNN LED INE HGR. Ancak eğitilmiş bir göz önce birinci harfleri, sonra ikinci harfleri okuyarak, bu yöntemle şifrelenmiş bir metnin anlamını kısa sürede çıkarabilecektir. Bu nedenle harflerin daha karmaşık bir biçimde kaydırılmaları gerekir.

Yukarıda anlattığımıza benzer diğer bir basit kaydırmalı şifreleme yöntemi de her harfin yerine, alfabetik sırada o harfin yanında yer alan harfin konmasıdır. A'nın yerine B, B'nin yerine C vb. Dolayısıyla ''*MALI HEMEN GÖNDERİN*''*, iletisi bu şifrelemede NBMİ IFNFO ĞPOEFSJO*'' *biçimine dönüşür. Bu biçim ilk bakışta yeterince tanınmaz gibi görünüyorsa da, alfabeyi bilen biri için şifreyi çözmek son derece kolaydır.*

Bu tür şifrelemede her harfin yerine başka bir harf, ama hep aynı harf geçtiği için bu şifreler ''düz şifre'' *diye adlandırılırlar.*

Bir harfin yerine bir sonrakini koymaktansa, kaydırma miktarı her harf için, bir harf artırılabilir. Bu durumda ''*MALI*'' *kelimesi* ''*NCOL*'' *olacaktır. Bu şifreleme sisteminde, asıl metindeki belli bir harf yerine, şifrelenmiş metinde alfabenin herhangi bir harfi geçebilir. Böyle şifrelenmiş bir metnin çözümü biraz karmaşık bir matematiksel çözümleme gerektirir.*



Bunlardan başka, iki adım gerektiren daha karmaşık şifreleme yolları da vardır. Birinci adımda harfler önceden belirlenmiş bir şemaya göre kaydırılır; bu aşamada kaydırmanın ters yönde yapılması asıl metni elde etmeye yetecektir. Bilgisayar terminolojisinde bu önceden belirlenmiş şemaya ''algoritma'' denir. İkinci adımda ise şifrenin gönderileceği kişiye algoritmayı iletiye nasıl uygulayacağını gösterecek bir anahtar bulmak gerekir. Kaydırmalı şifrelemenin güvenlik derecesi büyük ölçüde şifre anahtarının uzunluğuna bağlıdır. Kuşkusuz en güvenilir şifre anahtarı, yalnızca karmaşık bir harfler, 2 ve 10 tabanlı sayılar, simgeler dizisi kullanmakla kalmaz, aynı zamanda sonsuz uzunlukta da olurdu. Çünkü anahtar sınırlı bir uzunlukta olduğu sürece keşfedilmesi ve dolayısıyla şifrenin de çözülebilmesi olasılığı hep varolacaktır. Yine de şifre asıl metnin elde edilmesini yeterince uzatıyor ya da pahalılaştırıyorsa, amaca ulaşılmış demektir.

Çoklu şifre sistemlerinin en büyük güvencesi şifrelemede tüm olası harflerin kullanılabilmesidir. Yani metnin değişik yerlerinde görülen A harfi için her defasında farklı bir karakter kod olarak kullanıldığı halde, anahtar sayesinde A harfine ulaşmak, sonunda yine mümkün olur.

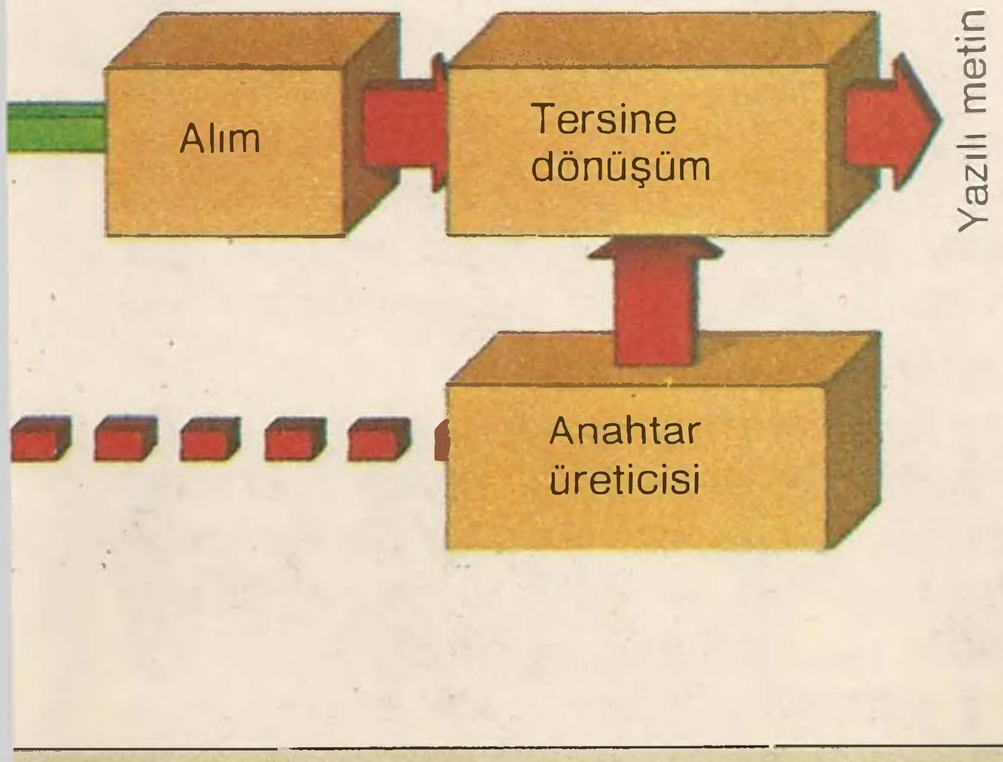
Başka bir şifreleme türü ise her harfin alfabenin 29 harfi tarafından rasgele karşılanmasıdır. Bunun için raslantı ilkesine göre oluşmuş sınırlı uzunlukta bir anahtar gereklidir. Yöntem raslansal diziye dayanır. Bu raslansal dizi, harflerden, 10 tabanlı sayılardan ya da 2 tabanlı sayılardan oluşabilir ve şifre çözümü sırasında her harfin ne kadar kaydırılması gerektiğini belirtir. Bu sistem son derece güvenilir bir şifreleme sistemidir. Kodu çözebilmek için şifre çözücünün olağanüstü bir çaba harcaması gerekir. Ancak aynı şey alıcı için de geçerli olduğundan bu sistem zor ve pahalıdır. Çünkü, haftalar ya da aylar boyunca gönderilecek her metnin her harfi için bir anahtar yapılması ve dağıtılması gerekir. Uygulamada çoklu şifrelerin çoğu Beaufort'un evrik karelerine dayanan sınırlı uzunlukta anahtarlar kullanır. Böyle bir durumda anahtar kısa bir cümle olabileceği gibi iletilerin kendileri de ek anahtarlar içerebilir.

Ancak çoklu şifre anahtarlarının çözülebilme olasılığı, günümüzün hızlı bilgisayarları sayesinde artmıştır. Ama çoğunlukla şifreleme sistemlerimizin tamamiyle güvenilir olmasını bekleyemeyiz. Daha önce de söylediğimiz gibi, şifre asıl metnin elde edilmesini yeterince uzatıyor ya da pahalılaştırıyorsa amaca ulaşılmış demektir.

Son on yılda şifreleme alanında, özellikle bilgisayarın kullandığı sayısal yöntemlerle doğrudan ilişkili çok hızlı birtakım sayısal yöntemler sayesinde dikkate değer yenilikler görüldü. Bilindiği gibi elektronik alanında bir sisteme sayısal denildiği zaman, bu sistemde, sinyallerin ve sinyal biçimlerinin sınırlı sayıda birtakım durumlara karşılık geldiğini kastediyoruz. Oysa benzeştirmeli sistemlerde sonsuz sayıda basamak ve biçimler olabilir.

2 tabanlı sistem, yalnızca iki farklı durum tanıyan bir sayısal sistemdir. Dolayısıyla bu sistemde sinyal, yalnızca iki boyut üzerinde değerlendirilir: VAR/YOK (yani, devre ''açık'' ya da ''kapalı''). İletişim dilinde söyleyecek olursak, sinyalin değeri, ya 1 olabilir ya da 0. Bir bakıma, radyo veya kabloyla yapılan tüm şifreli iletimler böyle bir sayısal kodlama gerektirir. Örneğin, ister Mors alfabesini kullansın ister harfleri doğrudan doğruya kullansın, bütün telgraf ve teleks sistemleri böyle bir kodlamaya dayanır. Önce hazır metin için sayısal bir anahtar hazırlanır. Bu sayısal anahtar darbe modülasyonlarına uygulanır. İkinci adımda, yani alıcı taraf şifreyi çözmeye çalışırken, anahtar tarafından belirlenmiş bitlerin akışı ile ilk başta kullanılmış sayılar arasındaki ilişki kurulur ve şifre çözülür. Günümüzün elektronik aygıtları sayesinde, ister yazılı bir metin olsun ister bir konuşma, bütün iletilere saydığımız bu işlemlerin uygulanması gayet hızlı bir biçimde yapılabilmektedir. Anahtar gerçekten raslantı ilkesine dayanıyorsa ve sınırlı uzunluktaysa bu sistemin çok güvenli olması gerekir.

BASİTLEŞTİRİLMİŞ BİR ÇİZİMİ



Bu yöntemin de ''raslantı anahtarı''nınkine benzer güçlükleri vardır. Çünkü uygulamada genellikle kullanılan anahtar tam anlamıyla raslansal değildir. Bu sanki-raslansal (pseudo-random) dizi, nispeten daha kısa (genellikle 100 bitin altında) ve gerçekten raslansal olan bir sıfırlar ve birler dizisinin tekrarından oluşur. Sanki-raslansal dizinin nasıl elde edildiğinin şifre sisteminin güvenilirliği üzerinde büyük etkisi vardır. Bugün ''doğrusal mantığa'' dayanan ilk sayısal sistemlerin ürettiği şifrelerin küçük bir bilgisayar tarafından çabucak çözülebileceğini biliyoruz. Bunun için haberin sadece bir kısmının (örneğin adresin) çözümleyici tarafından biliniyor olması yeterlidir.

Bu tehlike bazı kuruluşları, örneğin IBM'i az sayıda anahtar biti kullanan ama doğrusal olmayan bir algoritmayla çalışan şifreleme sistemleri bulmaya yöneltti. Bu sistemlerde her metin bölümü ve söz konusu bölümün anahtarı, bir dizi çok karmaşık değişikliğe uğratılmakta, harfler hem kaydırılmakta, hem de uzun anahtar dizileri aracılığıyla kodlanmakta. Bu şifreler, en azimli saldırılar karşısında bile çözülmeye direnebilmekte.

1977'de Amerikan Standartlar Bürosu, 56 bitlik bir anahtara dayanan ''Veri Şifreleme Standard''ı (Data Encryption Standart-DES) algoritmasını onayladı. Her veri öbeği 18 değişik işlemden geçiyor. Bu işlemler sırasında 56 bitlik temel bir anahtardan türetilen 16 değişik şifre anahtarı kullanılıyor ve böylece toplam anahtar uzunluğu 10^{17} bite ulaşıyor. Bu sistemin kullanıcıları bilinen bir algoritmayı kullanıyorlar. Ancak kendi anahtarlarını üreten 56 bitlik (iç kodlamayla birlikte 64 bitlik) temel anahtar gizli kalıyor.

Anahtar üretici, mühürlenmiş bir kutu şeklinde olabilir. Önemli olan onu sürekli olarak kilitli tutmaktır. Bu ise anahtarların dağıtımı ve denetimi sorununu gündeme getiriyor. Şimdilik, bu sistemin normal ekonomik tüm amaçları karşılayacak derecede güvenli olduğuna inanılmakta. DES gibi sayısal sistemler doğrudan bilgisayara bağlanabilmekte, böylelikle ''yazılım korsanlığı'' da önlenmektedir.

Aynı şekilde gizli kütüklere ya da telefon ve radyo yoluyla iletilen verilere izinsiz ulaşım da önlenebilir. Ancak bazı uzmanlar 56 bitlik temel anahtarın fazla kısa olduğunu ve bu sisteme sanki bilgisayarı kaba güçle bir saldırıya karşı bile koruyabilirmişçesine aşırı bir güven duyulduğunu düşünmekteler.

DES şifreli yazı sistemleri, büyük bankalar ve ekonomik kuruluşlar tarafından kullanıldığı ve bugüne kadar bu sistemlerden bilgi çalmanın mümkün olduğu kanıtlanmadığı halde, çok sayıda Amerikalı uzman başka bir sistemi salık veriyorlar. Bu sistemin daha güvenli olduğu, zira anahtarının herkesin kullanımına açık olduğu söyleniyor.

Böylece anahtar dağıtımı ve denetimi sorunu ortadan kalkıyor. Bu sistemlerde şifreleme ve şifre çözme ayrı. Gönderici şifre çözme koduna sahip değil, ancak iletiyi, alıcının da bildiği bir anahtar kod kullanarak şifreliyor. Bu sistemler anahtar üreticilerin dağıtımı ve güven-

Örnek 1

ANAHTAR		A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	L	M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z
KAYDIRMA		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
YAZILI METİN	M	A	L	I	H	E	M	E	N	G	Ö	N	D	E	R	İ	N												
SERBEST ANAHTAR	C	A	F	Z	L	H	T	R	P	A	S	H	R	Y	F	N	O												
KAYDIRMA	3	1	7	29	15	10	24	21	20	1	22	10	21	28	7	17	18												
ŞİFRELİ YAZI	Ö	B	S	I	U	P	I	V	G	B	İ	V	Ü	D	Y	Z	E												

Örnek
Anahtarın F harfi L'yi S ile veya R'yi Y ile değiştirir.
Anahtarın H harfi E'yi P ile veya N'yi V ile değiştirir.

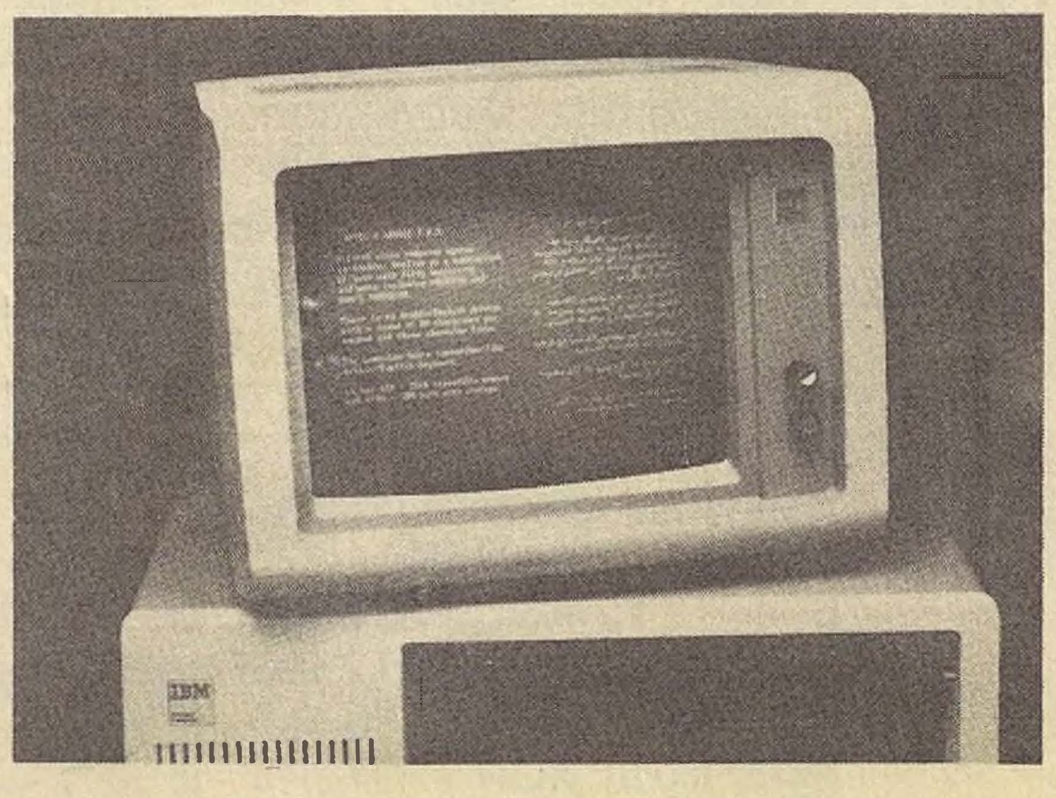
liği sorununu da ortadan kaldırdıkları için yüksek güvenlik taleplerini karşılamaları gerekiyor.

Ancak şifreleme alanındaki ilerlemelerin öneminin yalnızca güvenlikle ilgili olduğunu söylemek yanıltıcı olur. Örneğin belirli bir alfabeyle yazılmış bir yazının farklı bir alfabeye çevrilmesi sorununu ele alalım. Bunu basit bir şifre çözme problemine benzetebiliriz.

Diyelim ki Yunan alfabesiyle yazılmış bir metni Lâtin alfabesiyle yeniden yazmaya çalışıyoruz. Bunun için önce hem Lâtin harfleri hem Yunan harfleri için ortak olan bir kod anahtarı üretmemiz gerekecektir.

Bir kez böyle bir anahtar geliştirildikten sonra, artık göndericinin (örneğin bilgisayarın) yapması gereken Yunanca'daki her harf yerine anahtardaki sinyali yollamak, alıcının (örneğin bilgiyazarın) yapacağı ise gelen sinyale karşılık olarak, bu kez anahtar kodun gösterdiği Lâtin harfi yazmaktan ibaret olacaktır.

*Ancak durum daha da karmaşık olabilir. Örneğin Lâtin alfabesindeki **k** harfine karşılık Arap alfabesinde iki harf, **kaf** ve **kef** bulunmaktadır. Bu yüzden daha da karmaşık bir anahtarlama sistemi gerekecektir. Ancak bu yazıda anlatılan ilkelerle sorunun kolayca üstesinden gelinebilir.*



Örnek 2

A	00001	N	01110
B	00010	O	01111
C	00011	P	10000
D	00100	Q	10001
E	00101	R	10010
F	00110	S	10011
G	00111	T	10100
H	01000	U	10101
I	01001	V	10110
J	01010	W	10111
K	01011	X	11000
L	01100	Y	11001
M	01101	Z	11010

YAZILI METİN	**M**	**A**	**L**	**I**	**H**	**E**	**M**	**E**	**N**
İKİ TABANLI ÇEVİRİ	01101	00001	01100	01001	01000	00101	01101	00101	01110
SAYI ANAHTARI	01100	11001	00110	10101	11101	01011	00010	01010	00101
KODLANMIŞ ÇEVİRİ	00001	11000	01010	11100	10101	01110	01111	01111	01011
SİNYAL BİÇİMİ									

TEST 2

1 / Aşağıdaki doğruluk değerleri tablosunu tamamlayınız:

A	B	$\overline{A \text{ XOR } B}$	A AND ($\overline{A \text{ XOR } B}$)
1	0		
1	1		
0	0		
0	1		

---

2 / Aşağıdaki işlemleri çözünüz:

a - 11011 AND 011
b - 12 AND 7 (12 ile 7 on tabanlı sayılardır)
c - 12 AND 7 (12 ile 7 sekiz tabanlı sayılardır)
d - 1101 OR 0110
e - 1011 XOR 1101

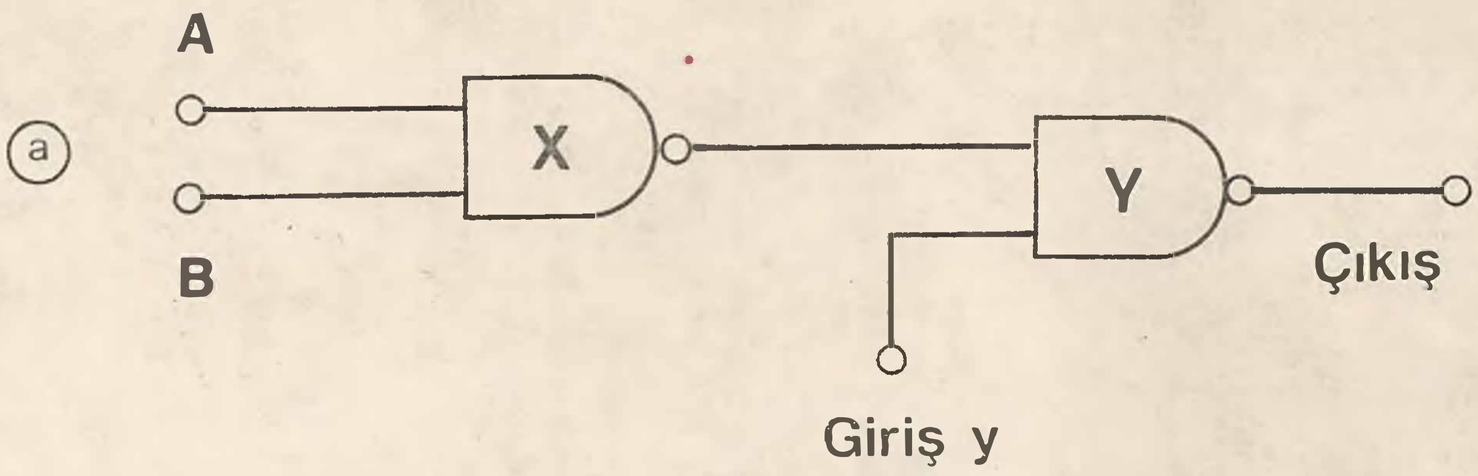
---

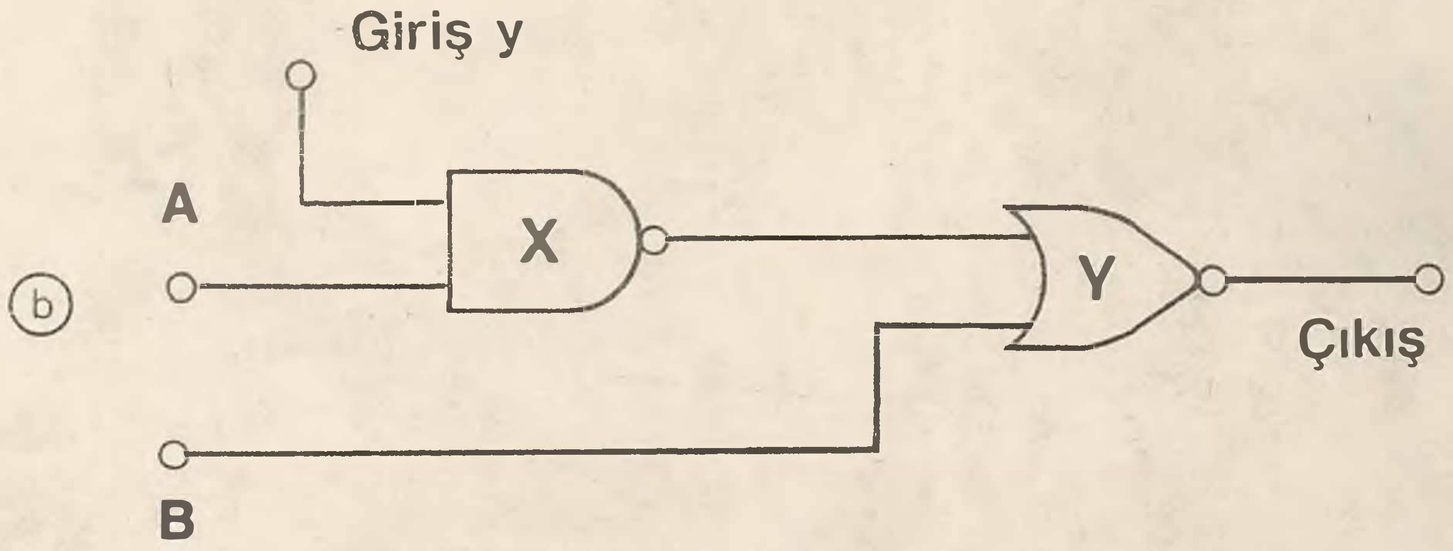
3 / Aşağıdaki devrelerin şemalarını ve doğruluk değerleri tablolarını çiziniz:

a - AND (VE) b - OR (VEYA) c - XOR (DARVEYA) d - NOT (DEĞİL)

---

4 / Aşağıdaki devrelerin doğruluk değerleri tablolarını gösteriniz:





3. ve 4. soruların çözümü, izleyen konuları anlamak için şarttır. 1. ve 2. soruların en az üç şıkkı çözülmüşse, önceki konular anlaşılmış demektir.

Veri Taşıma Kodları

Bir bilgisayarın kendisine verilen bilgileri işleyebilmesi gerekir. Bir yanda sayısal veriler iki tabanlı sayı sistemine çevrilirken, sayısal olmayan veriler de bilgisayarın anlayabileceği bir biçime sokulmalıdır. Bu amaçla verileri makineye taşırken bir kod kullanılır.

Bit ve Bayt

Bir Sayının İki Tabanlı Sayı Sisteminde Gösterimi: 2 tabanlı sayı sisteminde her sayı 0 ve 1 simgeleriyle gösterilir. Bu iki simgeden her birine **bit** adı verilir.

Örneğin 35 sayısının iki tabanlı gösterimi, 0 0 1 0 0 0 1 1 şeklindedir. (Bkz. Çizelge.)

Her bit, konumu tarafından belirlenir. Dolayısıyla 1 konumundaki bitin değeri 1, 3 konumundaki bitin değeri 0'dır. Bilgisayarda her bit elektriksel bir sinyaldir, bit 1 değerini aldığında sinyalin düzeyi yüksek (y), 0 değeri aldığında ise alçak (a) olabilir.

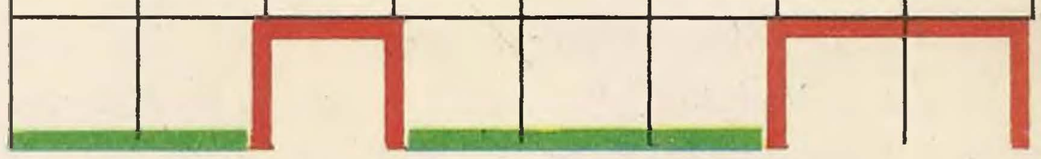
Konum	8	7	6	5	4	3	2	1
2'nin kuvveti	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Değer	128	64	32	16	8	4	2	1
(10 tabanlı) 35	0	0	1	0	0	0	1	1

Yüksek düzey = **y**

Alçak düzey = **a**

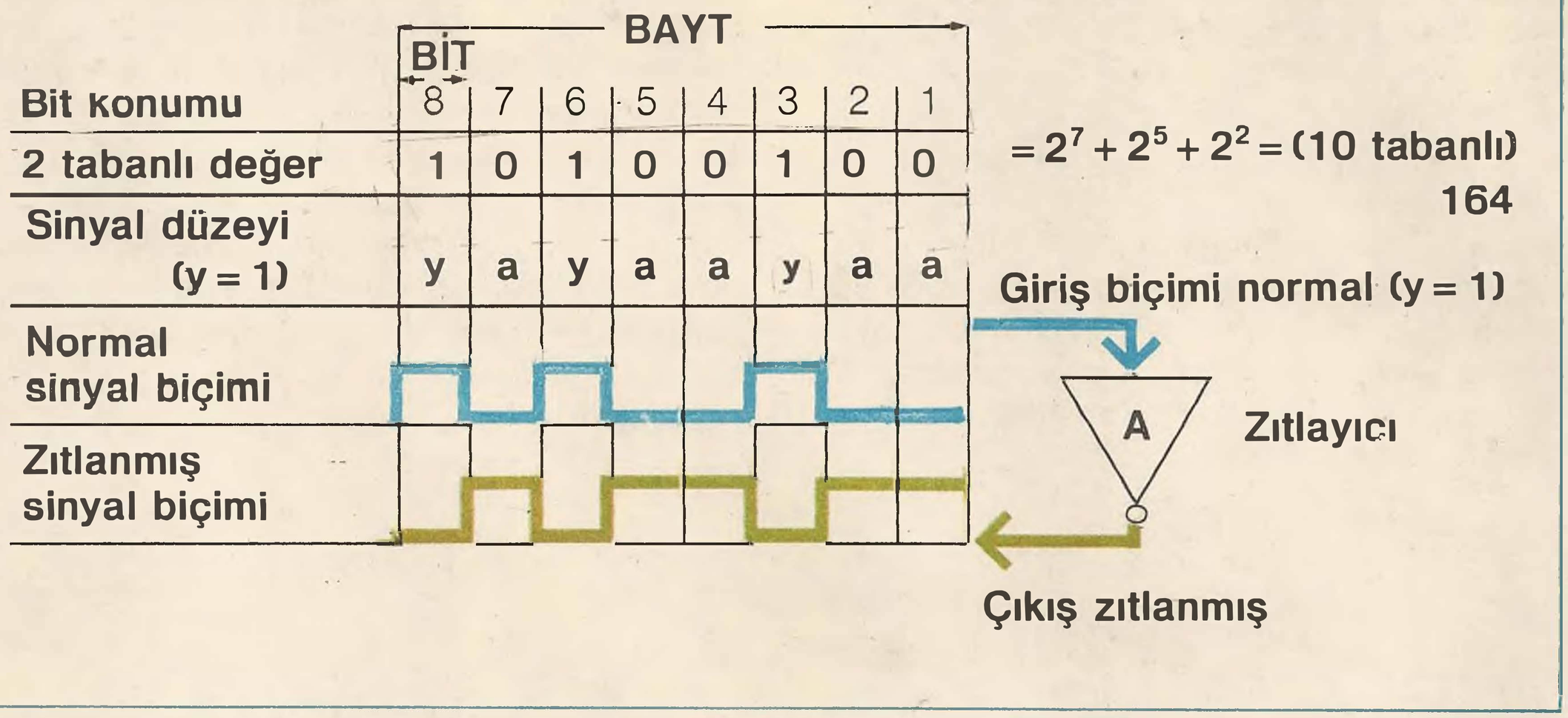
Aşağıdaki çizelgede 10 tabanlı sayı 35 için gerekli sinyaller gösterilmiştir:

Konum = bitin sayısı	8	7	6	5	4	3	2	1
2'nin kuvveti	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
10 tabanlı 35'e karşılık düşen 2 tabanlı değer	0	0	1	0	0	0	1	1
Sinyal düzeyi	a	a	y	a	a	a	y	y
Çizgisel gösterim								



İki tabanlı 1 değerinin, Yüksek düzeyde bir elektriksel sinyalinin karşılığı olduğunu varsaydık. Bunun tersini de yapabilir, iki tabanlı 1 değerine karşılık Alçak düzeyi seçebilirdik. Kimi bilgisayarların işleyişi böyle bir NEGATİF MANTIK temeline dayanır.

Şimdi 10 tabanlı 164 sayısının pozitif ve negatif mantıkta nasıl gösterildiğine bakalım:



Uygulamada bu iki farklı çalışma şeklinin varlığı bir güçlük yaratmaz. Yalnız birinden diğerine geçmek için bir zıtlayıcı gerekir. (Zıtlayıcı şekilde A ile gösterilmiştir.)

Şekilde de görüldüğü gibi, 10 tabanlı 164 sayısının 2 tabanlı karşılığı için 8 bit gerekiyor. 8 bitlik bir gruba **bayt (byte)** denir.

Hepsi 1 değerinde olan 8 bitlik bir dizi, bir baytla ifade edilebilecek en yüksek sayı değeridir.

Öyleyse bir baytın en yüksek değeri:

11111111 =

$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$

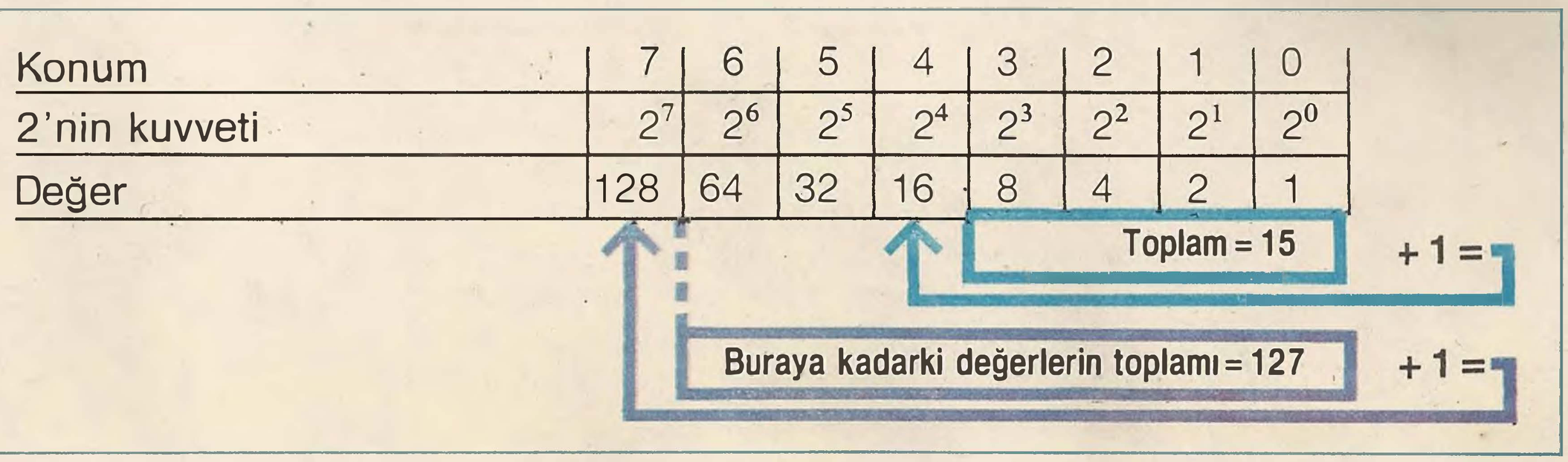
8 bitlik (0'dan 7'ye kadar) bir veri TAŞITI'yla, **(BUS)** (Bkz. s.109) bilgisayarın kullanabileceği en yüksek sayı 255'tir. Uygulamada bu yeterli değildir. Bu kısıtlamayı aşmak için özel devreler birkaç baytı birleştirip daha büyük sayıların işlenmesini olanaklı kılarlar.

Örneğin iki bayt birbirine eklendiğinde 16 bit elde edilir. Bu bağlantıyla ifade edilebilecek en büyük sayısal değer 65535'dir. Kullanılan en büyük bağlantılar 2 baytlık (16 bit) ve 4 baytlık (32 bit) bağlantılardır. Herhangi bir diziye tek bir bitin eklenmesi, o dizinin solundaki bit değerinin 2 ile çarpılmasına denktir. Her bitin sayısal değeri, kendinden önceki (sağındaki) bit değerlerinin toplamına 1 eklenerek bulunur.

Aşağıdaki şekilde dördüncü basamaktaki bitin değeri 16'dır; bu değer sağındaki değerlerin toplamının 1 fazlasına eşittir. (0,1,2,3 konumundaki bit değerlerinin toplamı:

1 + 2 + 4 + 8 = 15.)

Aynı şekilde 7 numaralı bitin değeri 128, 64 + 32 + 16 + 8 + 4 + 2 + 1 toplamına 1 eklenerek bulunur.

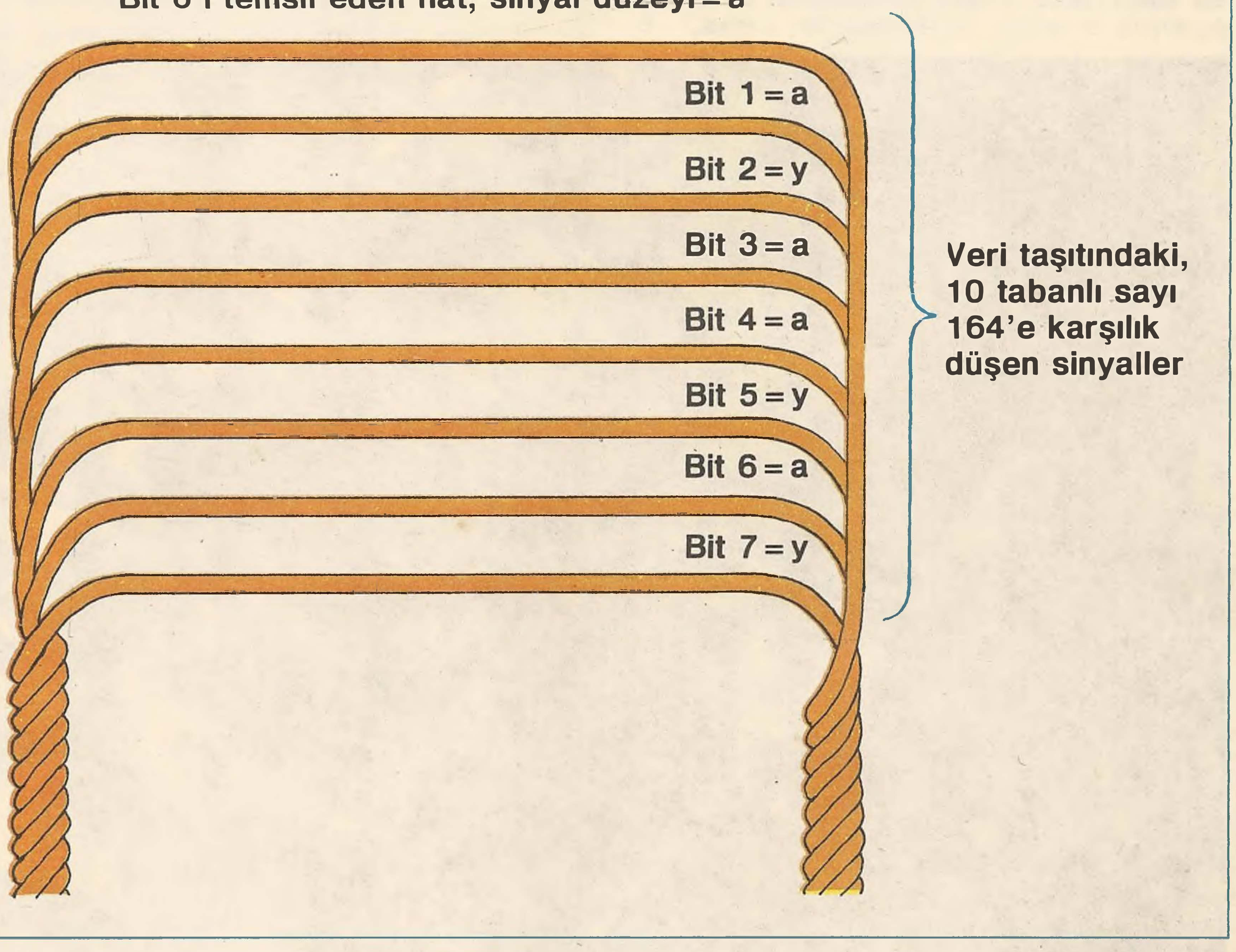


Bir sayının bilgisayarın içinde temsil edilebilmesi için, sayıyı oluşturan bit miktarı kadar elektrik bağlantısına gerek vardır. 8 bitlik bir sayıda 8 bağlantının her biri, temsil ettiği bitin durumunu belirtir.

Bu bağlantılar grubuna ''taşıt'' adını veriyoruz. Bilgisayarlarda veri taşımayı denetleyen iki taşıt vardır. Bunlardan biri verilerin taşındığı hattır; diğeri ise verilerin yerleştirileceği veya alınacağı adresi belirtir.

Minik-işlemcilerde genellikle 8 bitlik (1 bayt) bir veri taşıtı ve 16 bitlik bir adres taşıtı bulunur. Sayfa 108'de üstteki şekilde bir baytın 1'den 8'e kadar numaralanmış bitlerini görüyorsunuz. Ancak bitler genellikle sıfırdan başlayarak numaralanırlar (Bkz. sayfa 108, alttaki şekil); böylece her bit basamağı aynı zamanda 2'nin kuvvetini belirleyen üste eşit olur (bit = 2^0, bit = 2^1 vb).

Bu numaralama şekli bu sayfada gösterilmiştir. Aşağıda bir veri taşıtında 164 sayısına karşılık gelen sinyalleri görüyorsunuz:



- İki tabanlı sistemde en küçük nicelik birimi 0 ya da 1 değerini alabilen BİT'tir.
- 8 bitlik bir gruba BAYT (Byte) adı verilir. Bir baytın ifade edebileceği en büyük 10 tabanlı sayı 255'tir.
- Verilerin taşındığı elektriksel bağlantıya veri taşıtı denir. Kişisel bilgisayarların çoğunda 8 hattan oluşur ve dolayısıyla aynı anda 8 bit (= 1 bayt) taşıyabilir.
- Bir veriyi almak veya saklamak için, elektronik devrelere verinin bulunduğu ya da gideceği yerin (yani adresinin) bildirilmesi gerekir.
- Veri adresleri, adres taşıtı adındaki elektriksel bağlantı yoluyla taşınırlar. Bu taşıt genellikle 16 bitten (2 bayt) oluşur.

ASCII Kodu

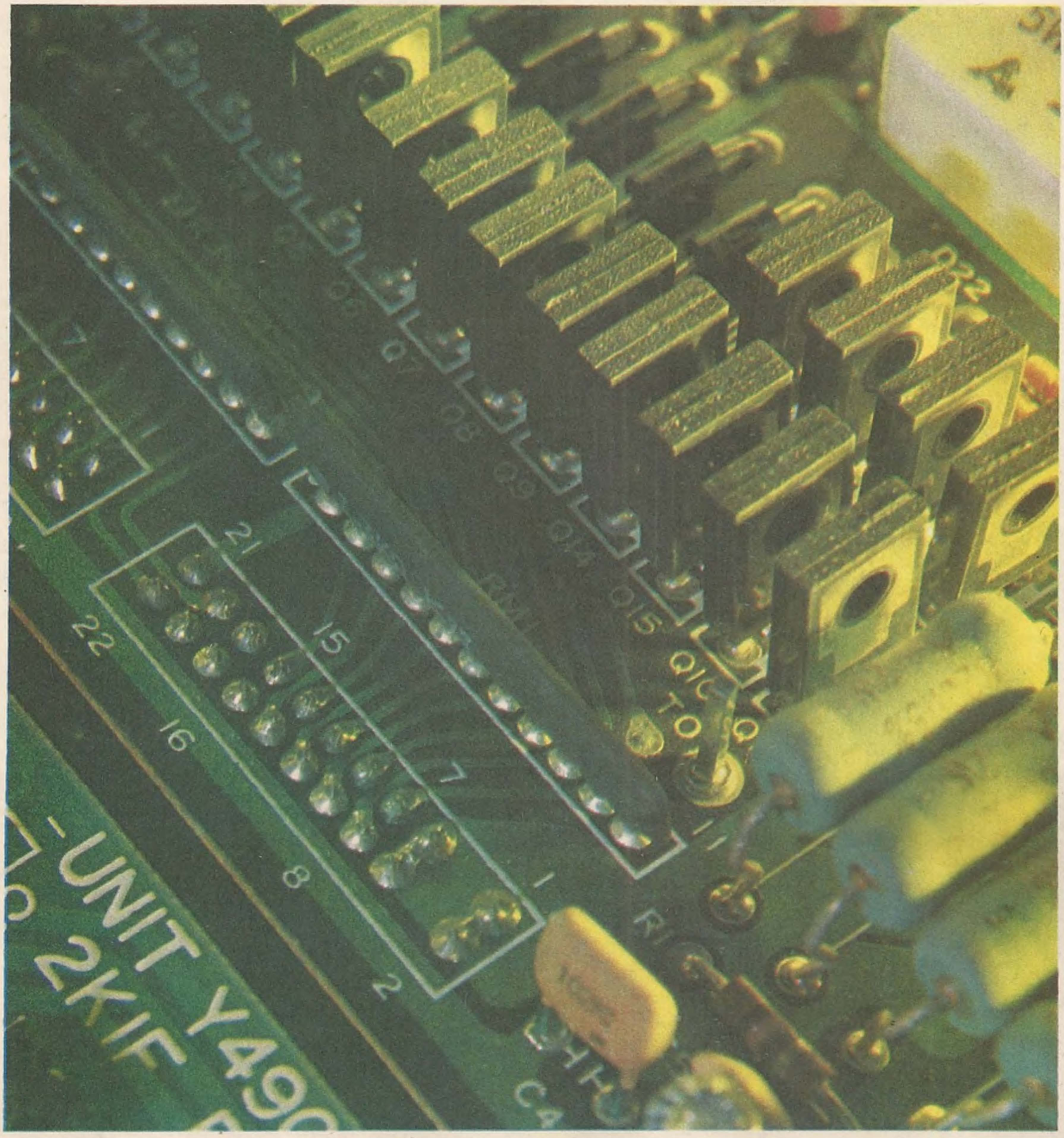
Tüm sayısal aygıtlar gibi bir bilgisayar da yalnızca 0 ve 1 simgelerini ayırdedebilir. Bu nedenle veriler (sözcükler ya da sayılar) bilgisayara girmeden önce bir sıfırlar ve birler dizisine çevrilir. Bilgisayarın içindeki verilere ulaşabilmek içinse bunun tersi işlemi yapmak, yani sıfırlar ve birlerden oluşan bir diziyi harflere ya da 10 tabanlı bir sayıya çevirmek gerekir.

Bu çevirme işlemlerinden birincisine **kodlama** (şifreleme), ikincisine de **kod çözme** (şifre çözme) denir.

Kodlamanın en basit şekli 10 tabanlı bir sayıyı 2 tabanlı eşdeğerine çevirmektir. Bu yeni biçiminde sayı yalnızca 0 ve 1 simgeleriyle ifade edilir.

Alfabenin harfleri için böyle ''matematiksel'' bir kodlama yoktur. Tek çare, tamamiyle isteğe bağlı olabilecek bir kodlama cetveli hazırlayıp harflerin (ya da başka simgelerin) birler ve sıfırlarla gösteriliş biçimini saptamaktır.

Bir bilgiyazarın güdüm devresinden bir kesit.



Bu gösteriliş biçimine KOD denir. Böylece her harfin kendi kodu bulunur ve bilgisayar kodlanmış harfi tanır.

Bilgisayarın verileri dışarıya vermesi için, iki tabanlı bilgileri kullanıcının anlayacağı yazı veya çizgi biçimine dönüştürecek devrelere gerek vardır. Gelişigüzel kodlamalar kullanışlı değildir, çünkü çeviri çizelgeleri farklı olursa bilgisayarlar veri değiş tokuşu yapamazlar, yapsalar da kodların birbirine çevirisi için karmaşık yöntemlere gereksinim doğar. Bundan başka gelişigüzel kodlamada bir verinin doğru çevrilip çevrilmediğini, hatasız olup olmadığını denetlemek mümkün olmaz.

Bu gerekçelerden ötürü üreticiler standart bir kodu, ASCII kodunu kullanıyorlar. ASCII ''Amerikan Standard Code for Information Interchange'' sözcüklerinin kısaltmasıdır. Bu koda göre bir simge 7 bitle gösterilir dolayısıyla 128 simge kodlanabilir
(0'dan 1111111=127'ye kadar).

ASCII kodları, her birinin belirli işlevi olan üç gruba ayrılır:

- **Güdüm işaretleri**
- **Simgeler ve sayılar**
- **Harfler**

Güdüm İşaretleri

Alfabede karşılıkları olmayan ve komutların iletiminde kullanılan kodlardır.

Bilgisayarla ona bağlı çevresel birimler arasındaki veri değişimi sırasında, çevresel birimlerin yönlendirilmesi için birtakım komutlar gereklidir. Örneğin bir bilgiyazarın işleyişinde satır ilerleme ve satır hareketi için bilgisayarın komut vermesi gerekir. Yazma komutlarının sırası şöyledir:

1 - Harf basım biriminin birinci satırın başında hazır olması.
2 - İlk satırın yazılması.
3 - Harf basım birimine ''sayfanın sol kenarına git!'' komutunun verilmesi.
4 - Kâğıdın bir satır ilerlemesi için komut.
5 - İkinci satırın yazılması.
6 - Çevrim yeniden (3) ile başlar ve böyle sürer.

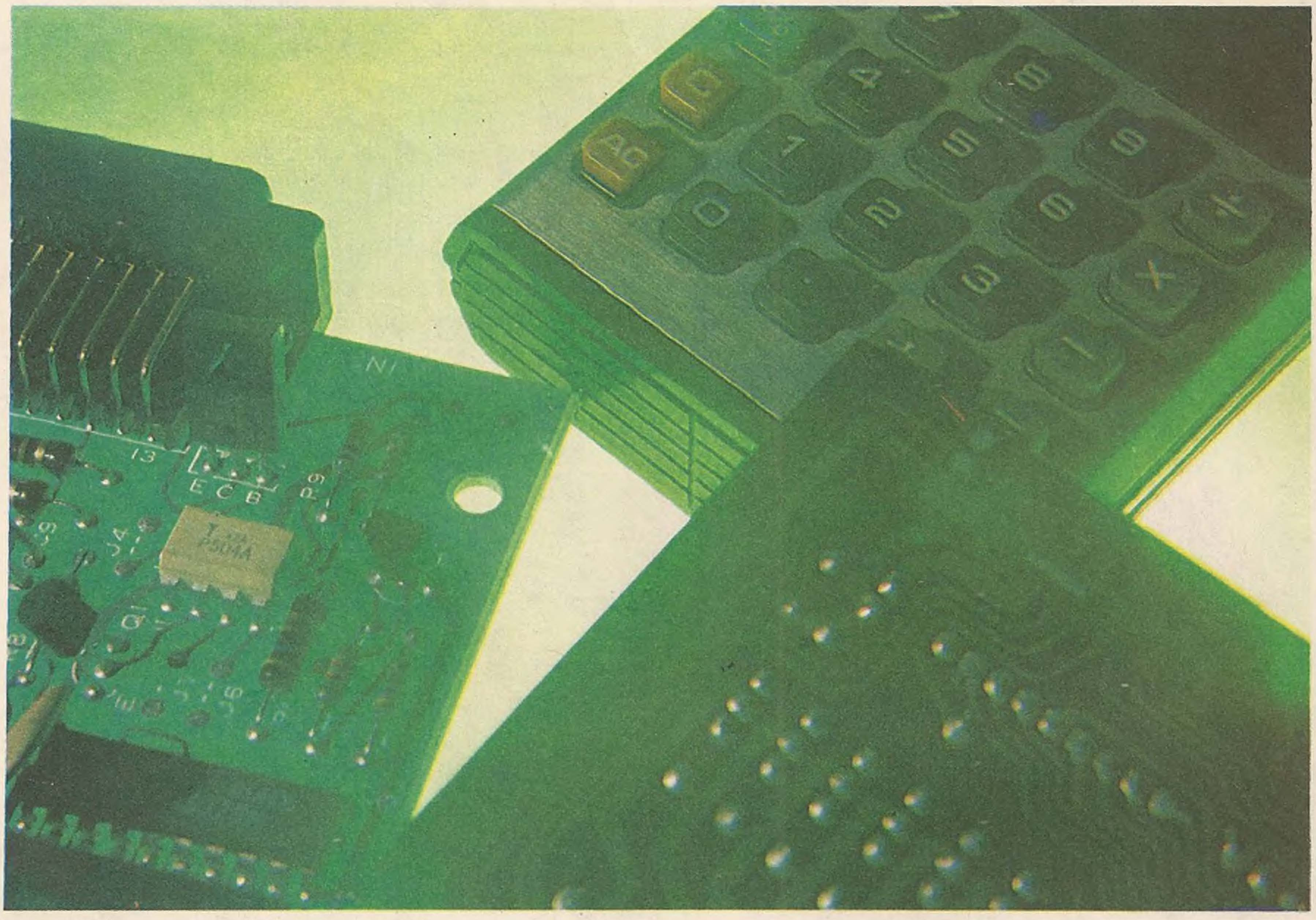
1., 3. ve 4. adımlarda iletilen kodlar, bilgiyazarın güdümünü sağladıklarından, güdüm işaretleri olarak anılır. Varışlarında yorumlanır ve güdümü sağlayan devrelere iletilirler. ASCII kodunda otuzbir tane güdüm işareti vardır ve ilk sıraları, yani 10 tabanlı 1 ilâ 31 (16 tabanlı 1 ilâ 1F) arasındaki değerleri alırlar.

Güdüm işaretlerinin çoğu bilgisayarlar arası veri değiş tokuşu gibi özel amaçlarla kullanılır. Kullanıcı programlarında yer almazlar. Uygulamada en sık kullanılan kodlar aşağıda gösterilmiştir.

Aşağıdaki çizelgede daha iyi anlaşılmaları için ASCII kodlarının Türkçe anlamları da yer alıyor, ancak kodlar normal olarak İngilizce ad-

BAŞLICA GÜDÜM İŞARETLERİ

	Kısaltma	10 tabanlı kod	16 tabanlı kod
1 - Bell Zil işareti	BEL	7	7
2 - Back Space Bir karakterlik geriye dönüş	BS	8	8
3 - Carriage Return Şaryonun geriye dönüşü	CR	13	D
4 - Escape	ESC	27	1B
5 - Form-Feed Sayfa ilerleme	FF	12	C
6 - Horizontal Tabulation Yatay cetvel	HT	9	9
7 - Line-Feed Satır ilerleme	LF	10	A
8 - Vertical Tabulation Düşey cetvel	VT	11	B



Hesap makinesi: Biri açılmış durumda ve içindeki devre görülebiliyor.

ları ile adlandırılır. Kısaltılmış simgeleri de İngilizce adlarından türetilmiştir:

1 - Bell: Terminallerin çoğunda kullanıcının dikkatini çekmek için bir zil bulunur. 7 kodunun verilmesiyle bu zil harekete geçirilir.

2 - Back Space: Bir karakter geriye gitme komutudur. Normal olarak ekrandaki hataları düzeltmek için kullanılır. BS (10 tabanlı 8, 16 tabanlı 8) komutu verilir ve yanlış simgenin üzerine doğrusu yazılır. Yalnızca ekranda kullanılmasının nedeni, ekranda doğru simge yazıldığında yanlışının silinebilmesindendir.

3 - Carriage Return: Şaryoyu geriye döndürme komutudur. Bilgiyazarlarda şaryonun hareketini, ekranda ise simge konumunun yerini belirten işareti (''imleç''i) etkiler.

4 - Escape: Bu kodun hemen görülebilir bir etkisi yoktur. Çevresel gereçlere (giriş-çıkış aygıtlarına) bir sonraki simgenin bir güdüm işareti olduğunu bildirir. Çizim yapan ekranlarda sık sık kullanılır. Çizim ekranlı bir terminal, tasarım çizimleri için de kullanılabilir. Bunun için çizgilerin şekillerini belirtmek üzere birtakım sayılar verilerek imleç kaydırılır.

Ancak bu sayıların ''imleci kaydırma komutu'' olarak verildiklerinin bir şekilde terminale bildirilmesi gerekir. Aksi takdirde terminal, bu sayıları normal yazım simgeleri olarak yorumlar ve istenenden çok farklı bir sonuç elde edilir. Sayıların bu özelliğini, yani imleçi yönlendirmede kullanıldıklarını belirtmek için Escape kodunun verilmesi gerekir.

Komutun biçimi şöyledir:

ESCAPE kodu + sayı

Escape kodu olmaksızın sadece sayı verilirse, imlecin konumunda bir değişiklik olmaz, yalnızca yazılan sayılar ekranda görünür.

5 - Form-Feed: Kâğıdın ilerlemesini sağlar. Her yazma evresinden önce ve sonra yazıcıyı bir sonraki sayfanın başına getirir ve böylelikle değişik bölümlerin (ya da sayfaların) arasında bir boşluk bulunması sağlanır.

6 - Horizontal Tabulation: Bilgiyazarlarda ve ekranda her satır belli sayıda simge alır. Bu sayı sabittir. Her simge, önceden belirlenmiş bir (ya da birkaç) konumu dolduracak şekilde yazılır ve simgeler arasındaki uzaklık sabit kalır. HT komutuyla bu konumlardan belli bir mik-

tarı atlanır. Bu komut, örneğin sol kenar çizgisi ile metin arasında biraz boşluk bırakmak için kullanılır. Bu komutun kodu olan 9 koduyla bilgiyazarın şaryosu (ya da elektron ışını) bir sonraki cetvel (tab) konumuna ilerletilir. Birbirini izleyen iki 9 koduyla ikinci cetvel konumuna ilerleme sağlanır.

7 - Line-Feed: Bu komutla bir sonraki satıra geçilir.

8 - Vertical Tabulation: İşlevi HT'ninkine benzer; ancak bu kez atlama dikey olarak yapılır (satır atlama). Bazı çevresel aygıtlarda HT ve VT kodu programlanabilir, başka bir deyişle, HT için atlanacak karakter sayısı, VT için atlanacak satır sayısı kullanıcı tarafından güdüm işaretlerinin kodlarıyla değiştirilebilir.

Simgeler ve Sayılar

Sayıları, harfleri ve diğer simgeleri kodlayan kodlar grubu 10 tabanlı 32'den (16 tabanlı 20) başlar, 10 tabanlı 64'te (16 tabanlı 40) sona

ASCII KODLARI

10 tabanlı	16 tabanlı	Kısaltma	Güdüm işlevleri
0	0	NUL	Kullanıcı programlarındaki işlevi
1	1	SOH	
2	2	STX	
3	3	ETX	
4	4	EOT	
5	5	ENQ	
6	6	ACK	
7	7	BEL	Ses işareti verir
8	8	BS	İmleç bir basamak sola kayar
9	9	HT	Yatay cetvel
10	A	LF	Bir sonraki satıra
11	B	VT	Düşey cetvel
12	C	FF	Bir sonraki sayfaya
13	D	CR	Satır başına
14	E	SO	
15	F	SI	
16	10	DLE	
17	11	DC1	
18	12	DC2	
19	13	DC3	
20	14	DC4	
21	15	NAK	
22	16	SYN	
23	17	ETB	
24	18	CAN	
25	19	EM	
26	1A	SUB	
27	1B	ESC	ESCAPE
28	1C	FS	
29	1D	GS	
30	1E	RS	
31	1F	US	

erer. Çizelgede harfler gibi 10 tabanlı sayılar da yeni bir biçimde gösterilmiştir. Bilgisayarda iki gösterim biçimi vardır:

- **Sayısal değer**
- **ASCII kod**

Sayısal değer, matematik işlemlerinde, ASCII kodu ise yazmada kullanılır. Örnek olarak 5 sayısı iki biçimde ifade edilebilir:

Sayısal değer = 1 0 1
ASCII kodu = 1 1 0 1 0 1 (16 tabanlı 35)

Harfler

Sayılar ve güdüm işaretlerinden başka, büyük ve küçük harflerle, yazarken kullanılan diğer bazı simgeler de ASCII koduna dahildir.

Uygulama Alanları

Şimdi, ''BU KOMUTA UYMALISIN'' cümlesini yazabilmesi için, bilgiyazara verilmesi gereken komutları görelim:

1 - Önce, ''Sayfanın başından başla'' (FF) komutu verilir.
2 - Bunu yazılacak cümlenin kodları izler.

SİMGELER VE SAYILARIN ASCII KODLARI

10 tabanlı	16 tabanlı	Simge veya sayı	
32	20	Boşluk	
33	21	!	
34	22	''	
35	23	#	
36	24	$	
37	25	%	
38	26	&	
39	27	'	
40	28	(	
41	29	)	
42	2A	*	
43	2B	+	
44	2C	,	
45	2D	—	
46	2E	.	
47	2F	/	
48	30	0	
49	31	1	
50	32	2	
51	33	3	
52	34	4	
53	35	5	
54	36	6	
55	37	7	
56	38	8	
57	39	9	
58	3A	:	
59	3B	;	
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	
64	40	@	

HARFLERİN ASCII KODLARI

10 tabanlı	16 tabanlı	Simge
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	—
96	60	`
97	61	a
98	62	b
99	63	c

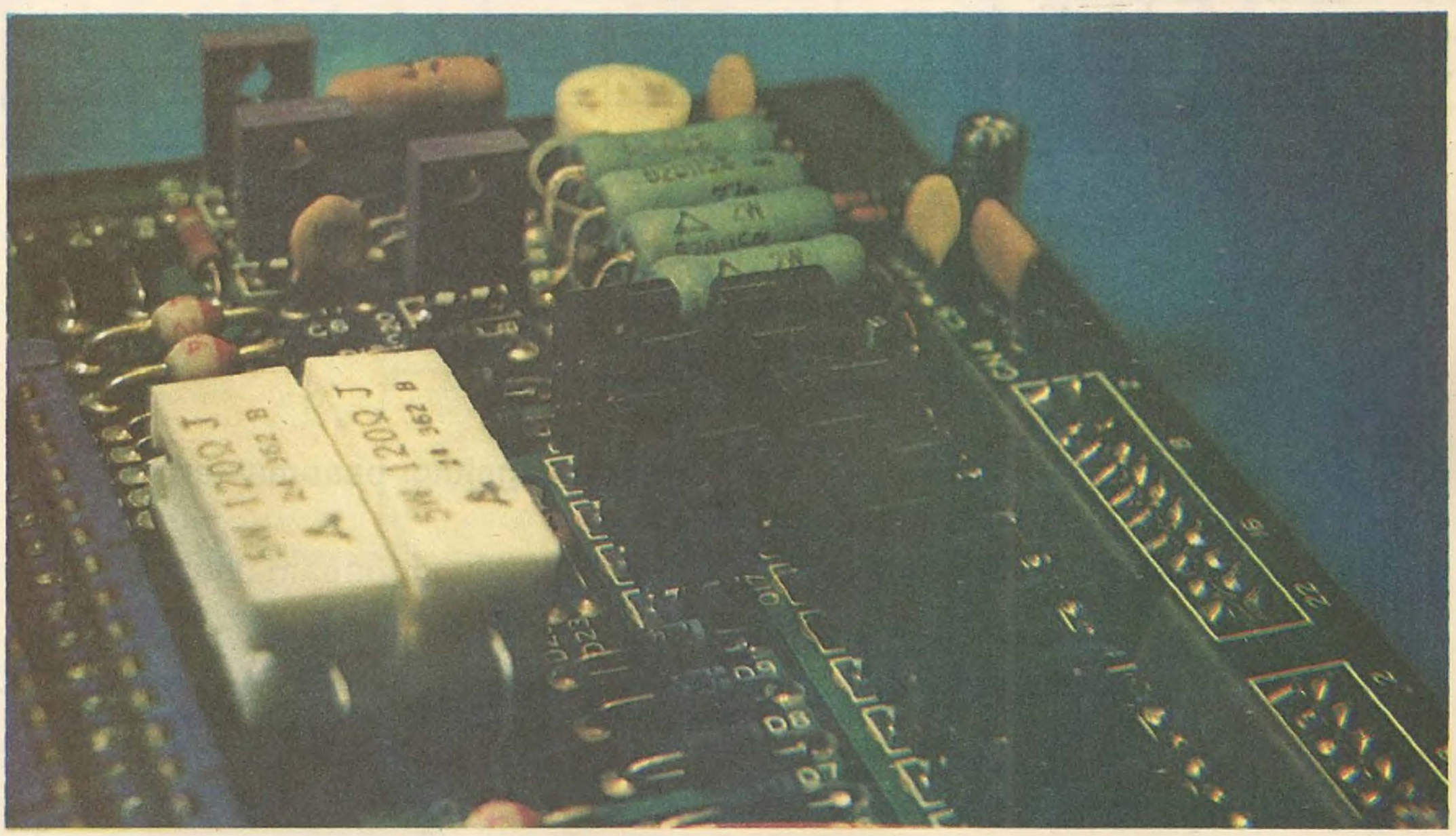
10 tabanlı	16 tabanlı	Simge
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL

3 - Yeniden bir sayfa başı komutu (FF) verilir.

Birinci ve üçüncü adımlarda kullandığımız FF komutunun kodu 10 tabanlı 12'dir (=16 tabanlı C). İkinci adımda ise cümle harf harf kodlanır. Sayfa 114'teki ASCII kodları tablosundan bu cümle için bize gerekli olan kodları buraya aktaralım:

Harf	16 tabanlı kodu
B	42
U	55
boşluk	20
K	4B
O	4F
M	4D
U	55
T	54
A	41
boşluk	20
U	55
Y	59
M	4D
A	41
L	4C
I	49
S	53
I	49
N	4E
	2E

Bir bilgisayarın besleme gerilimini sağlayan tümleşik devreler. Diğer devre elemanlarına göre daha büyük boyutlarıyla kolayca ayırt ediliyorlar.



FF	B U K O M U T A U Y M A L I S I N .	FF
12	42 55 20 4B 4F 4D 55 54 41 20 55 59 4D 41 4C 49 53 49 4E 2E	12

Toplu Halde ASCII Kodları

Uygulamada ASCII kodları tablosu daha çok aşağıda görülen sıklaştırılmış şekliyle kullanılır. Bu tabloda her simge kodunun 2 ve 16 tabanlı değerleri verilmiştir:

ASCII KODLARI

7 6 5 4 3 2 1 ⇨ Sütun / Satır		000	001	010	011	100	101	110	111
		0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC 1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC 2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC 3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC 4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10 (A)	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11 (B)	VT	ESC	+	;	K	[	k	{
1 1 0 0	12 (C)	FF	FS	,	<	L	\	l	|
1 1 0 1	13 (D)	CR	GS	-	=	M	]	m	}
1 1 1 0	14 (E)	SO	RS	.	>	N	^	n	~
1 1 1 1	15 (F)	SI	US	/	?	O	_	o	DEL

Bir örnek olarak 4. sütun ve 6. satırda yer alan **F** simgesini ele alalım. 4 ve 6 sayılarını birleştirirsek **F** simgesinin kodunun 16 tabanlı sayı sistemindeki değerini, 4 ve 6 sayılarının 2 tabanlı karşılıklarını birleştirirsek bu kodun 2 tabanlı sayı sistemindeki değerini buluruz:

6. Satırın değeri 0 1 1 0 — **Karşılık geldiği basamaklar: 4,3,2,1**

4. Sütunun değeri 1 0 0 — **Karşılık geldiği basamaklar: 7,6,5**

İkisinin birleştirilmesi: F = 1 0 0 0 1 1 0 **= 70 (10 tabanlı) = 46 (16 tabanlı)**

4 6

Bu sonucu sütun ve satır sayılarını ardarda yazarak elde ediyoruz:

F = sütun **4**, satır **6** = **46** (16 tabanlı).

Şimdi de 6. sütun ve 13. satırda (yani 16 tabanlı **D** satırında) yer alan **m** simgesinin değerini bulalım:

m = sütun **6**, satır **13**(= **D**) = **6D** (16 tabanlı).

ASCII Kodlarının Taşınması

Bilgisayarla çevresel gereçler arasındaki veri taşınması iki biçimde gerçekleşir:

- **seri taşıma**
- **paralel taşıma**

Seri taşımada bitler hedefe sırayla, birbirlerini izleyerek varır. Paralel taşımada ise belirli sayıda bit, hedefte aynı anda belirir. Bitler teker teker gönderildiklerinden, seri taşıma için tek bir elektrik hattı yeterlidir. Paralel taşımada ise, aynı anda kaç bit gönderilecekse o kadar hat gerekir. Sayfa 118'deki şekiller her iki taşıma yöntemini de göstermektedir.

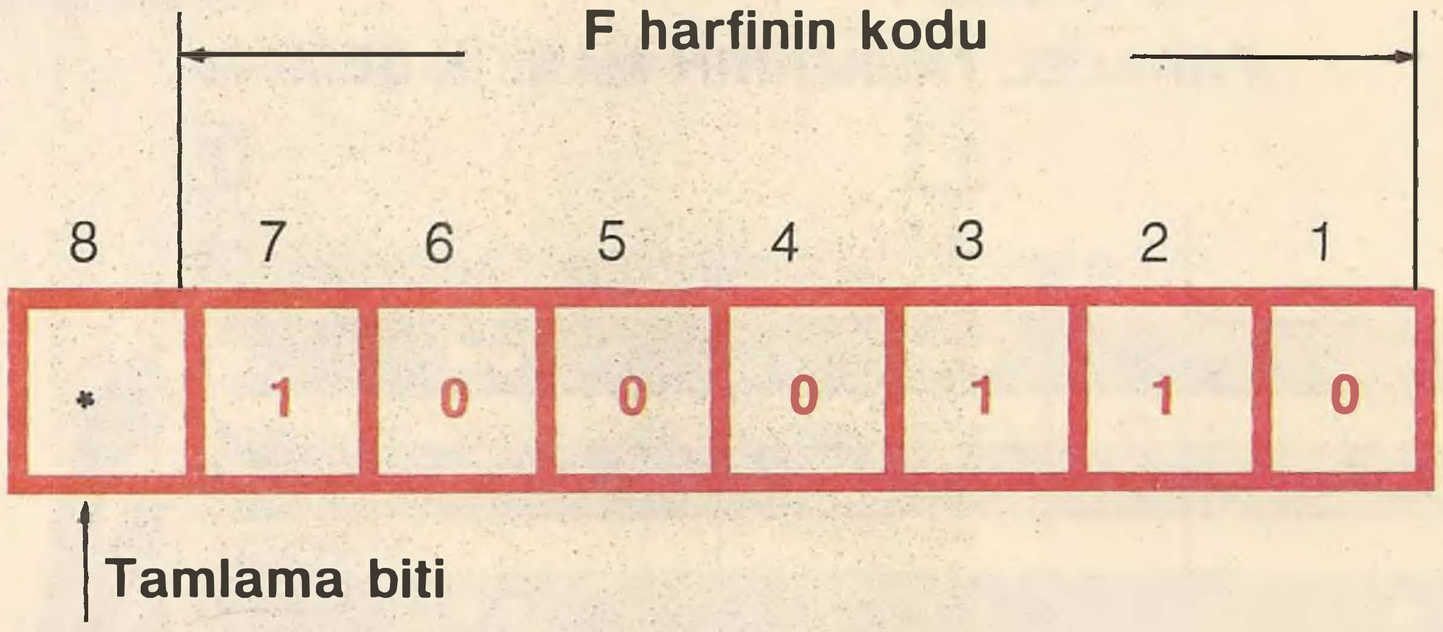
Paralel taşıma yüksek bir taşıma hızı sağlar, ancak çok sayıda hat gerektirdiğinden pahalıya mal olur; dolayısıyla uzun mesafelerde kullanılmaz. Buna karşılık seri taşıma yöntemi daha yavaş olmakla birlikte, ucuz olduğundan daha sık kullanılır. Bundan böyle biz de yalnızca seri taşımadan söz edeceğiz.

Veri taşıma yöntemleri başka bir özellikleri yönünden, **eşzamanlı** ya da **eşzamansız** olmaları bakımından da ikiye ayrılırlar.

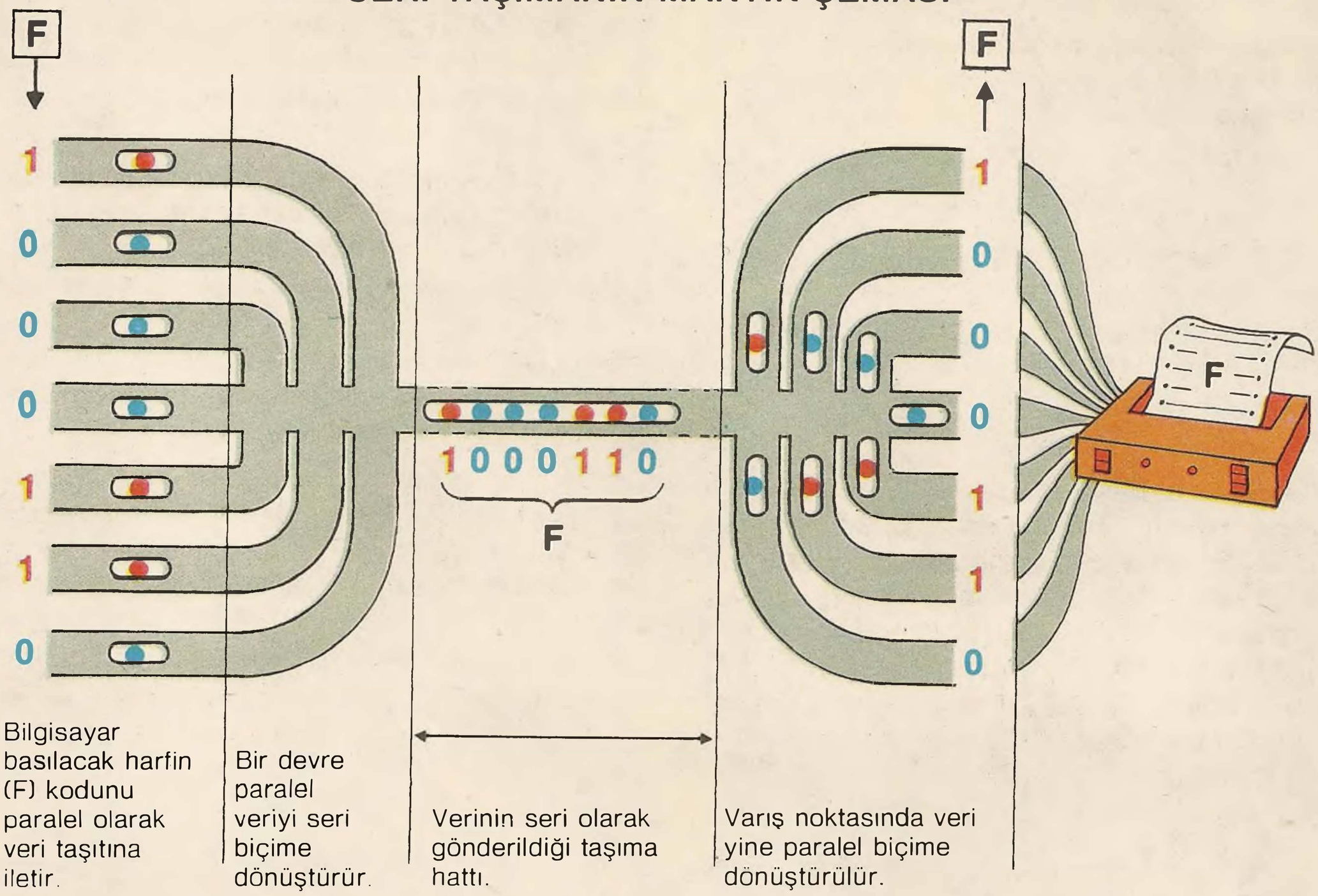
Eşzamansız veri taşımada iki işaret arasındaki zaman aralığı değişebilir, ancak tek bir işaretin bitleri arasındaki zaman aralığı değişmez, sabit kalır. Şekilde **F** ve **H** harflerinin eşzamansız olarak iletimi gösteriliyor.

Eşzamanlı taşımada ise veriler, veri taşıma aralığı ya da ÇERÇEVE denilen sabit zaman aralıklarında yollanır. Her çerçeve bir diğerinden eşzamanlamaya yarayan özel birtakım kodlarla ayrılır. Sayfa 119'daki **F** ve **H** harflerinin eşzamanlı iletimini gösteriyor.

Aşağıdaki şekilde ASCII kodundaki yedi bit dışında yanında bir de * simgesi ile gösterilen



SERİ TAŞIMANIN MANTIK ŞEMASI



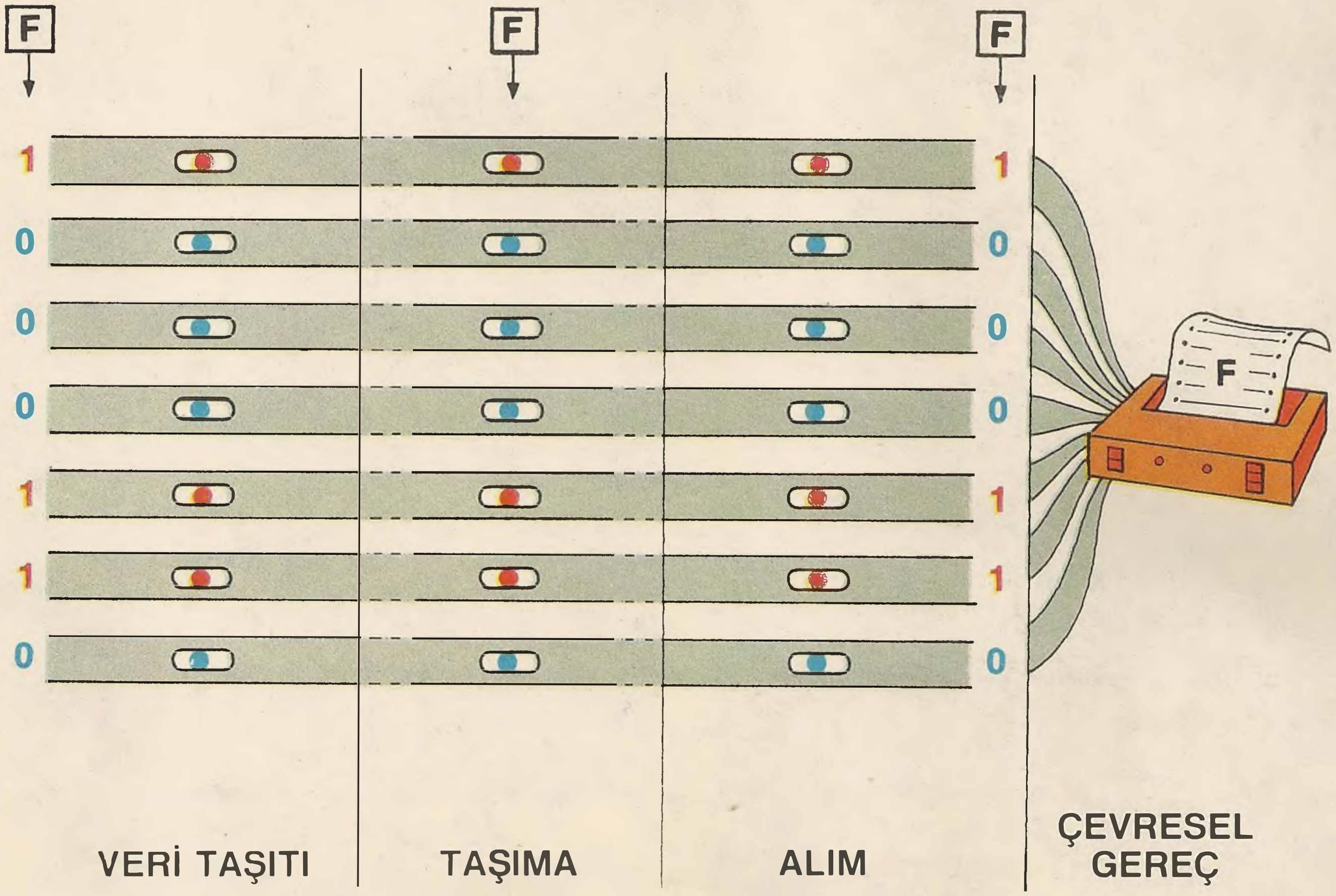
Bilgisayar basılacak harfin (F) kodunu paralel olarak veri taşıtına iletir.

Bir devre paralel veriyi seri biçime dönüştürür.

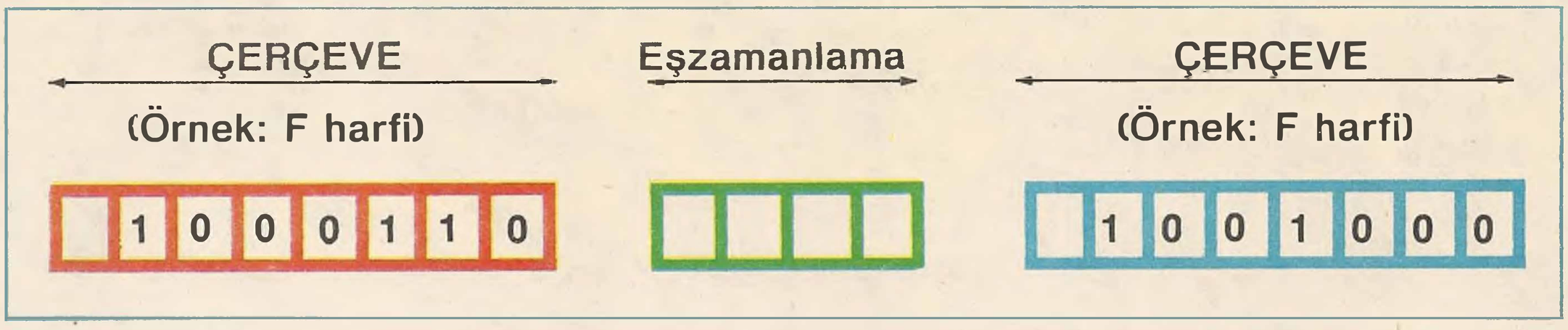
Verinin seri olarak gönderildiği taşıma hattı.

Varış noktasında veri yine paralel biçime dönüştürülür.

PARALEL TAŞIMANIN MANTIK ŞEMASI

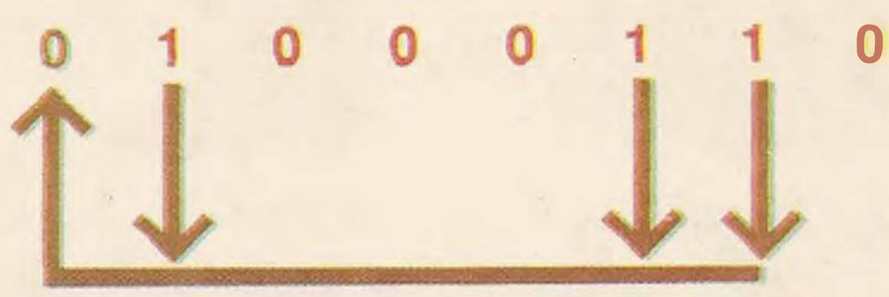


VERİ TAŞITI | TAŞIMA | ALIM | ÇEVRESEL GEREÇ

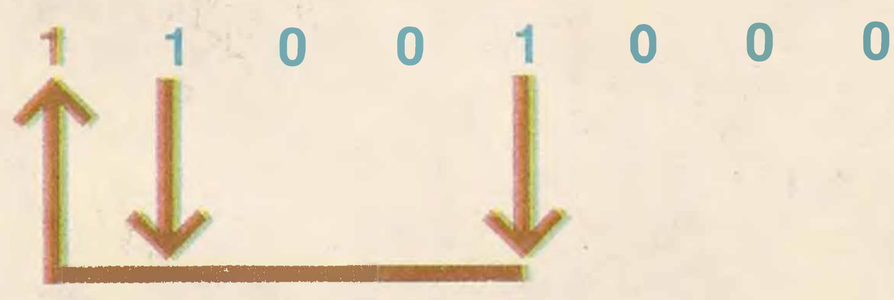


bir bit var. Gönderilen verinin doğruluğunu sınamaya yarayan bu bite TAMLAMA biti adı verilir. Bu sınama diğer 7 bit arasında 1 değeri almış olanların toplanmasıyla yapılır; eğer çıkan sonuç çiftse ve ''teke tamlama'' yolu seçilmişse sekizinci bite (tamlama bitine) 1 değeri, aksi takdirde 0 değeri verilir.

Yine sayfa 117'deki şekilde gördüğümüz gibi, **F** harfinin kodunda (1000110) üç tane 1 var; dolayısıyla tamlama biti 1 değerini alıyor. **H** harfinin kodunda (1001000) ise 2 tane 1 var, dolayısıyla tamlama biti olarak 0 ekleniyor. Bu işlemden sonra veri artık şöyle yazılacaktır:



Tek sayı (üç tane 1)
Tamlama biti = 0



Çift sayı (iki tane 1)
Tamlama biti = 1

Örneğimizde 1'lerin toplamı tek bir sayı ediyorsa, tamlama biti 0 değerini alıyor. Ancak bu örneklerdeki mantığı tersine çevirerek ''çifte tamlama'' yolunu da seçmek mümkün. O zaman tamlama bitinin alacağı değerler şöyle olacaktır:

1'lerin toplamı tek sayı ise, tamlama biti = 1
1'lerin toplamı çift sayı ise, tamlama biti = 0

Eşzamansız taşıma için tamlama biti dışında bir de verinin başlangıcını ve sonunu belirten tanımlama sinyalleri gerekir.

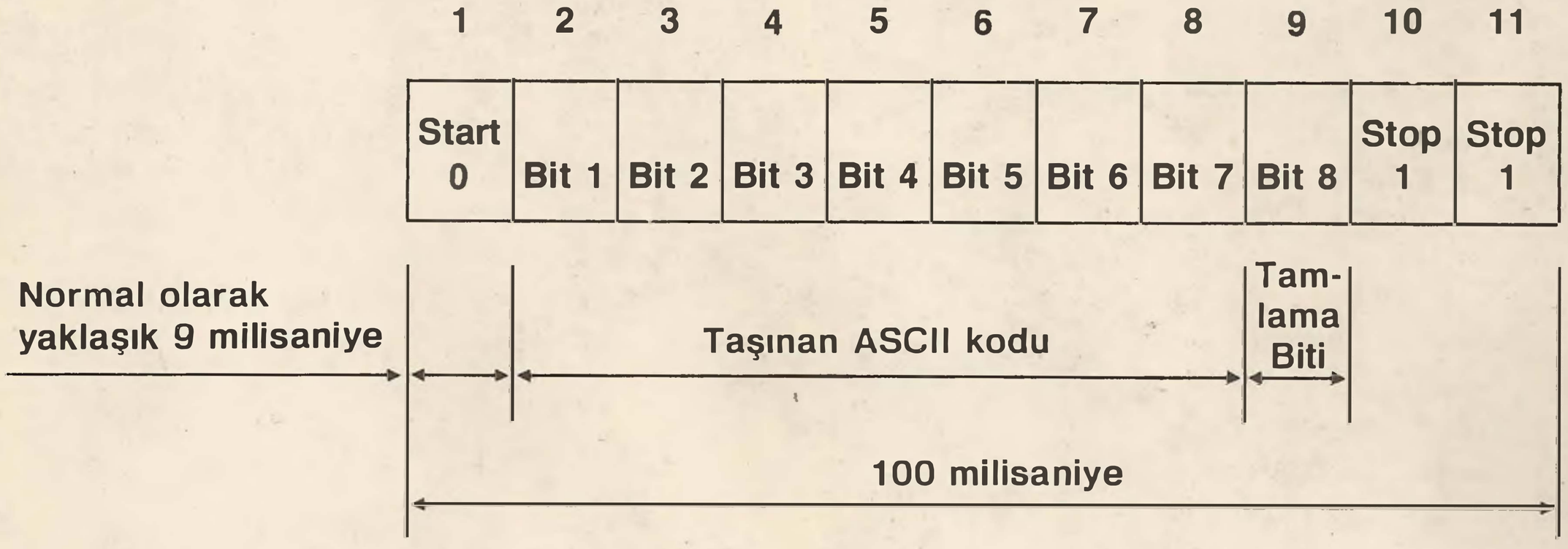
Başlangıç sinyali START (BAŞLA), bitiş sinyali ise STOP (DUR) simgeleriyle gösterilir. Genellikle verinin bitişini göstermek için iki tane STOP biti kullanılır. O halde tek bir karakter kodu şu kısımlardan oluşur:

- START biti
- 7 tane veri biti
- Tamlama biti
- Birinci STOP biti
- İkinci STOP biti

Taşıma hızı BAUD adı verilen bir birimle ölçülür; BAUD bir saniyede gönderilen veri miktarını belirtir.

Şekilde verinin 150 Baud'luk bir hızda nasıl gönderildiği gösteriliyor. Bu taşımada START biti 0, STOP biti ise 1 değerini alır. 1 durumuna MARK (işaret), 0 durumuna ise SPACE (boşluk) adı verilir.

SİNYALLERİN 150 BAUDLA TAŞINMASI



- **Veriler, (simgeler, sayılar, harfler) ASCII kodu ile kodlanır.**

- **ASCII kodu hem çevresel gereçlere (ekran, bilgiyazar) veri taşımada, hem de bilgisayarlar arası iletişimde kullanılır.**

- **Verilerin çevresel gereçlere taşınmasını düzenleyen kodlara Güdüm İşaretleri denir.**

- **Veriler seri ve paralel olmak üzere iki şekilde taşınabilir.**

- **Daha çok kullanılan seri taşıma eşzamanlı ya da eşzamansız olabilir.**

- **Eşzamansız taşıma için 7 bitlik veriye şu bitler eklenir:**

  - START biti (normal olarak 0 değerini alır);

  - Tamlama biti (7 veri bitinin sonundadır);

  - İki STOP biti (normal olarak 1 değerini alırlar).

Derste bilgisayar. Bir öğrenci bilgiyazar çıktılarını inceliyor.

Şifrenin Alım Sırasında Kontrolü

Bir işareti eşzamansız olarak iletmek için verilerin doğruluğunu denetlemek üzere bir tamlama biti eklenir. İşaretin varış yerinde verideki 1'lerin sayısıyla tamlama bitinin değeri karşılaştırılır.

Alıcı birimin giriş devresi birinci biti (START bitini) ve son iki biti (stop bitlerini) çıkardıktan sonra verinin yedi biti ile tamlama bitinden oluşan, toplam 8 biti çevresel gerece ulaştırır.

Alınan verinin doğru olup olmadığı üç adımda sınanır:

1 / Sekizinci bitin (yani tamlama bitinin) ayrılması,
2 / Kalan 7 bitteki 1'lerin sayılması,
3 / 1'lerin toplamı tek ve sekizinci bit 0 ise ya da 1'lerin toplamı çift ve sekizinci bit 1 ise taşınan verinin doğru olarak taşındığına karar verilmesi.

Bu işlemler dizisini bilgisayara bir kullanıcı programı olarak vermek mümkündür.

Sekizinci bitin 7 bitlik veriden ayrılmasında VE (AND) işlevi kullanılabilir:

Sekizinci bitin değeri = (varan bilgi) AND (10000000)

Kalan 7 bitin değeri = (varan bilgi) AND (01111111)

Herhangi bir iki tabanlı sayı (örneğimizde varan bilgi) ile 10000000 değerine AND (VE) işlemi uygulandığında sonuç, varan bilginin ilk bitinden, yani tamlama bitinden oluşur. Örnek olarak F (1000110) harfini ele alalım:

← 8 bit →	
1 0 0 0 1 1 0	0
Veri = F	Tamlama biti = 0, çünkü 3 adet 1 simgesi var.

Varan iletinin START ve STOP bitleri çıktıktan sonraki durumu

Verinin belirli bir kısmını ayırmak için VE (AND) işleminde kullanılan sabit değere (örneklerde 00000001) ''MASKE'' adı verilir. Bu maske sayı sayesinde veriyi, ancak belli bir kısmı okunabilecek bir şekilde örtmüş oluyoruz.

	← Varan ileti →	Tamlama biti
Veri = F	1 0 0 0 1 1 0	0
Maske = 1	0 0 0 0 0 0 0	1
	0 0 0 0 0 0 0	0
	İşlemden sonra ilk 7 bitin sonucu	Tamlama biti

Aynı yöntemi bir de H = 10010001 harfinde görelim:

← İleti →	
1 0 0 1 0 0 0	1
Veri = H	Tamlama biti = 1, çünkü 2 adet 1 simgesi var.

Veri AND 00000001:

Veri = H	1 0 0 1 0 0 0	1 Tamlama biti
Maske = 1	0 0 0 0 0 0 0	1
	0 0 0 0 0 0 0	1
	İşlemden sonra ilk 7 bitin sonucu	Tamlama biti

Varan bilgiden, iletilen harfin kodu olan 7 biti ayırmak için ise 11111111 maskesi kullanılır.

Veri AND 11111111

Veri = F	1 0 0 0 1 1 0	0
Maske =	1 1 1 1 1 1 1	1
	1 0 0 0 1 1 0	0
	İşlemin sonucunda verinin ASCII kodundaki 7 biti çıkar.	Bu maskeyle sonuncu bit hep 0 olur.

Veri = H	1 0 0 1 0 0 0	1
Maske =	1 1 1 1 1 1 1	1
	1 0 0 1 0 0 0	0

Bu örnekler, iyi hazırlanmış programlarla, verilerin ve tamlama bitlerinin birbirlerinden nasıl ayrılabileceğini gösteriyor. Modern alıcı devrelerinde bu işlevler, bir program gerektirmeksizin, ''donanım'' tarafından yerine getirilir. (örneğin tümleşik bir AND devresi)

Güvenlik Kodları

Sayfa 78'deki tablolarda, çeşitli Boole işlemleri, maske olarak kullanılan 11'le 65 ilâ 90 (hepsi 10 tabanlı) arasında değişen sayılara uygulandığında elde edilen sonuçlar yer alıyor. 65 ilâ 90 değerleri büyük harflerin ASCII kodlarıdır (A = 65, Z = 90).

Tablodaki XOR (DAR VEYA) sütunu mükem-

mel bir kodlama aracı işlevini görmekte (her harfin kodundan hareketle yeni bir kod üretiyor) ve kullanılan maske bilinmediği sürece kullanılan kodla üretilen kod arasındaki bağlantıyı keşfetmek olanaksız.

Sayfa 78'de çizelge 3'ün birinci satırında

11 XOR 65 = 74

işlemiyle A (65) harfinden çıkarak J (74) üretiliyor. Maske olarak 11'i değil 32'yi kullandığımız takdirde (bkz. sayfa 78 tablo 4)

32 XOR 65 = 97

yani A'nın karşılığı (a)'nın değeri olarak bulunuyor.

İşarete gürültü karıştırma yöntemi ile veri taşımada yeterli bir güvenlik derecesi sağlanabilir. Bununla birlikte kodlamaya dayanan güvenlik sistemlerini çözme yöntemleri de vardır. Bilgisayarın yüksek hızı sayesinde çok sayıda deneme ve birbirini izleyen yaklaştırmalar sonucu kod anahtarı bulunur.

Test 2'nin Çözümleri

1 / Önce A XOR B'yi, sonra onun değillenmişini bulmak çözümü kolaylaştırır.

A	B	A XOR B	$\overline{\text{A XOR B}}$	A AND ($\overline{\text{A XOR B}}$)
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**1**	**1**
0	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**

Görüldüğü gibi A AND (A XOR B) ile A AND B aynıdır.

---

2 / a - 11011 AND 11 = 00011

b - 12 ve 7 ondalık sayıları önce iki tabanlıya çevrilmelidir:

10 tabanlı 12 = 1100 10 tabanlı 7 = 0111

12 AND 7 = 1100 AND 0111 = 0100

Sonucu 10 tabanlı sayı sistemine çevirdiğimizde

(0100 = 4) = 12 AND 7 = (10 tabanlı 4)

c - 8 tabanlı 12 ve 7'nin 2 tabanlı karşılıkları:

sekiz tabanlı 12 = 001010

sekiz tabanlı 7 = 000111

Dolayısıyla:

12 AND 7 = 001010 AND 000111 = 000010

Ya da:

12 AND 7 = (sekiz tabanlı) 2

d - 1101 OR 0110 = 1111

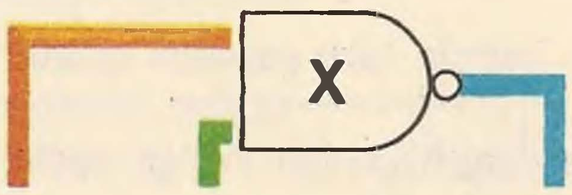
e - 1011 XOR 1101 = 0110

---

3 / Çözüm için AND, OR, XOR ve NOT Boole işlemlerini açıklayan bölümlerden yararlanınız. (Bkz. s.80-83, 88-90)

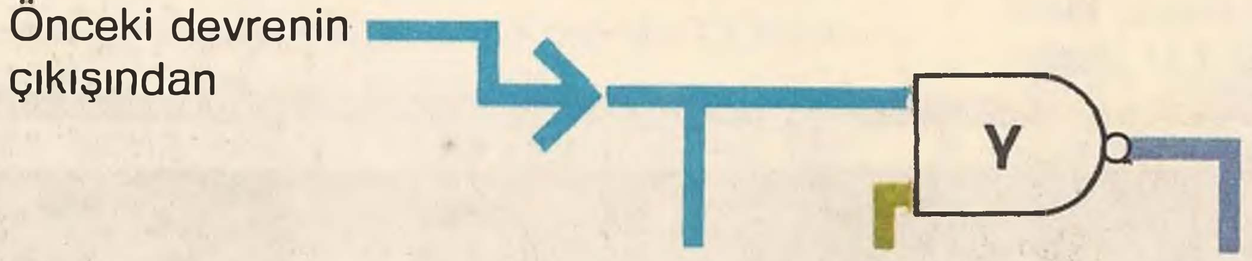
---

4 / a - Bileşik devrelerde önce ilk devrenin doğruluk değerleri tablosu bulunur ve bu tablo bir sonraki devreye uygulanır. X devresi için:



A	B	$\overline{\text{A AND B}}$
1	1	0
1	0	1
0	1	1
0	0	1

A AND B sütunundaki değerler (0111) ikinci devrenin girişlerinden biri olarak yazılır:



A	B	$\overline{\text{A AND B}}$
0	1	1
1	1	0
1	1	0
1	1	0

A	F	A NAND F	B	$\overline{\text{B OR (A NAND F)}}$
1	1	0	0	1
0	1	1	1	0
		0	1	0
		1	0	0

Tablonun ikinci kısmında A NAND F'nin değerleri, B nin değerleriyle birarada kullanılmayı mümkün kılmak için tekrarlanmıştır.

TEST 3

1 / 7, 9, 0, 3 rakamlarının 2 tabanlı karşılıklarını ve simge kodlarını bulunuz.

2 / Seri ve paralel taşıma arasındaki başlıca farklılıkları sayınız.

3 / K harfini iletmek için gereken iletinin tam yapısını gösteriniz.

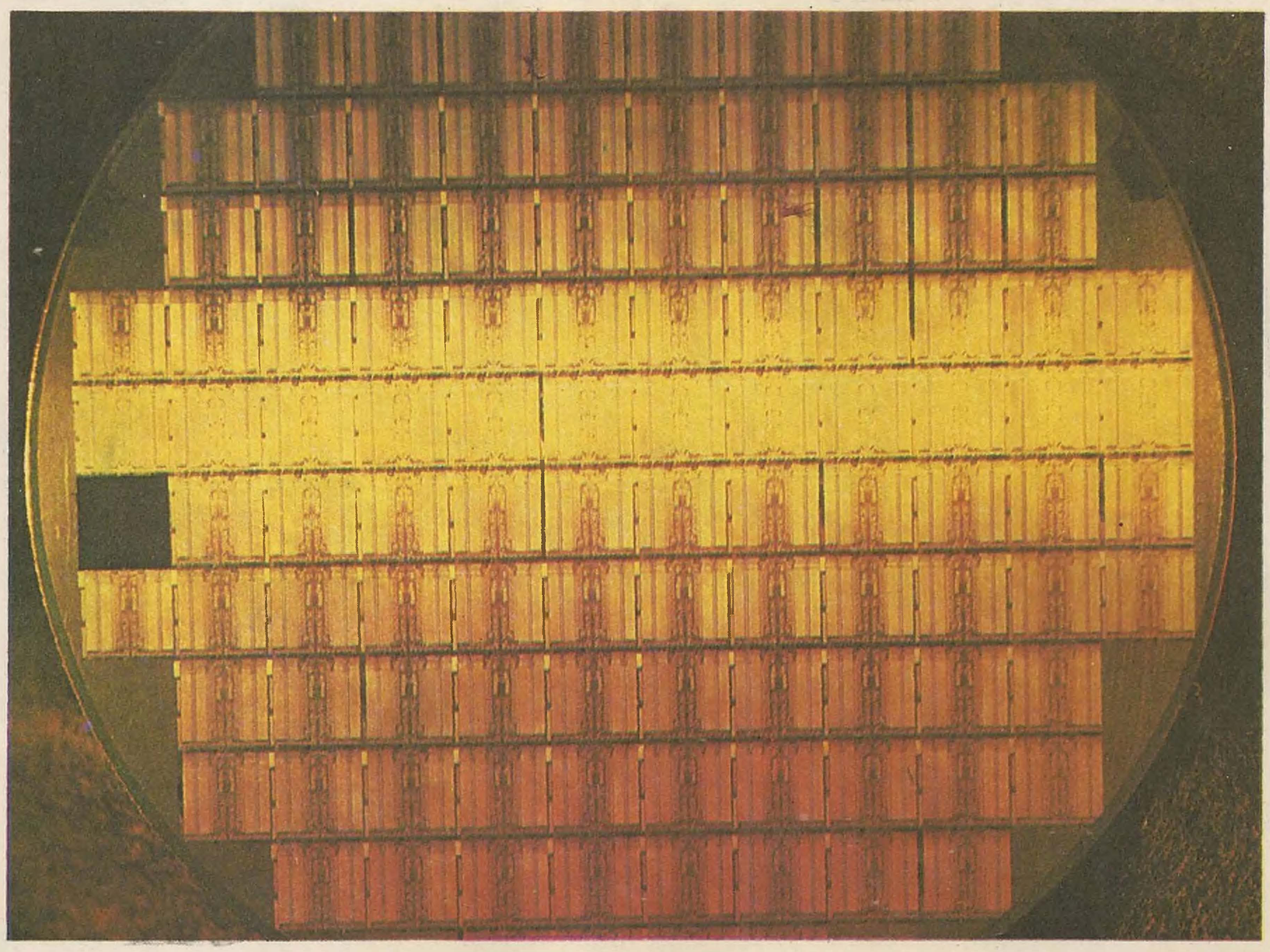
4 / Bilgiyazarın aşağıdaki görevleri yapması için hangi kodların verilmesi gerektiğini ve bu kodların 10 tabanlı değerlerini bulunuz:

a - Sayfa ilerleme
b - X ve Y harflerinin basımı
c - Satır ilerleme
d - x ve y harflerinin küçük basımı
e - sayfa ilerleme

5 / ASCII kodları tablosuna bakarak büyük bir harfin küçük harf haline getirilmesi için ne yapılması gerektiğini söyleyiniz.

6 / 0'dan başlayarak numaralanmış bitlerden oluşan bir verinin 3, 7 ve 1. bitlerini çıkaran maskeyi yazınız. Sonucun doğruluğunu 12, 21, 6 ve 15 (10 tabanlı) sayılarıyla denetleyiniz.

Testin çözümleri s. 150-151'dedir.



Josephson ilkesinden yararlanan bu sayısal makine çok yüksek bir hızda çalışır.

Kişisel Bilgisayar Sistemleri

Kişisel bilgisayar sistemlerini doğru kullanabilmek için bunların fiziksel yapıları hakkında bazı temel şeyleri bilmek gerekir.

Kişisel bilgisayarlar üç amaca hizmet edebilirler:

- **Hazır programlardan faydalanma,**
- **Hazır programların ihtiyaca göre değiştirilmesi,**
- **Yeni uygulamalar için program geliştirilmesi.**

Kuşkusuz farklı uygulamaların gerektirdiği bilgiler de farklıdır. Yine de bu sistemlerin fiziksel yapısını, en azından kuramsal olarak tanımak her durumda faydalıdır.

Programlamayla ilgili her şeye YAZILIM, (software) fiziksel yapıyla ilgili olanlara da DONANIM (hardware) denir. Kişisel bilgisayarların donanımı zamanımızda hızla gelişmekte. Giderek daha ucuz ve daha güçlü donanım sistemleri üretildiğinden, bazen programlar bilgisayardan daha pahalı olabilmektedir.

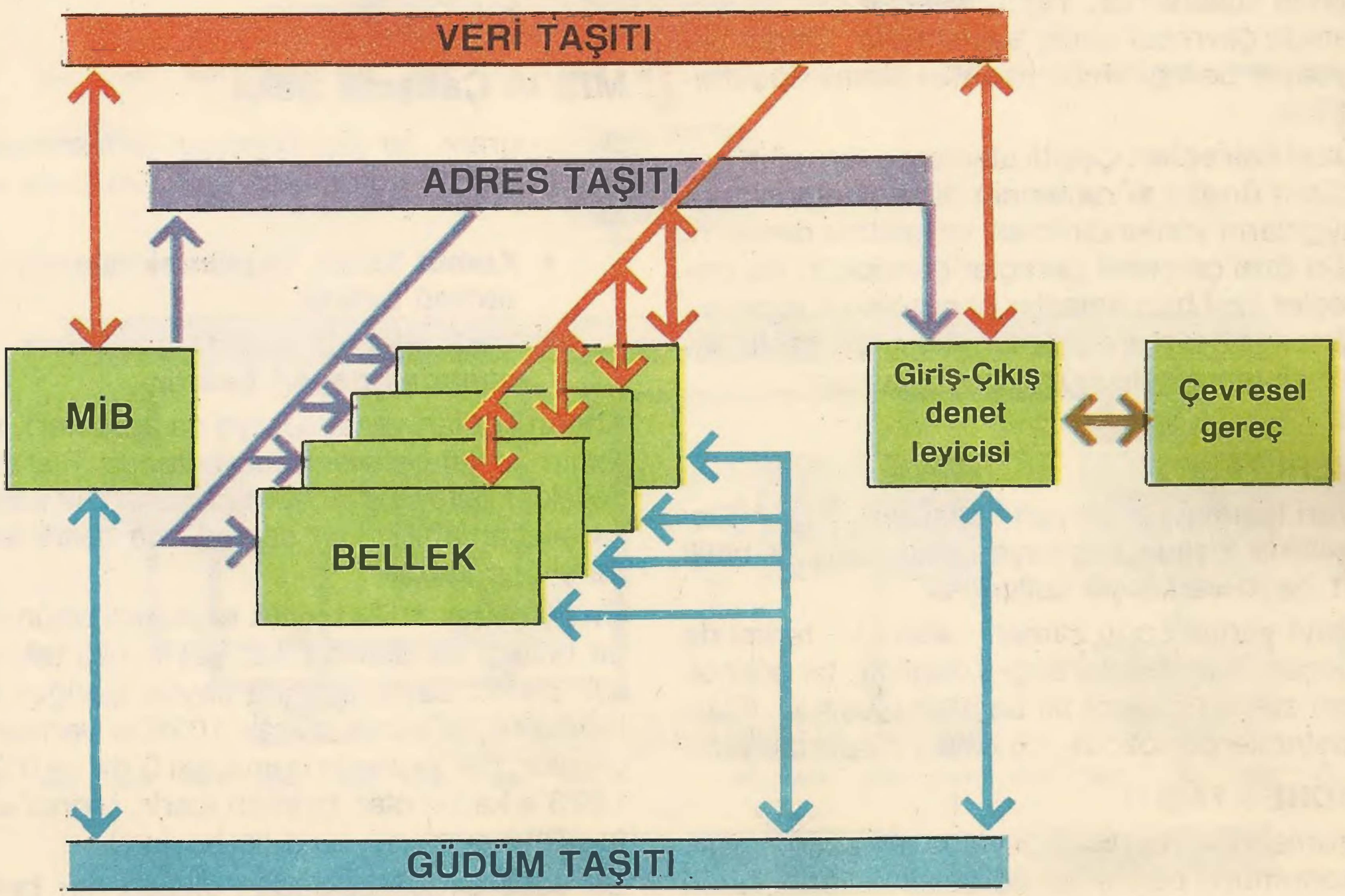
Maliyetin düşmesi ve verimliliğin artmasıyla, bilgisayarların uygulama alanları sürekli artıyor. Yakın gelecekte uygulama alanlarının sayısı birkaç katına çıkacaktır.

Minik İşlemcinin Temel Yapısı

Bir kişisel bilgisayar sisteminin başlıca ögeleri şunlardır (bkz. şekil):

- **MİB = Merkezi İşlem Birimi (MİB; Central Processing Unit -CPU)**
- **Bellek (Memory)**
- **Giriş-Çıkış Denetleyicisi (Input/Output Ports)**
- **Çevresel gereç (Peripheral devices)**
- **Veri taşıtı (Data bus)**
- **Adres taşıtı (Address bus)**
- **Güdüm taşıtı (Control bus)**

MİNİK İŞLEMCİNİN TEMEL YAPISI



MİB

Diğer tüm ögeleri denetlemek ve hesap işlemlerini yapmak gibi en önemli işlevleri yerine getirir.

BELLEK

Programları oluşturan komutların 2 tabanlı kodlarını ve bu programların işleyeceği verilerin 2 tabanlı karşılıklarını saklayan mekanizmalar bütünüdür.

GİRİŞ-ÇIKIŞ DENETLEYİCİSİ

Sistemin dış dünya ile bağlantısını sağlayan devrelerden oluşur. Çok defa ''Input/Output'' teriminden türetilmiş ''I/O denetleyicisi'' olarak da adlandırılır.

ÇEVRESEL GEREÇLER

İç devrelere dahil olmayan tüm ögeler. Başlıcaları şunlardır:

Ekran: Bilgisayarla işletici (kullanıcı) arasındaki iletişimi sağlar.

Klavye: Dış görünüşüyle bir daktilo klavyesine benzer. Veriler ve programlar bilgisayara onunla iletilir.

Bilgiyazar: Ekranınkine benzer bir işlevi olan çıkış aygıtıdır.

Disket ve Manyetik Şerit: Sistemin, büyük hacimde verilerin saklandığı yığın bellekleridir. Veriler manyetik olarak kaydedilir ve gerektiğinde kullanılırlar. Yığın bellekler gerçek anlamda çevresel gereç sayılmazlar. Onları bilgisayar belleğinin bir uzantısı olarak düşünebiliriz.

Özel Gereçler: Çeşitli alanlarda (örneğin endüstri üretim süreçlerinin düzenlenmesinde) aygıtların yönlendirilmesi ve işletme denetimi için özel çevresel gereçler gereklidir. Bu gereçler özel bazı amaçlar öngörülerek yapılmış birtakım devreler barındırdıkları için herhangi bir sisteme gelişigüzel bağlanamaz.

VERİ TAŞITI

Veri taşımayı sağlayan hatlardan oluşur. Genellikle kişisel bilgisayar sistemleri, 8 bitlik (1 bayt) veri taşıtı kullanırlar.

Bayt yerine çoğu zaman ''sözcük'' terimi de geçer. Bunun tersi doğru değildir, bir sözcük her zaman sadece bir bayttan oluşmaz. (Bazı sistemlerde sözcük, 16 bitlik - 2 bayt oluşur.)

ADRES TAŞITI

Verilerin kaydedileceği ya da alınacağı bellek konumunu belirleyen adres sinyallerini ileten devrelerdir.

GÜDÜM TAŞITI

Bir kişisel bilgisayar sisteminde mutlaka bulunması gereken devrelerden biridir. MİB'in diğer ögelere gönderdiği komutlar ve bunların yanıtları güdüm taşıtı yoluyla iletilir. (Bir kişisel bilgisayar sisteminin ayrıntılı yapısı için bkz. s.127, üstteki şekil)

MİB'in Yapısı

MİB, verilerin işlendiği ve diğer devreleri işletmek için gerekli sinyallerin üretildiği en karmaşık birimdir. MİB'in başlıca birimleri şunlardır (bkz. s.127 alttaki şekil):

İç Veri Taşıtı: İşlevi dış taşıtlarınkine benzer (s.127, üstteki şekil ile karşılaştırınız.) Veriler bu yolla iletilir.

Bellek ve Siciller: Veriler işlemden önce ve sonra bunlara kaydedilir.

Komut Kodu Çözücüsü: Bu devre program komutlarını bir dizi elektrik sinyaline dönüştürür.

Güdüm Birimi: Bu birim, komut kodu çözücüsünden gelen sinyallere göre MİB'in hem içindeki, hem de dışındaki birimleri (bellek, giriş/çıkış aygıtları gibi) yönlendiren çeşitli sinyaller üretir.

AMB = Aritmetik-Mantık Birimi (ALU = Aritmetic-Logic Unit): Aritmetik ve mantık işlemlerinin yapıldığı birimdir.

MİB'in Çalışma Şekli

Bir program, bir dizi komutun sıralanmasından oluşur. Her komutun en az iki kısmı vardır:

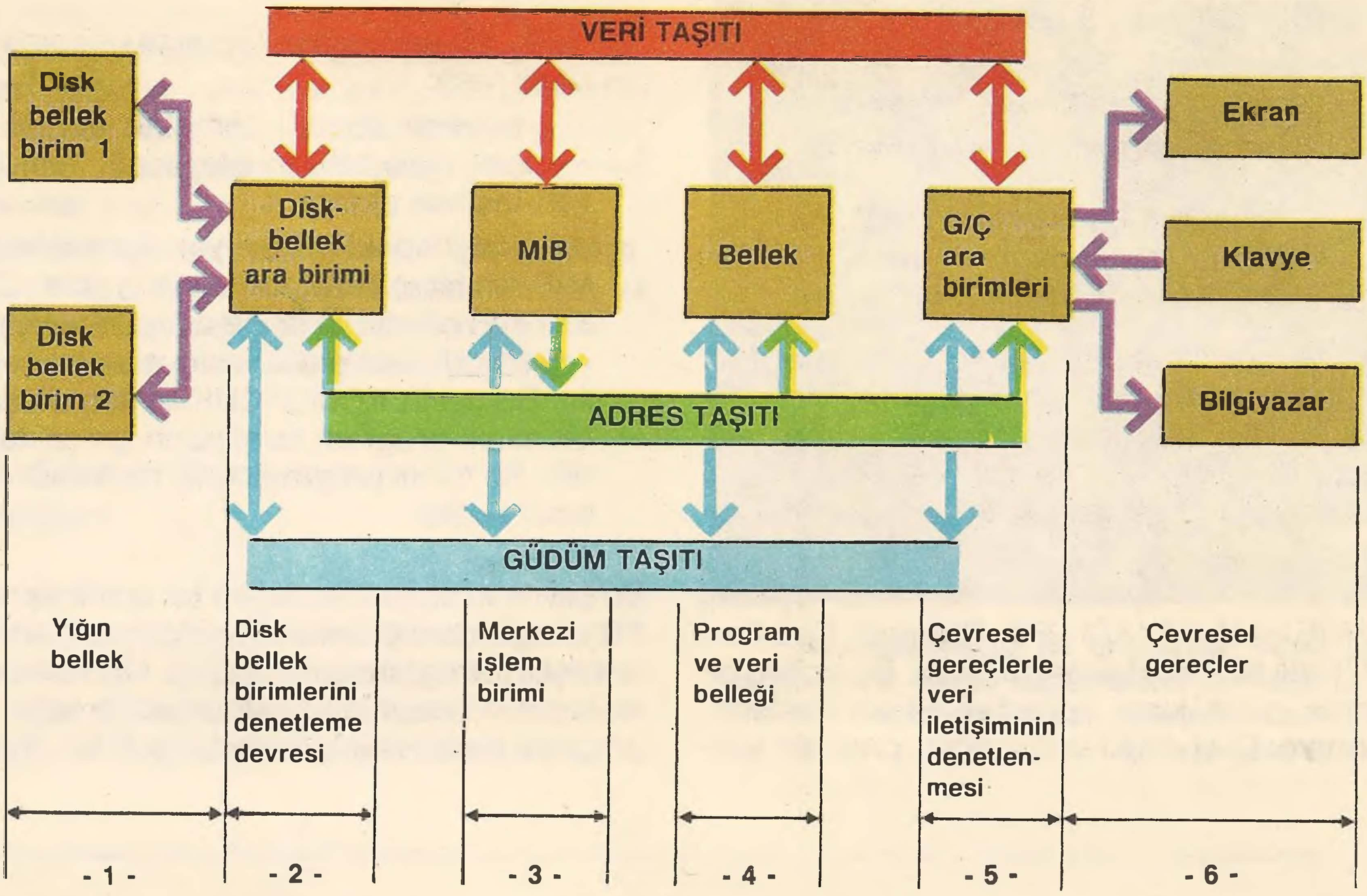
- **Komut Kodu: Yapılacak işlemin türünü belirtir.**
- **Veri: İşlem sırasında gözönüne alınacak değeri belirtir.**

Komut kodları ve veriler (ya da adresler), sistemin çeşitli belleklerinde bulunur. Sistemin bellekler bütünü sayfalardan oluşan bir kitapla karşılaştırılabilir: Her sayfada da belirli sayıda simge vardır.

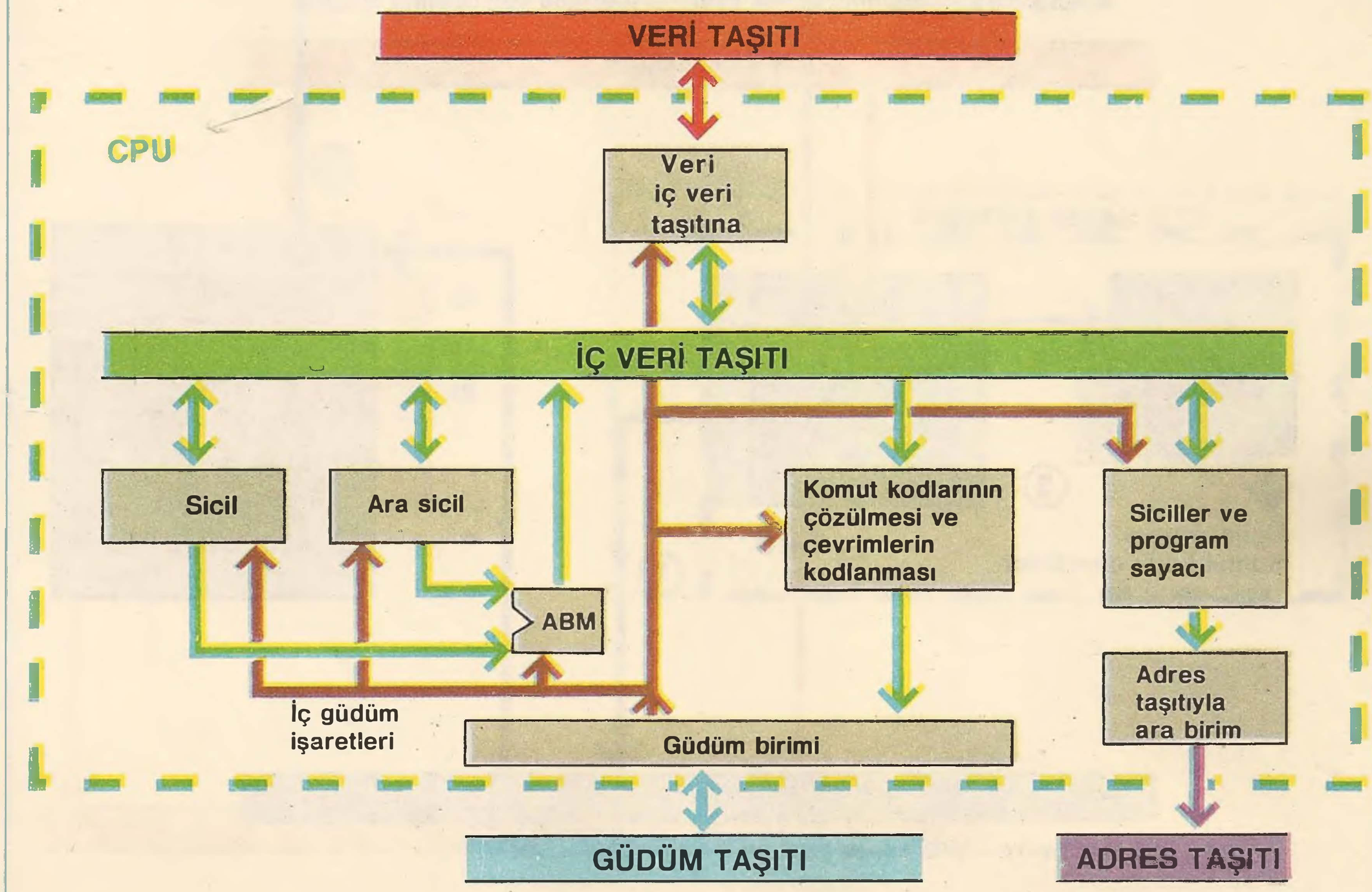
Örnek olarak 1024 baytlık sayfalara bölünmüş bir belleği ele alalım (bkz. sayfa 125'teki şekil). Birinci sayfa üçüncü baytın içeriğini öğrenmek için adres olarak 1026'yı vermemiz gerekir. (İlk sayfanın numarası 0'dır ve 0'dan 1023'e kadar olan baytları içerir, birinci sayfa 1024 numaralı bayt ile başlar.)

Bir veya birçok sayfadan oluşan her bellek bölgesinin belirli bir görevi vardır. Farklı böl-

MİNİK İŞLEMCİ SİSTEMİNİN AYRINTILI YAPISI



BİR MİB'İN TİPİK YAPISI



geleri ve uygulama amaçlarını gösteren tabloya BELLEK HARİTASI adı verilir. Bu harita, söz konusu bilgisayar sisteminin cinsine ve içerdiği programlara göre değişir.

Bir komutun uygulanması, ÇEVRİM denilen bir işlemler dizisi şeklinde gerçekleşir.

Bir çevrim genellikle şu evrelerden oluşur:

- **Komutun çağırılması (FETCH)**
- **Kodun çözülmesi ve işlemin yapılması (EXECUTE)**

Bu en basit komut biçimidir. Daha karmaşık komutlar, daha çok işlem gerektirebilir. Sistem işlemeye başlayınca veya ''BAŞLA'' komutu verilince, MİB ilk komutu alır, gereğini yapar; ikinci komuta geçer ve işleyiş bu şekilde sürer.

Komutların sırayla işlenmesi, MİB içindeki PROGRAM SAYACI (PS; Program Counter-PC) adlı sicil tarafından sağlanır. Bu sicilde bir sonra yürütülecek komutun adresi saklanır. Komutlar bellekteki sıraya göre, program içinde atlama olmaksızın işlendiğinde, PS'ye (program sayacı) 1 eklemek, bir sonraki komuta geçmek için yeterlidir.

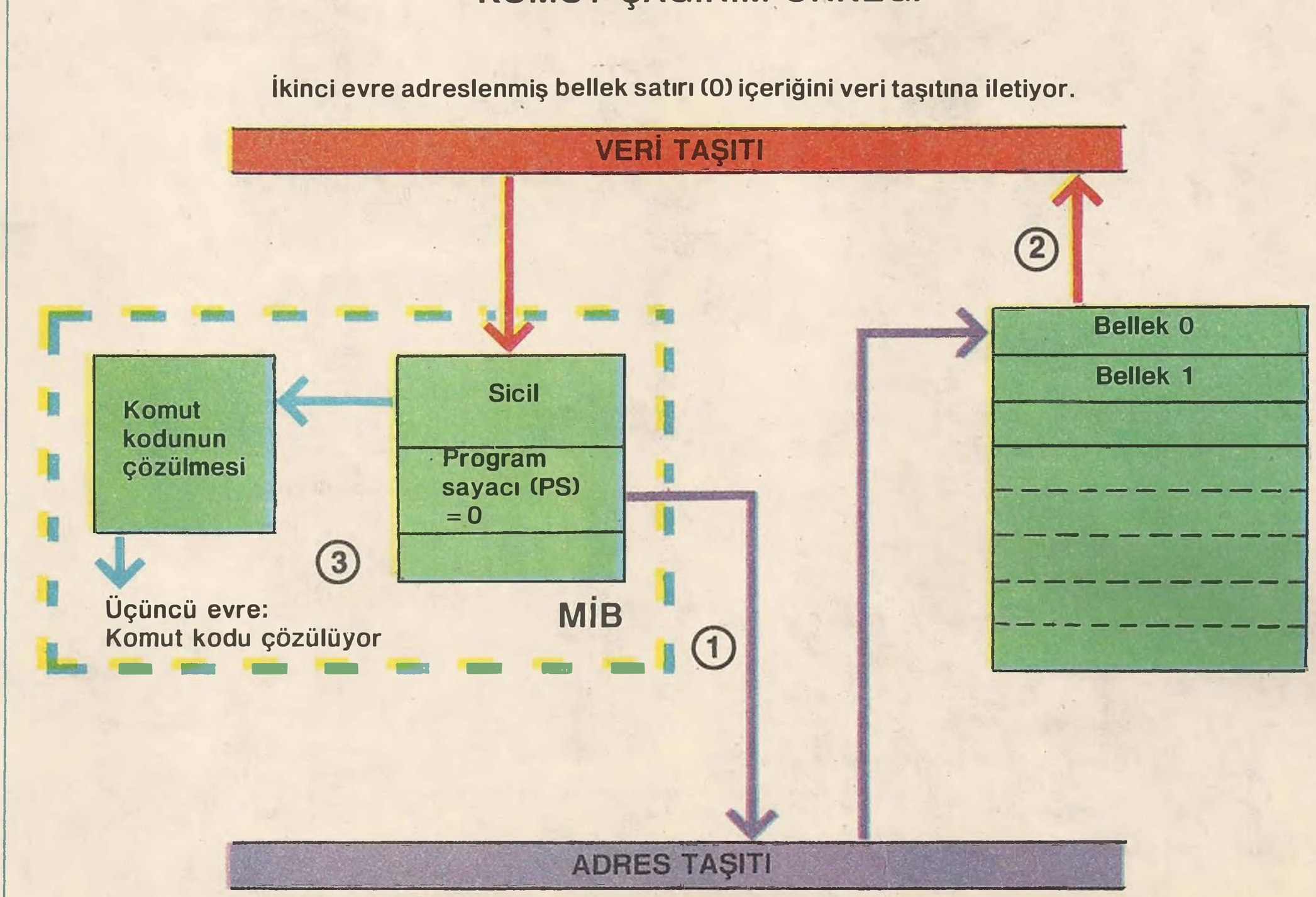
1- MİB, PS'deki değeri (örnekte 0) adres taşıtına verir.

2- Adreslenen sözcük (örnekte adresi 0) içeriğini (yani MİB'in işleyeceği komutu) veri taşıtına aktarır.

3- Veri taşıtındaki değer (yani komut kodu) MİB'nin bir iç belleğine (sicile) yazılır. Oradan kod çözme birimine aktarılır ve işlemi yaptıracak olan elektrik sinyallerine çevrilir. Bu çeviri KOMUTCUK'lardan oluşan, bir minik program tarafından gerçekleştirilir. Bu minik programı MİB'nin kendi üreticisi sağlar.

Bu çevrimin sonunda, değeri bir artırılmış olan PS'nin gönderdiği bellek sözcüğündeki komuta erişilir. Programlamada çoğu kez komutların işlenme sırasının değiştirilmesi, örneğin bir program parçasının atlanması gerekir. Bu gi-

KOMUT ÇAĞIRIM ÖRNEĞİ



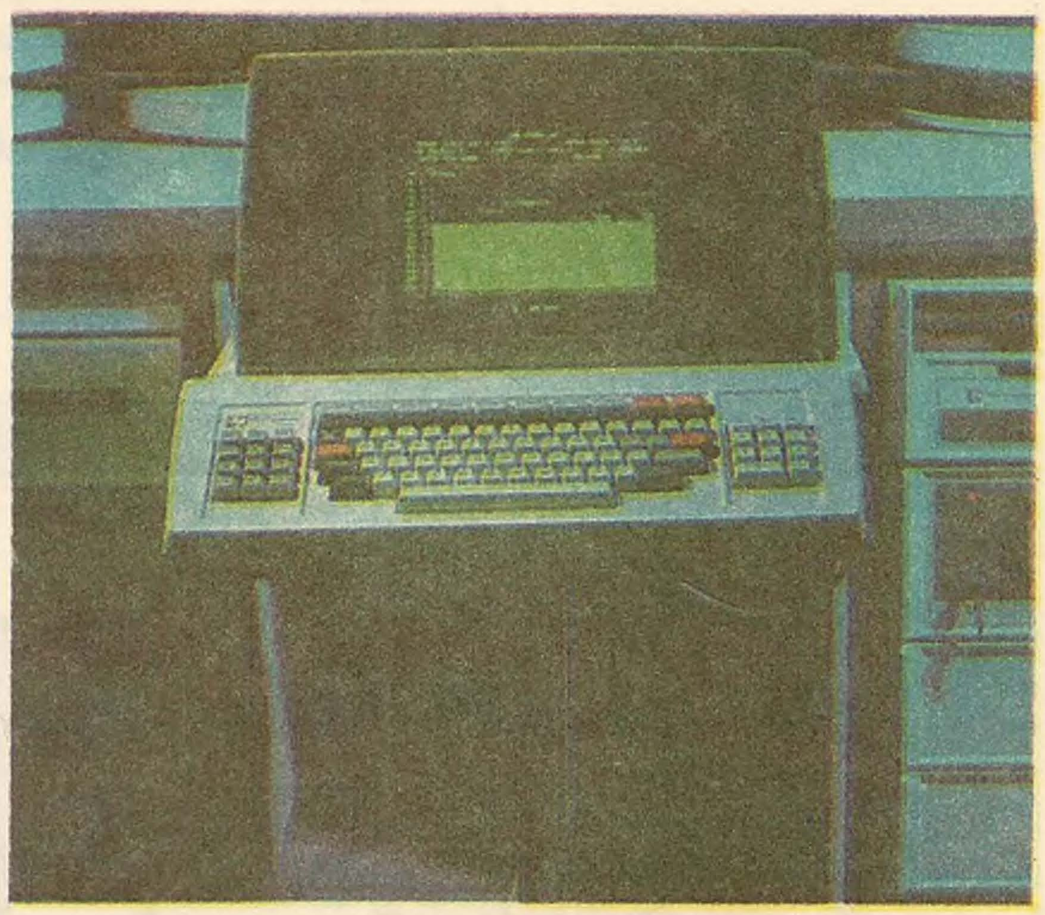
bi durumlarda program sayacının içeriğini korumak ve daha sonra yeniden PS'ye yükleyebilmek için Deste Bellek (STACK) adı verilen bir bellek bölümünden yararlanılır. Bir atlama koduna gelindiğinde PS'deki değer deste belleğe kaydedilir, atlama komutunun belirlediği yeni adres PS'ye yüklenir. Bu adresin gönderdiği komut bellekten çağırılarak, yeni adrese atlanmış olur.

Atlamanın gerektirdiği işlemler bitince MİB, PS'nin eski değerini deste bellekten alır ve programa kalındığı yerden devam edilir, tâ ki yeni bir atlama komutu gelinceye kadar.

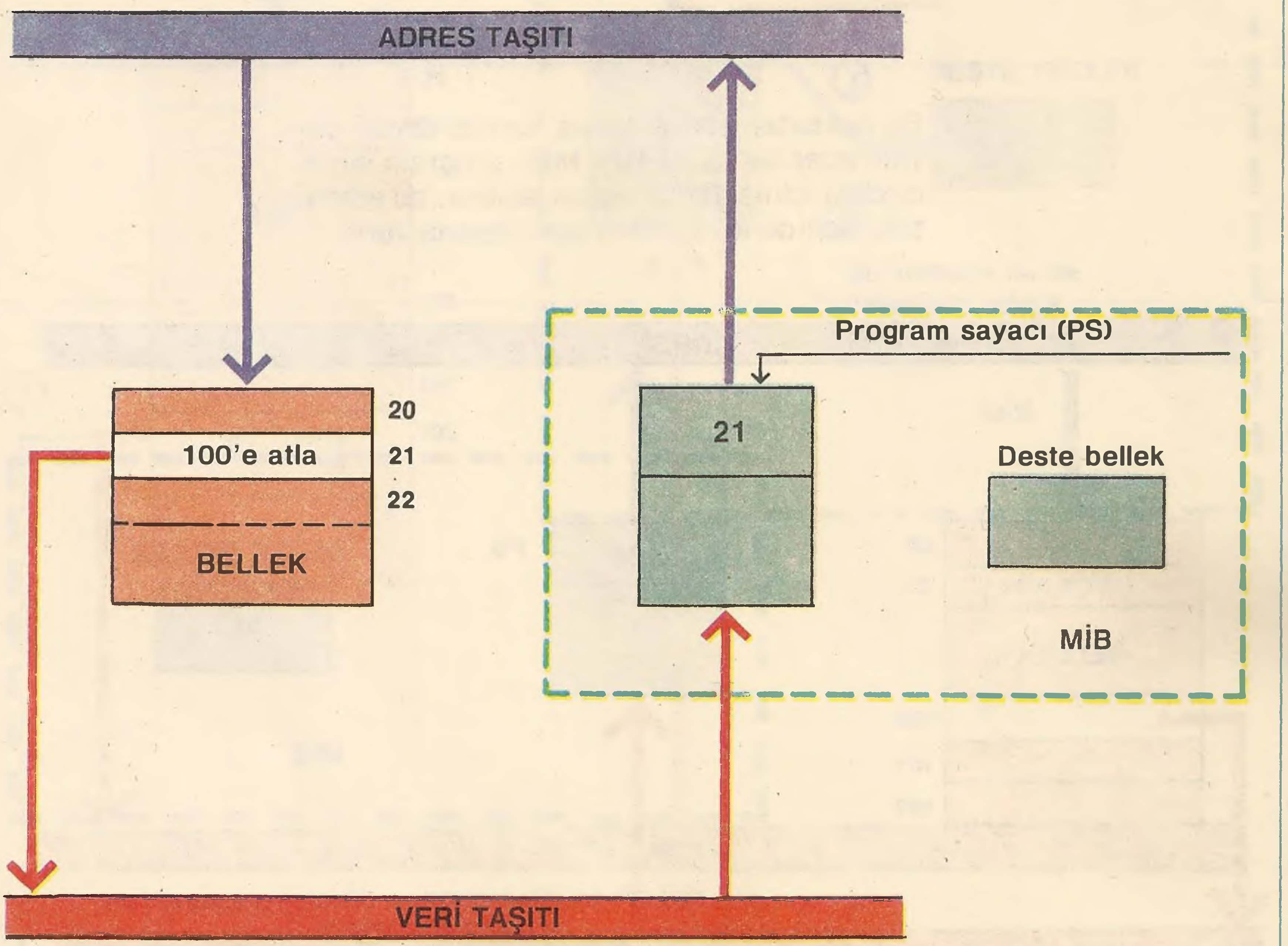
Bir Atlama Komutunun Evreleri

Birinci Evre

Bir programın yürütülmesi sırasında program sayacının 21 değerine ulaştığını varsayalım. Bu sayı, adres taşıtına verilir ve 21 numaralı bellek, sicili sözgelimi ''100'e atla'' komutunun kodunu veri taşıtına aktarır. Bu anda deste bellekte henüz bir değişme yoktur.

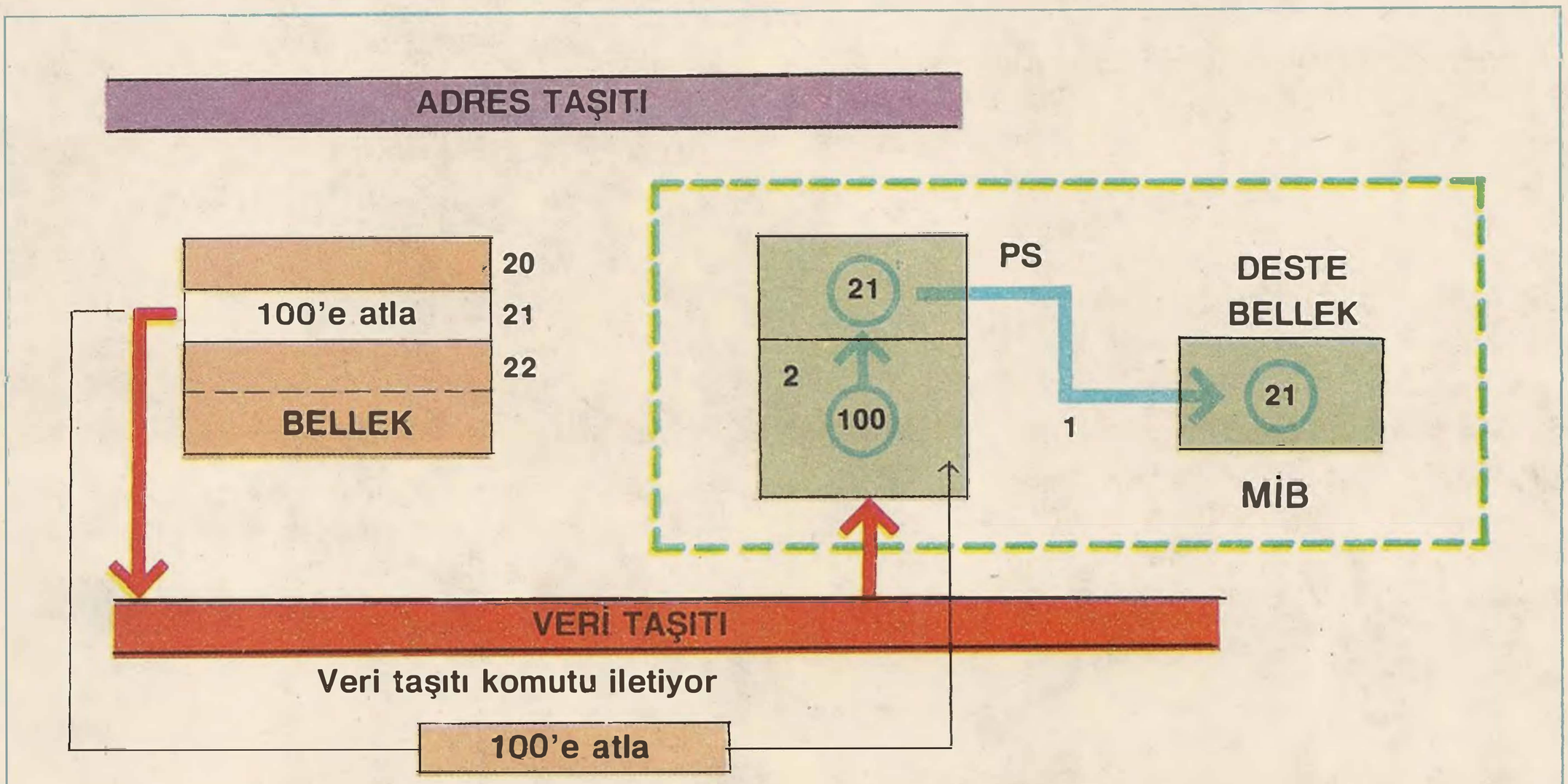


Resimde kişisel bir bilgisayar, böyle bir bilgisayarın bellek kapasitesini artırmak için kullanılabilecek manyetik disklerle birlikte görülüyor.

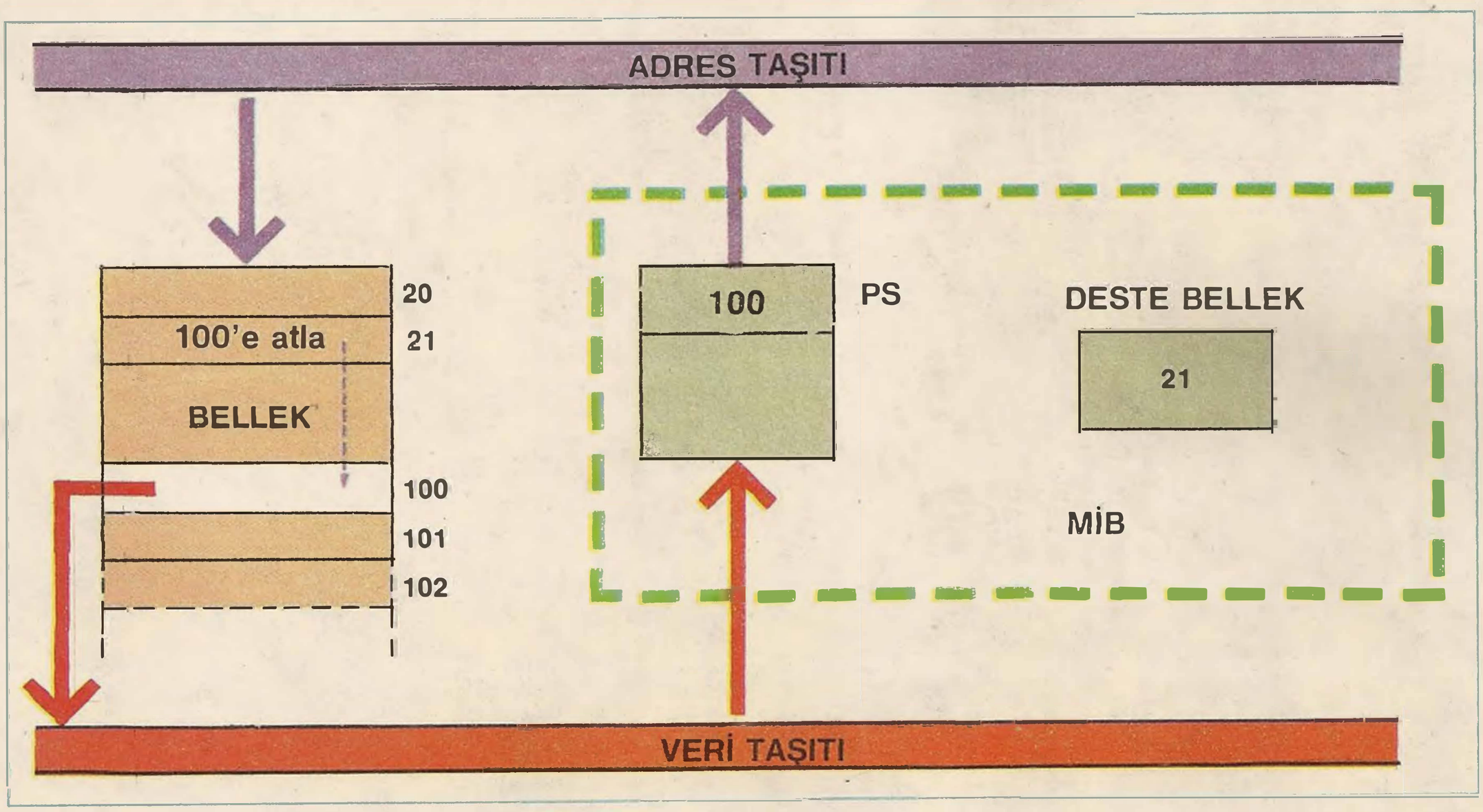


İkinci Evre

MİB, kodun bir atlama komutu olduğunu belirleyince, içindeki bilgiyi (yani atlamadan önceki son komutun adresini) DESTE BELLEK 1'e, yeni adresi de (100), PS 2'ye yazar. 100 sayısı şimdi işlenecek komutun adresidir (bkz. şekil).



Bu noktadan sonra işleyiş normal olarak devam eder (bkz. şekil). MİB, program sayacındaki adresi (100), adres taşıtına, bu adresteki sicil de içindekileri veri taşıtına verir.



Bu yeni komutun gereği yapıldıktan sonra MİB 101 numaralı adresteki komuta geçer ve ''DÖN'' (RETURN) komutuna varıncaya kadar işlemler böyle sürer. RETURN komutuyla birlikte MİB, deste bellekte saklanmış olan adresi alıp program sayacına yeniden yazar. Sonraki komutlar yine bir atlama komutuna raslanıncaya dek sırayla işlenir.

Bunu örnek üzerinde görelim (bkz.şekil). 100 numaralı adrese atlandıktan sonra sırasıyla 101, 102 ve 103 numaralı adreslerdeki komutlar çağırılır. 103'de RETURN komutu bulunduğundan MİB deste bellekteki adresi PS'ye geri getirir. Program 22 adresli komuttan devam eder.

Sicil 101'de de bir atlama komutu olsaydı, geri dönüş için iki adresin kaydedilmesi gerekirdi, biri 21'deki biri de 101'deki atlamalar için. Birinci dönüş 102'ye, ikinci dönüş ise 22'ye yapılırdı. Böyle bir durumda deste belleğin iki sicilden oluşması gerekirdi.

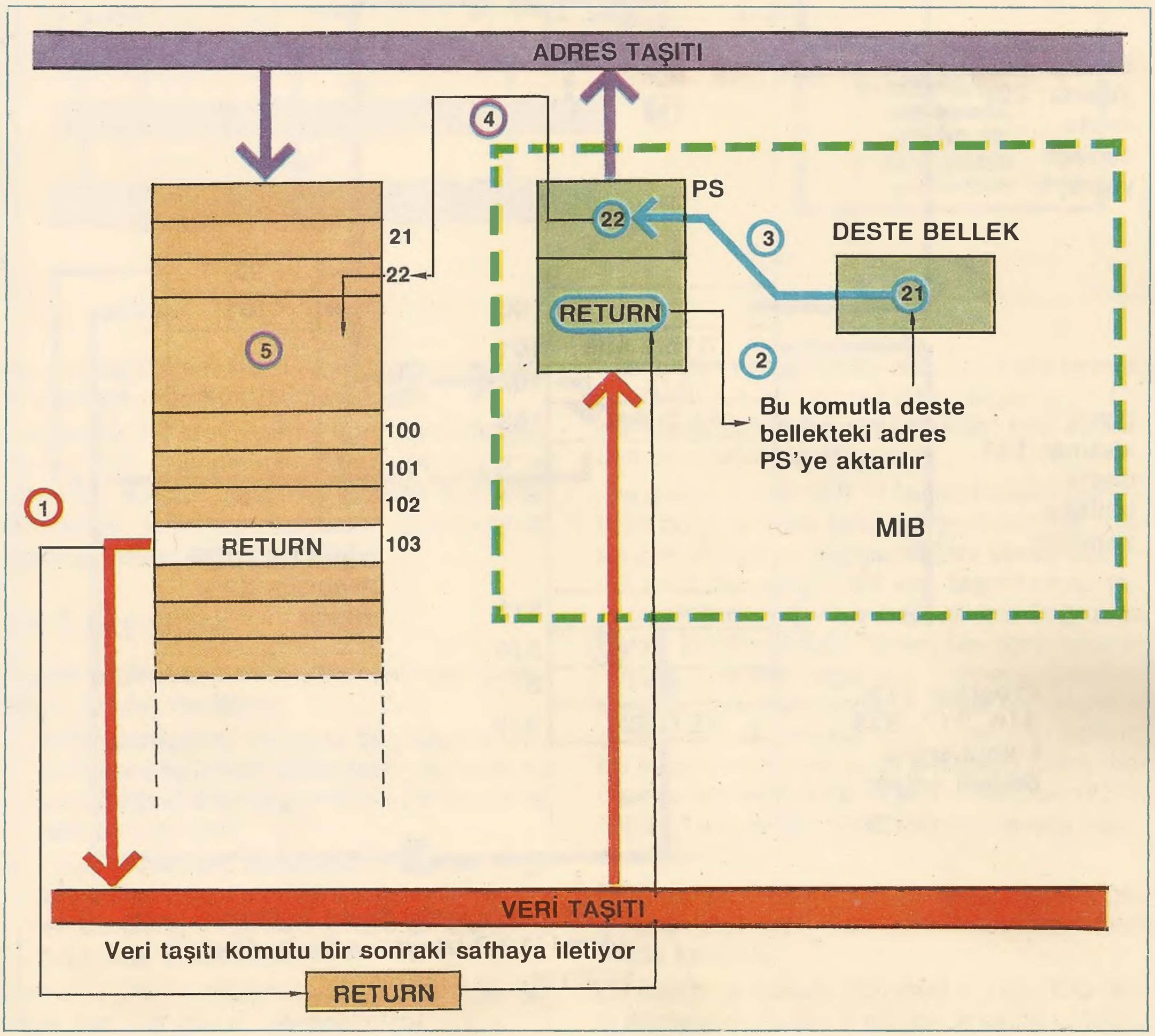
Bazı minik işlemcilerde deste bellek MİB'in içindedir. Bunun sağladığı birleştirme olanakları önemliyse de, birbirini izleyen atlamalar yapma olanağı kısıtlıdır.

Deste bellek, ana bellekte de yer alabilir, bu durumda sınırları ana belleğin sınırlarına eşit olur.

Bu sayfadaki şekilde birbirini izleyen iki atlamanın nasıl yürütüldüğü gösteriliyor. İzlenen adımların sırası 1'den 10'a kadar sayılarla belirtilmiştir.

Bir Programa Nasıl Ara Verilir

Normal program akışı ''ARAYA GİR'' (INTERRUPT) sinyaliyle kesilebilir. Bu arada başka

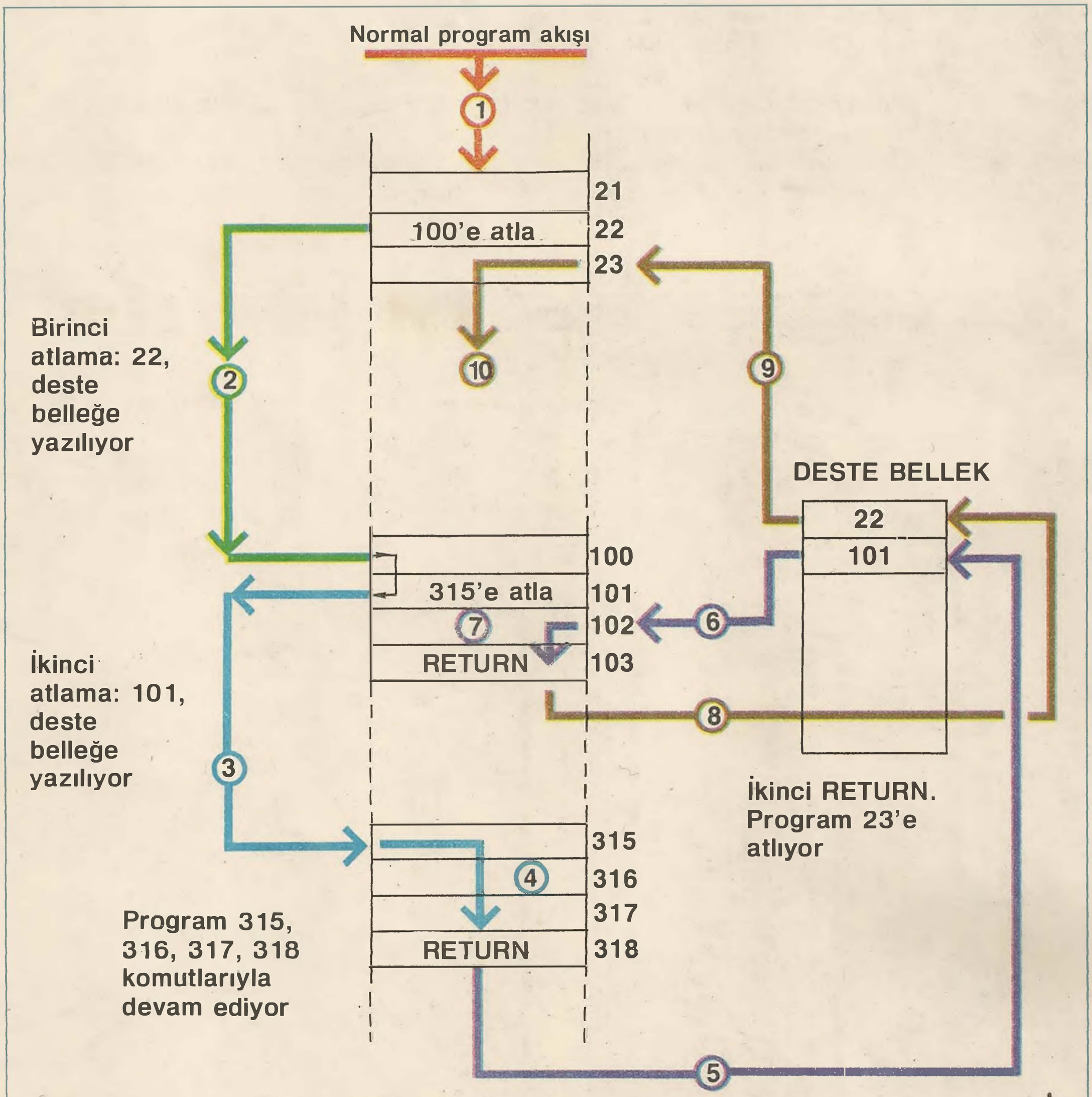


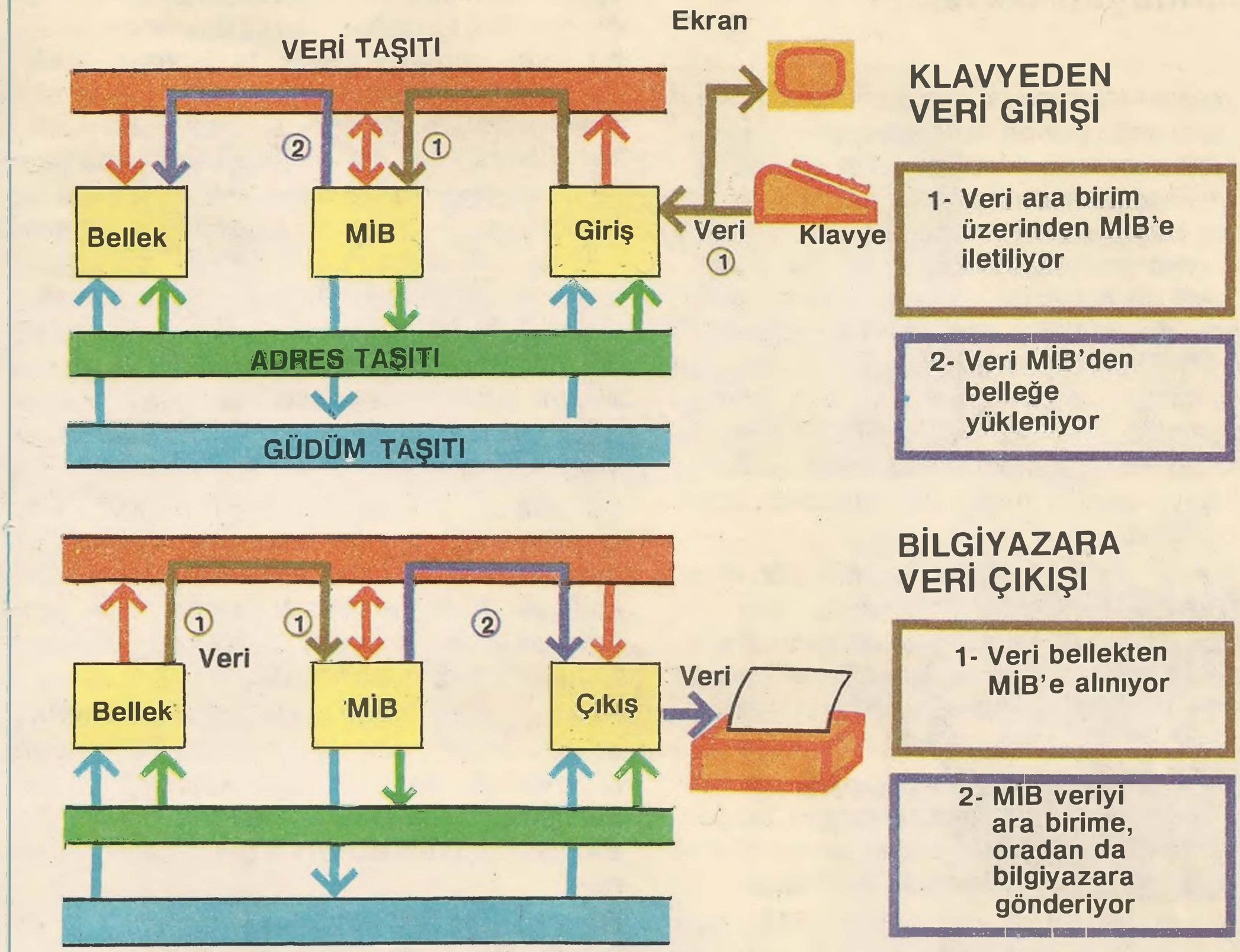
bir işlev yürütüldükten sonra programın durdurulduğu noktaya geri dönülebilir.

Kişisel bilgisayar sistemlerinde çeşitli çevresel gereçlerin araya girme öncelikleri farklıdır. Normal olarak komutları gönderen klavye ilk sırayı alır. Denetim klavyede olmadığı sürece diğer çevresel gereçler çalışabilir.

Araya girme işlemi programda öngörülmemiş olan ve dışarıdan gelen talepleri karşılamak için kullanılır.

Tipik bir örnek, basılmakta olan bir programa verilerde ya da sütun düzeninde bir yanlışlık olduğu için, ara verilmesidir. Böyle bir bilgiyazma işlemini yürütecek program sistemin tüm parçalarını kullanır: Verileri bellekten alır, kodlar ve bilgiyazara iletir. Programda veri girişinin öngörülmediğini ve dolayısıyla güdümün, klavyenin etkisinde olmadığını varsayalım. Bu durumda MİB bilgiyazma işlemini tamamen bitirip klavyeden yeniden komut veri-





lip verilmediğini kontrol etmeden, komut girişi yapmak mümkün olamayacaktır.

Klavyeden bir araya girme komutu verildiğinde ise MİB, kullanıcı programını durdurur ve güdümü istenen çevresel gerece terkeder. Böylelikle, kullanıcı bilgiyazma programında istediği değişiklikleri yapabilir.

Veri Alışverişi

Kişisel bilgisayar sistemlerinin çevresel gereçler iki gruba ayrılabilir:

- **Giriş gereçleri:** Verilerin bilgisayara yüklenmesini sağlayan gereçlerdir; bu işlev, kişisel bilgisayarlarda genellikle bir klavye tarafından görülür.
- **Çıkış gereçleri:** Bilgisayarın istenen bilgileri kullanıcıya iletmesini sağlayan gereçlerdir. Bunun için normal olarak ekran ve bilgiyazar kullanılır.

Verilerin giriş ve çıkışını sağlayan gereçler şekilde şematik olarak gösterilmiştir.

Klavyeden veri girişinde veri, önce ara birime, sonra MİB'e, son olarak da belleğe ulaşır. Çıkışı istendiğinde ise veri bellekten MİB'e, oradan da arabirime aktarılır.

Ara durak olarak MİB'in kullanılmasını gerektiren bu giriş-çıkış şekli çok hızlı değildir, daha çok klavye ve bilgiyazar gibi yavaş çevresel gereçler arasındaki veri taşınmasına uygundur. Daha hızlı veri değişimi için Doğrudan Bellek Erişimi (DBE) -Direct Memory Access (DMA)- yöntemi uygulanır. Böylece çevresel gereçle bellek arasında dolaysız veri alışverişi sağlanır. Bu olayda MİB'in bir rolü yoktur, bu nedenle veri alışverişi bitinceye dek çalışmasına ara verir. MİB'in bu durumuna HAZIR (HOLD) adı verilir. (DBE için bkz. sayfa 138'-deki şekil.)

DBE yöntemi büyük miktarda veri taşıma gerektiren manyetik disk birimlerine veri aktarmada kullanılır.

Bir disket genellikle 250.000-1.200.000, bazı modellerinde 5-10 milyon arasında simge

Unutmayan Bellekler

En yetenekli bilgisayarın merkezi işlem birimi bile aynı anda birden fazla iş yapamaz. Bir işlemi bitirir, sonucu kaydeder, ikinci işleme geçer ve ikinci sonucu başka bir yere kaydeder. Bazen birinci sonuca, ancak arada binlerce farklı işlem yaptıktan sonra döner. Bu yüzden her bilgi, gerektiğinde hemen erişilebilir bir konumda bulunmalıdır. Yani bellek yapısı öyle olmalıdır ki, bilgiler, defterlerdeki gibi saatlerce sayfa karıştırma gerekmeksizin, anında okunup yazılabilsin. YAZ-OKU BELLEK (RAM = Random Access Memory) içerdiği herhangi bir bilgiye istediği anda hızla ulaşabilen böyle bir bellek türüdür.

Bilgisayar, daha önce gördüğümüz gibi, iki tabanlı sayı sistemine çevrilmiş sayıları işler. Verileri oluşturan bu sayılar seri adresleme yolu ile belli bir sıra izleyerek teker teker belleğe gönderilir, ya da paralel adresleme yoluyla bütün veriler aynı anda belleğe iletilebilir.

Bellek, tek bir silisyum yonga içinde yer alan transistör sıraları ve sütunlarından oluşur. Yongada ayrıca, güdüm, seçim ve veri arama gibi işlemleri yapan devreler de bulunur.

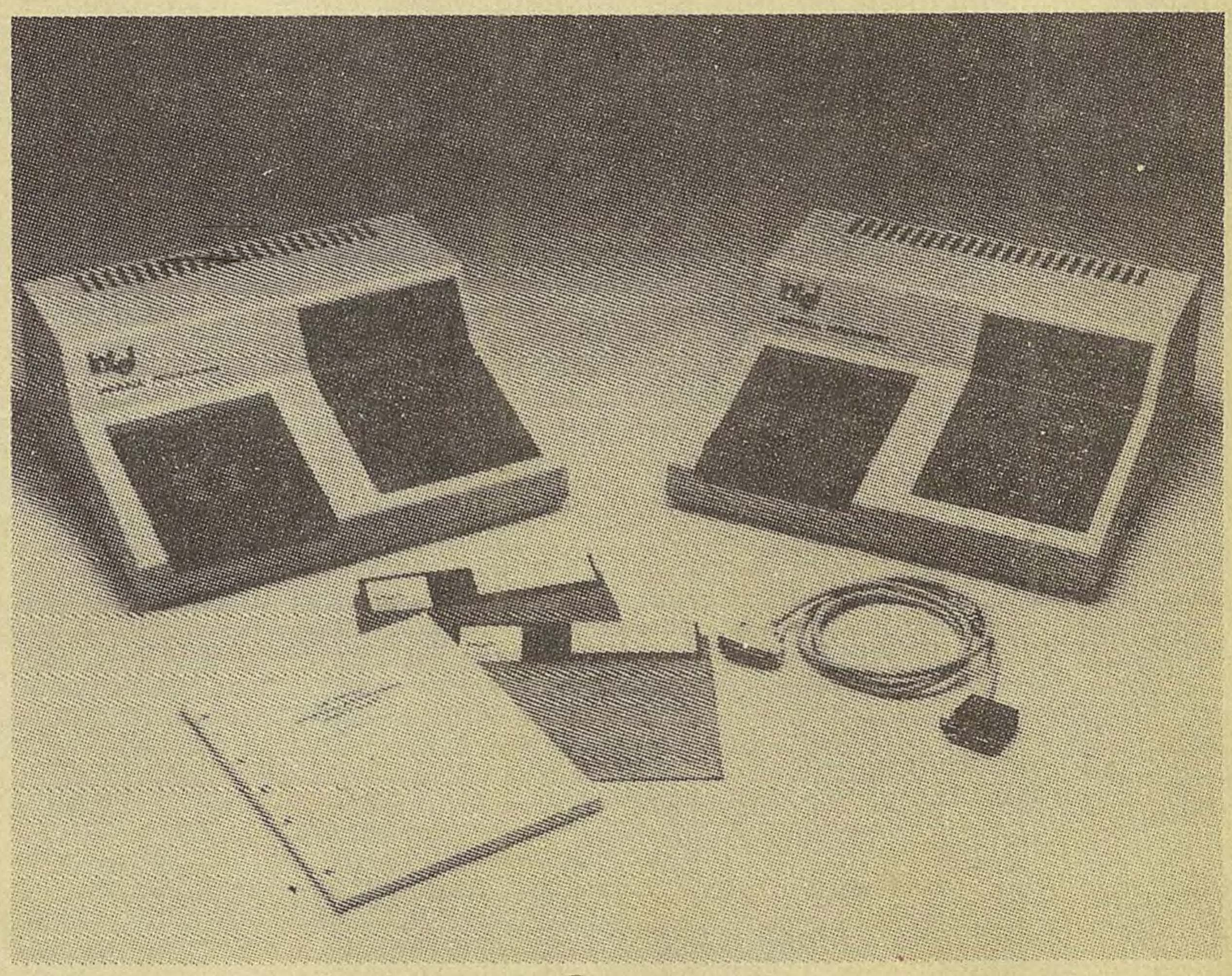
İki tür yarı-iletken yaz-oku belleği (RAM) türü vardır: Statik ve dinamik bellek. Dinamik bellekte bilgi, belirli bir süre geçtikten sonra kendiliğinden silinir. Bu tür belleklerde veriyi oluşturan 0 ve 1 rakamları çok küçük bir elektrik yükü olarak saklanır; kondansatörlerde olduğu gibi direnç değerinin küçük olması nedeniyle yük zamanla boşalır. Bu yüzden, saklanacak bilginin zamanlama sinyalleri ile sürekli olarak yeniden yüklenmesi gerekir. Bu yükleme işlemi için ek bir devre gerekir, ama yine de dinamik devre birimleri, her bit için en az iki transistör gerektiren statik bellek birimlerine kıyasla çok daha küçüktür.

Statik devrede ise veri ikili bir devrede (flip flop) *(ya da daha dakik söylemek gerekirse çift-kararlı bir mültivibratör) saklanır. İkili devre elektrik yükleri gerektirmez, devrelerin açık ya da kapalı olması sinyalin 0 ya da 1 olmasına karşılık gelir.*

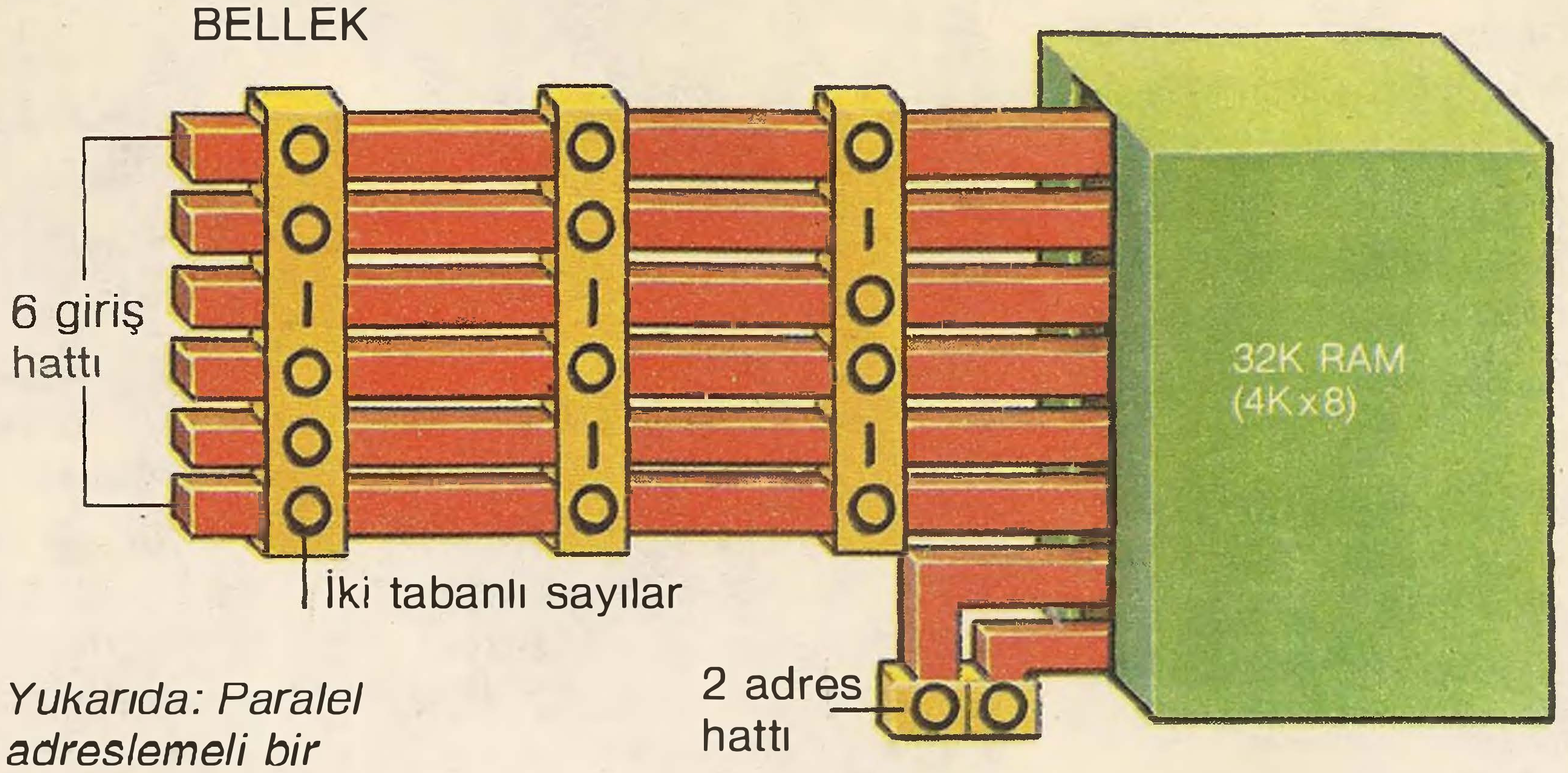
Belleklerde en çok MOSFET (Metal oxide semi-conductor with field effect transistör -alan etkili metal oksit yarı-iletken transistör) kullanılır. Fotoğraf tekniği sayesinde bir santimetrekarelik silisyum parçasına binlerce transistör sığdırılabilmekte.

Her transistör, bir ucundan diğerine elektrik akımını ileten mikroskopik boyda yarı-iletken bir kristaldir. Akaç (drain) ve kaynak (source) denilen uçlar arasındaki elektrik akımını düzenleyen üçüncü elektroda geçit (gate) adı verilir.

Geçit gerilimi belli bir değere ulaşınca, kaynak ve akaç arasında oluşan akım bir kondansatörü doldurur. Bu kondansatörün dolu olması 1'e, boş olması 0'a karşılıktır. İki tabanlı kodlanmış bilgi böylelikle depo edilmiş olur.

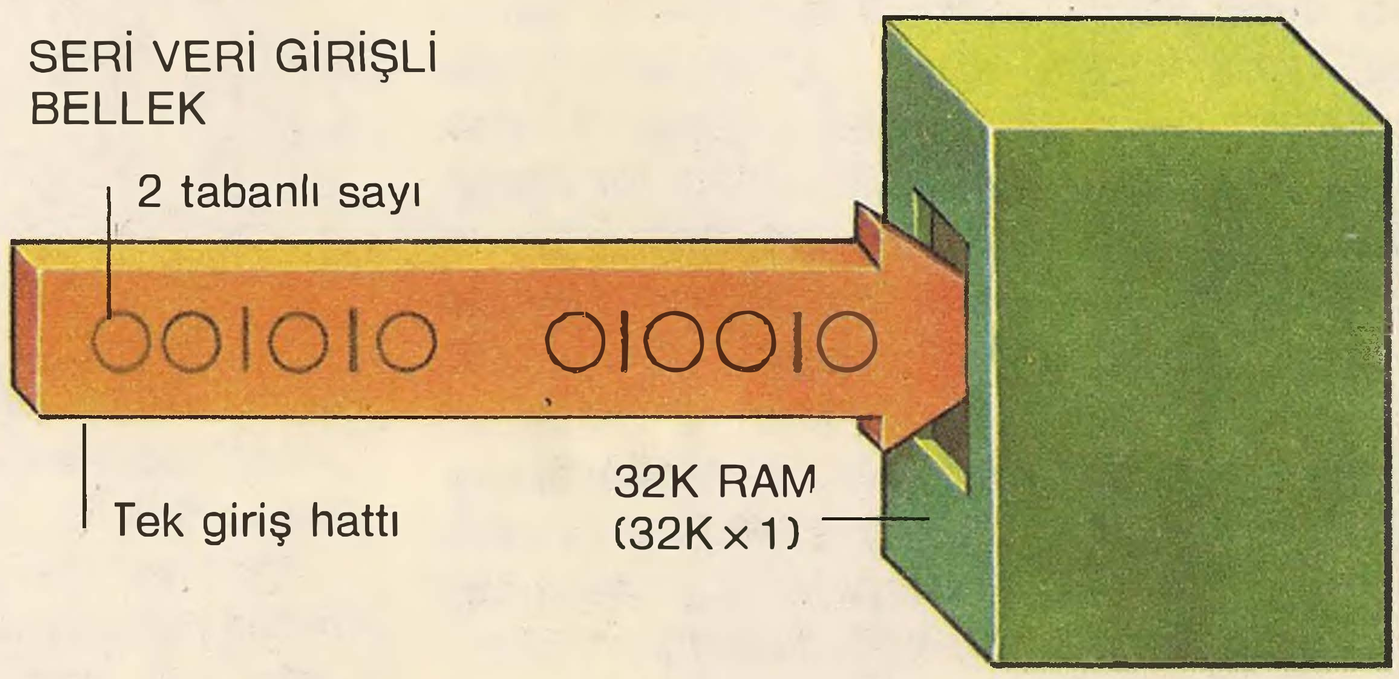


Yonga, plastik bir kılıfa yerleştirildikten sonra programlanabilir. Resimde gördüğümüz cihaz hem programlanabilir salt okunur bellekleri (PROM), hem silinebilir programlanabilir salt okunur bellekleri (EPROM) programlayabilir.



Yukarıda: Paralel adreslemeli bir bellek, bir verinin iki tabanlı rakamlarını birçok paralel giriş hattından aynı anda alıyor. Verinin yazılacağı yeri gösteren kodlanmış adres, iki ek hatla bilgisayara giriyor.

Altta: Seri adreslemeli bir bellekte tüm veriler gerekli adreslerle birlikte bir tek hattan girer. Bir mantık devresi onları ayırır.



MOSFET'in kondansatörle birlikte oluşturduğu birime bellek hücresi denir. Her hücrenin geçidi bir kondansatöre bağlıdır. Bir rakam yüklenmek istendiğinde, bu hat harekete geçer.

Seri veri girişi olan bir belleğe her sayı sırayla girer. Her iki tabanlı sayıya, verinin hangi satıra gideceğini gösteren bir adres eşlik eder. Bir bellek satırı, çoğunlukla 1, 2, 4 ya da 8 bitten oluşur. Bir mantık devresi verinin doğru adrese ulaşmasını sağlar.

Veri girişinin paralel olarak yapıldığı bir bellekte ise dört bitlik bir sayı, dört hattan aynı anda gönderilir. Hücrenin adres hattında 1 sinyali bulunuyorsa, sayı bu hücreye yazılır. Bu teknikle tek bir yonganın içinde çok sayıda bilgi saklanabilir. Dinamik yaz-oku bellekler (RAM) yonga başına 65.536 hücrelik (64K) bir kapasiteye erişmişlerdir, statik yaz-oku bellekler de ise bu sayı sadece 16K'dir.

Normal statik MOSFET RAM belleklerde direnç kullanılmaz, bunun yerine küçük bir alana doğrusal olmayan bir direnç gösteren transistörler yerleştirilir.

Bununla birlikte Mostek firması, kısa süre önce direnç kullanmaya tekrar başlamıştır. Bu yeni yöntemde, hâlâ yaygınlığını koruyan eski sisteme kıyasla daha fazla hacim kaplayan, iyonik olarak aşılanmış polimorf silisyum kullanılıyor.

Bütün yaz-oku bellekler (RAM) MOS (yarıiletken metal oksit - Metal Oxide semiconductor) teknolojisine dayanmaz. İkili devre çiftlerinden oluşan iki kutuplu bellekler de vardır.

Bunlar statik MOS belleklere benzer; yalnız her hücre için gereken yüzey daha büyüktür. Çift kutuplu bellekler dinamik belleklere kıyasla daha hızlı çalışırlar, ancak dinamik bellekler daha ekonomiktir.

Statik MOS yaz-oku belleklerin (RAM) bir değişik türü CMOS (complemetary MOS - tümleyici MOS) belleklerdir. Bu tür belleklerde, ikili devrelerdeki, transistörlerin ürettiği yükleme direnci kullanılmaz. Çapraz bağlantılı transistörlerin yerini, aktarıcı transistörlere göre ters kutuplu bağlanmış olan transistörler almıştır. Bu devrelerde yükleme transistörlerinin direnci sonsuza yaklaşır, bu nedenle akım geçmez

ve enerji kaybolmaz. CMOS RAM bellekler diğer bütün belleklerden daha az enerji harcar. Yavaş çalışmalarına karşın çok özel bazı amaçlar için gereklidirler. CMOS belleklerde akımın kuramsal olarak sıfır olması gerekir, ama yine de ikili devrenin durum değiştirdiği çok kısa süre içinde bir miktar akım geçer. Yakınlarda RCA firması, saçılma kapasitesinin yol açtığı bu sakıncanın üstesinden gelmeyi başardı.

Bu teknik sayesinde çalışma hızı dört kat artmıştır, ama kullanılan safir, silisyumdan çok daha pahalıdır. Dolayısıyla, araştırmalar sürmekle birlikte bu tekniğin geleceği pek parlak görünmüyor.

Tüm yarı iletken RAM belleklerin temel sakıncası, yazılanların ''uçucu'' olmasıdır, besleme geriliminin kesilmesiyle yazılanlar kaybolur. Buna karşı alınacak bazı önlemler vardır. Örneğin en önemli bilgiler, bir manyetik belleğe kaydedilirse, bellekteki bilgilerin sürekli olarak yeniden yüklenmesi gerekmez.

''Uçucu'' olmayan yarı-iletken bellekler de vardır. Bu tür bellekler, oyun programları, işletim programları, genel birtakım matematiksel işlemlerin nasıl yapılacağını gösteren programlar gibi uzun süre kullanılacak programların saklanmasında kullanılırlar. Bilgisayar bu bellekteki verileri kullanıcının aracılığı olmaksızın değerlendirip uygular. Değiştirilemeyecek ve sürekli kullanılan bilgileri sadece okunmak üzere sakladıklarından salt okunur bellek (ROM = Read Only Memory) adını alırlar.

Basit bir diyot matrisinden oluşan salt okunur bellekler (ROM) diyotların bir kereye mahsus olmak üzere yüksek gerilimle yakılmasıyla programlanır. Çalışma ilkeleri bu denli basit olduğundan ucuzdurlar ve verileri yoğun bir biçimde saklayabilirler.

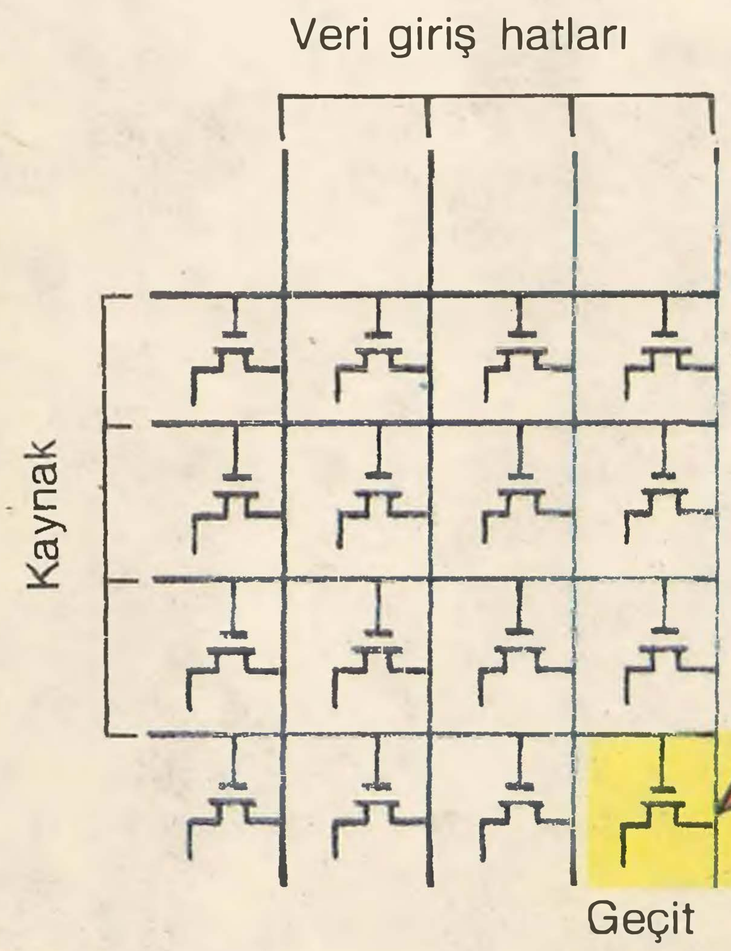
İlk salt okunur bellekler (ROM) bir kez programlandıktan sonra yeniden programlanamıyordu. Daha sonra diyot matrisine küçük bir pencereden mor-ötesi ışık vermekle silinebilen bellekler çıktı. Böylece programlanabilir salt okunur bellekler (PROM - Programmable Read Only Memory) ve silinebilir programlanabilir salt okunur bellek (EPROM - Erasable Programmable Read Only Memory) ortaya çıktı.

EPROM'lar 65.536 bitlik bir kapasiteye ulaştılar ve 128.000 - 256.000 bit hedefi de yakın görünüyor.

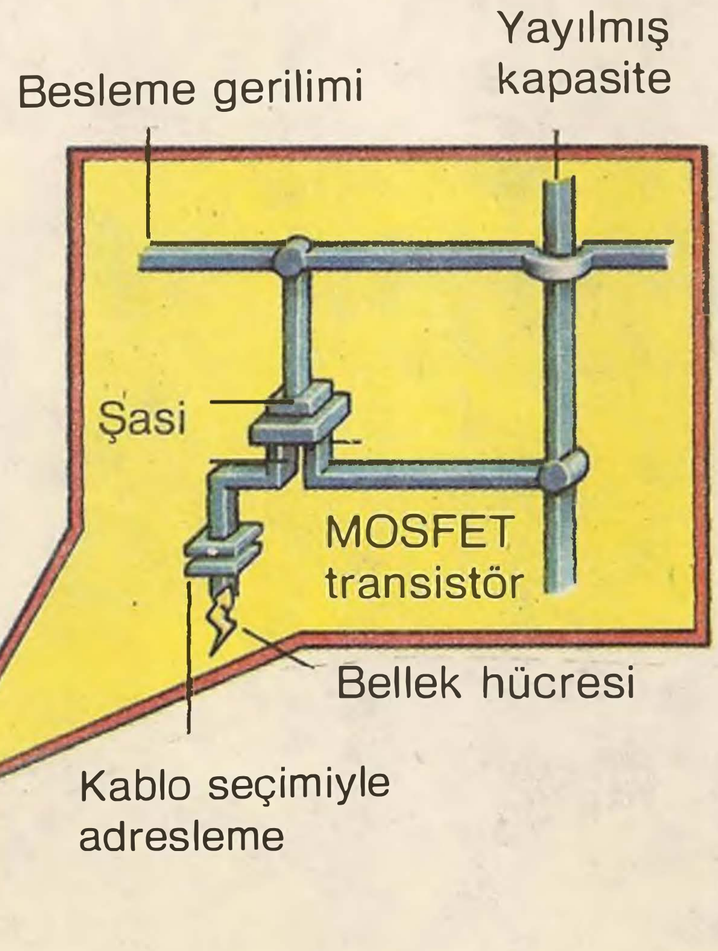
Salt okunur belleklerin (ROM) bir çeşidi de elektrikle değiştirilebilir salt okunur bellekler-dir (EAROM = Electrically Alterable Read Only Memory). Bu belleklerin bir kısmını yeniden programlamak için bütününü silmek gerekmez; normalin üstünde bir gerilim, uygulandığı hücreyi yeniden programlamış olur.

Sayısal veri saklamaya uygun başka sistemler de vardır, ancak bunlar henüz piyasada ye-

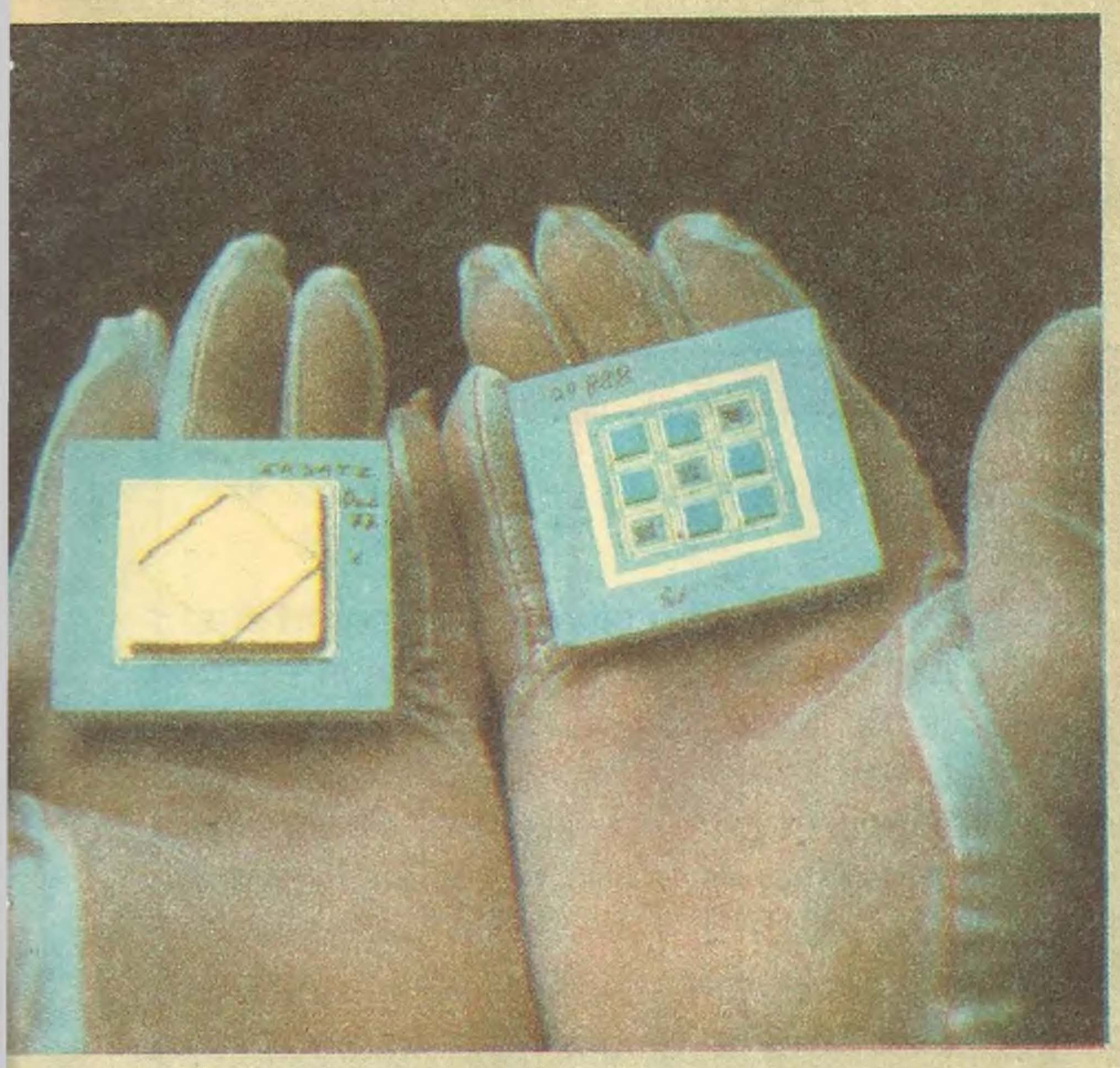
DİNAMİK RAM



BELLEK HÜCRESİ



Dinamik bir yaz-oku belleğin (RAM) yapısı. Bu bellek binlerce MOSFET adı verilen alan etkili yarı-iletken metal oksit transistör içerir. MOSFET transistörler hücre kapasitesinin gelen veriyi almasına izin veren bir anahtar işlevi görür.



Seramik taban üzerinde bellek yongası üretiminde iki evre: Sağda taşıyıcı üstünde olan yongalar, solda yalıtılıp kapatılmış. Birim bu haliyle hazırdır ve basılı devreye yerleştirilebilir.

terince denenmemiştir. En önemlileri manyetik kabarcıklı (Bubble) bellekler ve elektrik yükü akışına dayanan belleklerdir (CCD = Charge Coupled Devices).

Manyetik kabarcıklı bellekte bilgi, bazı kristallerin manyetik alan etkisi altında mıknatıslı kabarcıklara benzemesi özelliğinden yararlanılarak saklanır. Mıknatısların kutupları 1 ve 0 kavramlarına benzer, yani ara dereceleri yoktur. Dolayısıyla saklanacak sayının 0 ya da 1 olmasına göre bir kabarcık oluşturulur ya da oluşturulmaz. Bu yöntemle çok sayıda verinin saklanması mümkün olur. Yöntemin doğru sonuç vermesi için verilerin tamamıyla kapalı manyetik bobine benzer bir ortamda saklanması gerekir. Dolayısıyla bu bilgiyi doğru bir biçimde ''yazacak'' ve ''okuyacak'' ögelere (örneğin bir teyp kafasına) ihtiyaç vardır. Bilgi manyetik ögenin devirli dönmesi ile okunabilir.

Tüm bunlar bilgiye erişme sürecini uzatır, ancak manyetik ortamlarda saklanan bilgilerin ''uçucu'' olmaması, önemli bir avantajdır. Ayrıca bu sistemde saklanabilecek veri miktarı, bir yarı-iletken bellekte (MOS) saklanabilecek miktardan yaklaşık on kat fazladır.

IBM firması ve bazı Japon bilgisayar üreticileri, endüstriyel uygulamalarda bu tür bellekleri gitgide daha çok kullanmaktalar.

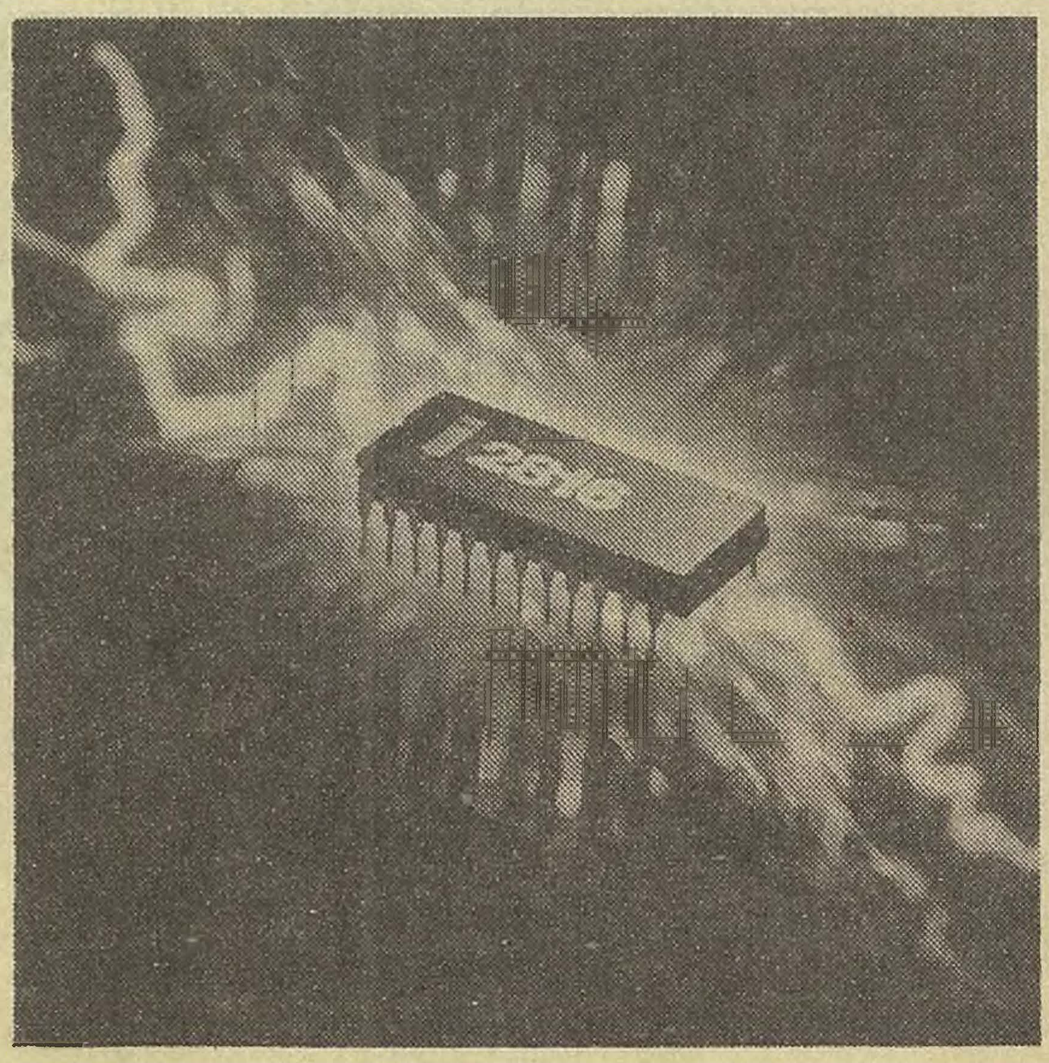
Yük akışından yararlanan beliekler, yarı-iletken teknolojisine dayanan ve bu yarı-iletkenlerden büyük arşivler gibi yararlanmayı sağlayan bir teknolojiye dayanır. Bilgiler uzun sıralar halinde saklanır ve kabarcık bellekte olduğu gibi ancak bir okuma kafasının önüne geldiğinde okunabilir.

CCD bellek uzun alan etkisi yaratan bir transistöre benzer. Bu transistörün iki ucu arasında çıkışları (ya da elektrotları) vardır. CCD bellekte kullanılan transistör, bahçe sulamada kullanılan delikli hortumlara benzetilebilir. Ancak transistörden geçen akım miktarı bahçe hortumundan geçen su miktarına göre daha iyi denetlenebilir ve çıkıştaki her elektrotta sinyaller dizisinin bir eşi yaratılır.

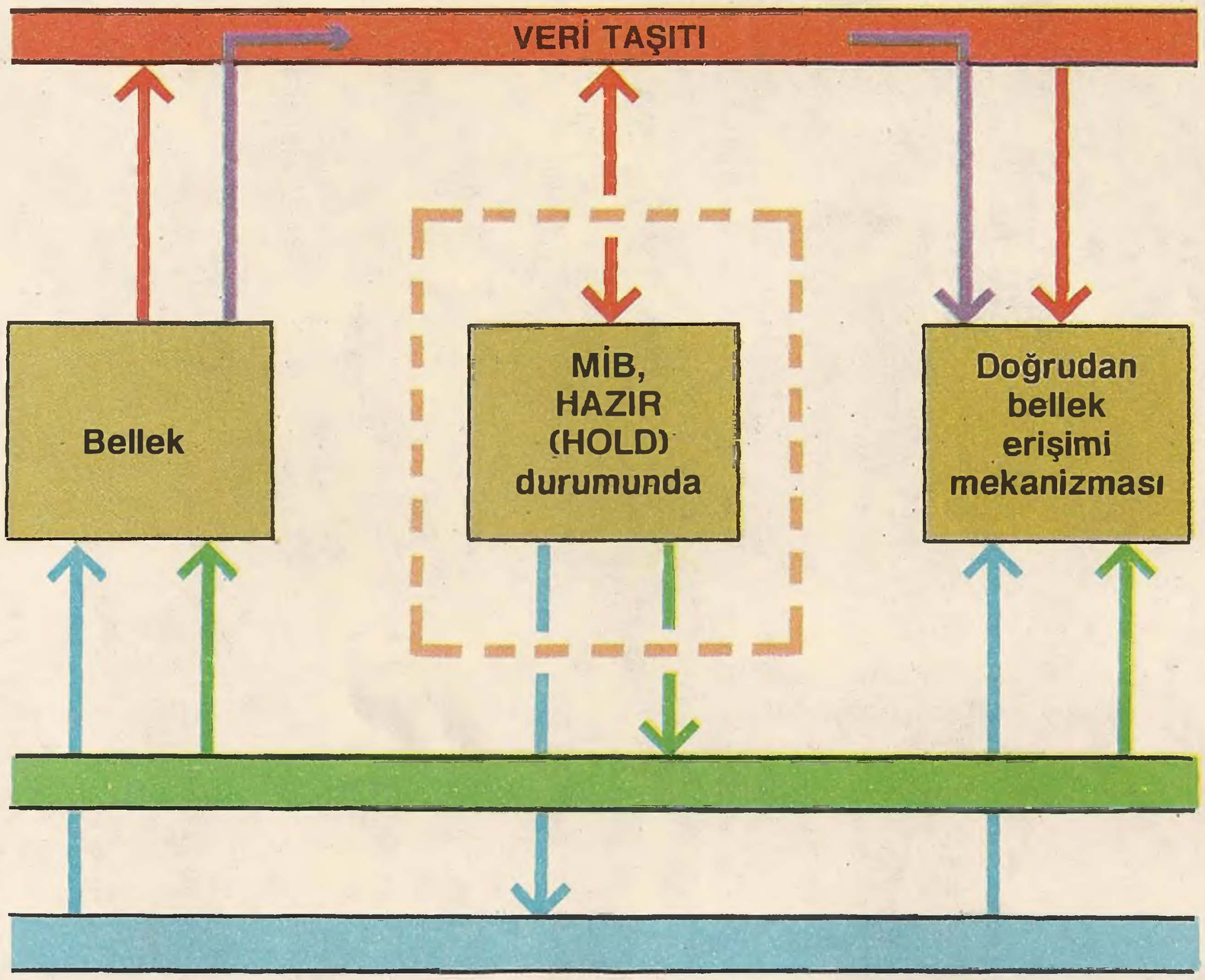
Verinin herhangi bir yerde takılmasını önlemek için düzenli olarak bir zamanlama sinyali vermek gerekir. Bu yüzden giren verinin saklanacak bir bilgi hücresi halini alması için gereken süre, diğer mekanizmalarda olduğundan daha uzundur.

Sistemi ucuzlatmak için çalışılmaktadır, ancak henüz geniş bir uygulama alanında kullanılabileceği kanıtlanamamıştır.

Bir EPROM YONGASI. 5 voltluk bir gerilim gerektiren ve 65.536 bitlik bir kapasitesi olan bu yonga, morötesi ışıkla programlanabilir.



DOĞRUDAN BELLEK ERİŞİMİNDE (DBE) VERİ AKIŞI



saklayabilir. Bu ise her bir simgenin taşınmasında elde edilecek küçük zaman tasarruflarının eklenerek disketin bütününde önemli bir kazanç oluşturacağı anlamına geliyor. Ayrıca, kullanıcı programları da manyetik disk birimlerinden sürekli olarak verilerin alınmasını ve bu verilerin işlenerek yeniden diske kaydedilmesini gerekli kılar. Bu gibi durumlarda doğrudan bellek erişimi zorunludur.

Çeşitli veri alışverişi yöntemleri arasında seçim otomatik olarak yapılır. Bazı özel programlar, alışveriş yönteminin seçiminden yürütülmesine kadar her şeyi kendi kendine halleder. Kullanıcı ya da programcının bu işlerle ilgilenmesine gerek kalmaz. Bu tür programlar, bilgisayarla birlikte verilen sistem işletim programının bir parçasıdır.

Bellekler

Programlar ve onların kullandığı ya da ürettiği veriler bellekte saklanır. Bu aslında geçici bir saklamadır, yeni bir program yazıldığında eskisi silinir. Kimi durumlar dışında belleğin içeriği bilgisayarın kapatılmasıyla kaybolur. Sürekli saklama ancak manyetik bir ortamda mümkündür, dolayısıyla yeniden kullanılacak program ve verilerin disklere ya da manyetik şeritlere kaydedilmesi gerekir.

Bilgisayar içindeki belleğin kapsamı dört ana bölüme ayrılır:

1 / Araya girme alanı,
2 / Sistem programları alanı,
3 / Yardımcı programlar alanı,
4 / Kullanıcı alanı.

Araya girme alanında, programın işlenmesine ara veren komutlar saklanır. Sistem programları ve yardımcı programlar alanı işletim sistemini oluşturur, bunlar bilgiyazma, klavyeden giriş, disk güdümü gibi temel işlevleri yürüten programlardır.

Kullanıcı alanı, kullanıcı programlarına ve onlarla ilgili verilere ayrılmıştır. Küçük ve kişisel bilgisayarların ortalama 32.000-64.000 bellek hücresi vardır. İşletim sisteminin karmaşıklı-

ğına göre bunlardan 28.000 ilâ 58.000 kadarı kullanıcıya ayrılır. Ancak bazı özel kişisel bilgisayarlarda ve küçük bilgisayarlarda bu sayı 170.000 ilâ 250.000'e, daha büyük bilgisayarlarda ise 2 ilâ 4 milyona ulaşır.

Yaz-oku Bellekler-RAM (Random Access Memory): Kısaca RAM terimiyle belirtilen bu bellekler ikiye ayrılır:

- **Statik**
- **Dinamik**

Statik bir belleğe verinin girilmesi, istendiği zaman yeniden bellekten alınabilmesi için yeterlidir. Dinamik bellek ise verilerin düzenli aralıklarla saniyede birkaç yüz kez yeniden yazılmasını gerektirir. Aksi halde kendiliğinden silinirler. Dinamik belleklerde verileri yenileme işlemine TAZELEME (REFRESH) denir.

Salt Okunur Bellekler-ROM (Read Only Memory): İçeriklerinin bilgisayar üreticisi tarafından yerleştirildiği adreslenebilir belleklerdir. Değiştirilmeyecek programlar bu tür belleklerde saklanır. İki tür salt okunur bellek vardır:

- **ROM**
- **EPROM**

ROM'un içeriği üretim sırasında belirlenir, değiştirme olanağı yoktur. EPROM ise mor ötesi ışıkla silinip yeniden programlanabilir.

Seri Bellekler: Çok özel amaçlar için uygulandıklarından seyrek raslanırlar. Kapasiteleri yüksek olmasına yüksektir ama çok karmaşık devreler gerektirirler.

Çevresel Gereçler

Ekran: Bir televizyon ekranına çok benzer. Gerçekten de bazı ucuz kişisel bilgisayarlarda normal televizyon ekranı kullanılabilir. Ancak bu takdirde ek bir devre gerekir.

İki tür ekran vardır:

- **Harfsayısal ekranlar**
- **Çizi (videografi) ekranları**

Harfsayısal ekranda sadece harfler ve sayılar, çizi ekranında ise harf ve sayılara ek olarak çizi ve tasarımlar da gösterilebilir. Başlıca farklılık görüntüyü oluşturan noktaların sayısındadır. Bu sayı ne kadar büyükse ekran çizim için o kadar elverişli olur.

Renkli çizi ekranlarındaki şekiller renkli olarak da gösterilebilir.

Bir ekranın en önemli özellikleri şunlardır:

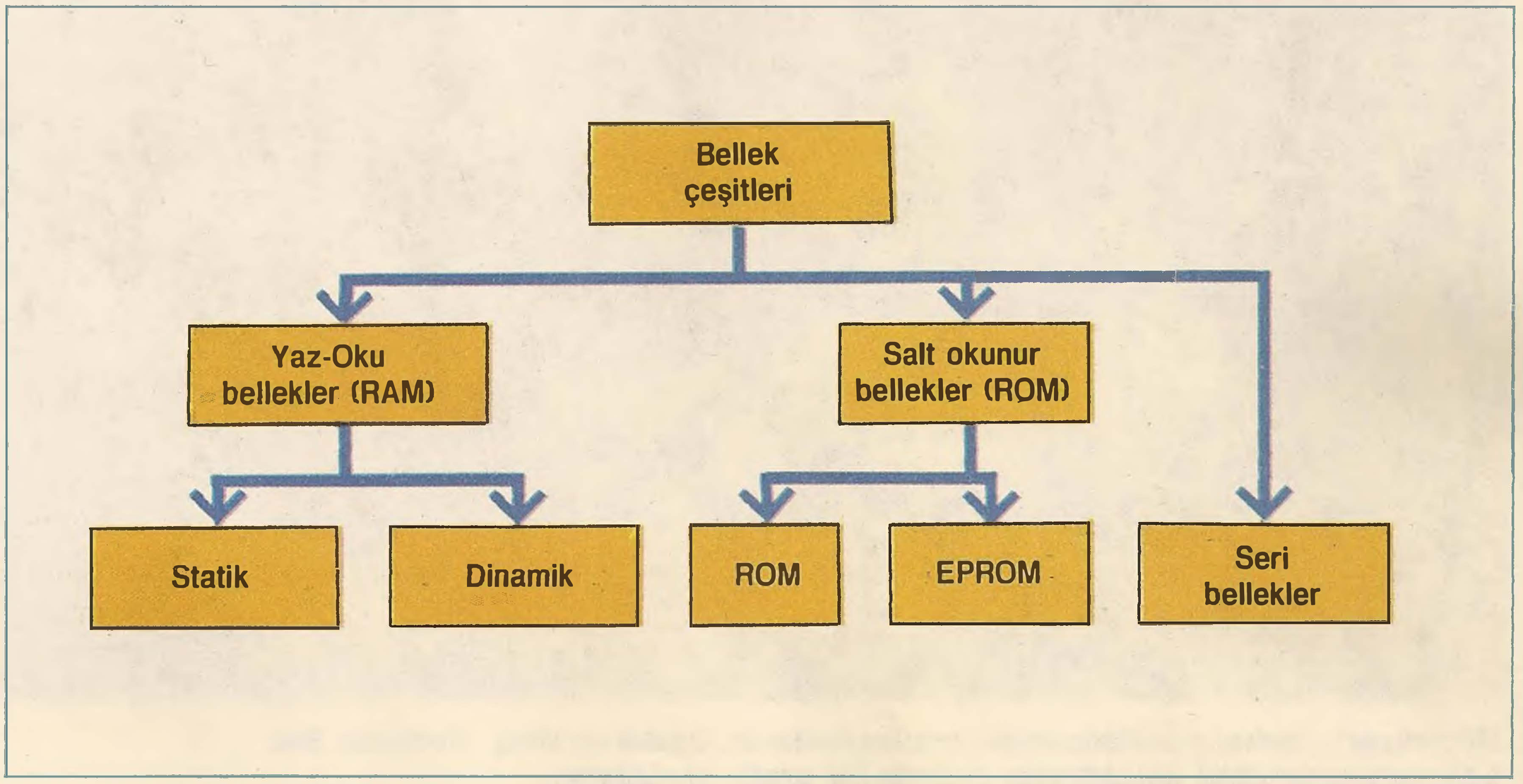
- **Büyüklük**
- **Satır ve sütun sayısı**
- **Renk**

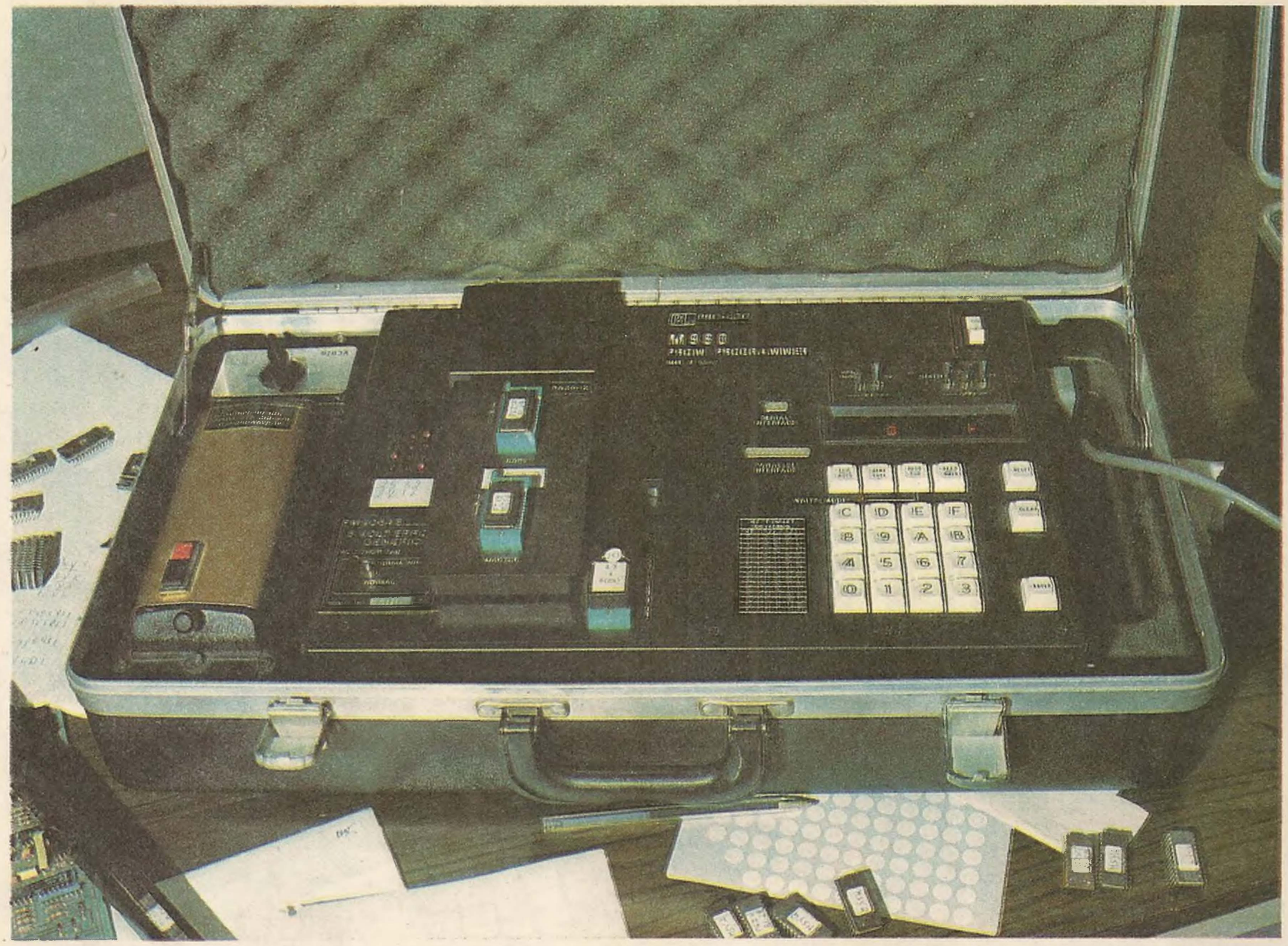
Ekranın satır ve sütun sayısı simgelerin büyüklüğüne bağlıdır. Yaygın olan bazı değerler:

16 satır × 64 sütun
25 satır × 80 sütun
40 satır × 132 sütun

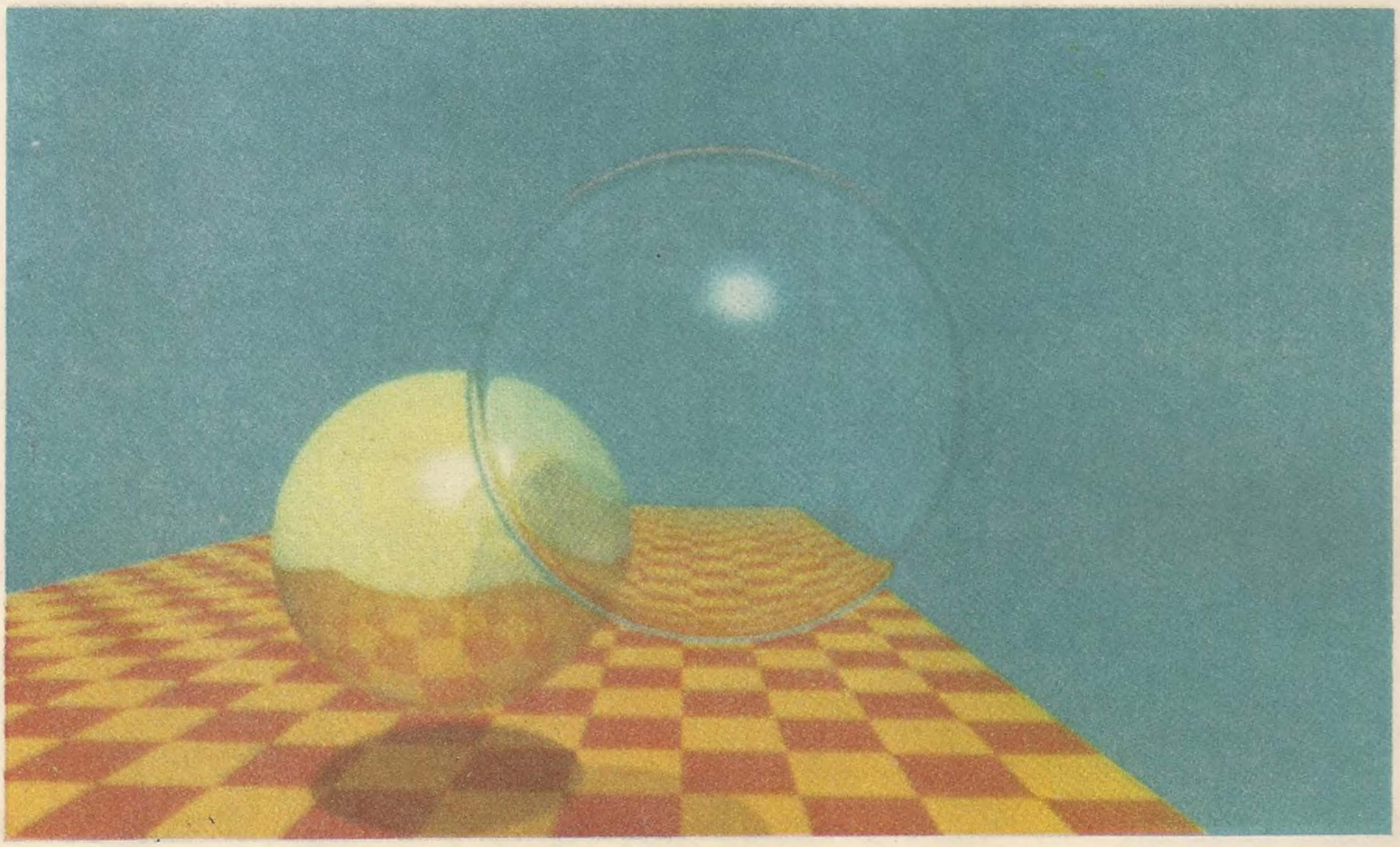
Bazı ekranlarda satır-sütun düzeni değiştirilebilir. Programcı, görünmesi istenen veri miktarına uygun düzeni kendisi seçer.

Çizi ekranlarında satır ve sütun yerine belirli şekil ögeleri vardır. Bunlar ekranın yatay ve

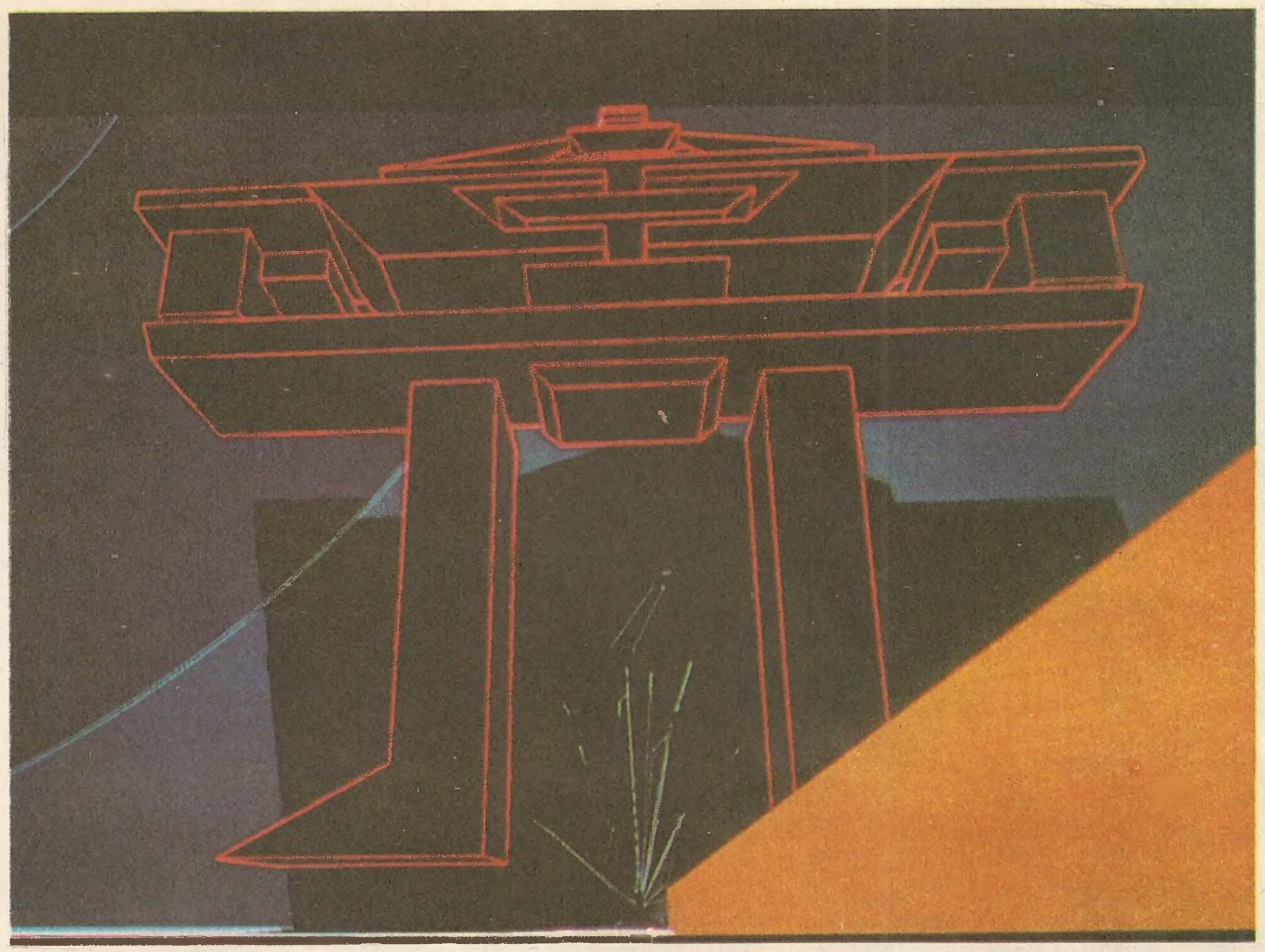


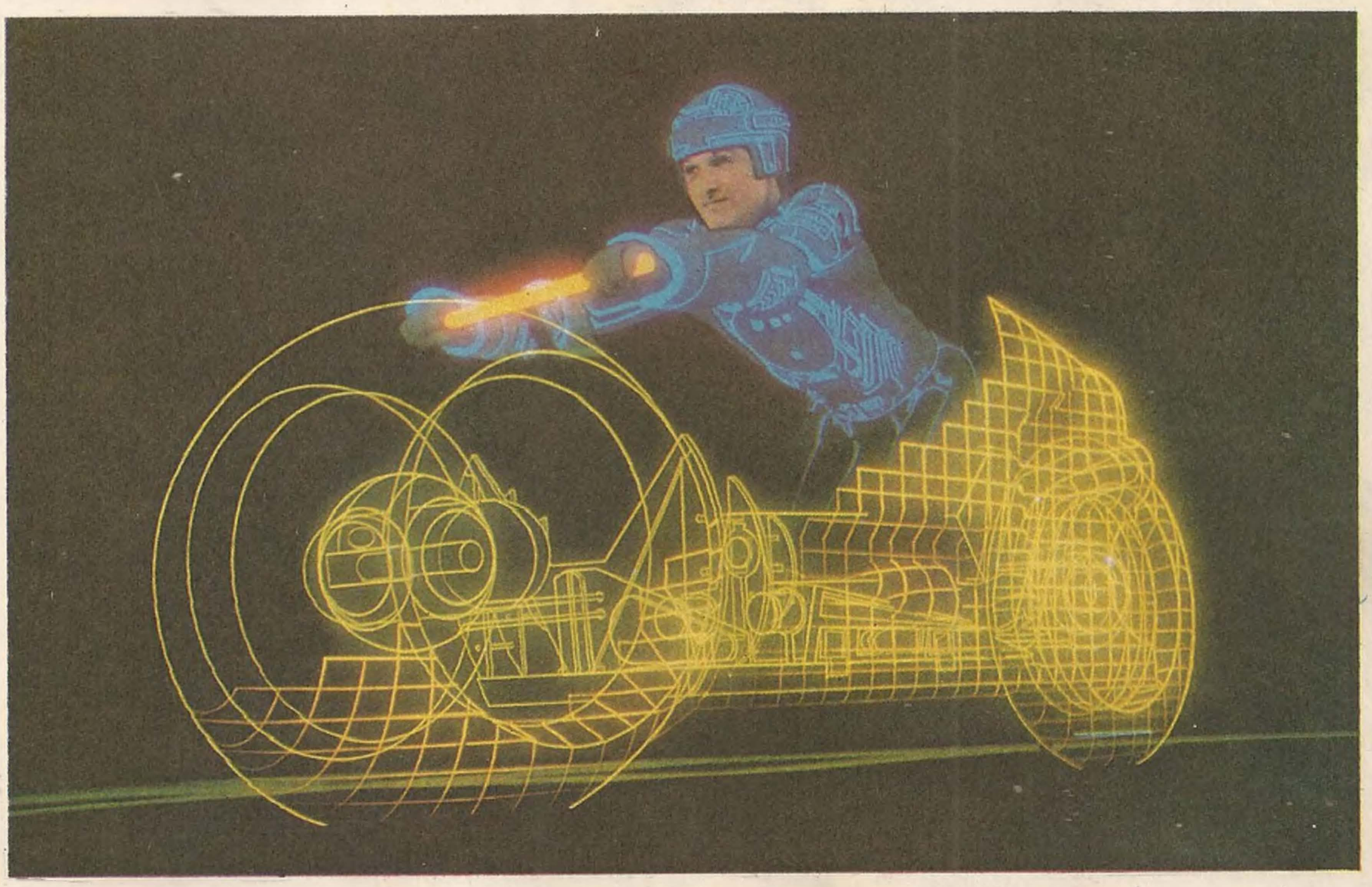


EPROM bellekleri programlama aygıtı.



Bilgisayarla çizim, sanatçıların ve grafikerlerin de ilgisini çekmiş. Resimde Bell Laboratuvarları'nda bilgisayarla çizilmiş bir grafik görülüyor.





**Bilgiyazar çizisinin uygulama alanlarından biri film hileleridir.
''Tron'' adlı filmden alınan iki görüntü.**

düşey kaç noktaya ayrıştırılabileceğini belirler. Tipik bir değer olarak 512 × 256 çizi noktası verilebilir.

Klavyeler

Genellikle iki tür klavyeden söz edilir: Hem harfleri hem de sayıları içeren harfsayısal klavyeler ve sadece sayıların bulunduğu sayısal klavyeler. Bazı bilgisayar modellerinde her tuşun iki işlevi vardır: Birincisi harfsayısal bir simge girişi sağlar, ikincisi ise önceden tanımlanmış bir komut verir. Bu ikinci işlev için özel bir kod kullanılır. Programlama sırasında önceden hazırlanıp kaydedilmiş komutların bir tek tuşla verilmesi büyük zaman tasarrufu sağlar.

Bilgiyazarlar

Kişisel bilgisayar sistemlerinde en sık kullanılan bilgiyazarlar şunlardır:

- **İğneli bilgiyazar**
- **Papatya bilgiyazar**
- **Paralel bilgiyazar**

İğneli bilgiyazar: Bu ad simgeleri basma şeklinden kaynaklanır. Yazma kafasındaki belirli sayıda iğne elektrikli bir mıknatısla hareket ettirilerek yazılacak simgeye göre öne çıkarlar. Bu şekilde yazılan simge bitişik birçok noktadan oluşur. (İğneli bilgiyazar yazı örnekleri için bkz. şekil.)

Bu bilgiyazarların, bir ölçüde ucuz olmak, ortalama bir hızla çok farklı simgeleri yazabilmek gibi yararları vardır. En büyük sakıncaları ise yazının pek güzel görünmemesidir.

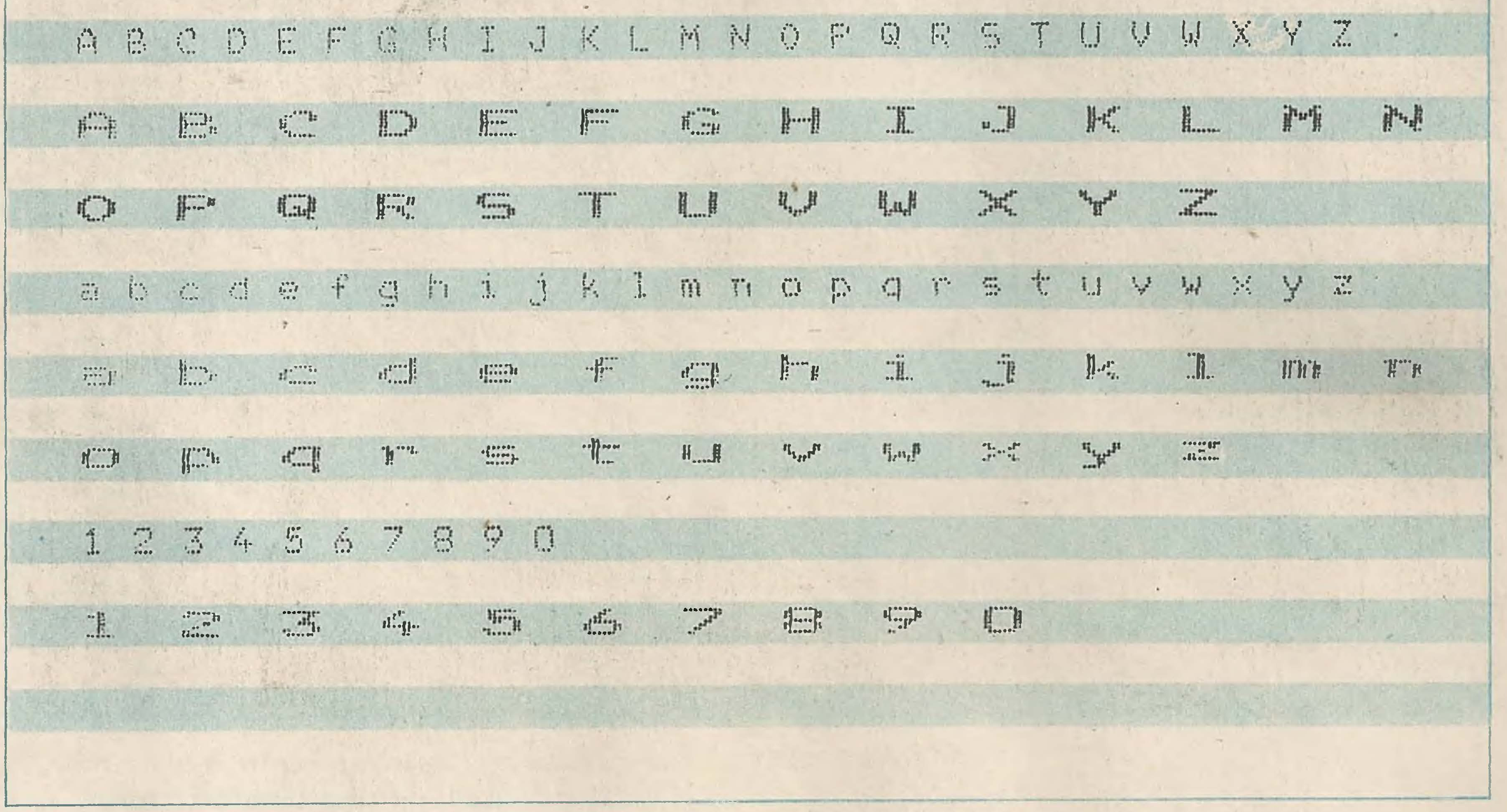
Papatya bilgiyazar: Bunda her simge hazır olarak bilgiyazarda bulunur ve yalnızca kâğıda basılması gerekir. Harfleri daktilonunkilerin aynısıdır. Bu bilgiyazarlar iğnelilerden daha pahalıdır, daha yavaş yazarlar ve tek bir simge değiştirmek mümkün değildir, değişiklik istendiğinde simge tekerinin bütünüyle değiştirilmesi gerekir.

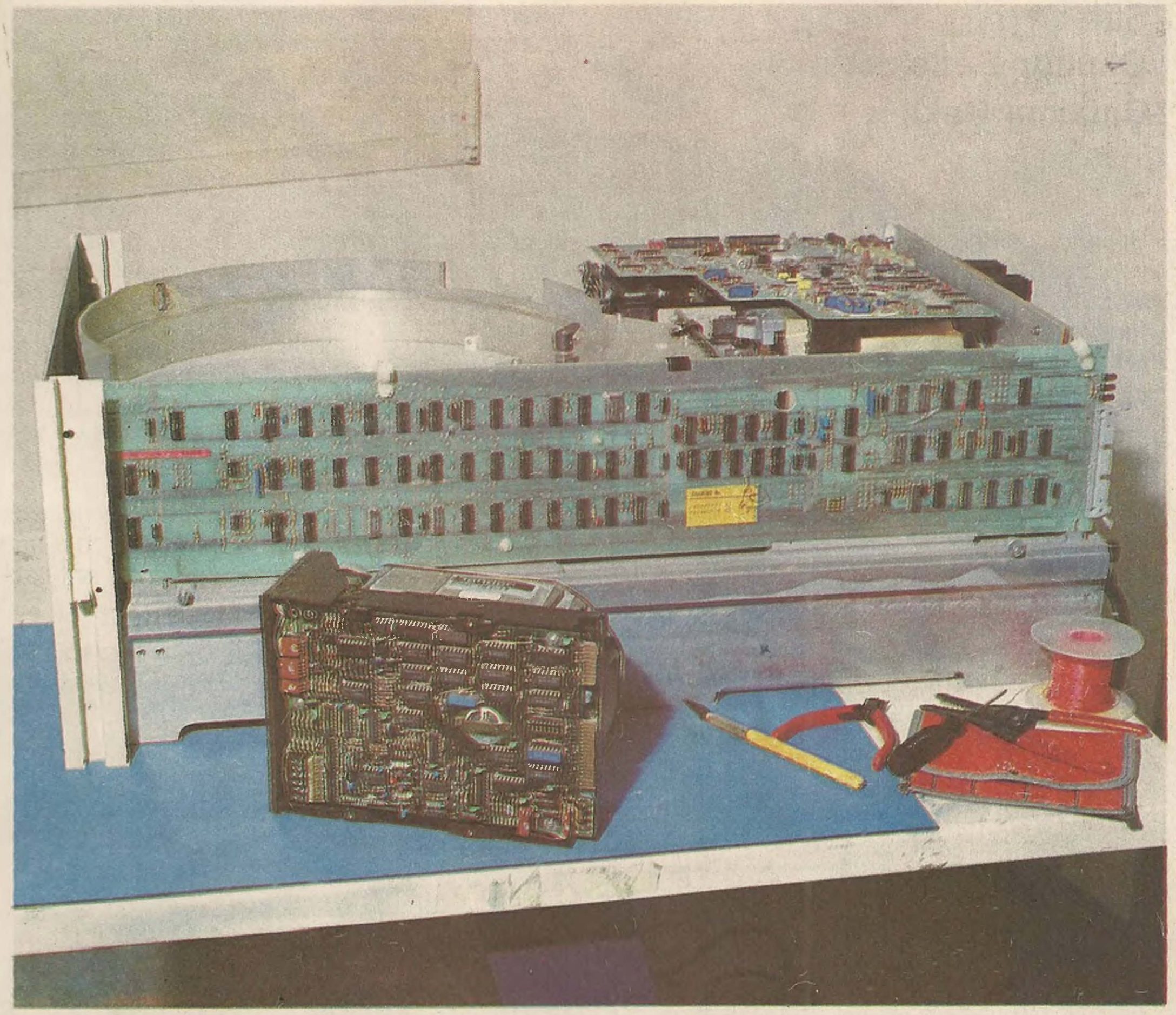
Paralel bilgiyazar: En hızlı tip olup daktiloda 600 satıra kadar yazabilir. Fiyatı da aynı oranda yüksek olduğundan küçük bilgisayarların nispeten daha büyükleri için uygundur.

Disk Bellek Birimleri

Kişisel bilgisayarlarda iki tür disk bellek kullanılır: disketler ve sert diskler.

İĞNELİ BİLGİYAZARLA YAZILMIŞ YAZI ÖRNEKLERİ





İki disk bellek biriminin karşılaştırılması. Boyutlarının çok farklı olmasına karşın bellek kapasitesi ikisinde de aynıdır.

Disketler makineden çıkarılabilir ve modeline göre 250.000 ilâ 1.200.000 simgelik kapasiteleri olabilir.

Disketlerin diğer bir türü normal olarak yerinden çıkarılmaz ve kapasiteleri 5-10 milyon simgedir.

Disketler:

- **Tek ya da çift yüzeyli olabilir,**
- **Tek ya da çift yoğunluklu olabilir.**

Yoğunluk, veri saklayabilme kapasitelerinin derecesi ile ilgilidir. Tek ya da çift yüzeylilik ise disketin ne kadarının veri saklamada kullanılabileceğini ifade eder. Açıktır ki en az veri alabilen disketler tek yoğunluklu ve tek yüzeyli disketlerdir. Buna karşılık çift yüzeyli, çift yoğunluklu disketler en büyük bellek kapasitesine sahiptir.

kullanır. Bunlardan ikincisi daha pahalı, ancak daha yüksek kapasitelidir. Bu nedenle sadece belirli büyük sistemlerde bulunur.

Daha ucuz olan sert taşınır disklerin bellek kapasitesi 10 milyon simge dolayındadır. Yukarıdaki fotoğrafta iki farklı disk bellek biriminin karşılaştırılması görülmektedir. Aralarındaki büyüklük farkına karşın her iki birimin de 10 Mbaytlık (10 milyon simge) bir kapasitesi vardır. Büyük olanı sert taşınır disk, küçük olanı ise çoklu disket birimidir. Fotoğraf, boyutlardaki farklılığı açıkça gösteriyor.

Salt Okunur Belleklerin Çalışma Şekli

Bildiğimiz gibi salt okunur bellekler değişmeyen algoritmaları kaydetmeye ve belirli süreçleri denetlemeye yarar. ''Kalıcı yazılım'' ya da ''katı yazılım'' denen bu programlar sürekli tekrarlanan, çok sayıda işlevi yürütür. Şimdi inceleyeceğimiz örnekte, bir ROM'un 7 parçalı bir göstergeyi denetlemesini açıklayacağız. Bu olayla, örneğin hesap makinesinde işlem yaparken karşılaşıyoruz.

Önce mantık kapılarından oluşan bir kod çözücü, iki tabanlı sayıları on tabanlıya çevirir. Örneğimizdeki dört giriş hattı, 0'dan 9'a kadar olan sayıların iki tabanlı karşılıklarını kod çözücüye ulaştırır. Ayrıca, 0'dan 9'a kadarki 10 tabanlı sayıları göstermek için de 10 tane hat gereklidir.

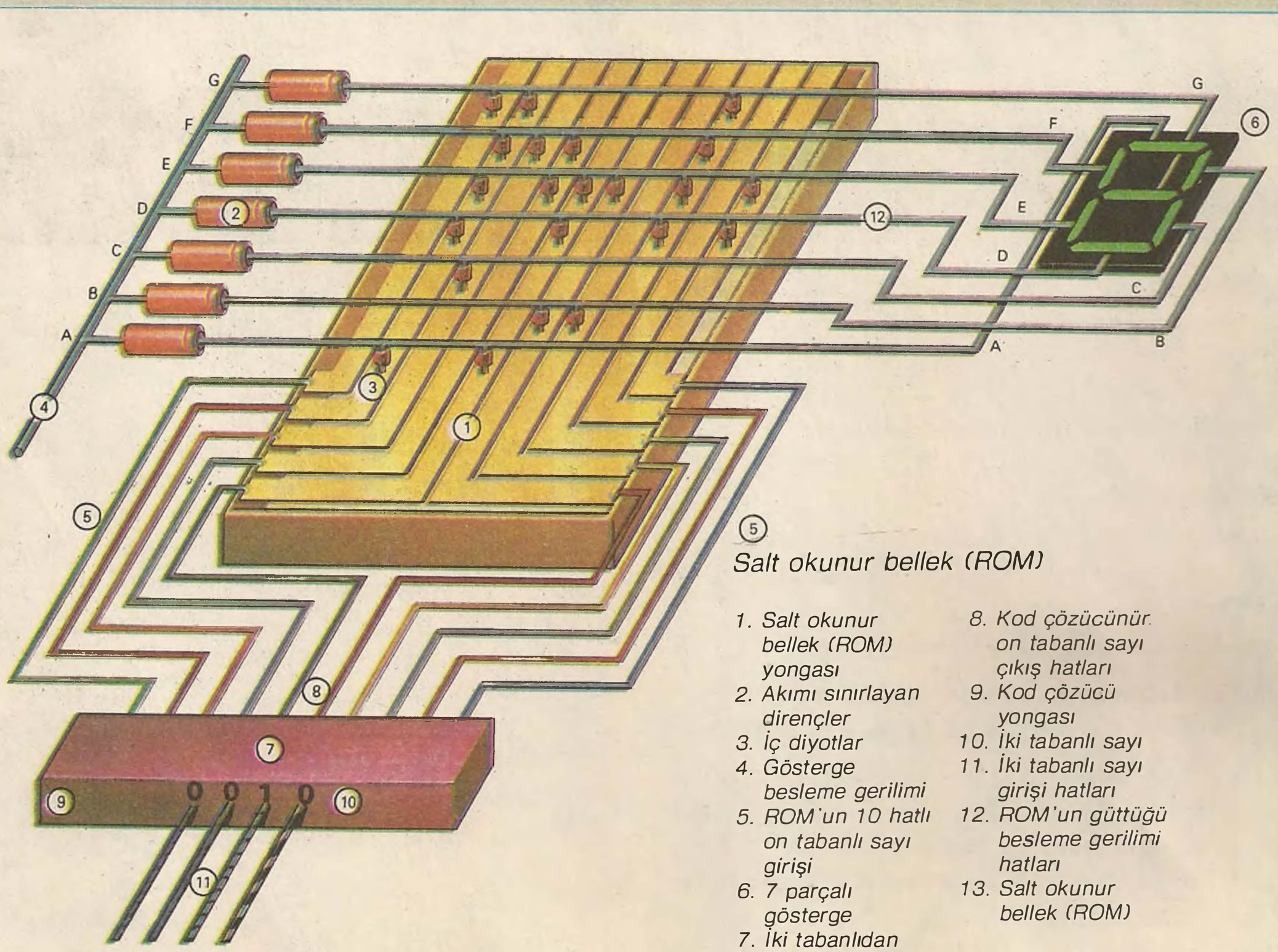
Kod çözücünün girişinde sinyal yoksa, çıkış hatlarının gerilimi sabit bir düzeyde (genellikle 5 volt düzeyinde) kalır.

Kod çözücü, sinyal olarak örneğin 0000 sayısını (yani 10 tabanlı 0 sayısının iki tabanlı karşılığını) alırsa, on tabanlı sayı çıkışının ilk hattında gerilim sıfır düzeyine düşer ve on tabanlı 0 sayısı üretilmiş olur. Bu sırada diğer hatlardaki gerilim, her zamanki sabit düzeylerinde bulunmaktadır.

Kod çözücü on tabanlı 1'in karşılığı olan iki tabanlı 0001'i aldığında ise, bu kez ikinci çıkış hattındaki gerilim sıfırlanır ve diğer bütün hatlardaki gerilim yine 1 (yani 5 volt) düzeyinde kalır.

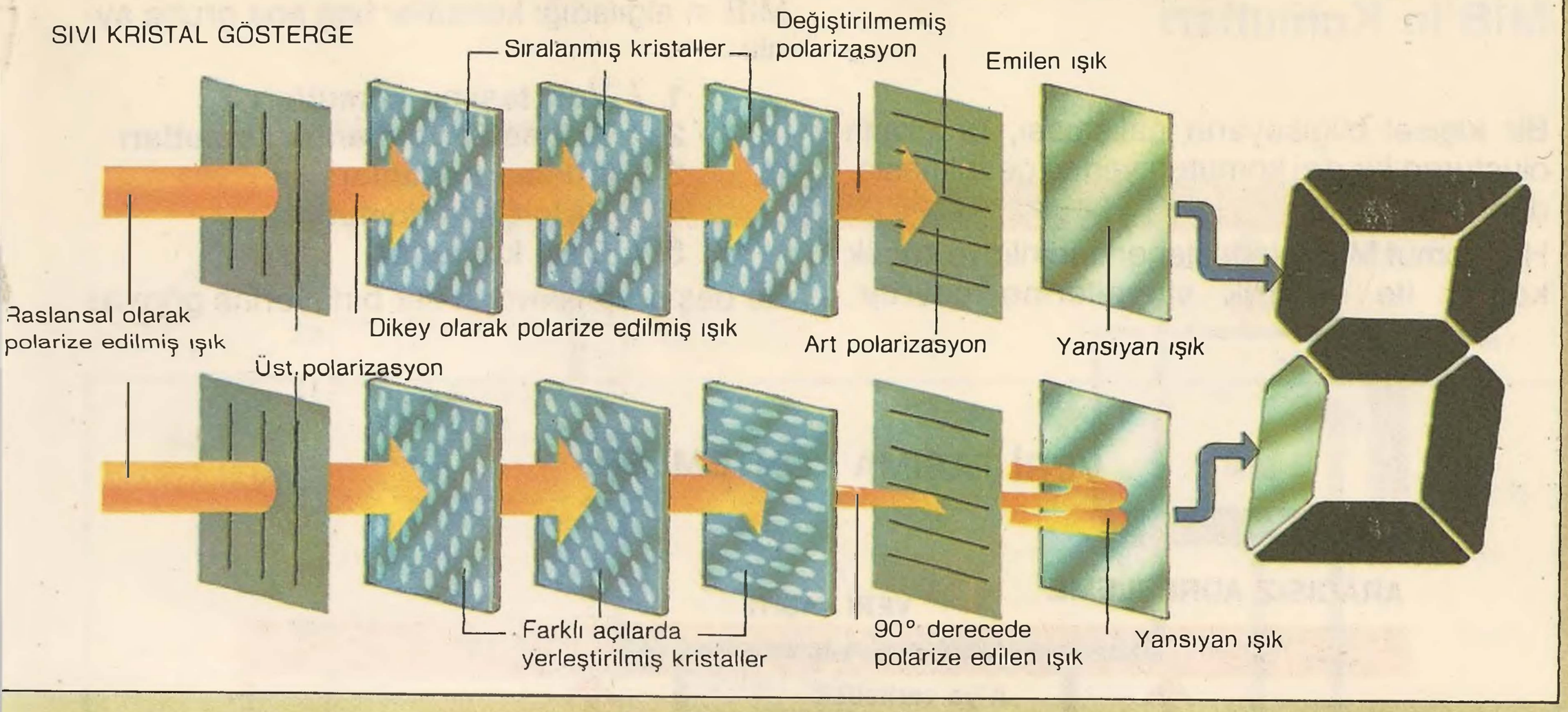
Aynı yöntem diğer on tabanlı sayılar için de uygulanır. Her on tabanlı sayı, kod çözücünün çıkışında, kendisine karşılık gelen hattın geriliminin sıfırlanmasıyla gösterilir.

Negatif mantıkta olduğu gibi, işaret olarak gerilimin varlığını değil, yokluğunu kullanmanın basit bir nedeni vardır.



Salt okunur bellek (ROM)

1. *Salt okunur bellek (ROM) yongası*
2. *Akımı sınırlayan dirençler*
3. *İç diyotlar*
4. *Gösterge besleme gerilimi*
5. *ROM'un 10 hatlı on tabanlı sayı girişi*
6. *7 parçalı gösterge*
7. *İki tabanlıdan ondalığa kod çözücü*
8. *Kod çözücünün on tabanlı sayı çıkış hatları*
9. *Kod çözücü yongası*
10. *İki tabanlı sayı*
11. *İki tabanlı sayı girişi hatları*
12. *ROM'un güttüğü besleme gerilimi hatları*
13. *Salt okunur bellek (ROM)*



Bir hatta gerilimin olmaması, o hattın topraklanmış olması demektir; dolayısıyla söz konusu hatta bağlı her hat kısa devre yapar ve böylelikle yanması istenmeyen gösterge parçalarının gerilimi sıfırlanır. Yedi parçalı bir göstergede 0'dan 9'a dek tüm sayılar bu şekilde elde edilir. Burada ROM'un görevi, kod çözücünün verdiği değerlere göre gösterge parçalarının bazılarını söndürmektir.

Örneğin kod çözücünün birinci çıkış hattında 0 varsa, on tabanlı 0 sayısının görünmesi için göstergenin ortasındaki yatay parçanın yanmaması gerekir.

Çok basit bir ROM'un üretimi çeşitli evrelerden oluşur.

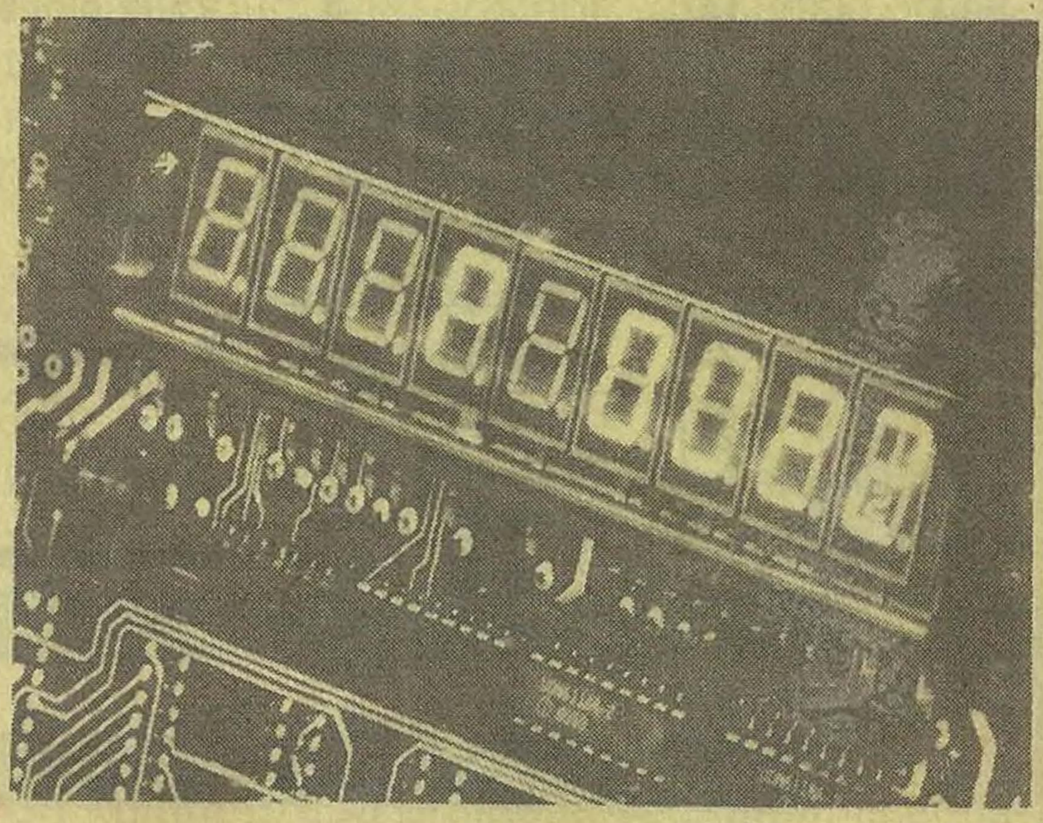
Önce girişteki dört bitin her bileşimi için hangi gösterge parçalarının yanacağını gösteren bir tablo hazırlanır.

Gerekli bağlantılar bir diyot matrisiyle sağlanır (bkz. şekil, s.144).

Kod çözücünün birinci çıkış hattıyla şekildeki G gerilimi hattı arasına bir diyot bağlanırsa, bu hatta gerilimin sıfırlanması, göstergenin orta parçasının gerilimini keser. A ile F arasındaki gerilim hatlarını, kod çözücünün 0 hattına bağlayan diyotlar bu işlemin tamamen dışında kalırlar. Bunu sağlamak için A-F hatları yakılarak söz konusu matristen çıkarılmışlardır. Bu daha ROM'un programlanması sırasında diyotlara yüksek gerilim bağlayarak yapılır. Böylece ROM basit bir yöntemle, öngörülen işleri yapacak şekilde programlanmış olur.

Daha önce de sözünü ettiğimiz gibi, sayısal bir göstergeyi, merkez bilgisayardan verileri alan, verilerin kodlarını çözen ve çıkışa gerekli sinyalleri gönderen bir ara birim yönetir. Yedi parçası ışıklı diyotlardan(LED = Light Emitting Diode) oluşan bir gösterge belirli diyotlara gerilim vererek on tabanlı rakamları gösterebilir.

Sıvı kristal göstergelerde de, aynı işlevleri yürüten yedi sıvı halindeki parça vardır. Bu göstergelerde, kristallerin özelliklerinden yararlanılır. Kristal öndeki saydam, arkadaki yansıtıcı iki plaka arasına yerleştirilir. Ön ve arka plakaların sayıya karşılık gelen parçalarına gerilim verildiğinde kristal sıvı saydamlığını kaybeder, yansıtma özellikleri değişir ve göstergenin bu parçası görünür hale gelir.



Diyotlu bir gösterge.

MİB'in Komutları

Bir kişisel bilgisayarın çalışması, programı oluşturan bir dizi komutun yerine getirilmesinden ibarettir.

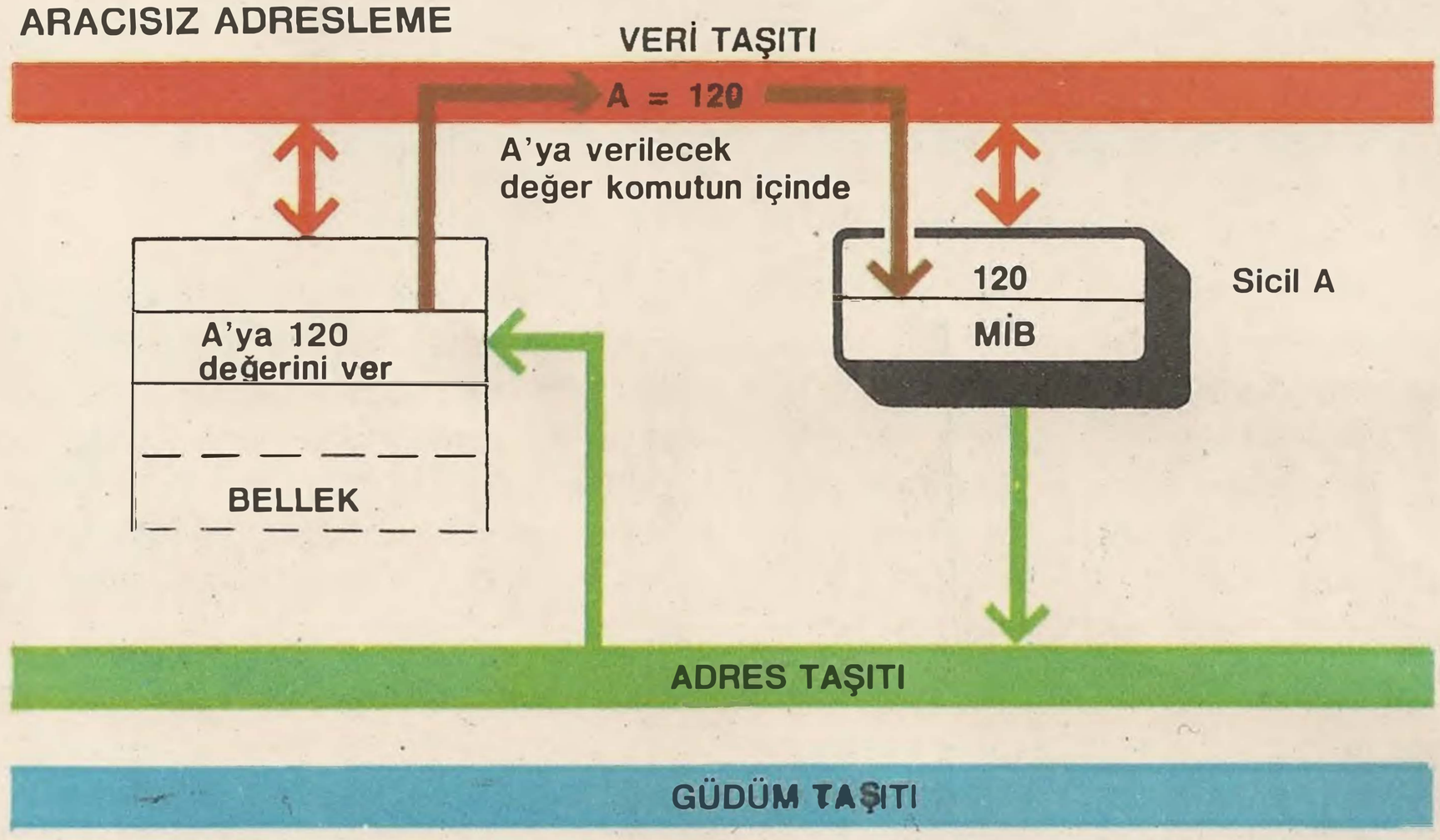
Her komut MİB içinde değerlendirilir ve küçük kodlar ile elektrik sinyallerine çevrilir.

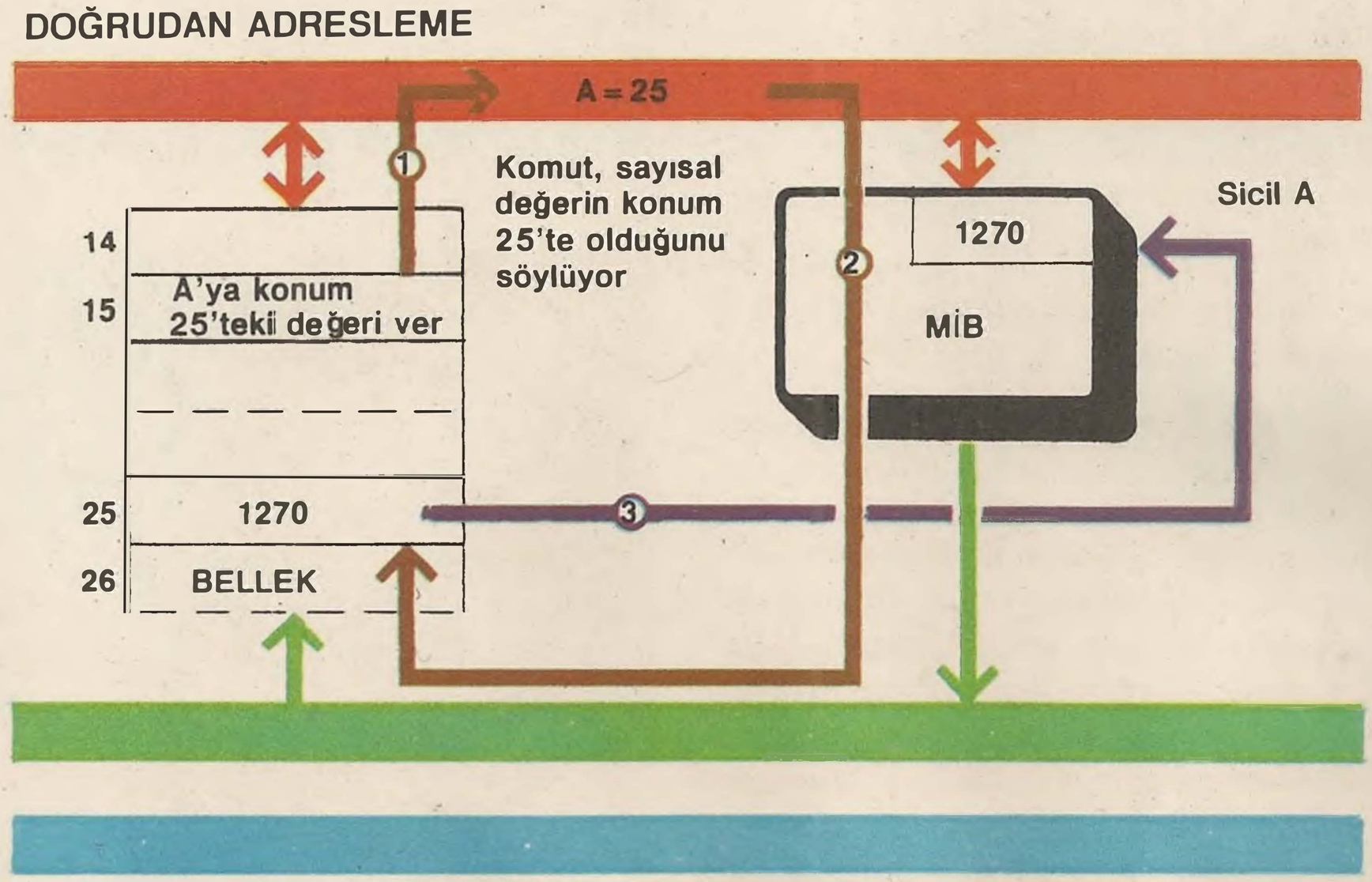
MİB'in algıladığı komutlar beş ana gruba ayrılır:

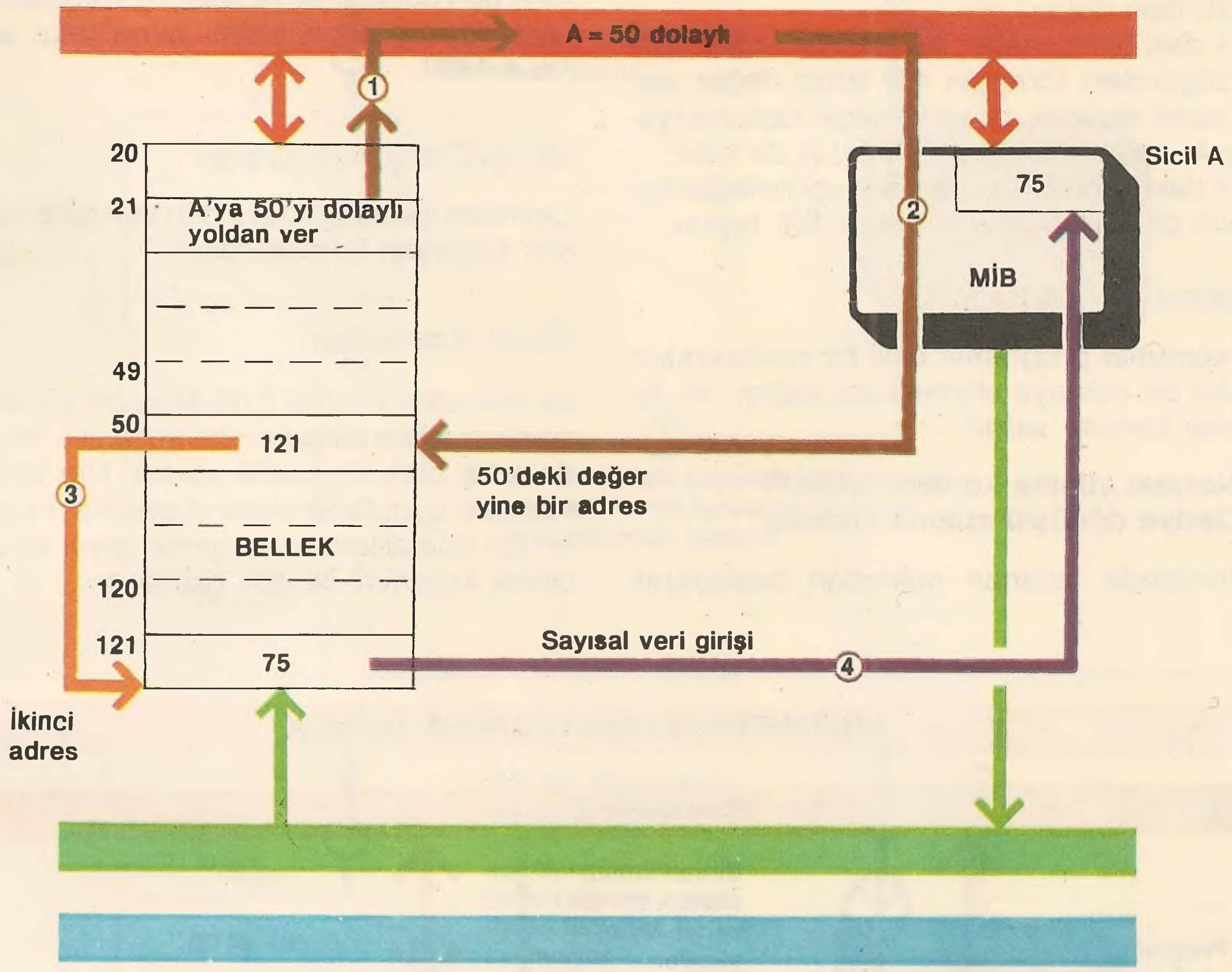
1 / Veri taşıma komutları
2 / Aritmetik ve mantık komutları
3 / Atlama komutları
4 / Giriş/Çıkış komutları
5 / Özel komutlar

Bu beş grup işlevden her biri, yerine göre az

VERİ TAŞIMA YÖNTEMLERİ







ya da çok karmaşık bir dizi komut tarafından yerine getirilir.

Veri Taşıma Komutları

Bellekle ilgili giriş/çıkış komutlarıdır.

Minik-işlemcilerde veriler, sistem belleği ile MİB'in özel bellek bölümü, SİCİL arasında değiş tokuş edilir.

Veri taşıma komutları üç çeşittir:

Aracısız veri taşıma (adresleme): Seçilen herhangi bir sicile sayısal bir değer verilir; örneğin, A siciline 120 değerinin verilmesi gibi.

Dolaysız veri taşıma (adresleme): Sicile belirli bir bellek sözcüğündeki depolanmış değer yazılabilir; örneğin A siciline belleğin 25. sözcüğündeki değerin yazılması. Bu durumda önceki komutun aksine sicile sabit bir değer değil, sözcük 25'in ''içeriği'' veriliyor.

Dolaylı veri taşıma (adresleme): Dolaysız veri taşımanın genişletilmiş bir şeklidir. Sicile arka arkaya iki adres getirilir, ikinci adres verinin bulunduğu yeri gösterir. Yukarıdaki örnekte A siciline dolaylı olarak 50 sayılı bellek veriliyor. Komut şöyle yerine getirilir:

① ② 50 sayılı bellek sözcüğündeki değer alınır.

③ Bu değer, örneğin 121, yeni bir adres olarak yorumlanır.

④ Sözcük 121'deki veri alınır.

Burada sadece bellekten sicile veri aktarmayı inceledik. Tabiî aynı şekillerde ters yönde veri aktarmak da mümkündür.

Aritmetik ve Mantık Komutları

Bu komut grubu aritmetik ve mantık işlemlerini kapsar. Herhangi bir işlemin yürütülebil-

mesi için, verilerin alınacağı ve sonucun yazılacağı adreslerin belirtilmesi gerekir. Gerekli komutlar şöyle sıralanmalıdır:

1 Sicile birinci veri yazılır (örneğin sözcük 26'daki değer).
2 A'daki birinci değer ile belirli bir bellek sözcüğündeki (örneğin 41) ikinci değer veri olarak alınarak, işlem (örneğin toplama) yapılır. İşlem bittiğinde sonuç A'da kalır.
3 A'daki yeni değer (yani sonuç) belleğin belirli bir sözcüğüne (örneğin 53) taşınır.

Atlama Komutları

Bu komutlar programın belli bir noktasından başka bir noktaya atlanmasını sağlar. İki tip atlama komutu vardır:

- **Normal atlama komutu (JUMP)**
- **Geriye dönüşlü atlama komutu**

Birincisinde, atlanan noktadan başlayarak programın sonuna kadar devam edilir (bkz. s.149, üstteki şekil). İkinci durumda atlamadan sonraki komutlar işlenirken RETURN (DÖN) komutuna raslanınca ilk atlama komutuna geri dönülür ve bir sonraki komuttan başlayarak programın işleyişi sürer (bkz. alt şekil, s.149).

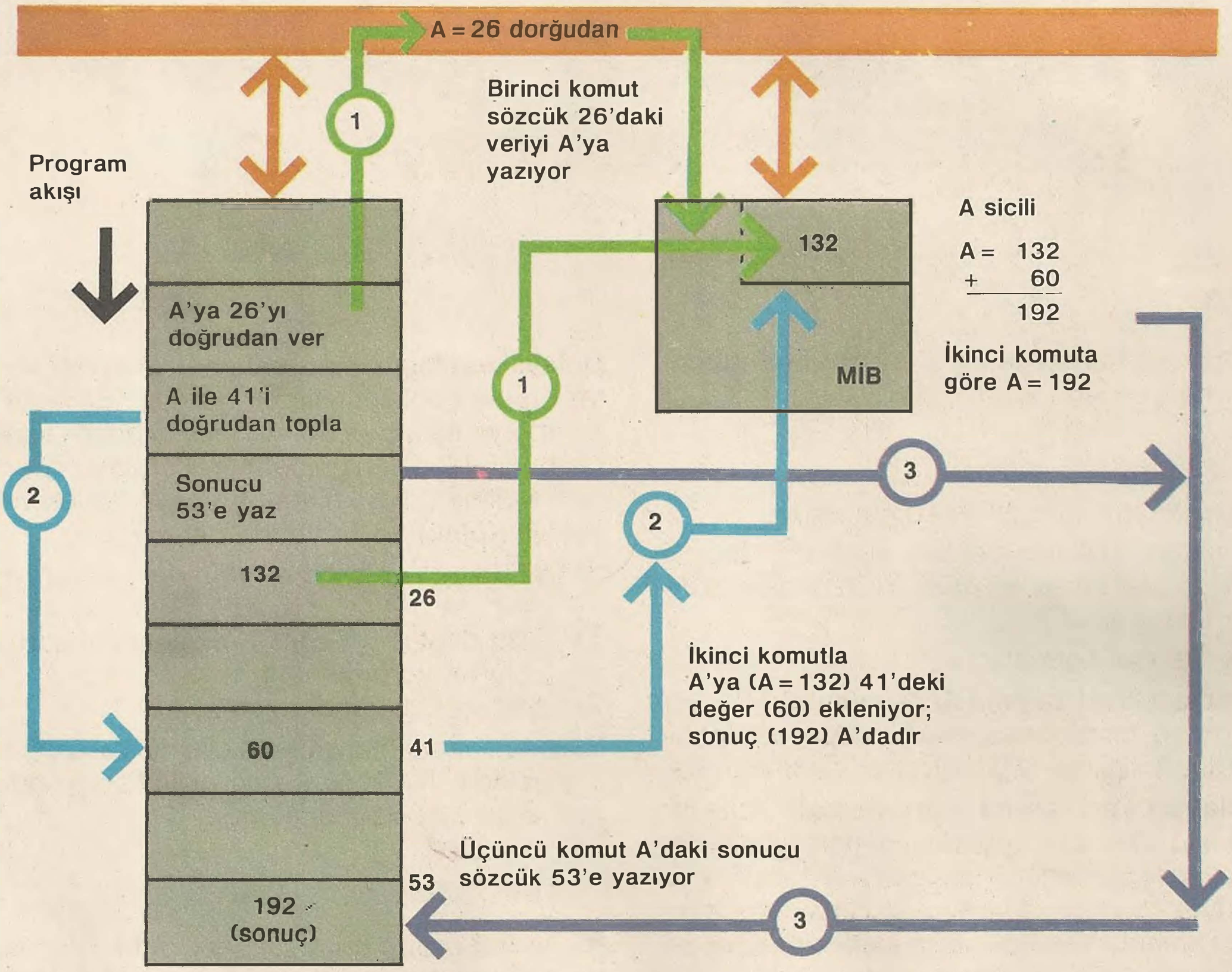
Giriş/Çıkış Komutları

Çevresel gereçler üzerinden veri giriş ve çıkışını sağlayan komutlardır.

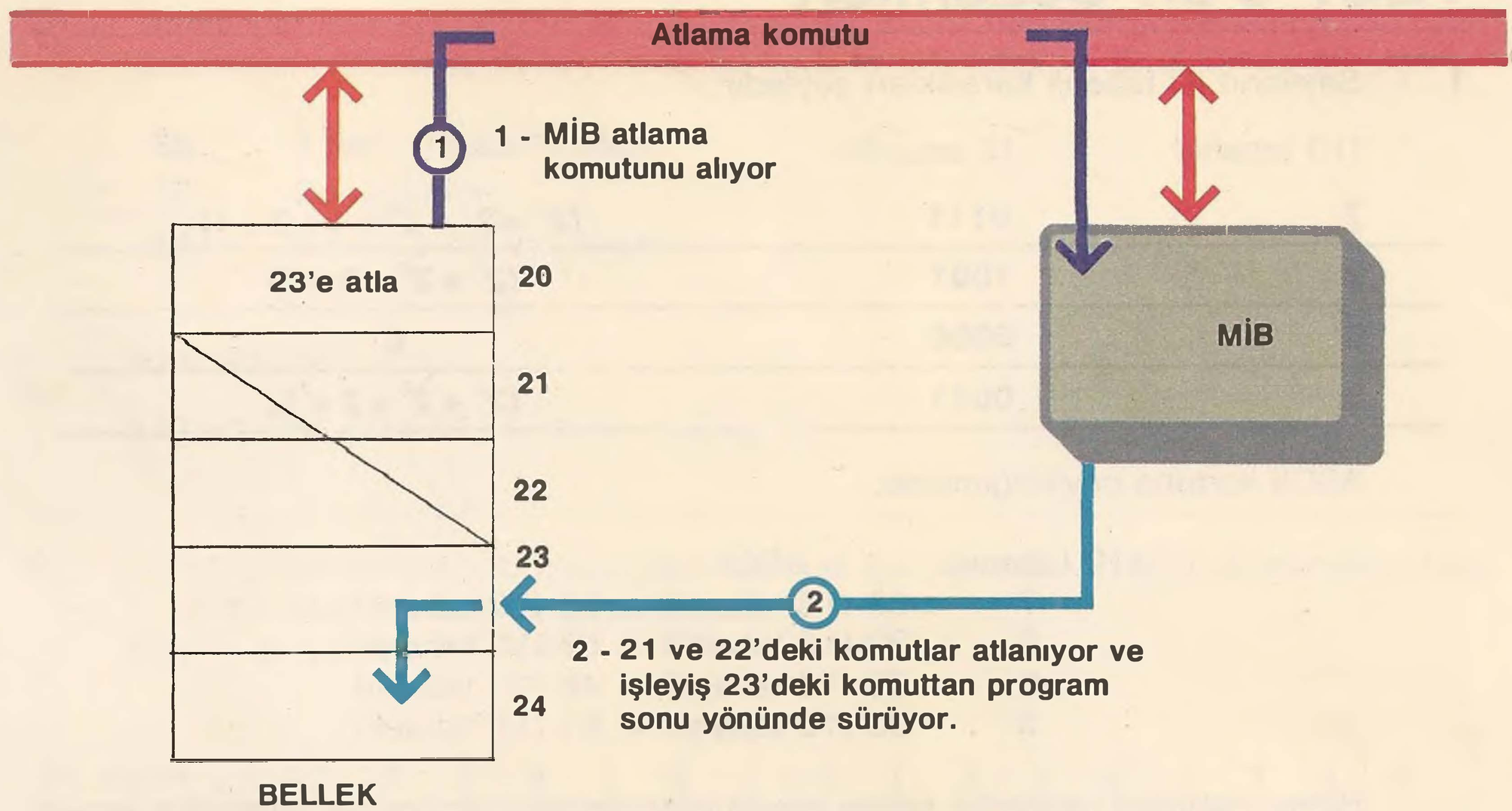
Özel Komutlar

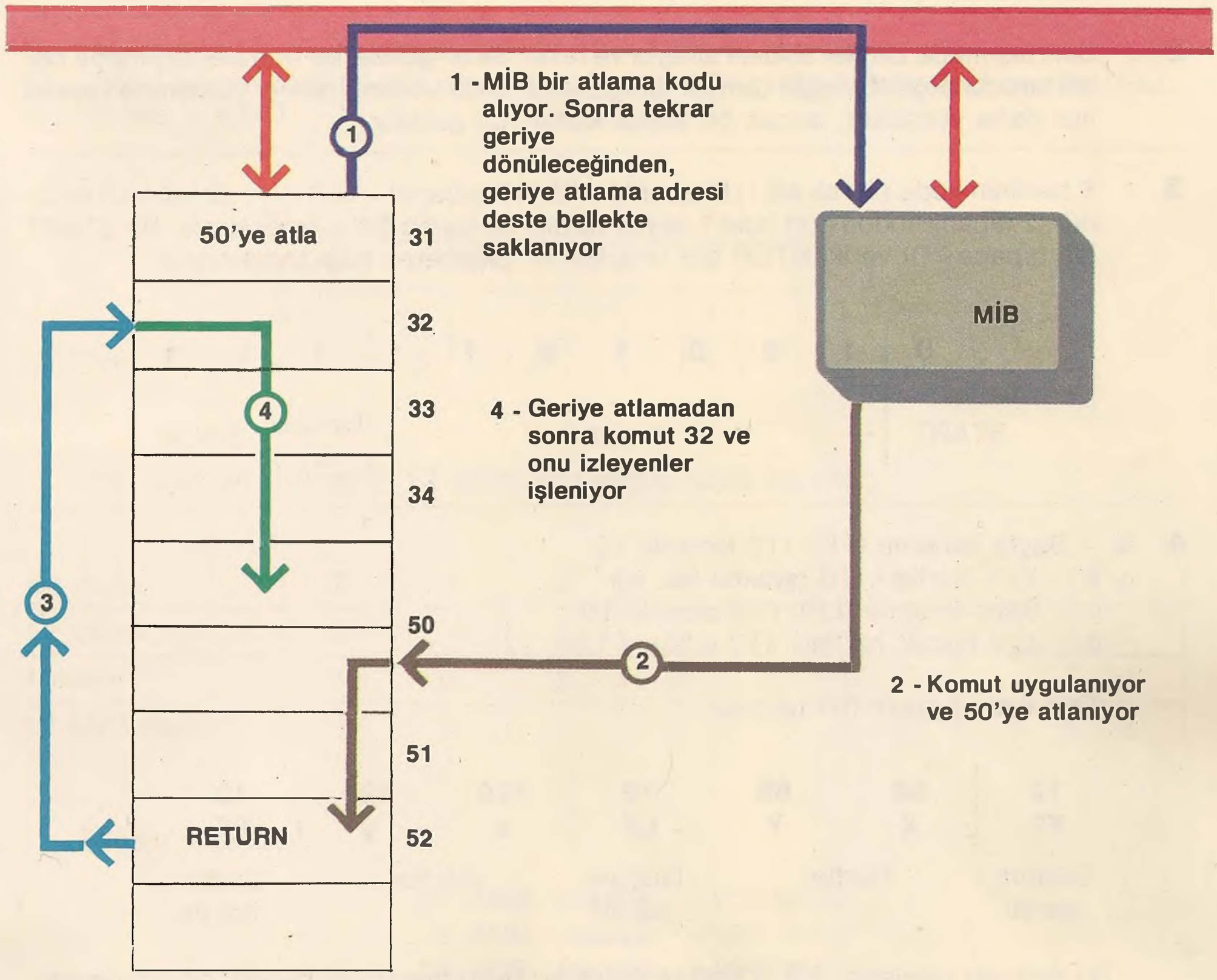
Bu komutların çoğu özel amaçlarla üretilmiş merkezi işlem birimlerinde kullanılır. Hepsinde ortak olan bir özellik yoktur. Her biri istenen özel uygulama alanının gereklerini yansıtır. En önemlileri araya girme işlevi ve deste bellek kullanımı ile ilgili olanlardır.

ARİTMETİK KOMUTLARINA ÖRNEK



ATLAMA KOMUTLARINA ÖRNEK





RETURN koduna gelindiğinde, atlama komutunu izleyen komuta geri atlanıyor

TEST 3'ün Çözümleri

1 / Sayıların iki tabanlı karşılıkları şöyledir:

(10 tabanlı)	(2 tabanlı)	
7	**0111**	**$(2^2 + 2^1 + 2^0 = 4 + 2 + 1)$**
9	**1001**	**$(2^3 + 2^0 = 8 + 1)$**
0	**0000**	**0**
3	**0011**	**$(2^1 + 2^0 = 2 + 1)$**

ASCII koduna çevirdiğimizde:

(10 tabanlı)	ASCII
7	37 (16 tabanlı) = 55 (10 tabanlı)
9	39 (16 tabanlı) = 57 (10 tabanlı)
0	30 (16 tabanlı) = 48 (10 tabanlı)
3	33 (16 tabanlı) = 51 (10 tabanlı)

Birinci (sayısal) gösteriliş biçimi hesap işlemlerinde, ikincisi ise giriş-çıkış ve veri saklamada kullanılır. Programcı, işlemlerde hata yapmamak için, hangi biçimi kullanacağını komutlarla belirlemelidir.

---

2 / Seri taşımada bilgiler (bitler) sırayla ve teker teker gönderilir. Paralel taşımada belirli sayıda bilgi, (örneğin tüm bir simge) aynı anda yollanır. İkinci yöntemde taşıma hızı daha yüksektir, ancak bit sayısı kadar hat gerekir.

---

3 / K harfinin kodu olarak 4B (16 tabanlı) = 75 (10 tabanlı) = 1001011 (2 tabanlı) bulunur. 2 tabanlı kodda dört adet 1 sayısı vardır. Tamlama biti 1 değerini alır. Bir START biti (space = 0) ve iki STOP biti (mark = 1) eklenerek bilgi tamamlanır:

0	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**1**
START	←			"K" harfi			→	Tamlama biti	STOP	

---

4 /a - Sayfa ilerleme (FF): (10 tabanlı) 12
b - X, Y harfleri: (10 tabanlı) 88, 89
c - Satır ilerleme (LF): (10 tabanlı) 10
d - x, y küçük harfleri: (10 tabanlı) 120, 121

Tam ileti şöyledir (10 tabanlı):

12	**88**	**89**	**10**	**120**	**121**	**12**
FF	**X**	**Y**	**LF**	**x**	**y**	**FF**
Güdüm işareti	Harfler		Güdüm işareti	Harfler		Güdüm işareti

Bu iletideki sayıların, ASCII kodları oldukları belirtilmeseydi, bilgiyazar için bir anlam taşımazlardı ve böylece bir hata yapılmış olurdu.

5 / Büyük harflerin kodları (10 tabanlı) 65 ilâ 90, küçük harflerin kodları ise 97 ilâ 122 arasındadır. Dolayısıyla büyük harfin koduna (10 tabanlı) 32 eklemek gerekir.

	65	A'nın 10 tabanlı değeri
+	**32**	Dönüştürme
	97	a'nın 10 tabanlı değeri

Aynı şekilde z için:

122 (z) = 90 (Z) + 32

---

6 / Belirli bitleri VE işleviyle birbirinden ayırmak için, ayrılacak bitlerin basamaklarında 1 rakamı olan bir maske gereklidir.
Maske şu şekilde belirlenir:

Bit sayısı	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Maske	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**

Maskenin 10 tabanlı değeri: $2^7 + 2^3 + 2^1 = 128 + 8 + 2 = 138$
İkinci kısmın çözümü için önce 12, 21, 6 ve 15 sayıları iki tabanlıya çevrilir. (Bkz. Tablo 1, s.64).

12 =	0	0	0	0	1	1	0	0
21 =	0	0	0	1	0	1	0	1
6 =	0	0	0	0	0	1	1	0
15 =	0	0	0	0	1	1	1	1

VE işlevi ve maskeyle, 12 sayısı için sonuç şöyle bulunur:

Bit sayısı	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
12 sayısı	0	0	0	0	1	1	0	0
Maske	1	0	0	0	1	0	1	0
12 AND maske	0	0	0	0	1	0	0	0

Benzer biçimde:

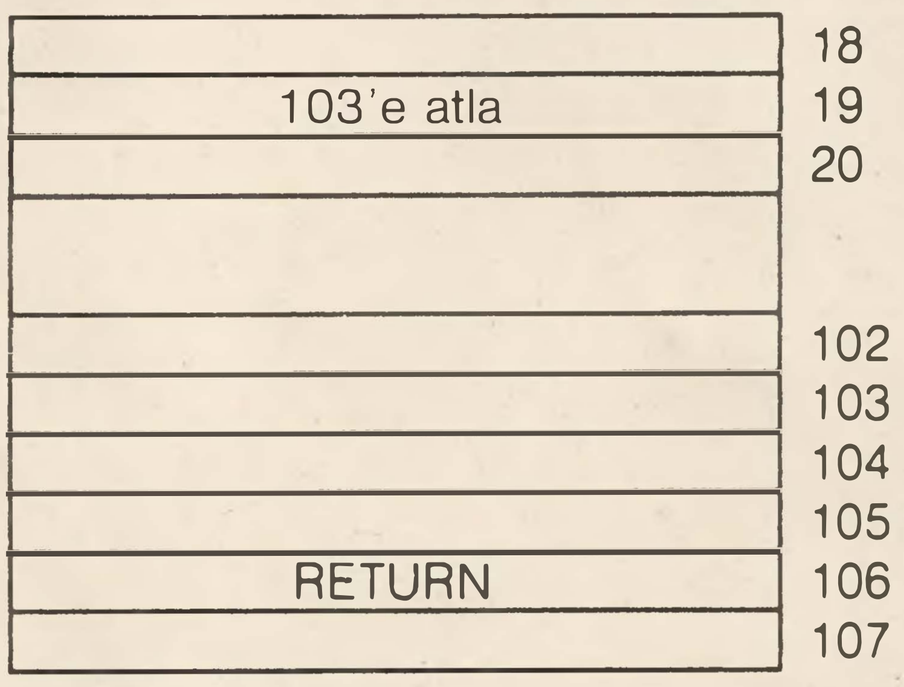
21 AND maske = 00000000
6 AND maske = 00000010
15 AND maske = 00001010

TEST 4

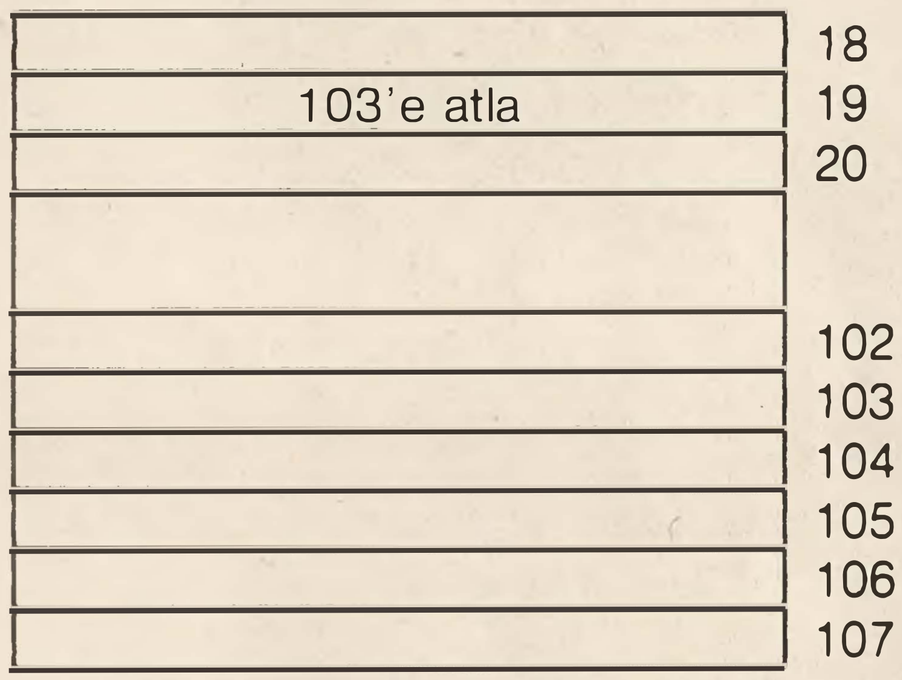
1 / Program sayacının işlevi aşağıdakilerden hangisidir?
a - İşlenmiş komutların sayısını tutar.
b - İşlenecek bir sonraki komutun adresini gösterir.
c - Toplama işlevi için yardımcı bir bellektir.

---

2 / Aşağıda iki bellek haritası görülüyor. Her iki durumda adres 103'e gelindikten sonra komutların hangi sırayla işleneceğini belirtiniz.



a



b

---

3 / Aritmetik işlemler MİB'in hangi bölümünde yapılır?
a - Deste bellekte
b - AMB'de
c - Program sayacında

---

4 / İşletim sistemi nedir?
a - Çevresel gereçlerin güdümü gibi temel işlevleri yürüten belirli sayıda program.
b - Hesap işlemlerini yürüten proramlar.
c - Bilgisayarın, doğrudan bellek erişiminde veri iletişimini denetleyen özel birimler.

---

5 / Doğrudan bellek erişimi ne işe yarar?
a - Kullanıcının bellekle doğrudan bağlantı kurmasını sağlar.
b - MİB tarafından belleğin içeriği değiştirilir.
c - MİB'in katkısı olmaksızın veri iletişimini sağlar.

---

6 / Aşağıda yine bir bellek haritası görüyoruz. Prgram konum 31'den başlıyor. Bir yanlış olup olmadığını, varsa ne olduğunu belirleyiniz:

A'ya 12'yi yerleştir	31
57'ye atla	32
RETURN	33
	56
RETURN	57
	58
	59

7 / Araya girmenin tanımı hangisidir?

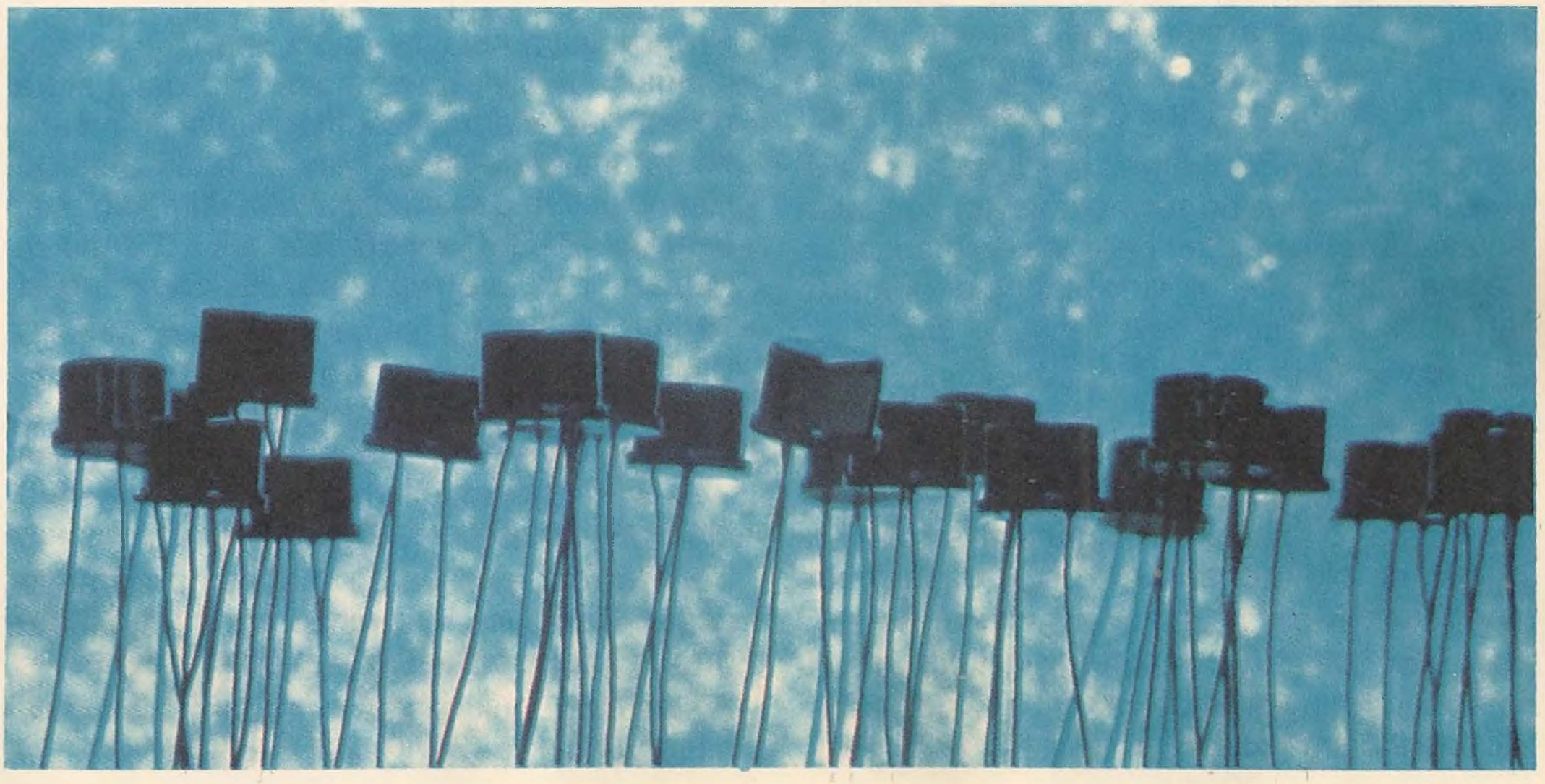
a - MİB'in işleyişine ara verilir.
b - Dışarıdan bir talep gelir gelmez programa ara verilir.
c - Bellekle veri iletişimine ara verilir.

8 / Aşağıdaki ifadelerden hangisi doğrudur?

a - Silinebilir Programlanabilir Salt Okunur Bellekte (EPROM) program saklanamaz.
b - Bir disket bellek biriminde yaklaşık 64.000 simge bulunur.
c - Salt Okunur Belleklere (ROM) program komutuyla değer yazdırılamaz.
d - Bir araya girme komutu verildiğinde MİB, HAZIR (HOLD) durumuna geçer.
e - Ekran ve klavye üzerinden araya girme komutu verilemez.
f - Disk bellek birimiyle veri iletişimi normal olarak doğrudan bellek erişimi yoluyla gerçekleşir.

Testin çözümleri s.168'dedir.

Günümüzün sayısal sistemlerinde transistörlerin yerini tümleşik devreler alıyor.



Programlama

Bir programın hazırlanması için, yapılacak bütün işlemlerin önce bilgisayarın anlayabileceği bir biçime çevrilmesi gerekir.

Örneğin, bir hesap makinesiyle problem çözmeye uğraşan bir kullanıcı makineye, hangi işlemin yapılacağını tuşlara basarak bildirir. İç devreler tuş darbelerini yorumlar ve gerekli sinyalleri üretir. Aynı işlemler ters yönde de tekrarlanır; sonucu bildiren sinyaller elektronik devreler tarafından sayılara çevrilip göstergeye iletilir.

Programlanabilen cihazlarda ise, hangi işlemlerin yapılacağının her defasında tek tek bildirilmesi gerekmez; işlemler belleğe yazılır ve istendiğinde, BAŞLA komutundan başka bir şeye gerek duymaksızın istenen işi yürütürler. Programlanabilen cihazların çözebileceği problemler, basit bir aritmetik işleminden çok daha karmaşıktır.

Bilgisayarın asıl işlevi, çok büyük miktarda verilerin işlenmesidir. Bu yüzden bilgisayar, verileri okuyabilmeli, saklayabilmeli, aralarından gerekeni seçebilmeli ve onların üzerinde gereken işlemi yapabilmelidir. Tüm bu işlemler, bilgisayara hangilerinin nasıl bir sırayla yürütülebileceğini gösteren bir ''program''la yerine getirilir.

Örneğin bir telefon rehberinde ''Seçkin'' soyadını arama programı şu adımları izlemek zorundadır:

1 / Rehberden bir soyadı oku.
2 / Bu soyadı ''Seçkin'' ise ''bulundu'' yaz ve arama işine son ver.
3 / Okunan soyadı ''Seçkin'' değilse 1. adıma geri dön (ve bir sonraki soyadını oku).

Ancak bilgisayar bu tür betimleyici cümleleri anlayamaz. Bu yüzden belleğe yazılmadan önce komutların elektrik sinyallerine dönüştürülebilecek bir biçime çevrilmesi gerekir.

Bilgisayar için en basit ifade biçiminin iki tabanlı sayı sistemi olduğunu söylemiştik. Oysa insanlar için en kolay yol, nasıl bulunması gerektiği üzerinde fazla kafa yormayı gerektirmeyecek bir biçimde, bulunması istenen sonucun ne olduğunun açık seçik bir biçimde yazılmasıdır.

Ancak bu iki aşırı uç arasında geliştirilmiş başka iş bildirme yolları da vardır. İşte bu yollara ''Programlama Dilleri'' denir.

Programlama Dilleri

Kullanılacak programlama dili çözülecek problemin niteliğine bağlı olarak seçilebilir. Bazı durumlarda bu seçimin fazla bir anlamı yoktur. Hangi dilin seçileceğini belirleyen etkenler, verilerin işlenmesinde zaman tasarrufu sağlamak ya da kullanılacak komut sayısını sınırlamak olabilir. Hatta zaman zaman, verimliliği azaltma pahasına kişisel tercihler bile belirleyici olabilir. Ancak belirli durumlarda ise diller arasında seçim yapmak mümkün değildir. Belirli problemler, ancak o tür probleme uygun bir dille çözülebilir.

Programlama dillerinin hepsi ''simgesel''dir, yani bilgisayarın yapacağı işlemleri insanların da anlayacağı simgelerle ifade ederler. Bu simgeler, özel programlar yardımıyla bilgisayarın anlayabileceği bir biçime çevrilirler. Bu programlara ''çevirici'' ya da ''derleyici'' adları verilir.

Simgesel dillerden başka, ''mutlak'' programlama dilleri de vardır. Bu dillerde komutlar iki tabanlı olarak ifade edilir. Mutlak diller kullanıcı programlarının hazırlanmasında değil, çok özel uygulamalarda kullanılırlar.

Programlama dilleri başlıca iki gruba ayrılır:

- **Yüksek düzeyli programlama dilleri**
- **Alçak düzeyli programlama dilleri**

Her iki türün de kendine özgü nitelikleri ve uygulama alanları vardır.

Yüksek Düzeyli Programlama Dilleri

Programcıya kolaylık sağlamak üzere hazırlanmış tüm diller bu gruba girer. Komutlar yapılacak işlemin İngilizce karşılığını anımsatıcı kodlardır. Örneğin bir simgenin ekranda görünmesini ya da kâğıda yazılmasını sağlayan **PRINT**...(BASIC) ve **WRITE**... (FORTRAN) komutları ''bas'' ve ''yaz'' sözcüklerinin İngilizce karşılıklarıdır.

Bilgisayar, komutları doğrudan bu biçimiyle işleyemez, çünkü MİB sadece sınırlı sayıdaki temel komutları tanır (bkz. ''MİB komutları'', s.146). Bu nedenle komutların, temel komutlara çevrilmesi gerekir.

Çeviriyi, derleyici ve çevirici adlı programlar yapar. Derleyici, kullanıcı programının tümünü, çevirici ise tek tek komutları çevirir.

Kişisel ve küçük bilgisayarlarda en çok kullanılan yüksek düzeyli diller şunlardır:

• **BASIC** • **FORTRAN** • **COBOL**

Daha az tanınan diğer dilleri İşletim Sistemleri bölümünde göreceğiz. FORTRAN çoğunlukla teknik ve bilimsel, COBOL ise işletme ve yönetim alanlarında uygulanır. BASIC dili ise öğrenme kolaylığı nedeniyle, kişisel bilgisayar sistemlerinde her iki alanda da kullanılır.

FORTRAN ve COBOL genellikle derleyici, BASIC ise daha çok çevirici tarafından uygulanır. Herhangi bir programlama dilinde yazılmış kullanıcı programına **"Ham Program"** (ya da kullanıcı programı) denir.

Yandaki şekilde, hem çevrilen hem derlenen dillerde yazılmış programların hazırlanmasındaki aşamalar gösterilmiştir.

Çevrilen dillerde yazılmış ham programlarda her komut, yürütme sırası geldiğinde yorumlanır, yani makinenin anlayacağı bir biçime çevrilir. Bu yüzden işleyiş biraz yavaştır. Ancak bu sayede hataların daha program girişi sırasında bulunması mümkündür. İşletim sistemi, tuşlarla verilen komutları belleğe yazar. Bu sırada yanlış tuşa basmak gibi dikkatsizliğin neden olduğu hataları, çevirici hemen gösterir ve anında düzeltme yapmayı mümkün kılar. Derlenen dillerde ise böyle bir yanlışın yapıldığı durumlarda çevirinin bitmesini beklemek gerekir. Bu tür hatalara biçimsel hatalar denir. Çevrilen dillerde çevirinin bitmesini beklemek gerekmediğinden biçimsel hatalar çok daha çabuk düzeltilebilir ve programın işleyişine zaman kaybetmeden devam edilir.

İki dil türünün en önemli niteliklerini şöylece özetleyebiliriz:

Çevrilen Dil: Ânında hata düzeltme; daha yavaş program yürütme.

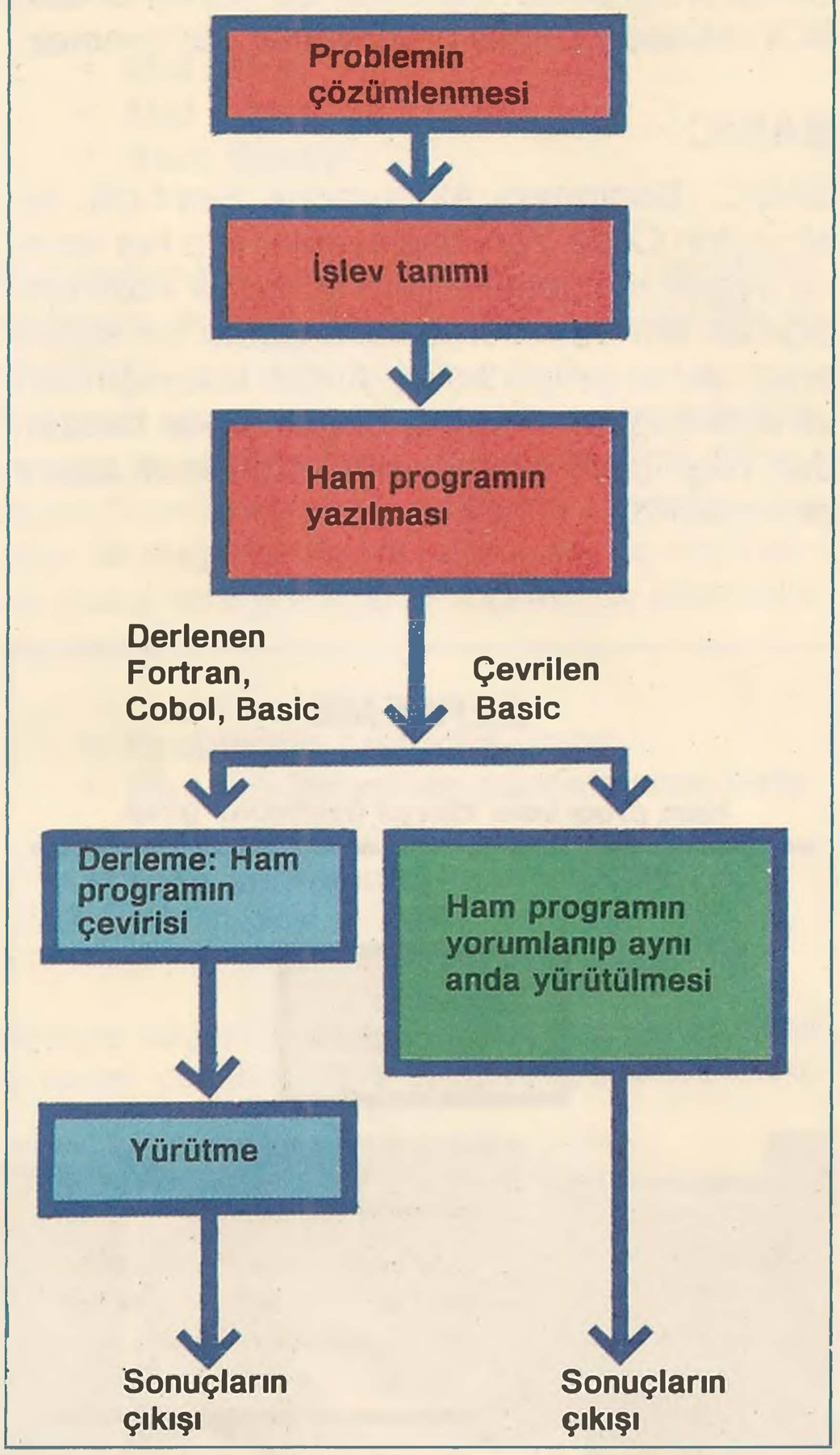
Derlenen Dil: Her düzeltme için yeni bir çeviri, dolayısıyla yeni bir bekleme süresi; ancak daha hızlı program yürütme.

Kuşkusuz hem çevrilebilen, hem de derlenebilen bir dil, ideal dil olurdu.

Bir çeviricinin bulunması, kullanıcı programının hazırlanmasında çok yararlıdır. Deneme ve düzeltmeler çevirici sayesinde zaman kaybetmeden gerçekleştirilebilir. Kullanıcı programı deneme evresi sonunda düzeltilmiş şekliyle derlendiği takdirde hızlı bir program elde edilir.

BASIC, yeni başlayanların dili olarak ortaya çıktığından programlama öğrenmeyi çok kolaylaştırır. Bu nedenle kişisel bilgisayarlarda en çok kullanılan dildir.

BİR PROGRAMIN HAZIRLANIŞI VE YÜRÜTÜLMESİ



Derleme

Bir ham programı derleme süreci iki adımdan oluşur:

- Önce bir ara biçim bulunur (çevrilmiş, ancak yürütmeye hazır olmayan programlar- **çevrilmiş program**).
- İkinci adımda ara biçimin çeşitli parçaları birleştirilir. Bu işleme Bağlama **(Link)** adı verilir. Sonuçta **derlenmiş program** (ikincil program, yürütülebilir program) ortaya çıkar.

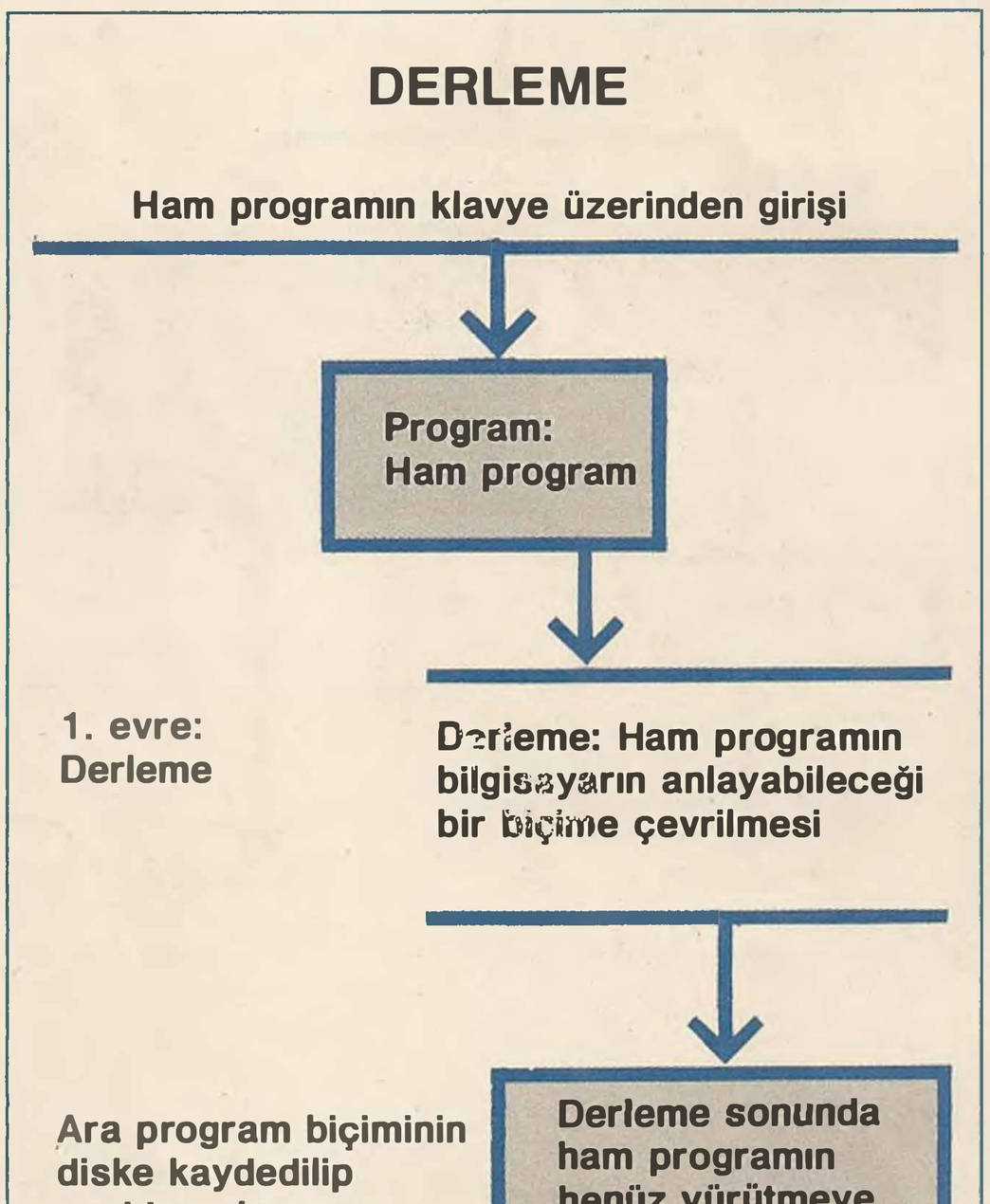
Derleyici sistemlerde, ara program biçiminin (yani çevrilmiş programın) kaydedilebileceği fazladan disk bellek birimlerinin de bulunması gerekir. Çevrilen diller için ise bazı durum-

larda bu koşul ortadan kalkar, çünkü çevirici, doğrudan doğruya bellekte yer alabilir.

Ayrıca, her yüksek düzeyli dil için ayrı bir derleyici gerekir. Örneğin FORTRAN dilinde yazılmış programları derleyen bir derleyici COBOL dilinde yazılmış programları derleyemez.

BASIC

BASIC (Beginners All-purpose Symbolic Instruction Code-Yeni başlayanlar için her amaca uygun simgesel eğitim kodu) ilk hazırlandığında üniversitelerde kullanılacak bir eğitim aracı olarak geliştirilmişti. Ancak kolaylığından ve esnekliğinden ötürü, birçok şirket tarafından bilgi-işlem ihtiyaçlarını karşılamak üzere benimsendi.



Bugün BASIC hem bilimsel, hem de ticari alanda uygulanabilen bir dildir, çünkü FORTRAN 66'da olmayan bazı özellikleri vardır.

Disket verilerini işleme yeteneği COBOL'unki kadar olmasa da, kişisel ve küçük bilgisayarlar için yeterlidir.

BASIC diğerlerine göre oldukça yeni bir dil olup, herkes tarafından aynı biçimde uygulanmamaktadır. Komutlar ve simgeler, cihaz üreticilerine bağlı olarak bazı farklılıklar göstermektedir. Gerçi 1978'de bir BASIC standardı üzerinde anlaşmaya varıldı ama bu standart BASIC komutlarının çok az bir kısmını kapsıyor.

FORTRAN

Bilimsel amaçlara yönelik olarak hazırlanmış olan FORTRAN (FORmula TRANSlator - Formül çevirici) dilinin başlıca özellikleri şunlardır:

— Az sayıda temel komut gerektirir.
— Problemin metin işleme payı az olmalıdır.
— Kullanılacak sayıların üstel yazı biçiminde gösterilmiş olması gerekir.
— İşleyebileceği veri miktarı genellikle azdır, ama daha kapsamlı işlemler yapılabilir.
— Belirli hesaplama araçları (matematiksel algoritmalar) mevcut olmalıdır.

Bu dili işletmelerde de uygulamak mümkündür, ancak harfsayısal simge katarlarını işleyebilecek komutların olmaması işi zorlaştırır.

Standart FORTRAN'ın (FORTRAN 66) bu sakıncaları, FORTRAN 77 dilinin geliştirilmesiyle hemen hemen tamamiyle ortadan kalkmıştır.

COBOL

COBOL (COmmon Business Oriented Language - Yaygın işe yönelik dil) geliştirilirken İngilizceyi andırması özellikle öngörülmüştür. İşletmeler için için hazırlanan bu dilin başlıca özellikleri şunlardır:

— Temel işlemler dışında hesap komutları yoktur.
— Veriler kolaylıkla disklere kaydedilebilir (kütükler).
— Metin yazımına elverişli komutları vardır.

Çözümleme ve Akış Şemaları

Bir problemi çözmeye çalışırken programı oluşturan komutların yazılması, problemi bilgisayarın çözebileceği bir biçime getirme çalışmasının son evresidir.

Problemin çözümlenmesi sırasında, önce karmaşık işlemlerin bir dizi temel işleme indirgenmesi gerekir. Bu işlemlerin sırası bir şemada belirli simgelerle gösterilir. Akış şeması denilen bu şemanın hazırlanmasından sonra komutların yazılmasına geçilir. Bu bölümde problem çözümleme ve akış şeması hazırlama yöntemleri ile akış şeması örneklerini göreceğiz.

Bir Programın Hazırlanışı

Her özel uygulama için bilgisayara ayrı bir program verilmesi gerekir. Karmaşık problemlerde sonuca ulaşmak için birden fazla program hazırlanabilir. Böyle programlara ''bileşik program'' adı verilir.

Stok yönetimi örneğini ele alalım. Burada bizi ilgilendiren üç konu vardır:

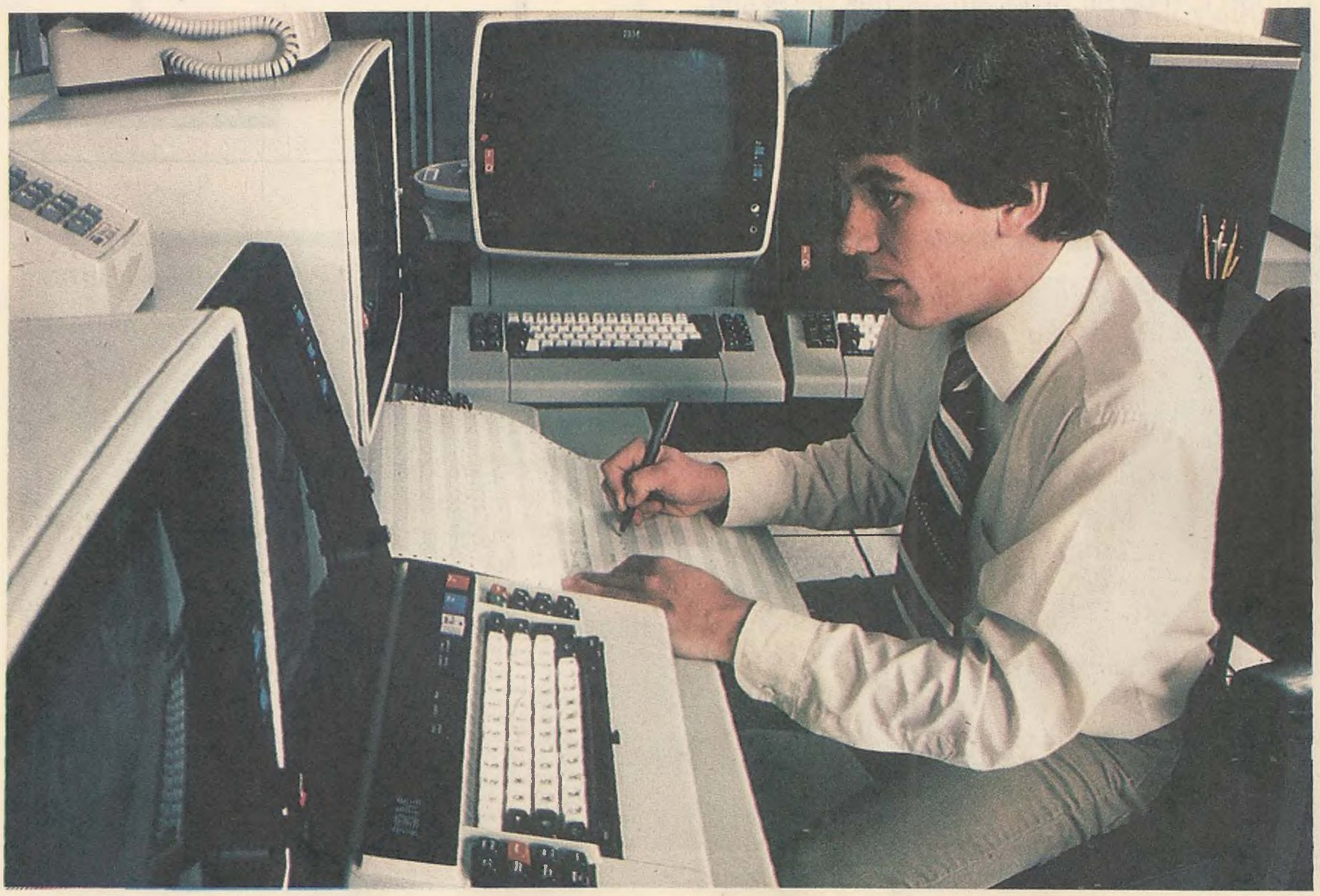
- **Mal girişi**
- **Mal çıkışı**
- **Stok düzeyi**

Bunlardan her biri için bir program yazılabilir ve hepsi birlikte Stok Yönetimi Programı'nı oluşturur.

Program yazmadan önce problemi iyi anlamak ve çözüm yöntemlerini bilmek gerekir. Programın hazırlanışı titiz bir çalışma gerektirir. En ufak bir değerlendirme hatası, yanlış veya eksik sonuçlara yol açar. Program üç aşamada hazırlanır:

- **Problemin çözümlenmesi**
- **Bireşim (program parçalarının birbirine bağlanması)**
- **Akış şemasının hazırlanması**

Yüksek bellek kapasitesi ve hızıyla bugün bilgisayar sigorta şirketlerinden avukat yazıhanelerine kadar çeşitli işletmelerde vazgeçilmez oldu.



Çözümleme

Her program yalnızca bilgisayara verilmiş verileri işleyebilir. Stok düzeyi örneğinde program ancak tüm mal giriş ve çıkışları bildirildiği takdirde doğru sonucu bulur. Ayrıca malın cinsi, önceden bellekte belirtilmeli, giren ve çıkan mallar, bilgisayarın tanıdığı (yani önceden belirtilmiş) cinsten olmalıdır.

Çözümlemenin ilk adımı, hangi sonuçların istendiğini ve verilerin yeterli olup olmadığını belirlemektir. Örnek olarak, stok düzeyinin aşağıdaki gibi gösterileceğini varsayalım:

1 / Mal cinsi kodu
2 / Malın tanımı
3 / Stok düzeyi
4 / Birim fiyat
5 / Stok toplam değeri

Yalnızca mal çıkışı ile ilgileniyorsak, bileşik programın tümünü oluşturan programlardan sadece bu kısmını hazırlamamız yeterlidir (bkz. şekil).

Örneğimizde ilk girdiler şunlardır:

1 / Mal cinsi kodu
2 / Malın tanımı

Bu verileri daha sonra kullanabilmek için, onları önceden bilgisayara vermemiz gerekir. Bunun için de bir program vermek gereklidir.

Bundan sonraki iş, her maldan stokta ne kadar bulunduğunun belirlenmesidir. Herhangi bir malın stok düzeyi iki duruma bağlı olarak değişebilir:

- Bir malın dışarı çıkması (mal çıkışı)
- Yeni bir malın girmesi (mal girişi)

Demek ki, çıkan mal miktarını stok düzeyinden çıkaracak, giren mal miktarını ise ekleyecek bir program gerekiyor.

Dördüncü sıradaki birim fiyatı, giren mal miktarıyla birlikte bilgisayara, özellikle de giren malın stok düzeyini belirleyen programa verilmelidir.

Demek ki üçüncü ve dördüncü işler için iki programa gerek var:

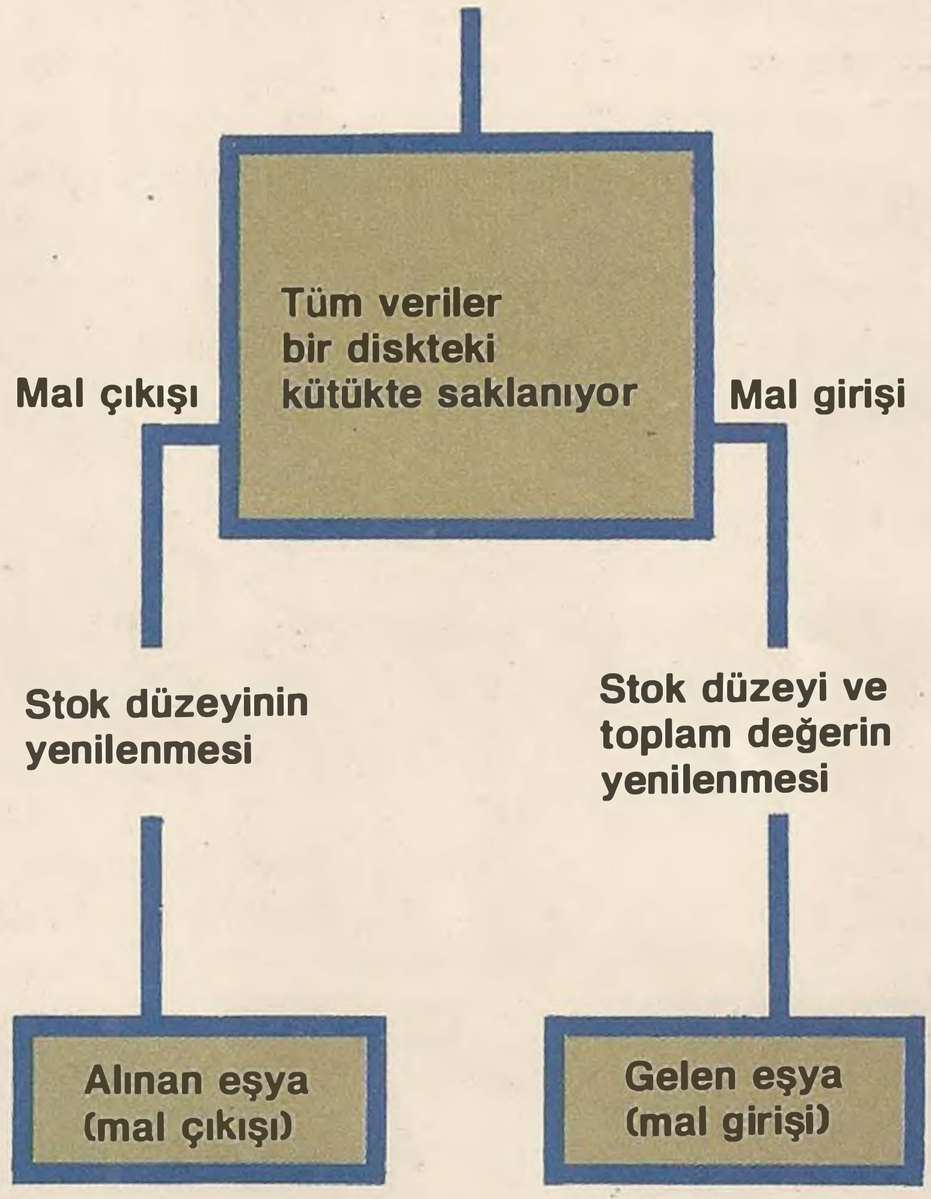
- Mal çıkışı
- Mal girişi

Beşinci madde ise, istenen hesaplamaları yapmak için buraya kadar verilenleri kullanan bir program gerektiriyor (stok düzeyinin toplam değeri).

Şimdi çözümlemenin sonucu gereken programın başlıca evrelerini bir tabloda gösterebiliriz:

1 / Stok düzeyi belirlenmesi (mal cinsi, kodu ve tanımı)
2 / Mal girişlerinin belirlenmesi (son stok durumunun hesaplanması)
3 / Mal çıkışlarının belirlenmesi (son stok durumunun hesaplanması)
4 / Stok durumunun bir tablo halinde basılması (toplam değerin hesaplanması)

STOK YÖNETİMİ PROGRAMININ MANTIK ŞEMASI



Gerekli programlar ve onlarla ilgili veriler aşağıda sıralanmıştır:

PROGRAM	VERİLER
Stok düzeyi belirlenmeli	Mal cins kodu ve tanımı
Mal girişi	Stok düzeyi ve malların birim fiyatı
Mal çıkışı	Stok düzeyi
Sonuçların bir tablo halinde basımı	Toplam değer

Bu programların her biri için gereken başlıca işlemler şöyle özetlenebilir:

Stok Düzeyinin Belirlenmesi: Kullanıcı, stoktaki her malın cinsini ve tanımını belirten kısaltılmış bir kod vermelidir.

Mal girişi: Burada amaç, giren her malı eklemek ve toplam değerin son durumunu bulmaktır. Bunun için şu işlemler yapılır:
Son miktar = önceki miktar + giren miktar
Fiyat = son giren malın fiyatı

Mal çıkışı: Depodan mal alındığında yeni stok düzeyi belirlenir:
Son miktar = önceki miktar—çıkan miktar

Tablonun basımı: Bu program, stok düzeyini kâğıda geçirmeden önce, her mal cinsi için toplam değeri hesaplamalıdır.
Toplam değer = stok düzeyi × birim fiyatı
(5. veri = 3. veri × 4. veri)

Bu işlemler sayfa 160 ve 161'deki şemalarda gösterilmiştir.

1, 2, 3 ve 4. maddelerde belirtilen veriler bir yığın-bellekte saklanır. Bu iş için en elverişli olacak bilgisayarı, gereken bellek kapasitesi, hız ve aynı anda çalışması gereken kullanıcı sayısı belirler. Hangi tür bir bilgisayar kullanılması gerektiğine, birinci evre tamamlandıktan, yani çözümleme, bireşim ve akış şeması hazırlandıktan sonra karar verebiliriz. Bu noktada üç tür bilgisayar ele alınabilir:

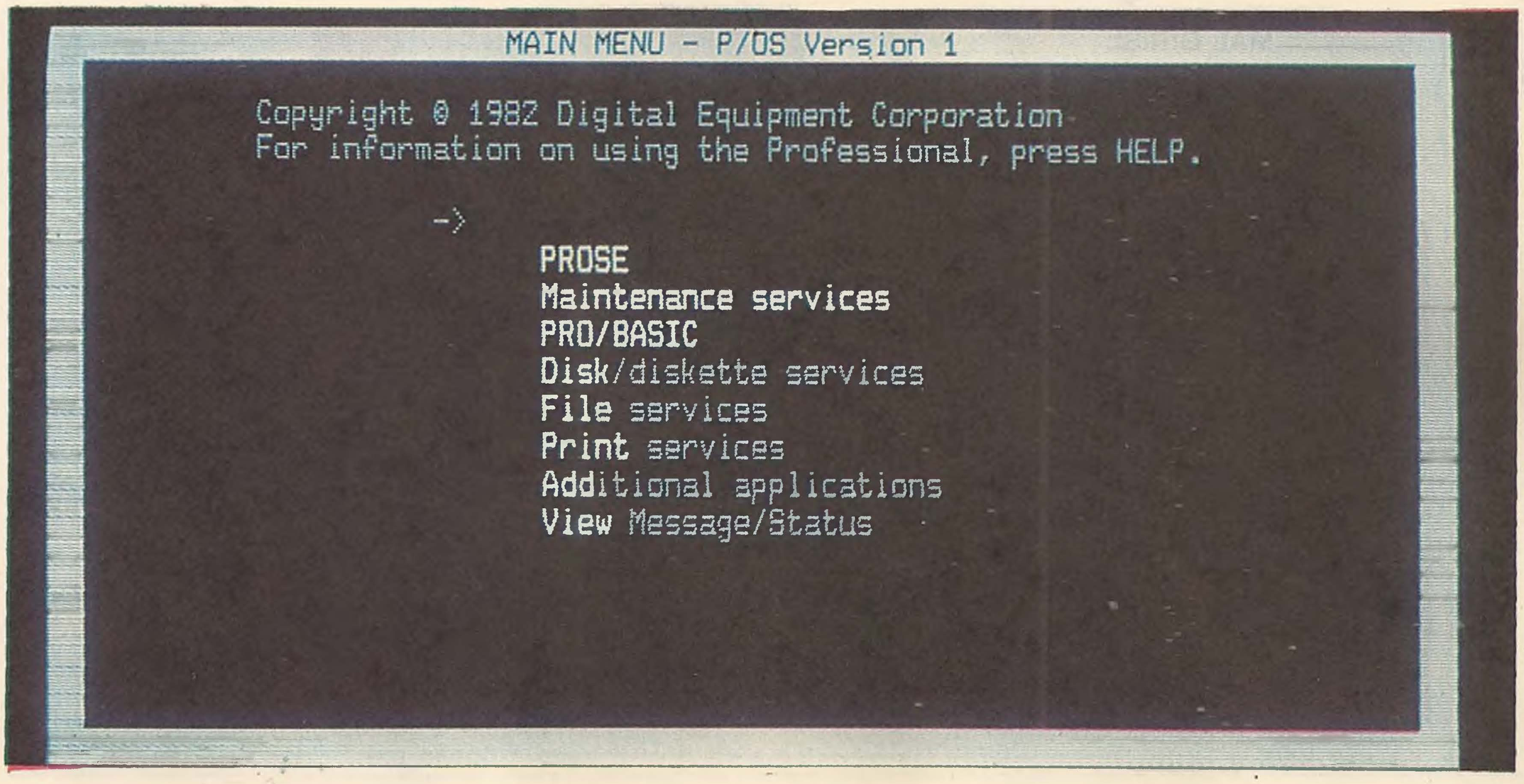
Kişisel bilgisayar: Tek kullanıcı için - Sınırlı miktarda veri belleği - Düşük hız - Düşük fiyat - Az genişletme imkânı.

İş bilgisayarı: Bir veya nadiren az sayıda kullanıcı - Daha kalıcı veriler - Yüksek işlem hızı - Vasat fiyat - Sınırlı genişletme olanağı.

Küçük bilgisayar: Birçok kullanıcının aynı anda değişik programlarla çalışabilme imkânı - İşlemlerde orta yükseklikte hız - Daha yüksek fiyatlar - Önemli ölçüde genişletme olanağı.

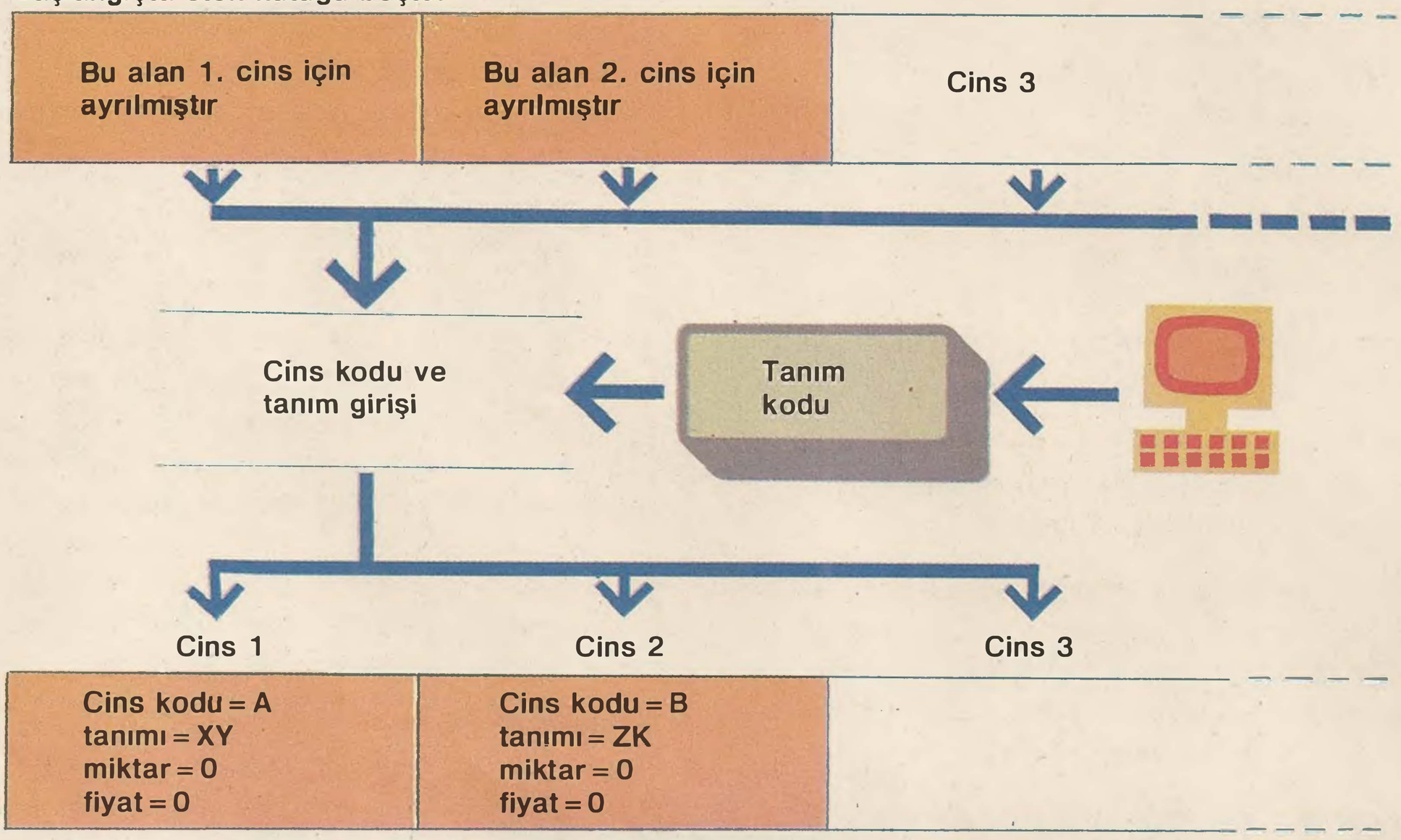
Bilgisayar seçerken gelecekteki ihtiyacı da düşünmek gerekir.

Bilgiler, bilgisayar ekranında görünür hale getirilir.



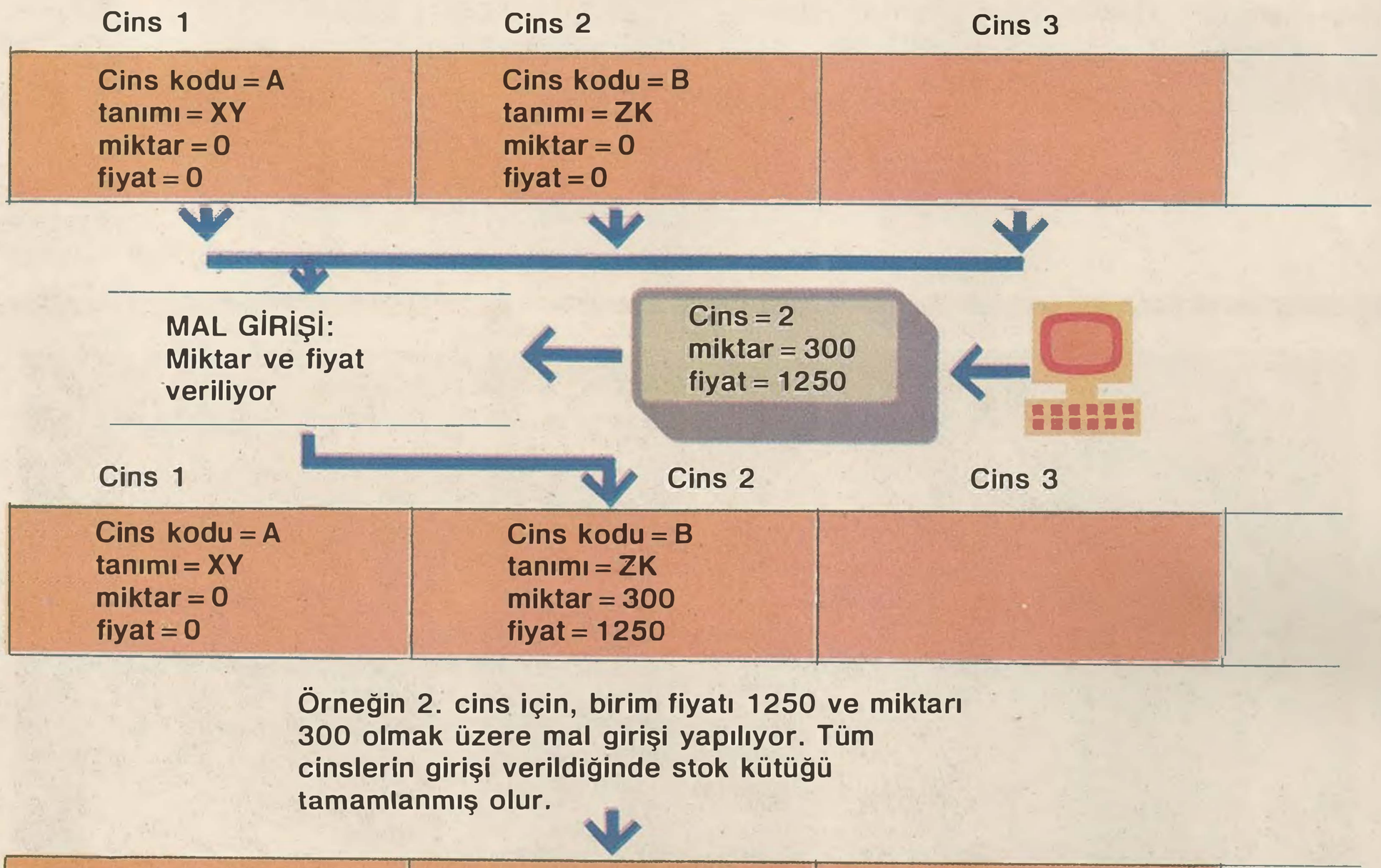
STOK DÜZEYİNİN BELİRLENMESİ

Başlangıçta stok kütüğü boştur



Stok kütüğüne her cinsin kodu ve tanımı verilir. Henüz mal girişi olmadığından miktar ve fiyat sınırlıdır.

MAL GİRİŞİNİN KAYDEDİLMESİ



Örneğin 2. cins için, birim fiyatı 1250 ve miktarı 300 olmak üzere mal girişi yapılıyor. Tüm cinslerin girişi verildiğinde stok kütüğü tamamlanmış olur.

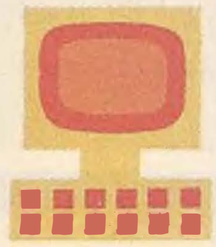
Cins kodu = B tanımı = ZK miktar = 300 fiyat = 1250	Cins kodu = A tanımı = XY miktar = 35 fiyat = 270	Cins kodu = C tanımı = LMN miktar = 20 fiyat = 500

MAL ÇIKIŞININ KAYDEDİLMESİ

Cins kodu = A tanımı = XY miktar = 35 fiyat = 270	Cins kodu = B tanımı = ZK miktar = 300 fiyat = 1250	Cins kodu = C tanımı = LMN miktar = 20 fiyat = 500

MAL ÇIKIŞI:
Çıkan miktarın
verilmesi

Cins 3
cins kodu = C
çıkan
miktar = 6



Stok düzeyi = önceki miktar-çıkan miktar
Stok düzeyi = 20—6 = 14

Cins kodu = A	Cins kodu = B	Cins kodu = C tanımı = LMN miktar = 14 fiyat = 500

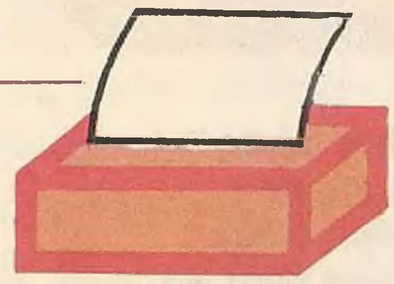
Mal çıkışında, alınan miktar stok düzeyinden çıkarılır ve buna göre kütük durumu yenilenir.

STOK DÜZEYİNİN BASIMI

Cins kodu = A tanımı = XY miktar = 35 fiyat = 270	Cins kodu = B tanımı = ZK miktar = 300 fiyat = 1250	Cins kodu = C tanımı = LMN miktar = 14 fiyat = 500

Bütün kütük okunur. Her cins için toplam değer hesaplanır:
Toplam değer = fiyat × miktar

Her mal cinsinin toplam değeri toplanarak genel toplam değer bulunur.



Cins kodu	Tanımı	Birim fiyatı	Miktar	Değer
A	XY	270	35	9450
B	ZK	1250	300	375000
C	LMN	500	14	7000
			GENEL TOPLAM	391450

Çözümlemenin başlıca aşamaları şunlardır:

- **Gerekli bütün verilerin sağlanmakta olduğu saptanmalıdır.**
- **Sonunda kullanılacak olan program tanımlanmalıdır.**
- **Hangi tür bir bilgisayarın kullanılacağına karar verilmelidir.**

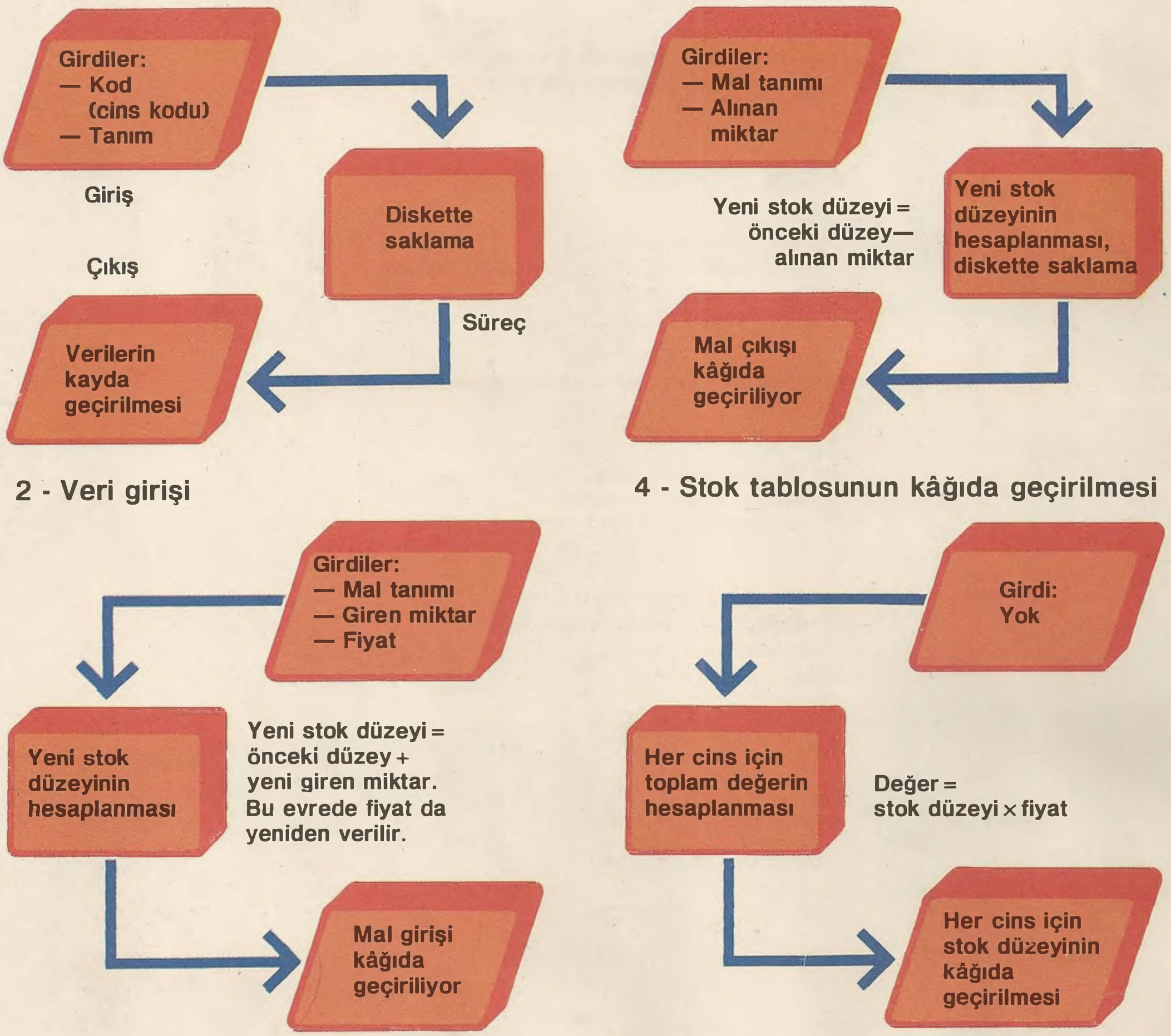
Bireşim

Bireşim evresinde, bir işlem dizisinin programlarının çalışma yöntemlerini gösteren şemalar hazırlanır. Bu aşamada en iyi yöntem HGSÇ tekniğinden yararlanmaktır (Hiyerarşik giriş-süreç-çıkış: HIPO = Hierarchy plus Input-Process-output). Bu teknikle her program veya program parçası üç kısma ayrılır.

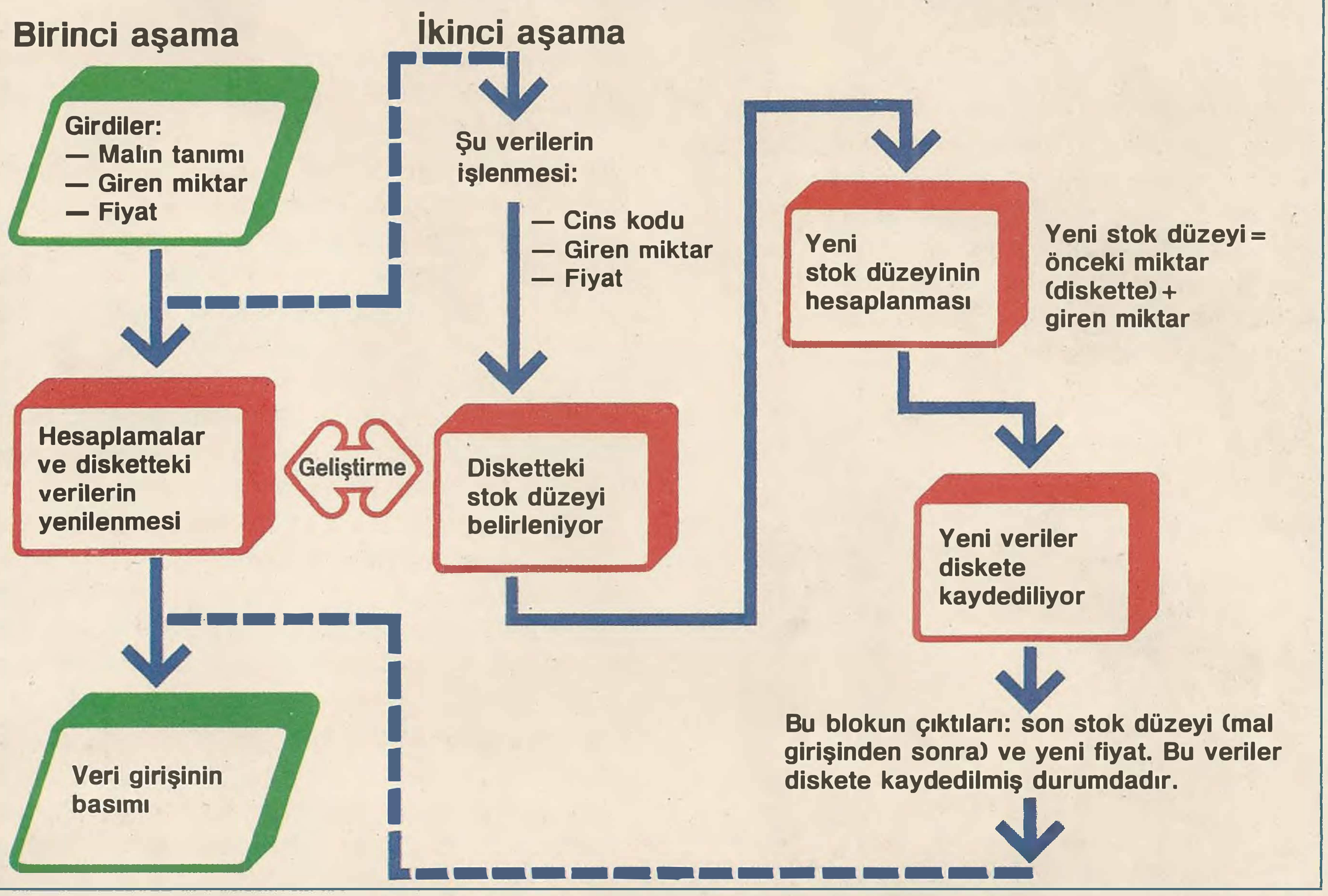
Birinci Blok: Veri Girişi (Giriş) - Program için gerekli olan tüm veriler tanımlanır.

İkinci Blok: Veri İşleme Süreci (Süreç) - Verilerin nasıl işleneceği gösterilir.

STOK YÖNETİMİ PROGRAMI



ÖRNEK: MAL GİRİŞİ İÇİN BİR ŞEMANIN GELİŞTİRİLMESİ



Üçüncü Blok: Veri Çıkışı (Çıkış) - Öngörülen tüm çıktılar tanımlanır.

Basit programlar için bir tek HGSÇ şeması yeterlidir. Karmaşık programlar ise, farklı ayrıntı düzeylerinde çeşitli şemaların hazırlanmasını gerektirir. HGSÇ yöntemi, çalışma sürecini kolaylaştırması bir yana, program dökümü için de en iyi yoldur. (Stok Yönetimi Programının HGSÇ şeması için bkz. s.162.)

Bloklar adım adım basitten karmaşığa doğru geliştirilir. Mal girişi örneğinde, programın süreci şöyle genişletilebilir:

a / Her mal kodu için stok düzeyi, disket bellekten getirilir.
b / Yeni giren mallar eski stok düzeyine eklenir.
c / Eski birim fiyatının yerine yeni değer verilir.
d / Bellekte, eski verilerin yerine yenileri kaydedilir.

Bu işlemlerin tümü belirli bir mal cinsi ile ilgilidir; girdi de zaten mal koduna göre alınmıştır. Veri işleme sürecinin sonunda, disketteki bilgiler yenilendiği için veri çıkışı işlemi yapılabilir.

Şekilde, hem şemanın birinci aşamadaki durumu hem ikinci aşamada geliştirilmiş biçimi görülüyor. İkinci aşamada programı yazabilmek için gereken ayrıntılar henüz yoktur. Bunun için bir aşama daha gerekir. Böyle bir şemanın hazırlanması, bir sürecin tüm ayrıntılarıyla görülmesini kolaylaştırır.

Şemada tüm ayrıntılar verilmemiştir. Dolayısıyla bu şekle göre hazırlanan bir programda kimi hatalar olabilir. Örneğin böyle bir program varolmayan, yani önceden tanımlanmamış olan bir mal kodu verildiği takdirde bir hataya yol açabilir. Böylesi hataları önlemek için girişe bir sınama birimi eklenebilir. Ayrıca çıkışta birkaç satırın başlık için boş kalacağı da şemada gösterilmemiştir.

İşlemleri bu ayrıntıda gösteren şekle AKIŞ ŞEMASI (Flow-chart) denir. Bu şema programın yazılmasından önceki son aşamadır.

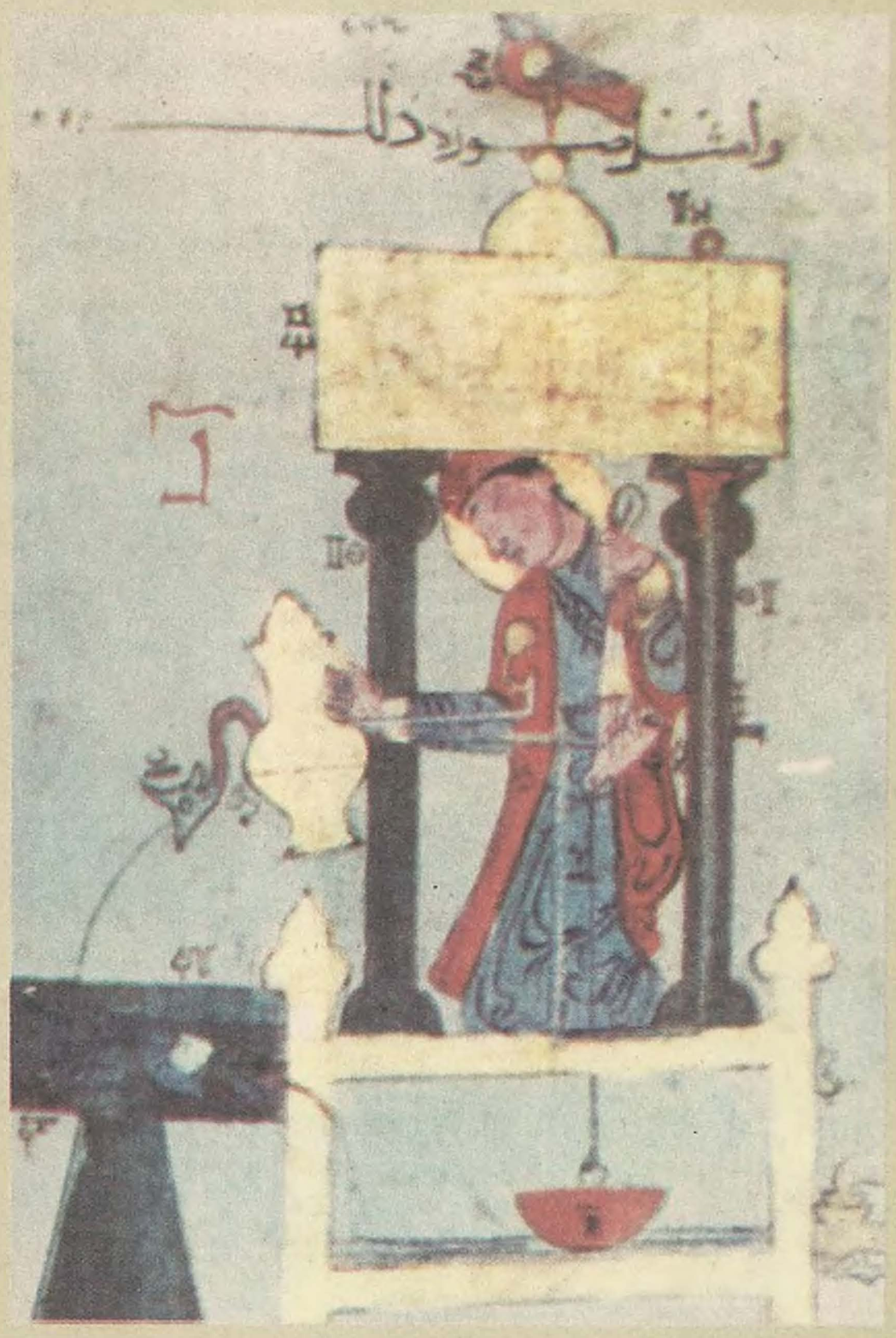
Eb-Ül-İz'in "Robot Adamları"

Selçuklu Sultanı Alpaslan'ın 1071 yılında Bizans İmparatoru Romanos Diogenes'i, Malazgirt'te yenmesi ile Selçuklu boylarına Anadolu kapıları açılmıştı. Doğu'dan Batı'ya, Kuzey'den Güney'e doğru dalga dalga akan Selçuklu boyları, beraberlerinde getirdikleri teknik ve kültür ile fethettikleri yerlerdeki teknik ve kültürleri karşılaştırarak yepyeni bir teknik ve uygarlık yaratıyorlardı. Türkler, kendilerine yeni vatan olarak seçtikleri Anadolu'nun her köşesine, bu yepyeni uygarlık ve teknolojiyi götürüyorlardı. Eskimiş Bizans kültür ve teknolojisi yerine Selçuklu kültür ve teknolojisi yükselmeye başlamıştı.

İran üzerinden Batı'ya doğru Türk boylarının bir kısmı Anadolu'nun ortasına kadar gelip Konya'yı başkent yaparken, Artuk Bey kumandası altındaki diğer bir Selçuklu boyu, Anadolu'nun güneyine doğru akıyordu. Güney Anadolu'yu da tamamen fetheden Selçuklular, Artukoğulları adı altında bir devlet kuruyorlar ve 1183 yılında da Diyarbakır'ı başkent yapıyorlardı. Diyarbakır hem Güney'e, hem Doğu'ya ve hem de Batı'ya giden yolların ortasında olduğundan, o tarihte çok önemli bir kültür ve teknik merkezi görevi de yapıyordu. Artuk Sultanları, çağın en ünlü bilginlerini saraylarına davet etmişler ve onlara, bilim ve teknolojideki buluşlarını ortaya koyabilmeleri için, bütün imkanları tanımışlardı.

Hemen hatırlatalım ki, aynı 1183 yılında Avrupa'da Engizisyon Mahkemeleri kuruluyor ve çağın bilginleri, dine karşı gelinmiştir gerekçesiyle yargılanıyor, işkenceye tâbi tutuluyor ve öldürülüyorlardı...

Selçuklu Türklerinin Sükman boyundan olup, Mardin'in Cizre kazasında kendi kendine buluşlar yapmaya ve otomatik sistemler kurmaya çalışan Abdülaziz İsmail bin Razzaz, Artukoğulları sarayında, kendisine tanınan imkânlardan yararlanarak, buluşlarını burada ortaya koyuyordu. Tam 25 yıl, Artukoğulları sarayında, ilginç otomat makineler meydana getiren bilgin, bu makineleri ile çağın ilerisine atladığını gösteriyor ve çağın harika bilgini Cizreli Eb-Ül-İz adı ile anılıyordu. Artuk Hükümdarı'nın önerisi üzerine de, bütün yaptıklarını (sistemlerin işleyişini resimler ve çizimlerle açıklayarak) gösteren bir kitap yazmıştı. 1206 yılında yazmış olduğu bu kitabında şöyle diyordu:



"Otomatik Apdest Alma Makinesi": Şekilde görülen, sütunlar üzerine oturtulmuş olan, büyük bir su deposu vardır. Bu deponun içindeki su, sağdaki sütun içinden, aşağıdaki havuz içine boşalmaktadır. Aşağıdaki havuza gelen su, bu havuz içinde bulunan şamandırayı harekete geçirmektedir. Şamandıra, yukarıya doğru yükselirken, sütunlar arasında bulunan "Robot Adam"ın, sağ kolunu harekete getirmektedir. Robot Adam da, sağ elinde tuttuğu ibriği, öne eğmektedir. Bu esnada, yukarıdaki depodan, incecik bir boru içinden gelen ve ibriğe dolmakta olan su, bu ibriğin ağzından dökülmektedir.

"...Ben, bu kitabı Artukoğullarından Diyarbakır Hükümdarı Eb-Ül-Feth Mahmut İbn-i Mehmet İbn-i Karaaslan adına yazdım. Ben, bu değerli hükümdarın babasına ve kardeşine hicretin 577. yılından beri (İsa'dan sonra 1183) tam 25 yıl hizmet ettim. Bir gün, yaptığım makinelerden birini göstermiştim. O, bu işimi büyük bir ilgi ile inceledi ve, '-Dünyada eşi bulunmayan bir şey yaptın. Emeğin boşa gitmeyecektir. Bana, bütün bu yaptıklarını gösteren ve içine alan bir kitap yaz...' dedi. Ben de bütün çabamı toplayarak, gücüm yettiğince ça-

lıştım ve bu kitabı yazarak kendilerine sundum...''

O tarihte Artuk Türkleri sarayında konuşulan bilim dili Arapça olduğu için, büyük bilim adamımız bu kitabını Arapça yazmıştı. Mevlana'nın, Konya Selçuk saraylarında konuşulan bilim dili Farsça olduğu için, Farsça yazması gibi.

Yukarıda, Anadolu'ya gelen Selçukluların, beraberinde getirdikleri teknik ve kültür ile fethettikleri yerlerde buldukları teknik ve kültürleri karşılaştırarak yepyeni bir uygarlık yarattıklarına değinmiştik. Artukoğulları sarayında görev yapan İbn Razzaz Cesari (Cizreli Eb-Ül-İz), hiç kuşku yok ki, kendisinden yüzyıllar önce Bizanslı bilgin Philon'un, su gücünden yararlanarak kurduğu otomatik sistemleri ve İskenderiyeli Heron'un hava basıncı ve su buharından yararlanarak meydana getirdiği otomatik aygıtları incelemişti. (Toygar Akman, ***Sibernetik Yaratıcılık,,*** *Bilgi Yayınevi, 1984.)*

Eb-Ül-İz'in, otomasyonun evrim tarihi içindeki önemli yeri; kendisinden öncekiler, yalnızca kendiliğinden çalışan otomatik sistemler kurduğu halde, onun, kendi kendine çalışırken, kendi kendine dengeler kuran ve ayarlamalar yapan sistemler kurabilmiş olması idi. Kısaca Eb-Ül-İz, otomatik sistemler arasında, kendiliğinden çalışarak dengeler kurabilen sistemler meydana getirebilmiş bir bilgin idi.

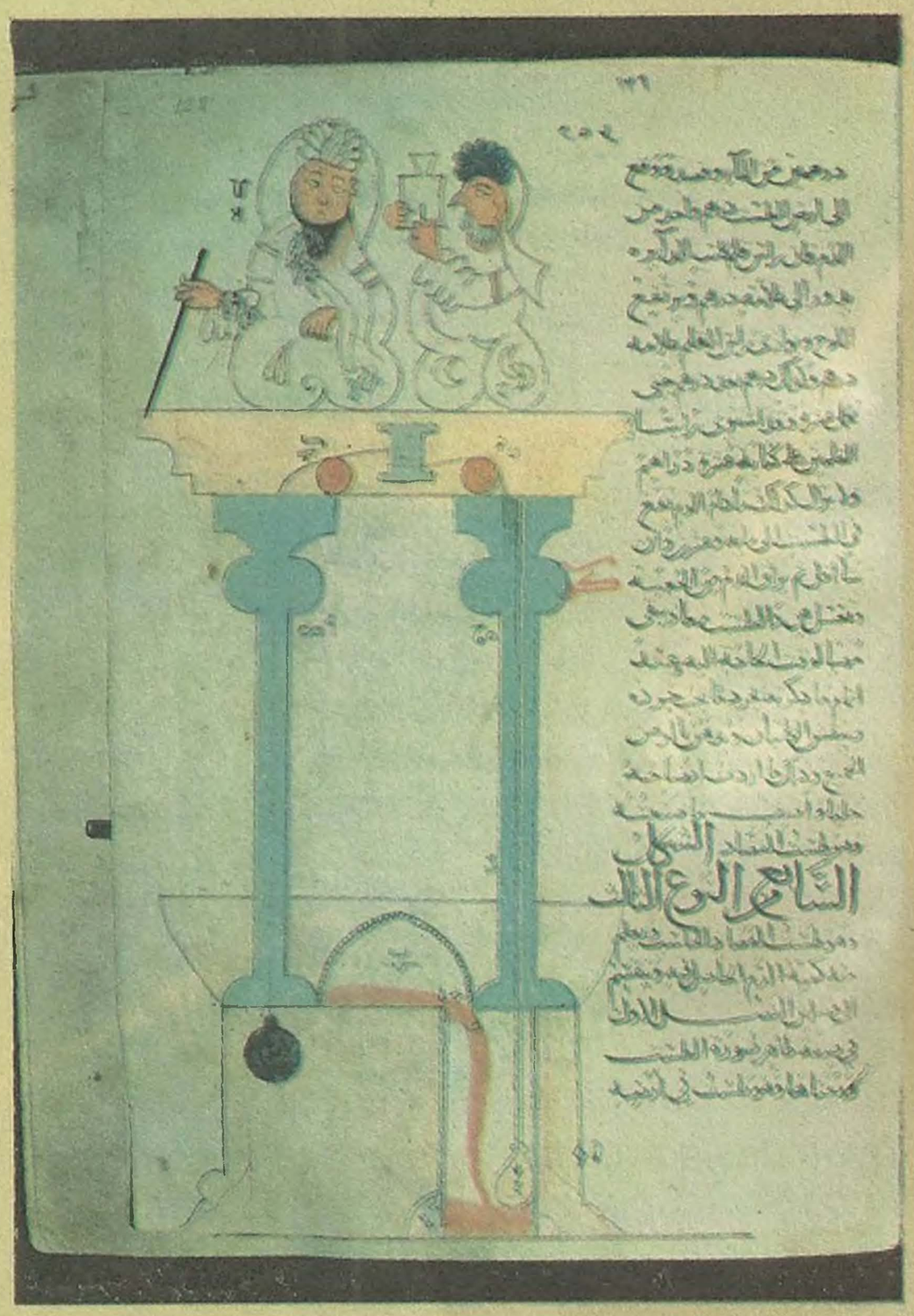
Çok iyi bilindiği gibi otomatik kelimesi, kendi kendine çalışan demek. Otomasyon kelimesi ise, sibernetik biliminin ortaya çıkmasından (yani 1943 yılından) sonra ortaya çıkmış bir kelime. Kısaca, sistemler arasında kendiliğinden denge kuran ve ayarlama yapan sistemler anlamına geliyor.

Eb-Ül-İz, kendisinden tam 800 yıl sonra ortaya atılacak olan sibernetik bilimini ve otomasyon teknolojisini bilmediği halde, yüzyıllar ötesinden, böylesine sistemler kurulabileceğini saptamış ve makineleri ile de kanıtlamış bir bilginimiz. Eğer bu kadar önemli olmasaydı, onun kitabının sayfaları bizim kütüphanelerimizden çalınır mı idi! Kitabının birçok istinsahi (el ile kopya edilmiş nüshaları) yabancı ülkelerin milli kütüphanelerinde saklanır mı idi!

''...Memleketimize dost gibi gelen, bizim misafirperverliğimizden en geniş derecede yararlanan bir müsteşrikin (Doğu dilleri bilgini) bu pırlanta eserin kıymetli yaprakları çaldığını ya da çaldırdığını, büyük bir teessürle öğrendik.'' (İ.Hakkı Konyalı, ***8 Asır Evvel Türk Sarayları Makineleşmişti,*** *Kara Amid Dergisi, Nisan 1969, Diyarbakır.)*

Biz, bu teessürle yetinir, Eb-Ül-İz'in kitabını (Türkçeye çevirmeksizin) saklamakla avunurken, Batı ülkeleri bilim adamları, onun bu değerli kitabını, kendi dillerine çevirmişler, istinsah edilmiş birçok nüshalarını da arayıp bularak kendi kütüphanelerine götürmüşler. Alman profesörlerinden Wideman tarafından, Eb-Ül-İz'in otomatik makinelerinden birkaç tanesi, aynen yapılarak işletilmiş. Bu makineler, Erlangen Üniversitesi'nde bulunuyor.

Oxford'da Bodleian Kütüphanesi'nde, Leiden Üniversitesi Kütüphanesi'nde, Dublin'de Chester Beatty Kütüphanesi'nde, Paris'te bibliotheque National'de, büyük bilginimizin kitabının el yazması kopyaları ya da çeviri olarak kopyaları bulunmakta. En son olarak da 1974 yılında Donald R.Hill tarafından İngilizce çevirisi ''The Book of Knowledge Ingenius Mechanical Devices'' adı ile yayınlandı. (Ibn al-Razzaz al Jasari, ***The Book of Knowledge of Ingenious Mechanical Devices,*** *çev. D.R. Hill, Hollanda, 1974.)*



''Hesaplayıcı Robot ile Yazıcı Robot''u Birlikte Çalıştıran Otomatik Sistem: ''Zamanın Akışı'', biri ''Elindeki Tahtaya Yazan'', diğeri de ''Yerdeki Çubukları Hesaplayarak Çizen'' iki Robot Adam'ın çalışması ile gösterilmektedir.

''Mum'un Yanarak Ağırlığını Kaybetmesiyle Oluşturulan'' ''Tek Otomatik Sistem''

Eb-Ül-İz'in otomatik makineleri ve robot adamlarının, yalnızca birkaç tanesinin işleyiş sistemlerini inceleyince, insan, bir an inanamıyor ve 800 yıl önce, böylesine bir otomasyon sisteminin nasıl kurulabilmiş olduğuna şaşırıyor!..

Büyük bilgin, bir yandan suyun itme gücü, bir yandan havanın basıncı, diğer bir yandan da mumun yanarak erimesi sonucu ağırlığını kaybetmesinden meydana gelen itme gücü vb. güçlerden yararlanarak karşılıklı ayarlama sistemleri kurabilmiş ve sistemler arasında da kendiliğinden işlemde bulunma durumunu sağlayabilmiş. Onun otomatik saatlerinin her biri, ilginç otomasyon sistemi örnekleri.

Suyun basıncı ile harekete geçen şamandıranın bağlı olduğu ip ile makaraları döndürmesini sağlamış, böylece de iki robot adama (biri yazıcı, diğeri de hesaplayıcı olarak) işlem yaptırabilen sistemleri kurabilmiş.

Hesaplayıcı robot ile yazıcı robotu birlikte çalıştıran sistemi şöyle kurmuş:

Altta bulunan çanak içindeki suyun, solda bulunan sütunun içine dolmaya başlaması ile birlikte, bu sütun içinde bulunan şamandıra yükselmeye başlamaktadır. Şamandıranın üstüne bağlı olan ip, soldaki sütun içinden yukarıya doğru çıkmaktadır. Orada iki robot adamın oturdukları kâidenin altında bulunan üç makara ile, robotları harekete geçirecek olan milli makara içinden dolanarak, sağdaki sütun içinden aşağıya doğru inmektedir. Orada da ipin ucu, bir ağırlığa bağlı bulunmaktadır. Soldaki sütun içindeki su yükseldikçe, sağdaki sütun içinde bulunan ağırlık ipi çekmektedir. Böylece de zamanın akışı, biri elindeki tahtaya yazan, diğeri de yerdeki çizgileri elindeki çubuğu ile deneleyerek gösteren iki ayrı robot adamın çalışması ile gösterilmektedir.

Eb-Ül-İz, mumun yanarak erimesi ve ağırlığını kaybetmesi ile yine kendiliğinden çalışan, çift otomatik sistemi de şöylece kurmuş:

Yanan mumun, eriyerek ağırlığını kaybetmesi ise, bu mumun üzerine oturduğu ağırlık kutusu harekete geçmektedir. Mum, yanıp ufaldıkça, ağırlığın bağlı olduğu dingil hem elinde kalem çubuk tutan otomat yazıcıyı harekete geçirmekte ve hem de yanda bulunan otomat kuşun pençesinde tuttuğu topların, belirli zaman aralıkları ile bırakılmasını sağlamaktadır. Eb-Ül-İz, bu makinesine ''Mumlu Saat'' adını vermiştir.

Yapmış olduğu makinelerde yer alan robot adamlar ile kuşları, ayrı bir estetik ile renklendirmiş ve süslemiş olması, büyük bilginin ne kadar titiz ve hassas bir yapıya sahip bulunduğunu belgelemektedir.

Yalnızca hava ve su basıncı ile kendiliğinden çalışan çok işlemli sistemler kurabilmiş olması ise, onun dehasını göstermektedir.

Batılı kaynakların, ''çağın doruğuna erişmiş Müslüman mühendis'' ya da ''çağın harikası Cizreli bilgin'' diye tanımladıkları, bizim bilginimiz Eb-Ül-İz, şaft sistemi ve dişli çarklar kullanmak suretiyle de ilginç otomasyon sistemi kurabilmiş. Otomatik yüzen kayık ve otomatik çalgıcılar bunun en güzel örneği. (Jasia Reichard, ***Robots [Fact, Fiction and Prediction],*** *Penguin, 1978.)*

Şekilde, erkek çalgıcıların bulunduğu bölmenin (deponun) içinden aşağıya doğru akan su, kepçelere gelerek onları döndürmektedir. Bu dönüş, kayığın arkasında ayakta duran otomat gemiciye, kürek çekme hareketi yaptırmaktadır. Burada ilginç olan durum, sistem, şaft ve dişli çarklarla sağlandığı halde, gemiciye yine de kürek çekme hareketinin yaptırılmakta olmasıdır. Bu şaft ve dişli çarklar aynı zaman-

da, kayıkta su deposunun üstünde bulunan kadın ve erkek çalgıcıların sazlarını çalmalarını da sağlamaktadır.

Yalnızca suyun kepçeleri döndürmesi ve şamandırayı kaldırması ile, bu kadar çok ve çeşitli hareketin meydana getirilebilmesi ve böylece üstün denge durumlu bir sistemin kurulabilmesinin 800 yıl önce nasıl düşünülebildiğini, insan kolayca kavrayamamaktadır.

Eb-Ül-İz'in, sıcak su ile soğuk suyu birlikte muhafaza eden testisi ile, bir otomat termostat icad etmiş olduğunu hayranlıkla öğreniyoruz. Hiç kuşku yok ki, onun en ilginç otomat makinelerinden bir tanesi de, Artukoğlu Hükümdarı'nın önerisi ve buyrukları üzerine, yapmış olduğu ''otomatik aptes alma'' makinesidir. Artuk Hükümdarı, makinenin yanına yaklaştığı anda, otomat adam elinde tuttuğu testiden hükümdarın eline su dökmeye başlıyordu. Hükümdar, aptes alma işini tamamladığında, aynı robot adam, bu kez diğer elinde tuttuğu havlu ve tarağı, hükümdara uzatıyordu. Artuk Hükümdarı, şimdi kurulanabilir saçını ve sakalını tarayabilirdi.

Buna benzer başka bir otomat makinesi ise, otomatik su döken meyve ve içki sunan makine idi. Artukoğlu Hükümdarı, hizmetçilerinin ve cariyelerinin ellerinin kirli olabileceği düşüncesi ile, misafirlerine ikramda bulunan bir makine yapmasını isteyince, Eb-Ül-İz bu makineyi de yapmıştı. Otomatik tavus kuşlu ikram makinesinde, tavus kuşunun ağzından dökülen su, hükümdarın ve konuklarının ellerini yıkamasını sağlıyordu. Bu yıkama işlemi süresince, aşağıdaki havuzda toplanan suyun kaldırdığı şamandıranın harekete geçmesi ile, yukarıda makinenin ortasındaki gizli bir kapıdan, otomatik hizmetçi, elinde içki ya da meyve çömleği ile çıkıyordu. Konuklar, hayret dolu bakışları ile bu içki ve meyveleri alınca, bu kez otomatik hizmetçi aynı gizli kapı içinden geri çekiliyordu. Onun yerine elinde havlu tutan otomat ortaya çıkıyordu. Konuklar, şimdi ellerini yıkayıp robot hizmetçinin uzattığı havlu ile kurulayabilirlerdi.

Ne kadar hazindir ki, bu ünlü bilginimizi, Batılı bilginler kendi dillerine çevirdikleri kitabını okuyarak tanıyorlar da biz, onun kitabını Arapçadan Türkçeye çevirerek kendi ulusumuz gençlerine tanıtamıyoruz.

(Toygar Akman, İlginç Otomasyon Sistemleri Kuran Üç Bilgin, Bilgisayar, Ocak 1985.)

Otomatik Yüzen Kayık ve Otomatik Çalgıcılar

Akış Şemaları

Akış Şeması **(flow-chart)**, bir programın grafik simgeler dizisi olarak gösterilmesine verilen addır. Programın akış şeması hazırlanırken, bu simgeler kullanılır. Bilgisayarın yapabileceği her işlev, grafik bir simge ile gösterilir. Akış şeması doğru yapıldığı takdirde, programın yazımı, akış şemasında yer alan simgelerin kullanılan programlama dilindeki karşılıklarına çevrilmesi gibi basit bir işlem haline gelir.

Kullanılan grafik simgeler, üç temel grupta toplanır:

- İşlemleri (hesaplama, karar verme vb.) gösteren simgeler
- Veri giriş ve çıkış işlevlerini gösteren simgeler
- Bellekle ilgili simgeler

Bu sayfada gösterilen grafik simgeler genellikle daha sık kullanılan simgelerdir, ancak küçük ve kişisel bilgisayarların programlanmasında kullanılan simge sayısı çok daha azdır.

Bu bölümde, sadece çok sık kullanılan ve örnek uygulamalarımızda görülecek grafik simgeler ayrıntılı olarak anlatılacaktır. Çok az kullanılan diğer simgeler, kısa bir özet verilerek geçilecektir.

Sık Kullanılan Simgeler

İşlev simgeleri en sık, programın ana gövdesi özellikle işlem yapmak için hazırlanacaksa kullanılır.

Aşağıdaki şekilde görülen **a** simgesi, programın başlangıç ve sonunu belirtmek için kullanılır. Program, ne hakkında ve ne kadar karışık olursa olsun, sayfa 169'da üstteki şekilde gösterilen genel yapıyı taşır.

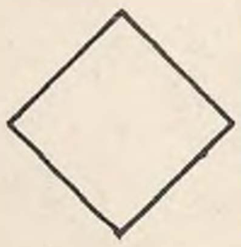
Ok işaretleri, işlemin yapılma sırasını belirtir.

AKIŞ ŞEMASI SİMGELERİ

İşler simgeleri

a Programın başlangıcı ve bitişi

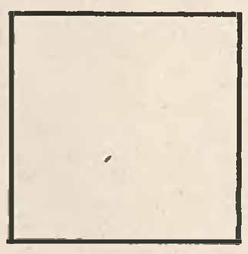
b Genel işlemler (hesaplama ve değer bildirme)

c  Sınama (karar, ilk biçimi)

d Sınama (ikinci biçimi)

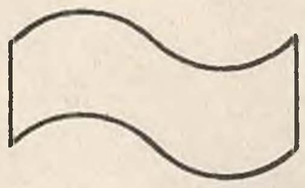
e Sayaç (İçindeki N harfi kaçıncı dönüşte olunduğunu bildiren sayıdır)

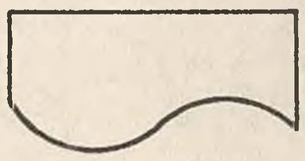
f Elle yapılacak işlemler

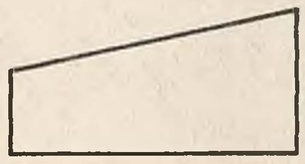
g  İkincil işlemler

Giriş/Çıkış simgeleri

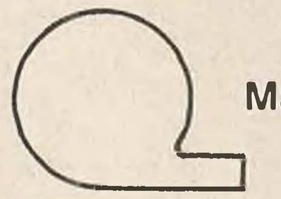
h Genel giriş/çıkış işlemleri

i  Delikli kâğıt şerit giriş/çıkışları

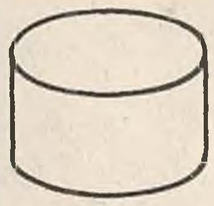
l  Özel yazı düzeni gerektiren belgeler için giriş/çıkış

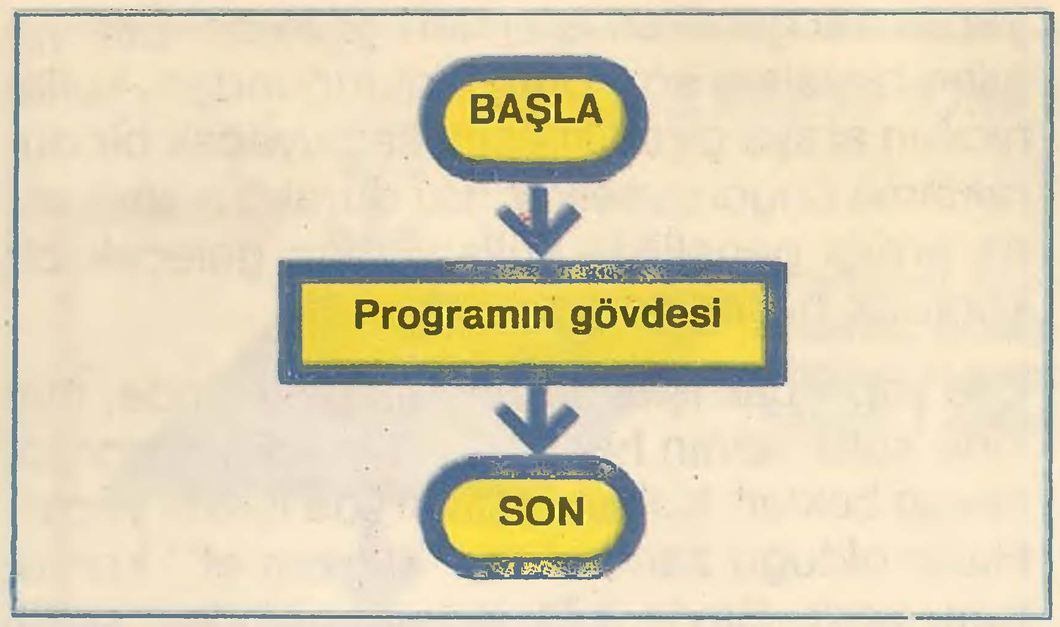
m  Verinin elle girileceğinin bildirilmesi

Bellek simgeleri

n  Manyetik alan

p Bellek

q  Disk birimi



İncelemekte olduğumuz stok yönetimi programında toplam maliyet, beher malın fiyatı, malın miktarı ile çarpılarak bulunur. Bu tür işlemler genellikle **b** harfi ile gösterilir:

Toplam maliyet = Miktar × Birim Fiyat

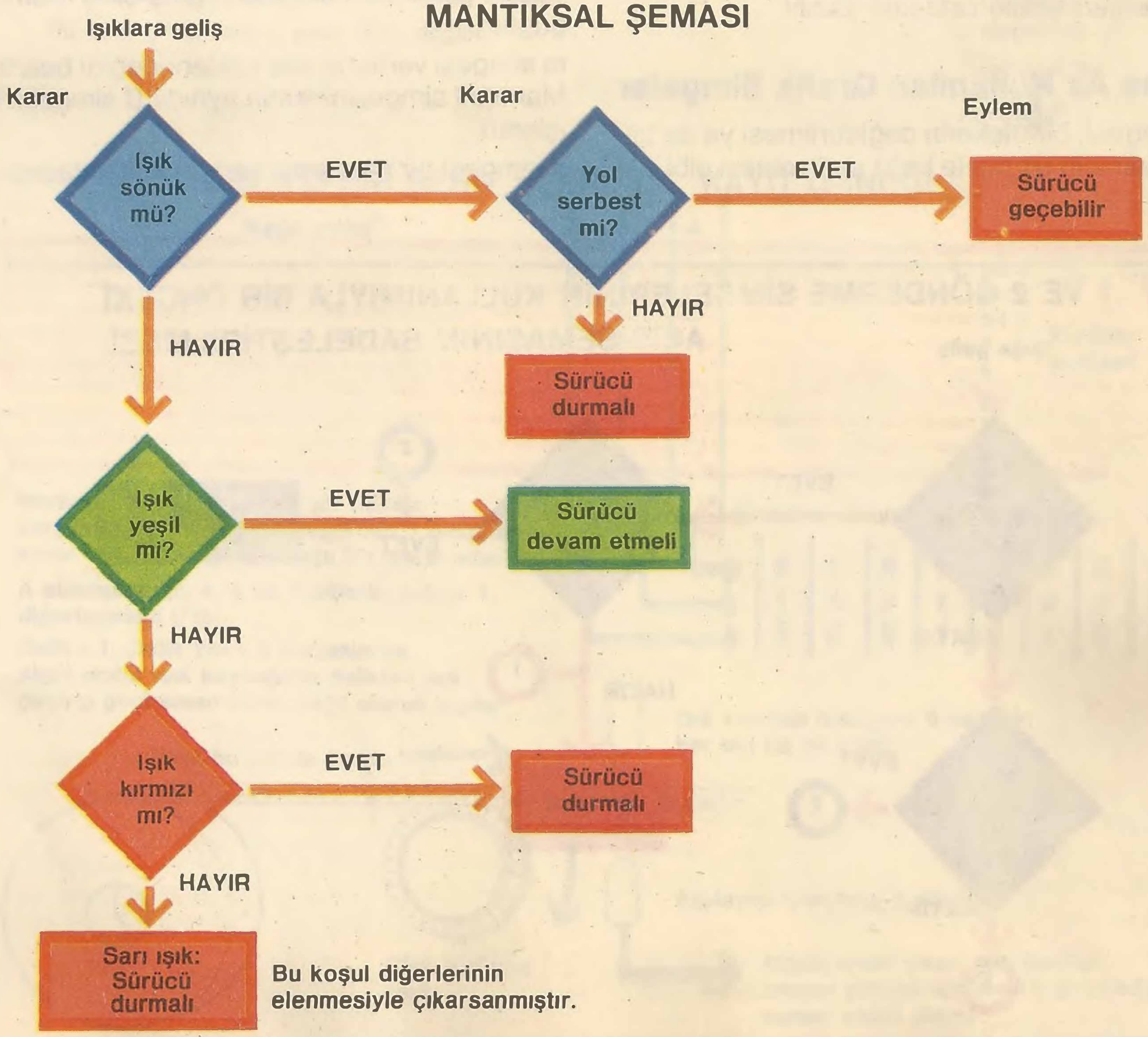
c ve **d** simgeleri ise bir karar verme (ya da **sınama**) işlemini göstermek için kullanılır. Hangi simgenin kullanılacağını, işlem hakkında eklenecek yorumun uzunluğu belirler. Sınama iki aşamadan oluşan bir işlemdir:

- Birinci aşama: İki (ya da daha çok) değer birbirleriyle karşılaştırılır.
- İkinci aşama: Karşılaştırmanın sonucuna bağlı olarak belirli bir yolun izlenmesine, yani belirli bir işin yapılmasına karar verilir.

Şimdi bu iki aşamayı trafik ışıkları örneğimizde görelim. Sürücünün değişik davranış olasılıklarını **b** ve **c** simgeleriyle şemalaştırmak mümkündür. (Bakınız: aşağıdaki şekil.)

Sayfa 168'deki şekilde **e** (gönderme) simgesi, gereği yerine getirilen bir dizi komuttan

TRAFİK IŞIKLARINA GELEN BİR SÜRÜCÜNÜN DAVRANIŞLARININ MANTIKSAL ŞEMASI



Bu koşul diğerlerinin elenmesiyle çıkarsanmıştır.

sonra programın hangi noktasına geri dönüleceğini belirtmek için kullanılır. Böylece simgeleri birbirine bağlayan çizgiyi (yani akış çizgisini) izlemekten kurtulmuş oluruz. Bu hem akış şemasının okunmasını kolaylaştırır, hem de birçok sayfa uzunluğunda şemalar çizebilme olanağını verir. Bu sayfadaki şema bu gönderme simgesinin nasıl kullanıldığını göstermektedir.

Bu şemada ''ışık kırmızı mı?'' sorusunu da atlayabiliriz, çünkü cevap ''Evet'' (= Kırmızı) de olsa ''Hayır'' (= Sarı) da olsa yapılması gereken, aynı şeydir: Beklemek.

Bir program ancak, işlenecek bütün bilgiler makineye verilmişse çalışabilir. Bu bilgilerin (örneğin bir klavye yardımıyla) verilmesi bir giriş (Input) işlemi ve işlendikten sonra (örneğin bir bilgiyazar tarafından) yazılmaları da bir çıkış (Output) işlemidir.

Genel bir giriş-çıkış işlemini (G/Ç) göstermek için en çok kullanılan simge **h** harfidir. (Bakınız: s.168.) Sözkonusu işlemin bir giriş işlemi mi, yoksa bir çıkış işlemi mi olduğu **h** ile gösterilen şeklin ortasına yazılır.

Daha Az Kullanılan Grafik Simgeler

f simgesi, bir disketin değiştirilmesi ya da bilgiyazar için bir deste kağıt sağlanması gibi elle yapılması gereken işlemleri gösterir. Elle yapılan bir işlem sözkonusu olduğundan, kullanıcının araya girebilmesini sağlayacak bir duraklama öngörülmelidir. Bu duraklamanın sona erdiği genellikle kullanıcıdan gelecek bir komutla bildirilir.

Elle yapılacak işlemin zamanı geldiğinde, makine kullanıcının hazır olup olmadığını sorup, cevap bekler: Kullanıcı bu arada işlemi yapar. Hazır olduğu zaman da, ''devam et'' komutunu verir. Sayfa 171'in yukarısındaki şekil, bu programın basitleştirilmiş akış şemasını vermektedir.

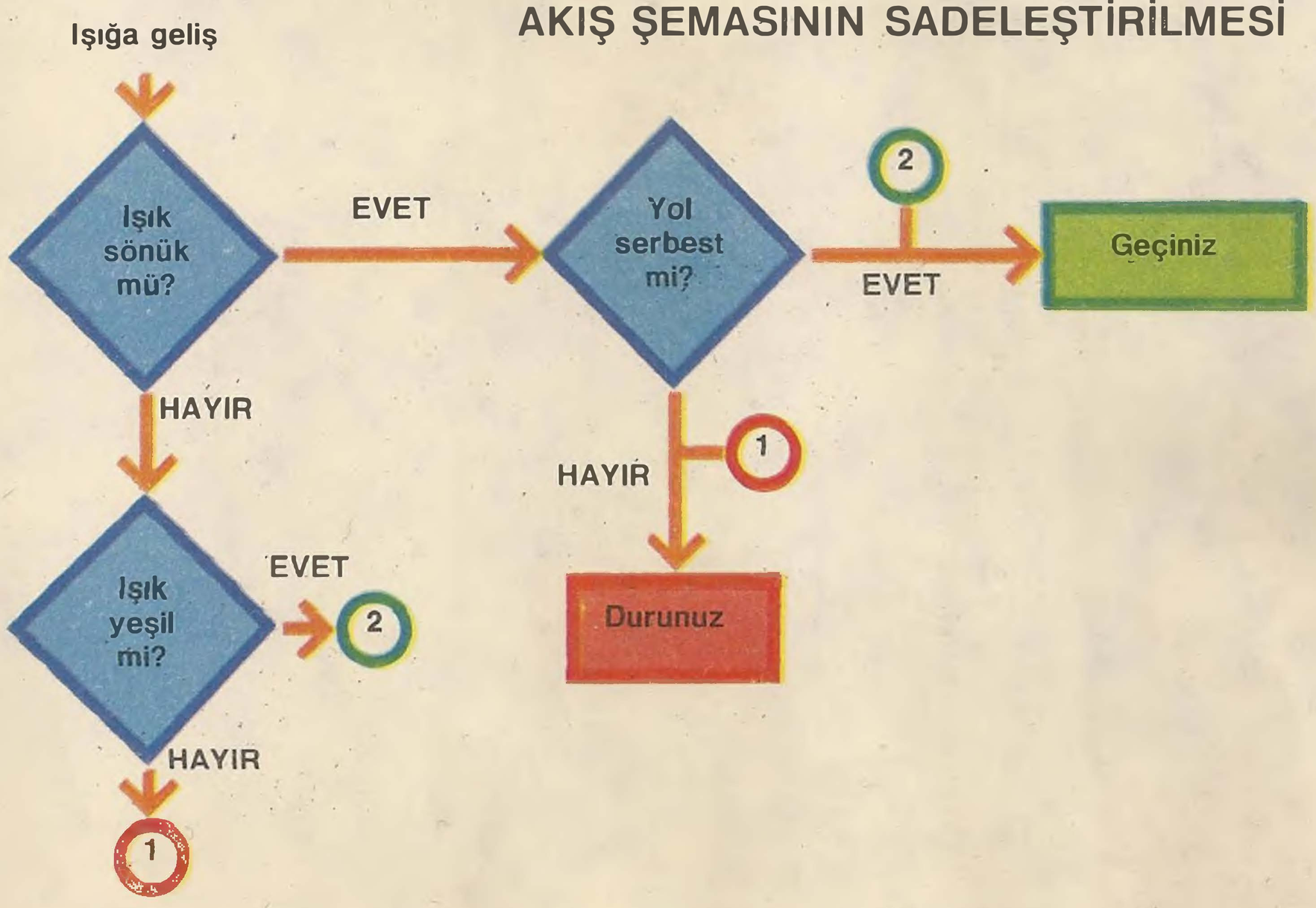
g simgesi programla yakın ilişkisi olmayan her türlü dış işlemi, **i** simgesi ise delikli bir şeridin okunacağını ya da yayınlanacağını gösterir. Bu şeritler verilerin delikler aracılığı ile kaydedilmesini sağlar (deliğin varlığı 1 düzeyine karşılık gelir, bakınız s.171'deki şekil). Bugün artık bu şeritler, yerlerini manyetik gereçlere bırakmıştır.

l simgesi ödeme makbuzları gibi özel bir yazı düzeni gerektiren belgelerin giriş-çıkış işlemini gösterir.

m simgesi verilerin elle yükleneceğini belirtir. Mantığı **f** simgesininkinin aynıdır (**f** simgesi: el işlemi).

n simgesi bir manyetik şeridin kullanılacağını

1 VE 2 GÖNDERME SİMGELERİNİN KULLANIMIYLA BİR ÖNCEKİ AKIŞ ŞEMASININ SADELEŞTİRİLMESİ



EL İŞLEMLERİ İÇİN BEKLEME YÖNTEMİ

Bilgisayar elle yapılması istenen işi bildiren, örneğin şöyle bir mesaj verir: Bilgiyazar kağıdını değiştirin ve bu iş bitince 1 değerini girin

Bilgisayar cevabı bekler

İşlemci kağıdı değiştirince 1 değerini girer. Diğer bütün değerler yeni bir beklemeye yol açar.

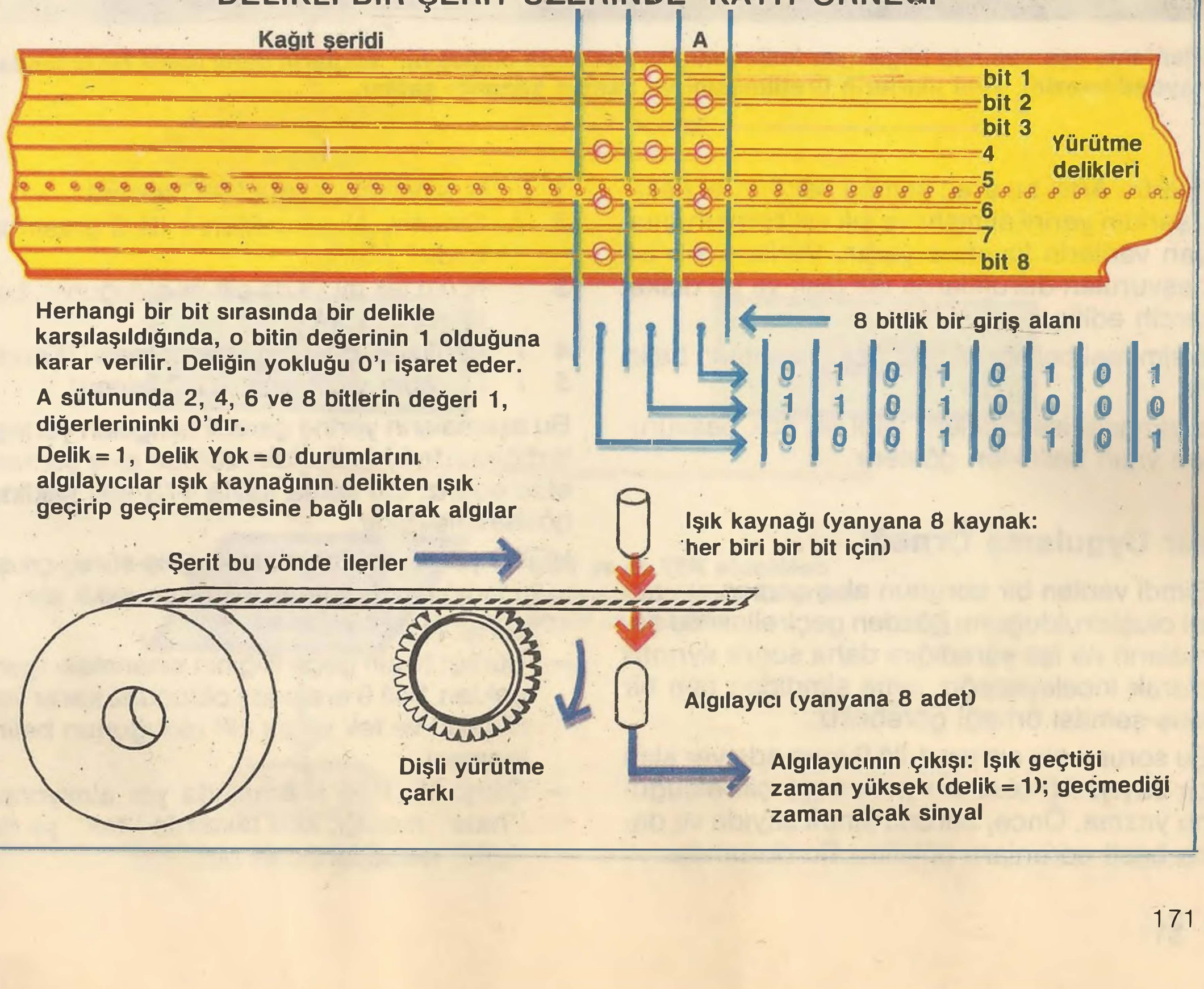
Bu karar şuna karşılık gelir: Giriş değeri 1 mi?

Kararlar

G/Ç İşlevleri

Ayrıntılı akış şeması tek bir el işlemi simgesi ile değiştirilebilir.

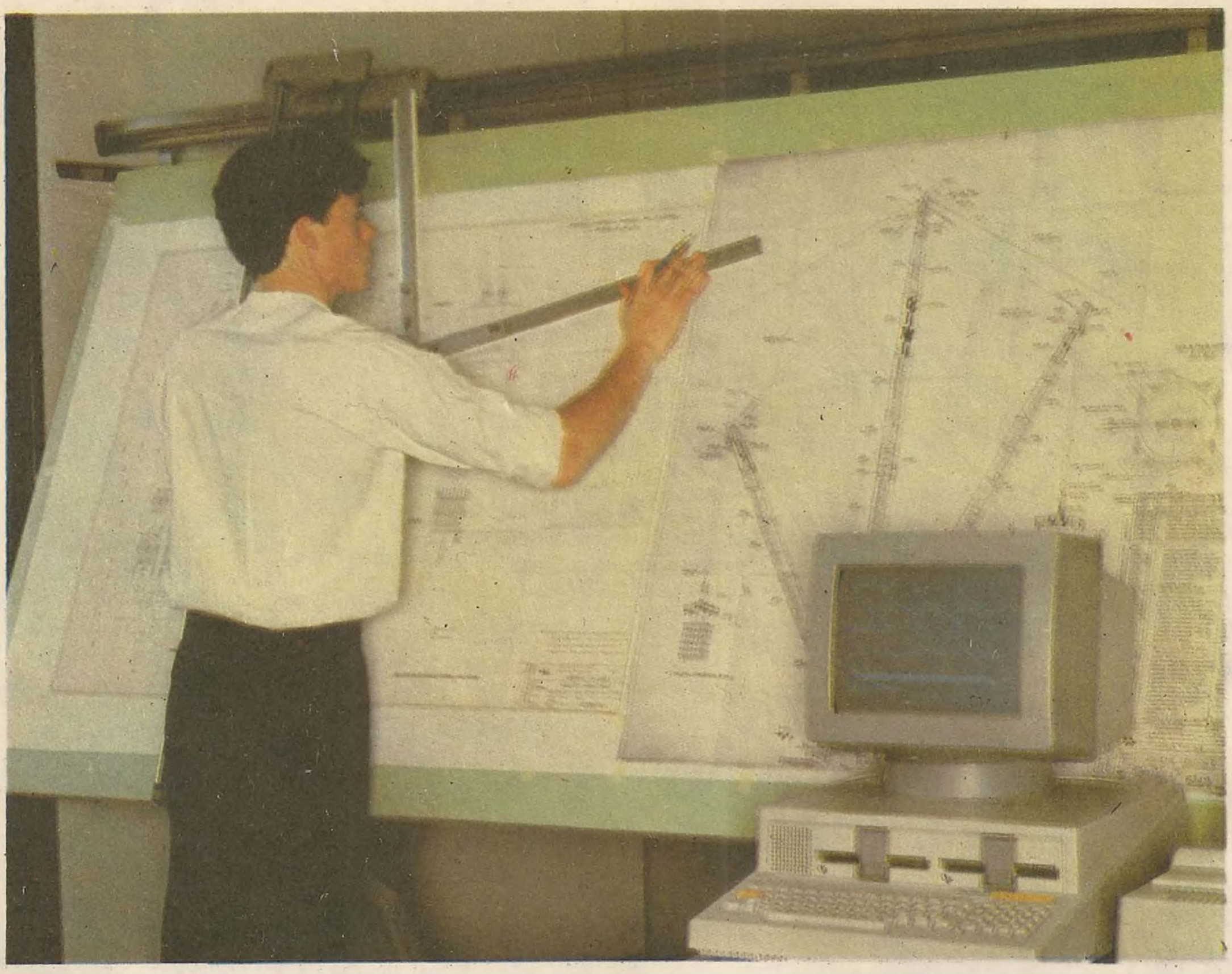
DELİKLİ BİR ŞERİT ÜZERİNDE KAYIT ÖRNEĞİ



Herhangi bir bit sırasında bir delikle karşılaşıldığında, o bitin değerinin 1 olduğuna karar verilir. Deliğin yokluğu 0'ı işaret eder.

A sütununda 2, 4, 6 ve 8 bitlerin değeri 1, diğerlerininki 0'dır.

Delik = 1, Delik Yok = 0 durumlarını algılayıcılar ışık kaynağının delikten ışık geçirip geçirememesine bağlı olarak algılar



Planlama aşamasında bilgisayar kullanımı: Bu aşamada bilgisayar, verilerin daha dakik bir biçimde kaydedilmesini, yeni fikirlerin üretilmesini ve zaman kazancı sağlar.

belirtir. Artık bu araç hemen her yerde, delikli şeridin yerini almıştır ve sık sık başvurulmayan verilerin kaydına yarar. Verilere sık sık başvurulan durumlarda ise disk ya da disket tercih edilir.

p simgesi belleği ve ona ilişkin işlemleri belirtir.

q simgesi ise diskleri, yani en çok başvurulan yığın bellekleri gösterir.

Bir Uygulama Örneği

Şimdi verilen bir sorunun akış şemasının nasıl oluşturulduğunu gözden geçirelim. Bu şemaların ne işe yaradığını daha sonra ayrıntılı olarak inceleyeceğiz, ama şimdiden tam bir akış şeması örneği görebiliriz.

Şu sorunu ele alalım: 1 ilâ 9 arasında yer alan bir sayıyı (N) okuma ve tek veya çift olduğunu yazma. Önce, sorunu sınırlı sayıda ve daha basit sorunlara bölelim. Bu durumda:

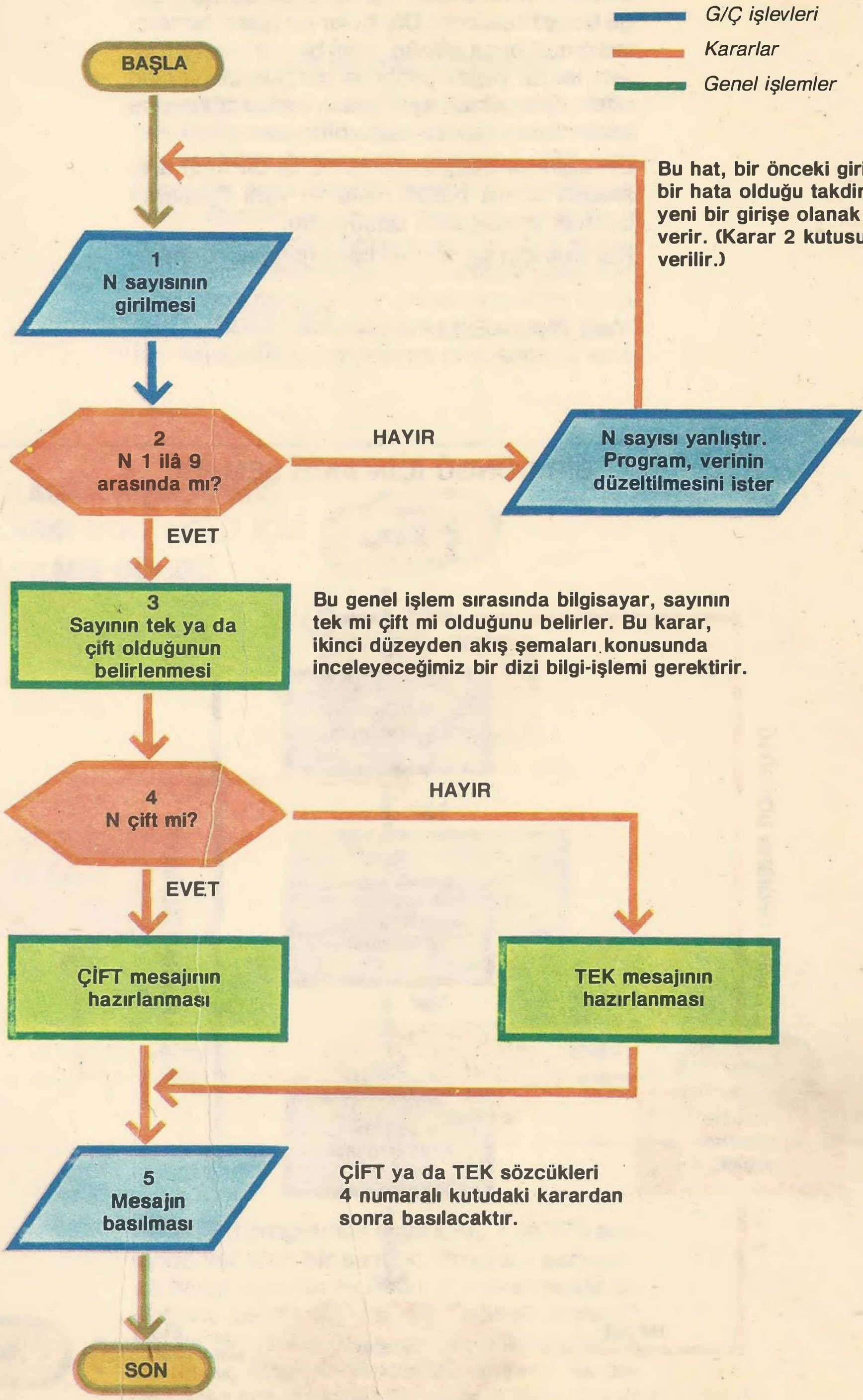
1 / N sayısını okuma (G/Ç işlemi)
2 / Sınama: N gerçekten 1 ilâ 9 arasında mıdır? (Karar)
3 / N'nin tek mi ya da çift mi olduğunun belirlenmesi. (Asıl iş)
4 / Verilecek cevabın belirlenmesi. (Karar)
5 / Cevabın yazılması. (G/Ç işlemi)

Bu aşamaların yerine gerekli simgeleri yerleştirdiğimizde birinci düzeyden bir akış şeması elde ederiz. Bu şema sayfa 173'teki şekilde gösterilmektedir.

HGSÇ yöntemini (hiyerarşik, giriş-süreç-çıkış) uyguladığımız takdirde şema şu şekli alır:

— Giriş: N
— Süreç: N'nin geçerliliğinin sınanması (gerçekten 1 ilâ 9 arasında olduğuna karar verilmesi) ve tek ya da çift olduğunun belirlenmesi
— Çıkış: N, 1 ilâ 9 arasında yer almıyorsa ''hata'' mesajı; aksi takdirde ''tek'' ya da ''çift'' mesajlarından biri.

BİRİNCİ DÜZEYDEN BİR AKIŞ ŞEMASININ UYGULAMA ÖRNEĞİ



Bu hat, bir önceki girişte bir hata olduğu takdirde yeni bir girişe olanak verir. (Karar 2 kutusunda verilir.)

Bu genel işlem sırasında bilgisayar, sayının tek mi çift mi olduğunu belirler. Bu karar, ikinci düzeyden akış şemaları konusunda inceleyeceğimiz bir dizi bilgi-işlemi gerektirir.

ÇİFT ya da TEK sözcükleri 4 numaralı kutudaki karardan sonra basılacaktır.

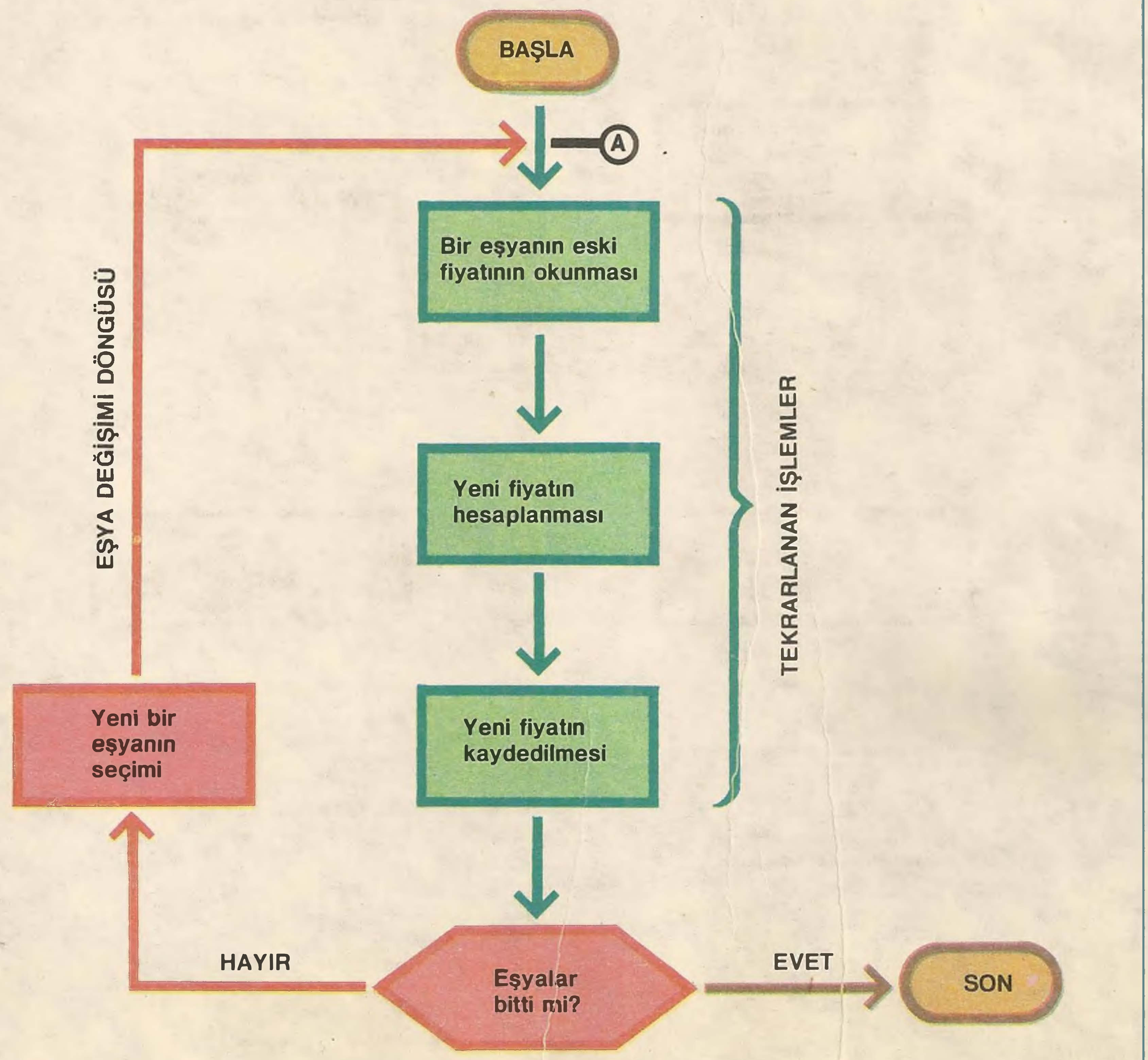
Döngüler

Aynı işlemin değişik verilerle üst üste tekrarlandığı durumlarda bir minik-işlemci sistemi özellikle elverişlidir. Bu durumlarda bir döngü **(loop)** kullanılır. Döngüler bir işlemi tamamladığında başa dönüp, yeni bir veriyle aynı işlemi tekrar yapan program parçalarına verilen addır. Döngünün aynı işlemi veriler tükenene kadar tekrarlaması sağlanmalıdır.

Örneğin bir mağazada % 12'lik bir fiyat artışından sonra bütün malların yeni fiyatlarını bulmak istediğimizi düşünelim.
Her mal için şu işlemin tekrarlanması gerekir:

$$\text{Yeni Fiyat} = \text{Eski Fiyat} + \frac{\text{Eski Fiyat}}{100} \times 12$$

BİR DÖNGÜ İÇİN AKIŞ ŞEMASI

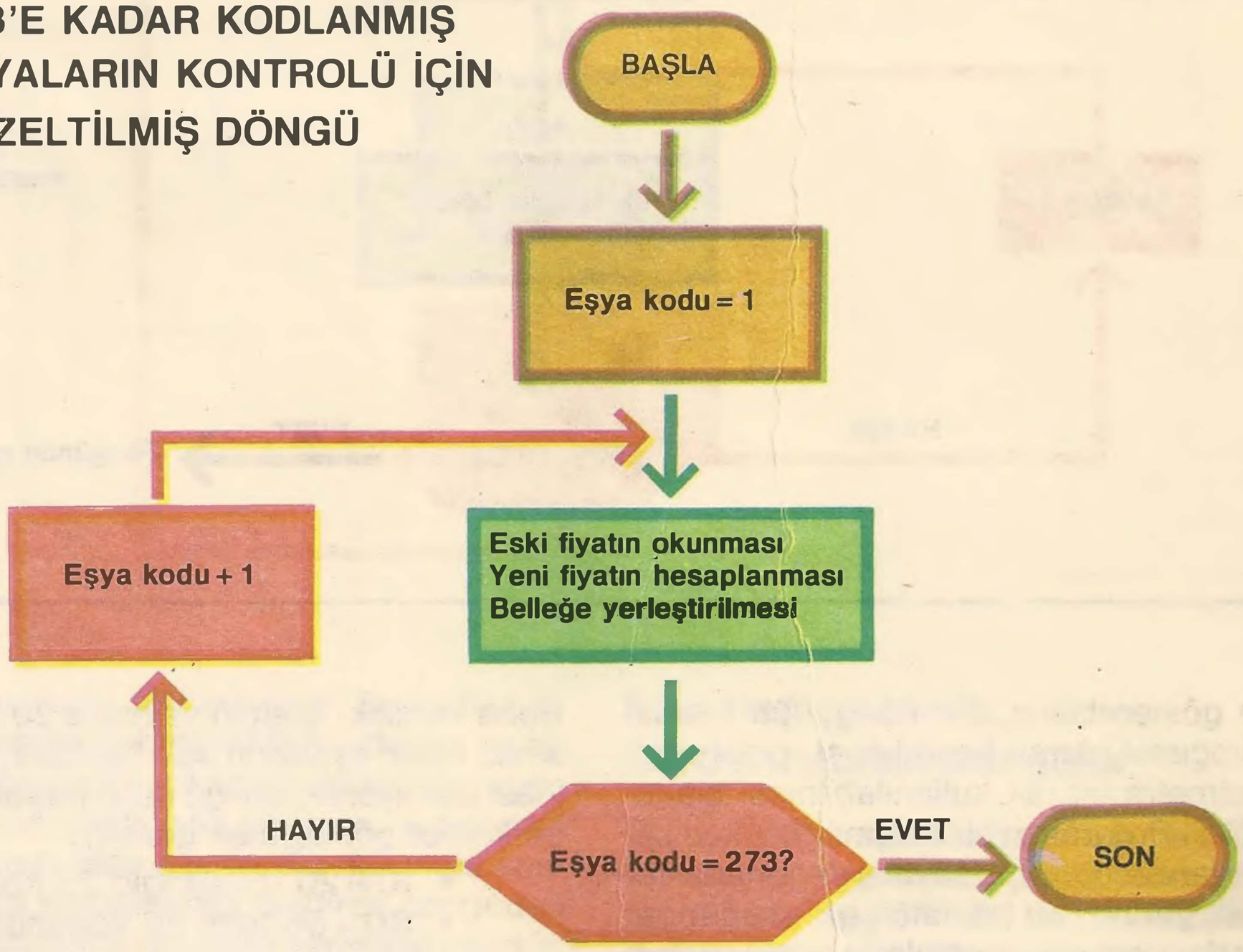


Bu işlemin birinci dereceden akış şemasını gösteren Sayfa 174'teki şemada döngüyü oluşturan bölüm kırmızıyla gösterilmiştir. Her yeni mal için, işlemin başına, yani A noktasına dönülmektedir.

Sınırları Belirlenmiş Döngüler

Bir mağazada 1'den 273'e kadar sayı verilerek kodlanmış 273 değişik eşya varsa, 1 sayılı eşyadan başlayarak döngüyü 273 kere (yani her eşya için bir kere) tekrarlamak gerekir. İşlemin bütün eşyalar için tekrarlanıp tekrarlanmadığını öğrenmek için, 273 sayısına varılıp varılmadığının sorulması yeterlidir. Eğer varılmamışsa bir önceki eşyanın kodu na bir ekleyerek yeni bir eşya seçmek gerekir. Aşağıda bu programın akış şeması veril-

273'E KADAR KODLANMIŞ EŞYALARIN KONTROLÜ İÇİN DÜZELTİLMİŞ DÖNGÜ

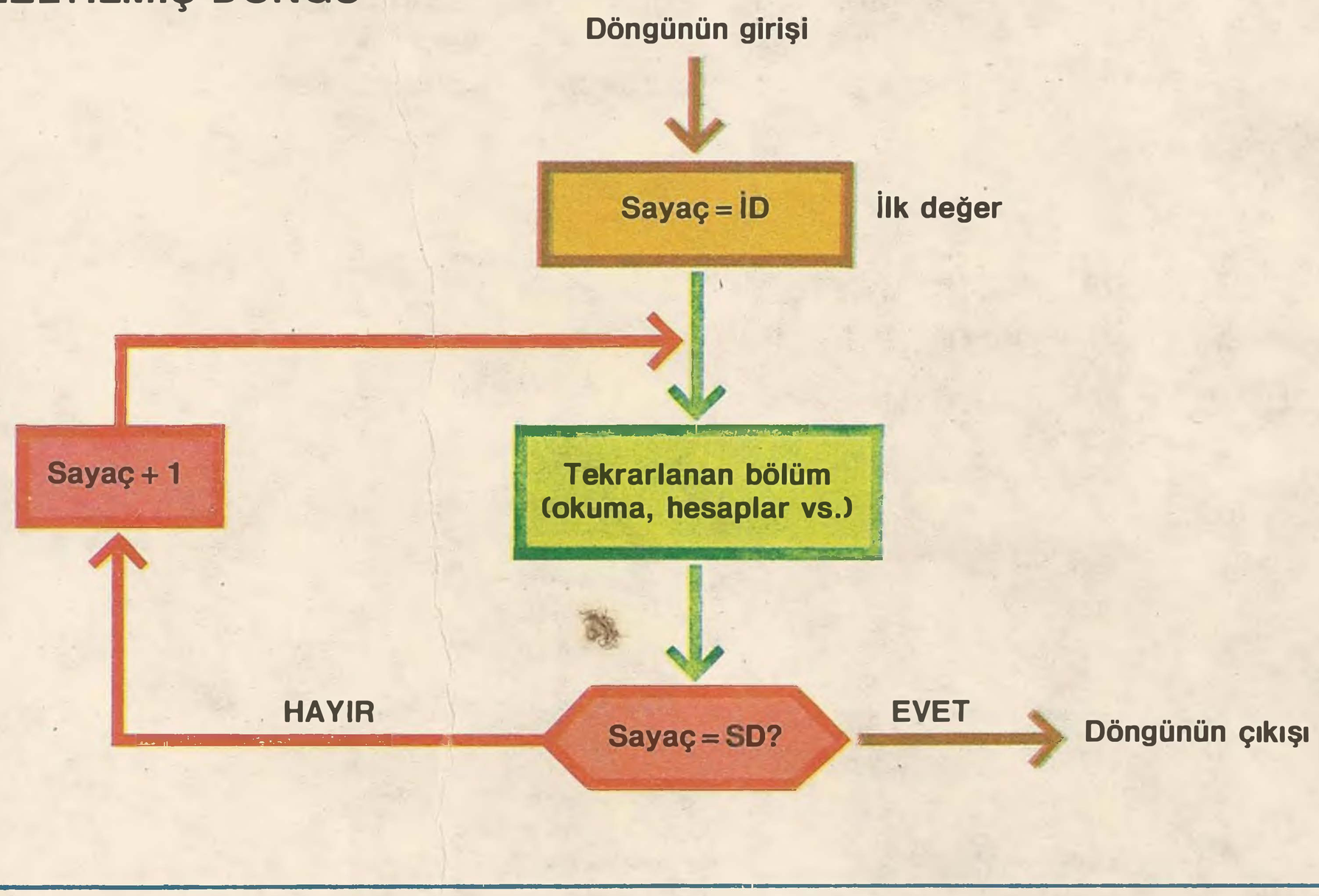


miştir. Bu programda eşya kodu 1 ile 273 arasında yer alan bir sayıdır. Program esnasında hangi eşyanın fiyatının hesaplanmakta olduğunu belirtir. Bu sayıya **"döngü sayacı"** adı verilir. Döngülerin büyük çoğunluğunda bir sayaç kullanılır ve döngü, sayacın ilk değerinden son değerine (burada, 1'den 273'e) kadar tekrarlanır.

Sınırları Parametrelere Bağlı Döngüler

Sayfa 175'teki şema üzerinde, döngünün sınırları, yani sayacın ilk ve son değerleri (1 ve 273) sayısal bir biçimde ifade edilmiştir. Ancak aynı değerlere birer simge vererek, bunları dolaylı yoldan göstermek de mümkündür. Bu iki sınıra **İD (İlk Değer)** ve **SD (Son Değer)** dersek, tipik bir döngünün akış şemasını şu

İD VE SD SINIRLARI ARASINDA ÇALIŞTIRILMAK ÜZERE DÜZELTİLMİŞ DÖNGÜ



şekilde gösterebiliriz. Bir döngünün sınırlarının simgesel olarak belirtilmesi, onların birer parametre olarak kullanılabilmesi olanağını sağlar. Programın bir bölümünün, kaç kere tekrarlanacağını açık sayısal değerlerle ifade etmek yerine, bu bölümün girişini simgesel adlarla, yani parametrelerle adlandırıyorsak sözkonusu bölüme ''parametreli'' denir. Bu parametrelerin sayısal değerlerinin bölümde yer alan komutlardan önce bildirilmesi gerekir. Bu yöntem aynı komutların değişik sayısal değerlere uygulanmasına olanak verir. Örneğimizde, işlemin bütün malların eski fiyatına uygulanması isteniyorsa, parametrelere şu değerler verilir:

- **İlk Değer = İD** = 1
- **Son Değer = SD** = 273

Buna karşılık, işlemin yalnızca 20 ilâ 56 arasında kalan eşyaların eski fiyatlarına uygulanması isteniyorsa, döngü daha başlamadan önce komut göndermek gerekir:

- **İD** = 20 (başlangıç 20 kodunda)
- **SD** = 56 (son 56 kodunda)

Verilerin Bittiğini Anlayan Döngü

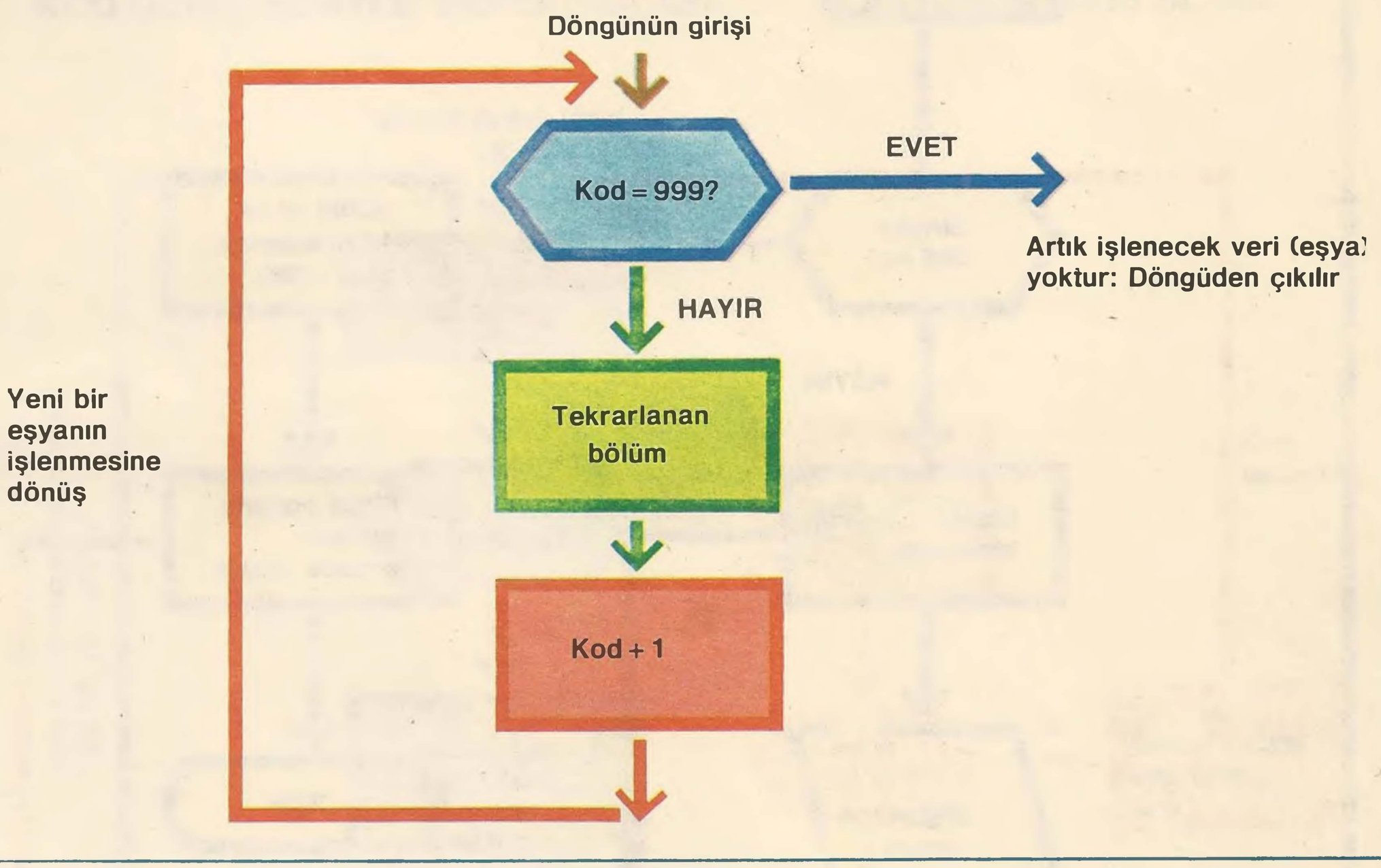
Şimdiye kadar gördüğümüz döngüler, yani sınırları belirli ya da sınırları parametrelere bağlı olan döngüler, ancak bütün veriler işlendikten sonra sona eriyorlardı. Bu döngülerin elverişsiz bir yanı toplam veri sayısının önceden belirlenmesini gerekli kılmalarıdır (örnekte 273).

Oysa, bütün uygulamalarda toplam veri sayısı önceden kesin olarak bilinmeyebilir. Bu du-

rumlar için belli bir verinin son veri olduğunu saptayabilecek bir döngü tipi geliştirmek gerekir. Bunun için işlenecek verilerin sonuna üzerinde anlaşılmış, tanımlanması kolay bir değer verilir; döngü bu değerle karşılaşınca durur. Bu simgesel değer için ''verilerin sonu'' ya da ''kütük sonu'' **(EOF, end of file)** veya ''son etiketi'' terimleri kullanılır.

Yine mağaza örneğimize dönecek olursak; yüklenecek verilerin bittiği, olmayan bir eşya-

VERİLERİN BİTTİĞİNİ ANLAYAN DÖNGÜ



nın kodu verilerek belli edilebilir. Bu kodla karşılaşmak verilerin bittiğini belli edecek ve bu sayede döngüden çıkılabilecektir. Bu koda 999 değerini verdiğimiz takdirde, yukarıdaki akış şemasını elde ederiz. Bu durumda yanlış bir kodla karşılaşıldığı takdirde döngüden çıkılması kararının önceden verilmesi gerekir. Aksi takdirde işlemlerin birtakım hatalı verilere de uygulanması sonucu doğabilir. Kütük sonu kodu verilerin yüklenmesi aşamasında belirtilir. Bu kodun değerinin başka bir verininkine eşit olmaması gerekir. Mağaza örneğinde, bu değer belirlenmeden önce, listeye yeni eşyaların eklenme olasılığı da gözönünde bulundurulmalıdır.

Sayfa 178'deki akış şeması bu olasılık gözönünde bulundurularak hazırlanmıştır.

Kod Değişmesiyle Durdurulabilen Döngüler

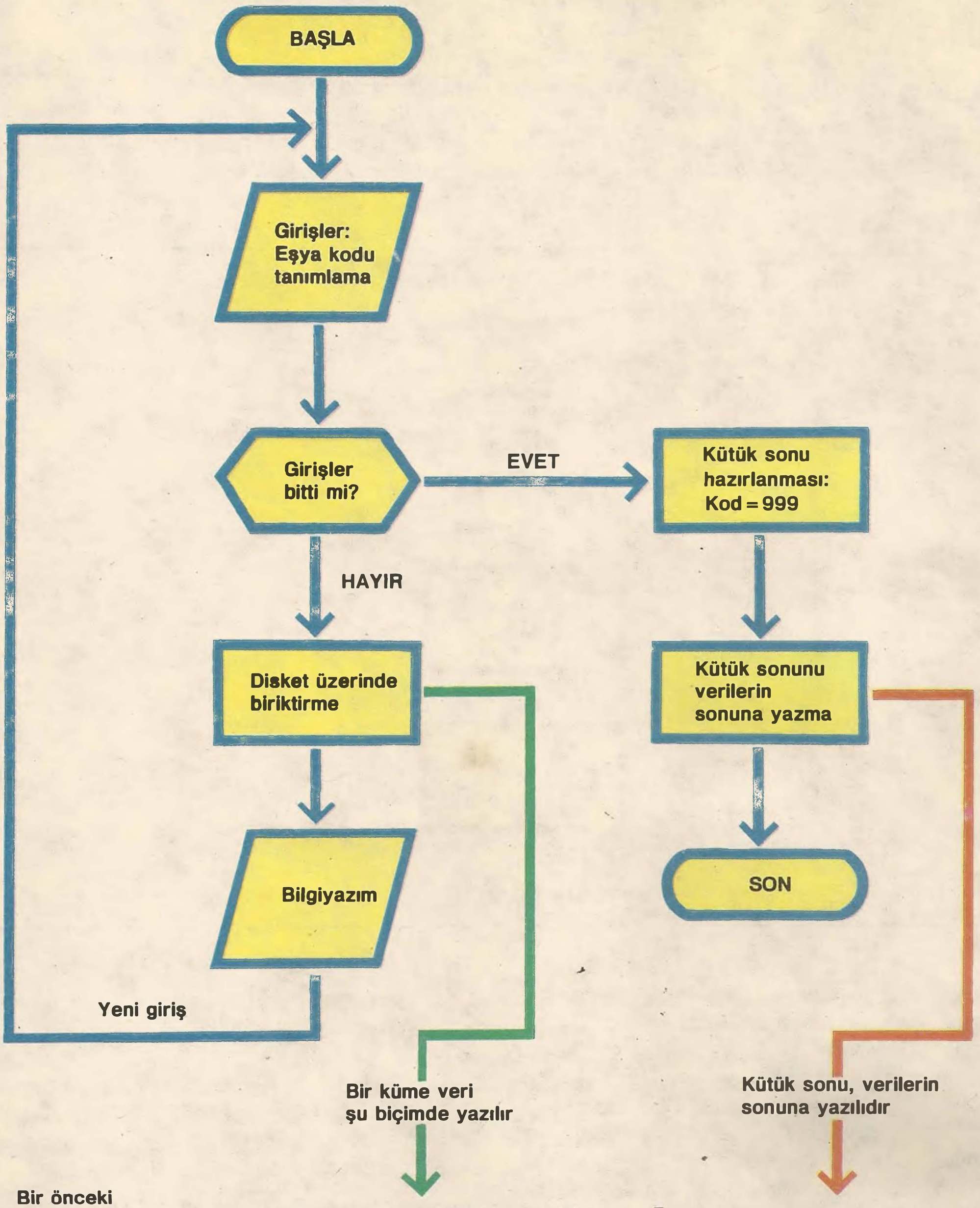
Bu tip döngü, işlemlerini ancak belirli bir parametre ilk değerini koruduğu sürece yapar. Bu değerini değiştirirse, bir kod bozulması olduğu söylenir ve döngü durdurulur.

Bu süreçten COBOL ve RPG* dillerinde sık sık yararlanılır. BASIC'de de böyle bir döngü oluşturma olanağı varsa da BASIC dilinde bu tür döngülerin özel bir işareti yoktur.

---

* Cobol'a çok benzeyen RPG **(Report Program Generator** - ya da Otomatik Program Yaratıcısı) şimdi daha az karmaşıktır ve yalnız belli bilgisayarlarda, örneğin IBM'-de kullanılmaktadır.

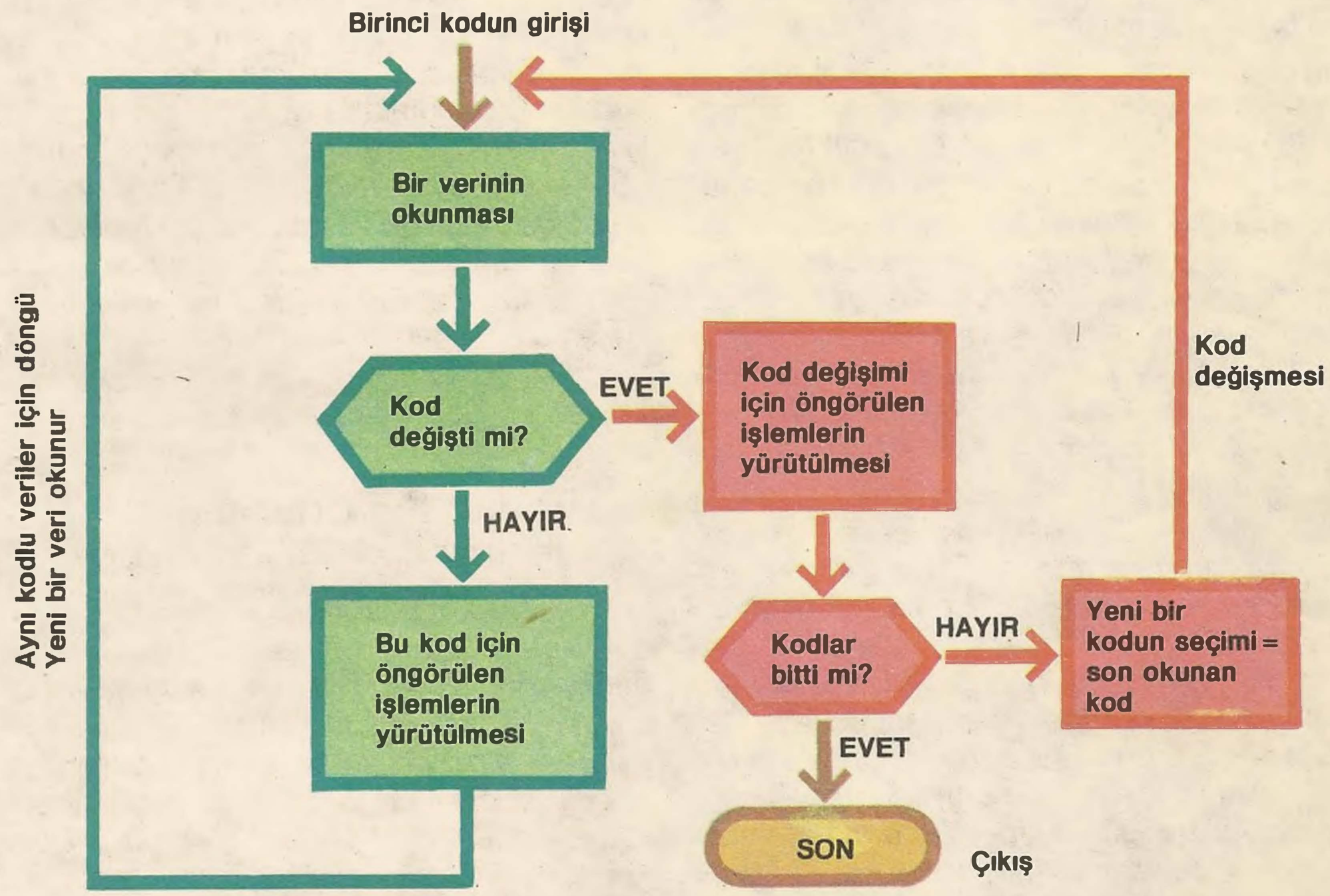
MAĞAZADA STOK DÜZEYİNİ BELİRLEMEK İÇİN AKIŞ ŞEMASI



Bu tür döngüler hep aynı türden verilerin işlenmesinin gerektiği, yani veri kodunun aynı kaldığı durumlarda çok yararlı olur. Veri girdisinin herhangi bir bölümü verinin kodu olarak yorumlanabilir. Örneğin mağaza yönetimi programıyla malların stok düzeyini belirlemeye çalıştığımızı düşünelim. Bu durumda mal cins kodu veri kodu olarak kullanılabilir.

İki çalışma hattından birincisi (A öbeği) aynı kodlu verilerin işlemlerini tamamlamak, ikincisi ise (B öbeği) döngünün durdurulması halinde yürütülmesi gereken işlemleri göstermek için kullanılmıştır. Döngünün durdurulması halinde, önce o noktaya kadar bütün kodların işlendiğinden emin olmak gerekir. Eğer gelen kod ''kütük sonu'' ise yani bütün veriler işlenmişse program durur. Aksi halde döngünün başına dönülerek, yeni malın stok düzeyi hesaplanmaya başlanır. Bu yöntemin başarılı bir biçimde uygulanabileceği bir örnek, bir grup satış temsilcisinin satışlarının çözümlenmesidir. Sorun belirli bir dönem içinde her

KOD DEĞİŞMESİYLE DURDURULABİLEN DÖNGÜNÜN AKIŞ ŞEMASI



İleti, İletişim Ortamının Kendisidir

Bazen, bizimkisi gibi kontrol edebilmek için her şeyi ayırmaya ve bölmeye alışmış bir toplumda, bir iletişim aracının kendisinin de bir ileti olduğu gibi, pratik ve işlevsel bir hakikatı anımsamak şoka benzer bir etki yaratıyor. Her iletişim aracı, ortamı ya da teknolojisinin ''ileti''*si, kişilerarası ilişkilerde, insanların işlerinde, yaşam hız ve ölçeklerinde yol açtığı değişikliklerin tümüdür. Başka bir deyişle bir iletişim ortamının ya da teknolojisinin iletisi onun toplam etkisiyle aynıdır. Örneğin hareketlilik ve ulaşım, insan toplumlarına ilk kez demiryolları ile birlikte girmedi, ancak, demiryolu var olan bu işlevlerin hızını ve ölçeğini değiştirdi.*

Ama çoğu zaman iletişim araç ve ortamlarının kendi niteliklerini, bu ortamların 'içeriği' yüzünden gözden kaçırıyoruz. Örneğin IBM yalnızca büro gereçleri yapan bir işletme olmadığının, asıl iş alanının bilgi-işlem olduğunun, ancak çok sonraları farkına vardı; bunu farketmekle de yönünü daha bir açıkgörüşlülükle belirleyebildi. Kârının önemli bir bölümünü elektrik ampullerinden sağlayan General Electric ise, hâlâ kendisinin de en az AT&T kadar bilgi taşıma işi içerisinde olduğunun farkına varamadı.

Birkaç yıl önce General David Sarnoff, Notre Dame Üniversitesi'nden bir onur derecesi alırken şunları söylemiş: ''Teknolojiyi kullananların günahlarını teknolojik araçların kendisine yüklemeye fazlasıyla yatkınız. Çağdaş bilimin ürünleri, kendilerinde iyi ya da kötü değildir; aksine, bunların değerlerini belirleyen nasıl kullanıldıklarıdır.'' *Burada konuşan çağdaş uyurgezerliğin sesidir. Ha bunları söylemişiz, ha ''çiçek hastalığı virüsü kendinde iyi ya da kötü değildir; değerini belirleyen nasıl kullanıldığıdır'' ya da ''ateşli silahlar kendinde iyi ya da kötü değildir; değerlerini belirleyen nasıl kullanıldıklarıdır'' demişiz. Yani demek oluyor ki, kurşunları doğru hedefleri buluyor, doğru insanları öldürüyorsa, silahlar iyidir. Televizyon ekranı, doğru cephaneyi gereken insanlara fışkırtıyorsa, iyidir.*

Teknolojiye gösterilen ve bir geri zekalının donuk tavrını andıran bu tepki şu gerçeği gözden kaçırıyor: Bir iletişim ortamının 'içeriği', alıcının aklını çelmek için kullanılan bir yem gibidir. Bir kere, her iletişim ortamının içeriği de bir iletişim ortamıdır ve bu yüzden de, ilk ortamın etkisi daha da yoğun ve güçlü olur. Örneğin sinemanın içeriği, ya bir roman, ya bir oyun, ya da bir operadır. Ancak sinema biçiminin etkisi filmin içeriğine bağlı değildir. Nasıl ki telgrafın içeriği baskı, baskının içeriği yazıysa, yazının da içeriği konuşmadır; ama okuyucu çoğunlukla baskının da, konuşmanın da farkına varmadan okur.

* * *

Gerçi Arnold Toynbee iletişim araç ve teknolojilerinin tarihteki belirleyici rolünü anlamaktan çok uzaktır ama dikkatli bir okuyucu, Toynbee'nin tarihinde bu konuda sayısız örnek bulabilir. Toynbee, şark toplumlarının çağdaş teknolojileri ve bu teknolojilerin siyasal sonuçlarını kabul ettiklerini söyledikten sonra, ''Ancak bu teknolojilerin kültürel etkileri hakkında böyle yaygın bir görüş birliği yok'' *diyor. Bu, okur-yazar ve çok aydın bir adamın, reklamlarla dolu bir ortamda ''Ben reklamlara aldırmam'' demesine benziyor. Şarklıların bizim teknolojimiz karşısında gösterdikleri manevi ve kültürel çekinceler hiçbir işe yaramayacaktır. Çünkü teknolojinin etkileri kanılar ve kavramlar düzeyinde değil, algılama biçimleri ve farklı duyumlar arasındaki oranlar düzeyinde gerçekleşir ve bu düzeydeki etkiye karşı konamaz.*

Ateşli İletişim Ortamları; Serinkanlı İletişim Ortamları

Radyo gibi, sinema gibi ateşli iletişim ortamlarını, telefon gibi, televizyon gibi serinkanlı ortamlardan ayıran temel ilke şudur: Ateşli bir ortam tek bir duyumun gücünü, çok tanımlayıcı bir biçimde artırır. Burada tanımlayıcılık ''çok fazla veriyle yüklü olma'' anlamında kullanılıyor. Örneğin bir fotoğraf, görsel olarak, bu anlamda çok tanımlayıcıdır. Oysa bir karikatür çok daha az tanımlayıcıdır, çünkü bu ortamda sağlanan görsel veri çok daha azdır. Bu yüzden seyircinin daha çok yorum yapması gerekir.

Konuşma da serinkanlı bir ortamdır, çünkü konuşmada konuşanın sağladığı veriler azdır, dinleyenin etkin olarak katkıda bulunması gerekir. Oysa ateşli ortamlar böyle etkin bir katılım olanağı sağlamazlar. Bunu genelleyebiliriz: Serinkanlı ortamlar alıcının katılımının yüksek olduğu ortamlardır, oysa ateşli ortamlar, verilerin daha yüksek bir oranının ortamın kendisi tarafından sağlandığı, dolayısıyla alıcının daha edilgin ve katılımının daha az olduğu ortamlardır. Bu ölçüte göre, örneğin bir

ders, bir seminere kıyasla daha ateşli bir ortamdır.

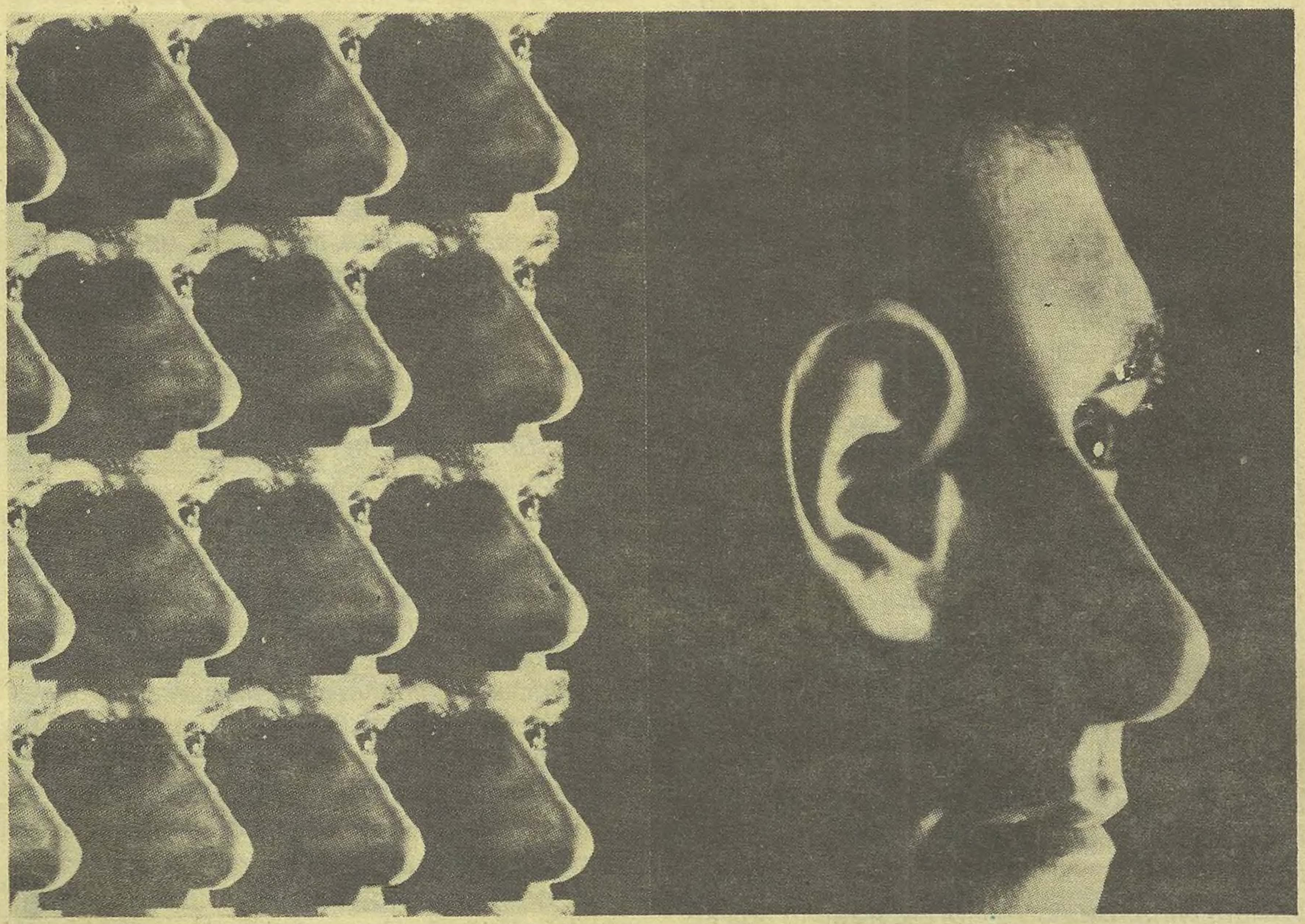
Tarih boyunca insanın icad ettiği belki de en ateşli iletişim teknolojisi alfabetik yazı sistemidir. Bildiğimiz fonetik alfabe kelimelerin anlamı ile görsel biçimleri arasındaki ilişkiyi tamamen koparır ve ilettiği bütün bilgileri yalnızca görsel veriler aracılığıyla sağlar. Alfabenin mümkün kıldığı düşünce ve bilim sistemleri, kendileri görsel olmayan kişilerarası ilişkilerin ve toplumsal süreçlerin görselleştirilmesinden ibarettir. Simgeci şair Edgar Allan Poe'nun dediği gibi, böyle bir alfabeyi kullanarak, ''yalnızca yazı yazma eyleminin bile düşünceyi daha mantıksal kılmak gibi bir etkisi vardır''.

Alfabe Kültürleri

Yanlış anlaşılmasın, burada sözkonusu edilen herhangi bir yazı sistemi değil, fonetik alfabeye dayanan yazı sistemi. Ancak fonetik alfabe insanda gözün yerini kulağın almasına ve aşiret düzeninin uygarlığa dönüşmesine yol açabilir. Serinkanlı bir iletişim ortamı olan hiyerogliflerin ve Çin ideogramlarının etkisi, ateşli ve hatta patlayıcı bir ortam olan fonetik alfabenin etkisinden çok farklıdır. Örneğin Çin kültürü Batı dünyasından çok daha incelmiş ve duyarlı bir kültürdür ama yine de Çinliler kapalı bir aşiret düzeni içerisinde, gözle değil kulakla yaşayan insanlardır.

Karl Popper ***Açık Toplum ve Düşmanları*** *üzerine kaleme aldığı incelemesinin ilk bölümünde, Eski Yunan'da kapalı aşiret düzeninin çözülmesini ve bu çözülmeye gösterilen tepkileri anlatıyor. Ancak ne Eski Yunan'dan, ne de çağdaş toplumlardan söz ederken duyumlarımızın teknolojik uzantıları olan iletişim ortamlarının toplumların açık ya da kapalı toplumlar olması üzerindeki etkilerini gözönünde bulunduruyor. Bütün betimlemelerini ve açıklamalarını ekonomik ve siyasal bir bakış açısından yapıyor.*

Popper'a göre kapalı toplumlar (ya da aşiret düzenleri) biyolojik bir birliğe sahip olan top-



Her teknoloji belli bir insan yetisinin ya da duyumunun geliştirilmesi ve uzmanlaştırılmasıdır. Belli bir duyum, teknoloji sayesinde çok fazla güçlendiği takdirde duyumlarının arasındaki oran bozulur. Fonetik alfabenin ve özellikle baskı tekniğinin icadından beri yalnızca göz tarafından biçimlendirilmiş bir dünyada yaşıyoruz. Ancak elektronik çağında bütün yetilerimizi birden kullanmamız gerekeceğinden, göz ile kulak arasındaki bozulmuş denge yeniden kurulabilir.

lumlardır. Oysa ''çağdaş açık toplumun işleyişi değişim ve işbölümü gibi soyut ilişkiler tarafından belirlenir''. *Kapalı toplumların açılmasında (ya da soyutlaşmasında) farklı bir yazı biçimi ya da teknolojisindense fonetik alfabenin benimsenmesinin başlıca etken olduğu,* ***Gütenberg Galaksisi*** *adlı kitabımın temel savıdır.*

Nasıl çağdaş toplumlar alfabetik toplumlar ise, aşiret düzenine dayalı kapalı toplumlar da konuşmanın, tamtam davullarının, gözden çok kulağın uzantısı olan teknolojilerin egemen olduğu toplumlardır.

Bu toplumlarda, fonetik yazı, düşünce ile eylemi birbirlerinden ayırmadığı sürece, kendisini kendinden daha büyük bir organizmanın bir parçası olarak duyan her insanın eylemlerinden olduğu kadar düşüncelerinden de sorumlu tutulması kaçınılmazdır. Ne zaman ki, fonetik yazı gözün tarafsız dünyasını, kulağın ve tensel temasın tılsımlı dünyasından kopardı, işte o zaman bu kopuştan bireysel insan doğdu. Bu yüzden, fonetik alfabenin icadından sonra ilk kez tarih sahnesine Eski Yunan'da ortaya çıktığından bu yana okur-yazar insan hep bölünmüş bir insan, bir şizofren olmuştur.

Gütenberg Galaksisi

Hızlı bir geçiş dönemi dediğimizde, iki kültürün, çatışan iki teknolojinin birbirleriyle karşılaştığı sınırda yer alan bir dönemi kastederiz. Rönesans, ikibin yıllık bir alfabe ve elyazması kültürü ile tekrarlanabilirliği ve sayıya dökümü mümkün kılan yeni mekanizmalar çağının karşılaştığı sınırın adıdır. Bugün geçmiş beşyüz yıllık bir mekanizma çağının mirası olan birörneklik ile yeni elektronik teknolojilerin getirdiği eşzamanlılığın karşılaştığı sınırda yaşıyoruz.

Makine çağına özgü bir teknoloji olan baskı tekniği ile hazırlanmış bir metni okuyan kişi ile metnin yazarı arasındaki ilişki, elyazması bir metni okuyan kişinin yazarla kurduğu ilişkiden çok farklıdır. Bir kere, baskı, eskiden evrensel okuma tarzı olan ''yüksek sesle okuma''yı giderek gereksiz kıldı. Daha da önemlisi, baskı, okurun kendisini tamamen yazarın ''ellerine teslim etme'' eğilimini alabildiğine güçlendirdi.

William Ivins, ***Baskı ve Görsel İletişim*** *adlı kitabında basılı metinlerin görsel alışkanlıklarımız üzerindeki etkisini şöyle açıklıyor:* ''Her yazılı ve basılı kelime aslında, belirli bir çizgisel sıra içinde kaslarımızı, son çözümde ses çıkaracak şekilde kımıldatmak için verilmiş bir komuttan ibarettir. Ancak son çözümde çıkarılacak sesler lehçeye göre farklılık gösterebilir, dolayısıyla okumanın özünü çıkarılması düşünülen sesler oluşturmaz.'' *Bu bölümde Ivins, yalnızca baskı tekniğinin bize kazandırdığı çizgisel ve dizisel alışkanlıklara dikkat çekmekle kalmıyor, daha da önemlisi, bir tipobaskı kültüründe yaşantıların yalnızca görsel boyutlarının öne çıkarılarak birörnekleştirildiğini, seslerin ve diğer karmaşık duyumların geri plana itildiğini gösteriyor. Bütün yaşantıların tek bir duyuma, görsel olana indirgenmesi ise, bir ''bakış açısı benimseme'', dünyaya değişmez bir noktadan bakma alışkanlığını kazandırır.*

Bilişim Çağı

Geride bırakmakta olduğumuz makine çağının temel ilkesi, kullanışlılık ve gelişme adına biçimlerin ve işlevlerin ayrışması ve bölünmesiydi. Buna bağlı olarak kültür/teknoloji, sanat/ticaret, tatil/iş gibi kutuplaşmalar doğdu. Tatille işin kutuplaşması, tatilin 'boş zaman' ve hatta giderek tembellik olarak yaşanmasına yol açtı. Oysa bilişim çağı bütün yetilerimizi eşzamanlı olarak kullanmamızı gerektirdiğinden, bundan böyle, bütün çağların büyük sanatçıları gibi biz de en keyifli tatilin kendimizi sevdiğimiz bir işe kaptırdığımız zaman yaşandığını farketmeye başlayacağız.

Bugün hızla insanın teknolojik uzantılarının son aşamasına varıyoruz - bilincin, merkezi sinir sistemi teknolojik olarak yeniden yaratılması. Böylelikle, nasıl çeşitli iletişim araçları sayesinde sinir uçlarımızın ve duyu organlarımızın kapsamını genişlettiysek, artık bilincin yaratıcı süreçlerinin de bütün topluma yayılabilmesi mümkün olacak.

Gütenberg teknolojileri okur-yazarlara vahim toplumsal sorunlara soğuk bir yansızlıkla yaklaşma olanağı sağladı. Ama bu yansızlık bir katılımsızlık tavrıdır. Elektrik çağında merkezi sinir sistemi bizi bütün insanlığın yazgısına ortak edecek biçimde gelişmişken, her eyleme zorunlu olarak katılmak durumundayız. Artık Batılıların soğuk ve yansız tavrını sürdürmek olanaksız hale geldi. Bu yüzden bütün sınırlılığı ve uzmanlaşmışlığı ile 'bakış açısı' kavramı işlevini yitirdi. Duyumlarımız arasında yeni bir oran ve denge kurulmak üzere. Elektriğin küçülttüğü dünya, insanlığın ortak köyü haline gelecek.

(Marshall McLuhan'ın UNDERSTANDING MEDIA ''İletişim Ortamlarını Anlamak'' ve GUTENBERG GALAXY ''Gütenberg Galaksisi'' kitaplarından derlenmiştir.)

TEST 4'ün Çözümleri

1 / b

---

2 / a - 106 komutu RETURN sözcüğünü içermektedir. Bu, atlama komutundan sonra geri dönme mekanizmasını harekete geçirir. 106 komutundan sonra, program 20 komutuna geçer.

b - Burada, RETURN komutu yoktur; bu durumda program 106 komutundan sonra sırayı takip eder: 107 ve sonraki komutları yürütür.

---

3 / b - AMB, aritmetik ve mantık biriminin kısaltmasıdır. Adından da anlaşılabileceği gibi, aritmetik ve mantık işlemlerini yürütür.

---

4 / a

---

5 / c

---

6 / Hata, 33. komuttaki RETURN sözcüğünün varlığından kaynaklanmaktadır. Bu sözcük yalnızca bir atlamadan sonra yer alan bir geriye doğru atlama (ya da ilk atlama noktasına geri dönme) için kullanılır. Oysa bu program 33. satıra gönderen hiçbir atlama içermemektedir. Diğer yandan 57. komuttaki RETURN sözcüğü doğru olarak kullanılmıştır, çünkü daha önce (32. komutla) bir atlama yapılmıştır.

---

7 / b - Müdahale (araya girme), sistemin kaynaklarını, istenilen bir hizmete tahsis etmek üzere işlemi dışardan askıya alma olanağı verir.

---

8 / a - Yanlış. EPROM'lar programları biriktirmeye ayrılmışlardır. Özel aygıtların aracılığı ile bunları silme veya yeniden yazma olanağı vardır ama, verilerin belleğe yerleştirilmelerine hizmet etmezler.

b - Yanlış. Disk birimi 250.000'den birkaç milyona kadar karakter barındırabilir (büyükleri için bu birkaç yüz milyona varır). Bir kişisel bilgisayarın bellek kapasitesi 64.000 karakter düzeyindedir.

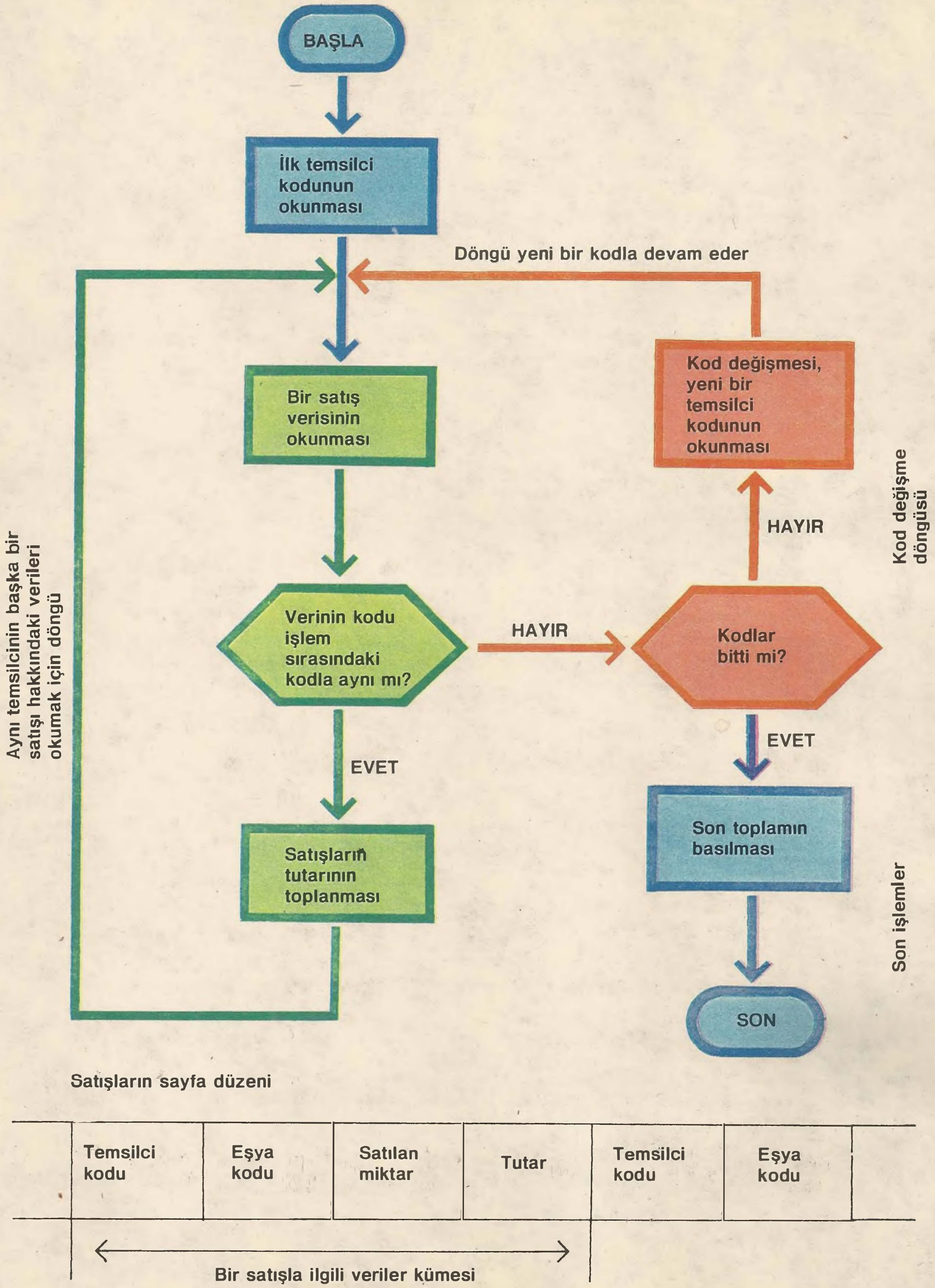
c - Doğru. Bir ROM'u programlamak için özel donanım gerekir.

d - Yanlış. MİB (merkezi işlem birimi) yürütme sırasındakileri belleğe yerleştirir ve yeni bir eyleme döner. Araya girmenin sonunda programın normal akışını sürdürür.

e - Bu yargı kısmi olarak doğrudur. Ekran, klavyenin aracılığı olmaksızın, kendi başına bir müdahaleye yol açamaz.

f - Doğru. Bu çevresel gereç ile değiş-tokuş edilen verilerin sayısı çok fazla olduğundan DBE (doğrudan bellek erişimi) aktarım hızını artırmanın tek yoludur.

KOD DEĞİŞMESİNE UĞRAMIŞ BİR DÖNGÜNÜN AKIŞ ŞEMASI: UYGULAMA ÖRNEĞİ



Satışların sayfa düzeni

	Temsilci kodu	Eşya kodu	Satılan miktar	Tutar	Temsilci kodu	Eşya kodu	

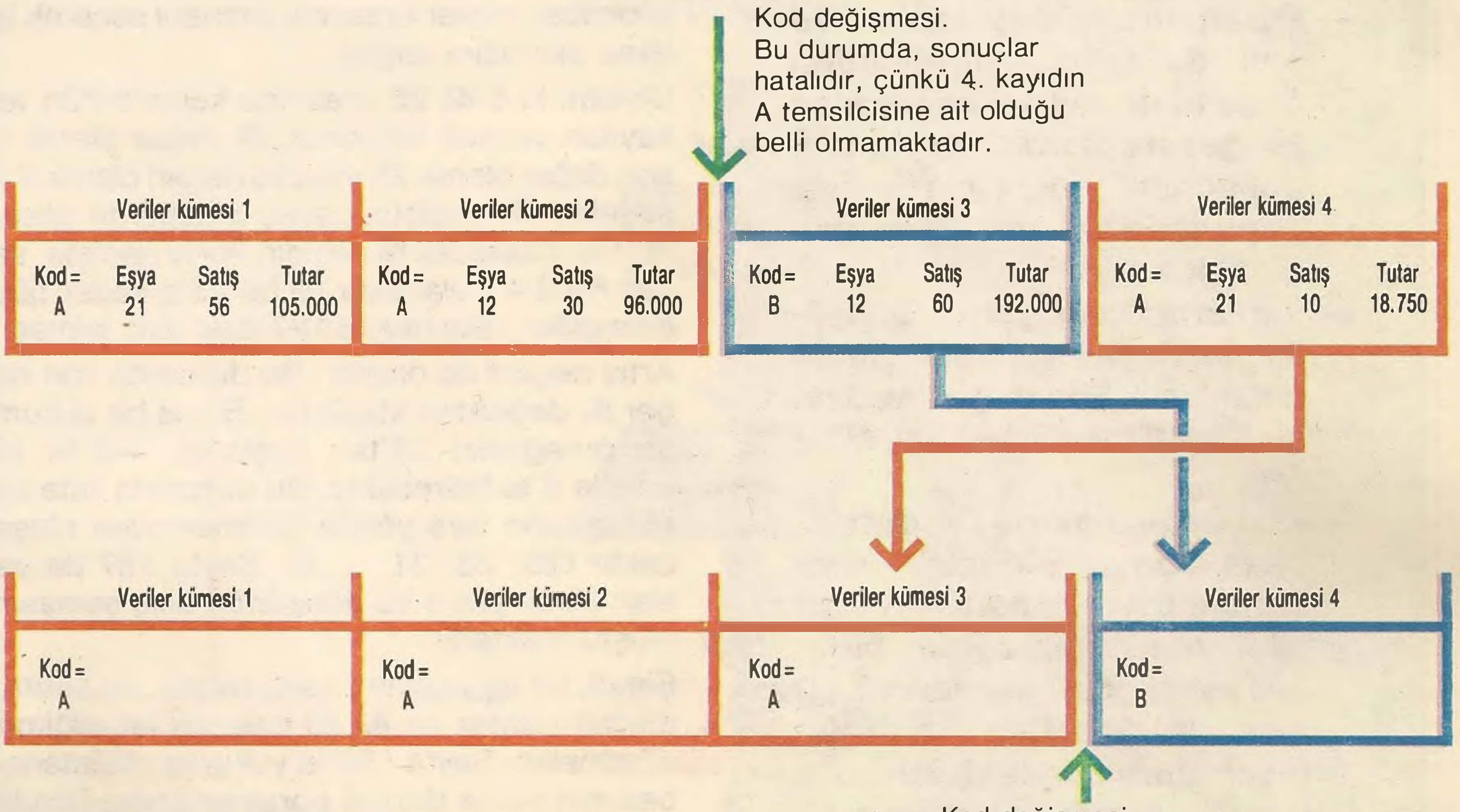
Bir satışla ilgili veriler kümesi

bir satıcının yaptığı satışların sayısını belirlemektir. Kod olarak her satıcıya bir numara verilir.

Yapılması gereken işlem aynı satıcı koduna sahip bütün satış kayıtlarının toplanmasıdır. Satıcı kodu değiştiği takdirde toplama döngüsü durdurulacak ve toplam yazılacaktır.

Sayfa 184'te bu sorunun akış şeması ve sonuçlarının nasıl gösterilebileceğine dair bir örnek yer almaktadır. Bu yöntemin kullanıldığı durumlarda, kodların sıralanmış olması gerekir. Kodlar artan ya da azalan bir sıraya sokulmazsa cevaplar hatalı olur. Diyelim ki yalnızca iki temsilci (kodları A ve B) var ve satışlarla ilgili bilgiler aşağıdaki şemada göreceğimiz gibi sırasız olarak veriliyor. Kod değişmesine bağlı olarak döngünün durdurulması halinde birinci satıcının bütün satışları dikkate alınmayacak ve sonuç olarak 4. veri kaydı dikkate alınmadığı için de toplam satışlar hatalı olacaktır. Öyleyse döngünün yürütülmesinden önce, verilerin, örneğin artan bir biçimde sıralanması gerekir. Böylece 3. ve 4. veri kayıtları yer değiştirmiş olacak ve kod değiştiğinde A satıcısının bütün satışları toplanabilecektir.

KOD DEĞİŞMESİ



Büyük bilgisayar terminallerinde çalışan işletmenler.

Özet olarak başlıca döngü tipleri şunlardır:

— Belirli Sınırları Olan Döngüler: Başlangıç ve son değerleri, döngüde sayısal biçimde belirtilirler.

— Parametreli Döngüler: İlk ve son değerleri, yürütmeden önce birer sayısal değer alan simgesel adlardır. Bu yöntem döngünün, programın değişik yerlerinde kullanılabilmesine olanak verir.

— Verilerin Sonu İşaretli Döngüler: İlk değer normal biçimde tanımlanır. Son değer ise özel bir kodun tanınmasından çıkarılır.

— Kod Değişmesiyle Durdurulabilen Döngü: Birincinin veya sonuncu veri kodlu kayıtları işler ve iki dala ayrılır; birincisi aynı kodlu işlemleri yürütmek için, diğeri ise kod değişmesi durumundaki işlemler için. Bu durumda verilerin sınıflandırılması kaçınılamayacak bir şeydir.

Çok Adımlı Döngüler

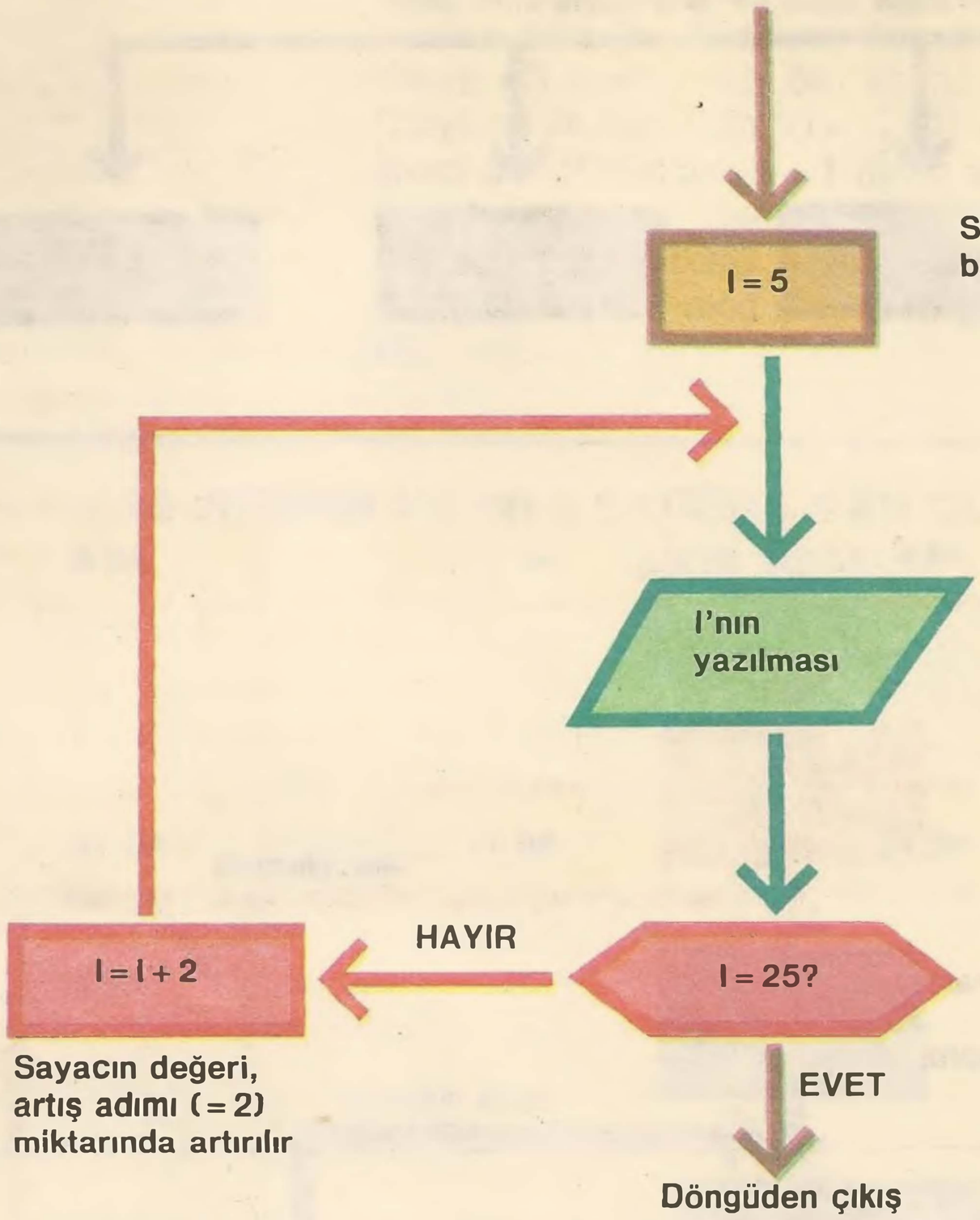
Bir döngünün sayacı, yani tekrarlanan kısmın her geçişinde artırılan değer (bakınız, s.187'-nin yukarısındaki şema), her defasında 1 artmak zorunda değildir. Sayacın artış değerine adım **(step)** denir. Adım değerinin birden fazla olması, veriler arasında isteneni seçerek işleme olanağını sağlar.

Diyelim ki 5 ilâ 25 arasında kalan bütün tek sayıları yazmak istiyoruz: İlk değer olarak 5, son değer olarak 25 ve artış değeri olarak 2'yi seçelim. İlk geçişte, sayaç 5 değerini gösterir. Bu, basılacak ilk sayıdır. İkinci geçişte, sayaç 5 + 2 = 7 olacaktır ve bu 25'e kadar böylece gider. (Bakınız, s.187'deki akış şeması.)

Artış negatif de olabilir: Bu durumda son değer ilk değerden küçüktür. Böyle bir durumda örneğimizi 25'ten başlayıp, —2'lik bir adımla 5'te bitirebiliriz. Bu durumda liste aynı sayıların ters yönde dizilmesinden oluşacaktır (25, 23, 21 ... 5). Sayfa 187'de yer alan ikinci şema bu döngünün akış şemasını oluşturmaktadır.

Şimdi, bir uygulama örneği olarak, üç satırda dörder dörder on iki ad basmak istediğimizi düşünelim. Sayfa 188'in yukarısında istenen basımın sayfa düzeni görülmektedir. Basılacak veriler (yani, 1'den 12'ye kadar numaralanmış olan adlar) bir belleğin bir yerinde aşağıdaki şekilde görüldüğü gibi, dizisel olarak kayıtlıdır.

2 ARTIŞLI DÖNGÜ ÖRNEĞİ

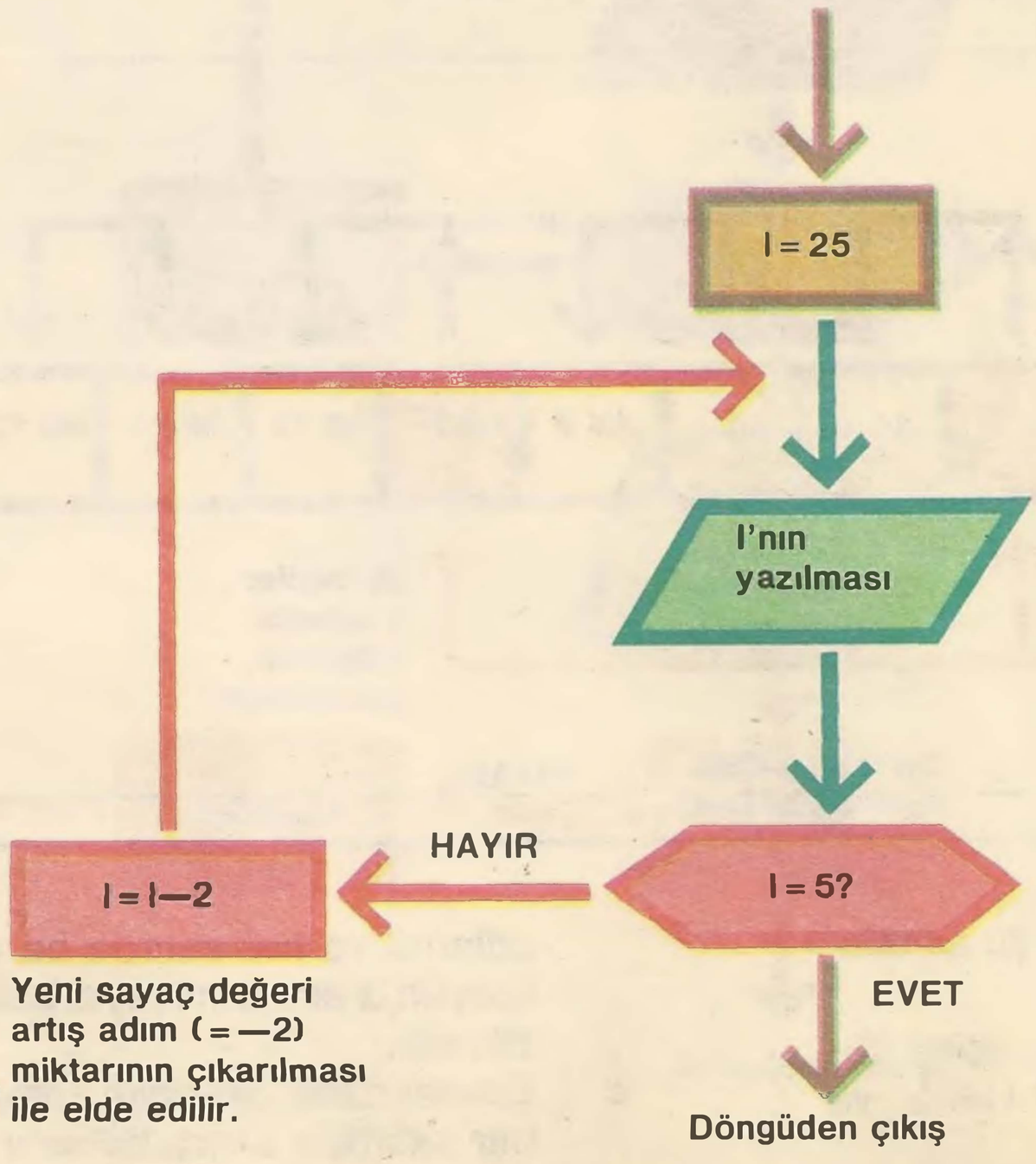


Sayacın başlangıç değeri

Bu örnekte (1 ile 25 arasındaki tek sayıların yazılması), sayacın değeri istenen sonucun kendisidir. Bilgiyazımda şu sıralama elde edilecektir: 5, 7, 9, 11 ve 25'e kadar.

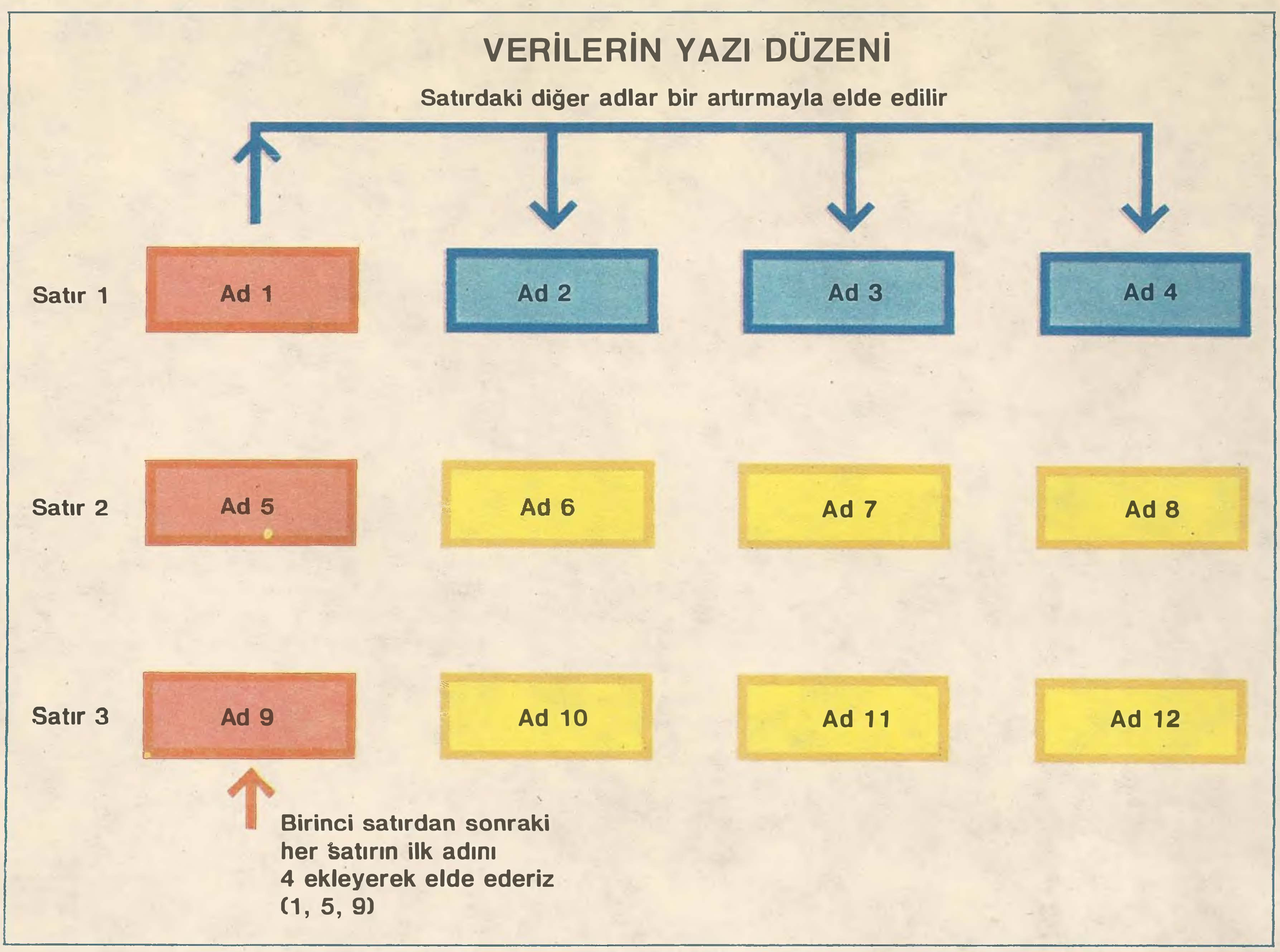
Sayacın değeri, artış adımı (= 2) miktarında artırılır

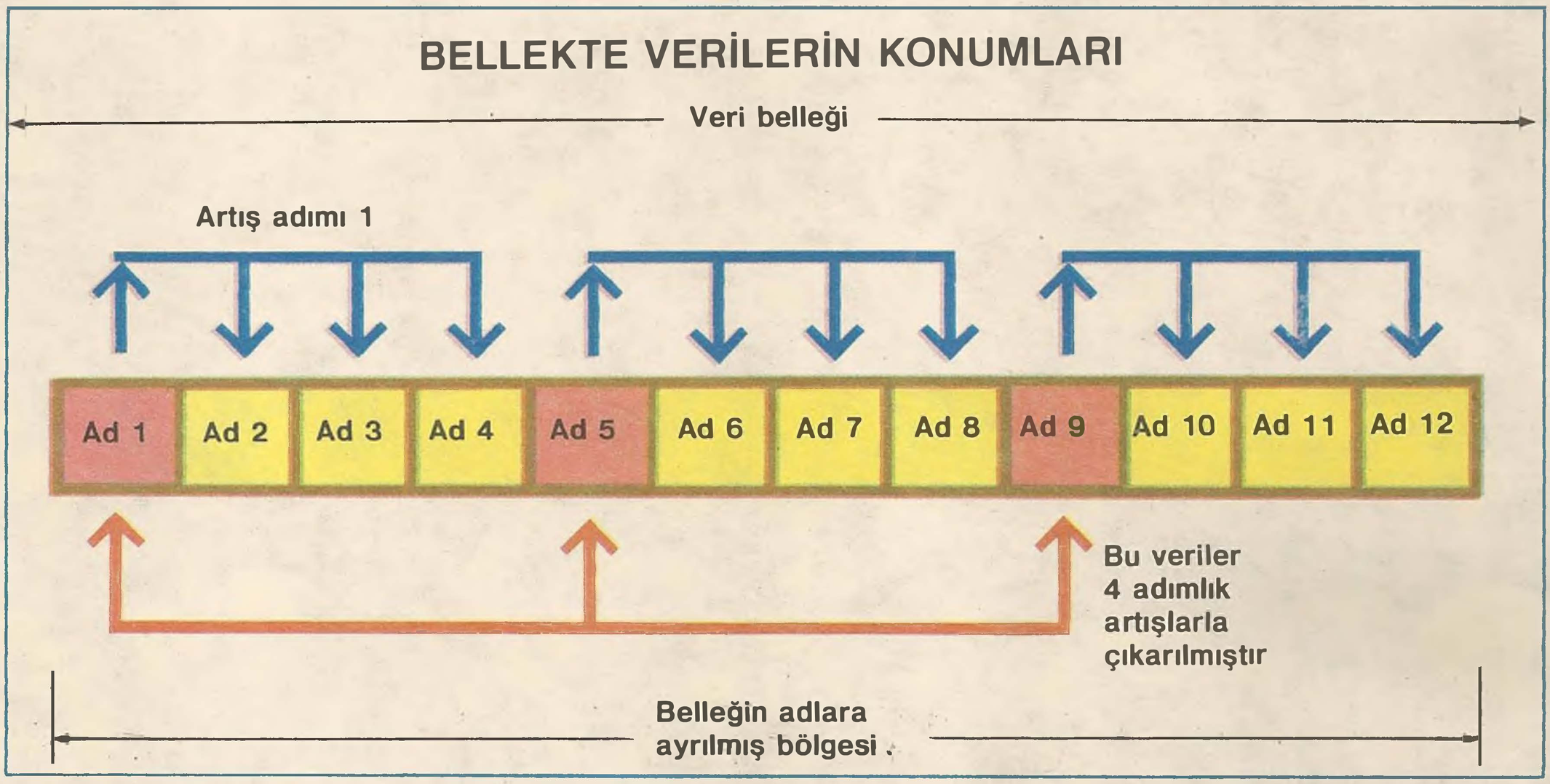
—2 ARTIŞLI DÖNGÜ



Bu durumda, sonuçların çıkması bir önceki örneğin tersi yönünde olur: 25, 23, 21 ve 5'e kadar.

Yeni sayaç değeri artış adım (= —2) miktarının çıkarılması ile elde edilir.





İstenen düzenin elde edilmesi için şu adımların izlenmesi gerekir:

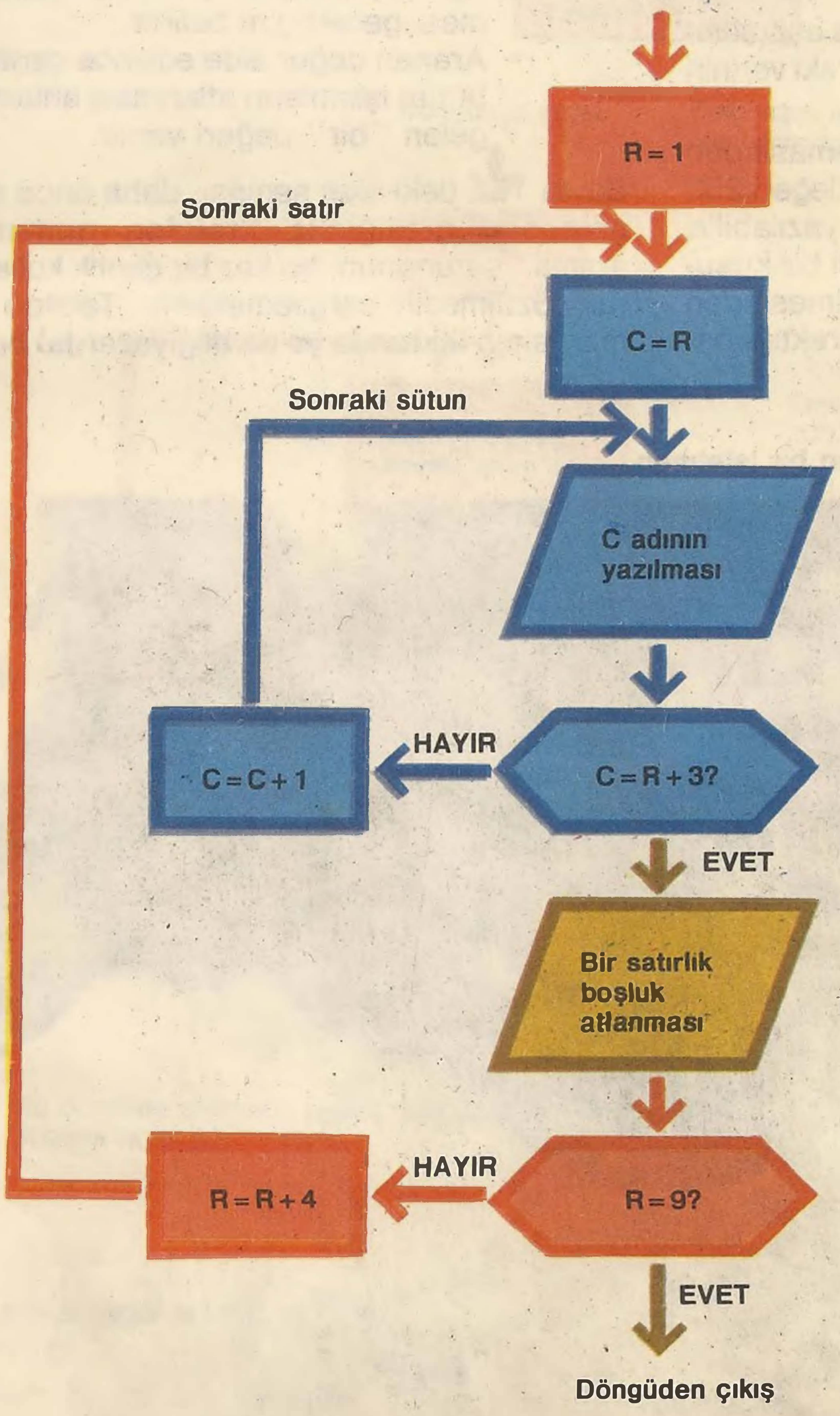
a / Satır başındaki verinin seçimi: Sırayla 1., 5. ve 9. adlar (1., 2. ve 3. satırlar için).

b / Seçilen verinin (yani 1., 5. ve 9. adların) ve her satırda bu verileri izleyen 3'er verinin aynı satıra yazılması.

c / Listeyi daha okunaklı kılmak için her satırdan sonra bir satır atlanması.

a aşamasını, ilk değeri = 1, son değeri = 9 ve adımı = 4 olan bir döngüyle halledebiliriz. **b** aşaması için ise, ilk değeri birinci döngünün sayacı olan (1, 5, 9) ve son değeri de bu sayaç + 3 olan ikinci bir döngü oluşturulur (böylece ilk satır için İD = 1, SD = 4 olur). Bu ikinci döngünün adımı = 1'dir. **c** aşaması için, iki satır arasında ''satır atlama'' (LF) kodunun gönderilmesi gerekir. Aşağıdaki şekil üç aşamanın tümünün akış şemasını göstermektedir.

12 VERİNİN (ADIN) 3 SATIRDA 4'ER OLARAK BASILMASININ AKIŞ ŞEMASI



Döngüden Çıkış

Döngüler belirli bir verinin aranması gerektiğinde sık sık kullanılır. Diyelim ki arkadaşlarımızın adlarını ve telefon numaralarını belleğe kaydettik ve toplam 200 veri kaydımız oldu. Her bir kayıt soyadı + numara şeklinde düzenlenmiş. Mahmut Güzelce'nin telefon numarasını öğrenmek istiyoruz. Şu işlemleri gerçekleştiren bir döngü yazmamız gerekir:

a / Bellekteki bir veri kaydının okunması.
b / ''Güzelce'' soyadı ile veri kaydının içerdiği soyadının karşılaştırılması.
c / İki soyadı aynıysa telefon numarasının kullanıcıya bilgiyazar aracılığıyla aktarılması.
d / İki soyadı aynı değilse a aşamasına geri dönerek bir sonraki verinin okunması.

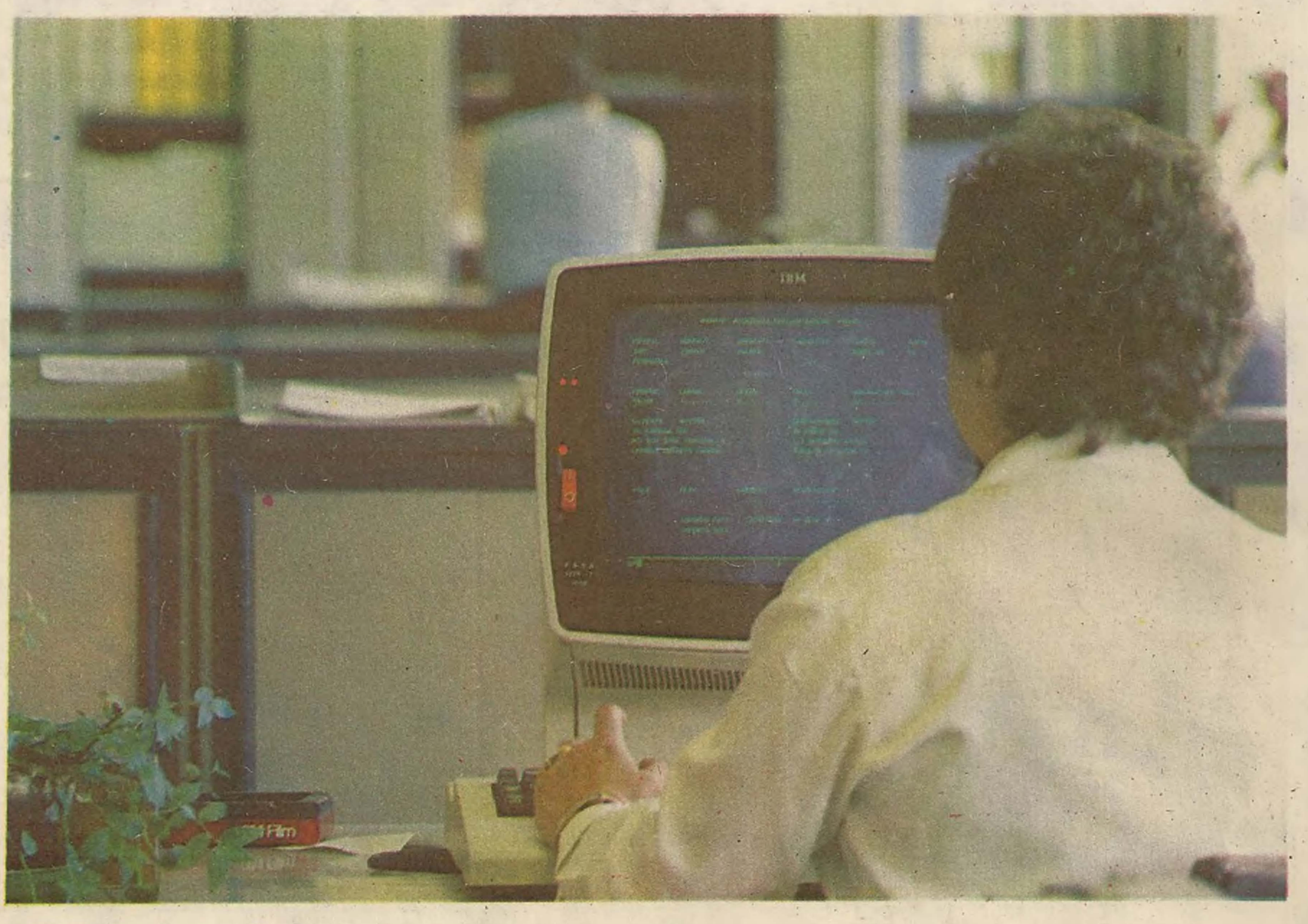
Bu program sayfa 191'deki akış şemasından yararlanarak ve ilk değeri 1 ve son değeri 200 olan bir döngü kullanarak kolayca yazılabilir. Gene de böyle bir çözümün önemli bir kusuru vardır: İstenen bilginin elde edilmesinden sonra arama işleminin durması gerektiği öngörülmemiştir. Dolayısıyla boşuna zaman harcanacaktır. Özellikle veriler bir diskette ise okuma için harcanacak vakit önemsiz değildir. (Bu aşama akış şemasında ''I verisinin okunması'' diye gösterilmiştir.) Bu şemaya göre okuma üçüncü veri kaydında durdurulabileceği halde iki yüzüncü veri kaydına kadar sürdürülmektedir. Bu süreyi azaltmak için bir ''belirtici'' ya da ''çentik''e **(flag)** başvurulur. Çentik aramanın sona erdiğini belirtir. Çentik 1 değerini aldığında okuma ve karşılaştırma işlemleri sona erer ve döngüden çıkılır.

Şimdi genel olarak çentiklerin nasıl kullanıldığını görelim:

- Başlangıçta çentik sıfırdadır ve çentiğin sıfır değeri işlemlerin yürütülmesi gerektiğini belirtir.
- Aranan değer elde edilince çentiğe, bütün işlemlerin atlanması anlamına gelen ''bir'' değeri verilir.

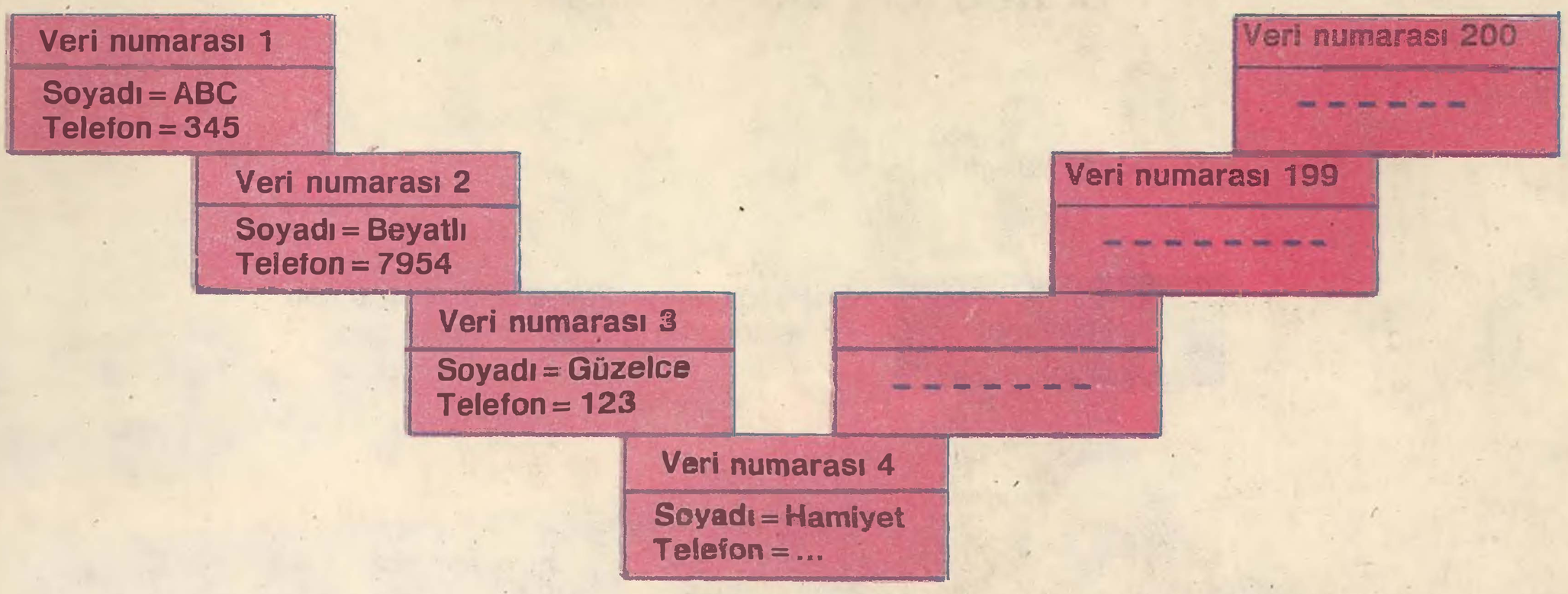
Sayfa 192'deki akış şeması, daha önce akış şemasını çizdiğimiz ''telefon numarası arama'' sorununun, bu kez bir çentik kullanarak çözülmesini sergilemektedir. Telefon numarasının (ekranda ya da bilgiyazarda) basıl-

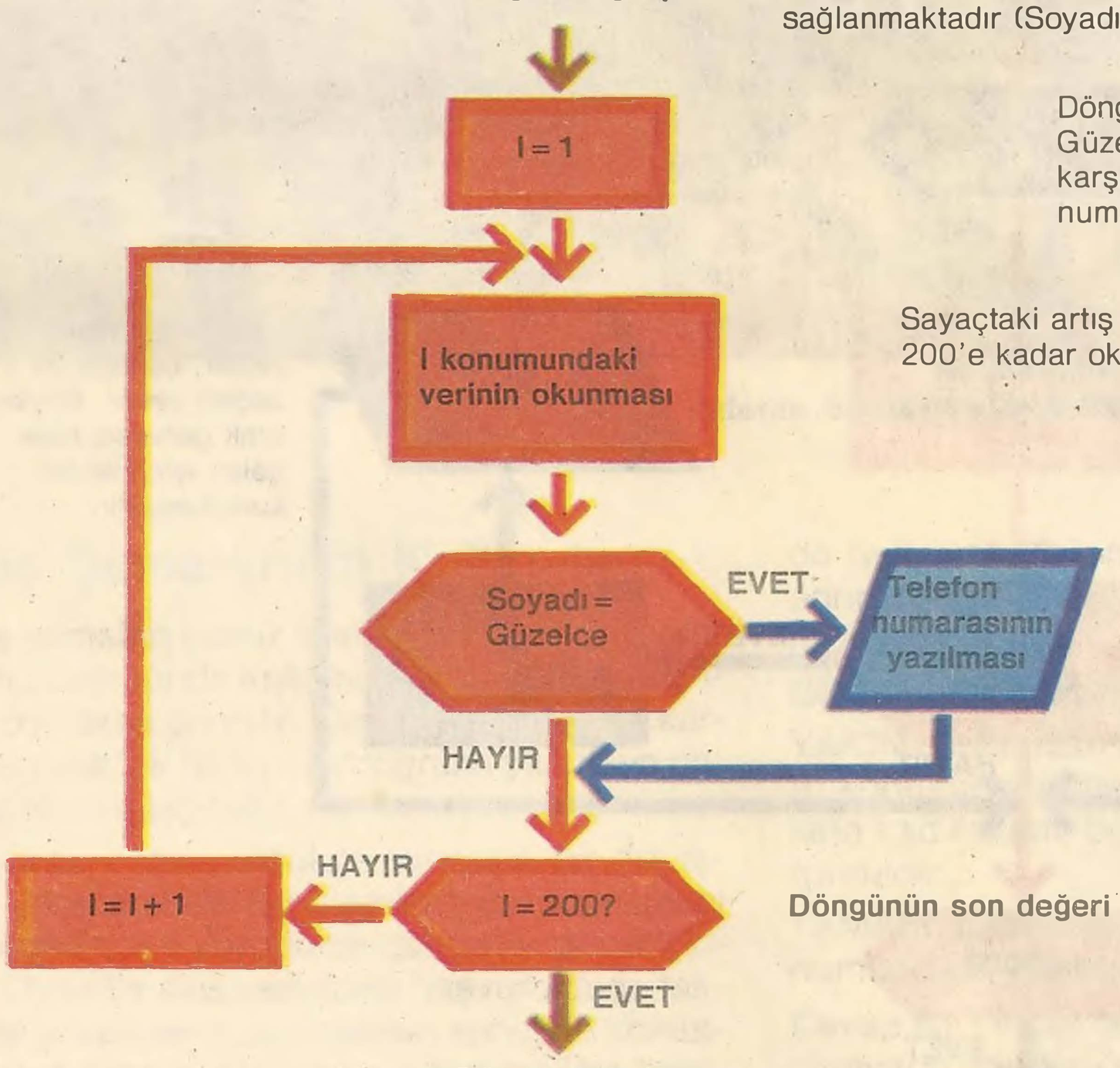
Bir arama döngüsünün akışını izleyen bir işletmen.



VERİ ARAMA DÖNGÜSÜNÜN AKIŞ ŞEMASI

Akış şemasında kullanılan verilerin yapısı. Aranılan veri (Güzelce, tel: 123) 3 numaradadır.





Bulunması istenen karşılaştırma terimi sağlanmaktadır (Soyadı = Güzelce)

Döngünün girişi: I sayacı Güzelce soyadı ile karşılaştırılacak verinin numarasıdır

Sayaçtaki artış (I) tüm verilerin 1'den 200'e kadar okunmasını sağlar

Bu işlemden sonra, döngü sona erebilir, çünkü aranılan soyadı bulunmuştur

Döngünün son değeri

Bu döngüde işlemlerin sıralanması düzeltilebilir: Aksi takdirde veri bulunmuş olsa da, arama sonuna kadar sürdürülür.

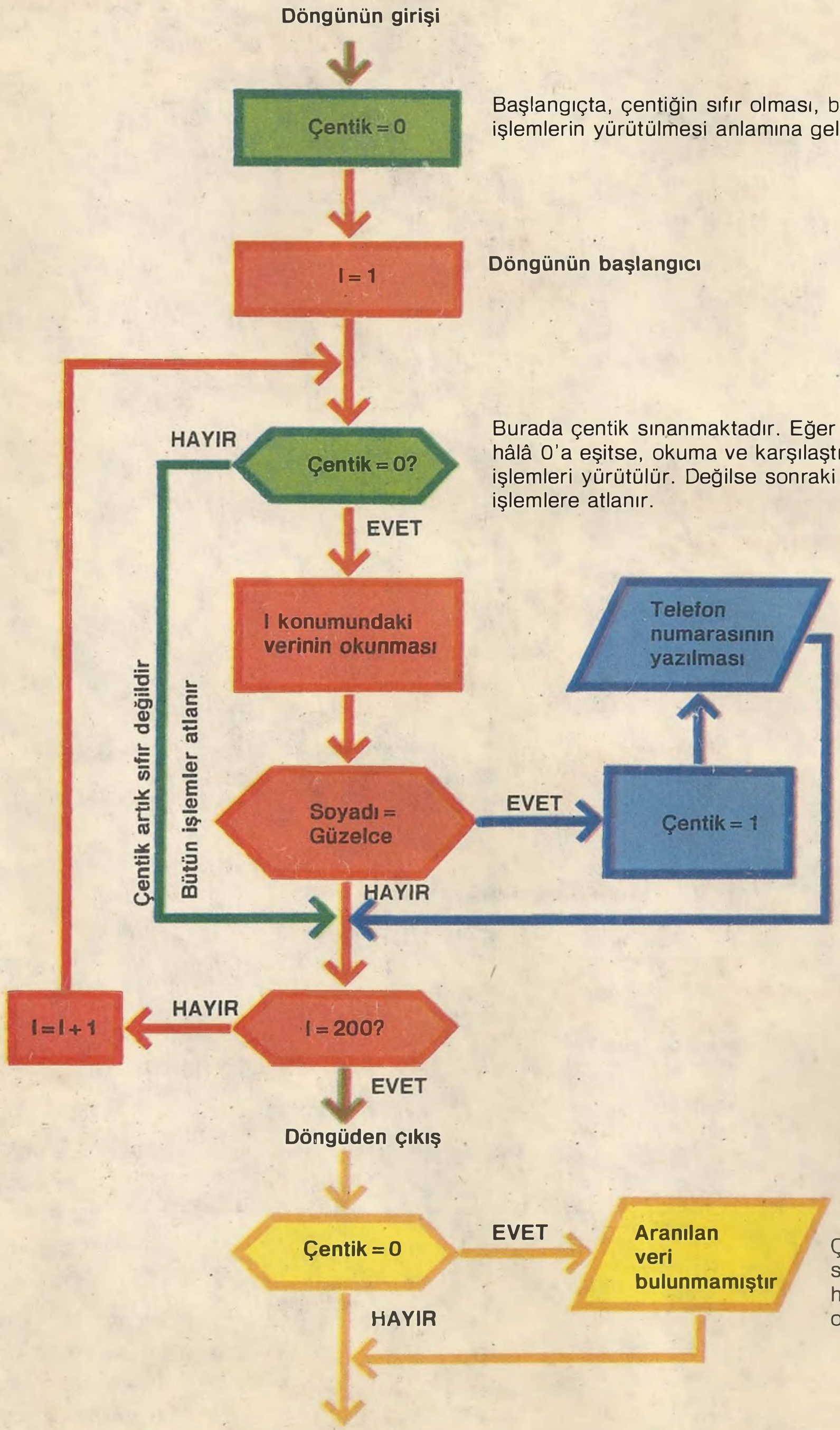
---

ması sayesinde aramanın bittiği hemen belli olmaktadır.

Program bir basım işlemini öngörmeseydi, arama başarılı olmadığı takdirde işlemlerin sürdürülmesi birtakım hatalara yol açabilirdi.

Bir basım işlemini öngörmeyen programlarda bu tür hataları önlemek için bir çentik kullanarak programın devamı belli bir koşulun gerçekleşmesine (ya da gerçekleşmemesine) bağlanır.

BİR ÇENTİK SAYESİNDE ARAMA ZAMANININ KISALTILDIĞI BİR DÖNGÜ İÇİN AKIŞ ŞEMASI



Döngünün girişi

Başlangıçta, çentiğin sıfır olması, bütün işlemlerin yürütülmesi anlamına gelir

Döngünün başlangıcı

Burada çentik sınanmaktadır. Eğer hâlâ 0'a eşitse, okuma ve karşılaştırma işlemleri yürütülür. Değilse sonraki işlemlere atlanır.

Veri bulunmuştur. Veri yazılır, çentiğe de 1 değeri verilir. Böylece, artık gereksiz hale gelen işlemlerden kurtulunabilir.

Çentik aramanın sonunu belli eder. Eğer hâlâ 0 ise, diğeri de olumsuz demektir.

Videolu bilgisayar oyunları evlerde de yaygınlaşıyor.

Akış Şemalarının Kullanılışı

Akış şemaları yalnız işletme yönetim sorunlarının çözümünde kullanılmaz; uygulama alanları çok daha geniştir. Örneğin bilgisayara karşı oynanacak bir oyun programı yazmaya çalıştığımızı düşünelim.

Böyle bir sorun, oldukça karmaşık bir dizi işlem gerektirebilir. Bu işlemlerin çözümlenmesi ise çok titiz ve yorucu bir çalışmayı gerekli kılar. Oysa bir akış şemasına başvurulduğu takdirde çözümleme aşamasının sonunda komutların yazılması, artık büyük bir yanılma tehlikesi taşımayan bir çevirme çalışmasından ibaret olacaktır.

İlginç bir uygulama örneği olarak, ünlü Mastermind oyununun akış şemasını çizmeye çalışacağız. Bu akış şemasının program komutlarına nasıl çevrilebileceğini ise daha ileride inceleyeceğiz.

Önce oyunun kurallarını açıklayalım. Elektronik oyun arkadaşımız (bilgisayar) bizim için, rastlansal dört rakamdan oluşan bir sayı yaratmakta ve bizden bunu en çok on denemede bulmamızı istemektedir. Her denemeden sonra, bilgisayar şu üç cevaptan birini vermektedir:

BOŞ: Hiçbir rakamı bilememişsek.

YAKIN: Rakamları bulmuş, ama bunları doğru konuma getirmemişsek. Bu durumda makine kaç rakamı bulduğumuzu söylemek zorundadır.

TAMAM: Bazı rakamlarla birlikte bunların konumunu da bulmuşsak.

Cevap iki YAKIN bir TAMAM olabilir. Bu, bilgisayarın yarattığı sayıdan, oyuncunun iki rakam bulabildiğini, ama belirttiği konumların yanlış olduğunu, bir üçüncü rakamı ise hem bulduğunu, hem de doğru konumuna getirdiğini gösterir.

Görüldüğü gibi oyunun kurallarının anlatılması, sorunun çözümlenmesine doğru atılmış ilk adımdır.

Çözümlemenin kendisi ise yürütülecek işlevlerin açıkça, ayrıntılı ve titiz bir biçimde sıralanmasından ibarettir. Bu yöntemle her sorun çözümlenip çözülebilir.

Fiber-Optik Teknolojisi

Günümüzde bilgisayarın kitle iletişiminde kullanılmasının yaygınlaşması büyük ölçüde fiber-optik teknolojisindeki gelişmelere bağlı. Çünkü bugüne kadar bütün iletişim ağlarının can damarı görevini yüklenegelmiş bakır telefon hatlarının veri taşıma hızı, bilgisayarın veri işleme hızını karşılayamayacak kadar yavaş. Oysa dalga boylarının kısalığı sayesinde çok büyük sayıda veri kodlayabilen ışık dalgalarının elektrikten yaklaşık % 50 daha hızlı hareket edebilmek gibi bir üstünlüğü var. Herhangi bir hat boyunca taşınan elektronların yarattığı elektrik yükü (elektronik gürültü) akımın yavaşlamasına yol açıyor; oysa ışık parçacıklarında (fotonlarda) böyle bir etki yok. Aradaki hız farkını biraz olsun somut kılabilmek için şöyle bir örnek verelim: Son yıllarda ATT'nin Bell laboratuvarlarında lazer ışınlarıyla yapılan en yeni deneylerde, saniyede 2 milyar dalga içeren ışınlar üretilebiliyor. Böyle bir ışını kıl inceliğinde cam tüplerle saniyede 70 km'ye yakın bir hızla iletmek mümkün. Böyle bir ışınla kodlandığı takdirde elinizdeki fasikülün 2400 kopyasını 1 saniyede 70 km'ye ulaştırmak mümkün olacaktır. Oysa aynı veriyi elektrik sinyallerine dönüştürerek bakır telefon telleriyle iletmek 21 saatimizi alacaktır.

Fiber-optik teknolojisi sanıldığı kadar yeni değildir. Tıbbın bazı özel alanlarında, örneğin endoskopi'de (iç organların incelenmesi) görüntü iletmek için cam liflerden öteden beri yararlanılıyordu. Ancak cam liflerin bu ilk örneklerini uzak yerler arasında bilgi iletiminde kullanabilmek için, önce hem daha ''saydam'' hem de gönderilen iletiyi çarpıtmayacak ''ışık hatları''nın üretilmesi gerekiyordu. Tıptaki kullanım böyle bir uygulama gerektirmediğinden, bu alanda özel bir çalışmanın yapılması için, önce bilgisayarların yol açtığı, uzak mesafelere hızlı veri taşıma ihtiyacının doğmasını beklemek gerekecekti. Gerçi daha bundan 30 yıl önce ışık ''borularının'' yepyeni bir iletişim sisteminin temelini oluşturacağını söyleyen genç araştırmacılar vardı, ama genellikle, kuşkuyla, inanmazlıkla karşılanıyorlardı.

Ancak telematiğin doğurduğu ihtiyaçla birlikte Japon ve Amerikan üreticileri bu alanın olanaklarından yararlanmak için amansız bir yarışa girdi. Bu yarış sayesinde sodyum borosilikat gibi çok bileşenli camların daha hassas bir biçimde arıtılması başarılarak istenen saydamlık derecesi elde edildi. Öte yandan özel bir yöntemle 2000°C ısıda neredeyse tamamıyla silikattan (asit silisit tuzu) yapılan camlar, gönderilen sinyali en düşük ölçüde çarpıtan optik liflerin üretilmesini mümkün kıldı.

Ama Japon ve Amerikan üreticiler arasında yarış sona ermiş değil. Şu anda bu yarış özellikle üç alanda yoğunlaşmış durumda.

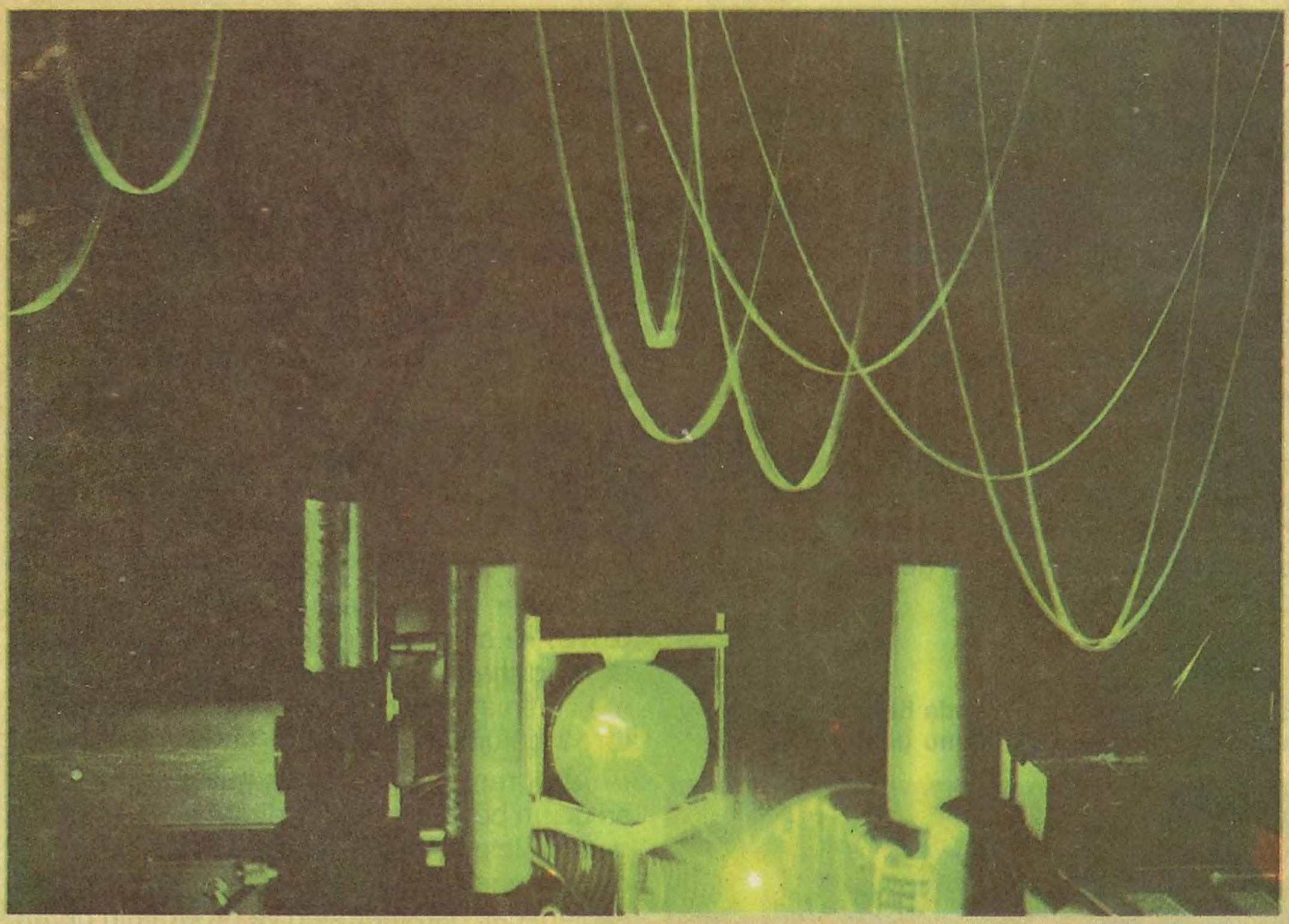
— 1. Daha hızlı, yani bir saniyede daha çok sayıda dalga içeren lazer ışınları üretebilmek. Başta da değindiğimiz gibi bu alanda rekor şimdilik saniyede 2 milyar dalga ile Bell laboratuvarlarında.

— 2. Belirli bir fiber-optik hattında kullanılması gereken ve ''tekrarlayıcı'' adı verilen özel güçlendiricilerin sayısını azaltmak. Tekrarlayıcılar öteden beri telefon şirketlerinin başbelası olagelmişlerdir. Bakır telefon hatlarıyla iletilen sinyal çok kısa bir süre içerisinde tamamıyla bozularak anlamsız bir gürültüye dönüşür. Bu yüzden her 1.5-2 km'de bir, sinyali güçlendirerek tekrarlayan bir tekrarlayıcı konulması gerekir. Oysa fiber-optik hatlarda, daha şimdiden, 45 km'ye bir tekrarlayıcı yetiyor. Bell laboratuvarlarında, 150 km'ye bir tekrarlayıcı yerleştirilmesinin yetebileceği de deneysel olarak kanıtlanmış durumda.

— 3. Üreticiler arası yarışın yoğunlaştığı üçüncü alan ise tümleşik optik devre üretimi. Bugün kullanılan bilgisayarlı fiber-optik uygulamalarının hepsinde veriler önce bir elektrik sinyali haline getiriliyor ve bilgisayar tarafından bu halleriyle işleniyor. Daha sonra bu elektrik sinyallerinin iletilmek üzere ışık sinyallerine çevrilmesi gerekiyor. Bu ise zaman kaybına yol açıyor.

Oysa henüz denenmekte olan bazı ''ışık anahtarları'', bilgisayarların yarı iletken yongalarındaki transistörlerden 10.000 kere daha hızlı durum değiştirebiliyor. Ancak, yalnız bu tür transistör benzeri ışık anahtarlarından oluşacak bir tümleşik devre yapmak, sorun olmakta devam ediyor. Çünkü bu güne kadar üretilen tümleşik optik devreler, içlerinde sadece 10 adet ışık anahtarı barındırabilen, elektronik devrelerle kıyaslandığında oldukça kaba kalan örnekler. ''Ancak'' diyor GTE Laboratuvarları araştırma yöneticisi Paul E. Ritt, ''bir iki 10'luk daha eklesek 100'e ulaşmamız zor olmaz. Çığ koptu artık bir kere...''

Öte yandan Japonlar, içinde hem optik hem elektronik ögelerin bir arada bulunduğu tümleşik devreleri yeğ tutuyorlar. Böylelikle elektrik sinyali halindeki verileri işleyen bir devre, aynı zamanda, başka bir devreye gerek duymaksızın, elektrik sinyalini optik bir sinyale çe-



Yeşil lazer ışınında parıldayan optik lifler.

virebilecek. Japonlar bu tür devreleri, gelecek için tasarladıkları ekranlı telefon, çift yönlü televizyon gibi olanaklarla donanmış iletişim ağının temel taşı olarak görüyorlar.

Kuşkusuz fiber-optik teknolojisi ile ilgilenenler yalnızca Japon ve Amerikan şirketleri değil. Amerikan ordusu da bilgisayarların gelişmesinde yaptığı gibi fiber-optik teknolojisine büyük yatırımlar yapıyor. Bu günlerde özellikle donanma fiber-optik sayesinde su altında radarın ve akustik duyargaların yerini alabilecek bir sistemle ilgileniyor. Böyle bir sistemin temel unsurlarını üretmek için 3 milyon dolarlık bir ihaleyi McDonnel Dougias şirketi kazanmış bile.

Bu sistemde deniz dibine 100 km'lik aralarla yerleştirilmiş fiber-optik hatlardan oluşan bir ağ döşenecek. Bu ağ ise karadaki bilgisayarlara bağlanacak. Aşırı duyarlı bir duyarga işlevi gören fiber-optik hatlar, sürekli olarak, ışık dalgası sinyalini ikiye bölerek birini deniz dibindeki hattan, ötekini ise karşılaştırma hattı adı verilen ikinci bir hattan karaya yollayacak. Normal koşullarda karada karşılaştırılan iki sinyalin birbirine tıpatıp uyması gerekiyor. Su altında bir denizaltı bulunduğu takdirde deniz dibinden iletilen sinyalin yerine ulaşması, çok kısa bir süre de olsa, gecikecek. Bu ise, karada bilgisayarın yapacağı karşılaştırmada derhal ortaya çıkacaktır. Böylelikle de su altındaki denizaltının varlığı saptanmış olacak.

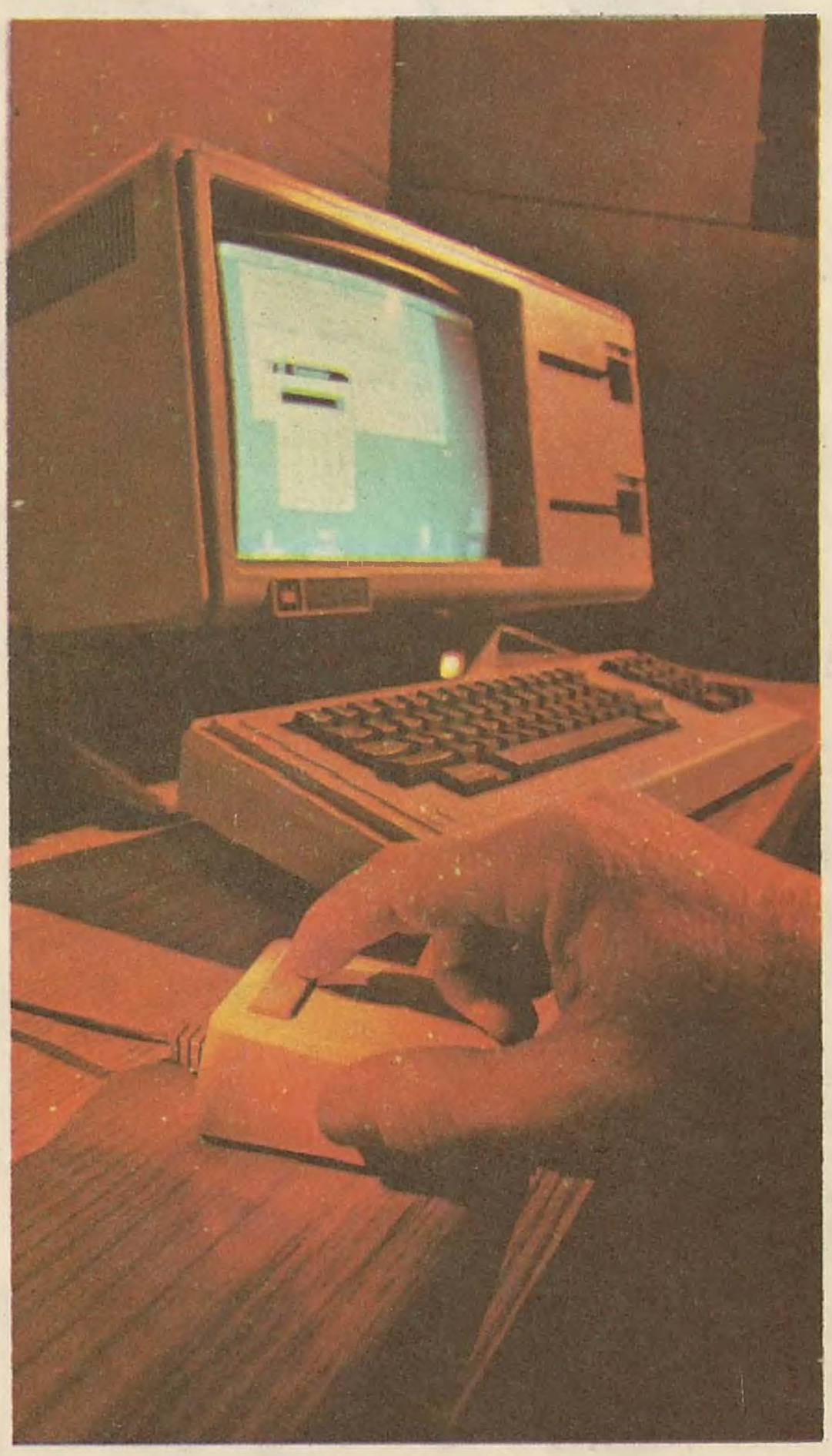
Kuşkusuz, aynı sistem askerî olmayan amaçlarla da kullanılabilir. Örneğin birçoğumuzun, telekomünikasyon teknolojisinin en ileri aşaması olarak düşündüğümüz ''uydular aracılığıyla iletişim'' bile fiber-optik teknolojisi tarafından eskitilmek üzere. Çünkü uydular, ancak çok uzun (1500 km.yi aşan) mesafeler arasındaki iletişimde kullanıldıklarında kârlı oluyor. Oysa fiber-optik teknolojisinin böyle bir kısıtlılığı yok; aynı yatırım hem kısa hem uzun mesafeler arası iletişim sorununu çözüyor.

(International Business Week, 21 Mayıs 1984 ''Fiber-optics: The Big Move in Commanications'' ve McGraw Hill Encyclopedia of Computers and Electronics'den derlenmiştir.)

Çözümleme sonucunda ulaştığımız işlevler sırasının akış şeması, birinci düzeyden bir akış şeması oluşturur. Sayfa 197'de bu sorunun birinci düzeyden akış şeması gösterilmiştir. Her simgenin yanına sayısal ''etiketler'' koymak, en azından başlangıçta faydalı olacaktır: Bu hem şemanın okunmasını kolaylaştırır, hem de programı daha çabuk yazabilme olanağı sağlar.

Sayfa 197'deki akış şemasında çeşitli adımları ayırabiliriz: 1000'den küçük sayılarla işaretlenmiş olan kutular ana programın akışını, 1000'den 1500'e kadar numaralanmış kutular ise bir oyunu bitirip yenisini başlatmak için gereken işlemleri göstermektedir; 2000'den 2400'e kadar numaralanmış olan üçüncü sütun ise makinenin vermesi gereken cevapları içermektedir.

Bu kişisel bilgisayar ekranda henüz işlenmemiş adımların hangileri olduğunu gösteriyor.



Örneğin bu şekilde 600 numaralı kutu, bir karar (ya da sınama) işlemini göstermektedir. Bu durumda makine kullanıcının verdiği cevabın şu durumlara uyup uymadığını sınar. Toplam TAMAM ve YAKIN'ların sayısı 0'a mı eşit? Durum böyleyse oyuncu hiç puan toplayamamış demektir. Dikkati çekeceğimiz konu, ''TAMAM=0 AND YAKIN=0'' cümlesinin hem toplam TAMAM cevaplarının, hem de toplam YAKIN cevaplarının sayısının 0 olduğunu göstermek için kullanılmasıdır; yani AND işlemcisi ile gösterilen bir ''VE'' mantık işlemi sözkonusudur. Bu şekilde yazıldığında akış şeması bütün işlevleriyle bir programa benzemektedir. Ancak bu program henüz iskelet halinde olduğundan, oyunun kurallarını bilmeyen bir oyuncu tarafından kullanılamaz.

Bu yüzden akış şemasının A ile başlayıp SON **(END)** ile biten ana gövdesine girmeden önce, bilgisayarın oyuncuya (oyunun kurallarını bilmemesi halinde) kuralları anlatacağı bir anlatı bölümü gereklidir.

Bu program hangi amaçla kullanılacak olursa olsun böyle bir bilgilendirme işlemi her uygulama için vazgeçilmez bir adımdır. Programın kullanıcısının (hatta bizzat programcının) sağlanacak parametreleri ve programın bütün işlevlerini tam olarak hatırlaması çok zordur. Başlangıçtaki açıklayıcı anlatı bölümü, bellek tazelemeye ve programı kullanan kişinin işini kolaylaştırmaya yarar.

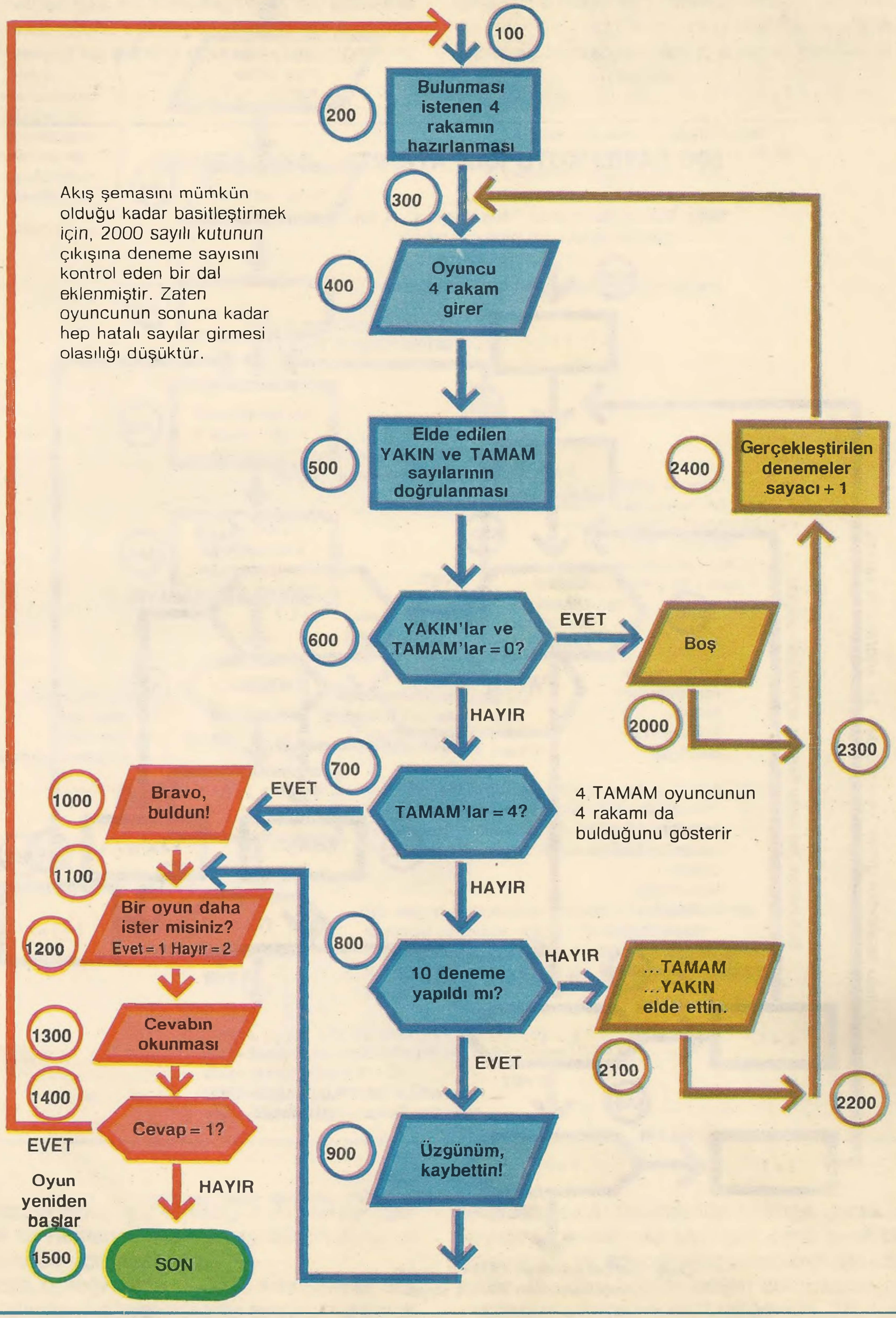
Sayfa 197'deki şema, karmaşık bir dizi işlemi gösteren iki öbek (200 ve 500) içermekle birlikte, bu işlemlerin nasıl gerçekleştiğini belirtmemektedir. Bu daha sonraki akış şemalarında açıklığa kavuşturulacaktır. Daha ayrıntılı açıklamalar sonraya bırakılarak, ana akış şemasının karmaşıklaşması önlenmektedir. Bu sayede daha karmaşık işlemler teker teker daha ayrıntılı bir biçimde geliştirilebilir. Akış şemasının böyle genel terimlerle yazılması, aynı şemayı değişik uygulamalarda kullanma olanağı sağlamaktadır. Bu teknik, programlamanın altın kuralını oluşturur.

Sayfa 198'deki şekilde programın özellikle toplam YAKIN ve TAMAM cevaplarının hesaplanacağı kısmının ayrıntılı akış şeması verilmiştir; bu kümede yer alacak bütün işlemlerin 500 ilâ 590 arasında numaralanması gerekir. (Bu sayıları belirleyen, sayfa 197'deki ilk akış şemasındaki 500 numaralı kutudur.) 600 numaralı kutu bir başka kümeye ait bir işleme işaret eder.

Artık bu programın akış şemasının tamamlanması için yapılması gereken tek şey, ilk şema-

METİNDEKİ OYUNUN BİRİNCİ DÜZEYDEN AKIŞ ŞEMASI

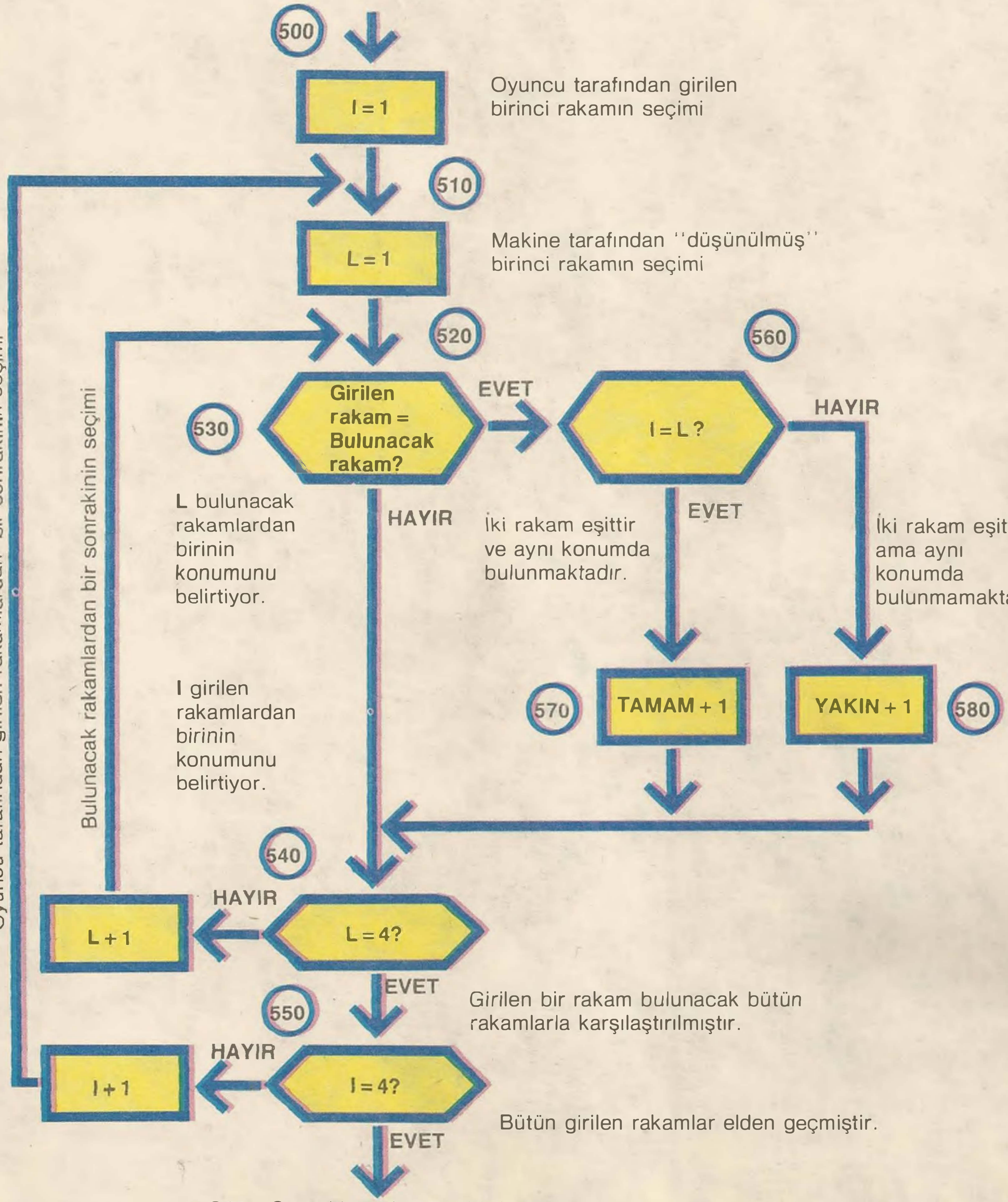
Akış şemasını mümkün olduğu kadar basitleştirmek için, 2000 sayılı kutunun çıkışına deneme sayısını kontrol eden bir dal eklenmiştir. Zaten oyuncunun sonuna kadar hep hatalı sayılar girmesi olasılığı düşüktür.



da 200 sayısı ile gösterilen kutunun ayrıntılı şemasını çizmektir. Buradaki işlemler oyunu oynayan kişinin bulması beklenen 4 rakamın belirlenmesi için gerekmektedir.
Bu birimin başlıca işlevi, rastlansal biçimde, yani birbirleriyle hiçbir ilişkisi olmayan sayılar sağlamaktır. Bu bölümde bilgisayara 0 ilâ 1 arasında yer alan rastlansal bir sayı sağlaması için bir komut verilebileceğini söylemekle yetineceğiz. Bu yolla elde edilmiş bir sayının as-

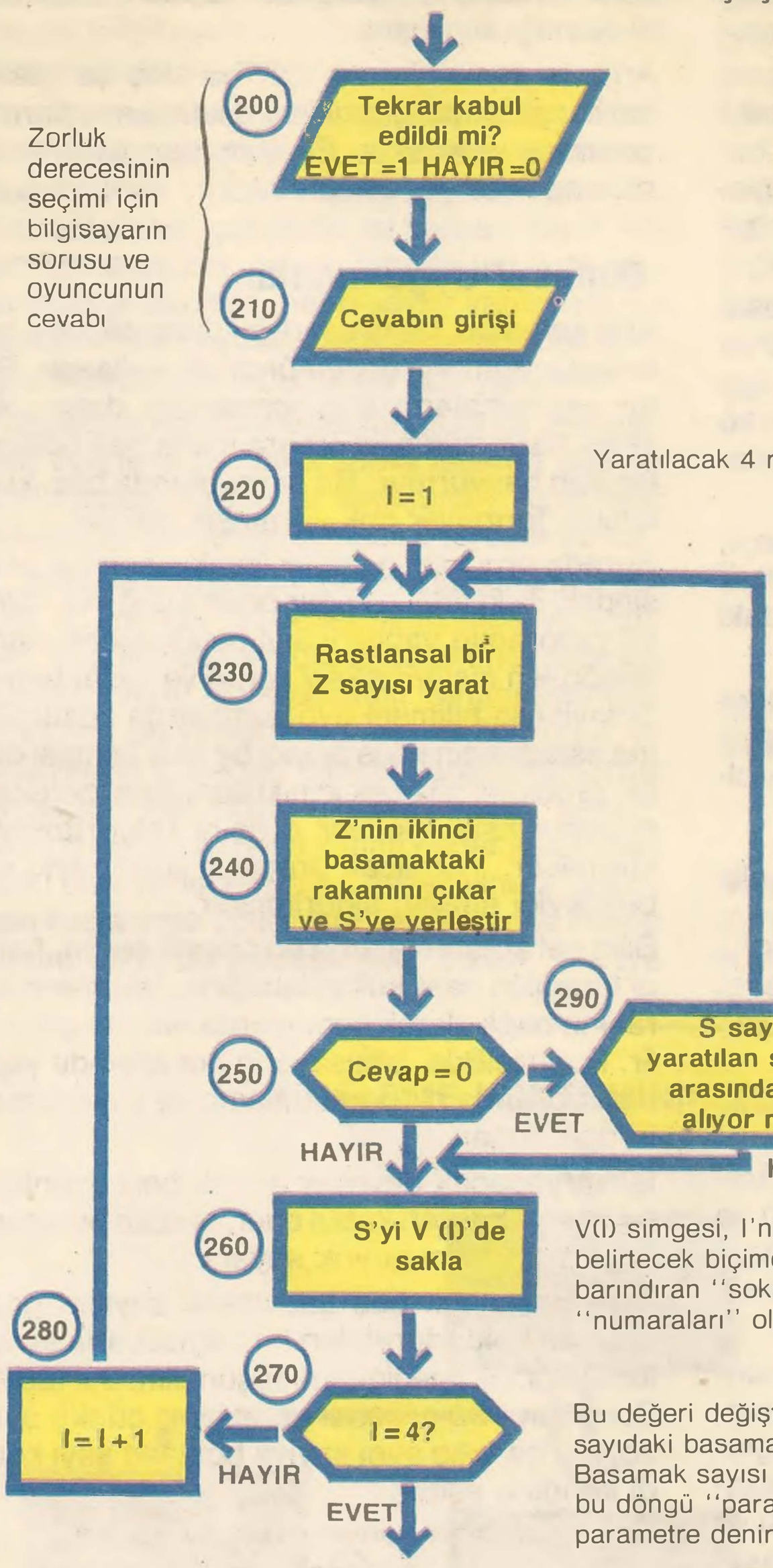
500 SAYILI KUTU İÇİN AYRINTILI AKIŞ ŞEMASI

Giriş: Makine tarafından ''düşünülmüş'' 4 rakam ve oyuncu tarafından girilen 4 rakam



Çıkış: Gerçekleştirilen TAMAM sayısı
Gerçekleştirilen YAKIN sayısı

200 SAYILI KUTUNUN AYRINTILI AKIŞ ŞEMASI



Zorluk derecesinin seçimi için bilgisayarın sorusu ve oyuncunun cevabı

240

C1 = (Z × 10)'un tümü
C2 = (Z × 100)'ün tümü
S = C2—C1 × 10

Örnek:
Z = 0,837915
C1 = 8
C2 = 83
S = 83—80 = 3

240 sayılı kutunun geliştirilmesi; burada rastlansal C sayısının ilk iki basamağı sayının geri kalan kısmından ayrılır.

Yaratılacak 4 rakamdan birincisinin seçimi.

EVET: Rakam daha önceki bir basamakta kullanıldığı için kullanılamıyor, dolayısıyla yeni bir sayı daha yaratmak gerekiyor. Oyuncu tekrarsız oyunu seçmiş (S kabul edilmiyor).

V(I) simgesi, I'nın değişmesine göre, birçok değeri belirtecek biçimde yazılmış; V bu değerleri barındıran ''sokak'', I da sokaktaki adres ''numaraları'' olarak düşünülebilir.

Bu değeri değiştirerek (örneğin I = 7) bulunacak sayıdaki basamak miktarı da değiştirilebilir. Basamak sayısı bu noktada tanımlandığına göre bu döngü ''parametreli''dir. Bu değişken değere parametre denir.

Çıkışta şunlar görülecektir:
V(1) = Birinci çevrimde elde edilen, birinci rakam (I = 1)
V(2) = İkinci rakam (I = 2)
V(3) = Üçüncü rakam (I = 3)
V(4) = Dördüncü rakam (I = 4)

lında ne olduğunu ve hangi bakımlardan gerçek bir rastlansal sayıdan ayrıldığını ilerki bölümlerde açıklayacağız.

Bizim örneğimizde 200'ler kümesinde dört çevrimde dört basamaklı bir sayı elde edilmektedir. Bu sayı, oyuncunun bulması gereken sayıdır. 4 basamaklı sayı çok daha basit bir yoldan, tek bir adımda elde edilebilirdi. Bu takdirde, ilk adımda elde edilen çok basamaklı rastlansal sayının ilk dört rakamının, ''bulun-

ması istenen sayı'' olarak kabul edilmesi gerekirdi. Oysa biz örneğimizde daha dolambaçlı bir yol izleyerek her basamak için ayrı bir çevrim kullandık.

Aslında bu yol, göründüğü kadar dolambaçlı değildir. Biraz daha dikkatli bakıldığında, bulunması istenen sayıda tekrarlanan rakamların olduğu durumlarda bu yöntemin daha basit olduğu görülür.

Bilgisayardan bize 0 ile 1 arasında 6 basamaklı bir sayı vermesini istediğimizi düşünelim. Bu basamaklardan yalnızca biri bizi ilgilendirdiğinden, bilgisayarın verdiği sayıyı 10 ile çarpıp virgülün sağında kalan basamakları atmamız gerekir.

Bilgisayarın bize verdiği 6 basamaklı sayı, 0,837915 olsun. Sayıyı 10 ile çarparak 8,37915 elde ediyoruz. Virgülün sağındaki basamakları atınca elimizde 8 kalıyor.

Şimdi ikinci basamağı da, yani 3'ü de sonunda elde etmek istediğimiz 4 basamaklı sayıya dahil etmek istiyorsak, yapmak zorunda olduklarımızı gözden geçirelim.

1 / Yukarıda yaptığımız gibi ilk basamağı alıp 8 elde ediyoruz.

2 / Şimdi bilgisayarın verdiği sayıyı 100'le çarpmamız (0,837915 × 100 = 83,7915) ve virgülün sağındaki basamakları atmamız gerekiyor. (83)

3 / Birinci adımda elde ettiğimiz sayıyı 10 ile çarpıyoruz. (8 × 10 = 80)

4 / 2. ve 3. adımlarda elde edilen sayılar arasındaki farkı buluyoruz. (83—80 = 3) ve bu sonucu bilgisayarın sağladığı rastlansal sayının ikinci basamağı olarak kabul ediyoruz.

Bu adımları atarak bulduğumuz sayı, sonunda ''bulunmasını istediğimiz sayı'' adıyla saptamaya çalıştığımız sayının dört basamağından yalnız biri. Üçüncü ve dördüncü basamaklar için de bu işlemlerin tekrarlanması ve her sonucun belleğe istenen sayının basamaklarından biri olarak yerleştirilmesi gerekiyor. Bu noktada, oyunda istenen güçlük derecesinin belirlenmesi de gerekli.

Bunun için dört basamaktaki değerlerden birinin başka bir basamakta da tekrarlanıp tekrarlanmayacağına karar verilmesi gerek. Bu soru, oyunun başında oyuncuya sorulacak ve kararı oyuncunun kendisi verecektir.

Sayfa 199'daki şekil, bulunması istenen sayıyı belirleyecek olan 200 sayılı birimin ayrıntılı akış şemasını gösteriyor. Burada 4 basamaktan herhangi ikisinin aynı değeri tekrarlamasına izin verilmiyor ve dolayısıyla her basamağın ayrı bir rastlansal sayıdan üretilmesi olanağı tanınıyor.

Artık işi tamamlamak için bu akış şemasını herhangi bir programlama dilinin komutlarına çevirmek yetecektir. Bu komutları daha ilerki bölümlerde göreceğiz.

Bilimsel Uygulamalar

Akış şemaları, daha sınırlı biçimde de olsa, bilimsel sorunların çözümünde de kullanılır. Bu tür uygulamalarda akış şemalarına, daha çok, zaten hazır olan bazı programların belli bölümleri için başvurulur. Bu sınırlı alanda bile, kullanılan formüller çok karmaşık olabilir.

Burada programın yapısı ile akış şeması arasındaki ilişki artık o kadar önemli değildir, çünkü programın yapısını asıl belirleyecek olan, öngörülen matematiksel sorun ve hesaplardır. Dolayısıyla bilimsel uygulamalarda çözümleme aşamasının esas amacı bir akış şeması değil, problemi çözmek için kullanılması gereken matematiksel işlemler dizisini **(algoritmayı)** kurmaktır. Yazılacak program üzerindeki en belirleyici etmen, algoritmadır.

Bilimsel araştırmalarda en önemli sorun, hangi formülün nasıl kullanılacağıdır. İşlemlerin sırasına bağlı olarak sonuçlarda hatalar çıkabilir. Bu, özellikle, bilgisayarın her işlemde yapmak zorunda olduğu tamlama ve yuvarlamalardan doğar.

İşlem yapacak devreler ancak belli uzunluktaki değerleri veri kabul eder, verinin bu uzunluğu aşan kısmını yok sayar.

Örneğin ancak beş basamaklı sayıları gözönünde bulundurabilen bir bilgisayarla şu işlemi yapmak istediğimizi düşünelim: 3 × 100/3. Sonucun 100 çıkması gerekiyor, çünkü aynı sayıyla çarpılıp aynı sayıya bölünen sayı kendi kendine eşittir.

Önce çarpımın sonucunu elde edelim:

3 × 100 = 300

Bilgisayar bu sonucu doğru olarak kaydedebilir, çünkü sayı beş basamağı aşmamaktadır. Dolayısıyla sonucu 3'e bölebiliriz:

300 ÷ 3 = 100

Bu sayıyı da bilgisayar doğru olarak kaydedebilir. Ancak şimdi işlem sırasını tersine çevirerek, bölme ile başlayalım:

100 ÷ 3 = 33,333333...

Bu sayı beş basamağı aştığından ''taşan'' sayılar yok sayılır ve sonuç 33,333 kaydedilir, ama bu, yanlış bir sonuçtur.

Elbette kabul edeceği basamak sayısı bu kadar sınırlı olan bir bilgisayar yoktur; normal olarak bilgisayarlar çok daha yüksek sayıda basamakları kaydedebilir.

Yine de çoğu bilimsel uygulamalarda çözümleme aşamasında yuvarlama ve tamlamadan doğabilecek hataların öngörülmesi gerekir. Normal olarak, çok dakik ve hassas büyük bilgisayarlarda bile, böyle hatalar üst üste tekrarlandığı takdirde, hatırı sayılır yanlışlara yol açılabilir. Bu olasılık, özellikle, yinelemeli yöntem sözkonusu olduğunda çok yüksektir.

Yinelemeli yöntemde başlangıçta bir bilinmeyen belirlenir ve daha sonra istenen işlemin yürütülmesi sırasında değiştirilir. Aşağıdaki şekilde bu yöntemin şemasını görüyorsunuz.

1. adımda rasgele bir değer, sözgelimi 0 seçilir. 2. adımda, bu değer bir formüle yerleştirilir. 3. adım ise formülün çözümünün istenen sonucu verip vermediğinin sınanmasından ibarettir. Bu örnekte sonuç = 0 doğrudur ve dolayısıyla 0 değeri bilinmeyenin aradığımız değeridir. Hal böyle olmasaydı bir ''kalan'' (sonuç değeri ile hesaplanan değer arasındaki fark) elde edilecek ve bu değer yine 2. adımdan başlanmak üzere formüldeki yerine konup sınanacaktı. Bu yaklaştırma, ''kalan'' 0'a (ya da 0'a çok yakın bir değere) varıncaya kadar yinelenecektir.

Şimdi yinelemeli yöntemin bir tartı mekanizmasına nasıl uygulanacağını görelim. Ağırlığını bilmediğimiz bir çanak içersinde 10 adet top var; ancak topların da ağırlığını bilmiyoruz. Çanağı tartıya koyduğumuzda toplam ağırlığın 24 kg olduğunu görüyoruz. Çanaktan 5 top çıkardığımızda bu kez toplam ağırlığın 14 kg olduğunu görüyoruz. Şimdi çanağın ve tek bir topun ağırlıklarını hesaplamaya çalışalım.

Çanağın ağırlığını **x**, topun ağırlığını **y** ile gösterirsek ilk tartıştan sonra şunu yazabiliriz:

$$\text{Çanak} + 10 \text{ külçe} = 24 \text{ kg, yani}$$
$$\mathbf{x} + 10\mathbf{y} = 24$$

İkinci tartıştan sonra ise

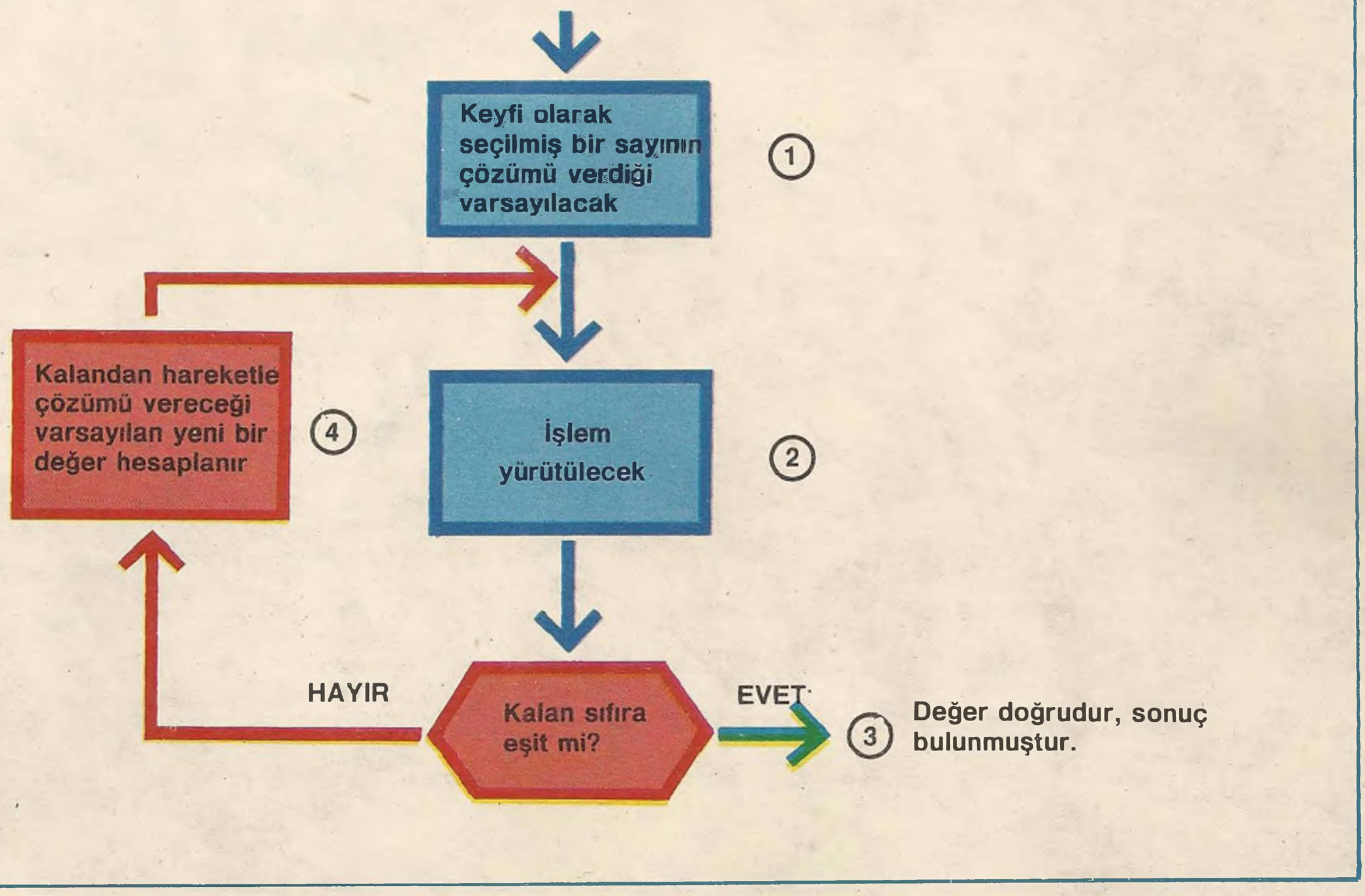
$$\text{Çanak} + 5 \text{ külçe} = 14 \text{ kg, yani}$$
$$\mathbf{x} + 5\mathbf{y} = 14$$

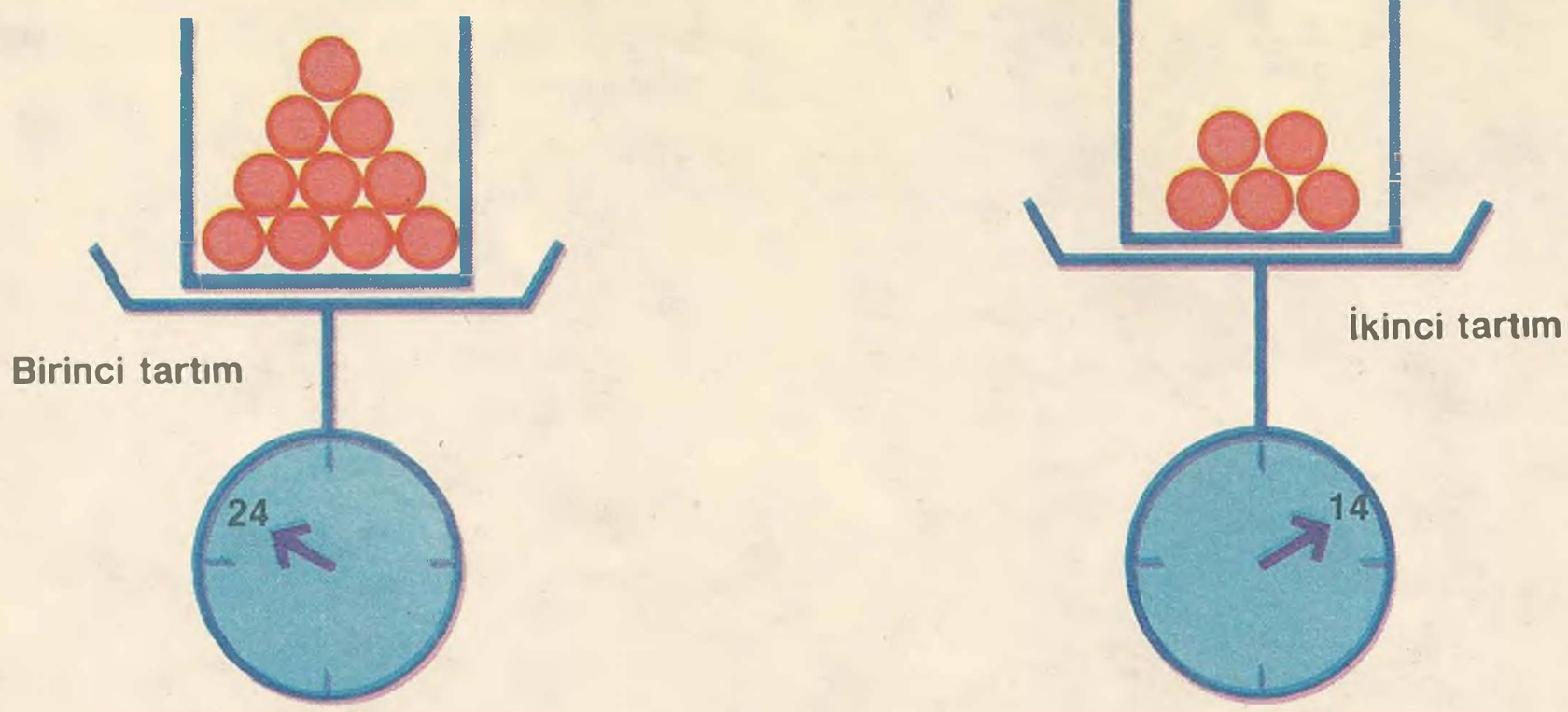
Sonuç bu iki cebirsel denklemin birleştirilmesiyle bulunur:

$$\mathbf{x} + 10\mathbf{y} = 24$$
$$\mathbf{x} + 5\mathbf{y} = 14$$

Sonuç çok basit bir cebirsel işlemler dizisiyle (örneğin iki denklem arasındaki farklardan yola çıkarak) bulunur. Bu yolla $\mathbf{x} = 4$ ve $\mathbf{y} = 2$ de-

YİNELEME YÖNTEMİNİN TEMEL ŞEMASI





ğerlerini elde ediyoruz. Şimdi aynı sonucu bir yinelemeli yöntem aracılığıyla bulmaya çalışalım:

1 / Bir bilinmeyene herhangi bir değer verilebilir, örneğin $\mathbf{x} = 0$.

2 / Bu değer birinci denklemde yerine yerleştirilir:

örneğin $\mathbf{x} + 10\mathbf{y} = 24$

Bundan;

$10\mathbf{y} = 24 \div 10 = 2{,}4$

sonucunu elde ederiz.

Yinelemeli yöntemde bir hata yapıldığı takdirde, kelimenin gerçek anlamında bir ''kağıt denizi''nde boğulmak işten değildir.

3 / Bu yolla **y** için elde edilmiş değer (**y** = 2,4) ikinci denklemde yerine yerleştirilir ve bu kez **x**'in değeri hesaplanır:

$$\mathbf{x} + (5 \times 2{,}4) = 14$$

Bundan;

$$\mathbf{x} + 12 = 14$$

$$\mathbf{x} = 14 - 12 = 2$$

sonucunu elde ederiz.

4 / Bu değer yine (1. adımda yapıldığı gibi) birinci denklemdeki yerine yerleştirilir.

$$2 + 10\mathbf{y} = 24$$

$$10\mathbf{y} = 24 - 2$$

$$10\mathbf{y} = 22$$

$$\mathbf{y} = 22 \div 10 = 2{,}2$$

Bu adımların tekrarı aşağıdaki şekilde görülen değerleri verecektir.

$\mathbf{x} + 10\mathbf{y} = 24$ ve $\mathbf{x} + 5\mathbf{y} = 14$ denklemlerinin her ikisi ancak $\mathbf{x} = 4$ ve $\mathbf{y} = 2$ değerlerini aldıklarında 0 sonucunu vermektedir. Dolayısıyla bu iki değer istenen çözümlerdir.

Ancak adımları tersine çevirip önce 5 topu, sonra 10 topu tartsaydık elde ettiğimiz değerler, örneğimizde olduğu gibi, gitgide sonuca yaklaşmayacak, aksine ''ayrılacaktı'', yani giderek istenen sonuçtan uzaklaşan değerler alacaktı. Dolayısıyla böyle bir yinelemeli süreç kullanıldığı zaman ayrılan değerler elde edilmesini önlemek için, işlemlerin sırasının çok doğru ve açık olarak öngörülmesi gerekir.

Gerçi örneğimiz biraz abes görünüyor, çünkü bu sonucu zaten cebirsel yollarla elde edebilirdik, dolayısıyla yinelemeli yönteme gerçekte ihtiyacımız yoktu. Ama bilimsel uygulamalarda problemlerin çözümü bizim örneğimizin çözümü kadar basit değildir. Çözülmesi güç problemlerde yinelemeli yöntem çok yararlı olduğundan, ayrılan sonuçların nasıl önlenebileceği konusunda birbirinden çok farklı, fakat hepsi de aynı derecede karmaşık birtakım kuramlar geliştirilmiştir. Bunu izleyen bölümde, yinelemeli bir yöntem kullanarak bir ısı probleminin nasıl çözülebileceğini inceleyeceğiz.

Yinelemeli Yöntemin Uygulama Alanları

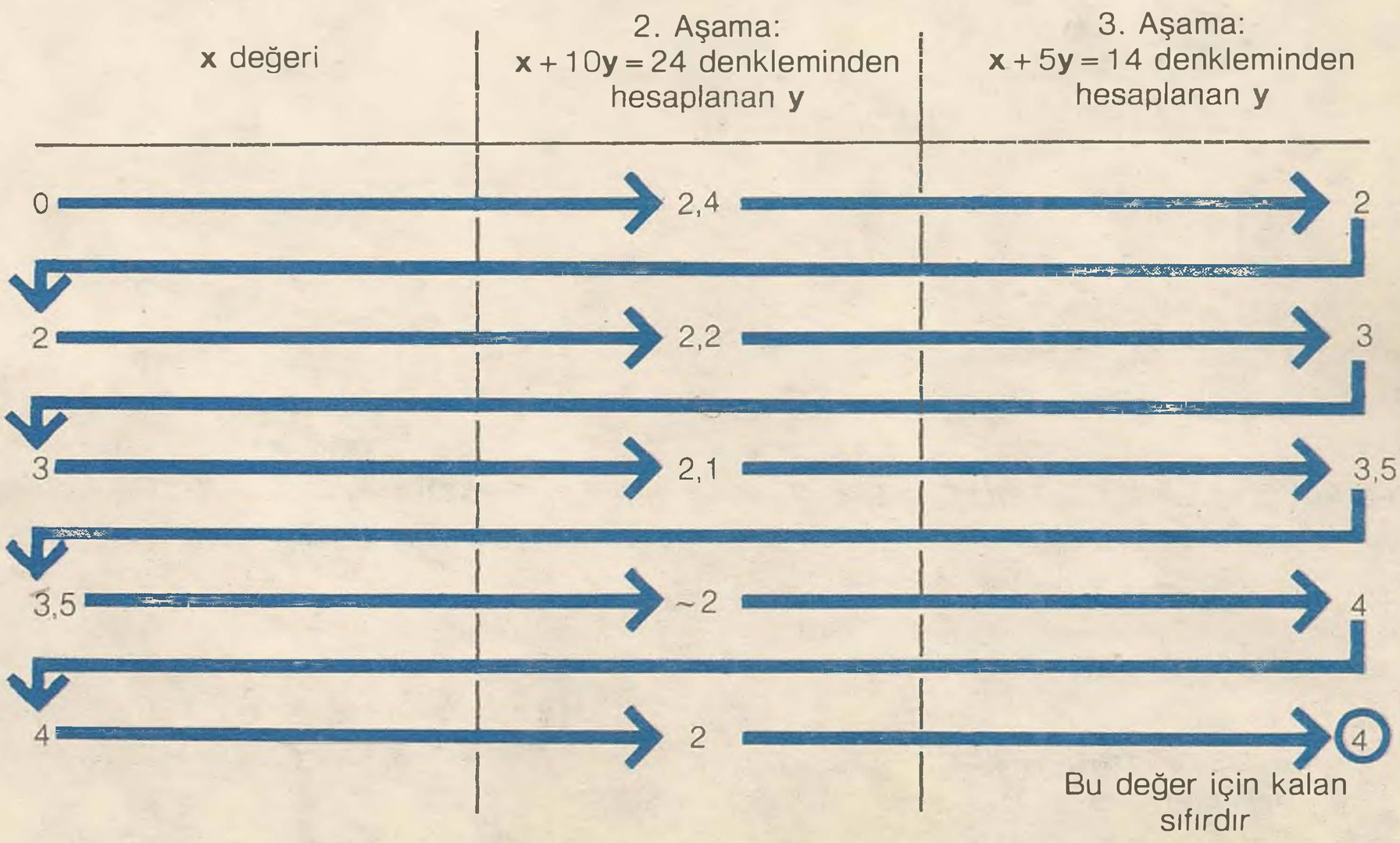
Bir önceki örneği (iki bilinmeyenli iki denklemin çözümü) açıklayıcı olsun diye seçtik. Cebirsel bir çözümün mümkün olduğu bu problemde yinelemeli yönteme başvurulması gerekmiyordu.

Bu türden birinci dereceden (yani bilinmeyenlerin kuvvetlerinin 1 olduğu) denklemler ne kadar karmaşık olursa olsun, bilinmeyenlerin sayısı denklemlerin sayısına eşit olduğu sürece, cebirsel yollardan çözülebilir, dolayısıyla yinelemeli yönteme başvurmayı gerektirmezler.

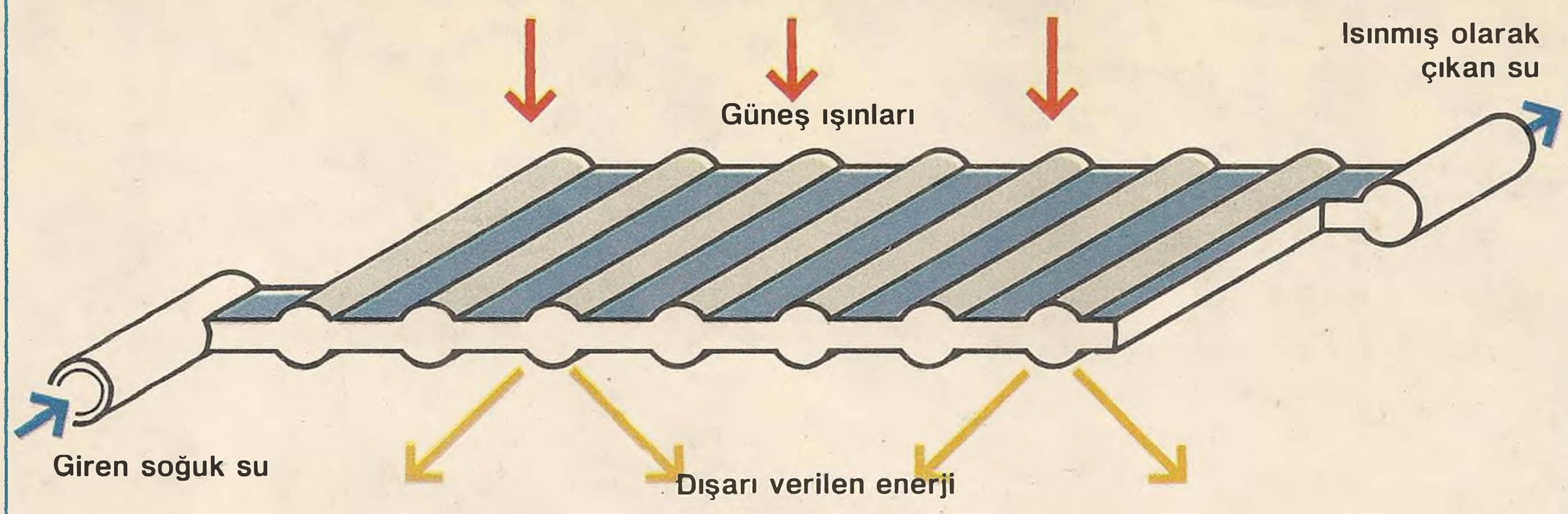
Ancak cebirsel bir çözümü olmayan denklem türleri de vardır. Bu denklemlerde yinelemeli yöntem, kullanılabilecek tek yöntemdir. Bu türden tipik bir denklem şudur:

$$A \times T^4 + B \times T = C$$

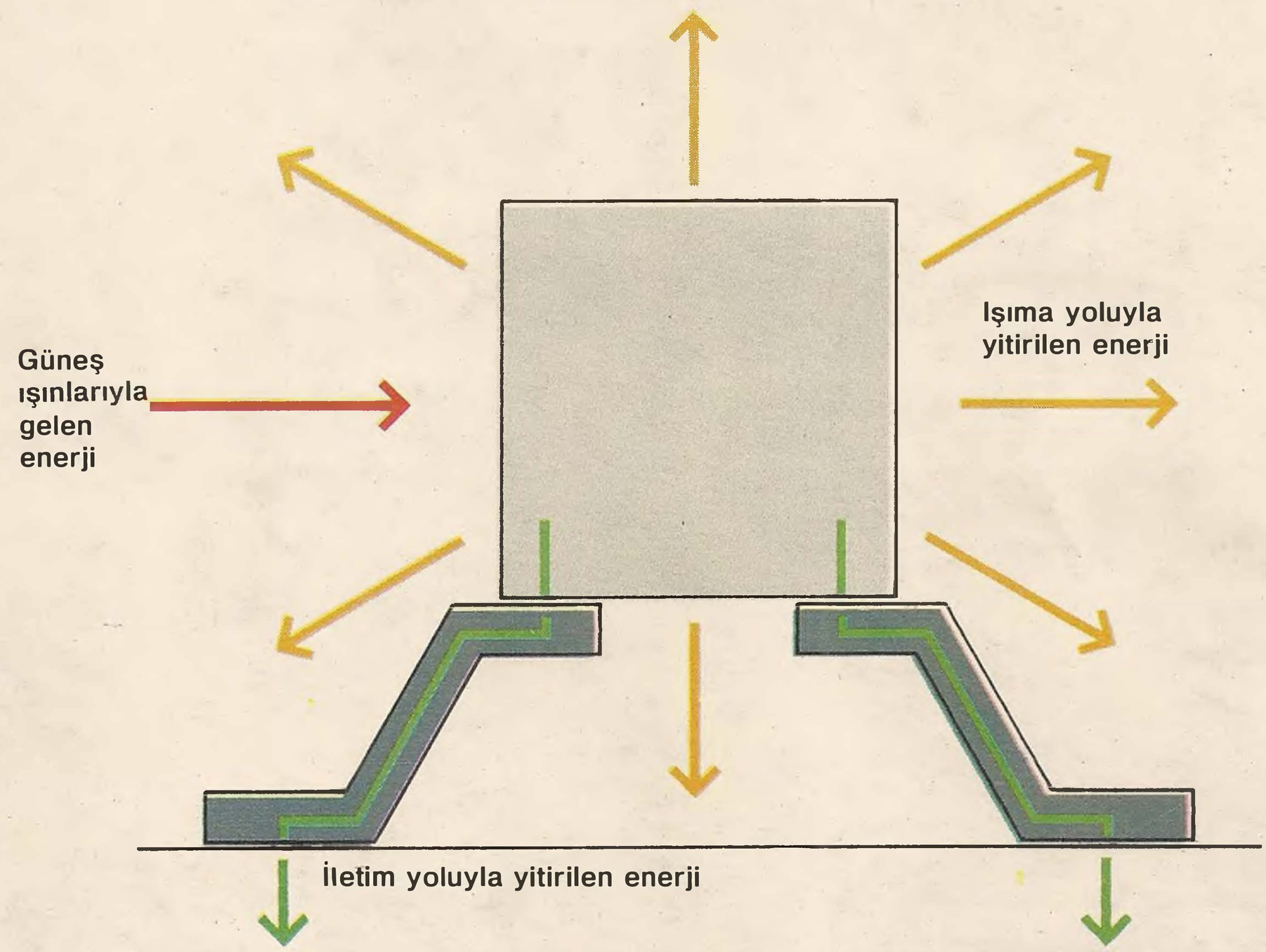
Burada, bilinmeyen, T'dir. A, B ve C verilmiş sabit değerlerdir. Bu denklemin nasıl çözüle-



BİR GÜNEŞ LEVHASININ ŞEMASI



FİZİK OLAYININ ŞEMASI



ceğini incelemeden cebirsel bir çözümün neden olanaklı olmadığını görelim.

Endüstri toplumlarının karşı karşıya kaldıkları enerji sorununa hepimiz aşinayız. Kömür gibi geleneksel enerji kaynakları gitgide tükenmekte ve dolayısıyla pahalılanmakta, üstüne üstlük, çevre kirliliğine yol açmaktadır. Bundan ötürü, enerji kaynaklarının aranması zorunlu hale gelmektedir. Bu kaynaklardan biri, belki de en önemlisi, güneş enerjisidir. Hepimiz bir su kovasının güneşte bırakıldığı takdirde ısındığını biliriz. Bu, güneş ışınlarındaki enerji suya geçtiği için olmaktadır. Su ısınınca, bu enerjiyi değişik yerlerde ve özellikle evin çeşitli ihtiyaçlarını karşılamak için kullanabiliriz. Bugün, evlerde, gerek merkezi ısıtma için, gerek sıcak su elde etmek için gereken güneş enerjisini özel güneş levhaları aracılığıyla toplamak mümkün. Bu uygulama birçok ülkede ticari bir nitelik bile kazanmış durumda. Ancak bu levhalardan elden geldiğince yararlanabilmek istiyorsak, bunların nasıl yerleştirileceklerini çok iyi hesaplamamız gerekir. İşte yukarıda verdiğimiz formül bu hesaplarda kullanılmaktadır.

Burada sözkonusu olan fizik olayı, bu sayfa-

da görüldüğü gibi şemalaştırılabilir. Şeklin yukarısında bir güneş levhasının işleyiş şeması görülüyor. Aşağıdaki şema ise gerçek fizik olayını gösteriyor. Hareketsiz duran bir su kabını ele alalım. Kaptaki (hareketsiz) suyu belli bir sıcaklıkta (T) tutmak istersek, kabın yitirdiği kadar enerjiyi suya vermek gerekir. Daha az enerji sağlarsak su soğur, daha fazla sağlarsak da gerektiğinden fazla ısınır. Sabit sıcaklıktaki hareketsiz su için ''durgun'' koşullardaki enerji dengesi şöyledir:

Alınan enerji = verilen enerji

Alınan enerji güneş ışınlarıyla taşınan ısı miktarıdır. Bunun değeri C ile gösterilir. Verilen enerji ise kabın çevresindeki ortama bıraktığı enerjidir (kayıp). Isı kaybının ilk nedeni, varolan enerjinin dışarıya iletilmesidir.

İletimden kaynaklanan ısı kayıpları daha sıcak bir kütleden onunla temas halindeki daha soğuk bir kütleye olan ısı iletimi miktarına bağlıdır (örneğin kap ile onu taşıyan kirişler arasında). Bunlar iki kütle arasındaki sıcaklık farkı ile orantılıdır:

İletimden kaynaklanan ısı kaybı =
B × (birinci kütlenin sıcaklığı—
ikinci kütlenin sıcaklığı)

Ancak örnek olsun diye anlattığımız bu problemi daha da basitleştirmek için ikinci kütlenin sıcaklığını gözönünde tutmuyoruz.

İletimden kaynaklanan ısı kaybı = B × T

B kullanılan malzemenin tipine ve sistemin geometrik biçimine bağlı olan bir parametredir.

İkinci kayıp türü olan ışınım, bir kütlenin ısısının onu çevreleyen (daha soğuk) ortama doğru elektromanyetik dalgalar halinde (radyasyon) iletimidir. Bu tip ısı iletiminde, iletilen enerji, sıcaklıkların dördüncü kuvvetleri arasındaki farka bağlıdır.

Işınımdan kaynaklanan ısı kaybı =

A × [(birinci kütlenin sıcaklığı)^4—
(ikinci kütlenin sıcaklığı)^4]

Bu durumda da, ikinci kütlenin sıcaklığını yi-

Güneş levhaları.

ne gözönünde tutmayarak denklemi şöyle yazıyoruz:

Işınımdan kaynaklanan ısı kaybı = $A \times T^4$

A parametresi kütlelerin fizik ve geometrik özelliklerine bağlıdır.

Toplam ısı kaybı bu iki kaybın toplamıdır (iletim + ışınım), yani:

Toplam ısı kaybı = $A \times T^4 + B \times T$

Sıcaklığı (T) sabit tutabilmek için, su kabına toplam kayba eşit miktarda enerji sağlamak gerekir. Diğer bir deyişle, güneş ışınlarıyla taşınan enerji (C), toplam kayıp enerjiye eşit olmalıdır:

$$C = A \times T^4 + B \times T$$

Elde edilen denklem basitleştirilmiş biçimdedir, çünkü kabın, çevresindeki hava ile temasından doğan ısı kayıpları (''konveksiyon'' tipi ısı kaybı) hesaba katılmamaktadır. Birçok durumda, T'nin değeri büyük değilse $A \times T^4$ terimi de gözönünde tutulmaksızın formül daha da basitleştirilebilir. Bu formülle, kabın özelliklerine bağlı olan sabitlerin değerleri bilindiği takdirde, suyun giren ve çıkan enerjilerin etkisi altındaki denge sıcaklığı (T) bulunabilir. Formülde sıcaklık (T) derecesi santigrat cinsinden değil, Kelvin cinsinden ifade edilmiştir (0 Kelvin, —273,2 derece santigrat karşılığıdır). Öyleyse Kelvin cinsinden bir sıcaklık, santigrat değerine 273,2 eklenerek bulunabilir. Görüldüğü gibi Kelvin derecelerinin kullanılması, çok büyük sayılar vermektedir. İşlemleri kolaylaştırmak için, genellikle bu değer 1000'e bölünür.

Böylece bütün işlemler küçük sayılarla yürütülmüş olur. Örneğin 373,2 K'nın 4 dördüncü kuvveti (T^4) 19398428000 sayısını vermektedir. Oysa sıcaklığı 1000'e bölmüş olsaydık (yani 0,3732) 0,0194'e yuvarlayabileceğimiz, 0,0193984 sayısını elde edecektik. İşlemin sonunda, tabiî, cevap 1000 ile çarpılacaktır. Bu yöntem, özellikle ısıl işlemlerde başvurulan bazı sabit değerler için de kullanılır.

Isıl denge formülünü yeniden yazalım:

C Güneş tarafından sağlanan ısı	=	$A \times T^4$ Işınım yoluyla kayıp	+	$B \times T$ İletim yoluyla kayıp

Denklem şu şekilde yeniden yazılabilir:

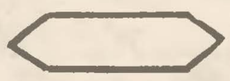
$$T = C — A \times T^4 \div B$$

Bu denklemin çözümü, yani kabın denge ısısının bulunması için şu yinelemeli süreç kullanılabilir:

1 / Denklemin sağ tarafındaki işlemleri yürütmek için, $A \times T^4$ terimine rasgele bir T değeri (örneğin T_1) değerini veririz. İşlem bitince denklemin sol tarafında T_2 adını vereceğimiz yeni bir T değeri elde ederiz:

$$T_2 = C — A \times T_1^4 \div B$$

TEST 5

1 /  simgesi şunu belirtir:
a - Bir kararı; b - Bir işlemin yürütülmesini; c - Programın sonunu.

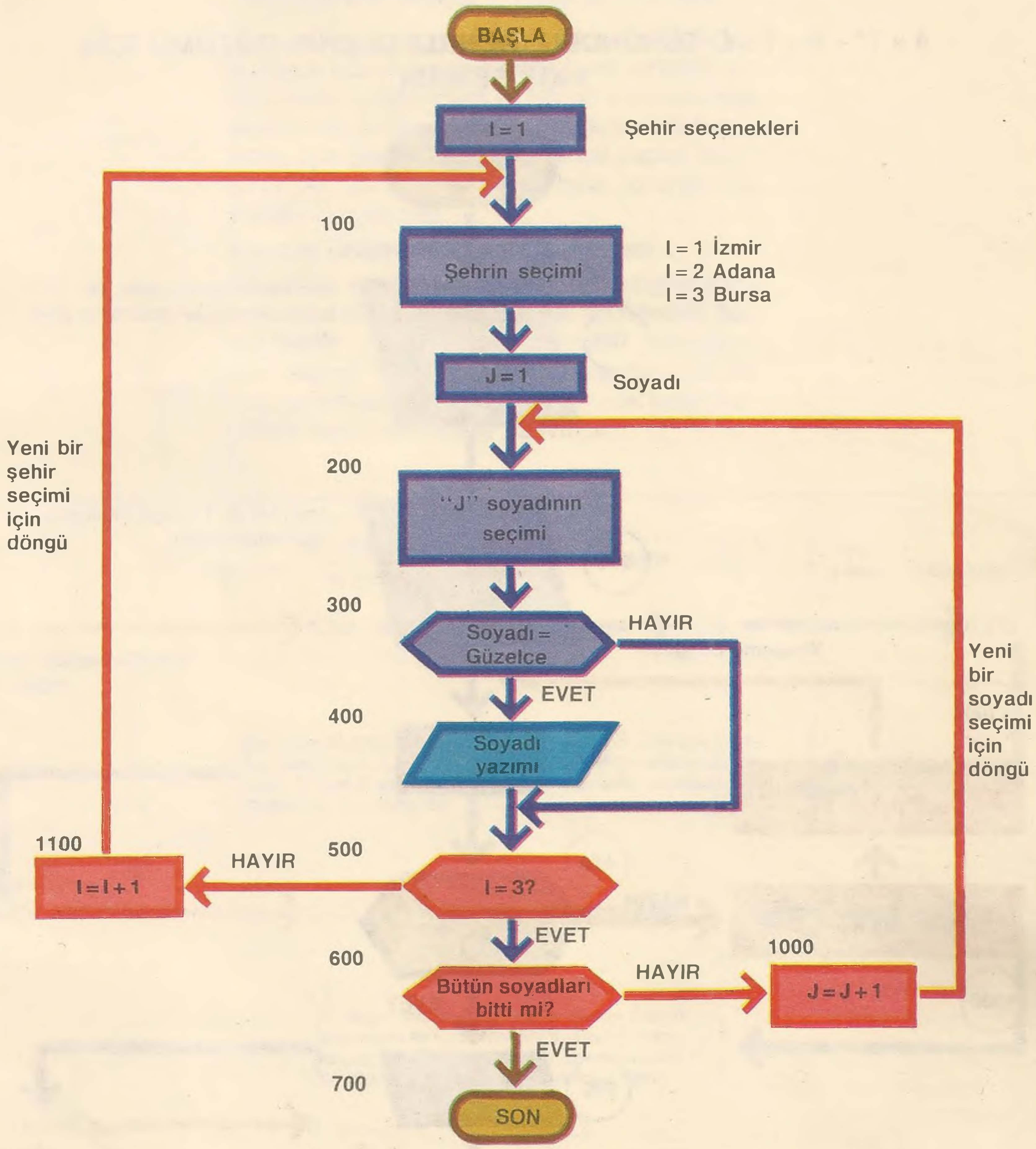
2 / Şu eylemleri gösteren simgeleri çiziniz:
a - Genel bilgi-işlem; b - Bir hesap işlemi; c - Bilgiyazım.

3 / Şu simge nerede kullanılır?

4 / Çentik (flag) nedir?
a - Bir belirteç; b - Bir sayaç; c - Bir merkez birim sicili.

5 / Yüzdeleri hesaplayan ve girişte yüzdelerin 100'den büyük olmadığını kontrol eden bir programın akış şemasını çiziniz.

6 / Aşağıdaki akış şeması bir hata içermektedir. Ne olduğunu bulun ve şemanın doğru biçimini çizin. Sorun şöyledir: Elimizde üç küme adres var. Birincisi İzmir, ikincisi Adana ve üçüncüsü de Bursa şehirleriyle ilgili. Her küme on adres içeriyor. Güzelce soyadlı bütün kişilerin adreslerini seçmek istiyoruz (iki döngü gerekmekte, birincisi şehri, ikincisi de soyadını seçmek için).



Cevaplar 214 ve 215. sayfalardadır.

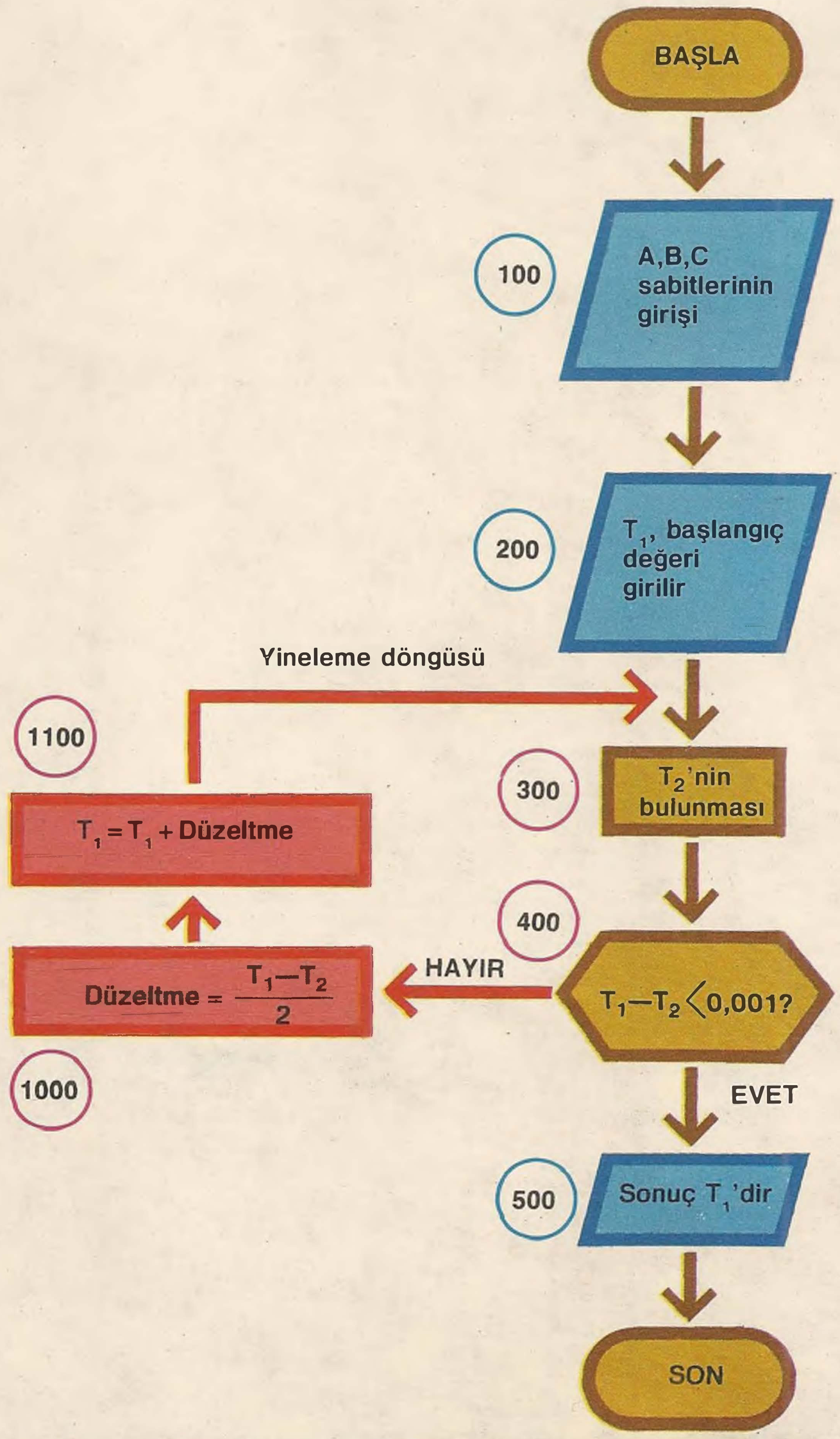
2 / T_1 ve T_2 değerleri arasındaki fark çok küçükse (örneğin 0,001) sonucunu bulduk demektir.

3 / Yoksa T için T_1 ve T_2 değerlerinin ortalaması olan yeni bir düzeltme değeri hesaplanır:

Düzeltme = $T_1 - T_2 \div 2$

4 / Bu düzeltme başlangıç (rastlansal) değeri

$A \times T^4 + B \times T = C$ TÜRÜNDEN DENKLEMLERİN ÇÖZÜMÜ İÇİN AKIŞ ŞEMASI



olan T_1'le toplanarak T_1 + düzeltme elde edilir.

(Yeni) T_1 = (Eski) T_1 + Düzeltme

Sayfa 208'deki akış şeması bu yöntemi göstermektedir.

Aşağıdaki tablo A = 300, B = 26,58, C = 10 değerleriyle işlemlerin nasıl geliştiğini göstermektedir. Bu değerlerle denklem şu biçimi alır:

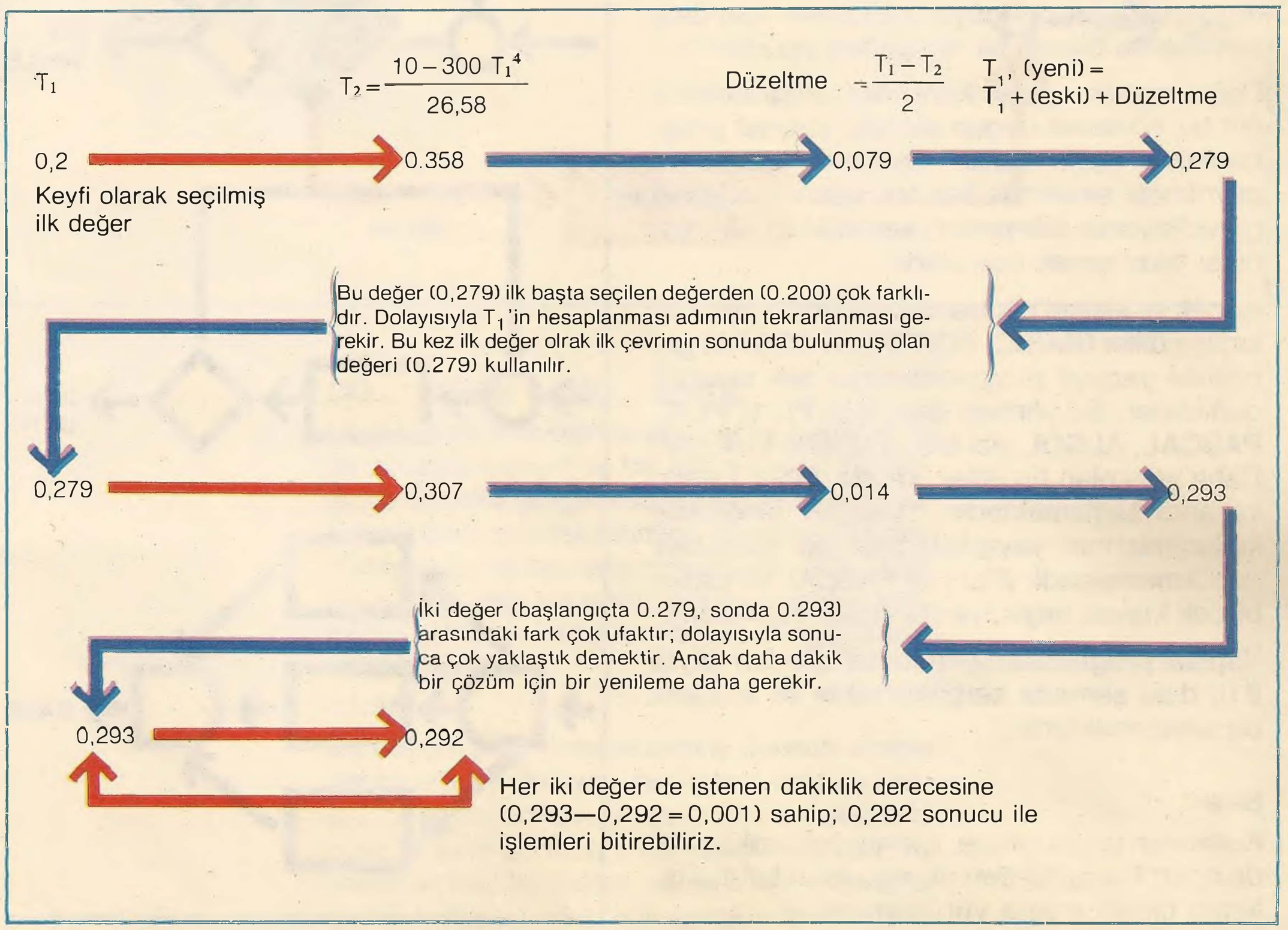
$$300 \times T^4 + 26{,}58 \times T = 10$$

Yapılandırılmış Programlama

Buraya kadar incelediğimiz işlemler, yani bir sorunun çözümlenerek daha basit birtakım alt sorunlara ayrıştırılması ve her alt sorunun akış şemasının çizilmesi, çoğu pratik sorunun çözümü için gerekli olan mantıksal yapıyı oluşturur. Ancak çok karmaşık bazı sorunlar bu yoldan çözülemez.

Yapısal programlamanın simgeleri

Küçük ve kişisel bilgisayarların programlanmasında HGSÇ ve akış şeması yöntemleri, büyük yarar sağlar. Daha karmaşık sistemler için, kişisel yoruma daha az yer bırakan ve ekip çalışmasına olanak verecek çok kesin kurallara bağlı tekniklere başvurulur.



Genellikle bir kişisel ya da küçük bilgisayar tek bir kişi tarafından programlanır. Dolayısıyla bu çalışma daha esnek olabilir. Ancak bilgisayarın verimi ve sorunların karmaşıklığı arttıkça, görev alan kişilerin sayısında da bir artma olacağından, açık bir döküm ve birörnek hale getirilmiş yöntemler kullanma gereksinimi de artar.

Ayrıca bu karmaşık sistemlerde, zamandan kazanmayı sağlayan, hata tehlikesini azaltan tekniklere de başvurmak gerekir. Bu yöntemlerin en modern ve gelişmiş olanı yapısal programlamadır.

Bu karmaşık yöntem, büyük bilgisayarlar ve küçük bilgisayarların gelişmiş tipleri ile yapılan uygulamalarda çok büyük değer taşır. Beş temel işlev üzerine kurulmuş olan bu programlama tekniğine daha sonra dönme olanağı bulacağız, ama şimdilik, bazı temel özelliklerine değinelim.

Akış şemasını oluşturan simgelerden her biri kendi başına bir temel işleme işaret ettiği için, akış şemaları genellikle çok sayıda böyle simge içerirler.

Yapısal programlamada ise aksine, yalın simgeler değil, karmaşık işlemleri belirten bileşik simgeler kullanılır. Örneğin kod değişmesi ile durdurulan bir döngüyü göstermek için akış şemasında bileşik bir simgeden yararlanılır.

Doğal olarak, ancak kullanılan programlama dili bu yönteme uygun olursa, yapısal programlama kullanılabilir. Şemanın komutlara çevrilmesi sırasında her tek birim ayrı işlem gerektiriyorsa bireşimsel (sentetik) bir akış şeması hazırlamak boşunadır.

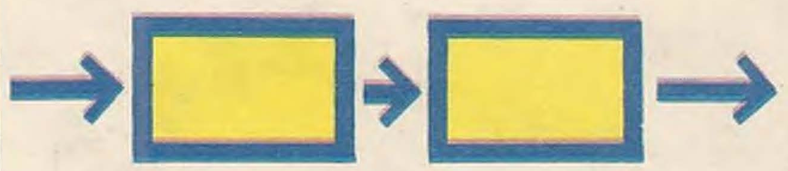
Küçük ve kişisel bilgisayarlarda daha çok kullanılan diller (BASIC, FORTRAN, COBOL) genellikle yapısal programlamaya pek elverişli değildirler. Bu yöntem daha çok **PL/1, PL/C, PASCAL, ALGOL** gibi başka dillerle kullanılır. Daha yeni olan bu diller, kayda değer zaman kazancı sağlamaktadır. Yakın bir gelecekte kullanımlarının yaygınlaşması da olanaksız gözükmemektedir (PL/1 ve PASCAL şimdiden birçok kişisel bilgisayarda kullanılmaktadır).

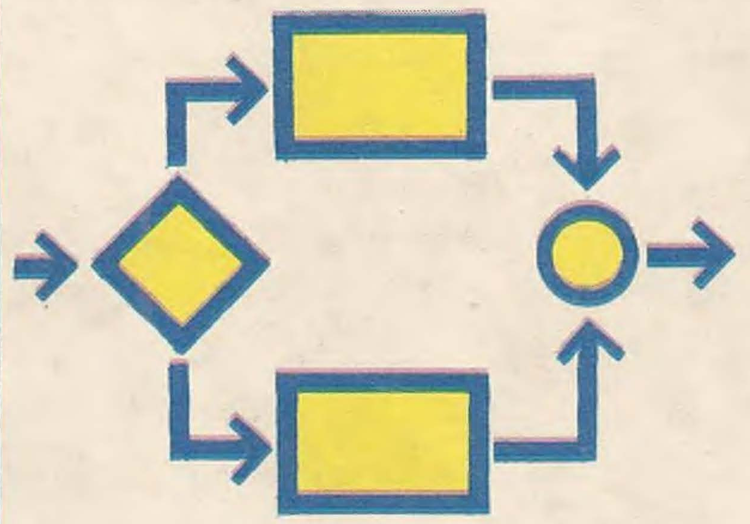
Yapısal programlamanın temel işlevleri sayfa 210'daki şemada sergilenmekte ve aşağıda da anlatılmaktadır.

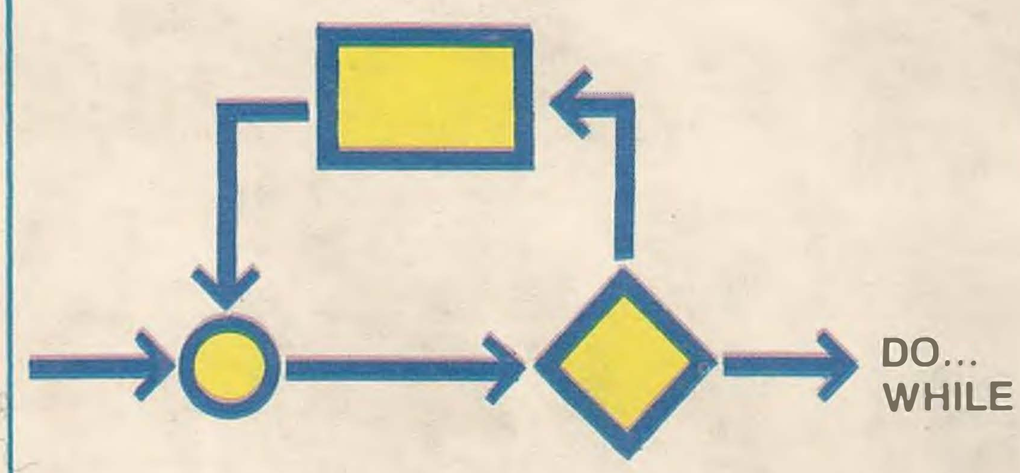
Sıra

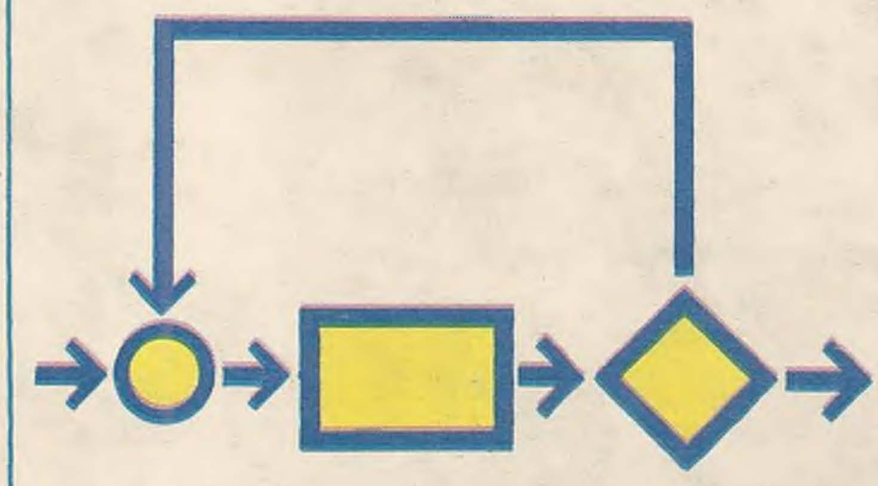
Kullanılan grafik simge, işlevini çok açıkça ifade eder: Hangi türden olursa olsun bir dizi işlemin birbiri ardına yürütülmesi.

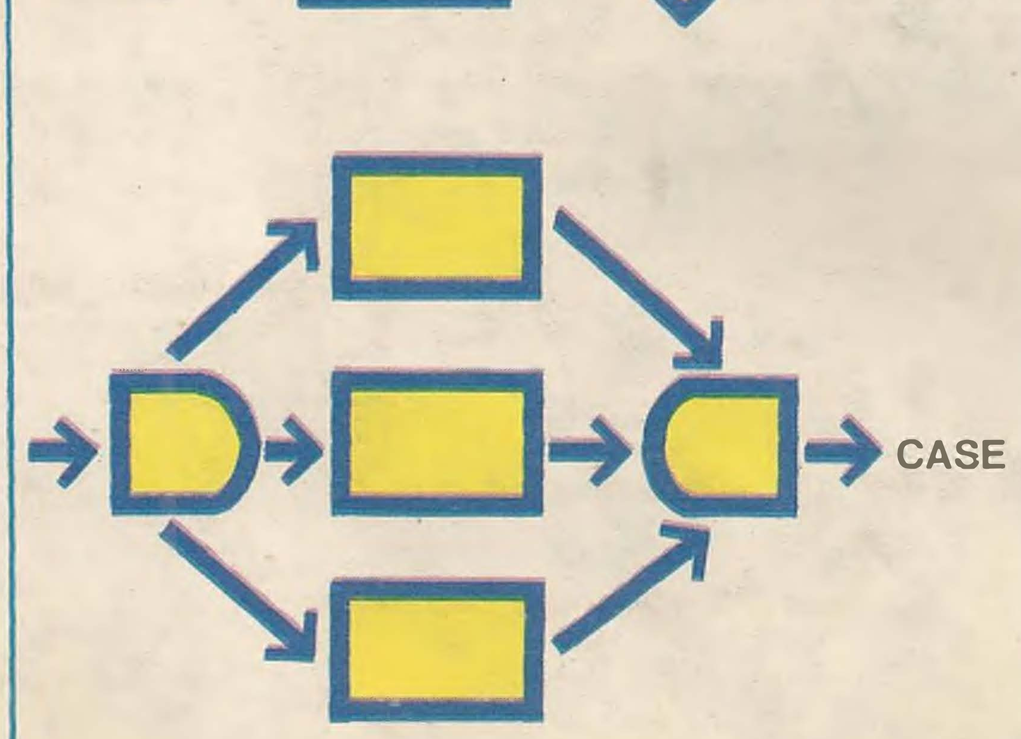
YAPISAL PROGRAMLAMADA KULLANILAN SİMGELER











IF... THEN... ELSE...

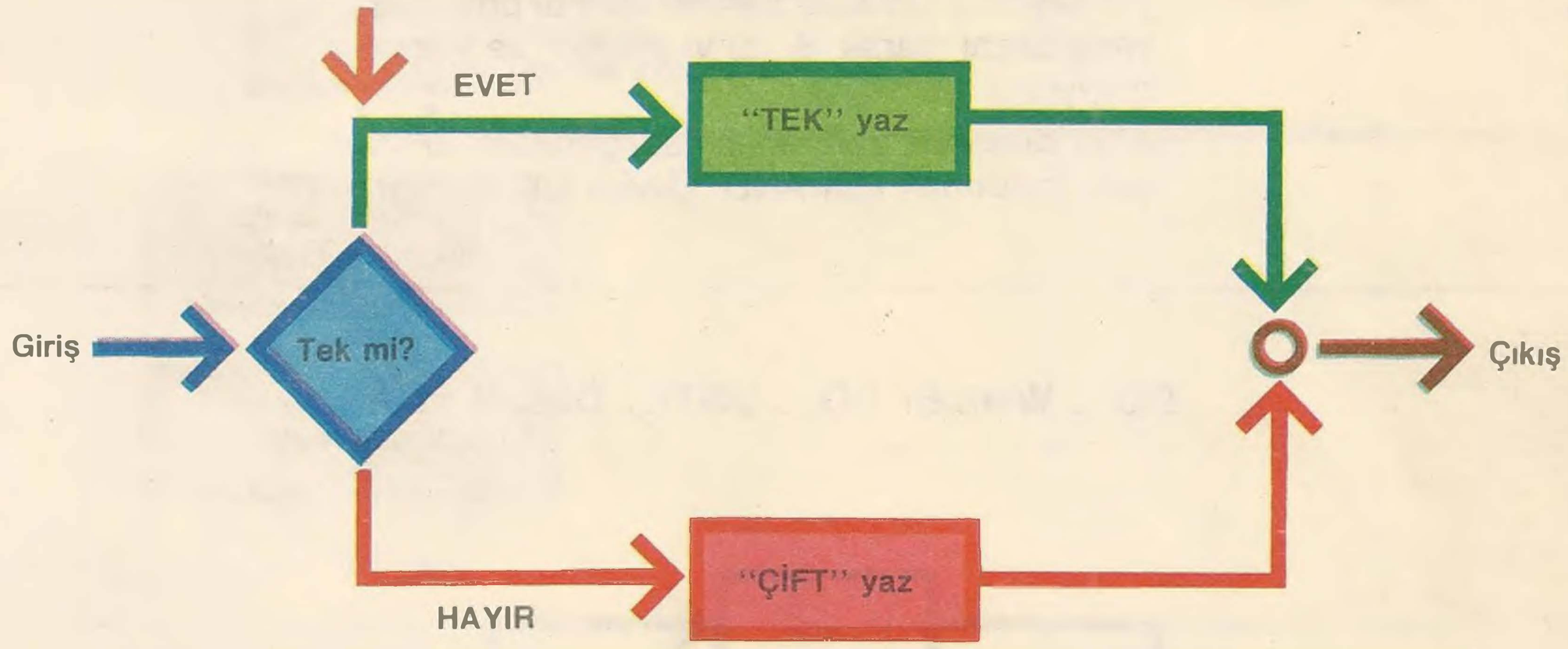
Bu bileşik bir simgedir. Bir kararla başlar (bir koşulun sınanması); sonuca bağlı olarak iki yoldan biri seçilir (koşul doğru ise biri; yanlış ise öbürü). İki yolun da çıkışı ortaktır. Yukarıdaki şema bir uygulamayı gösteriyor. Girişteki kutu bir sayı alır; bu sayı çift ise ''çift'', değilse ''tek'' diye bir iletinin yazılması gerekir.

Kutuların içerdiği cümleleri yazıya dökersek şunu elde ederiz:

IF ''sayı çift ise'', THEN ''çift yaz'',
ELSE ''tek yaz''.

Demek ki alışageldiğimiz sınama simgelerine çok benzeyen bir biçim ile karşı karşıyayız.

SINAMA (KARAR) İÇİN YAPISAL PROGRAMLAMA SİMGELERİ KULLANILARAK HAZIRLANMIŞ AKIŞ ŞEMASI



DO... WHILE ve DO... UNTIL

Bu iki işlem birbirlerine benzer ve her ikisi de bir sınama (karar) ve bir işlem içerir. Birincisinde sınama girişten sonra yapılır: İşlem yapılır ve yine sınama adımına dönülür. Sınama olumsuz bir sonuç veriyorsa çıkılır. DO... UNTIL kutularında ise işlem, sınamadan önce yapılır. Bunlar aslında iki yoğun döngü olarak düşünülebilir.

CASE

Bu öbeğin akış şemalarında kullanılan simgesel mantık ile dolaysız bir ilişkisi yoktur. Çoklu bir karşılaştırmayı simgeler. Bir sınama simgesi ile aynı sonucu elde edebilmek için, sınamanın kaç tane olası sonucu varsa o kadar CASE kutusu kullanmak gerekir. Bir uygula-

ma örneği olarak gelir vergisini alalım: Gelir vergisi gelir seviyesine bağlı olarak değişir. Yalnızca dört gelir basamağı olduğunu varsayalım:

Gelir:

— 25.000'den az	vergi = % 15
— 25.000 ile 39.999 arasında	vergi = % 20
— 40.000 ile 59.999 arasında	vergi = % 25
— 60.000 ve yukarısı	vergi = % 30

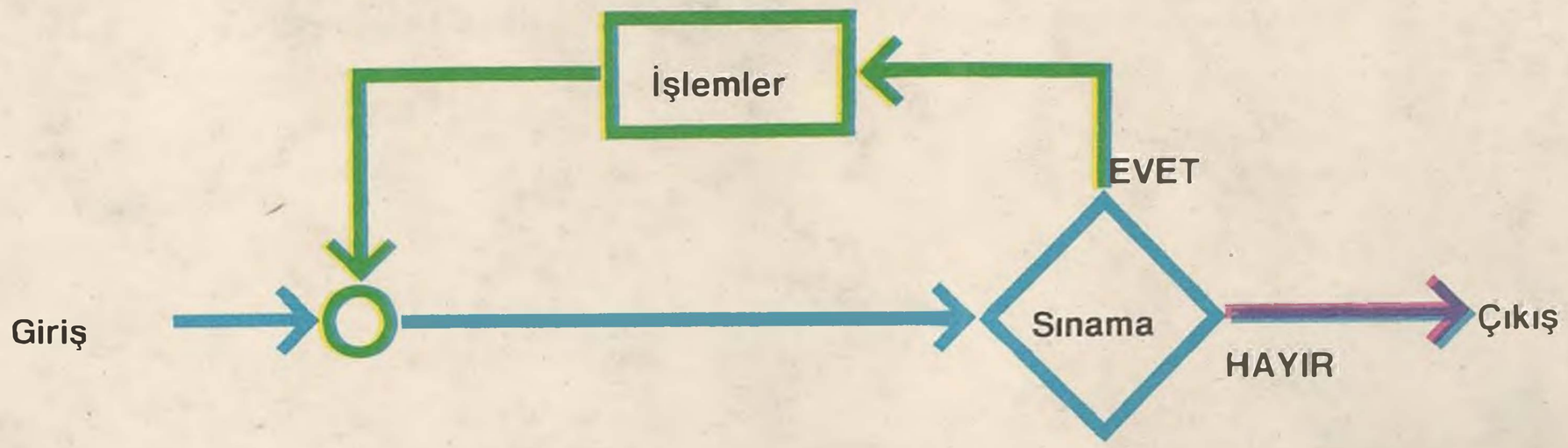
Sorun, uygulanacak oranları belirleyebilen ve net gelir vergisini hesaplayan bir programın yazılmasıdır. Yukarıdaki şema normal simgelerle sorunun akış şemasını gösteriyor.

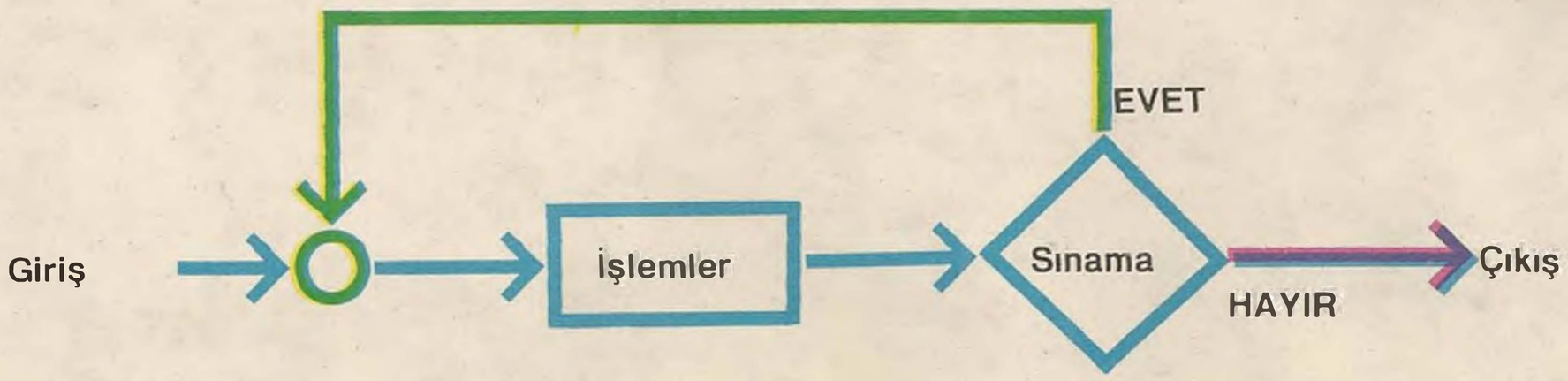
Karar 1, gelirin 24999'dan az (25.000'den çok) olup olmadığını sınıyor; gelir, 24999'dan azsa vergi oranı % 15'dir. Vergi oranını belirledikten sonra program doğrudan işleme (aşama 5) sıçrıyor.

1 kutusunun cevabı olumsuz ise karar 2'ye geçiliyor. Sözkonusu gelir ikinci basamakta yer alıyorsa (25.000'den 39.999'a) program, vergi oranı olarak % 20'yi seçiyor ve işleme geçiyor.

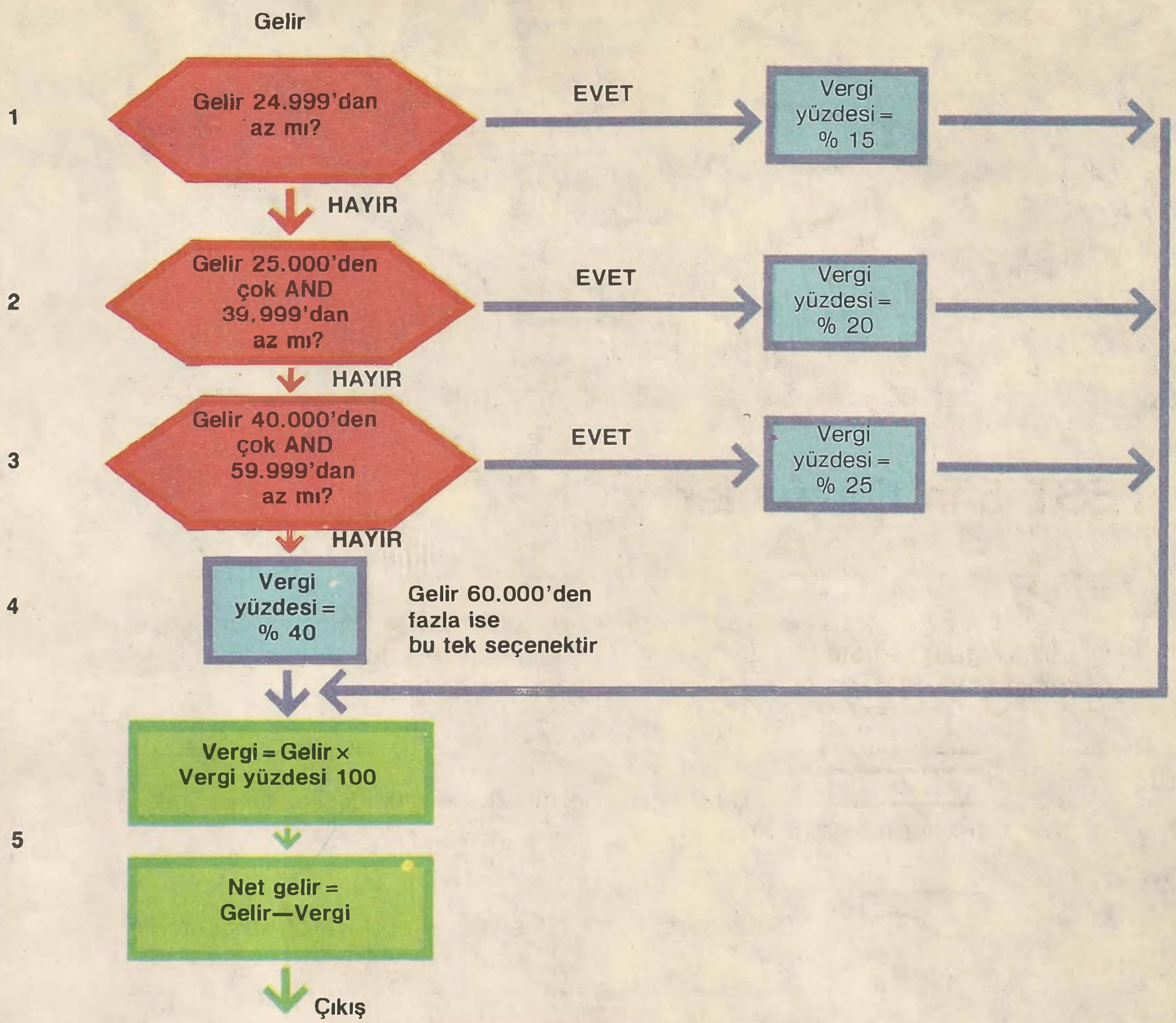
Aynı zamanda gerçekleşmesi gereken iki koşulu belirtmek için AND işlevini kullanıyoruz

DO... WHILE/ DO... UNTIL ÖBEKLERİ





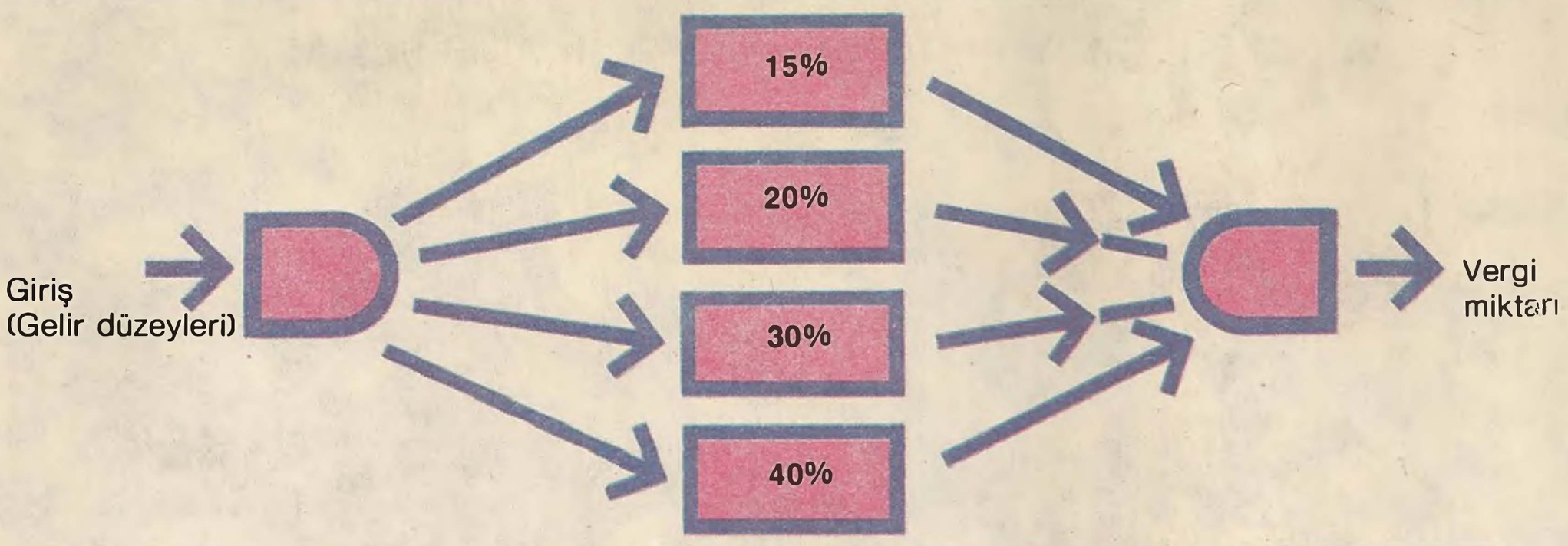
VERGİ DİLİMİNİN HESAPLANMASI İÇİN AKIŞ ŞEMASI



(24.999'dan az ve 40.000'den az gelir, yani 25.000'den 39.999'a). Üçüncü aşama hiçbir sınama gerektirmiyor, çünkü 3. aşamaya gelindiğinde gelirin 60.000'den az olmadığı belirlenmiş oluyor. Bu durumda otomatik olarak en yüksek vergi oranını kullanıyoruz.

CASE biçimini kullanarak aynı mantığı daha kısa bir yoldan ifade edebiliriz. Çünkü CASE üç sınama simgesinin birleşmesi olarak düşünülebilir. Bu ilkenin şeması sayfa 214'te bulunabilir.

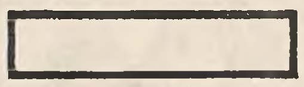
BASIC ve FORTRAN dilleri, mantığı bu kadar yoğun bir biçimde ifade eden komutlar içermezler.



TEST 5'in Çözümleri

1 / a - Simge bir kararı belirtiyor. İşlemlerin yürütülmesi genel bilgi-işlem simgesiyle (dikdörtgen) belirtilir. Bu dörtgenin içine de gerçekleştirilecek işlem yazılır. Programın sonu SON ya da END yazısı taşıyan bir elipstir.

---

2 / a-b -  Bu simge genel olarak işlemler için kullanılır. Dolayısıyla hem a hem b bu simge ile gösterilir.

c - Gerek giriş gerek çıkış işlevleri bu simge ile gösterilir.

---

3 / Bu simge gönderme simgesidir. Akış şemasında bir noktadan, o noktaya uzak başka bir yere gidilmesi gerektiğinde, iki noktayı bir akış çizgisiyle birbirine bağlamak şemayı çok karmaşıklaştırırdı. Böyle durumlarda akış çizgisinin yerine bu simge kullanılır.

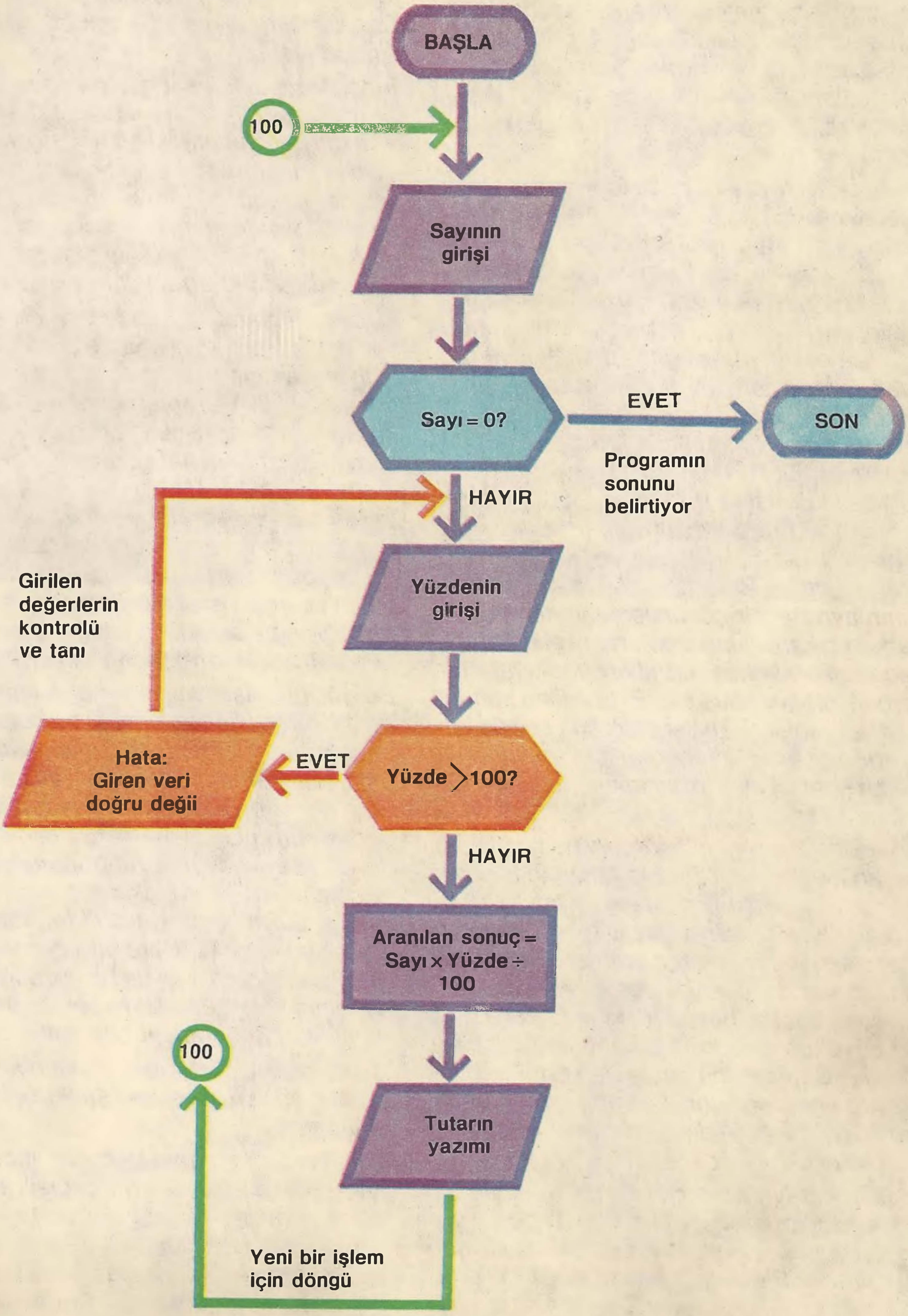
---

4 / a - Çentik bir koşulun gerçekleşip gerçekleşmediğini sınayan bir belirteçtir.

---

5 / Önce problem şu şekilde çözümlenir:
— Yüzdesi alınacak sayının girilmesi.
— Sayının 0'a eşit olup olmadığının sınanması (0 değeri, programın durdurulmasını sağlayabileceğinden bir çentik işlevi görebilir).
— Hesaplanması istenen yüzdenin girilmesi ve sınanması.
— Hesap. Sonuç = sayı × istenen yüzde/100.
— Bilgiyazım.
Akış şeması yandaki sayfada verilmiştir.

6 / Hata **I** ve **J** döngülerinin kesişmesinden doğuyor. 100 sayılı kutuda başlayan **I** döngüsünün, **J** döngüsü kapandıktan sonra kapanması gerekiyor. Hata 500 ve 600 sayılı kutulara yer değiştirtilerek düzeltilebilir.



Konuşan Makineler

Teknolojideki ilerlemelerin bilişime uygulanması, birkaç yıldır, insan ile makine arasındaki sözlü alışveriş denemelerini mümkün kılmıştır.

Bu denemelerin önünde duran sorun iki yönlüdür: makine bir yandan insan sesini duymalı ve anlamalı, diğer yandan da tutarlı bir şekilde konuşabilmeli ve sorulan sorulara cevap oluşturacak cümleler kurabilmelidir. Konuşabilen elektronik aygıtlar iki temel ilkeye dayanırlar.

Bu ilkelerden birincisi ''bağlı ***(concatenated)*** *sözcük bireşimi''dir. Bu ilkeyle bilgisayardan, belli bir dilin bütün hecelerini içeren bir tür sözlük gibi yararlanılır. Bunu yaparken mümkün bütün ünlü-ünsüz ve ünsüz-ünlü birleşimleri belleğe yerleştirilir. Bir sözcük oluşturmak için, örneğin, (''bil-gi-sa-yar'' demek için) bilgisayar belleğinden, gereken sözcükleri çıkarır ve bunları doğru sırada birbirine bağlar. Bağlı konuşma yöntemi çok katıdır ve sağladığı konuşma biçimi de çok kötü tınlar. Bununla beraber konuşma hızlıdır ve göreli olarak ekonomiktir. ''Kurallarla bireşim'' denilen ikinci yöntemde ise bellekte ses ve ses birleşimleri saklanamaz. Bunun yerine insan konuşmasının ayrıntılı bir çözümlemesinden ve konuşurken çıkardığımız seslerin üretilme biçiminden çıkarsanmış kuralları toplayan bir elektronik rehber kullanılır. Elde edilen konuşma daha iyi tınlar. Buna karşılık çok sayıda kuralı yerine getirmek gerektiği için, makinenin böyle bireşimsel bir yoldan konuşması çok vakit almaktadır.*

''Bağlı sözcük bireşimi'' yöntemini uygulayan konuşan makineler, özellikle Amerika Birleşik Devletleri'nde kullanılmaktadır. Konuşan hesap makineleri, tuşlara basıldığında, sayıları göstermek yerine telaffuz ederler. Dolayısıyla, sonuçları da yüksek sesle okuyabilirler. Bunlardan başka, hesap makinesinden biraz daha büyükçe olan küçük konuşma aygıtları da vardır. Bunlardan çocuklara konuşma öğretmekte yararlanılıyor. Satranç oynayan bir diğer küçük makine de hamlelerini sözlü olarak açıklamaktadır.

Konuşan bir aygıt, normal olarak insanlar tarafından sağlanan bazı hizmetlerin gerek verimini artırmak gerek maliyetlerini düşürmek için, Amerika Birleşik Devletlerinde Bell Telefon Şirketi tarafından kullanılan bir sistemin temel ögesini oluşturmaktadır. Bu sistemde, daktilonunkine benzeyen küçük bir klavye telefon cihazına yerleştirilir. Herhangi bir abonenin numarasını bulmak için, abonenin adı ve adresi klavyeyle yazılır. Hattın öbür ucundaki bilgisayar, belleğinden istenen numarayı arayıp bulur ve sesli olarak cevabı bildirir. Ancak soruların klavyeyle yazılmasının elenmesini öngören bir sonraki aşama daha büyük bir sorun çıkarmaktadır. Çünkü bu aşamada, makinenin sesli komutu anlayabilmesi de gerekir.

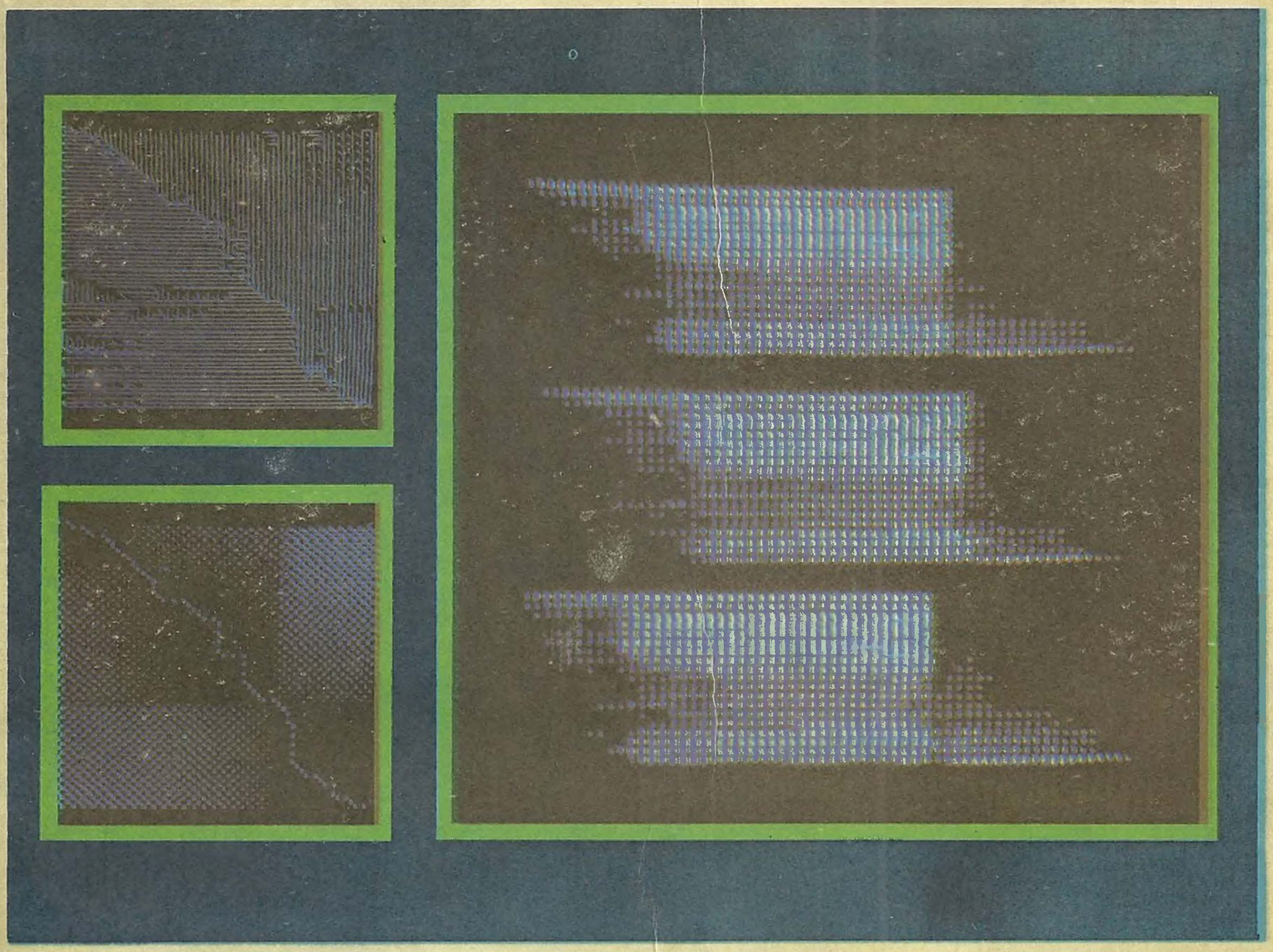
İkinci Dünya Savaşı'nın başında keşfedilen akustik spektrograf, makinenin konuştuğumuz dili anlayabilmesi olanağını araştıran uzmanlar için temel bir aygıttır. Söz konusu olan, ses dalgalarını çözümleyerek, bunların kalıcı bir çizisini elde eden elektronik bir aygıttır. Bu çizimlerin elverişli yanı bilgisayarın belleğinde (sayısal olarak) saklanabilmeleridir.

Kuramsal olarak, bilgisayarın, aldığı seslerin spektrogramı ile belleğinde saklı bilgileri karşılaştırarak bir sözcüğü tanımasının kolay olması gerekir. Bütün sorun aynı yerel lehçeyi kullansalar bile, kişilerin bir sözcüğü tamamıyla aynı biçimde telaffuz etmemelerinden kaynaklanmaktadır. Değişik kişilerce telaffuz edilen aynı sözcükler, farklı spektrogramlar oluşturmakta, bu da asıl güçlüğü doğurmaktadır. Ancak bu durum, bilgisayardan bir sesin sahibini tanıması istendiği zaman çok yararlı olabilir (örneğin sahibinin isteği üzerine kapılarını elektronik olarak açan bir arabada).

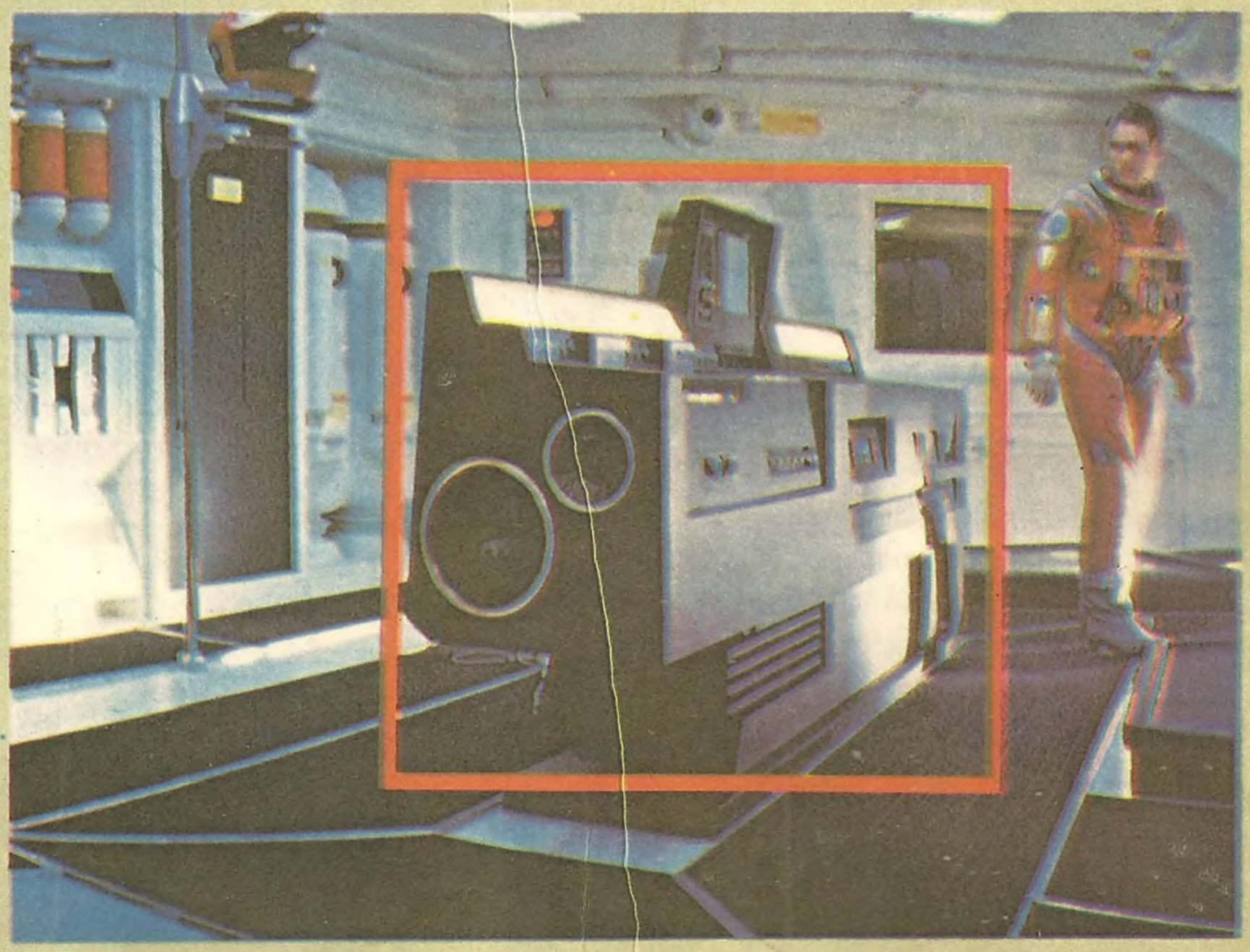
Bu güçlüğü aşmak için şimdiye kadar bulunan en etkili yol, bilgisayarı tıpkı bir çocuğa konuşma öğretirmişcesine ''biçimlemektir''. Bu eğitim yerel lehçeyi kullanan yüz kişiden oluşan bir grupla başlar. Kaydedilen değişik spektrogramlar, daha sonra bilgisayara verilir. Bilgisayar aynı sözcüğün çizisel tablolarını çözümler ve bir tipik tablo çıkarır. Son olarak da bu tabloyu, gelecekte, karşılaştırmalı olarak kullanmak üzere belleğinde saklar. Bilgisayarın diğer bir işi de bir sözcüğün her söylenişinin tipik tablodaki spektrograma oranla oluşturduğu sapmayı belirlemektir.

Daha sonra, başkaları makineye başvurduğunda, sesleri bellekteki spektrolarla karşılaştırılacaktır.

Bellekte biriken spektrolar bir model rolü oynamakta ve sözcüklerin değişik telaffuz hızlarına kendilerini uydurabilmek için uzayıp kısalarak şekil değiştirebilmektedirler. Makinede kayıtlı bu spektrolardan biri bilgisayarın işittiği sözcüğün spektrosu ile çakıştığında, bilgisayar telaffuz edilen sözcüğü tanımış olacaktır.



Sağda, yukarıdan aşağıya: Aynı kişi tarafından telaffuz edilen ‘‘sum’’ (toplam) sözcüğünün iki spektrosu ve bilgisayar tarafından gerçekleştirilen bireşim. Solda: Aradaki farkın çözümlemesi.



Hal, ‘‘2001, the Space Odissey’’ filminde ‘‘oynayan’’ bilgisayar, ve kurgu-bilimin klasik konuşan makine örneği.

Bu şekilde hazırlanmış makineler kendilerine söylenen sözcüklerin % 80 ile 90'ını anlıyorlar. Eğer konuşan, sözcükleri makineye ''öğretmiş'' olan kişi ise bu oran % 98'e çıkıyor. Bundan böyle basit emirleri anlayan ve bu emirlere göre davranan makineler yapılabilecektir. Örneğin, şekerli bir bardak sütlü çay istediğimizde, yalnızca ''çay'', ''süt'', ''şeker'' dememizin yeteceği bir içecek dağıtıcısını hayal etmek hiç de zor değildir. Bize olağanüstü gibi gelen bu durum, bir sonraki aşama gözönüne alındığında yalnızca bir başlangıçtır. Bundan sonra ulaşılması istenen aşama bilgisayarın yalnızca tek tek kelimeleri değil, süregiden bir konuşmanın cümlelerini de anlayabilmesi.

1970'te Amerikan Savunma Bakanlığı, bir grup araştırıcıdan, insanın konuştuğu dili anlayabilen bir elektronik sistemden, mantık sınırları içinde, ne beklenilebileceği konusunda bir rapor istedi. Raporun sonuçlarına göre, bu tip bir sistem 1.000 sözcüklük bir dağarcıkla sınırlı kalacak (ikinci sınıfa kadar öğrenim görmüş bir kişinin dağarcığı 5.000 ile 10.000 arasında terim içerir) ve dilbilgisi açısından yalnızca çok basit bir sözdizimi ile kullanılabilecekti. Sözcüklerin makineye yüklenmesinin de bilgisayar programlamayı andırması gerekiyordu. Sözcükler hecelere bölünmeden ve sürekli olarak telaffuz edilmediği takdirde, makine her heceyi ayrı bir sözcük olarak yorumlayabilirdi. Öyleyse, makineyi bu şekilde yükleyecek olan kişinin sesi pürüzsüz, kullanılan mikrofon iyi nitelikli olmalı ve kayıt tümüyle sessiz bir ortamda yapılmalıydı.

1970'lerin sonuna doğru, bazı gelişmelere rağmen, sözcükleri tanıyabilen makineler sözkonusu raporun dile getirdiği sınırlı amaçları bile gerçekleştirebilmekten çok uzaktılar. Sözcük dağarcıkları son derece fakirdi. Yapılabilmiş en iyi iki sistem, sırasıyla 39 sözcük (Amerikan alfabesinin 26 harfi, 0 ile 9 arasındaki rakamlar ve üç emir) ve 54 sözcüklü dağarcıklara sahipti. Makine, üst sınır olarak açıklanan 1.000 terimin çok uzaklarındaydı.

Buna rağmen 1980'lerde bir IBM araştırma ekibi amaca ulaşıldığını açıkladı. Bu araştırmacılar bir IBM 370'i konuşulan cümleleri bilgiyazara basabilecek şekilde programlamışlardı. Ama bir kere kullanılmadan önce bir işletmenin makineyi ''alıştırmak'' için ona 900 cümle okuması gerekiyordu. İkincisi bazen otuz saniyede telaffuz edilen bir cümlenin makinede işlenişi 100 dakikayı bulabiliyordu. Hata oranı da hâlâ çok yüksekti. Diğer bir deyişle, bu kullanışlı bir girişim değildi. 1970'deki Amerikan heyeti kuşkusuz böyle bir projenin zorluğunu öngörememişti.

Aslına bakılırsa, insan dili son derece karmaşıktır ve en basit, en açık konuşma bile yüzlerce küçük sözel parçadan ve olaydan oluşur. Makine bunları yorumlamadan önce özenle tanımlamalıdır. Aynı zamanda bir sözcüğün nerede bittiğini ve bir sonrakinin nerede başladığını belirlemelidir. Oysa kişilerin çoğu, sözcükleri tam ayırmadan, hızlı konuşur (Örneğin bazı dillerde sözcükleri birbirine bağlama alışkanlığı vardır).

Ayrıca, dil bilmek dediğimiz ve bizim konuşabilmemizi ve konuşulanı anlayabilmemizi sağlayan beceri, yalnızca milyonlarca sözcüğün ve hecenin ezberlenmesinden ibaret değildir. Aksine bir dili bilmek demek, her şeyden önce çeşitli sözel parçaları birbirine doğru bir biçimde bağlamamızı ve bu şekilde oluşturulan cümleleri doğru biçimlerde kullanmamızı sağlayan kuralları bilmek demektir.

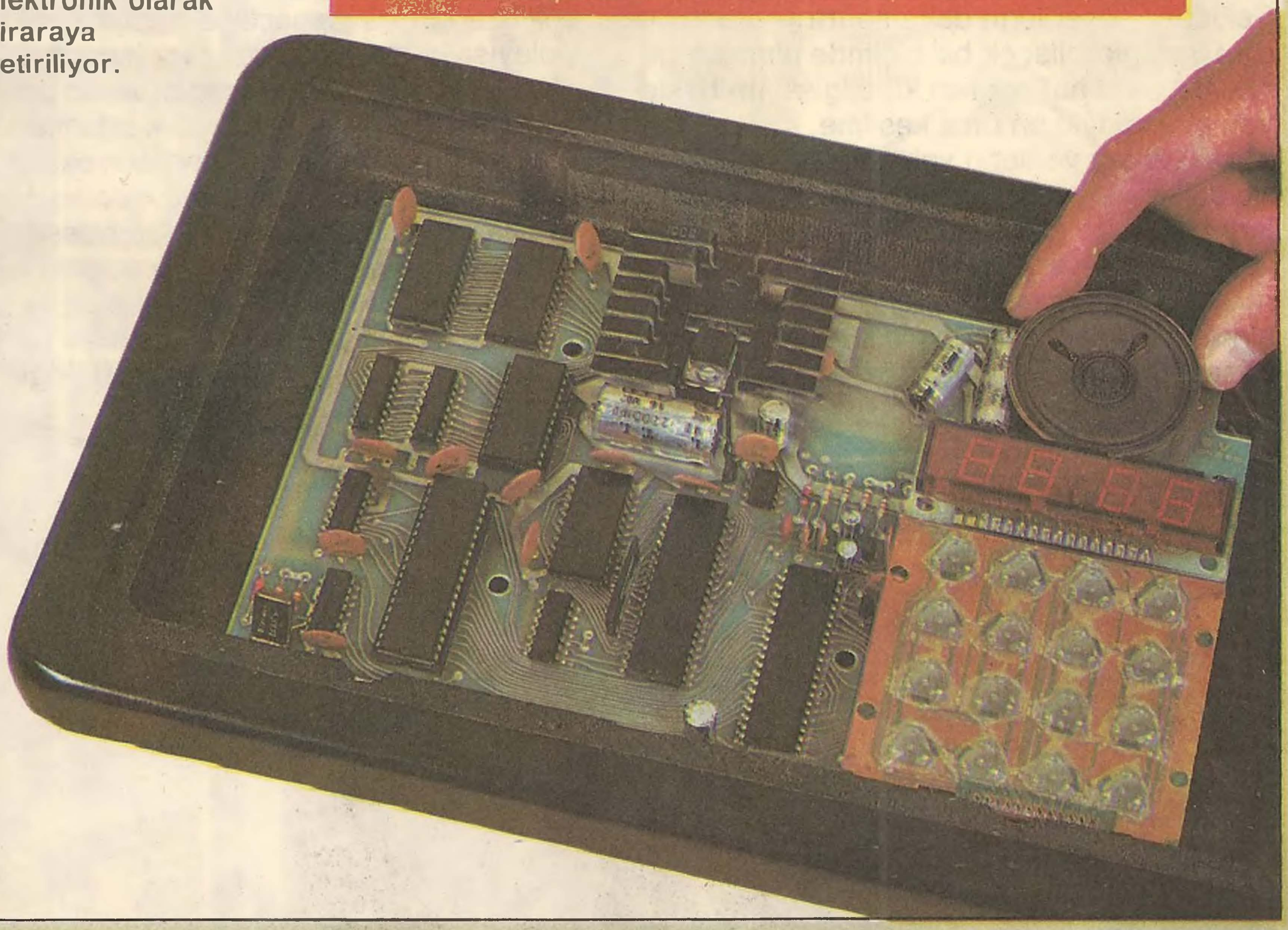
Bu kuralların çok küçük bir kısmı bize okulda, eğitimimiz sırasında öğretilir. Daha büyük bir çoğunluğunu ise kendi yetişmemiz, gelişmemiz sırasında keşfederiz. Ama bu yüzden de çoğumuz doğru bir biçimde kullandığımız bu kuralların içeriğinin ne olduğunu bilmeyiz. Başka bir deyişle, Türkçe konuşanların, yani Türkçe'nin kurallarını doğru olarak kullananların çoğu bu kuralların ne olduğunu bilmez. Kısacası, bir dili bilmek dediğimiz zaman kastettiğimiz kurallar bilgisi **örtük bilgi** dediğimiz türdendir.

Oysa bu bilginin bilgisayara yüklenip, makinenin konuşması ve konuşmayı anlayabilmesini sağlamak için önce bu örtük bilginin açık seçik bir hale getirilmesi gerekir.

Örneğin bir makine sözcükleri birbirinden ayırabilecek ve tanımlayabilecek yetkinlikte olsa bile, bir sözcüğün değişik anlamlarını da ayırımsayabilmelidir.

Ancak her kelimenin bütün farklı anlamlarıyla birlikte bilgisayara yüklendiği bir sistem de çok yavaş işleyen bir sistem olurdu. Aslında daha bilimsel olan başka bir çözüm, bilgisayara bütün bir dili ezberletmektense yalnızca o dilin kurallarını, yani o dilde cümlelerin na-

Bu makine yalnızca mükemmel bir şekilde satranç oynamakla kalmıyor, konuşuyor da. Yani oynayacağı hamleleri ''telaffuz ediyor''. Sözlerin bireşimi belleğinde kayıtlı ses parçalarının birbirine zincirlenmesi ile elde ediliyor: Sesler, sözcükler ve cümleler oluşturmak üzere elektronik olarak biraraya getiriliyor.



sıl kurulduğunu öğretmek olurdu. Bunun için örneğin, bir isimden önce ve sonra hangi sözcük türlerinin kullanılabileceğinin, fiillerin cümlelerin neresinde yer alabileceğinin belirlenmesi gerekir. Örneğin, bilgisayara Türkçe'de zamirlerin fiillerden önce kullanılamayacağına dair bir kural verilebilir. Ancak böyle bir kural bile yüzde yüz geçerli değildir.

Bugünkü duruma baktığımızda bilgisayarla karşılıklı konuşmayı mümkün kılacak sistemlerin daha yıllar alacağı ve araştırmacıların belkide yepyeni yollar denemek zorunda kalacağını söyleyebiliriz. Yine de kullanıcısıyla konuşabilen bilgisayarların üretimi fikri çok heyecan vericidir. Bunun salt bilim-kurgu olarak kalıp kalmayacağını zaman gösterecektir.

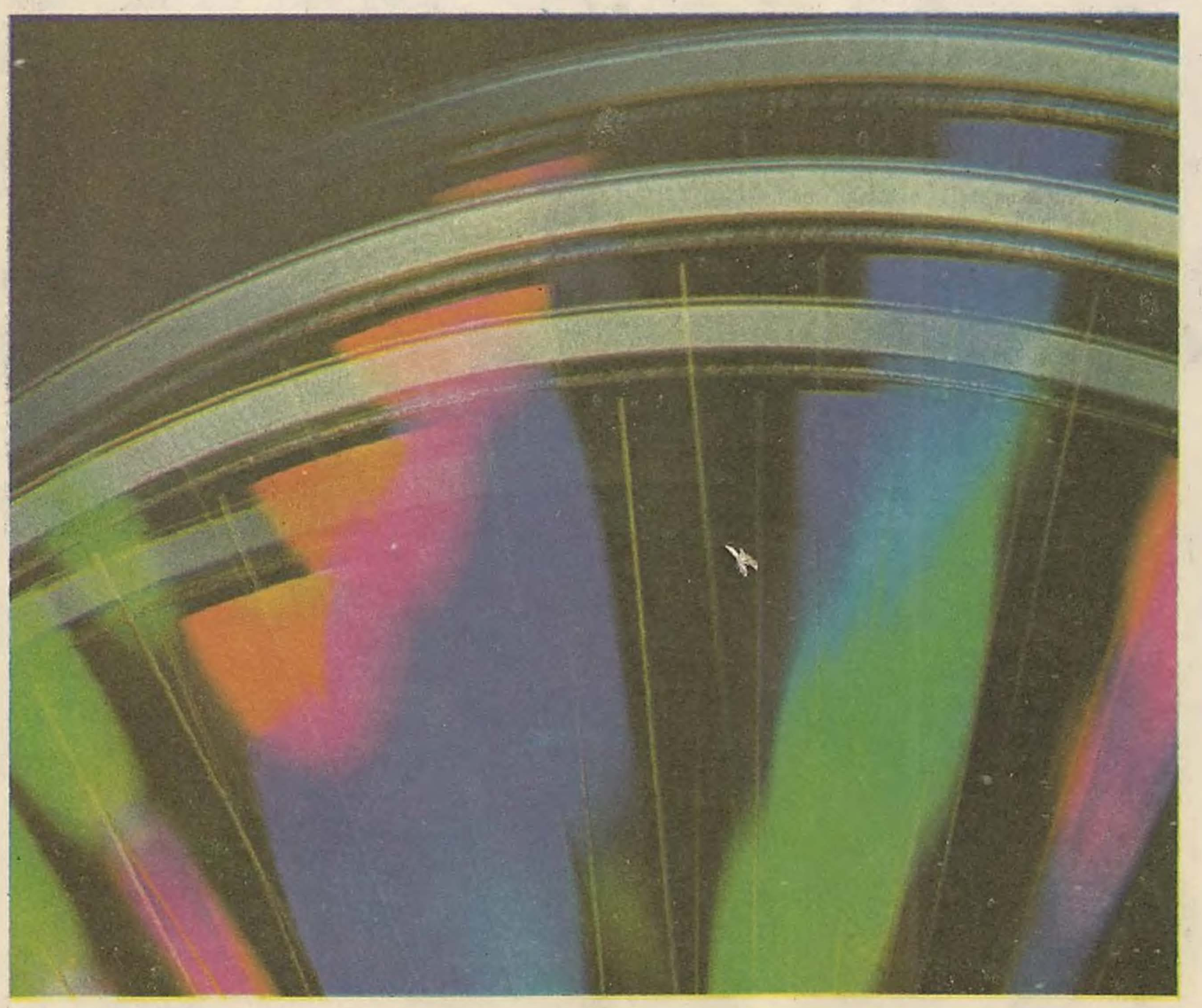
Verilerin Saklanması

Bilgisayarca işlenecek veriler, merkez bellekte sürekli olarak tutulamaz. Veriler yalnız işlenmeleri sırasında burada kalabilir, daha sonra yerlerini, işlenecek başka bilgilere bırakırlar. Bu durumda, makinenin yeni bir kullanımı belleğin içeriğini silmeden, önceki bilgi-işlemle ilgili bütün bilgilerin saklanması gerekir. Bunun için, daha sonra anlatacağımız, bu işe elverişli ortamlar kullanılır. Bir bilgisayar çok büyük sayıdaki bilgiyi, son derece büyük bir hızla alıp, işleyip üretebilir. Dolayısıyla bu bilgileri saklayacak yüksek kapasiteli ortamların da olması gerekir.

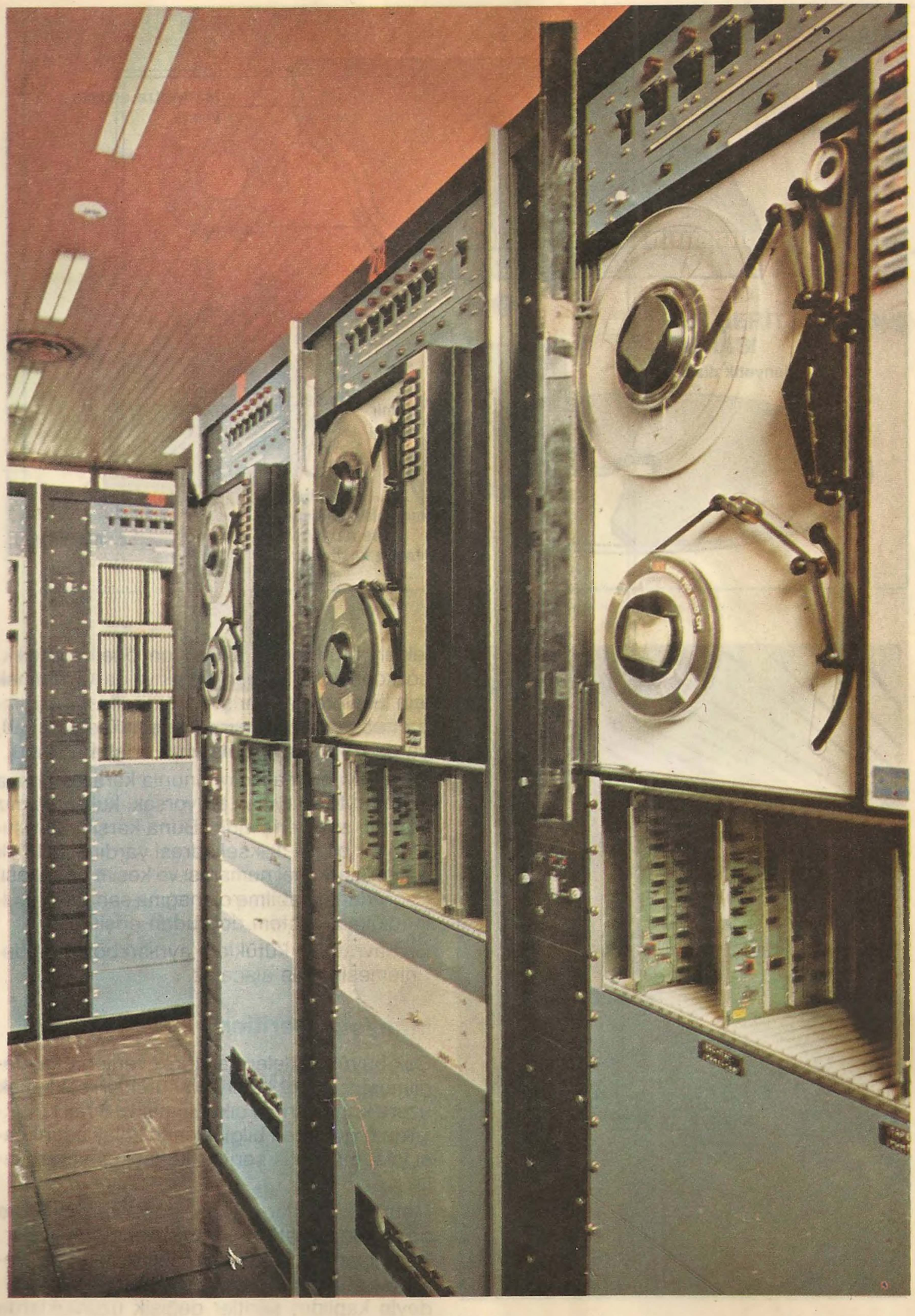
Saklanacak verilerin daha sonra yeniden bilgisayara girebilecek bir biçimde olmaları gerekir. Ayrıca hatırlatalım ki, bilgisayarı besleyen elektrikteki en ufak kesilme, merkez bellekteki bütün verilerin yok olmasına yol açar. Bu da verilerin, tümüyle güvenilir, içeriğini yıllarca koruyabilen, elektrikle beslenmesini bile gerektirmeyen ortamlarda saklanması gerektiğini göstermektedir. Bir örnek verelim: Bir şirkette çalışan işçilerin aylıkları ayda bir kez hesaplanmaktadır. Bu veriler her ayın yalnız belirli bir gününde (örneğin 20 ya da 21 'inde) kullanılacaktır. Ayın geri kalan bölümünde, ne kadar büyük olursa olsun, bilgisayarın belleğinin, milyonlarca karakterle doldurulması düşünülemez. Üstelik, Ahmet beyin aylığı hesaplandıktan sonra, hesap sonucunun dışarıda arşivlenmesi, gereken bellek bölümünün başka birim hesapları, örneğin Mehmet Bey'in aylığı için kullanılması, daha uygun olacaktır. Bu örnekte verilerin, yardımcı bellek ya da yığın bellek denen, bilgisayarın dışındaki ortamlarda saklanmasının önemi açıkça görülmektedir.

Yığın Bellekler

Verilerin delikli kart ya da şeritlerde saklanması üzerinde durmayacağız. Günümüzde artık geniş ölçüde terkedilmiş bu ortamlara ancak çok özel birtakım amaçlarla başvurulmaktadır. Dolayısıyla biz daha çok, verilerin manyetik olarak kaydedilmesi olanağını veren ortamları inceleyeğiz. Bütün manyetik ortamların bazı ortak özellikleri vardır. Verilerin okunma ve yazılmaları, okuma ve yazma kafaları denilen aygıtlar içeren özel araçlarla gerçekleşir. Manyetik ortamın temel parçalarına ulaşım, veri taşıma ortamının mekanik bir hareketi ve manyetik yüzeyin yakınına yerleştirilen bir kafa aracılığı ile olur (bkz. şekil, s.223). Yığın bel-

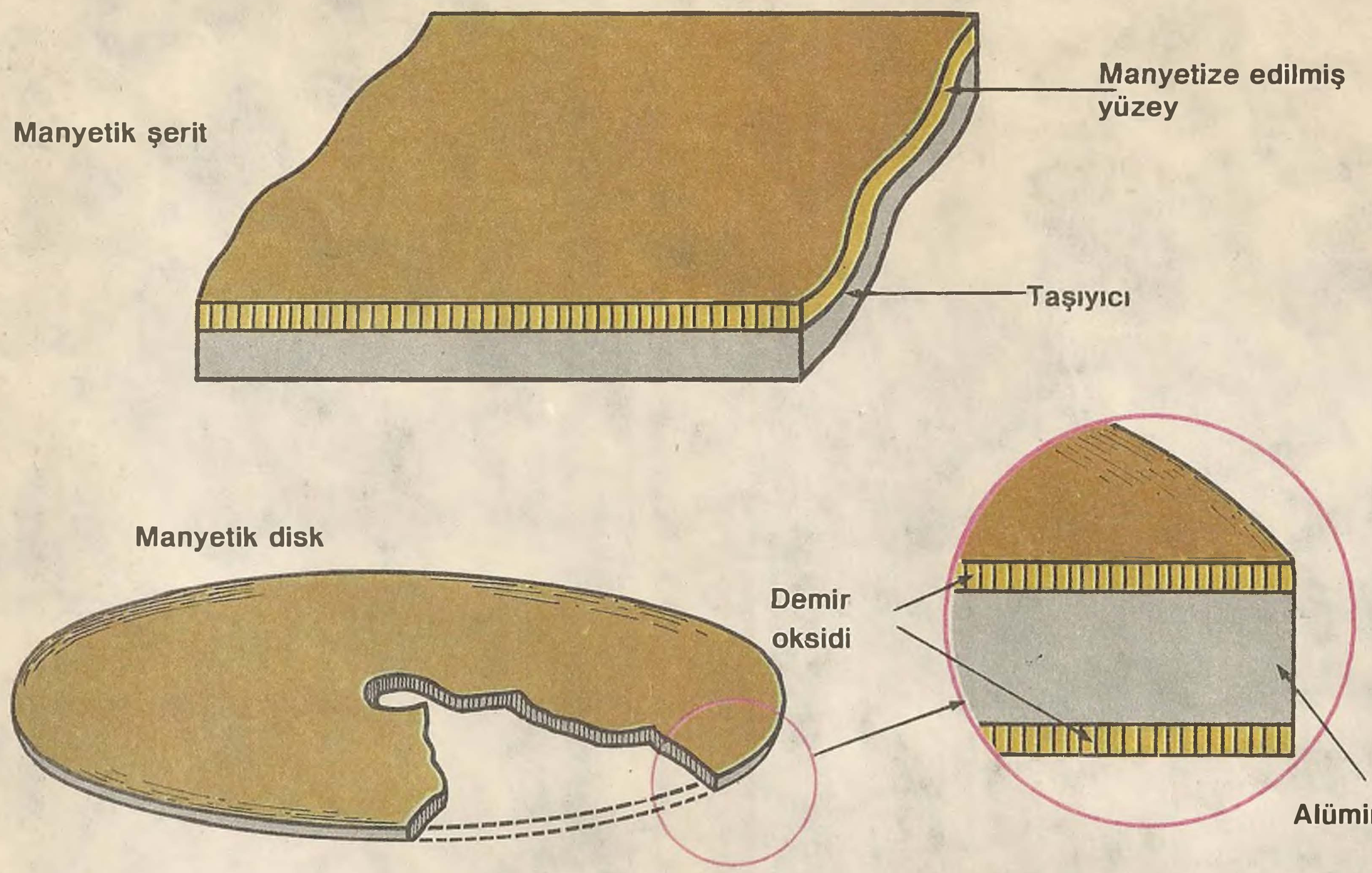


Teknolojideki son gelişme, optik disklerin bellek olarak kullanabilme olanağıdır. (Bkz, s.194-195'teki çerçeve yazı.) Resimde görülen 12 inçlik disketle 40.000 daktilo sayfası yazı saklanabilir.



Bir bilgi-işlem merkezinde manyetik şerit sürücüleri.

MANYETİK ORTAMLAR



Bir yığın bellek örneği.

lekleri birbirinden ayıran en önemli özellik, sözkonusu bellekte, okumak ya da kaydetmek için bilgilere nasıl erişileceğidir. İki tür erişimden söz edilebilir: **sıralı erişim** (sequential), ya da **doğrudan (dolaysız) erişim** (direct).

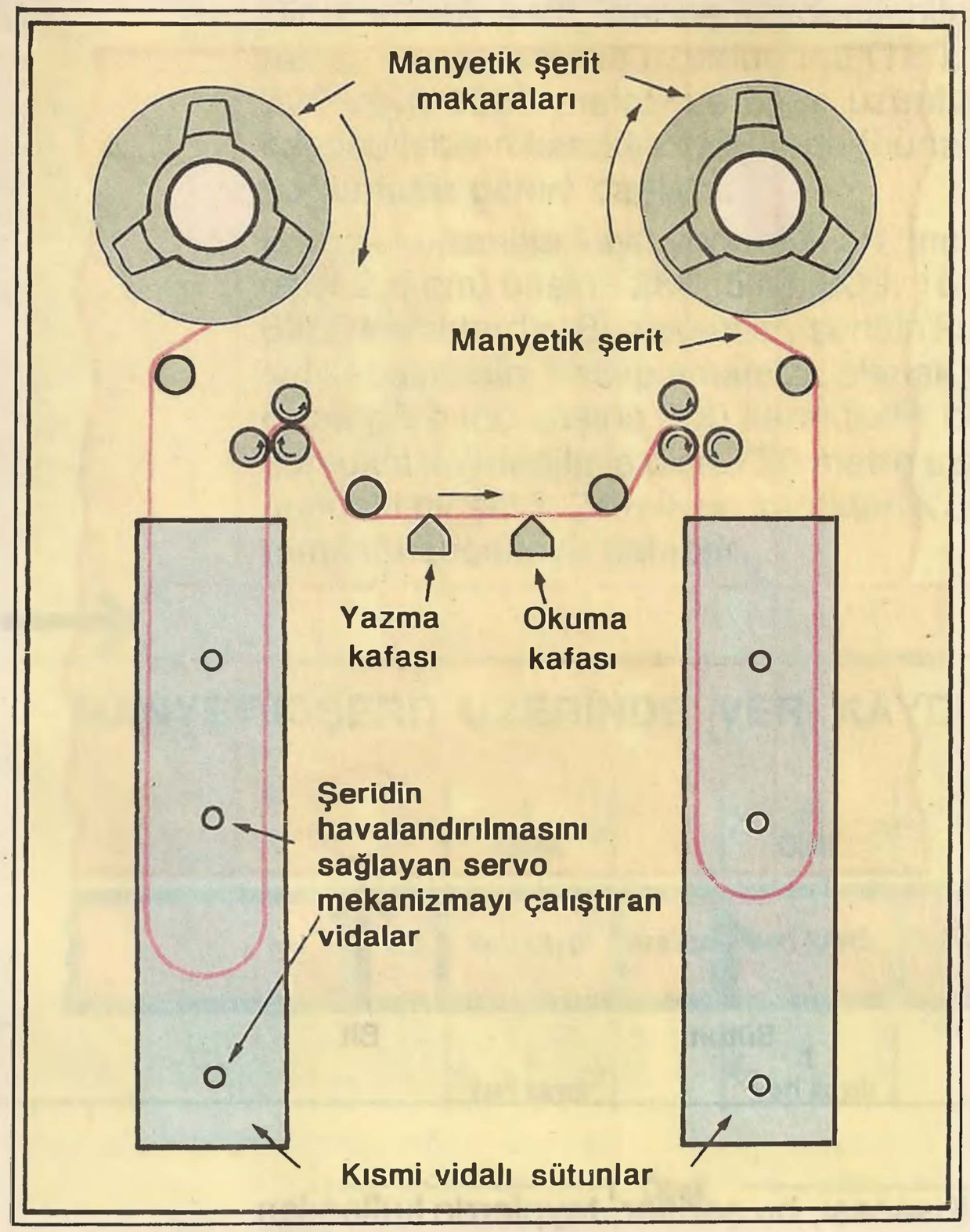
Bir bilgiyi bulmak için, onunla karşılaşıncaya kadar bütün kayıtları tarıyorsak, kullandığımız yöntem sıralı erişimdir. Buna karşılık, araştırılan bir bilgiyi fiziksel adresi yardımıyla (disk numarası, kanal numarası ve kesim numarası) doğrudan bulabilme olanağına sahipsek, kullandığımız yöntem doğrudan erişimdir.

Bu kavramları kütüklere ayrılan bölümde derinlemesine ele alacağız.

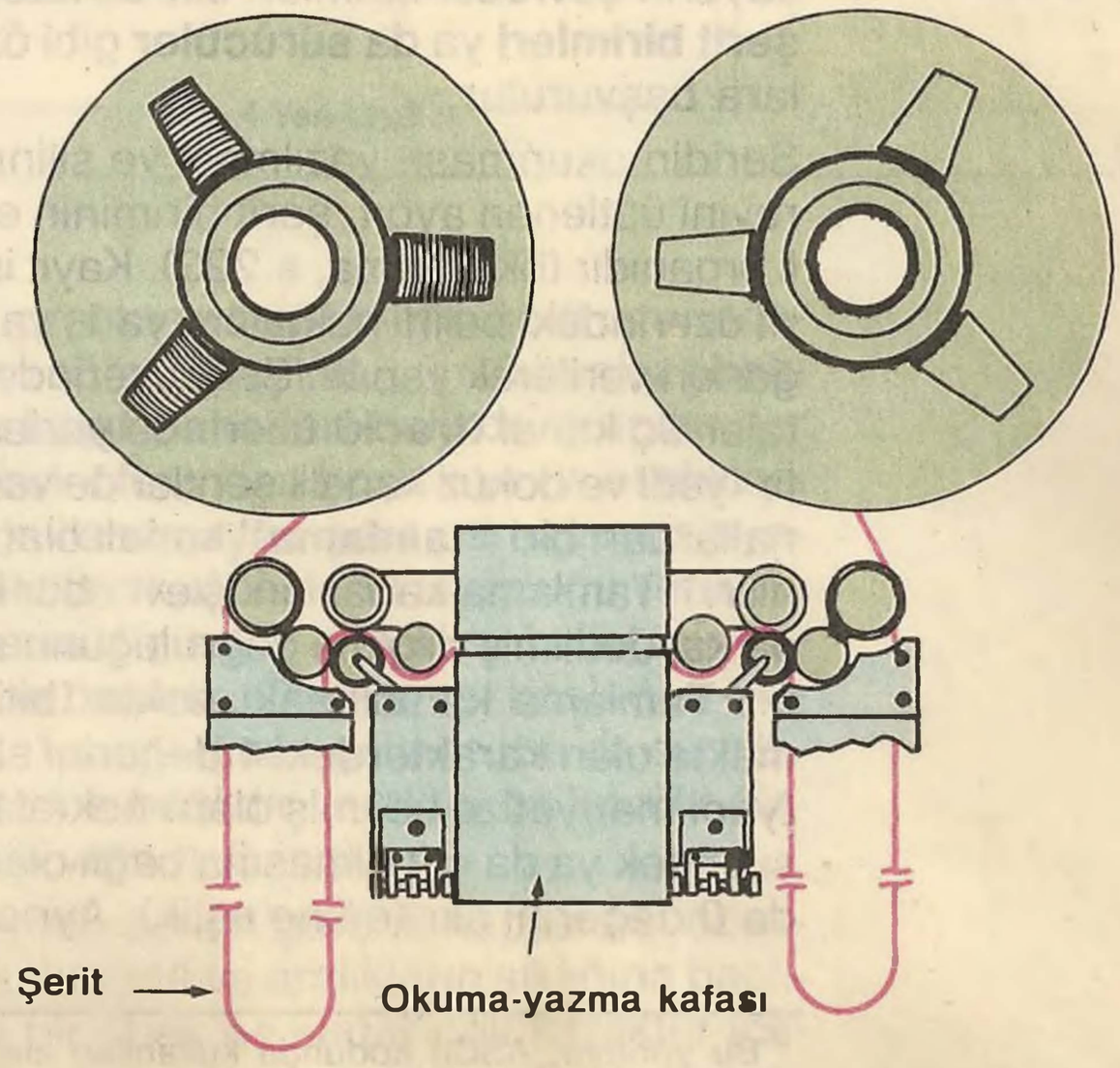
Manyetik Şeritler

Çok büyük kütleler halindeki verilerin kaydedilmesi, çok yer kaplayan kütükleri saklamak üzere kopyalarının çıkarılması ve sıralı kütüklerde örgütlenen bilgi-işlemin kolaylaştırılması için, manyetik şeritlerden çokça yararlanılır.

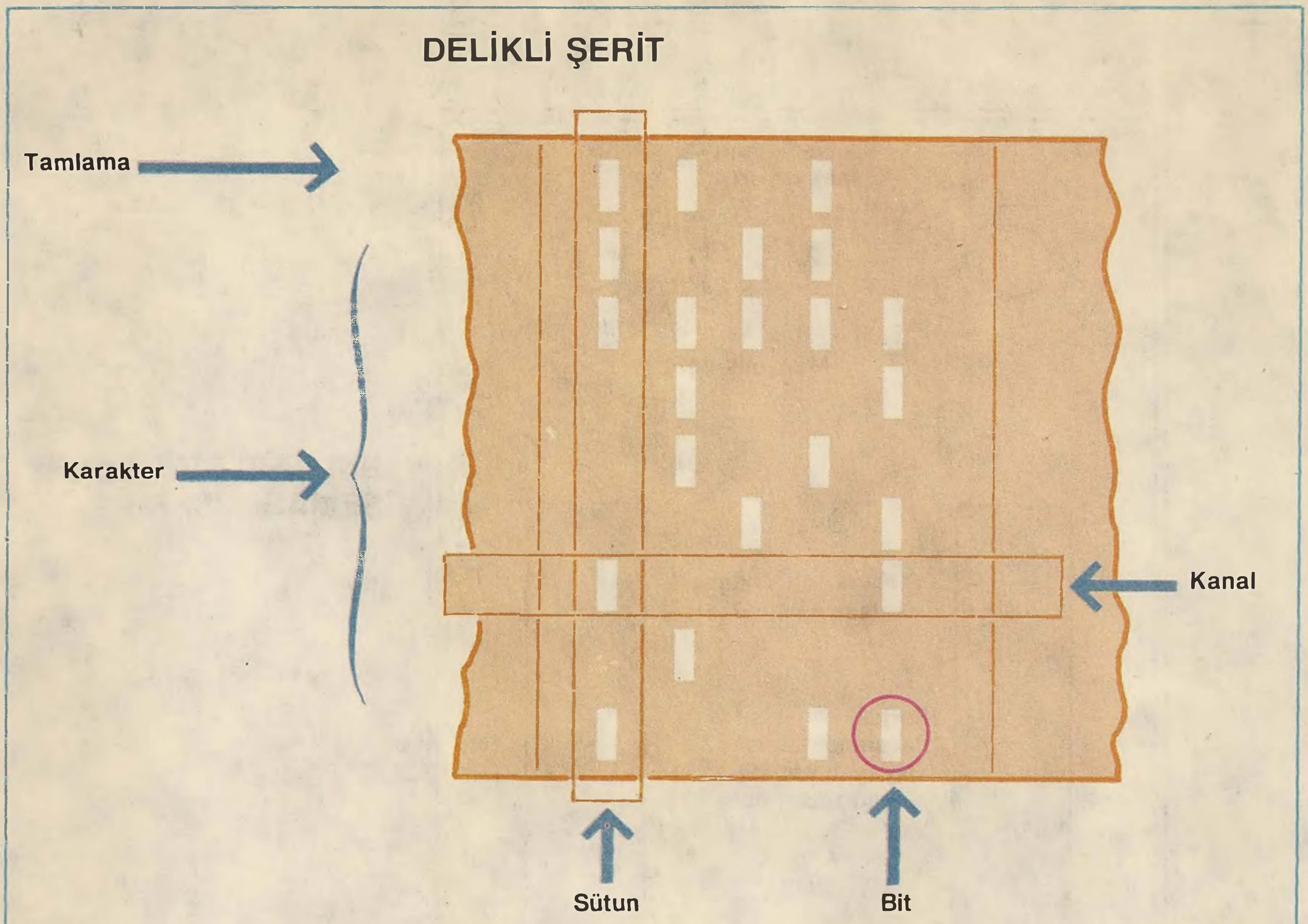
Manyetik şeritler naylon, polyester, milar ya da selüloz asetatı gibi maddelerden yapılır. Yarım parmak (12.7 mm) eninde olan şeritlerin bir yüzü manyetize edilebilecek bir maddeyle kaplıdır; şeritler değişik uzunluklarda olabilir (sık kullanılan bir uzunluk 730 metre-



BİR ŞERİT BİRİMİNİN ŞEMASI



BİR ŞERİT BİRİMİNİN YAPISI



dir). Kısacası, bu şeritler, teyplerde kullanılan bantlara çok benzer.

Bu manyetik şeritleri izlemek (okuma) ya da üzerine verileri kaydetmek (yazma) için, bilgisayarın çevresel birimleri olarak kabul edilen **şerit birimleri** ya da **sürücüler** gibi özel araçlara başvurulur.

Şeridin okunması, yazılması ve silinmesi görevini üstlenen aygıt, şerit biriminin en önemli organıdır (bkz. şema, s.223). Kayıt işlemi şerit üzerindeki belirli noktalara ya 1, ya da 0 değerleri verilerek yapılır. Şerit üzerinde bu noktalar üç kanal **(track)** üzerinde yerleştirilmiştir (yedi ve dokuz kanallı şeritler de vardır). Kanallardan biri **"tamlama"** kanalı olarak kullanılır.* Tamlama kanalının işlevi öbür kanallarda kaydedilmiş bilginin doğruluğunu sınamaktır. Tamlama kanalındaki nokta (bit), sınanmakta olan karakterdeki 1 değerini almış olan (yani manyetize edilmiş olan) noktaların sayısının tek ya da çift olmasına bağlı olarak, 1 ya da 0 değerini alır (enine eşlik). Ayrıca tek bir

---

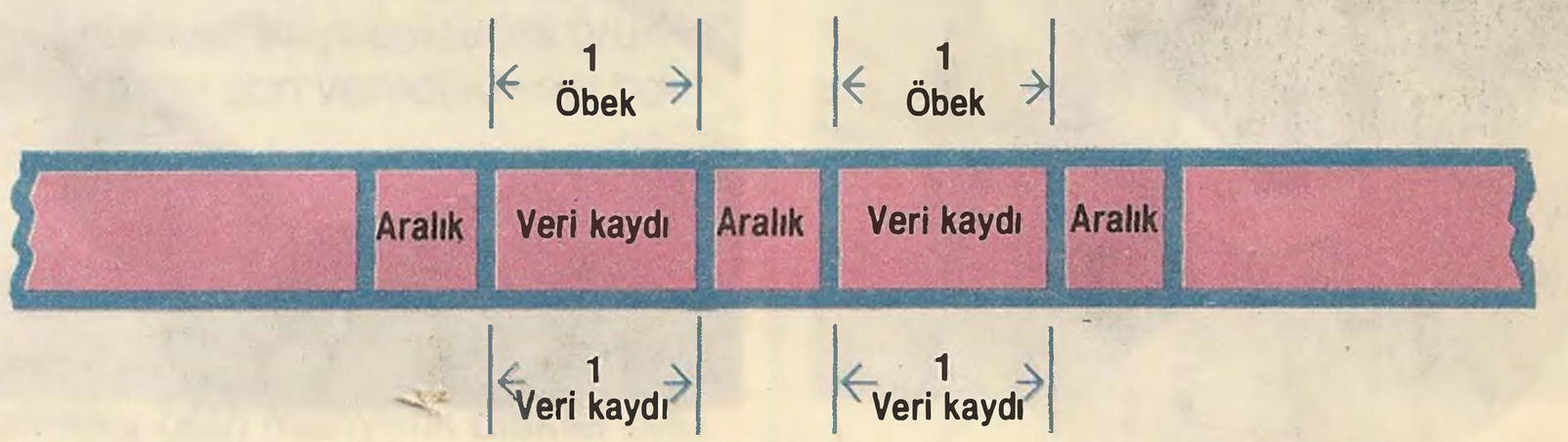
* Bu yöntem, ASCII kodunda kullanılan alış sırasında kontrol yöntemine çok benzemektedir (tamlama biti, bkz. s.117'den s.119'a).

kanal üzerindeki noktalar da benzer bir biçimde sınanır (boyuna tamlama).

Bir manyetik şerit üzerine kaydedilebilen karakter sayısı, şeridin uzunluğuna (180, 360, 720 veya 1080 metre) ve birim uzunluğuna kaydedilebilen karakter miktarına (buna **kayıt yoğunluğu** denir) bağlıdır.

En çok kullanılan kayıt yoğunlukları, inç (yaklaşık 2.5 cm) başına 256, 512, 800, 1600 ve 6250 karakterdir. Bu rakamlar, şeridin kuramsal kapasitesini hesaplamamıza olanak verir. Buna göre inç başına 800 karakterlik bir yoğunlukla kaydedilmiş olan 720 metre uzunluğundaki bir şerit, 24 milyon karakterlik bir kuramsal kapasiteye sahiptir.

MANYETİK ŞERİT ÜZERİNDE VERİ KAYDI ŞEMASI



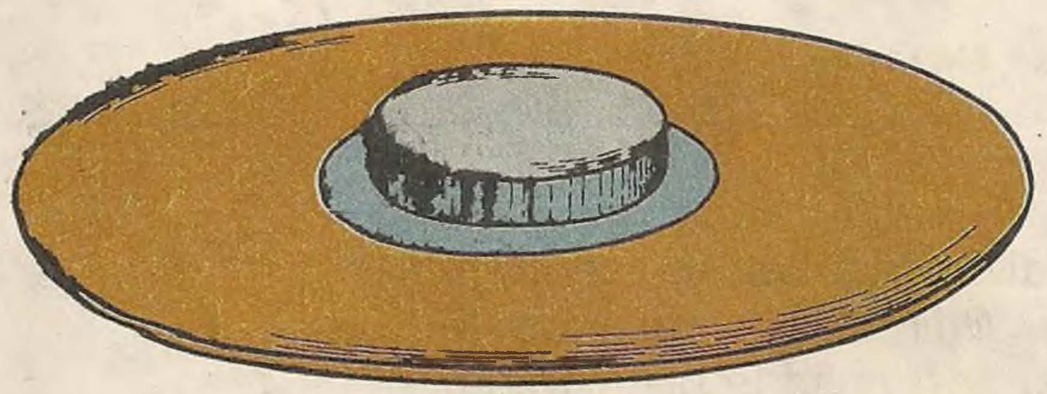
Aslında, pratikte karakterlerin şerit üzerindeki kayıtları sürekli değildir, çünkü bunlar **öbekler** veya **kayıt paketleri** halinde gruplanırlar ve her öbek ötekinden boşluk ya da **aralık** adı verilen, kullanılmayan şerit parçalarıyla ayrılır. Genel olarak, öbeklerin uzunluğu verilerin düzenine ve bilgisayarın merkez belleğinin kapasitesine bağlıdır. Öbek arası boşluğun (yani aralığın) uzunluğuna gelince, bu da şerit biriminin tipine bağlıdır ve oldukça kısadır (örneğin 0.6 veya 1.2 inç).

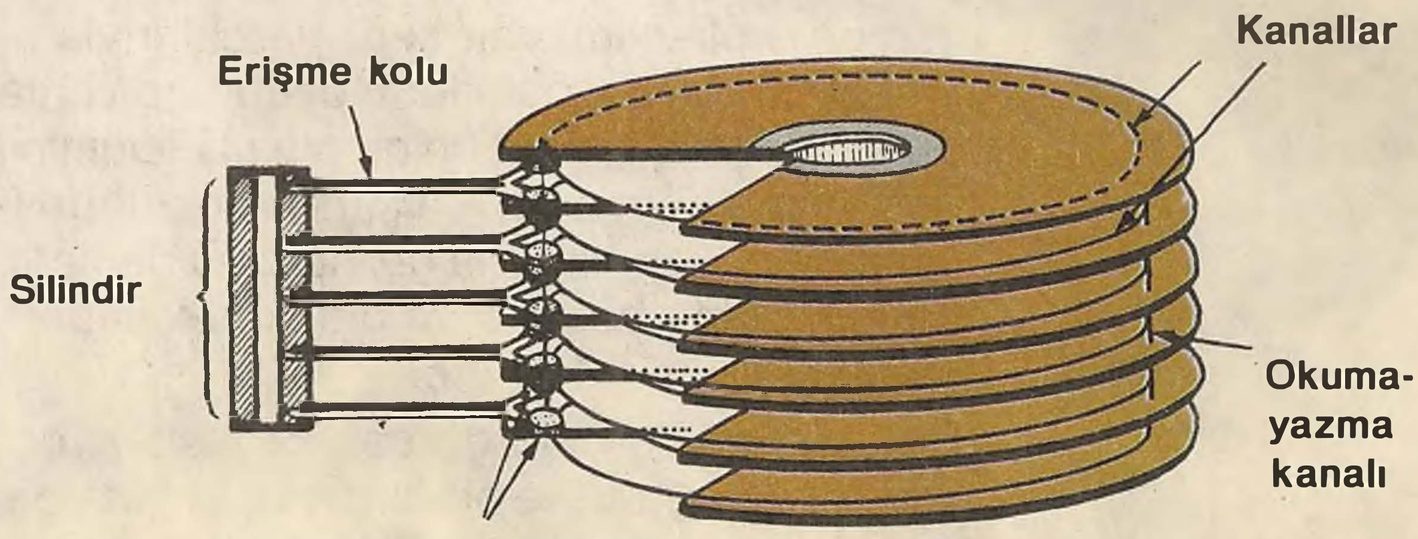
Bir manyetik şeritin gerçek kapasitesi öbeklerin uzunluğuna ve aralıkların sıklığına bağlıdır. Belli bir öbek ne kadar çok karakter içerebiliyorsa, kapasitesi o kadar yüksek demektir.

[redacted] mekanizmasının (drive) dönüşü [redacted]nda, iki yüzeyininde dengelenmesi, diske [redacted]el bir kararlılık sağlar.

Sayısal doğrulama (kalitenin bir osiloskop yardımı ile kontrolü) katmerli disk ile bilgisayarın birbirlerine tam bir uyum halinde çalışmasını sağlar.

KATMERLİ DİSK

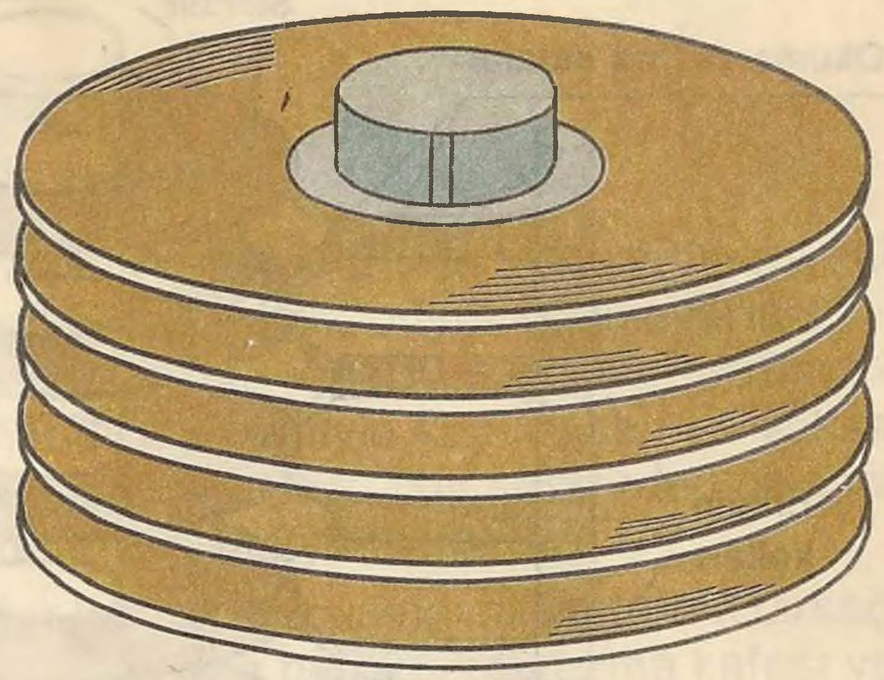




Daha önce de söylediğimiz gibi, şerit birimleri bilgisayarın çevresel birimleridir. Bilgisayar ile şerit birimleri arasındaki alışveriş, kablolar aracılığıyla, büyük bir hızla gerçekleşir. Veri taşıma hızı saniyede 20.000 ile 60.000 işaret arasında değişmekle birlikte 120.000, hatta 320.000 işarete ulaştığı da olur. Şerit kullanımının güvenliğini artırmak amacıyla, şeritlerin başında ve sonunda yaklaşık 3-4 metrelik boşluklar bırakılır. Bu, şeritin başında hareket geçirme mekanizmasının çalışmasını, sonunda da makaraların tamamen boşalmasını önlemeye olanak verir. Şeridin kullanılan kısmının başı ve sonu, **şeridin ön göstergesi ve son göstergesi** denilen yaklaşık 25 mm'lik alüminyumdan işaretlerle belirtilir. Bunlar, şeridin manyetize edilmemiş yüzüne yapıştırılan ve **etiket** (sticker) adı verilen yansıtıcı küçük levhalardır. Etiketlerin işlevi, şerit birimine yerleştirilen fotoelektrik hücrelerin yazma ve okuma işlemlerine nerede başlayacaklarını (yükleme noktası) ve nerede son vereceklerini belirlemektedir.

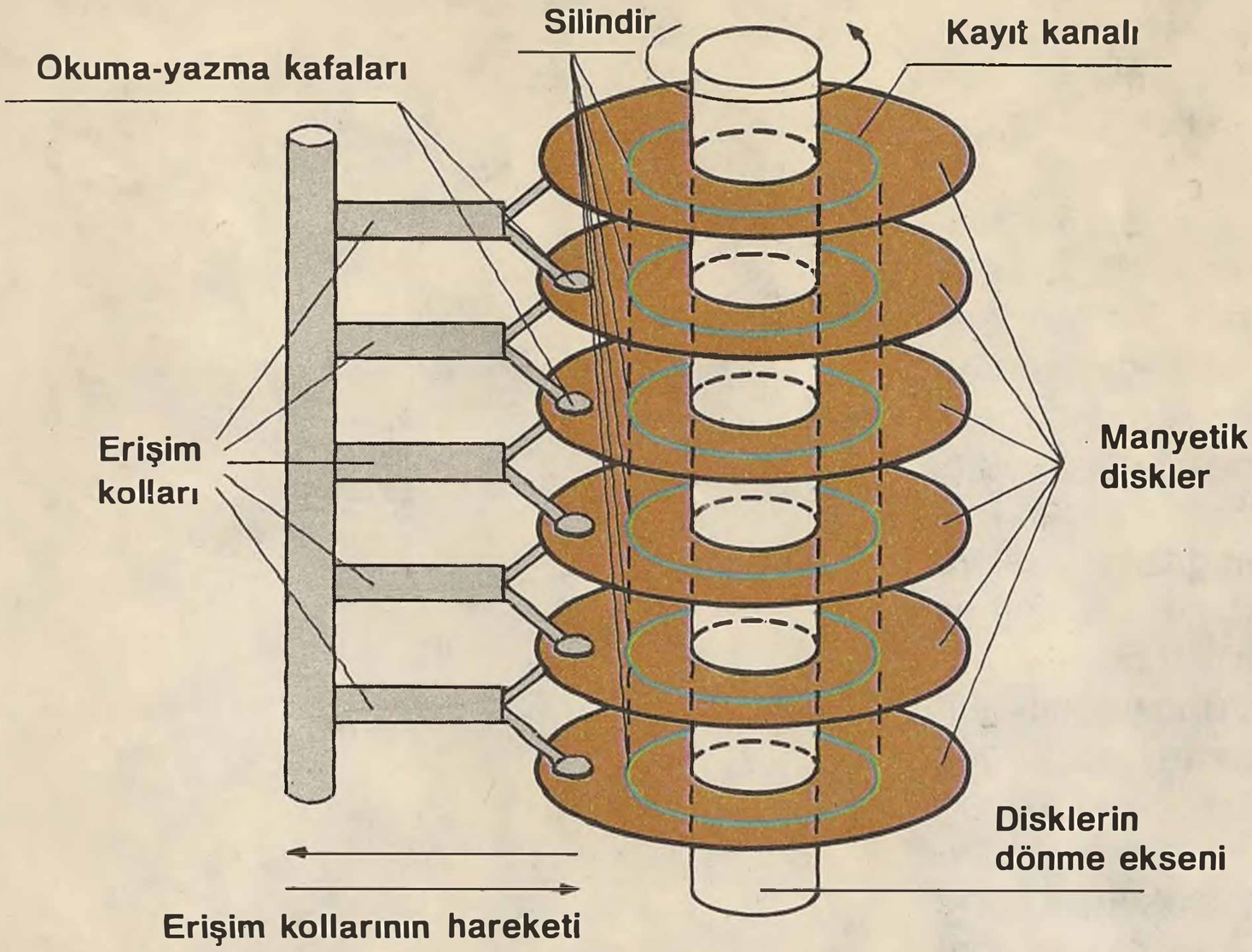
Manyetik Diskler

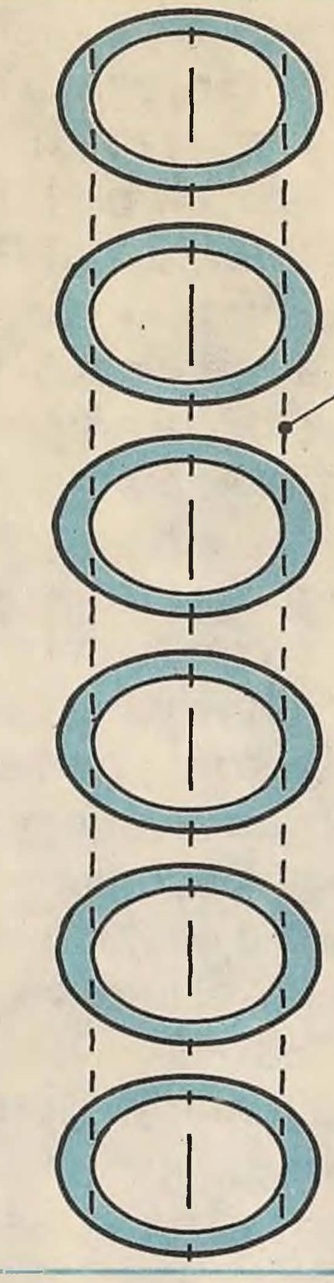
Büyük sayıda veriyi saklamak üzere doğrudan erişimli yığın bellekler olan manyetik diskler de kullanılır. Manyetik şeritlere kıyasla bilgiye erişmede daha yüksek bir hız sağlarlar. Manyetik disk, iki yüzü manyetik bir örtüyle sıvanmış, genellikle alüminyumdan yapılma ince bir diskten oluşur (bkz. s.226'daki şekil).



Katmerli disk büyük sistemlerde doğrudan erişimli yığın bellek işlevi görür.

KATMERLİ DİSKİN ŞEMASI

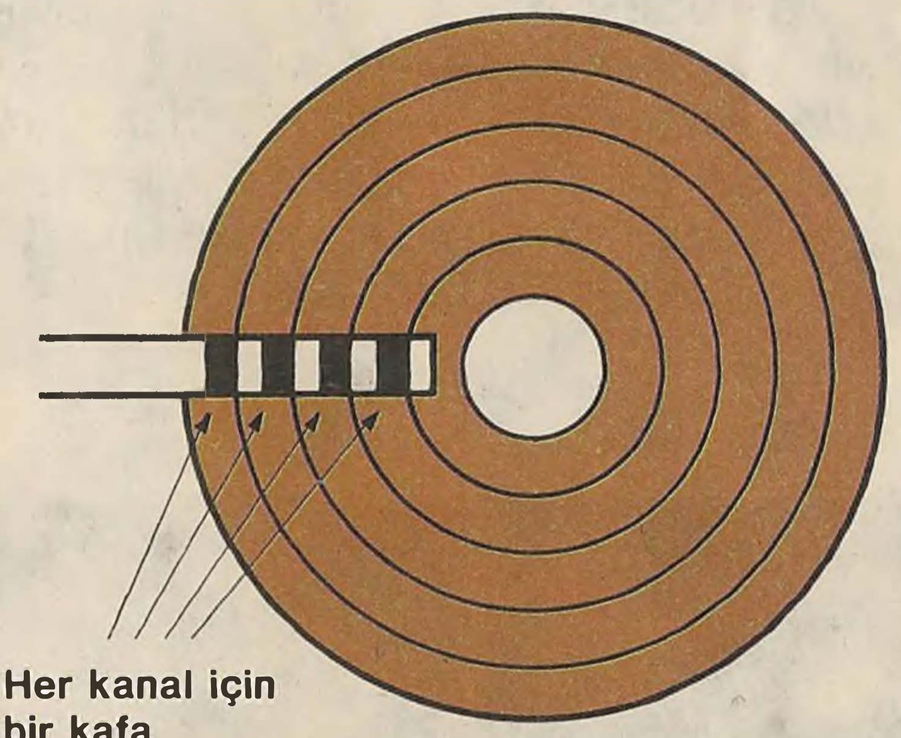


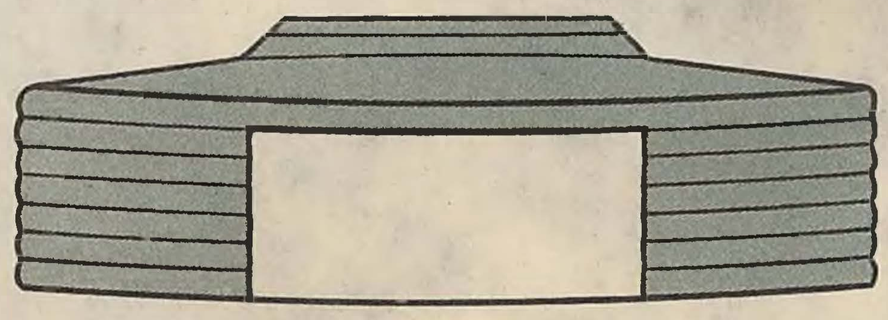


Birçok disk bir merkez eksen boyunca üst üste getirilip aynı hızda döndürülebilir. Böyle bir bütüne **katmerli disk** (disk-pack) denir (bkz. şekil). Bir katmerli disk oluşturan disklerin aralarında belirli bir mesafenin bulunması gerekir. Bu mesafe, okuma-yazma kafalarının disklere ulaşmasını sağlar. Kafalar önden arkaya doğru yatay bir hareketle kanalları tararlar. Okuma ve yazma kafaları geri çekilebilir cinstense, katmerli diski oluşturan diskleri değiştirmek mümkün olur. Buna da yerinden oynatılabilir diskli birim denir. Diskler, birbirinden, okuma ve yazma kafalarının yer aldığı boşluklarla ayrılırlar.

Bilgiler, şeritlerdeki gibi aynı fizik ilkelerine göre, iç içe kanallara kaydedilirler. Her disk de çeşitli kesimlere ayrılmıştır. Kanallar bu kesimlere yerleştirilir. Her diskte yüzlerce kanal bulunur. Kanallar birbirinden, içine hiçbir verinin kaydedilmediği aralıklarla ayrılır.

Bir katmerli diskte ise, kollardan oluşan tek bir mekanizma, okuma-yazma kafasını üstteki diskin alt yüzünün ve alttaki diskin de üst yüzünün karşısına getirir. Bu mekanizmadaki kollar birlikte hareket ettiğinden, bir dikey hat üzerinde sıralanmış kafaların her biri her diskin aynı kanalının önüne yerleşmiş olur. Bu kanalların tümü bir **silindir** oluşturur (bkz. yukarıdaki şekil). Böylece kolu oynatmadan bir silindire ait bütün kanalların üstüne aynı anda yazmak veya bunları okumak mümkün olur





ki, bu da erişim süresini oldukça kısaltır. Bu süre erişim kolunun hareketini gerektirmeyen sabit kafalar kullanılarak daha da kısaltılabilir (karşıdaki şekil). Disklerin dönmesi, bir kesimin içeriğinin, okuma-yazma kafaları bu kesimin üzerinden (veya altından) geçmedikçe okunamamasını ve yazılamamasını sağlar. Öyleyse belirli bir kesimdeki verilere erişim süresi şu ögelere bağlı olarak değişir:

- Kafanın aranılan kesimin bulunduğu kanala yerleştirilmesi için gereken süre,
- Kesimin, kafanın altında ya da üstünde bulunmasına bağlı olan bekleme süresi,
- Verileri okumak ve yazmak için gereken süre.

Zaten bu süreler on ile yüz milisaniye arasında oynayan çok kısa sürelerdir.

Bir birimin kapasitesi 500 ile 900.000.000 karakter arasında değişir.

Bilgilerin disk birimi ile bilgisayar arasındaki alışverişi saniyede birkaç bin karakterden 1.800.000'e ve hattâ 2.000.000'a kadar değişir. Sabit diskler, genellikle, üzerinde erişim kolları için delikler bulunan plastik kaplarda bulundurulur (s.228 üstteki şekil).

Manyetik Davullar

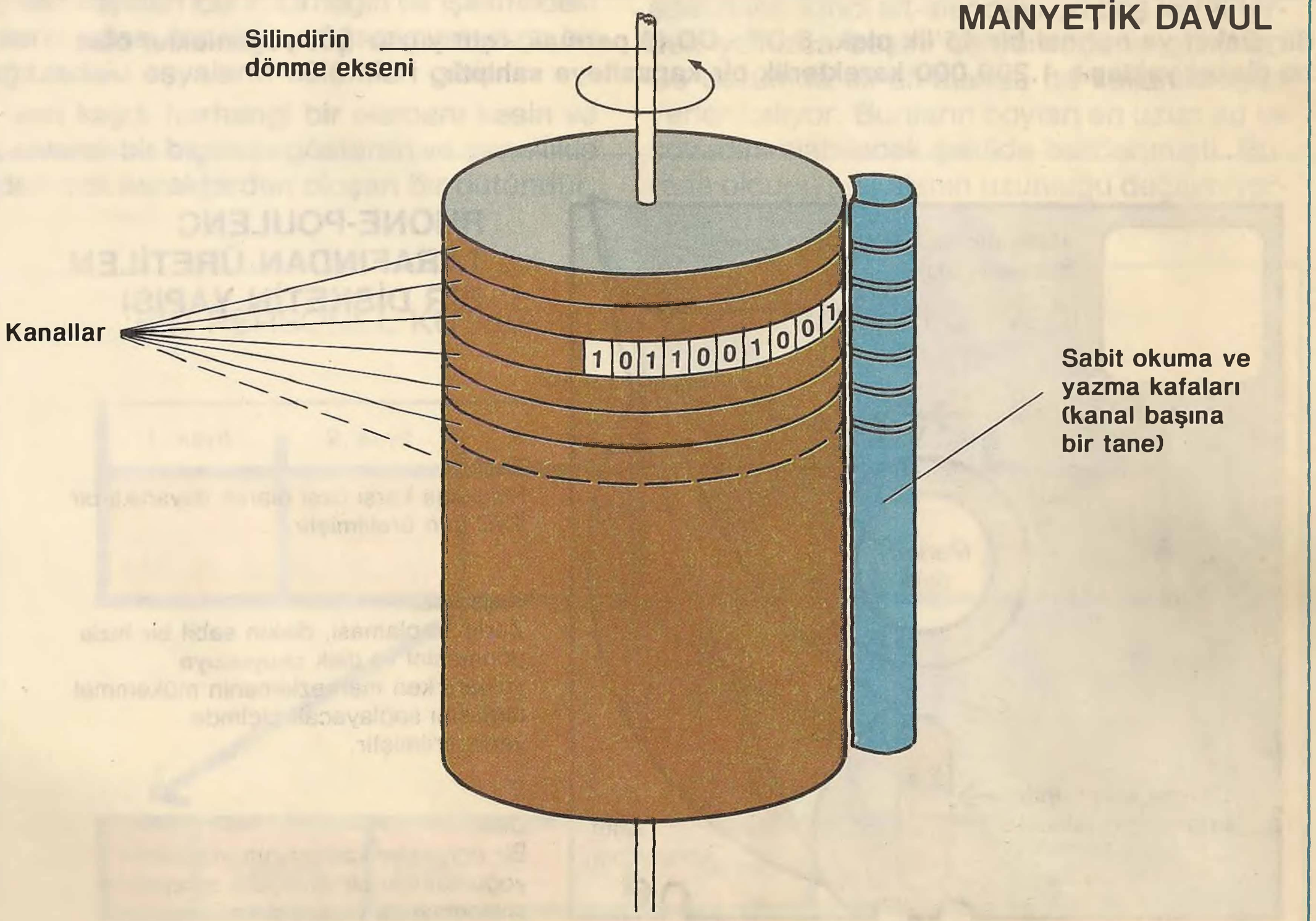
Yüksek maliyetlerinden dolayı, manyetik davullar, daha çok, büyük bilgisayarlarda kullanılır. Bunlar yanal yüzleri bir ferromanyetik madde katmanıyla kaplı, sabit ve yüksek bir hızda dönen, metal bir silindirden oluşan doğrudan erişimli belleklerdir.

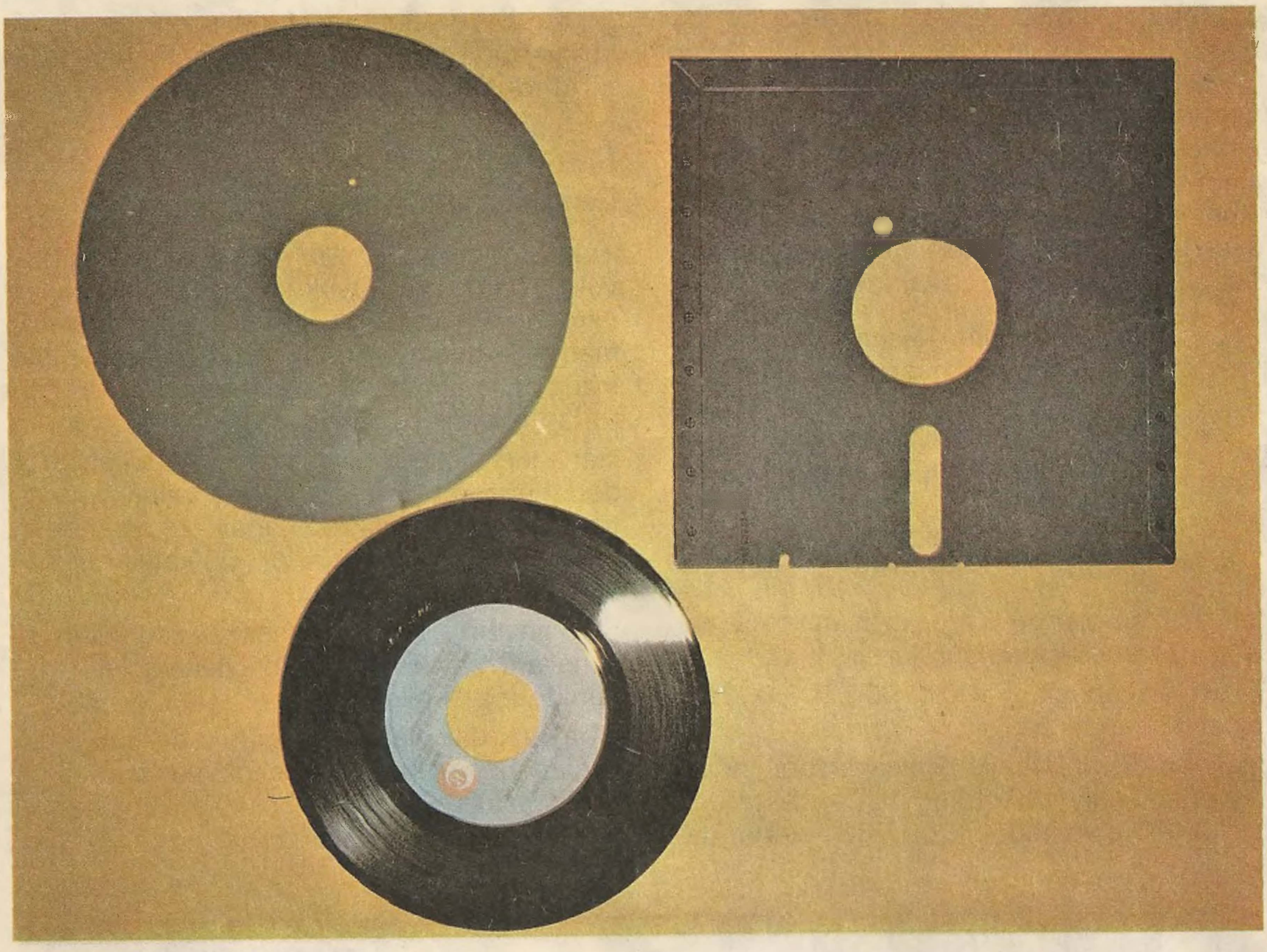
Her kanalın karşısına milimetrenin birkaç yüzdebiri arayla bir okuma-yazma kafası yerleştirilmiştir. Dolayısıyla kanal sayısı kadar okuma-yazma kafası vardır. Bu kafalar sabittir.

Bir manyetik davulun kapasitesi, uzunluğuna, kanalların sayısına ve kaydedilen verilerin yoğunluğuna bağlıdır.

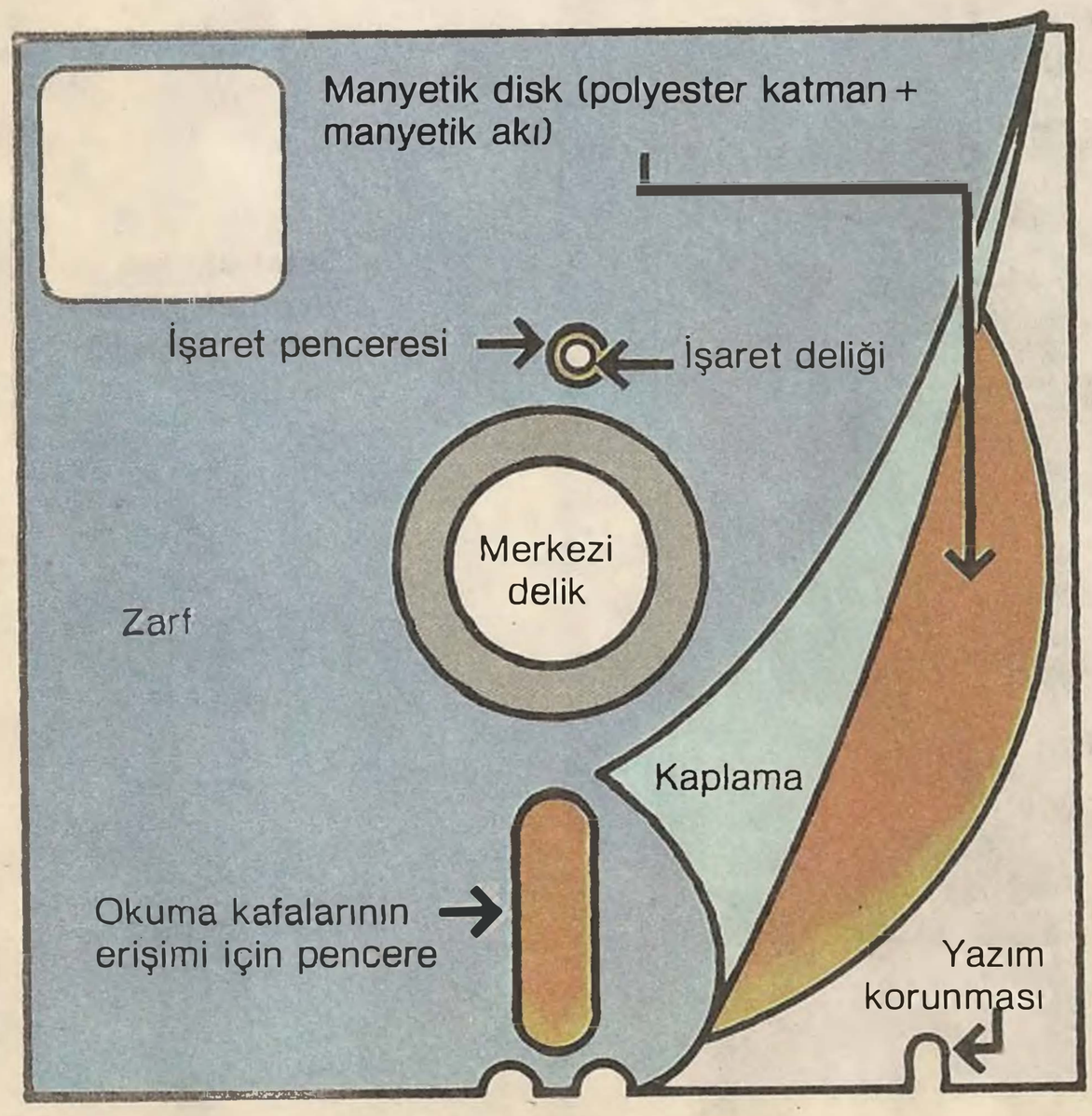
Manyetik davullarda saklanmış herhangi bir veriye erişme süresini hesaplamak için, okuma süresini ve ilgili kanal bölümüne erişme süresini toplamak gerekir.

MANYETİK DAVUL





Bir disket ve normal bir 45'lik plak. 8:DF - DD (8 parmak - çift yüzlü- çift yoğunluklu) olan bu disket yaklaşık 1.200.000 karekterlik bir kapasiteye sahiptir.



RHONE-POULENC TARAFINDAN ÜRETİLEN BİR DİSKETİN YAPISI

Zarf:
Sıcaklığa karşı özel olarak dayanıklı bir PVC'den üretilmiştir.

Kaplama:
Zarfın kaplaması, diskin sabit bir hızla dönmesini ve disk okuyucuya yüklenirken merkezlemenin mükemmel olmasını sağlayacak biçimde yerleştirilmiştir.

Disk:
Bir polyester katmanının yüksek yoğunluktaki bir manyetik sıvayla sıvanması ile yapılmıştır.

Disketler

Disketler (floppy disk) disklerle aynı ilkeye dayanırlar, ama bunlar daha ucuz ve daha kullanışlıdırlar.

Esnek bir maddeden yapılan disketler, bir zarfla korunurlar. Bu zarfta okuma-yazma kafalarının erişimine izin veren küçük bir pencere vardır.

Disketler de, belli kesimlere yerleştirilmiş kanallara bölünmüştür. Sağladıkları olanaklara göre disketler çeşitli tiplere ayrılırlar:

1 / Tek ya da iki yüze kayıt
2 / Basit ya da çifte yoğunluklu kayıt

Bu ögelerin çeşitli uyuşumlarına göre, bir disketin kapasitesi 80.000 ile 1.200.000 sekizli (karakter) arasında değişir. Bundan başka, aynı yükleyiciye birden fazla disket yerleştirilebilir. Disketler küçük ve kişisel bilgisayarlarda yaygın olarak kullanılmaktadır. En sık kullanılan disketler 5.1/4 ve 8 inç boyutlarındadır.

Kütükler

Yığın belleklere kaydedilen verilerin bütünü bir kütük **(file)** oluşturur. Bir kütük genellikle türdeş veri kayıtları içerir. Örneğin bir işletmedeki işçilerin adları, bir okulun öğrencileri ya da bir mağazadaki eşyaların bölümleri gibi.

Bir veri kaydı herhangi bir elemanı kesin ve tek anlamlı bir biçimde gösteren ve genellikle birden çok karakterden oluşan bir bütündür.

Örneğin:

- **Bir kişi:** Atik Engin
- **Bir sayı:** 235
- **Bir kısaltma:** A2

Öyleyse veriler:

- Yalnız **harfler** ve boşluklardan oluşan veriler
- Yalnız sayılardan oluşan **sayısal** veriler
- Harflerden, sayılardan ve boşluklardan oluşan **harfsayısal** veriler

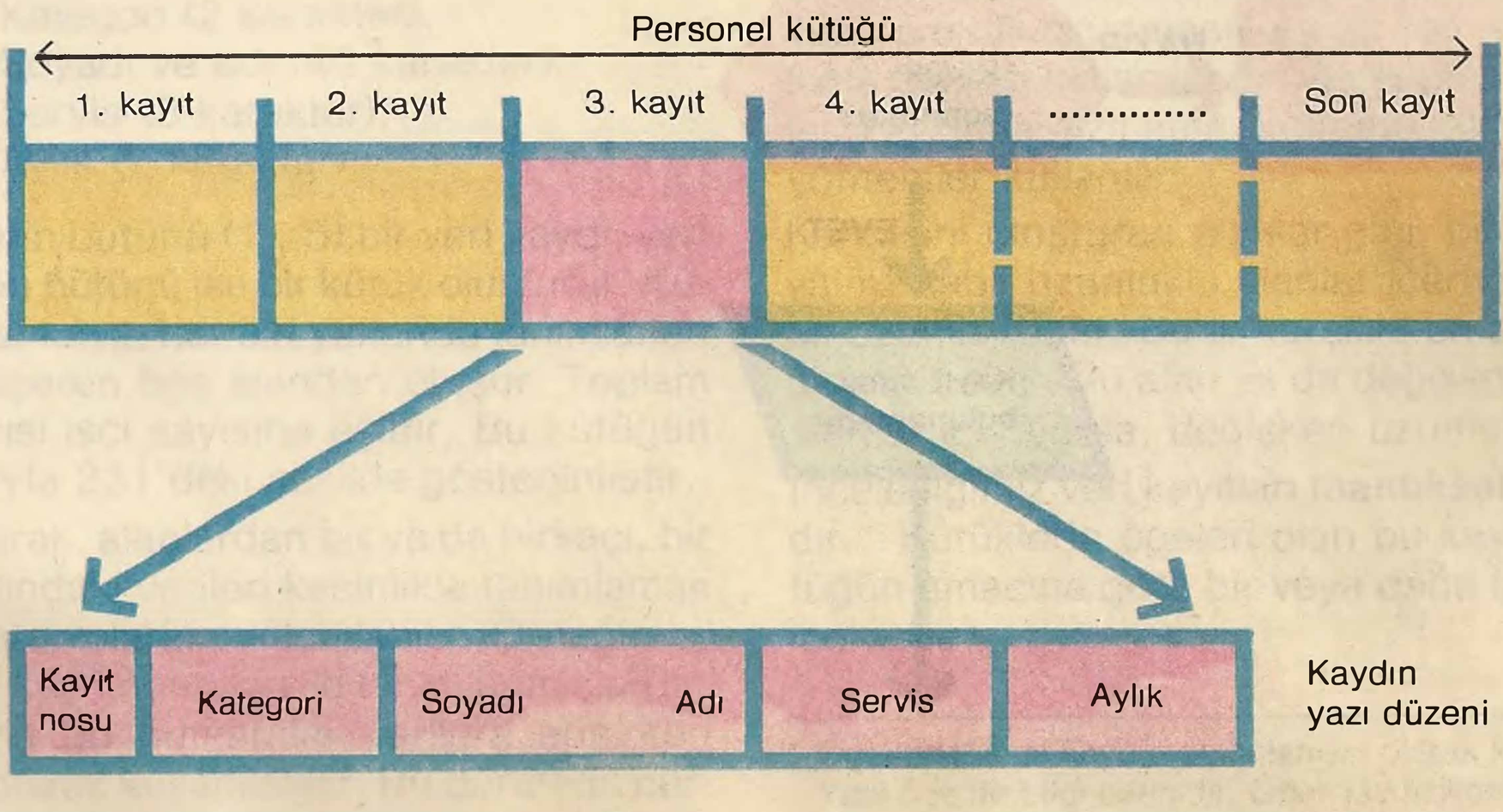
olmak üzere üçe ayrılırlar.

Bir verinin bellekte kapladığı yere **alan** (field) denir ve genellikle her alan bir adla anılır.

40 karakterlik ad	
25 kar.soyadı	15 kar.ad

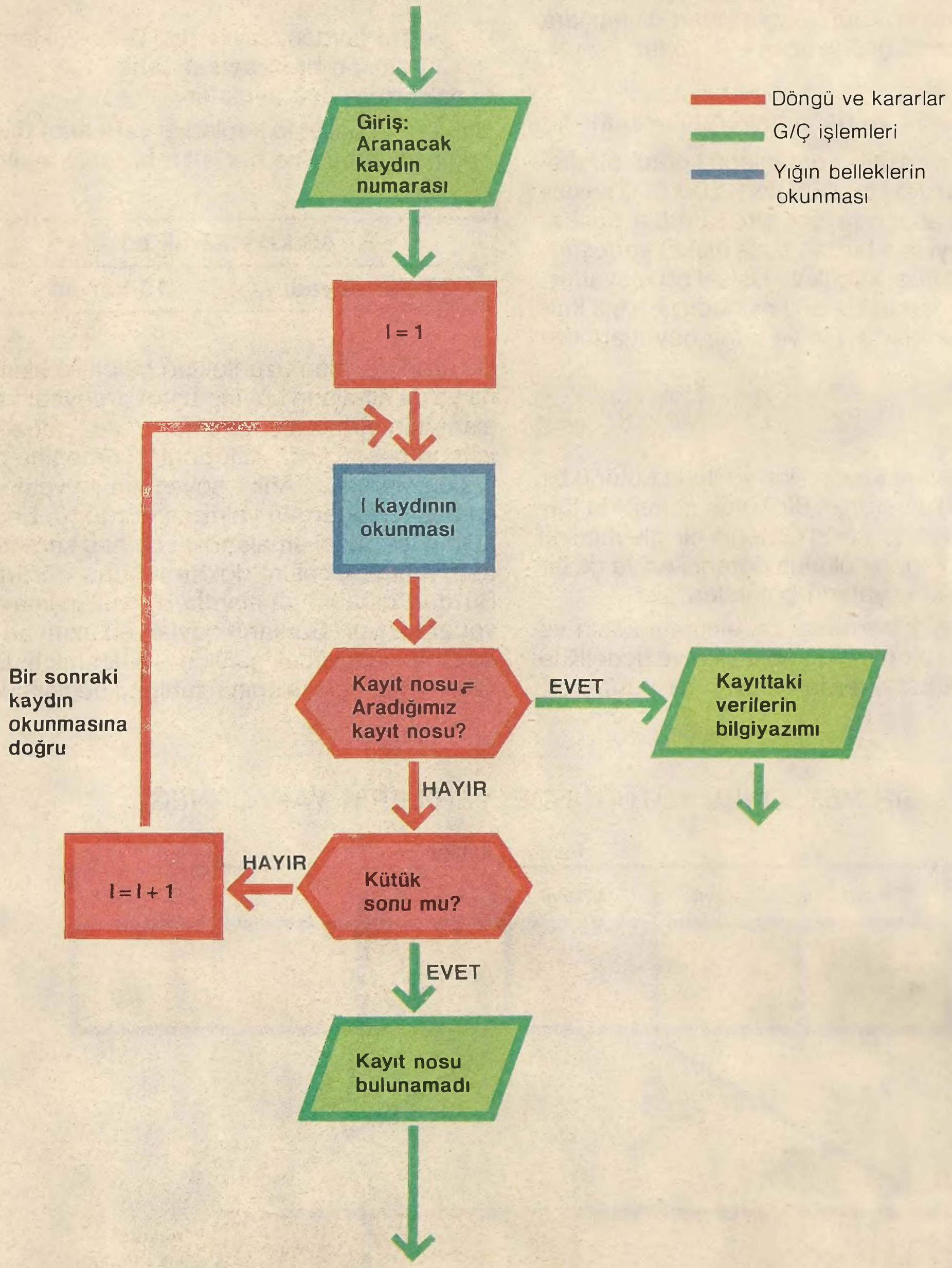
Bu örnekte, alan uzunlukları belirli iki elemana ya da alt-alana bölünmüştür: ''soyadı'' alt-alanı için yirmi beş karakter, ''ad'' alt-alanı için on beş. Eğer ''Atik Engin'' örneğine dönecek olursak, ''Atik'' soyadının mevcut yirmi beş karakterden yalnızca dördünü, Engin adının ise ikinci alt-alandaki on beş karakterden yalnızca beşini doldurduğunu görürüz. Bu durumda iki alt-alanda da kullanılmayan yerler kalıyor. Bunların boyları en uzun ad ve soyadını alabilecek şekilde belirlenmişti. Burada olduğu gibi alanın uzunluğu değişmiyor-

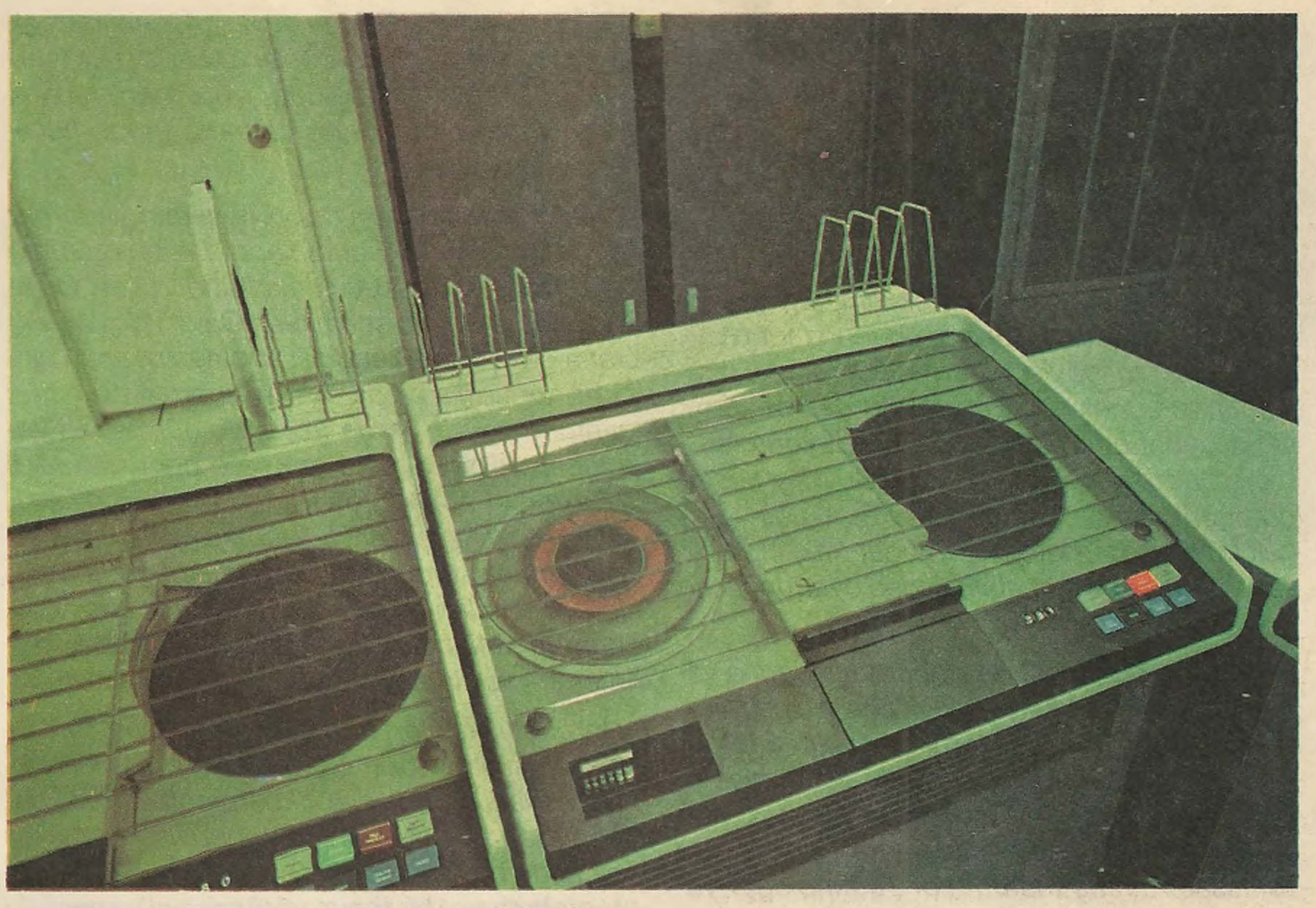
BİR PERSONEL KÜTÜĞÜNDE VERİLERİN YAPI ÖRNEĞİ



sa, buna **sabit uzunluklu** alan adı verilir. Buna karşılık, kullanılan bellek alanı karakter sayısına bağlı ise buna da **değişken uzunluklu** alan denir. ''Atik Engin'', 10 karakter (boşluk

ANAHTARLI ARAMA İÇİN AKIŞ ŞEMASI





Bilgisayara ilişkin manyetik kayıt cihazları en mükemmel kayıt cihazları arasında yer alırlar.

dahil), ''Bayındır Kadir'', 14 karakter (boşluk dahil).

Belli bir veri hakkındaki bütün bilgileri içeren alanlar kümesine, **veri kaydı** (record) denir. Şimdi bir şirketin personel yönetimi hakkında temel bilgileri içeren bir kütük oluşturmak istediğimizi düşünelim.

Kaydedilecek veriler şunlardır:

1 / Kayıt numarası (3 karakter),
2 / Kategori (2 karakter),
3 / Soyadı ve adı (40 karakter),
4 / Servis (2 karakter),
5 / Aylık (7 karakter).

Bu bilgilerin bütünü (1...5) bir veri kaydı, veri kayıtlarının bütünü ise bir kütük oluşturur. Kütükteki her kayıt, her biri yukarıda tanımlanan 5 bilgiyi içeren beş alandan oluşur. Toplam kayıt sayısı işçi sayısına eşittir. Bu kütüğün yapısı sayfa 231'deki şekilde gösterilmiştir.

Genel olarak, alanlardan bir ya da birkaçı, bir veri kaydındaki verileri kesinlikle tanımlamak için **anahtar** olarak kullanılabilir. Örneğimizde işçiler kayıt numaları ile tanımlanmışlardır. Dolayısıyla bu numaralar verilere erişirken anahtar olarak kullanılabilir. Bu durumda personelden herhangi bir kişi ile ilgili bilgileri bulmak için, kayıt numarasına gitmek yeterli olacaktır. Bilgisayar kütüğü birinci kayıttan başlayarak okurken, her veri kaydının numarasını kullanıcının girdiği kayıt numarası ile karşılaştıracaktır. Bu arama iki kayıt numarası arasında bir eşitlik bulununcaya kadar devam edecektir. Bu noktada bilgisayar kütükteki bilgileri sergileyerek işini sona erdirecektir. Bu basit programın akış şeması sayfa 232'de gösterilmiştir. Bu arama yöntemi (bütün kayıtların sıralı okunması) ancak veri sayısı çok fazla değilse kullanılabilir. Daha büyük kütükler için daha hızlı ama o ölçüde de karmaşık yöntemler kullanılır.

Kendisini oluşturan alanlar gibi, bir kayıt da yalnız sabit uzunluklu alanlar içeriyorsa, sabit uzunlukludur. Buna karşılık, en az bir değişken uzunluklu alan ya da değişken sayıda alanlar içeriyorsa, değişken uzunlukludur.

İncelediğimiz veri kayıtları **mantıksal** kayıtlardır.* Kütüklerin ögeleri olan bu kayıtlar, kütüğün amacına göre bir veya daha fazla bilgi içerebilir.

---

* Öge mantıksal kaydın eş anlamlısı olarak kullanılıyor. Yani öge bir bilgi birimidir. Öbek ise fiziksel kaydın eş anlamlısıdır, yani fiziksel bir birime işaret eder.

Video Oyununu İcad Eden Kişi

Bell, Edison, Fermi: Bu adlar size ne hatırlatıyor? Peki ya Higinbotham adı? Hayır mı? Her neyse, Higinbotham, 1958'de bir gün osiloskobu oluşturarak video oyununu yaratan kişidir. Bu sonuca ulaşabilmek için dağ gibi bir kitap devirdiğimi ve video oyununun tarihi hakkındaki fikrimi böylece berraklaştırdığımı böylece itiraf etmeliyim. Eğer 1958'den önce bir video oyunundan bahsedildiğini duyduysanız bana da haber verin. Aksi halde Willy Higinbotham'a tarihte düşen yeri kabul edin. Birçok büyük şirket istedikleri kadar hakkında savaş vermeye devam etsin, video oyununu icad eden odur.

1950 yıllarında araştırma laboratuvarlarına yapılan ziyaretler oldukça sıkıcıydı: Size çalışmalarının birkaç yönünü sergileyen bir iki fotoğraf gösterilebilirdi, o kadar. İşte bu yüzden fiziğe olan yeteneğini Cornell Üniversitesi'nde ve elektronikle ilgili yeteneklerini de MIT'de (Massachusetts Institute of Technology) yeni keşfetmiş olan Willy, bunları daha da geliştirmeye karar verdi. Kendi kendine, ''Ziyaretçiler neden ekran üzerindeki küçük oyunlarda yer almasın? Bu şekilde aygıtlara dokunma ve kollarla oynayarak eğlenme olanağı bulmuş olurlar.'' diye düşündü. Düşünür düşünmez uyguladı: Willy ile arkadaşları bir osiloskobun beş parmak enindeki ekranı üzerinde, bir tenis maçının akışını yarattılar.

O devirde, ilk sayısal bilgisayarlar ortaya çıkmıştı ve Willy'nin çalıştığı araştırma birimi de bunlardan bir tane yapmıştı. Willy, ilk aşamada küçük oyunu için, ON/OFF (açık-kapalı) darbeleri ile çalışan sayısal bir bilgisayar yerine bilgileri değişik değerli gerilimler olarak gösteren bir benzeştirmeli bilgisayardan yararlandı. Bilgisayar elektromekanik istasyonlar, potansiyometreler, dirençler, kondansatörler ve işlemsel yükselticilerden oluşan programlanamaz bir bütüne bağlıydı. Willy'de çarçabuk oluşturulan bu ''yamalı bohçanın'' pek zarif olmadığını, ama gayet iyi çalıştığını söylüyordu.

Ekran bir tenis kortunu yandan gösteriyor ve tersine çevrilmiş bir ''T'' de ağı simgeliyordu. Oynamak isteyen kişiler, üzerinde bir düğme ve bir kol olan bir kutu kullanıyorlardı. Düğme vuruşun açışını yönetiyor, kol da topa vurmayı sağlıyordu. Bunun dışında, oyun, gerçek oyundaki bazı fiziksel ögeleri de dikkate alıyordu: Top ağa takıldığında, yere çarpmasına oranla daha az zıplıyor ve daha yavaş hareket ediyordu.

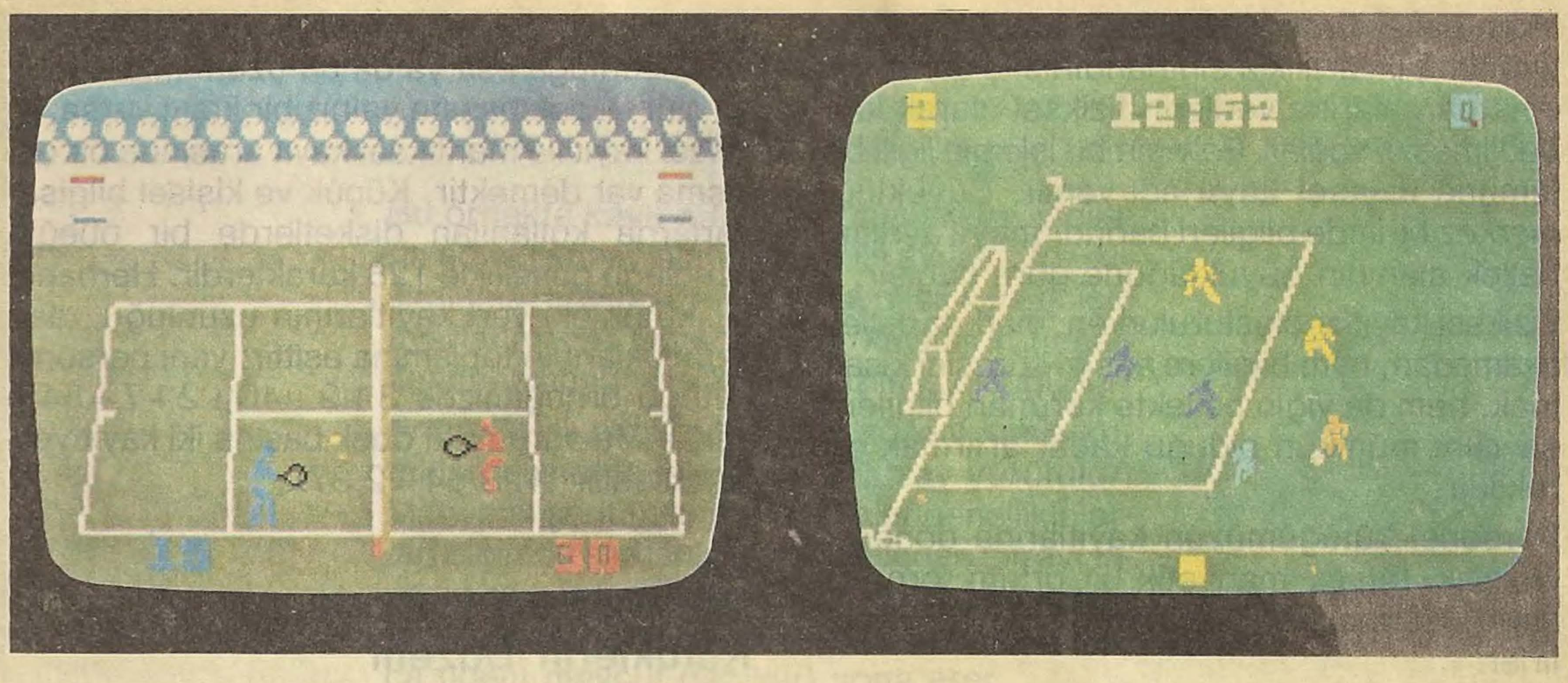
Basit olmasına karşın, bu heyecan verici bir oyundu. Higinbotham'ın arkadaşları duydukları çocuksu coşkuyu hâlâ hatırlıyorlar. O zamanlar benim de şefim olan Dave'i gözleri ekrana çakılmış ve kendini tamamen oyunun akışına kaptırmış bir halde hatırlıyorum. Sahanın sınırları ve top ekran üzerinde durmadan yeniden çiziliyordu. Bu tempo o denli yüksekti ki, resmin aradaki titreşmesi gözle fark edilmiyordu. Bu yöntem bugün de kullanılmaktadır. Ancak, topun hareket edişini göstermek için kullanılan sistem alınmamıştır. Higinbotham'ın kullandığı ilkeyi teknik ayrıntılara girmeden açıklamaya çalışacağım.

Hareketli çizgileri görselleştirmek için bir osiloskop kullanılabilir. Görünecek olan ışıklı noktanın (yani topun) x ve y koordinatlarındaki değeri, daha girişte gerilim değeri ile orantılı bir biçimde belirleniyordu. Bu ilkeden hareketle, Higinbotham, bir topun yörüngesini ve yerdeki sıçrayışını taklit etmek için işlem güçlendiricileri kullandı. Bir darbe anında, bir istasyon harekete geçiyor ve diğer güçlendiricinin kutbunu tersine çeviriyordu. Böylece, top sıçrama izlenimi verecek biçimde geri hareket ediyordu. Basit ama etkin olan bu yöntem, topun fileye değip değmediğini belirlemeye de yarıyordu çünkü bu durumda top, değişik bir hızla (daha düşük) sıçrıyor ve böylelikle oyuna gerçekçi bir görüntü verilmiş oluyordu. Topun hızının rüzgârın hızı yüzünden gittikçe azalması ise 10 megaomluk bir direnç ile doğrudan taklit ediliyordu.

Ağın yüksekliği ve sahanın uzunluğu ayarlanabiliyor ve bir düğme oyunculara servis köşesini seçme olanağı tanıyordu. Topa vuruş kutunun üzerindeki bir kolla oynanarak gerçekleştiriliyordu. Ama gene de, doğru anda hareket etmek ve vuruş açısını iyi ayarlamak gerekiyordu. Aksi halde, top karşı sahaya ulaşmıyordu.

Bu ayrıntılar o denli iyi incelenmişti ki, oyun öylesine başarılı olmuştu ki yıl boyunca Brookhaven laboratuvarlarının merkezi ve tüm ziyaretçilerin can alıcı noktasını oluşturdu. Derken bir gün birdenbire kaldırıldı.

Hükümetin satın aldığı yirmiden fazla paten-



tin sahibi olan Willy'ye nasıl olupta icadının patentini satın almadığını sordum. ''Biz gerçi pek eğlenmiştik buluşumla ama yönetimi ilgilendirmedi'' diye cevap verdi, ''Fena da olmadı; aksi halde, bugün yalnızca hükümetin lisansı altında video oyunu üretilebilecekti''. Bu düşünce Willy'i eğlendirmiş gözüküyordu.

Higinbotham'ın zamanında hükümet bu oyunlara fazla ilgi göstermemiş olabilir, ama yanlışını kısa zamanda anladı. Bugün, özellikle Amerikan ordusu personel eğitiminde videolu savaş oyunlarından yoğun bir biçimde yararlanıyor. Bu oyunlarda geleceğin komutanları, bilgisayarın onlar için hazırladığı gerçek savaşı andırır durumlarda karar verme yeteneklerini geliştiriyor.

Ayrıca nişancılık eğitiminde de video oyunlarından yararlanılıyor. Son gelişme ise bilgisayarlara tankları temsil ettirmek. Küçük bir odacıkta oturan tankçı, ekranda hedefleri görüyor ve oyun kolu ve düğmeler sayesinde namluyu ''ateşleyebiliyor''. Böylelikle bir yandan personelin hünerleri gelişirken, gerçek hiçbir mühimmat harcanmadığından, milyonlarca dolarlık bir tasarruf sağlanmış oluyor.

Buna karşı Magnavox şirketi ise daha o zamandan bunu fazla gülünç bulmamış olsa gerek, çünkü video oyununun hakları milyonlarca dolar demek. Ve bu şirket, bir topun sıçramasına dayanan video oyunları için patent talebinde bulunuyor... Higinbotham'ın bu işten kazanç sağlamaya niyeti olmasa da ününü ve kişisel prestijini korumaya karar vermiş durumda.

Şimdi bu kişiyi biraz daha yakından tanıyalım.

İkinci Dünya Savaşı patladığında, Higinbotham daha yeni Cornell Üniversitesi'nden fizik diploması almıştı. MIT'den ışıma konusundaki araştırmalara katılması için bir davet alınca, büyük başarı vaad eden ve o zaman Radio Detecting and Ranging adıyla anılan yeni bir teknik üzerinde çalıştı. Bu ad size bir şey hatırlatmıyorsa, -RADAR- kısaltması mutlaka hatırlatacaktır. Willy daha sonra Manhattan District Project'e katılarak, fizikçilik yeteneklerini büyük umut beslenen yeni bir teknolojide kullandı. 1945'de Elektronik Bölümü'nün başındaydı ve birkaç yıl sonra ilk atom bombasının patlamasından önceki son milisaniyeleri sayan, bir zaman ayırıcı devre tasarlıyordu.

Bomba Los Alamos'da patladığında, Willy oradan 40 km kadar uzakta bulunuyordu. Böylece patlamanın değişik evrelerini gözleme olanağını buldu. Bütün gözlemcilerle birlikte arabalara binip büyük bir sessizlik içinde geri döndüğünü söyledi son olarak. ''Kimsenin söyleyecek hiçbir şeyi yoktu''.

Bunu izleyen iki sene boyunca Willy, Amerikan Araştırmacıları Federasyonu'nun sekreterliğini yürüttü ve Kongre ile araştırmacılar arasında aracı sıfatı ile nükleer silahların sınırlandırılması konusunda etkin olarak savaştı. Bugün Brookhaven Laboratuvarları'nın başkanı olan Willy, arkadaşları ile birlikte nükleer korunma sistemleri üzerine dünyadaki en eksiksiz döküm çalışmalarını topladı. Ve video oyununu icad eden kişi işte bu kişiydi.

(Data Manager'de John Andeson'un yazısından özet, Nisan 1983....)

Yığın belleklerde kullanılan okuma ve yazma aygıtları, iki aralıkla sınırlandırılmış şerit veya disk parçalarına verilerin fiziksel olarak kaydedilmesini sağlar. Belleğin bu işlemle ilgili bölümüne **fiziksel kayıt** adı verilir. Büyüklüğü merkez birimde bilgileri barındırmaya ayrılmış bellek alanının büyüklüğüne göre değişir.

Fiziksel kayıtlar oluşturulurken, merkez belleği tıkamadan, hem bilgilere erişim süresini kısaltmak, hem de yığın bellekte korunan bilgilerin sayısını mümkün olduğu kadar artırmak gereklidir.

Personel kütüğümüzün kayıtlarına dönecek olursak, bunlar manyetik bir ortam (örneğin şerit) üzerinde iki değişik biçimde gözükebilirler:

Örnek 1

Aralık	Kayıt B	Aralık	Kayıt A	Aralık

|← Öbek →|

Örnek 2

Aralık	Kayıt C	Kayıt B	Kayıt A	Aralık

|←——— Öbek ———→|

Birinci örnekte aralıklar bütün kayıtları ayırmaktadır. Oysa ikinci örnekte, birçok kayıt iki aralık arasında bir öbek oluşturacak bir biçimde biraraya getirilmiştir. Bu ikinci kayıt yöntemi bilgi içermeyen kısımların uzunluğunu kısaltmaya ve böylece, bir şeride sığdırılabilen mantıksal kayıt sayısını artırmaya olanak vermektedir.

Bundan başka, şerit birimi üç mantıksal kaydı aynı anda erişilir kıldığı için, okuma ve yazma işlemleri için gereken süre de kendiliğinden kısalmaktadır. Bu şekilde tutulan kayıtlar kümesine **öbek** ya da **fiziksel** kayıt, bir öbekteki kayıt sayısına da **öbek katsayısı** adı verilir. (2. örnekteki öbek katsayısı 3'tür.)

Belirtelim ki, sabit uzunluklu kayıtlar aynı öbek katsayısı ile kümelenir, böylece, bütün öbekler aynı sayıda veri içerir. Kayıtlar değişken olduğu takdirde ise, kayıtları bölmekten kaçınılır. Bunun için bir en-büyük-öbek uzunluğu belirlenir ve her öbekte mümkün olduğunca çok sayıda kayıt biriktirilir. Bu durumda, her öbekteki veri sayısı öngörülmüş olan en büyük uzunluğa eşit ya da bu uzunluktan daha az olur. Öbek başına yalnız bir kayıt varsa, fiziksel kayıt ile mantıksal kayıt arasında bir çakışma var demektir. Küçük ve kişisel bilgisayarlarda kullanılan disketlerde bir öbeğin uzunluğu genellikle 128 karakterdir. Herhangi bir kütükteki veri kayıtlarının uzunluğu, alan uzunluklarının toplamına eşittir; yani personel kütüğü örneğimizde 3 + 2 + 40 + 2 + 7 = 54'e eşittir. Bu durumda, öbek başına iki kayıt yerleştirilebilir (128/54 = 2,37).

Manyetik Şerit Üzerinde Kütüklerin Düzeni

Manyetik şerit üzerindeki bir kütük, okuma ve yazma kafaları altında bilgilerin birbirini izlediği bir mantıksal kayıtlar dizisidir. Kütüklerin en önemli özelliği kayıtların nasıl yapılacağını belirleyen sıralı yapısıdır. Bu yapıya bağlı olarak kayıtlar düzensiz bir biçimde yapılabileceği gibi artan ya da azalan bir sırada da düzenlenebilir. Düzensiz bir biçimde yazılan tutanaklar, genellikle, sistem-programı içinde öngörülen bir **sıralama** (sort) programı ile sınıflandırılır.

Personel kütüğünün verileri aşağıdaki gibi değişik biçimlerde kaydedilebilirler:

1 / Kayıt numarası düzeni gözetilmeden
Num. 15 Yılmaz Ahmet
Num. 2 Çataloğlu Efruz
Num. 7 Tezcan Alper
Num. 5 Fellah Hasan
Num. 20 Yiğit Meltem
...

2 / Artan tutanak numarası düzeninde
Num. 2 Çataloğlu Efruz
...
Num. 5 Fellah Hasan
...
Num. 7 Tezcan Alper
...
Num. 15 Yılmaz Ahmet
...
Num. 20 Yiğit Meltem

3 / Azalan tutanak numarası düzeninde
Num. 20 Yiğit Meltem
...
Num. 15 Yılmaz Ahmet
...
Num. 7 Tezcan Alper

...
Num. 5 Fellah Hasan
...
Num. 2 Çataloğlu Efruz

Bu örnekte kayıt numaralarının sırası, işçilerin soyadlarının alfabe sırası ile aynıdır. Bu ikili sıralama, ancak, örneği daha basit kılmak için uyarlanmıştır. Önemli olan, kayıtların hesap defterine hangi sıra içinde yazılmış olduğudur.

Artan ya da azalan biçimde düzenlenmiş kütüklerde arama işlemi daha kolay yürütülür. Manyetik şerit üzerinde kütükler genellikle artan biçimde sıralanır. Arama istenen şifreyi taşıyan kayıtla karşılaşınca (kayıt bulunmuştur), ya da istenenin bir büyüğü şifreyle karşılaşınca (kayıt mevcut değildir) sona erer.

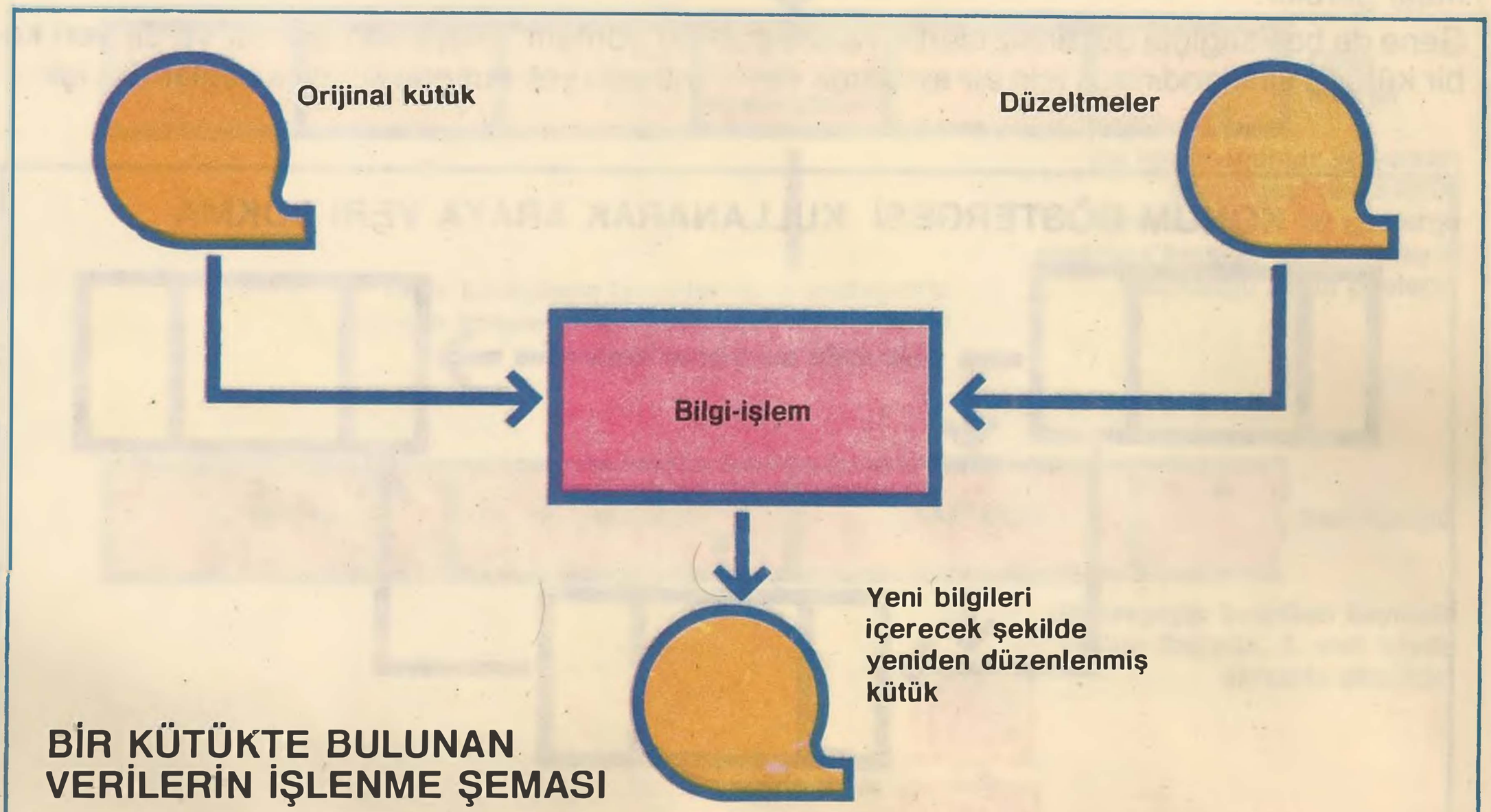
Taşıdığı verilerin düzeni biliniyorsa yeni gelen bilgileri içerebilmesi için bir kütük, kolaylıkla yeniden düzenlenebilir.

Diyelim ki artan kayıt numarasına göre düzenlenmiş personel kütüğünde Mart ayı içinde şu değişiklikleri yapmak istiyoruz:

Num. 7 adres değişikliği
Num. 15 Yılmaz Ahmet istifa ediyor
Num. 21 Zapçı Süheyla işe alınıyor

Aşağıdaki şekil, yürütülmesi gereken işlemi gösteriyor.

Yenilenmiş kütüğümüz şu veri kayıtlarını içerecektir:



BİR KÜTÜKTE BULUNAN VERİLERİN İŞLENME ŞEMASI

Num. 2 Çataloğlu Efruz
...
Num. 5 Fellah Hasan
...
Num. 7 Tezcan Alper
...
Num. 20 Yiğit Meltem
...
Num. 21 Zapçı Süheyla

Manyetik Diskler Üzerinde Kütüklerin Düzeni

Şerit üzerine kaydedilen bilgi kütüklerine erişim, zorunlu olarak, sıralıdır. Oysa fiziksel özelliklerinden ötürü manyetik disklerin taşıdığı bilgilere farklı yöntemlerle erişilebilir. Aslında değişik alanlarda bulunan verileri çabucak bulmak için kayıtlar birbirine bağlanarak yapılar oluşturulursa, vakit kazancı sağlanabilir.

İşte bir kütüğün disk üzerinde düzenlenmesinde en çok kullanılan dört yöntem:

- Sıralı düzen
- Zincirleme sıralı düzen
- Anahtarlı sıralı düzen
- Doğrudan erişimli düzen

Sıralı düzen: Veri kaydının anahtar alandaki sınıflandırma koduna uyularak kayıtlar birbiri ardına yazılır. Bu düzen, manyetik şeritte kullanılan düzene benzemektedir: Bir kayda erişmek için ondan önceki bütün kayıtların okunması gerekir.

Gene de başlangıçta düzensiz olarak yazılmış bir kütüğü sınıflandırmak için bir ayıklama yapılabilir ya da bazı sınırlar içerisinde, kütükteki bazı kayıtlara kütüğün hepsinin yazılmasına gerek kalmaksızın ulaşılabilir. Manyetik şeritle, bu, olanak dışıdır.

Böyle düzenlenmiş bir kütük, başka bir kütük gerektirmeksizin yeni verileri içerecek şekilde yeniden düzenlenebilir. Bunun için belli bir kaydın bu işe ayrılmış alanına özel bir işaret konarak kütükte bir boşluk yaratılır. İşaretlenen kayıt fiziksel olarak yok edilmemiştir; yalnız geçici olarak, okunmaz kılınmıştır.

Herhangi bir veri kaydının içeriğini değiştirmek için ise kaydın içinde bulunduğu öbeğin (yani fiziksel kaydın kendisinin) toptan olarak yeniden yazılması gerekir. Ayrıca, yeni veri eklemek için, kütüğün sonundaki boş alanlar da kullanılabilir.

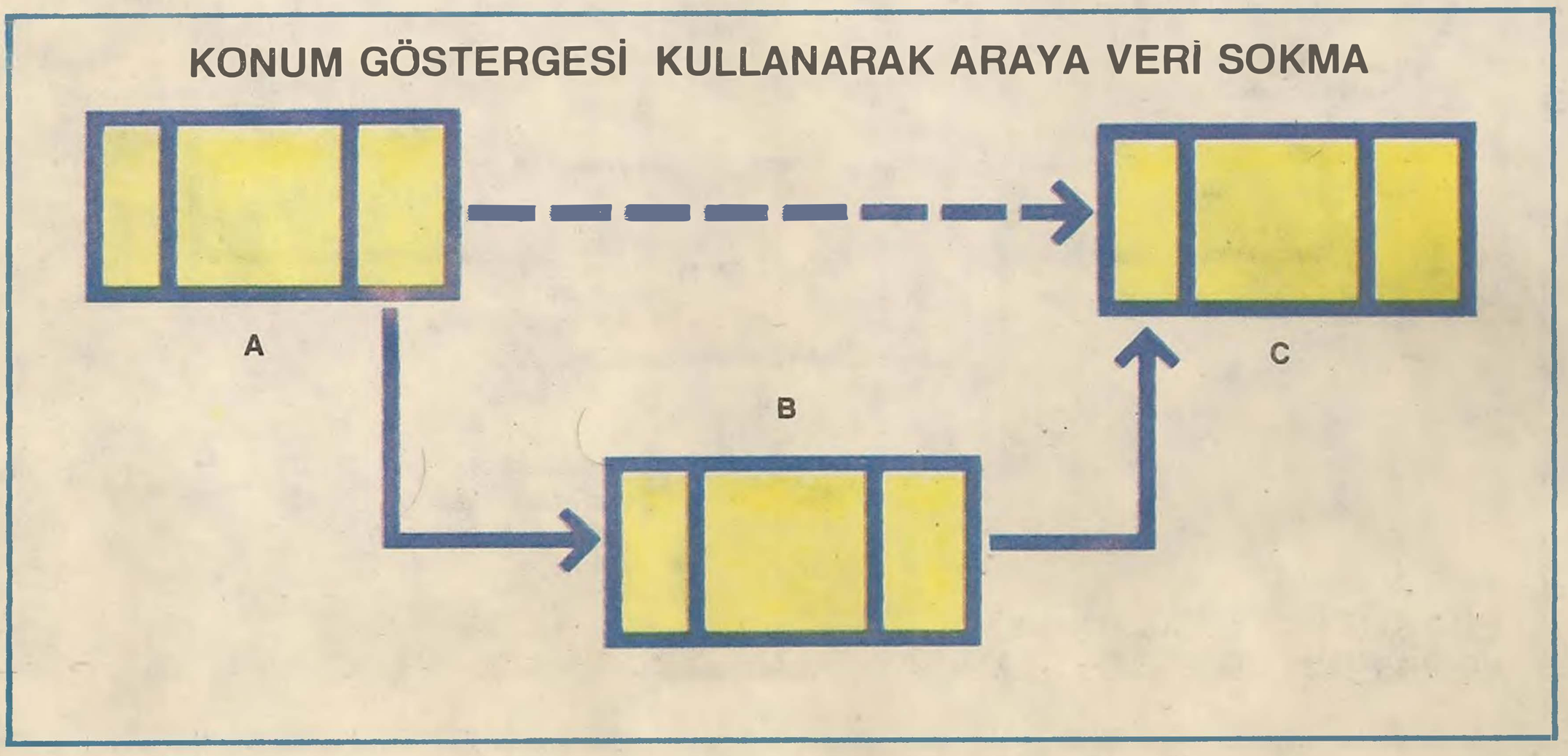
Yine de açıkladığımız tüm üstünlüklerine ve sağladığı hıza karşın, sıralı yöntem manyetik disklerin tümünden yararlanamamaktadır.

Zincirleme sıralı düzen: Her kayıtta, alanlarının yanında, o kaydı bir sonraki kayda bağlayan ya da bu kayıtla ilgili başka veri kayıtlarının bulunmasını mümkün kılan bir ek alan bulunur.

Bu zincirleme alanlara **konum göstergeleri (pointer)** adı verilir. Bu durumda kayıtlar şu şekilde oluşturulur:

Anahtar	Veriler	Konum göstergesi

Bu yöntem, araya veri sokma ya da veri kayıtlarını yok etmede kolaylık sağlar. Bu işlem-



ler, konum göstergeleri zincirindeki basit düzenlemelerle gerçekleştirilir.

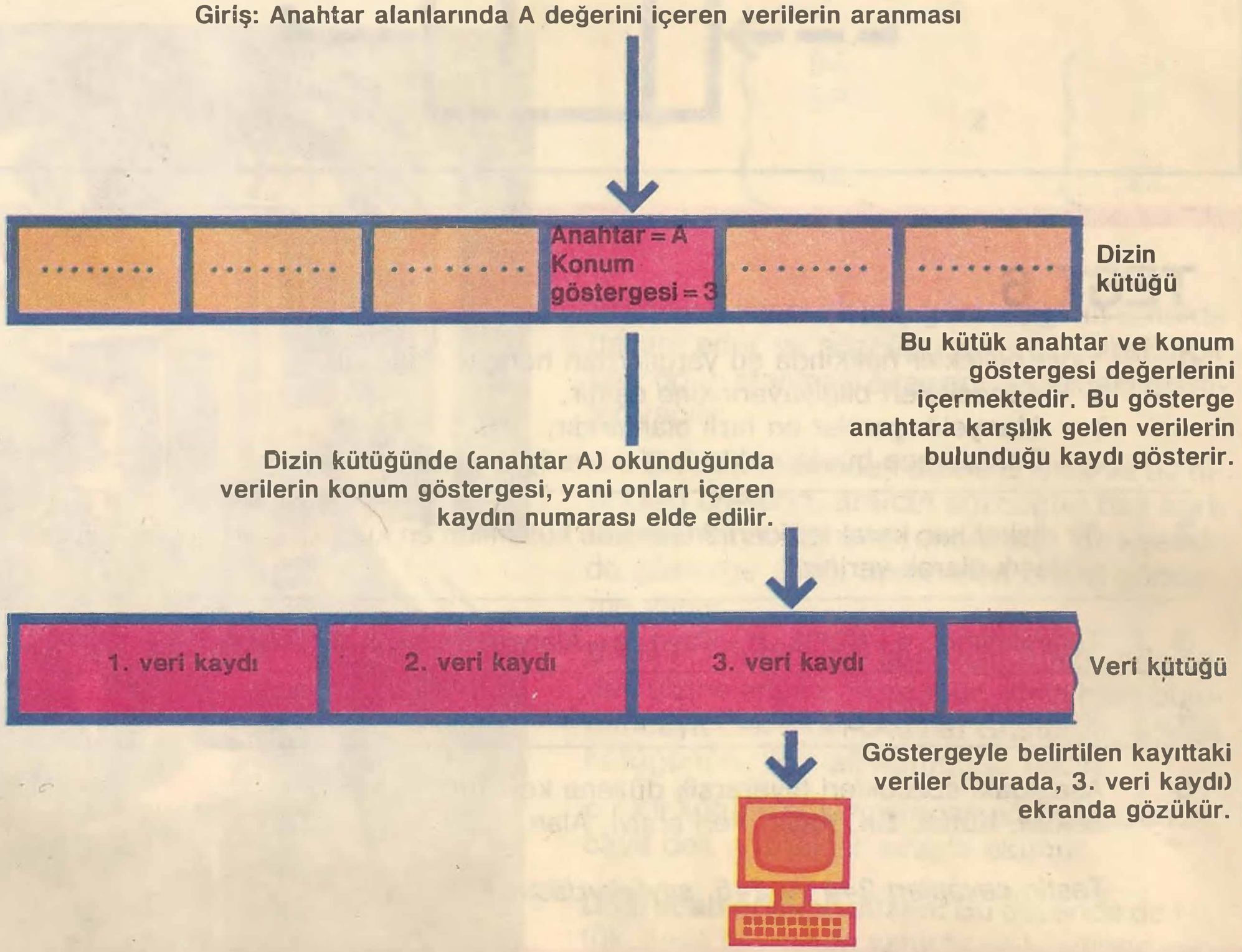
A / Araya sokma örneği:
A ve C kayıtları arasına bir B kaydı eklemek için, A'nın konum göstergesi B'yi gösterecek biçimde düzeltilir, B'ye de C'yi gösteren bir konum göstergesi yerleştirilir.

B / Yok etme örneği:
A B C zincirinden B kaydını çıkarmak için, A'nın konum göstergesini C'ye yöneltmek yeterlidir.

Kayıtların fiziksel sıralanması değişmez ve yeni kayıtlar, **kapasite artığı alanına** eklenir; kütüğün sıralı olarak yeniden yazılması ile de periyodik olarak temizlenir.

Anahtarlı sıralı düzen: Kütükte böyle bir düzenin oluşturulmasını sağlayan temel unsur, anahtar alanıdır: Aldığı değer belli bir veri kaydını ikinci bir kayıtla karıştırılmayacak biçimde tanımlar.

ANAHTARLI SIRALI BİR KÜTÜKTE VERİLERİ ARAMA YÖNTEMİ

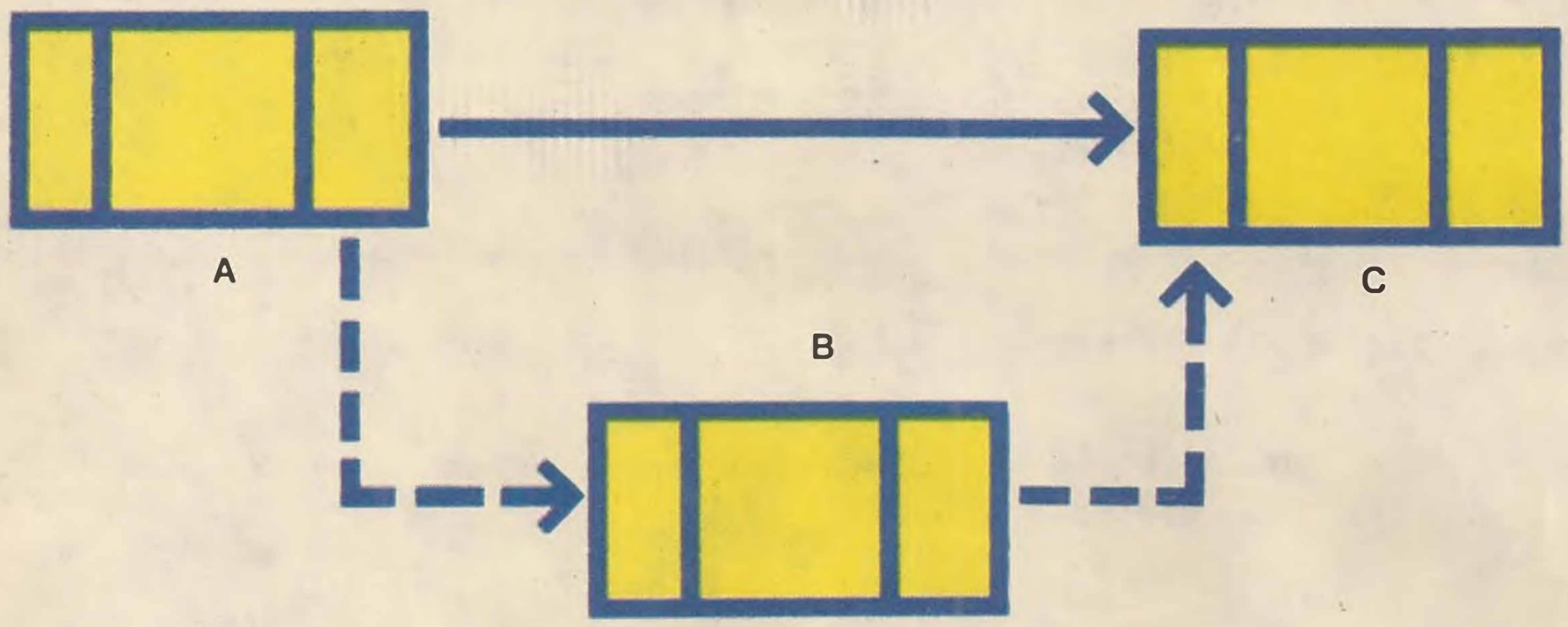


Bu yöntemde de, sıralı bir biçimde düzenlenen kütükte sıralama, erişim anahtarlarına göre yapılır. Ayrıca, bir **dizin** (index) işlevi görecek ikinci bir kütük açılır. Dolayısıyla anahtarlı sıralı bir kütük iki unsurdan oluşur:

- Bir dizin kütüğü
- Bir veriler kütüğü

Sayfa 239'daki şekil bu yöntemin ilkesini göstermektedir. Dizin işlevini gören kütüğün kendisi de bir tablo olarak, sıralı bir kütük düzeninde, diske kaydedilir. Aynı kütük birçok dizine bağlanabilir, bu da erişim vaktinin biraz daha azalmasına yol açar.

KONUM GÖSTERGELERİ ARACILIĞIYLA VERİLERİN SİLİNMESİ



TEST 6

1 / Yığın bellekler hakkında şu yargılardan hangisi doğrudur?
a - Kapasiteleri bilgisayarınkine eşittir,
b - Manyetik şeritler en hızlı olanlarıdır,
c - Son derece büyük miktarlardaki veriyi saklamaya yararlar.

2 / Bir disket kaç karakter içerir? Genelde kullanılan en küçük ve en büyük miktarları yaklaşık olarak veriniz.

3 / Tanımlayınız: a - Kütük, b - Kayıt, c - Alan.

4 / Manyetik şeritler genelde ne işe yararlar?

5 / Aşağıdaki sözcükleri hiyerarşik düzene koyunuz:
Sekizli, Kütük, Bit, Kayıt, Veri arşivi, Alan.

Testin cevapları 244 ve 245. sayfalardadır.

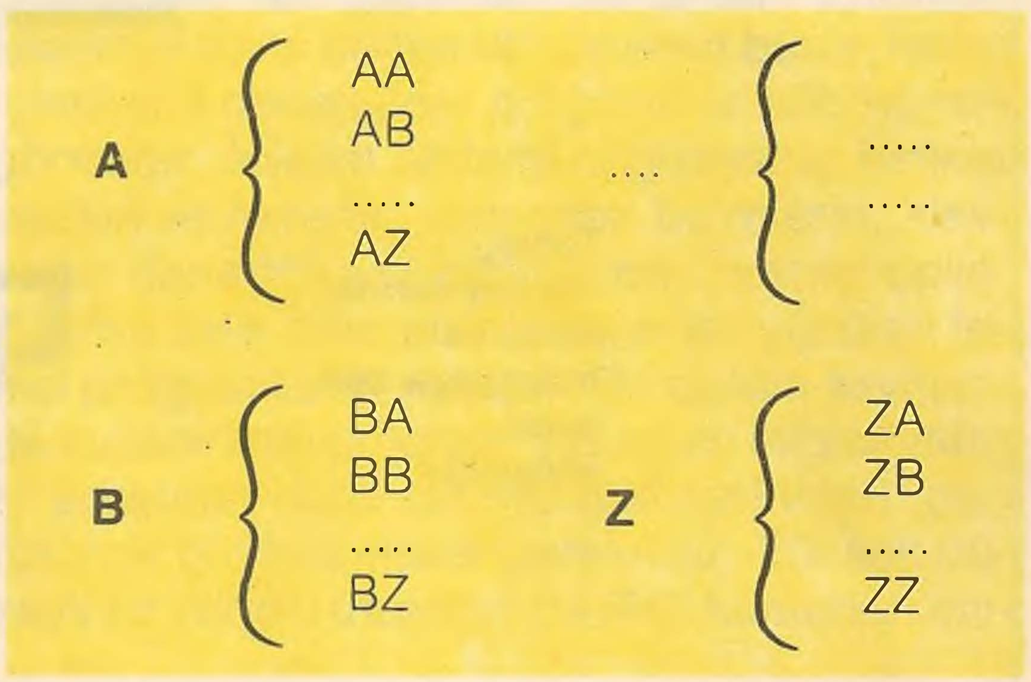
Bu düzeni daha iyi açıklamak için iki klasik örneği ele alalım. Diyelim ki, bir depoda bulunan ürünlerle ilgili bir kütük oluşturacağız. Kayıt, 3. silindirden başlayarak bir monodisk üzerine yapılacak. Bunun için aşağıda görülen dizin kütüğünü hazırlamamız gerekir:

Dizin Kütüğü

Veri göstergesi (Silindir numarası)	Veri erişim anahtarları (Her silindirdeki son anahtar verilmiştir)
3	1450
4	2972
5	4250
6	5587
7	6999

Belirli bir anahtar koduyla kodlanmış verilere erişmek için önce dizin kütüğü sıralı olarak okunur ve bu esnada aranan veri anahtarı dizin kütüğündeki bütün anahtarlarla karşılaştırılır. Aranan anahtarla aynı olan veri anahtarı bulunduğunda, o anahtara karşılık gelen veri göstergesi (yani silindir numarası) okunur. Böylelikle bütün veri kütüğünü okumak yerine, dizin kütüğü sayesinde bulunan tek bir silindirin kayıtlarının okunmasıyla yetinilmiş olunur. Örneğin, anahtarı 5350 olan bir malla ilgili bilgiler (bu anahtar 4250 ile 5587 arasında yer aldığından) yalnızca 6. silindirdeki kayıtlar arasında aranır.

İkinci örneğimiz bir sözlükteki sözcükleri içeren bir kütük. Sözlük, ilk olarak sözcüklerin baş harflerine göre alt-kütüklere ayrılarak, 29 harfli bir tablo oluşturulur (en alt kademeden dizin). Her harf grubu, sözcüklerin ikinci harflerine göre kendi içinde de alt-kütüklere bölünebilir. Böylece her biri alfabenin harflerinden oluşan 29 tablo daha ortaya çıkar (ikinci kademeden dizin). Daha sonra en alt kademeden dizinin her ögesiyle, sözcüklerin ikinci harflerinin saklandığı ikinci tablonun adresi eşleştirilir. Bu şekilde aşağıdaki bağıntılar elde edilir:



Aramaya, ikinci dizinin gösterdiği alt-kütükte devam eder ve sözcüğü buluruz. Özetle bir sözcüğün aranması sırasında aşağıdaki işlemler yapılır:

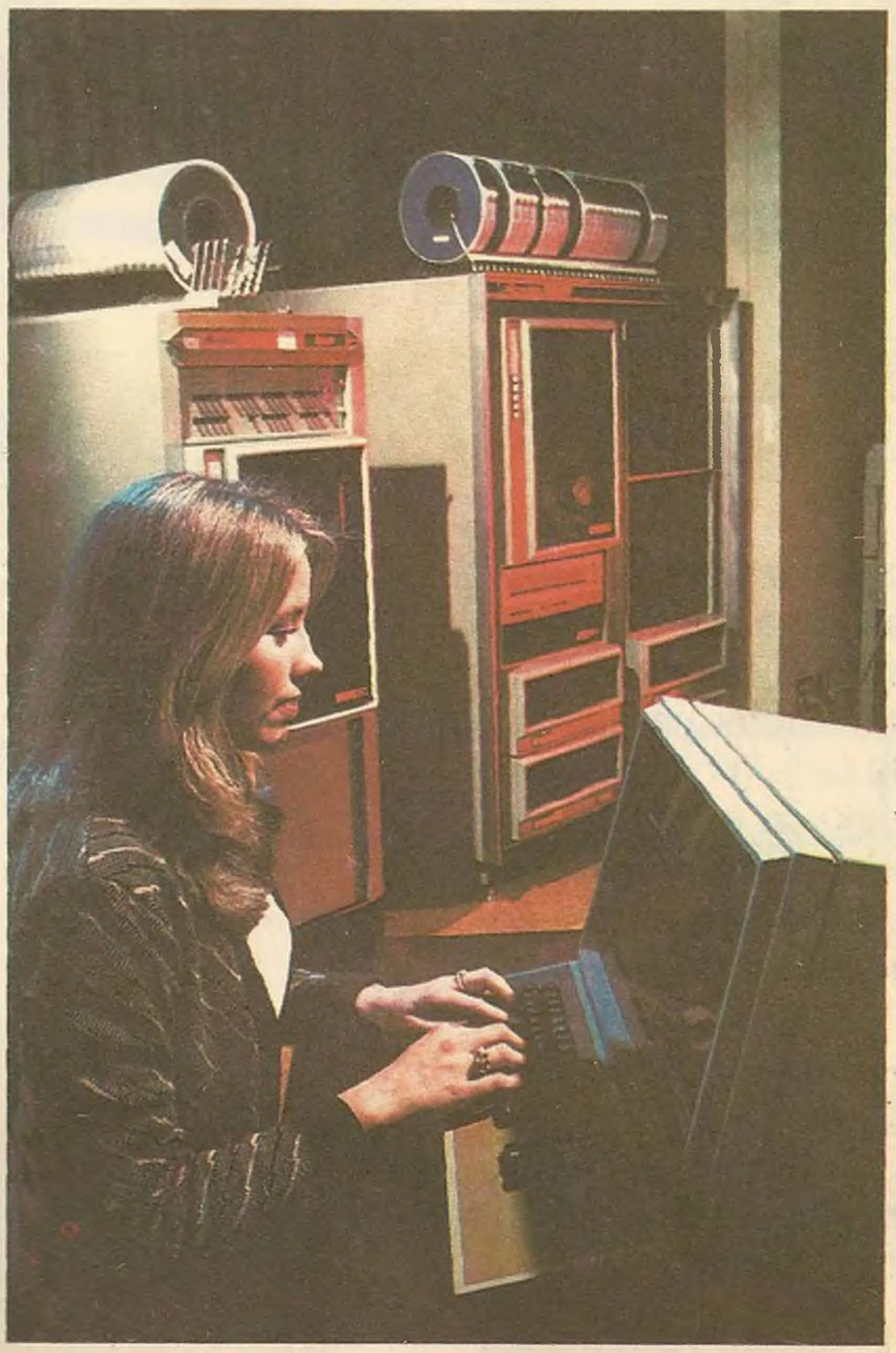
a - En alt kademeden dizine erişme ve bu dizindeki ögelerin, aranan sözcüğün baş harfi bulununcaya dek sıralı okunması. Bu aşamada gösterge ikinci dereceden dizine gönderme yapar.

b - İkinci harfle ilgili tablo için ikinci kademeden dizine erişim. Sözcüğün ikinci harfi bulununcaya dek sıralı okuma. Gösterge, sözlük kütüğünün, ilgili alt-kütüğünü bildirir.

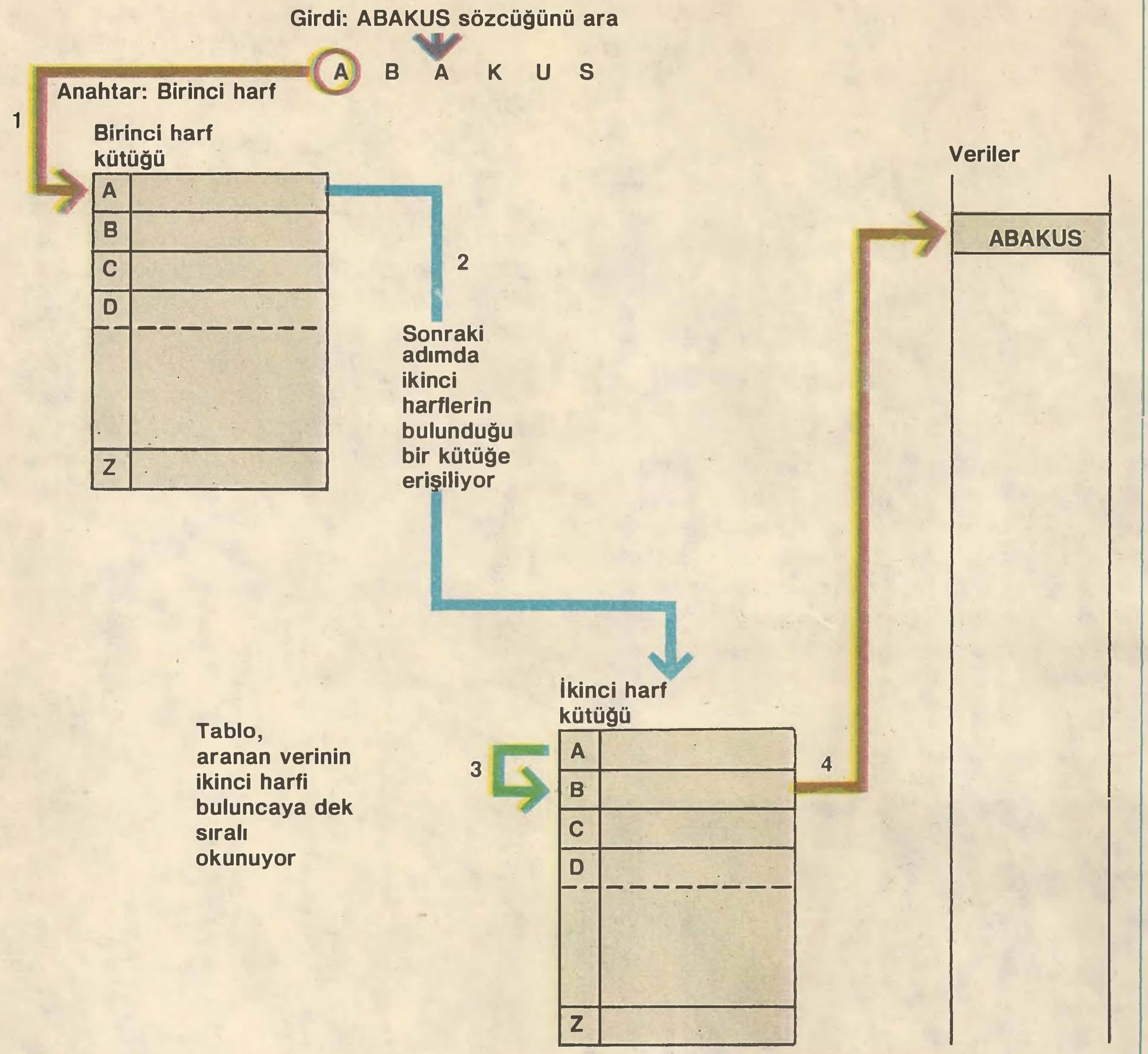
c - Alt-kütüğe erişim; aranan sözcük bulununcaya dek sözcükler sırayla okunur.

Doğrudan erişimli düzen: Bu düzende de kütük, sıralı bir yapıya sahiptir ve kayıtların yerleri önceden belirlenmiştir.

Veri yazıcı, klavye başında.



İKİ DİZİNLİ BİR SÖZLÜK KÜTÜĞÜNDE ARAMANIN MANTIK ŞEMASI



Kayıtların kütüğün içindeki fiziksel konumu, bir **işlem algoritması** (raslansal işaretlere dönüştürme formülü) tarafından belirlenir. Kullanılacak algoritmanın okunacak ya da yazılacak verilerin anahtarına uygun olması gerekir. Ayrıca kütükte bu şekilde adreslenecek tüm kayıtlar için yer bulunması gerekir. Başka bir deyişle kütük, gerçekte varolandan daha fazla kaydın sığabileceği kadar büyük olmalıdır.

Bu düzenin büyük sakıncası, manyetik diskte yer savurganlığına neden olmasıdır.

Örnek olarak telefon rehberi kütüğünün yapısını ele alalım. Saklanacak veriler şunlardır:

1	/ Soyadı	20 karakter
2	/ Ad	10 karakter

3 / Şehir	10 karakter	
4 / Cadde	20 karakter	
5 / Sokak	8 karakter	
6 / Bina numarası	3 karakter	
7 / Şehir telefon kodu	5 karakter	
8 / Telefon numarası	8 karakter	

Her veri için belirli sayıda karakter öngörülmüştür. Bellek yalnızca sıralanan veriler sakladığından bir kütük yeterlidir. Farklı veriler içeren daha karmaşık bellekler ise, her birinde verilerin bir kısmının saklandığı birçok kütükten oluşur. Örneğimize mantıksal veri kaydının uzunluğu, toplam 8 veri alanındaki simgelerin sayısı olan 84'tür.

Aşağıdaki şekil bu örnekte kullanılan mantıksal veri kaydı yapısını gösteriyor. Mantıksal veri kayıtlarının yazılacağı fiziksel kaydın uzunluğu küçük ve kişisel bilgisayarlarda en sık kullanılan uzunluk olan 128 karakterdir.

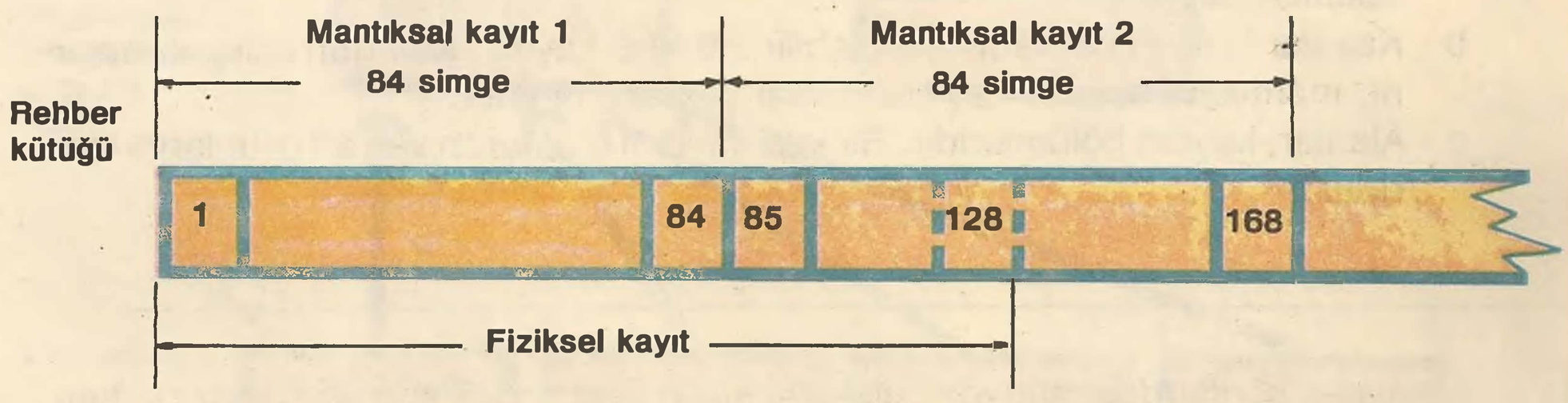
Görüldüğü gibi fiziksel kayıt uzunluğu mantıksal kayıt uzunluğundan fazladır. Bunun sonucu olarak, örneğin kayıt 1'i okumak gerektiğinde, gerçekte mantıksal kayıt 2'nin bir kısmı da belleğe getirilir.

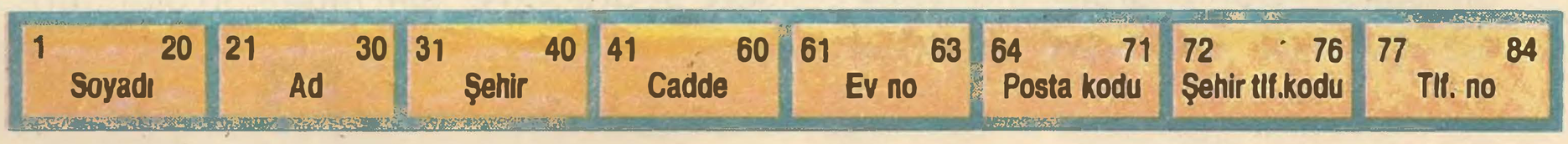
İstenen verileri gereksiz olanlardan ayırmak, işletim sistemi denen ve sistemin bütün temel işlevlerini denetleyen programlar bütününün görevidir. İşletim sistemi bilgisayarla birlikte verilen ve ekranın, manyetik birimlerin, klavyenin denetimi gibi bilgisayarın normal çalışmasına eşlik eden, bütün işlevleri yürüten temel programlardan oluşur. Bu sistem sayesinde kullanıcının program yazarken bilgisayarın iç yapısının nasıl olduğu gibi ayrıntıları gözönünde bulundurması gerekmez. Yüksek düzeyli bir dildeki (örneğin BASIC) komutları ma-

Özetlersek:

- **Verilerin saklandığı başlıca manyetik bellekler: Manyetik şeritler, manyetik tamburlar, sabit diskler ve disketler.**
- **Belirli bir parametreye göre türdeş olan verilerin toplamına kütük denir.**
- **Okuma yazma işleminin sona ermesinden sonra, kütükler, programlama için fiziksel ve mantıksal kayıtlara bölünürler.**
- **Veri alanlarının uzunluğuna bağlı olarak mantıksal bir kayıt, birden fazla fiziksel kaydı içine alabilir veya fiziksel bir kayıtta birden fazla mantıksal kayıt bulunabilir.**
- **Mantıksal kayıtlar, her biri belirli bir bilgi içeren alanlara bölünmüştür.**

TELEFON REHBERİ KÜTÜĞÜNÜN YAPISI





Mantıksal kayıtta alanların konumu

TEST 6'nın Çözümleri

1 / c - Veriler yığın bellekte sınırsız bir süre için saklanabilir. Bilgisayar belleğinden, verilerin bir bölümünün işlenmesi sırasında yararlanılır. Çıktılar (işlenmiş veriler), saklanmak üzere ya da başka bir problemin çözümü için program belleğini boşaltmak amacıyla yeniden yığın belleğe verilir.
Ara sonuçları geçici olarak manyetik disklere kaydetme yöntemi, aynı anda birçok kullanıcıya hizmet eden merkezi bilgisayarlarda sık sık kullanılır (manyetik disk en hızlı doğrudan erişimli yığın bellektir).
Bir kullanıcı çalışmasına ara verince, başka bir işin yürütülebilmesi için, veriler manyetik diske kaydedilir.
Kesintiye uğramış bir programa (ya da onun bir kısmına) merkezi işlem biriminin işi azalır azalmaz devam edilir.
Veri ve programların merkezi birimle yığın bellekler arasında gidiş gelişinin işletim sistemi tarafından otomatik olarak yönetilmesi, veri işleme sisteminin tüm olanaklarından en iyi biçimde yararlanılmasını sağlar.

---

2 / Küçük modellerde yaklaşık 250.000, büyüklerde 1.2000.000 karakter. Bu ansiklopedinin tümüyle yazı dolu her sayfasında iki sütun halinde 50 satır vardır. Her satırda ortalama 45 karakter bulunduğuna göre böyle bir sayfanın toplam simge sayısı $2 \times 50 \times 45 = 4.500$'dür. Fotoğraf ve şekillerin tüm ansiklopedinin yaklaşık üçte birini kapladığını varsayarsak sayfa başına ortalama 3.000 karakter düşer. 24 sayfadan oluşan her fasikülde $24 \times 3.000 = 72.000$ karakter bulunur. Sonuç olarak 1.200.000 karakterlik bir diskette 16 fasikül, 5 diskette ise ansiklopedinin bütünü saklanabilir.

---

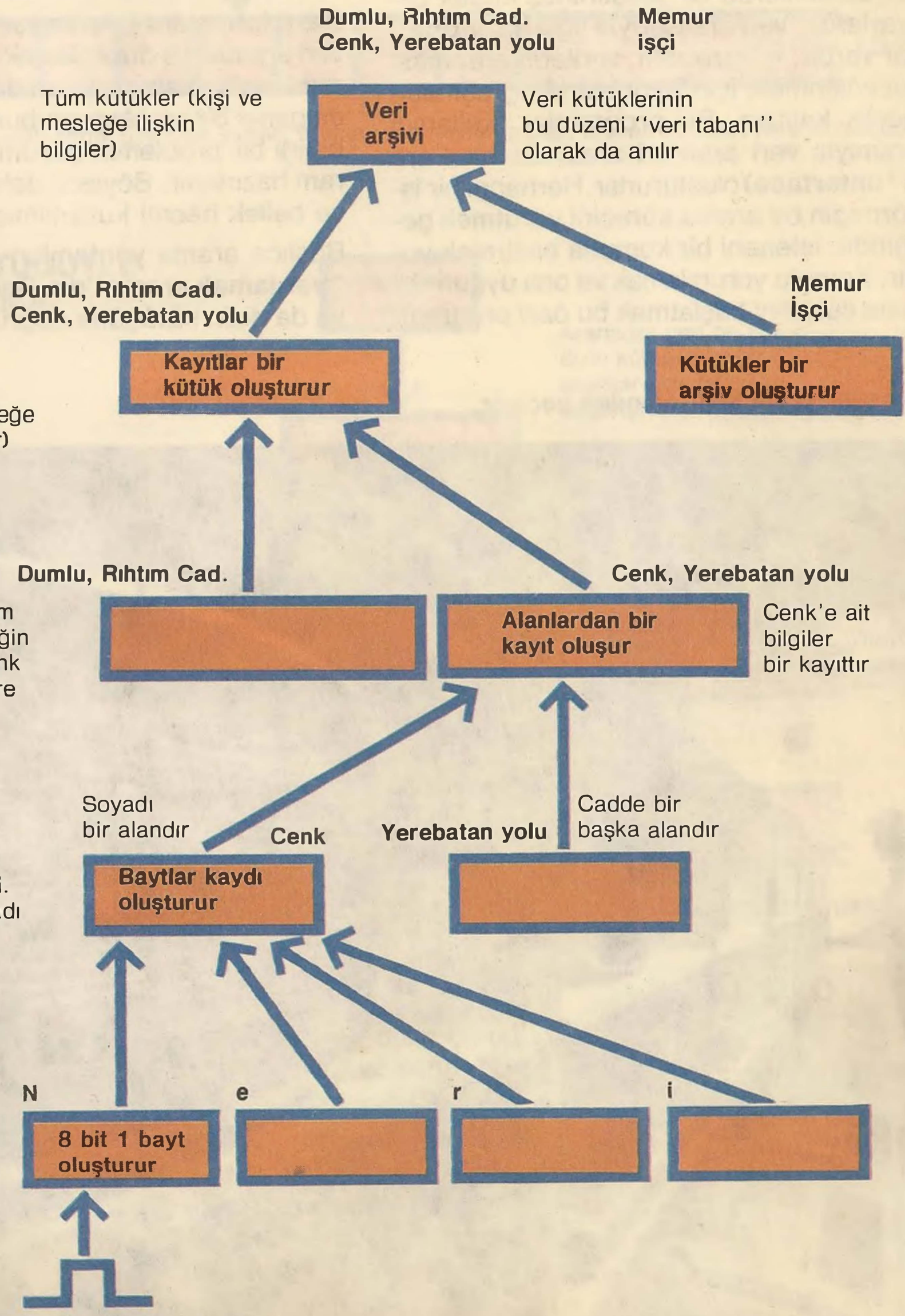
3 / a - Kütükler, yığın belleğin, aynı ad (kütüğün adı) altında türdeş veriler içeren bölümlerdir.
b - Kayıtlar fiziksel ve mantıksal olabilir. Fiziksel kayıtlar kütüğün fiziksel yapısını, mantıksal kayıtlar ise mantıksal yapısını belirler.
c - Alanlar, kaydın bölümleridir. Bir veri kaydını oluşturan veri alt gruplarını içerirler.

---

4 / Manyetik şeritlerden manyetik diskteki kütüklerin yedek kopyasını hazırlarken yararlanılır. Verileri sıralı sakladıklarından sürekli kullanılan kütüklere uygun değildirler. Buna karşılık manyetik disklere doğrudan erişim mümkün olduğundan çok hızlı veri alışverişi sağlarlar.

5 / Bit, bayt, alan, kayıt, kütük, arşiv. Şekilde bu hiyerarşi görülüyor.

BİR KAYDIN HİYERARŞİK DÜZENİ



kine dilindeki komutçuklara işletim sistemi çevirir. Gerçekte işletim sisteminin bu işlevleri yürütmek için kullandığı mekanizmalar daha karmaşıktır. Bu mekanizmaları işletim sistemleri bölümünde daha ayrıntılı olarak inceleyeceğiz.

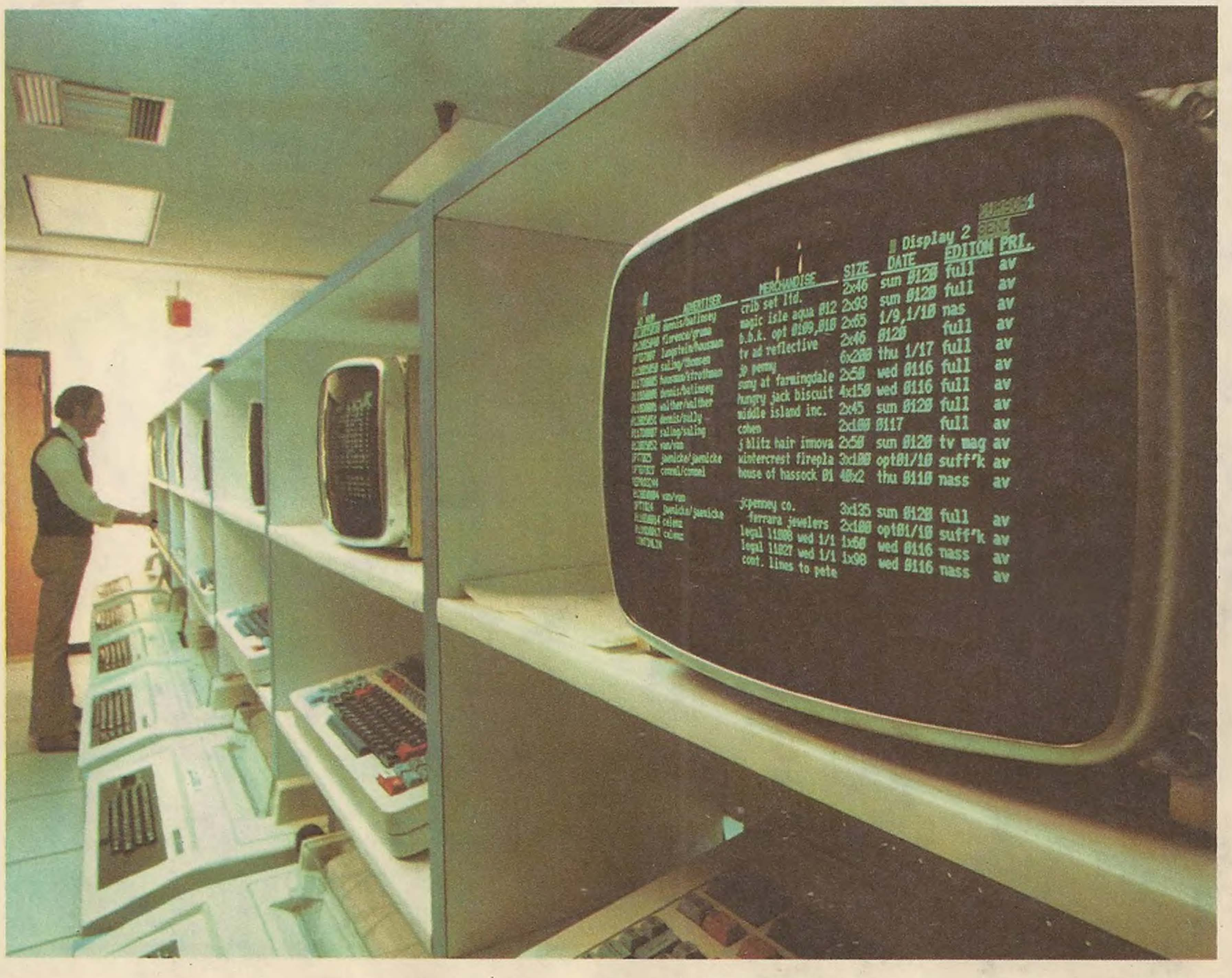
Küçük ve Kişisel Bilgisayarlarda Arşiv Düzeni

Büyük sistemlerde ve geliştirilmiş küçük bilgisayarlarda, veri arşivleriyle ilgili hazır programlar vardır; kullanıcının, verilerin aranması ve düzenlenmesi için ayrıntılarla uğraşmasına gerek kalmaz. Bu programlar, kullanıcı programıyla veri arşivleri arasında bir ''ara birim'' **(interface)** oluştururlar. Herhangi bir işlevi, örneğin bir arama sürecini yürütmek gerektiğinde, isteneni bir komutla bildirmek yeterlidir. Komutu yorumlamak ve ona uygun veri arama sürecini başlatmak bu özel programların görevidir. Birkaç istisna dışında daha küçük sistemlerde (küçük ve kişisel bilgisayarlarda), bu türden programlar yoktur. Çünkü bu programların maliyeti özellikle bilgisayar fiyatıyla kıyaslandığında çok yüksektir. Ayrıca başka bir sakıncaları da vardır. Birbirinden çok farklı uygulamalarda kullanılabilmeleri için işletim sistemi programlarının olabildiğince geniş kapsamlı tutulmaları gerekir. Geniş kapsamlı program ise kaçınılmaz olarak disketlerde ve bellekte çok fazla yer kaplar. Küçük sistemlerde ise bellek hacmi pek büyük değildir.

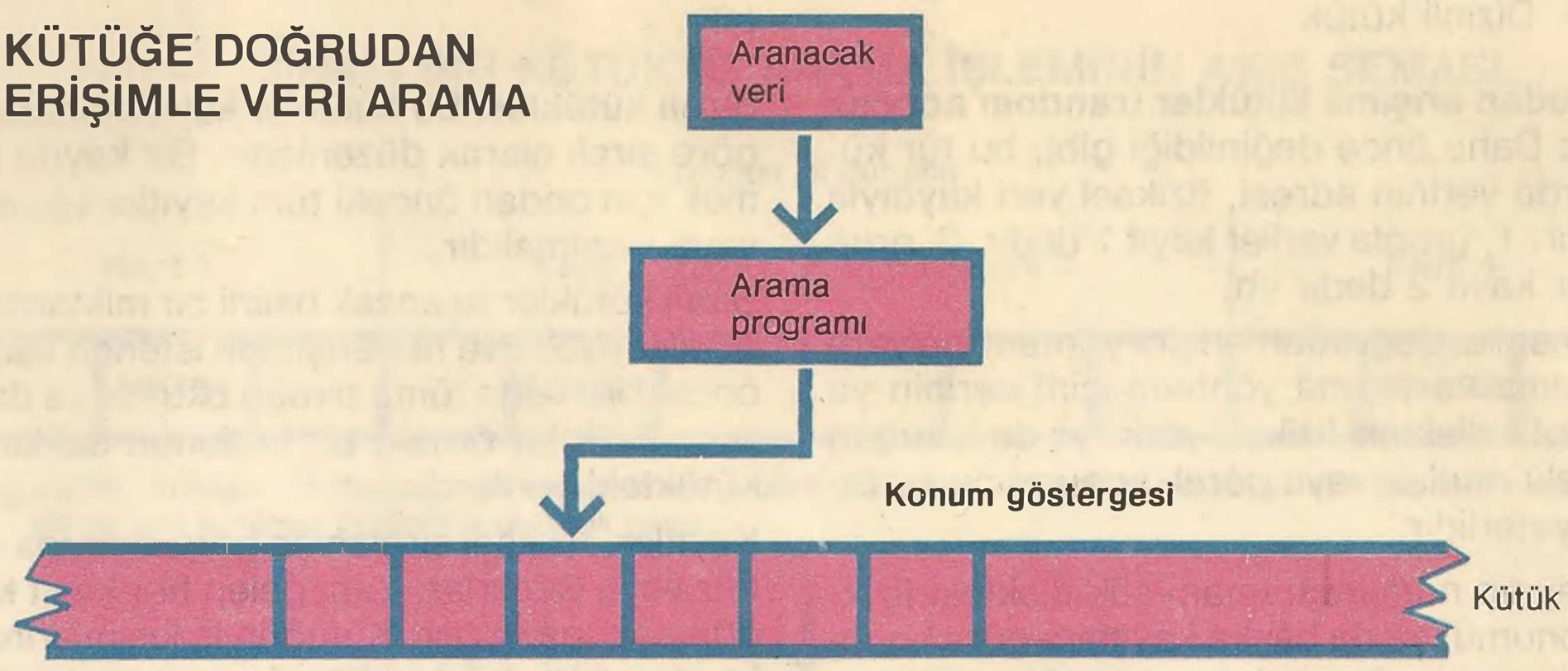
Bazı durumlarda, kullanıcı programı yanında veri aramak ve düzenlemek için gereken programları da yazmak yararlıdır. Böyle durumlarda genel bir problem grubuna değil, doğrudan belirli bir problemin çözümüne yönelik program hazırlanır. Böylece daha küçük bir disket ve bellek hacmi kullanılmış olur.

Başlıca arama yöntemleri **''seri arama''** ve **''yarılamalı arama''**dır. Seri aramada kütüğe ya da dizin kütüğüne doğrudan erişilir. İkinci

Terminaller kalite kontrolünden geçiyor.

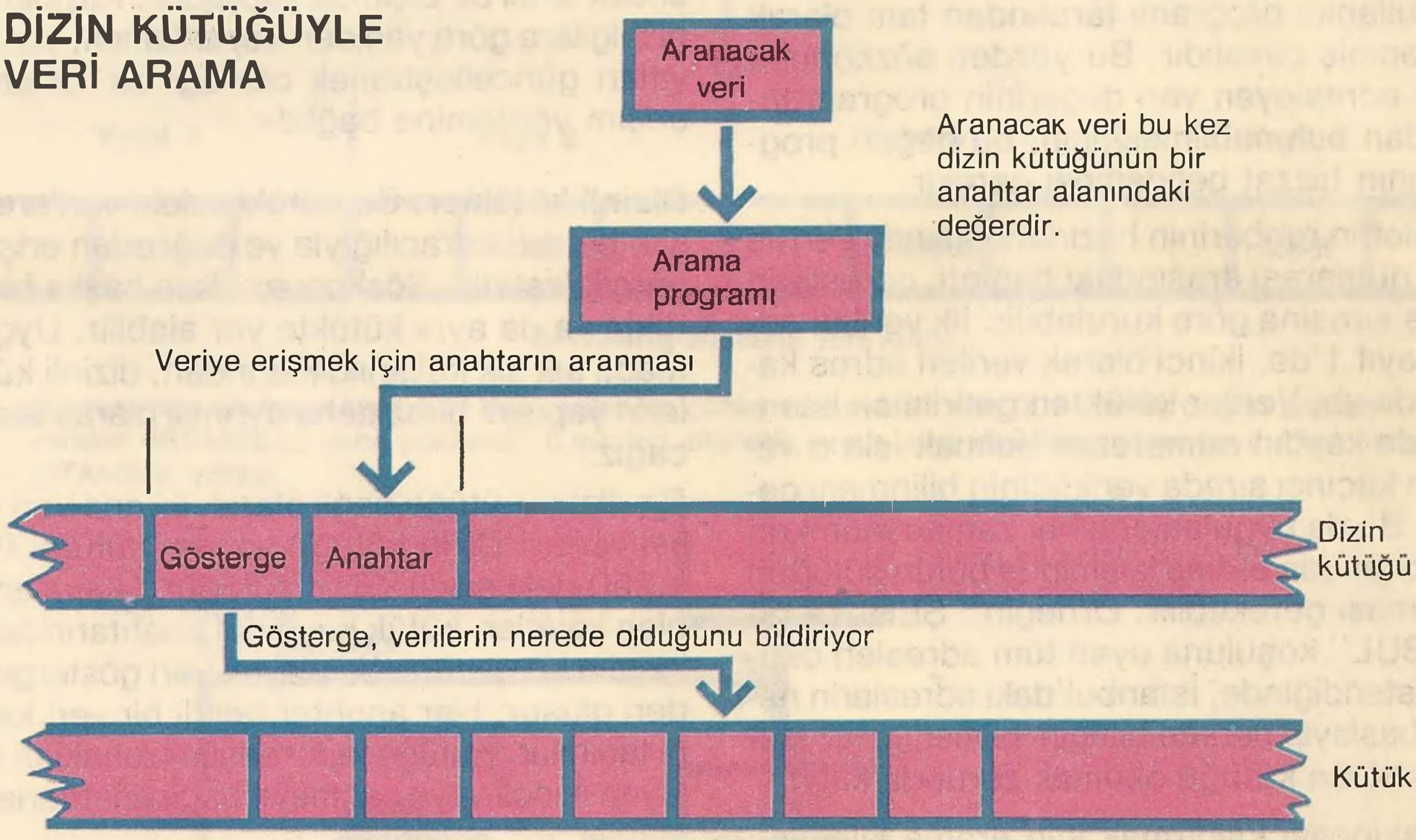


KÜTÜĞE DOĞRUDAN ERİŞİMLE VERİ ARAMA



Arama doğrudan kütük üzerinde yapılır. Arama programının çıktısı, aranan verilerin nerede olduğunu gösterir.

DİZİN KÜTÜĞÜYLE VERİ ARAMA



Aranacak veri bu kez dizin kütüğünün bir anahtar alanındaki değerdir.

Kütük dizinli ise, önce dizin kütüğündeki anahtar değeri aranır, böylece elde edilen gösterge başka bir kütükteki verilerin konumunu bildirir.

yöntemde ise önce dizin kütüğündeki anahtar değeri bulunur ve sonra buradaki göstergenin bildirdiği konumdaki veriye erişilir.

Kütüğe veya dizine doğrudan erişimli bu iki arama yöntemi de şekilde gösterilmiştir.

Çeşitli arama yöntemlerini açıklamadan önce küçük ve kişisel bilgisayarlarda kullanılan başlıca kütük türlerini yeniden hatırlayalım:

- Doğrudan erişimli kütük (yığın bellek kütüğü)

- Sıralı kütük
- Dizinli kütük

Doğrudan erişimli kütükler (random access files): Daha önce değinildiği gibi, bu tür kütüklerde verinin adresi, fiziksel veri kaydıyla aynıdır; 1. grupta veriler kayıt 1'dedir. 2. grup veriler kayıt 2'dedir vb.

Dolayısıyla, doğrudan erişim yöntemiyle veriyi okuma ve yazma yöntemi için, verinin ya manyetik diskteki fiziksel yerini ya da kütüğün içindeki mutlak veya göreli konumunu bildirmek yeterlidir.

Bir kaydın numarası (manyetik diskteki fiziksel konumu) ya da başka kayıtlara göre konumu biliniyorsa kayıt doğrudan okunabilir.

Göreli adreslemede kaydın numarası anahtar olarak kullanılabilir. Ancak bu takdirde veri kaydının numarasının aranacağı kütük konumu kullanıcı programı tarafından tam olarak belirlenmiş olmalıdır. Bu yüzden sözkonusu kaydı adresleyen veri değerinin program tarafından bulunabilmesi için, bu değeri programcının bizzat belirlemesi gerekir.

Bir telefon rehberinin hazırlanmasında veri ile kayıt numarası arasındaki bağıntı, adreslerin veriliş sırasına göre kurulabilir: İlk verilen adres kayıt 1'de, ikinci olarak verilen adres kayıt 2'de vb. Veriler kütükten getirilmek istendiğinde kaydın numarasını bulmak için o verilerin kaçıncı sırada verildiğinin bilinmesi gerekir. Bu da uygulamada her zaman mümkün olmadığından arama işleminde bütün kütüğün okunması gerekebilir. Örneğin "ŞEHİR = İSTANBUL" koşuluna uyan tüm adresleri okumak istendiğinde, İstanbul'daki adreslerin nerede başlayıp nerede bittiğini bilmediğimiz takdirde bütün kütüğü okumak zorunda kalırız.

Bu sakıncayı kaldırmak için arama işlemine başlamadan önce kütükteki şehirler alfabetik sıraya sokulur. Böylece adreslerin aranmasında hatırı sayılır bir kolaylık sağlanır. İSTANBUL'dan daha "yüksek" bir ögeye (örneğin İZMİR) gelindiğinde aramaya son verilir, çünkü bundan sonraki kayıtlarda İSTANBUL'un bulunmadığı bilinmektedir.

Bu sürecin mantıksal işleyişi s.249'daki şekilde görülüyor. Şekildeki akış şeması yalnızca ön bilgileri sunmaktadır. Arama yöntemi daha sonra tüm ayrıntılarıyla açıklanacak. Şekilde akış şeması verilen yöntemin başlıca yararı, kayıtlara doğrudan erişimi sağlamasıdır. Herhangi bir kaydı, istenen herhangi bir sırada yazmak ya da okumak mümkündür. Örneğin önce kayıt 100, sonra kayıt 50 incelenebilir.

Sıralı kütükler: Bu kütükler kayıt numarasına göre sıralı olarak düzenlenir. Bir kayda erişmek için ondan önceki tüm kayıtlar okunmalı veya yazılmalıdır.

Sıralı kütüklerde ancak belirli bir miktarda veri saklanabilir ve her erişimde istenen veriden önceki verilerin tümü birden okunur ya da yazılır. Tipik bir örnek, bir tablonun saklandığı kütükteki verilerdir.

Kayıtlar, fiziksel sıralanışa göre ardarda okunur veya yazılırlar. Yeni gelen her kayıt kütüğün sonuna yazılır. Kütüğün iç kısımlarına bir kayıt yazılmak istendiğinde ise tüm kütüğün yeniden yazılması gerekir. Yeni bir veri kaydını istenen herhangi bir konuma yerleştirmenin başka yolu yoktur. Kayıtların okunması da ancak sıralı bir biçimde yapılabilir. Kayıtları yeni bilgilere göre yeniden düzenlemek, yani kayıtları güncelleştirmek olanağı da kullanılan erişim yöntemine bağlıdır.

Dizinli kütükler: Bu kütüklerdeki verilere erişim bir dizin aracılığıyla ve doğrudan erişimle gerçekleştirilir. Sözkonusu dizin başka bir kütükle ya da aynı kütükte yer alabilir. Uygulamada sık sık kullanıldıklarından, dizinli kütüklerin yapısını biraz daha ayrıntılı olarak ele alacağız.

Bir dizinli kütüğe bağlı olarak iki ana veri dizilişi vardır: Dizin kütüğü ve veri kütüğü (bkz. s.250'deki şekil). Dizin kütüğünde sıralanmış olan kayıtlar, kütük kaydının anahtarından ve kaydın konumunu belirleyen veri göstergesinden oluşur. Her anahtar belirli bir veri kaydını tanımlar. Kütüğe farklı erişim olanakları sağlayan ikincil veya seçmeye bağlı anahtarlar da olabilir.

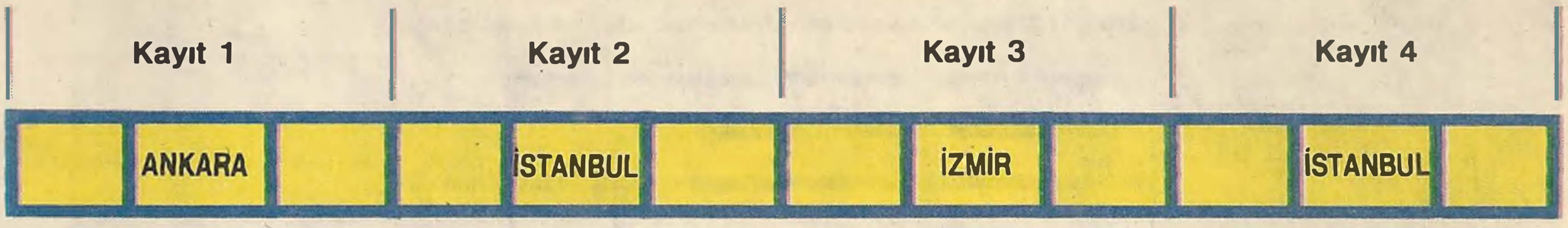
Bir dizin kütüğü içindeki bir anahtar tekrarlanarak da kullanılabilir. Örneğin kişilerin soyadı anahtar görevi yapıyorsa, aynı soyadını taşıyan birçok kişi bulunabilir. Her erişimde sadece tek bir değere ulaşılmak isteniyorsa, aynı anahtara sahip kayıtların, tüm kayıtlar içindeki sayısının tam olarak belirlenmesi tavsiye edilir.

Veri kayıtlarını içeren kütük, kullanılan tekniğe bağlı olarak anahtar değerine göre veya başka bir biçimde düzenlenmiş olabilir.

İkincil dizinler kullanılarak, birincil anahtarlardan farklı olan anahtarlar yardımıyla da kayıtlara erişilebilir. Örnek olarak bir müşteri kütüğündeki kayıtları ya müşterinin kod numa-

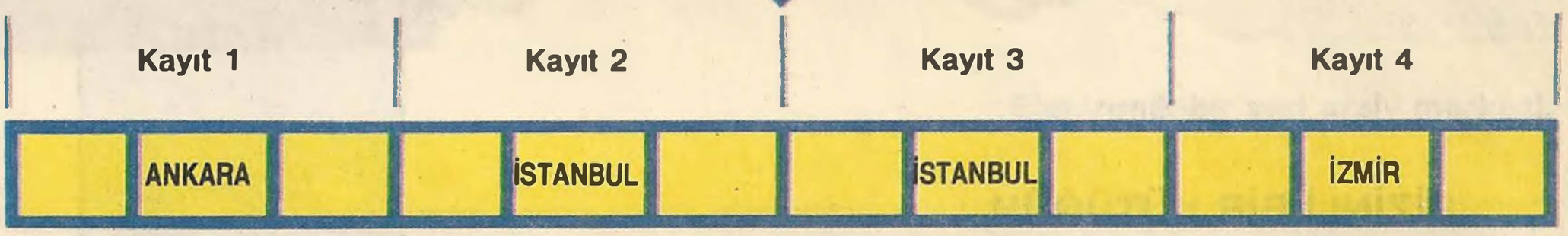
SIRALI BİR KÜTÜKTE ARAMA İŞLEMİNİN AKIŞ ŞEMASI

Kütüğün ilk durumu



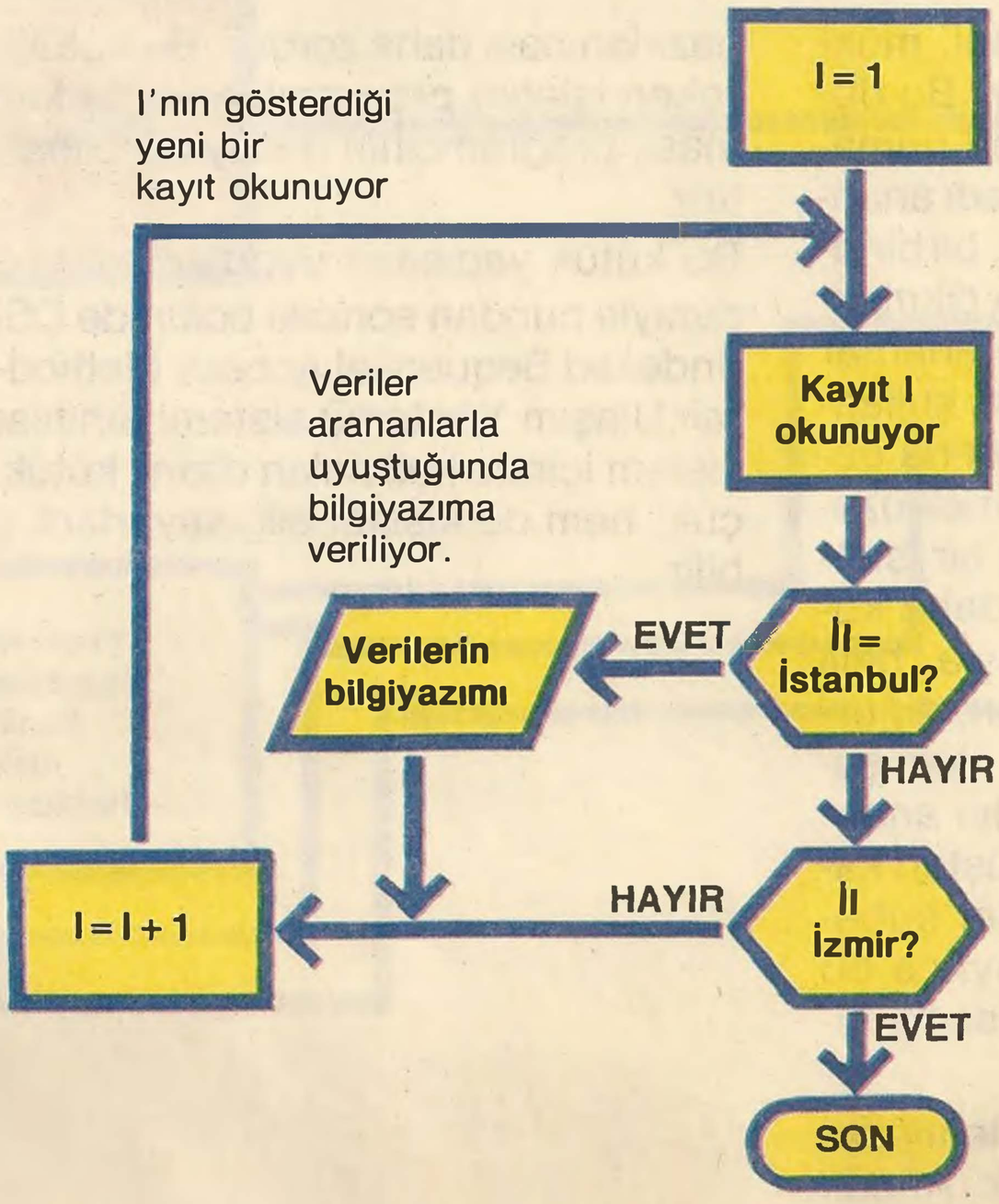
Veriler, kütüğün ilk durumunda veriliş sırasına göre yer alıyor. Herhangi bir ile ait tüm kayıtları elde etmek için kütüğün bütününü okumak gerekir.

Yeni sıralama alfabetiktir.



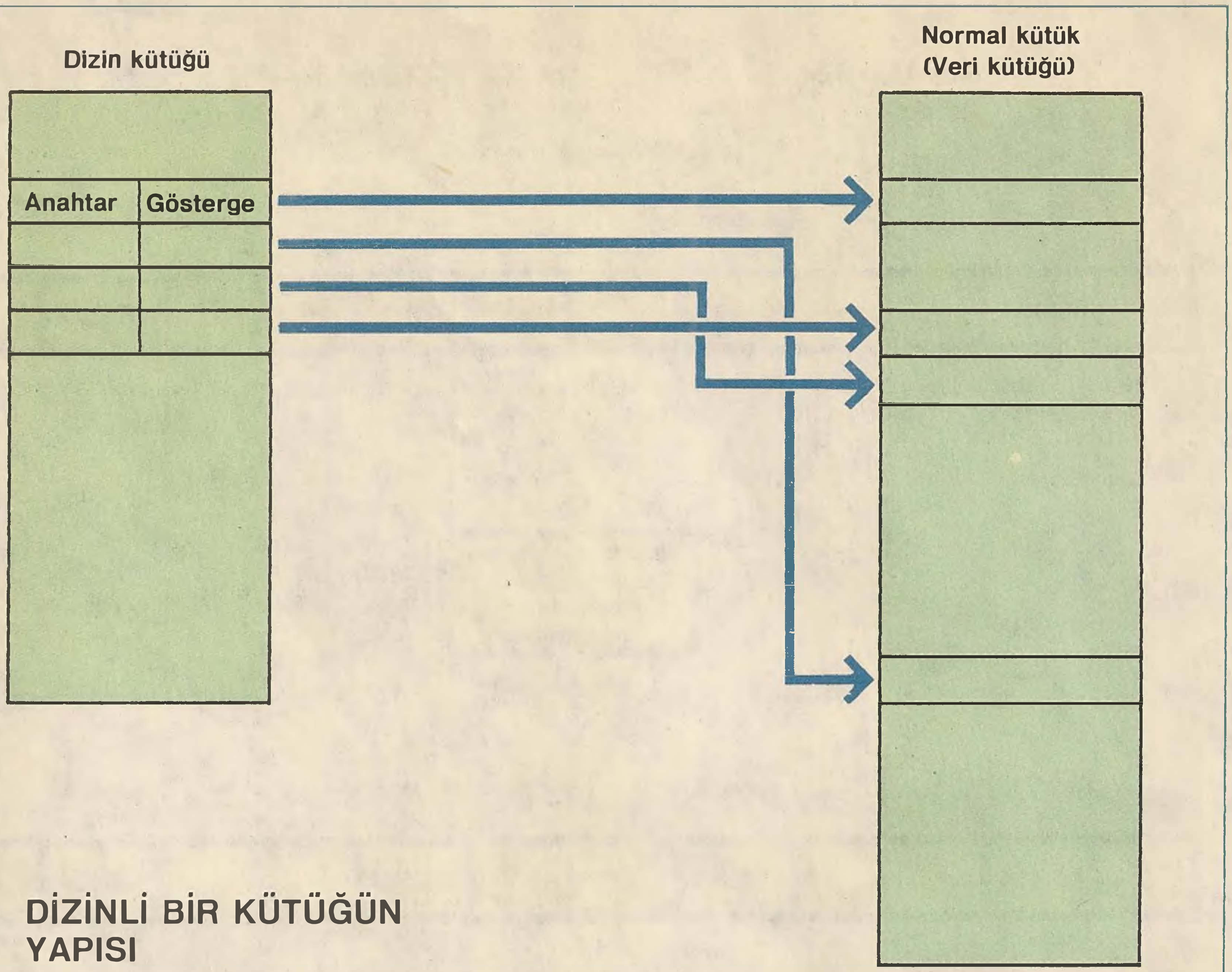
Sıralanmış alanlarla yeni kütük

Bu durumda aramaya kayıt 4'te son verilebilir, çünkü bu kayıttaki veri (İL = İZMİR) değeri aranan veriden (İSTANBUL) daha yüksektir. İl isimleri alfabetik sıraya göre dizildiklerinden, kayıt 4'ten sonra İSTANBUL yoktur.



Arama sürecinin başladığı ilk kayıt için gösterge

İstanbul'u içeren kayıtlar bitmiştir. Yürütmeye son verilebilir.



DİZİNLİ BİR KÜTÜĞÜN YAPISI

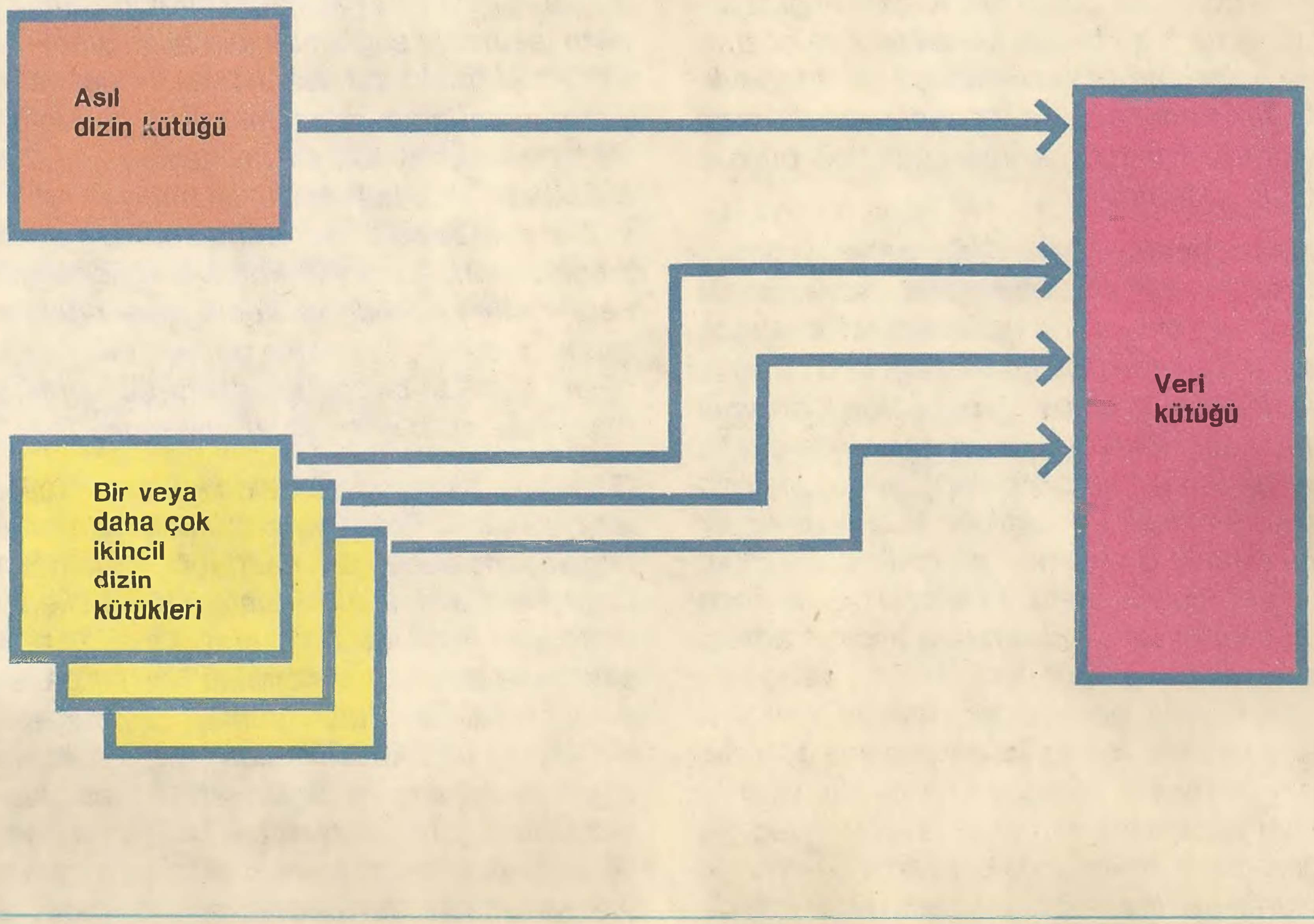
rasından hareketle okunabileceği gibi, müşterinin soyadı verilerek de okunabilir. Bu durumda birincil dizinde müşterinin kod numarası, ikincil dizinde de müşterinin soyadı anahtar görevini yapıyor. Böylece kütüğe, birbirinden bağımsız iki erişme biçimi ortaya çıkmaktadır (bkz. şekil). Bazen kayıtlar için anahtar olarak birden fazla değerin bileşkesini kullanmak elverişli olabilir. Bu amaçla iki ya da daha fazla veri, anahtar değerini elde etmek üzere birleştirilebilir. Örneğin çok şubeli bir işletme, her şubede tutulan hesapların daha kolay denetlenmesini sağlamak amacıyla, belli bir şubenin müşterilerine ait kayıtların, diğerlerinden ayrı saklanmasını isteyebilir. Hem şube kodu, hem de müşteri kodu kaydın anahtarı olarak kullanılırsa, bir şubenin müşteri kayıtları aynı grupta toplanabilir ve diğer şubelerden bağımsız olarak işlenebilir. Ayrıca bu düzen, müşteri kütüğünü geniş kapsamlı inceleme olanağını dışlamaz.

Dizinli kütükler çok farklı arama yöntemlerini mümkün kıldığından sık sık kullanılır. Ancak dizinli kütüklerin kullanıldığı veri arşivlerinin hazırlanması daha zordur. Bu kütükler için gereken işletim programlarının da karmaşık olması, programcının deneyimli olmasını gerektirir.

Bu kütük yapısının uygulamadaki önemi nedeniyle bundan sonraki bölümde DSEY (ISAM -Indexed Sequential Access Method-Dizinli Sıralı Ulaşım Yöntemi) sistemi tanıtılacaktır; bu sistem içinde kullanılan dizinli kütük, hem küçük, hem de kişisel bilgisayarlara uygulanabilir.

Elektronik bir veri arşiv merkezi.

İKİNCİL DİZİNLERİ OLAN BİR KÜTÜĞÜN YAPISI



Otomatik Bürolar

Büro, bugün hâlâ her endüstri ve ticaret kuruluşunun kalbidir. 50'li yıllardan bu yana üretimde yer alan işgücünün büyük ölçüde azalması, buna karşılık bürolarda çalışan personelin aynı oranda artması bu gerçeği vurgulamaktadır. Amerika Birleşik Devletleri'nde bugün bilgi işlem sektöründe çalışan insanların sayısı, başka herhangi bir işkolunda çalışanların sayısından fazladır. Bürolarda başlıca üç tür veri işlenir: Sayılar, şekiller ve sözcükler.

İlk aşama bürolara elektronik bilgi-işlem aygıtlarının girmesiyle gerçekleşti. Böylelikle kağıt ve yazılı belgelerden vazgeçilebileceği kanıtlanmış oldu. Bunu, bilgisayarla çizim ve metin işlem alanında devrim yaratan yenilikler izledi. Modern elektronik hesaplayıcılar, sayılarla uğraşmanın yükünü büyük ölçüde hafifletince de, muhasebeci ve satış yöneticileri daha önce hayal bile edemeyecekleri birtakım bilgi-işlem ve istatistik yöntemleri geliştirebildiler.

Uygun fiyatlı bilgisayarların ve minik işlemcilerin yaygınlığı otomatikleştirilen işlemler alanındaki gelişmelerin yönünü de gösteriyor. Kişisel bilgisayarların fiyatı, büyük bilgisayarlarınkine göre çok düşüktür. Kişisel bilgisayarlar çalışma hızı, bellek kapasitesi ve çözümleme olanakları bakımından büyük ağabeyleriyle yarışamasalar da, hesapların tutulması, ana defterlerin hazırlanması gibi tipik büro işleri için yeterlidirler.

Otomatik metin işlemcilerin geliştirilmesiyle otomatik veri işleme olanakları, yazılı metinleri de içerecek şekilde genişledi. Modern teknoloji sayesinde, bildiğimiz kağıtlara basılmış metinler çözümlenip mikroelektronik ortamlara taşınabilmektedir. Yakın gelecekte daktilo makinelerinin ''köle orduları'' yorucu işlerinden kurtarılacak ve daktilolar müzelerdeki yerlerini alacak. Bir memur otomatik metin işlemcinin yardımıyla, metni şifreleyip gizlilik önceliğini belirler ve bir klavyeden alıcının adresini vererek haberi gönderir. Haber, varış yerine ulaştığında, gelen diğer verilerle birlikte sıralanır ve bildirilen acillik derecesine göre sistemin belleğine yerleştirir. Alıcı kişi yine bir klavye yardımıyla veriler arasından istediğini seçer, metni bir ekranda okuyarak önemsiz kısımlarını ayırır ve diğer bilgileri, gerekirse değişikliklerle ve ek yorumlarla bilgisayarın yığın belleğine kaydeder.

Artık büroya kağıt gelmemekte, kağıtlar saklanmamakta, atılmamaktadır ve haber getirip götüren kuryelere de gerek kalmamıştır.

Tanıtım numarası bilinen herhangi bir haber, istendiğinde otomatik olarak bellekten alınır ve otomatik metin işlemci tarafından ekranda gösterilir. Böylece gerekli düzeltmeleri yaptıktan sonra kullanıcı istediği sayıda yanlışsız kopyayı bilgisayara hazırlatacak komutları verebilir. Bütün bunlar aynı sayfaları bir sekreterin daktiloda yazması için gereken süreden çok daha kısa bir sürede gerçekleşir. Günümüzde gönderilmesi istenen bir belge önce genellikle kağıda basılır; oysa aynı belge bir otomatik metin işlemcinin belleğinden bir telefon hattıyla doğrudan bir alıcı terminalin ekranına gönderilebilir.

Yeterli bellek hacmine sahip iletişim ağlarının kurulmasıyla kağıdın bürolardan tamamıyla kalkması mümkün görünüyor. Kodlanmış lazer darbelerini ileten fiber optik bağlantıları ya da iletişim uydularıyla, bu uygulamalar yaygınlık kazanabilir.

Bilimsel konularda uzmanlaşmış yayınevleri şimdiden, yazılı bilgilerin geleneksel depolanma ve hazırlanma şekillerini bırakıp zaman ve para tasarrufu sağlamak için bilgi-işlem merkezleri kurmaktadır. Bu merkezler sayesinde yazar, hazırladığı metni manyetik şeride kaydeder ve telefon yoluyla yayınevine iletir. Eser burada bir otomatik metin işlemciye verilir ve veri arşivinde saklanır. Daha sonra, merkeze ulaşma olanağı olan ve konuyla ilgilenen herkes, metni seçebilir ve kendi işyerindeki ekranda görebilir. Teknik ve bilimsel dergilerden çeşitli alıntılar birçok kitaplıkta bu yöntemle manyetik şeritlerde saklanmaktadır.

Gazetelerin yazı işlerinde de köklü değişiklikler olmuştur. Ekran ve terminallerle merkezi bilgisayara bağlı olan muhabir, redaktör ve dizgiciler, veri işleme sisteminin belleğiyle doğrudan ilişki kurarak, bütün metinleri, resimleri ve başlıklarıyla birlikte tüm bir gazeteyi çıkarabilirler. Bütün bunlar, gerekirse tek bir sayfa kağıt kullanılmadan yapılabilir. Hazırlanmış gazetenin, bilgisayarın içinde oluşan elektronik görüntüsüyle baskı işlemine geçilir. Bu, haberleri doğrudan ekranlara gönderme seçeneğinden önceki son adımdır. Bir

sonraki adımda haberler halka açık yerlerde veya evdeki televizyon alıcısında gösterilebilir. Böylece okurlar kahvaltı sırasında en yeni haberleri alabilir, istediklerini kendi manyetik bantlarına kaydedebilirler.

Şimdiden bazı büyük yayınevleri özel uzmanlık alanlarına hitap eden bazı konularda böyle elektronik bültenler yayınlamaya başladı. Günümüzde bu uygulama iş dünyasını ilgilendiren borsa haberleri gibi haberleri içeren bültenlerle ve kimi bilimsel araştırma raporları ile sınırlı. Ancak yayınevleri bu elektronik yayıncılığın sınırlarını ve ticari olanaklarını genişletmek için hummalı bir faaliyet halinde.

Yeni bir ''iş istasyonu'' tasarısında büyük bir büro içinde, bir veya iki kişilik, sabit duvarlarla ayrılmayan küçük birimler kurulması öngörülmektedir. Bu birimlerin her birinde iş için gerekli malzeme ile bilgisayara ve işyerinin santraline erişmeyi sağlayan ekran, terminal, arabirim gibi aygıtlar bulunur. Her iş istasyonuna ayrılan alan, belirli ölçüde kişisel rahatlık sağlamak amacıyla insan yüksekliğinde bir bölmeyle kapatılmıştır. Bilgisayar yardımıyla yürütülen ayrıntılı araştırmalar, hangi çalışma koşullarında en verimli çalışmanın yapılabileceğini de saptamamızı mümkün kılıyor. Bu koşullara örnek olarak ışıklandırma derecesi ve yazı masasının kalorifere uzaklığı sayılabilir.

Bilgisayarın birçok bölümü gürültüsüz çalışır, ancak bazı parçalar insan kulağının duyamadığı, ama rahatsızlığa neden olabilen yüksek frekanslı gürültüler yayar. Bu gibi etkilerin de gözönünde bulundurulması gerekir. Kağıdın kullanılmadığı bir büroda insan imzasının yerini de doğal olarak yasal belgeler tutar.

Bunun dışında kamuoyunun elektronik belleğin güvenirliliğine inanması için, insan hatalarının bilgisayara girip orada toplanarak büyük olumsuz sonuçlara yol açması önlenmelidir. Kısa süre önce İngiltere'deki bir fabrikadan bir görevlinin hatası yarım milyon dolarlık bir zarara neden olmuştu. Ancak bu türden sorunların tarihe karışması uzun sürmeyecektir. Görünüşe bakılırsa, uluslararası ve kamu telekomünikasyonunun ulaştığı dev boyutların ve çok yönlü veri arşivlerinin yarattığı sorunlar, çalışma yaşamının yeniden düzenlenmesinin yarattığı sorunlardan daha önce çözümlenecek. Bu düzenlemelere bağlı olarak arşiv memurları, sekreterler ve kuryeler başka işlere kaydırılacak ve bu görevlilerin sayısı azalacaktır.

Bilgisayarların girişi, bir ''açık'' büro için yeni olanaklar yarattı.

Yöneticilerin yeni kavramlara ve gelişmelere alışmaları gerekiyor. Bu alışma sürecinin psikolojik birtakım değişiklikler de içereceği açıktır. Çağdaş gelişmeleri uygulayan bürolarda çalışanlar geleneksel hiyerarşilerde belirli bir çözülme görüyorlar.

Elektroniğin etkisiyle amirler her gün biraz daha fazla otorite kaybederken, ekranın otoritesine karşı yeni bir tür ''saygı'' doğuyor.

Dizinli Kütüğün İşletimi (DSEY Sistemi)

Dizinli kütükte kayıtlara erişmek için hem sıralı, hem de doğrudan erişim yöntemleri kullanılabilir. Her iki yöntemde de temel öge ya kaydın anahtarı ya da belirli bir göstergedir.

Her iki yöntemin de uygulama türüne göre kendine özgü yararları vardır. Anahtarlı bir kütük, özellikle kayıtlara sıralı ya da doğrudan erişildiği durumlarda yararlıdır.

Anahtarlı bir kütük diğer organizasyon şekillerine göre daha esnektir. Ayrıca üstün nitelikler gerektiren etkileşimli ve gerçek zaman işletmelerine de uygundur.

Anahtarlı kütük kullanımına özellikle elverişli bir uygulama, veri arşivlerinde yapılan aramalardır. Bu aramalarda okunacak verileri içeren veri kaydı yalnızca anahtar yardımıyla aranır.

Örneğin bir stok kontrol sisteminde, bir siparişi sonuçlandırmadan önce ilgili malın stok düzeyini bilmek gerekir. Mal cinsi kodunun anahtar olarak kullanıldığı bir kütükten yararlanmak için kullanıcının hazırladığı program, mal cinsi kodu sayesinde ilgili kaydı bulur, okur ve ekranda gösterir. İkinci bir dizin de kullanılmak isteniyorsa, her mal cinsi için ikinci bir anahtar saptanır (örneğin malın tanımı). Malın kodu bilinmiyorsa tanımı verilerek, o mala ya da aynı tanıma uyan diğer mallara ilişkin kayıtlar bulunur.

Anahtarlı kütüklerin yararlı olduğu başka bir kullanım alanı da veri denetimidir. Bu uygulamada verinin değeri anahtar işlevi görür. Kullanıcının verdiği değer, kütükteki anahtar değerle karşılaştırılır. Böyle bir anahtar yoksa, durum kullanıcıya bildirilir ve bu durumda kullanıcı yeni bir değer girer. Örneğin bordro hesapları için hazırlanmış bir sistemde her kayıt, haftalık çalışma saatleriyle birlikte işçinin *sicil numarasını* içerir. *Sicil numarasını* kullanıcı belirler ve bu numara işçinin kaydına erişirken anahtar olarak kullanılır. Kullanıcı yeni bir sicil numarası girdiğinde, o numarayla aynı bir anahtar değeri bulunamamışsa aygıt bunu kullanıcıya bildirir ve kullanıcı bu kez doğru değeri makineye verir.

Bilgisayarın kullanılmasıyla, bir arşivin geleneksel görüntüsü değişmekle kalmadı, şimdi daha düzenli bir yerleşimle arama işlemleri daha çabuk sonuçlandırılıyor.

Verilere Erişme İşlemleri

Dizinli kütükte bir kaydı okumak için beş farklı yöntem kullanılabilir (bu noktadan itibaren dizinli kütük, anahtarlı kütükle eşanlamlıdır):

- **Anahtar üzerinden doğrudan erişim:** Bu yöntemde bir anahtar değerinin verilmesinden sonra, önce gösterge değeri, sonra da gösterge değeri yardımıyla aranan kayıt bulunur. Bir kez anahtar belirlendikten sonra geriye kalan süreç dizin kütüğünde kaydın göstergesinin aranmasıdır. Arama olumlu sonuçlanırsa gösterge bildirilir, aksi halde hata işareti gönderilir.

  Anahtar üzerinden doğrudan erişim yöntemi, özellikle anahtarı bilinen belirli bir kaydı değiştirerek kütüğü güncelleştirmek istendiğinde yararlıdır.

- **Anahtarla genel okuma:** Bu da öncekine benzeyen bir yöntemdir. Aradaki fark, aranan kayıt bulunamadığında, bir sonraki anahtar değerinin göstergenin yerini almasıdır.
  Bu yöntemi kısmi anahtarlarla uygulayarak, kayıtlar mantıksal gruplara ayrılabilir. Yöntem ayrıca kayıtların elenip tasfiyesi ve düzenlenmesi amaçlarına da uygundur.

- **Anahtarla sıralı okuma:** Kayıtların, kütükteki fiziksel sıralanıştan bağımsız olarak, anahtar değerlerinin yardımıyla mantıksal sıraları içinde okunmalarını sağlayan bir yöntemdir. Okumaya kütüğün birinci mantıksal kaydıyla başlanabileceği gibi başka herhangi bir noktadan da başlanabilir. Kayıtlara belirli bir sıraya göre erişmek istendiğinde bu yöntem uygulanır. Kullanılacak örneği anahtar değeri belirleyebilir. Genel okumayla aynı anda uygulandığı takdirde mantıksal kayıt grupları da okunabilir.

- **Fiziksel sıralı okuma:** Kütükteki kayıtlar, dizin kütüğünü kullanmaksızın fiziksel sıralanışa göre okunur. Bu yöntem, toplam bilgi işleme sürecinde kayıtların okunma sırası, asıl veri çözümleme işlemleri ile kıyaslandığında önemsiz bir yer tutuyorsa, yarar sağlar. Dizin kütüğünde arama işlemi kalktığından hatırı sayılır bir zaman kazanılır.

- **Kayıt numarasıyla doğrudan okuma:** Aranan kaydın numarası kullanılarak verilere doğrudan erişilir. Böylelikle, kütükteki verilere ya doğrudan erişim yöntemiyle ya da bir anahtarla ulaşmak mümkündür.

Veri girme ve silme işlemlerinde de çeşitli yöntemler kullanılır.

Dizinli bir kütükte kayıtlar veya anahtarlar temel yapıdan kaldırılabilir. Böylece kayıtları silmek için bir program hazırlamaya, kütüğü yeniden düzenlemeye gerek kalmaz ve manyetik diskte yer kazanılır.

Bir silme işlemi, bir kütükteki belirli kayıtları iptal eder. Bu yöntemle, silinen kayda ait anahtar, dizin kütüğünden silinir. Yeni kayıtlar eklenmek istenirse, anahtarı silinen kayıt yeniden kullanılabilir. Yalnızca kaydın anahtarı silinip kaydın kendisi korunmak isteniyorsa, anahtar silme işleminin uygulanması yetecektir. Bu işlem, anahtarı dizin kütüğünden kaldırır, kaydı ise değiştirmez. Anahtarı silinen kayıt, sıralı erişimle okunabilir. Bir veri kaydının kendisini ise ancak programcı silebilir. Arşiv hazırlamada olduğu gibi kayda erişimin birbirini izleyen adımlarla gerçekleştiği uygulamalarda anahtarın silinmesi çok yararlıdır. İkincil dizinli kütüklerde, silinmiş kayıtların anahtarlarını iptal etmek için de bu yöntemden yararlanılır.

Dizinli kütüklerin en yararlı oldukları kullanım alanı, verilerin gerçek zamanda (anında veri işleme ve çıkışı) yeni duruma göre değiştirilebilmesidir. Bir kayıttaki veriler, kullanıcı programının gelişmişlik derecesine bağlı olarak değiştirilebilir. Veriler böylece gerçek zamanda anlamlı bilgilere dönüşür. Bir verinin yenilenmesi, normal olarak kaydın okunması sırasında gerçekleşir.

Örneğin bankalarda para yatırma ve çekme işlemleri bu yöntemle doğrudan müşterinin hesabına kaydedilir; bir anahtarın (hesap numarasının) girişi yapılır, ilgili miktar toplama eklenir veya ondan çıkarılır ve cari hesaplar kütüğündeki kayıt yeniden yazılır. Bundan sonraki işlemlerde hesabın yeni durumu gözönüne alınır.

DSEY (ISAM) Kütüğünün Genel Yapısı

DSEY (ISAM) adı verilen dizinli sıralı erişim yönteminde iki veri yapısı kullanıldığını söylemiştik: Dizin kütüğü ve veri kütüğü. Kayıt erişimi için birden fazla anahtar belirlenmişse, bu anahtarların yer aldığı başka kütüklerden de yararlanılabilir. DSEY kütüklerine BASIC diliyle erişilir ve BASIC'le yapılabilecek bütün işlerde kullanılabilirler. Dolayısıyla sıralama (sort) ve birleştirme (merge) gibi hizmet programları, herhangi bir değişiklik gerektirmek-

sizin, DSEY kütüklerine uygulanabilir. Sıralama programı verilerin düzenlenmesine yarar, birleştirme programı ise birçok kütüğü bir araya toplayarak bir tek kütükten oluşan bir veri arşivi oluşturur. Değinilmesi gereken bir nokta, bu programların veri ile dizin kütüğü arasındaki mantıksal bağıntıları bozabileceğidir.

Veri kütüğü doğrudan erişimli bir BASIC kütüğüdür. Bu nedenle kendisiyle ilgili BASIC komutlarına yanıt verir.

Dizin kütüğünü oluşturan kayıtların her birinde belirli sayıda anahtar vardır. Her anahtarın bir kayıt numarası vardır. Çift anahtarlı dizin kütüğünde arama mekanizması daha değişik olduğundan bu kütükler özel bir yöntem gerektirir. Böylece dizinli kütüklerin iki türünün olduğunu görmüş oluyoruz: Çift anahtarla kullanılabilenler ve yalnızca basit anahtarların yer aldığı kütükler.

Her iki durumda da anahtarlar fiziksel bir sıralamaya göre saklanmazlar, harfsayısal bir sıralamayla göstergelere bağlanırlar. Bu sıralanışa ''ağaç yapısı'' adı verilir. DSEY kütüğünde kullanılan ağaç yapısına ''iki tabanlı ağaç'' denir.

İki Tabanlı Ağacın Yapısı

DSEY'de kullanılan ağacın yapısının ayrıntılarına girmeden önce, işleyişini daha iyi anlamak için basit bir biçimini inceleyelim.

İki tabanlı ağacın bu basit biçiminde her anahtar ağacın bir dallanma noktasını gösterir ve her dallanma noktası iki göstergeyle bağlantılıdır. Bunlardan birincisi bir önceki anahtarı, ikincisi de bir sonraki anahtarı gösterir. Bu iki anahtar bir alt kademededir. Aşağıdaki şekil, önceden varolan anahtar 7'ye, 4,5 ve 9 anahtarlarının eklendiği iki tabanlı bir ağacı gösteriyor.

Kütüğe her yeni kayıt eklenişinde, eklenen kaydın anahtarı, ''ağacın tepesi''ndeki, yani ağacın en üst noktasındaki, anahtarla karşılaştırılır. Bu değere ağacın ''kök''ü adı da verilir. Yeni gelen kaydın anahtarı kökten büyükse, kökün sağındaki göstergenin anahtarıyla karşılaştırılır. Yeni kaydın anahtarı kökten küçükse, karşılaştırma kökün solundaki göstergeyle yapılır.

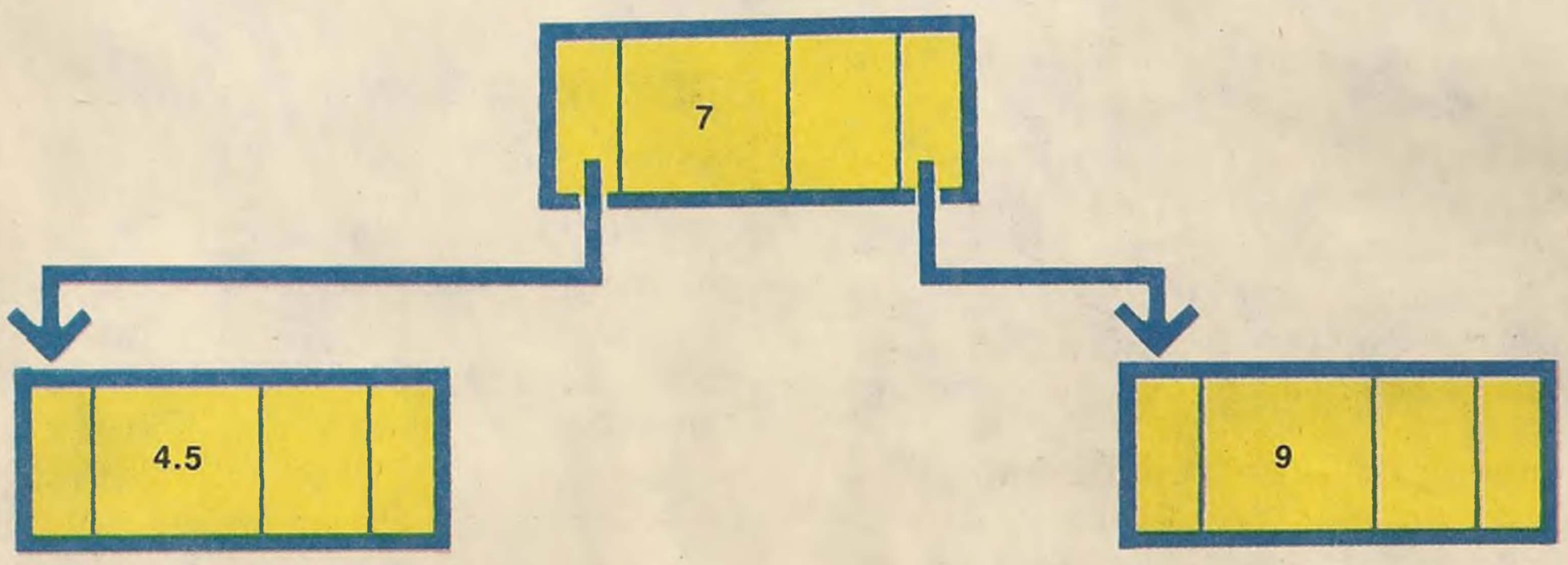
Anahtarın doğru konumu bulununcaya kadar karşılaştırmalar, ağacın en alt kademesi yönünde böylece sürer. Örnek olarak sayfa 256'daki yapıya 2, A3, A5, 4, 5A ve 9,9 anahtarlarını eklemek istersek, sayfa 257'deki şekle benzer yeni bir yapı elde ederiz.

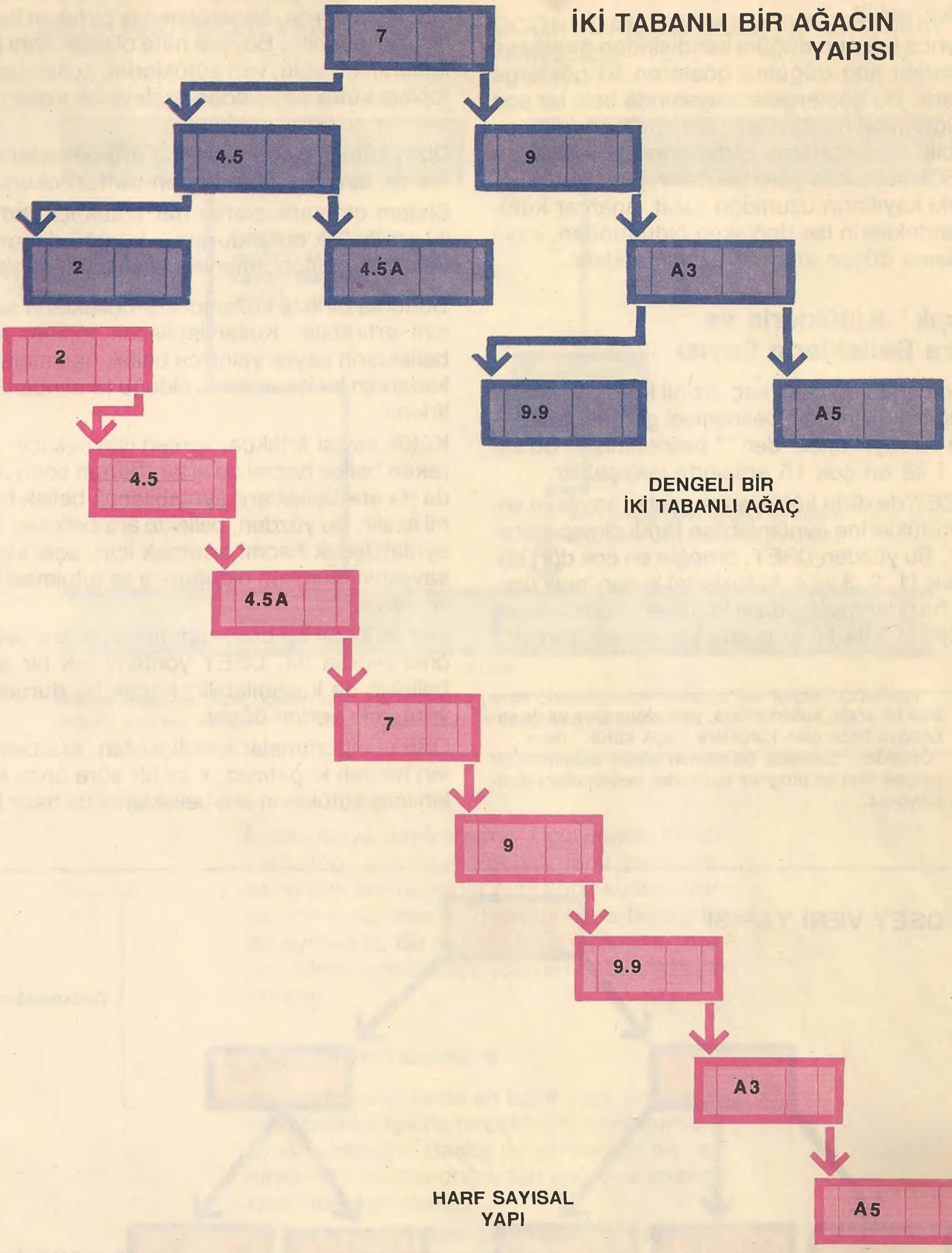
Ağaç dengede kaldıkça, yani bir dallanmanın sol tarafındaki anahtar sayısı sağ tarafındakilerin sayısına eşitse, herhangi bir anahtarı arama süresi eniyilenmiş olur. Örneğin sayfa 257'deki üst şekilde A5 anahtarına 7, 9 ve A3 düğümleri (dallanmaları) incelendikten, yani üç karşılaştırma yapıldıktan sonra erişilir. Anahtarlar harfsayısal sıralamaya göre verilirse, sayfa 257'nin altındaki gibi bir yapı elde edilir. Bu durumda A5 anahtarına erişmek için yedi karşılaştırma yapmak gerekir. Yapının dengesizliğinden kaçınmak için ağaç uygun algoritmalarla kırılarak **(splitting)** dengeli bir biçime getirilir.

DSEY Ağaç Yapısı

DSEY ağaç yapısı bir önceki bölümde açıklandığı kadar basit değildir. DSEY'de iki tür ka-

İKİ TABANLI BİR AĞAÇ YAPISINA ÖRNEK





İKİ TABANLI BİR AĞACIN YAPISI

DENGELİ BİR İKİ TABANLI AĞAÇ

HARF SAYISAL YAPI

yıt yer alır: Dallanmalar ve son düğümler **(nodes).** Ağacın en alt kademesinde bulunan son düğümlerde tüm anahtarlar ve karşılıkları olan kayıt numaraları vardır. Dallanmalar ise en alt kademe dışındaki bütün kademelerde yer alır

ve son düğümlere erişme yolunu gösterir (bkz. şekil).

Ayrıca her son düğüm kendisinden önceki ve sonraki son düğümü gösteren iki gösterge içerir. Bu göstergeler sayesinde belli bir son düğümden hareketle bütün anahtarlara erişilebilir. Anahtarların birbirlerine göre sıraları ASCII kodlarına göre belirlenir. Dizin kütüğündeki kayıtların uzunluğu sabit, anahtar kütüğündekilerin ise değişken olduğundan, kayıt başına düşen anahtar sayısı farklıdır.

Açık* Kütüklerin ve Ara Belleklerin Sayısı

Aynı anda en fazla kaç dizinli kütük açılabileceğini kullanıcının belirlemesi gerekir; DSEY'de bu sayı ''önce''den** belirlenmiştir. Bu sayı 1 ilâ en çok 15 arasında değişebilir.

DSEY'de dizin kütüklerine ayrılan sayıların veri kütüklerine ayrılanlardan farklı olması gerekir. Bu yüzden DSEY, örneğin en çok dört kütükle (1, 2, 3 ve 4. kütüklerle) kullanılmak üzere hazırlanmışsa, dizin kütüklerini gösterecek sayının 5 ilâ 15 arasında yer alması gerekir.

* Belli bir anda, kullanılmaya, yani okunmaya ya da yazılmaya hazır olan kütüklere ''açık kütük'' denir.

** ''Önceden'' demekle, bu sayının işletim sisteminin bir parçası olan bir program tarafından belirlendiğini kastediyoruz.

Bir BASIC programı 1, 2, 3 ve 4 sayılarını kullandığı takdirde, öngörülmemiş birtakım hatalara yol açabilir. Böylesi hata olasılıklarını önlemenin bir yolu, veri kütüklerini, kullanılacak toplam kütük sayısından başlayarak azalan bir sıra ile numaralamaktır.

Dizin kütüklerindeki kayıtlar ara-belleklerden (ya da tampon belleklerden-**buffer**) okunur.

Sistem otomatik olarak her kütük için en az iki ara-bellek bulundurarak, kütüğü dengeleyecek olan algoritmanın çalışmasını sağlar.

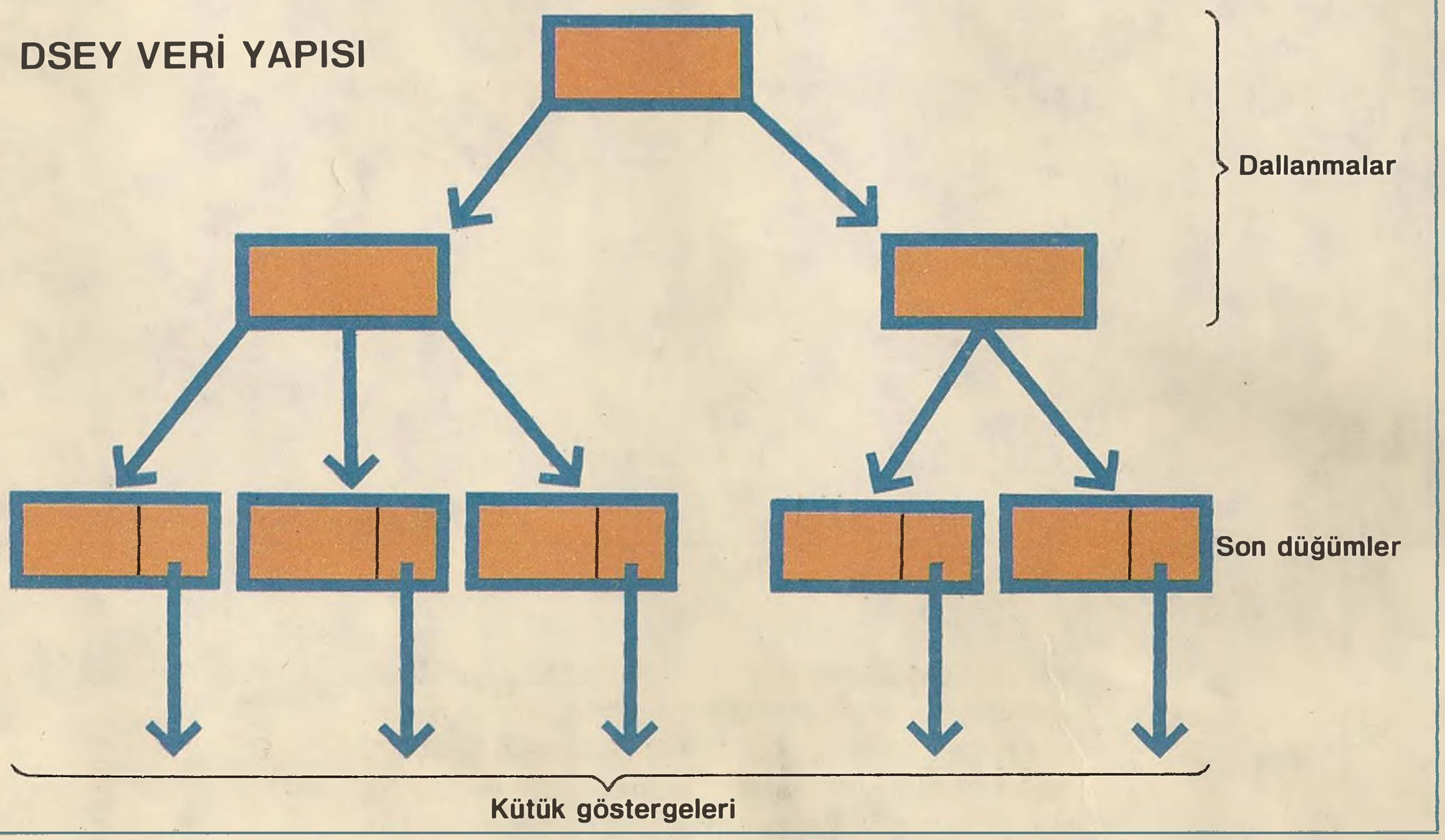
Bununla birlikte kullanıcı ara-belleklerin sayısını artırabilir. Kullanılabilecek toplam ara-belleklerin sayısı yalnızca bellek hacminin ne kadarının bu işe ayrılmış olduğu tarafından belirlenir.

Kütük sayısı arttıkça, verileri işlemek için gereken bellek hacmi de artar; bunun sonucunda da ara-belleklere ayrılabilecek bellek hacmi azalır. Bu yüzden, bellekte ara-bellekler için ayrılabilecek hacmi artırmak için, açık kütük sayısının mümkün olduğunca az tutulması salık verilir.

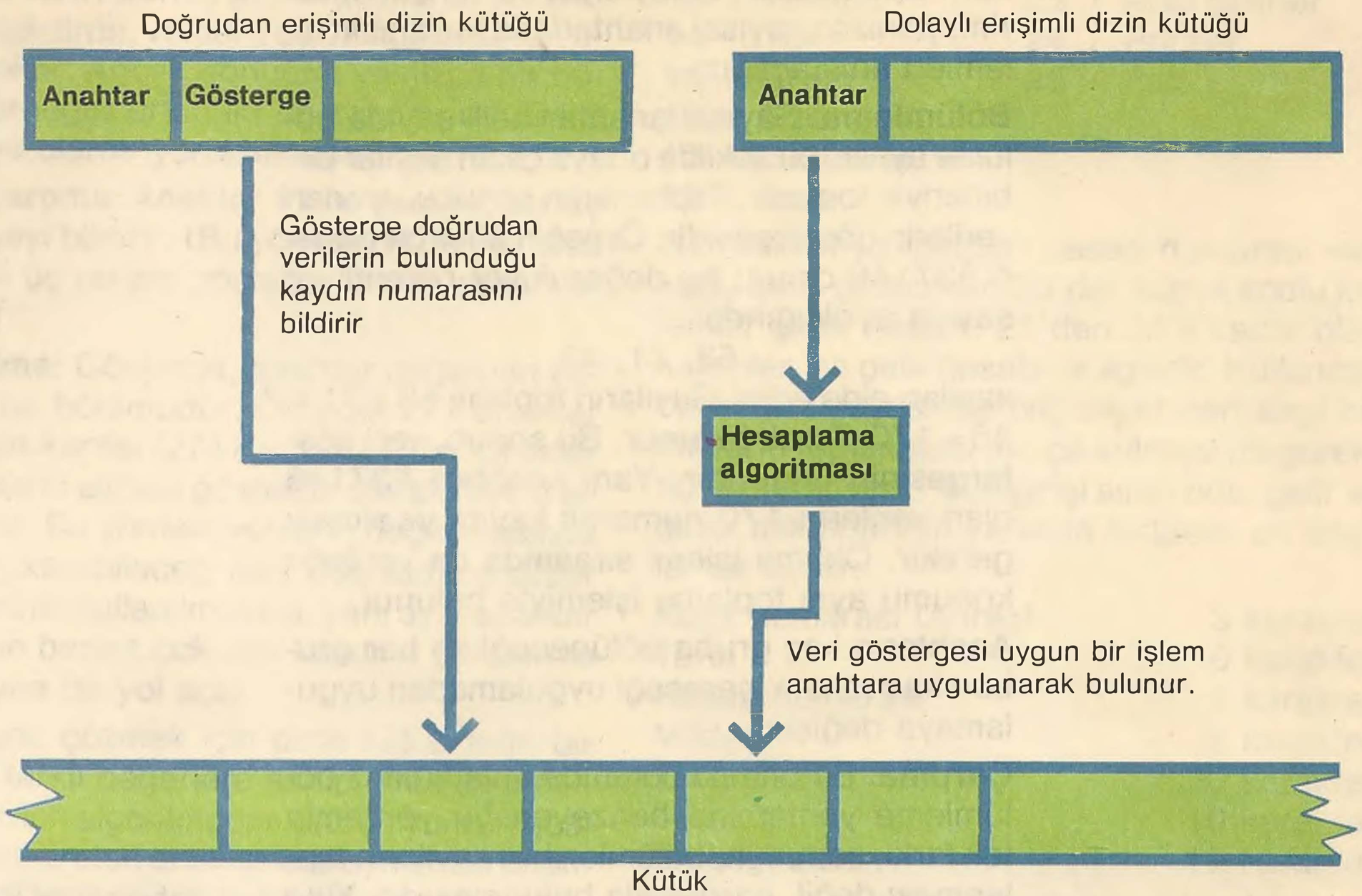
Her ne kadar bir dizin kütüğü için iki ara-bellek öneriliyorsa da, DSEY yöntemi tek bir ara-bellekle de kullanılabilir; ancak bu durumda yöntemin verimi düşer.

DSEY, algoritmalar içerdiğinden, ara-bellekleri hemen kapatmaz; kısa bir süre önce kullanılmış kütüklerin ara-belleklerini de hazır bu-

DSEY VERİ YAPISI



DİZİNLİ KÜTÜKLE VERİ İŞLEMEDE DOĞRUDAN VE DOLAYLI ERİŞİMİN KARŞILAŞTIRILMASI



Dolaylı erişimde, dizin, göstergeleri saklamasına gerek olmadığından daha az yer kaplar. Gösterge değeri, anahtar değerinin kendisine uygulanan bir algoritmayla bulunur.

lundurmaya devam eder. Dolayısıyla, bir dizin kütüğü için bir ara-bellek daha gerekiyorsa ve tüm ara-bellekler aynı anda kullanılıyorsa, son kullanılmış olan kütüğün ara-belleği bu işe ayrılabilir. Bu sayede DSEY yöntemi, ara-belleklere ayrılan hacimden en iyi biçimde yararlanır.

Dolaylı Adresleme

Anahtarlı kütüklerde en basit yapı, her anahtarın belli bir (ya da birçok) veri kayıt numarası saklamasıdır. Başka bir yöntemde ise veri kaydının konumu doğrudan doğruya anahtar tarafından gösterilir.

En karmaşık yöntem olan doğrudan adreslemede ise dizin kütüğünde yerden tasarruf etmek için veri kayıt numaralarına dizin kütüğünde yer verilmez.

Şekil her iki sistemi de gösteriyor: Birinci sistem daha önce incelemiş olduğumuz dizinli kütük sistemi; burada aranan veri dizin kütüğündeki göstergeler yardımıyla bulunuyor. Dolaylı adresleme ise verilere erişmek için ön-

ce anahtar değerinin bir işlemden geçirilmesi gerekir.

Bu yöntemde adres hesaplama işlemleri çok karmaşık olabilir. Dolayısıyla biz en önemlilerini, yalnızca sayısal anahtarlarla kullanılan işlemleri anlatacağız.

Bölümleme: Sayısal anahtar belli sayıda bölüme ayrılır, bu şekilde ortaya çıkan sayılar birbirleriyle toplanır. Toplamanın sonucu, aranan verilerin göstergesidir. Örneğin anahtar değeri 537146 olsun; bu değer ikişer rakamlı üç sayıya ayrıldığında;

53, 71, 46

sayıları elde edilir. Sayıların toplamı 53 + 71 + 46 = 170 olarak bulunur. Bu sonuç, veri göstergesinin değeridir. Yani anahtarı 537146 olan verilerin 170 numaralı kayıta yazılması gerekir. Okuma işlemi sırasında da verilerin konumu aynı toplama işlemiyle bulunur.

Anahtarın kaç gruba bölüneceği ve her grubun kaç rakam içereceği uygulamadan uygulamaya değişir.

Çarpma: Bir önceki bölümde anlattığımız bölümleme yöntemine benzeyen bu yöntemin tek farkı, sonucun, bölümlerdeki sayıların toplanması değil, çarpımıyla bulunmasıdır. Yine

Veri bankalarında dev hacimli yığın belleklerden yararlanılır.

537146 örneğine dönelim ve bu kez bu sayıyı üçer rakamlı iki gruba, 537 ve 146'ya ayıralım. Bunların çarpımı 537 × 146 = 78402 değerini verir. Sonuç gösterge olarak yorumlandığı takdirde, veriler 78402 numaralı kayıtta demektir. Ancak sonucun yalnızca bir bölümü, örneğin ortadaki üç rakamı da (840) gösterge olarak yorumlanabilir.

11 ile çarpma: Anahtar 11 ile çarpılır, sonuç göstergeyi bildirir. (Bu yöntemde de yalnızca ortadaki üç rakam gösterge olarak yorumlanabilir.)

Kare alma: Gösterge, anahtar değerinin karesinin bir bölümüdür. Örneğin 271 anahtar değerinin karesi $(271)^2 = 73441$'dir. Ortadaki üç rakamı alırsak gösterge olarak 344'ü elde ederiz. Bu yöntem verilerin dağıtılmasında güçlük çıkarabileceği gibi, eşanlamlı anahtar değerlerinin kullanılmasına, yani aynı anahtar değerinin birden çok veri kaydına gönderme yapmasına da yol açar.

Bu sorunu çözmek için dizin kütüğünün bir bölümü farklı değerlere sahip olmakla birlikte, kullanılan algoritmadan ötürü aynı veri göstergelerini üreten anahtarlara ayrılır. Bu anahtarların diğerlerinden farklı bir işlemden geçirilmesi gerekir. Verilerin dağıtımı ile ilgili sorun ise kullanılan algoritmanın sayısal değerleri birbirlerininkine çok yakın olan veri göstergeleri üretmesinden doğar. Bunun sonucu olarak bir kısmı tamamen boşken, diğer bir kısmı tıkış tıkış dolu olan birtakım kütükler üretilebilir. Böyle bir durumda yapılacak tek şey algoritmayı değiştirmektir.

Bir Uygulama Örneği: Aile Bütçesi

Buraya dek anlatılan kütüklerin uygulama olanaklarını incelemek için aile bütçesi için başvurulabilecek bir veri arşivinin nasıl hazırlandığını ve kullanıldığını inceleyeceğiz.

Birinci adımda (çözümleme) hangi sonuçları elde etmek istediğimizi ve bu sonuçlara ulaşmak için hangi verilerin gerekli olduğunu saptıyoruz.

Hesap Planı

Bir bütçe hazırlanırken önce gelir ve gider kalemlerinin saptanması gerekir. Sonra gelir (maaş vb.) ve giderin (kira, elektrik, gıda, giyim vb.) cinsi tanımlanır.

Şimdi tüm bu verileri "**hesap planı**" adını vereceğimiz şu tabloda gösterelim:

Hesap cinsi:

1 / Gelir
2 / Gider

Gider kalemleri (10)	**Gelir kalemleri (20)**
11 / Kira	**21** / Maaş
12 / Elektrik, su	**22** / Diğer ücretler
13 / Telefon	**23** / Arızi gelirler
14 / Yakıt	**24** / Primler
15 / Giyim	**25** / Faiz
16 / Eğlence	
17 / Gazeteler	
18 / Gıda	

Her kalemin yanındaki hesap numarası veri kodu işlevi görecektir. 20'den küçük kodlu kalemler gider hesabı, 21'den 25'e kadar olan kalemler ise gelir hesabı ile ilgilidir. Kullanılan bu kodlar sayesinde bilgisayar herhangi bir miktarın toplanması mı, çıkarılması mı gerektiğine karar verir. Veri girişi sırasında, gelir ve gider miktarlarının yanında birtakım ek bilgiler de verilir:

Kayıt numarası (Adres)	3 karakter
Tarih	6 karakter
Hesap numarası	2 karakter
Miktar	8 karakter
Yorum	20 karakter
İlgili kişi	10 karakter
Ek bilgi alanı	11 karakter

Şimdi bu alanların anlamını açıklayalım:

Kayıt numarası (Adres): Bu alanda veri giriş işleminin sıra numarası belirtilir. Adresin başlangıç değeri 1'dir ve her yeni veri girişinde bu değere 1 eklenir. Adres değeri hem veri kayıt numarasını, hem de bu verilere erişmede kullanılacak anahtarı verir. Hatalardan kaçınmak için bu değerin bilgisayar tarafından otomatik olarak belirlenmesi gerekir. Yani programın, adresin son değerini bellekte bulundurması ve her yeni veri girişinde bu değere 1 eklemesi gerekir. Böylelikle hep son girilen verinin adresi bellekte kayıtlı tutulur.

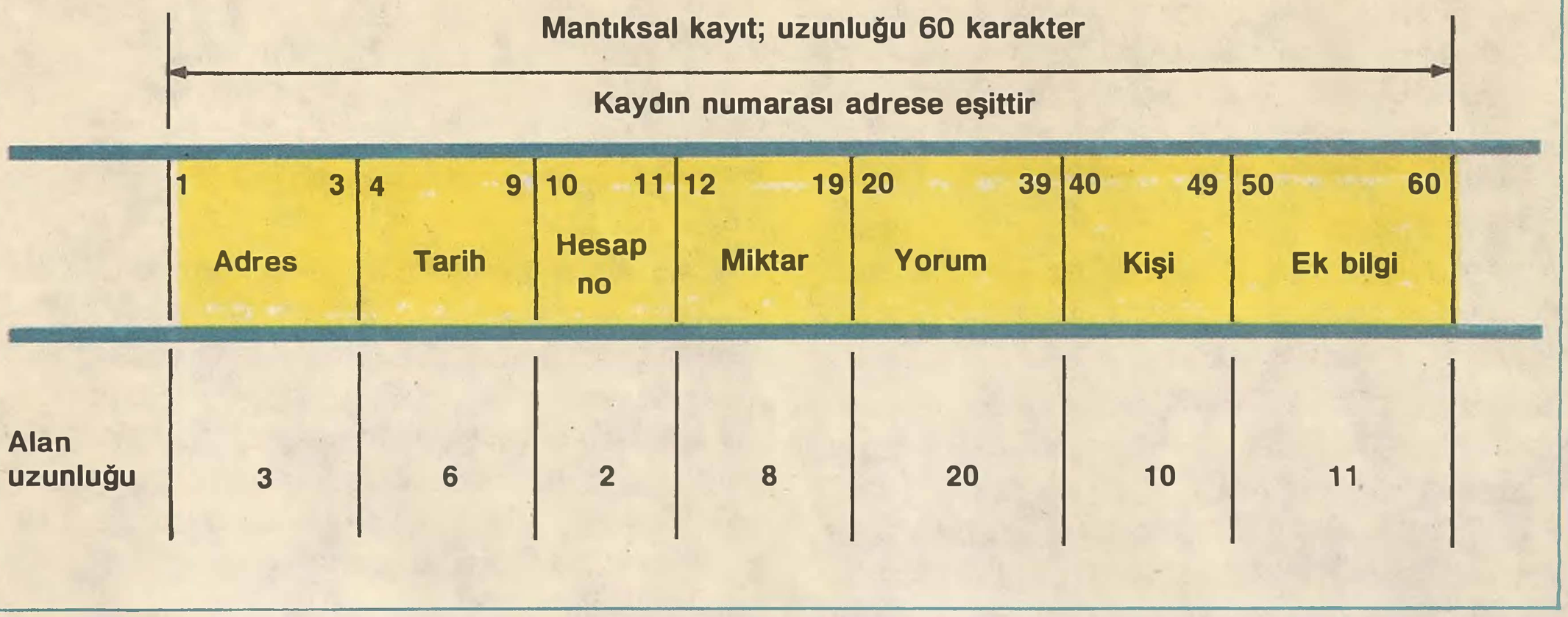
Tarih: Sarfiyatın tarihi şu şekilde kaydedilir: Gün (2 rakam), ay (2 rakam), yıl (2 rakam).

Bu kayıt yalnızca bize harcamaların yapıldığı tarihi hatırlatmakla kalmaz, bilgiyazım sırasında "seçici" işlevi de görebilir. Örneğin banka hesaplarımızı kontrol etmek için yalnızca belirli bir günde yapılan harcamaları öğrenmek istediğimizde bu veriden yararlanırız.

Hesap numarası: İşlemin gelir ya da gider olduğunu gösteren sayısal değerdir.

Yalnızca 11 ilâ 25 arasında yer alan kodlar kullandığımıza göre, bu alan için 2 karakter yeter. Veri girişi sırasında, hatalı veri girişlerini önlemek için bir sınama birimi, girilen kodun gerçekten bu değerler 11 ilâ 25 arasında yer alıp almadığını kontrol etmelidir.

AİLE BÜTÇESİ KÜTÜĞÜNDE VERİ YAPISI



Alan uzunluğu	3	6	2	8	20	10	11

Miktar: Gelir miktarını gösteren sayı.

Bu alan için 8 karakter öngörüldüğünden buraya girilebilecek en yüksek değer 99999999'dur. Ancak bu kısıt yalnızca veri girişi sırasında geçerlidir. Daha sonra genel toplamın hesaplanması gerektiğinde kullanılabilecek olan basamak sayısı çok daha yüksektir.

Yorum: 20 karakterlik bu alana sözkonusu işlem hakkında çeşitli açıklamalar yazılabilir. Örneğin gelir ve giderlerin türü burada daha ayrıntılı bir biçimde belirtilebilir. Sözgelimi prim koduyla (24) kodlanmış farklı birkaç gelir kalemi olabilir. Bu kalemlerden hangisinin hangi kaynaktan geldiği bu alana yazılabilir.

İlgili kişi: Gelirler için, parayı kimin kazandığı, giderler için ise harcamayı kimin yaptığı bu alanda yer alabilir. Bu alan sayesinde aile içindeki herhangi bir bireyin toplam gelirlere ve giderlere olan katkısı hesaplanabilir.

Ek bilgi alanı: Son olarak ek bilgiler alanı ise veri kaydımızı genişletme olanağı sağlar. Her veri arşivinde uzunluğu yaklaşık olarak belirlenmiş bir ek bilgi alanının bulunması gerekir. Aksi takdirde veri kaydı genişletilemez ve yeni bir alan gerektiğinde kütük kullanılamaz hale gelir.

Her alanın özellikleri aşağıda kısaca özetlenmiştir (karakter için K harfi kullanılmıştır).

Kayıt no (adres) - 3 K: Her yeni veri girişinde değeri 1 artar.

Tarih - 6 K: Gün ve ayın birbirlerine uygun olduğunun sınanması gerekir. (Örneğin 31/02 girildiğinde bir hata mesajının verilmesi gerekir.)

Hesap no - 2 K: 11 ilâ 18 veya 21 ilâ 25 arasında bir değer olmalıdır.

Miktar - 8 K: Bu alanda yalnızca sayılar yer alabilir.

Yorum -20 K: Herhangi bir sınama yok.

İlgili kişi -10 K: Herhangi bir sınama yok.

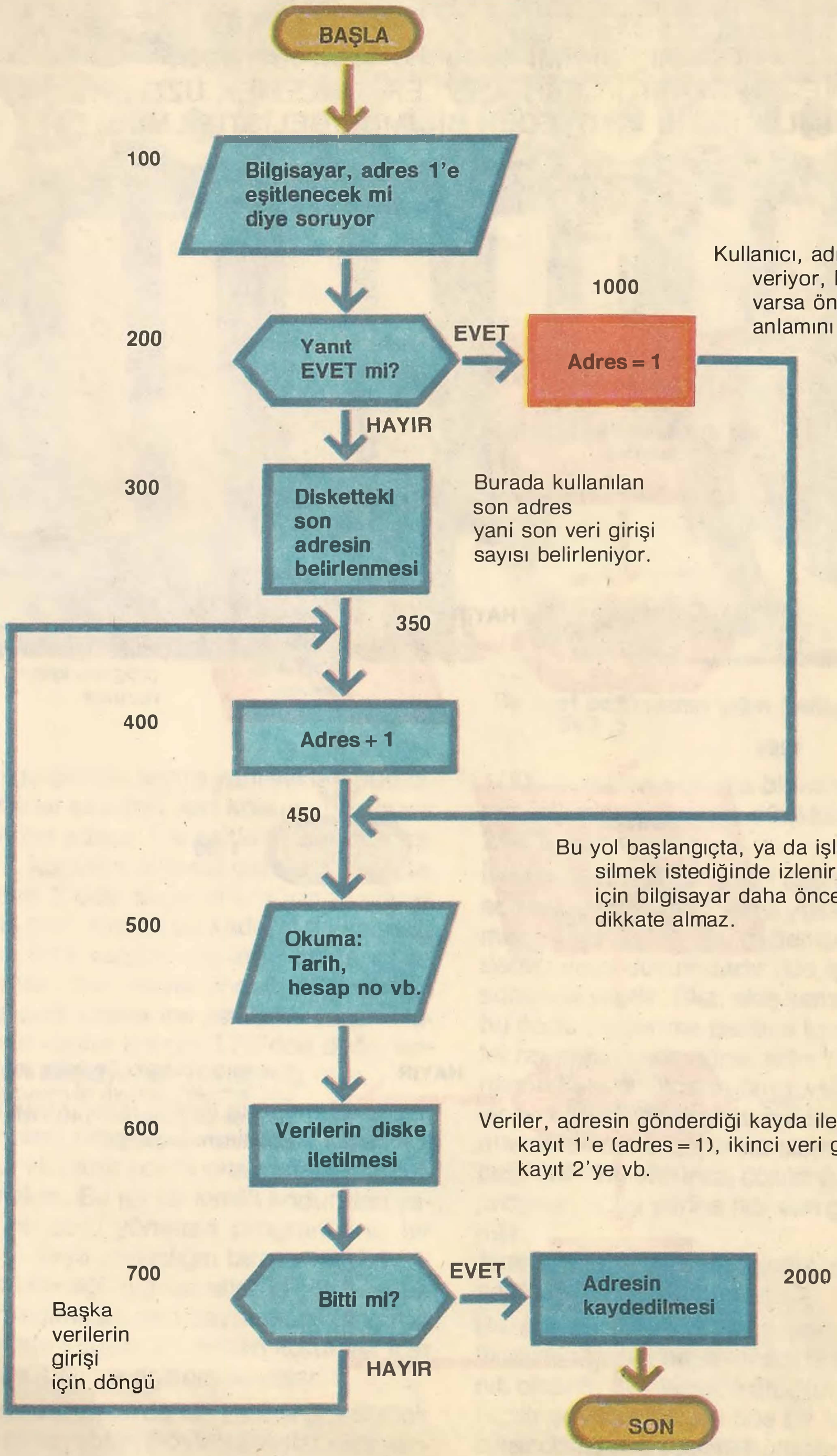
Bu listeden de görüldüğü gibi iki ayrı program biriminin hazırlanması gerekmektedir; biri her yeni veri girişinde adres değerinin 1 artmasını sağlamak için, diğeri ise tarih, hesap no. ve miktar alanlarındaki sınama işlemlerini yürütmek için.

Adresin Değerini Belirleyen Program Birimi

Yalnızca ilk veri girişinde adrese 1 değeri verilmelidir. Bundan sonraki her veri girişinde ise adresin otomatik olarak 1 artması gerekir. Veri girişleri sona erdiğinde, program adresin son değerini kaydetmeli ve daha sonra kullanıcı yeniden bir veri girişi yaptığında, program kaydedilmiş olan son adres değerini kullanmalıdır. Demek ki bu sırada adresin mutlaka bir manyetik diske (ya da şeride) yazılması gerekmektedir, çünkü bu iş için bilgisayarın kendi belleği kullanılırsa, bilgisayar kapatılır kapatılmaz bu değer silinecektir. (Anımsayalım: Elektronik bellekler ''uçucu''dur; akımla beslenmediklerinde içerikleri silinir.)

Bu programın akış şemasını yandaki sayfada görüyorsunuz. Ancak bu şemada resmedilen

VERİ GİRİŞİ AKIŞ ŞEMASI

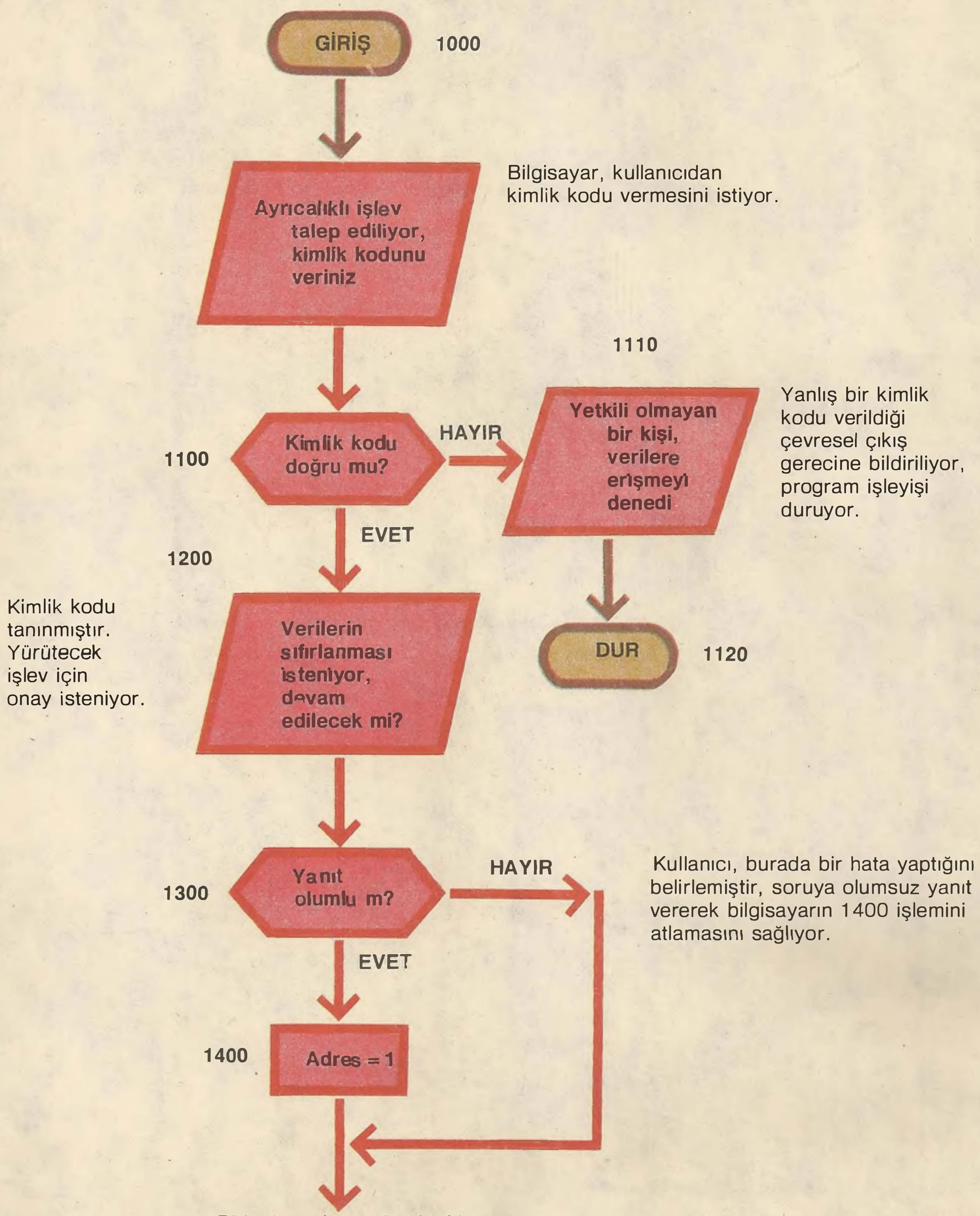


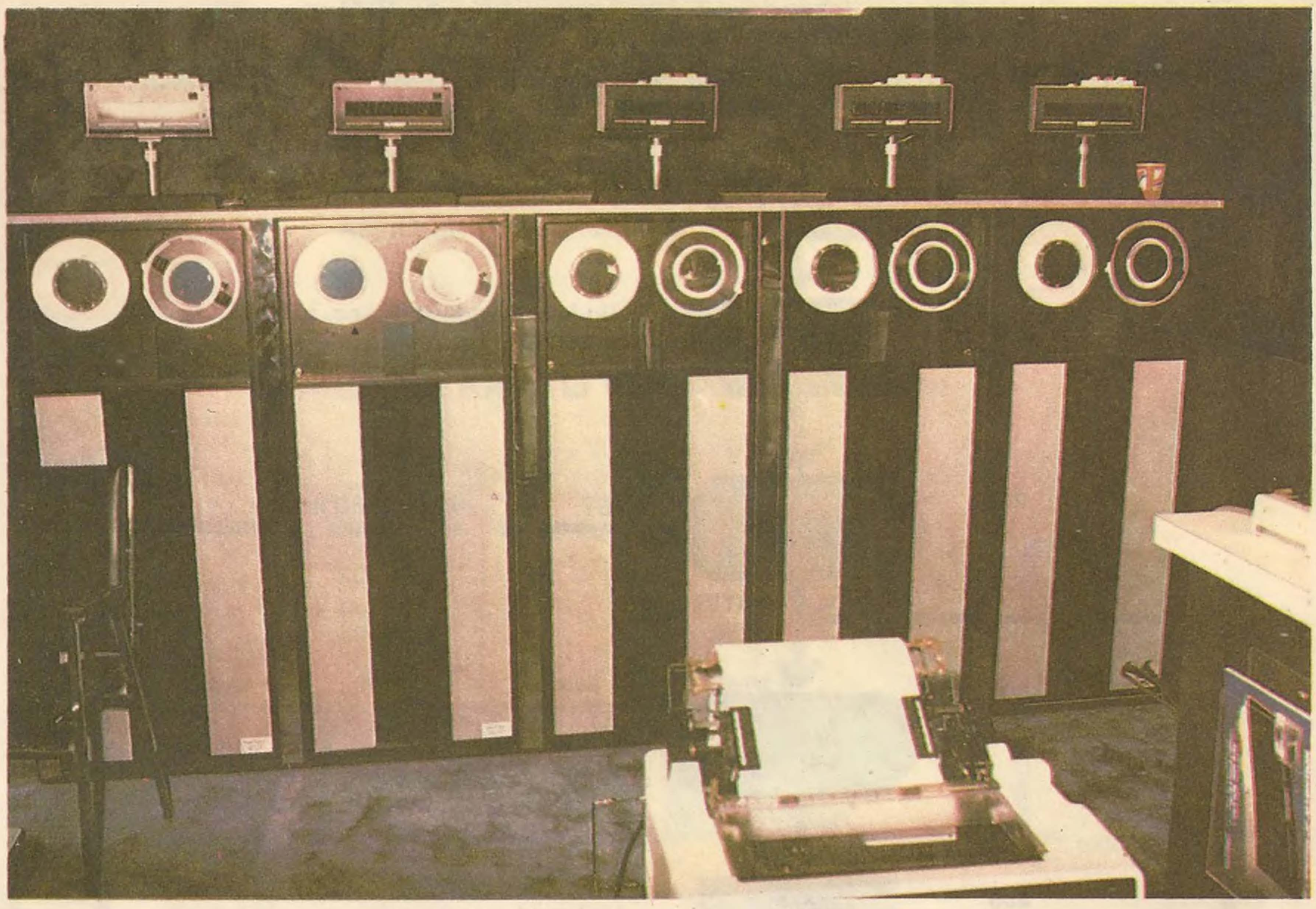
Bu sistemde güvenlik garantisi yoktur.

mantığın programın çalışmasını da etkileyecek iki büyük sakıncası vardır.

Adresi 1'e eşitleyen 1000 numaralı kutu, önceden girilmiş olan bütün verilerin silinmesine yol açar. Diyelim ki 176 adet veri girişi yaptık. Bilgisayar 176 sayısını adres değeri olarak saklayacaktır. Bir sonraki veri ise 176 + 1 konumundaki boş yere saklanacaktır.

PROGRAM BİRİMİ 1000'İN, VERİ KAYBINA NEDEN OLABİLECEK İŞLEMLERİ ÖNLEMEK ÜZERE KİMLİK KODU İSTEYECEK BİÇİMDE GELİŞTİRİLMESİ





Bir veri bankasının yığın belleğinden görünüş.

176. veri girişinden sonra yanlışlıkla 1000 birimi işletilirse sıradaki veri konum 1'e yazılır (1000 birimi adresi 1'e eşitler). Böylece aslında 177. konuma girmesi gereken yeni değer, konum 1'deki değerin üstüne yazılarak onu silmiş olur. Ancak bu kadarla da kalmaz; bilgisayar artık varolan verilerin miktarını da belirleyemez, dolayısıyla önceki 176 verinin tümünü silerek üzerlerine yenilerini yazar. Hatayı izleyen veriler konum 176'dan değil, konum 1'den başlayarak yüklenmiş olur.

Böyle hatalara karşı önlem almanın tek yolu, 1000 numaralı program birimine, bir kimlik kodu tanıma ve daha sonra onay isteme koşulları eklemektir. Bu tür bir kimlik kodundan yararlanan ve soru yönelten programlara, bir dikkatsizlik veya yanlışlığın bazen veri kaybına yol açabileceği uygulamalarda çok sık raslanır. Örneğimizde veri kaybı olasılığı varolmakla birlikte küçüktür; verileri korumak için adresi yine 176'ya eşitlemek yeter.

Başka bazı durumlarda ise yanlışı düzeltmek mümkün olmayabilir. Böyle yanlışlar kimi yerlerde, örneğin büyük işletmelerde ticari verileri saklayan veri arşivlerinde korkunç sonuçlara yol açabilir.

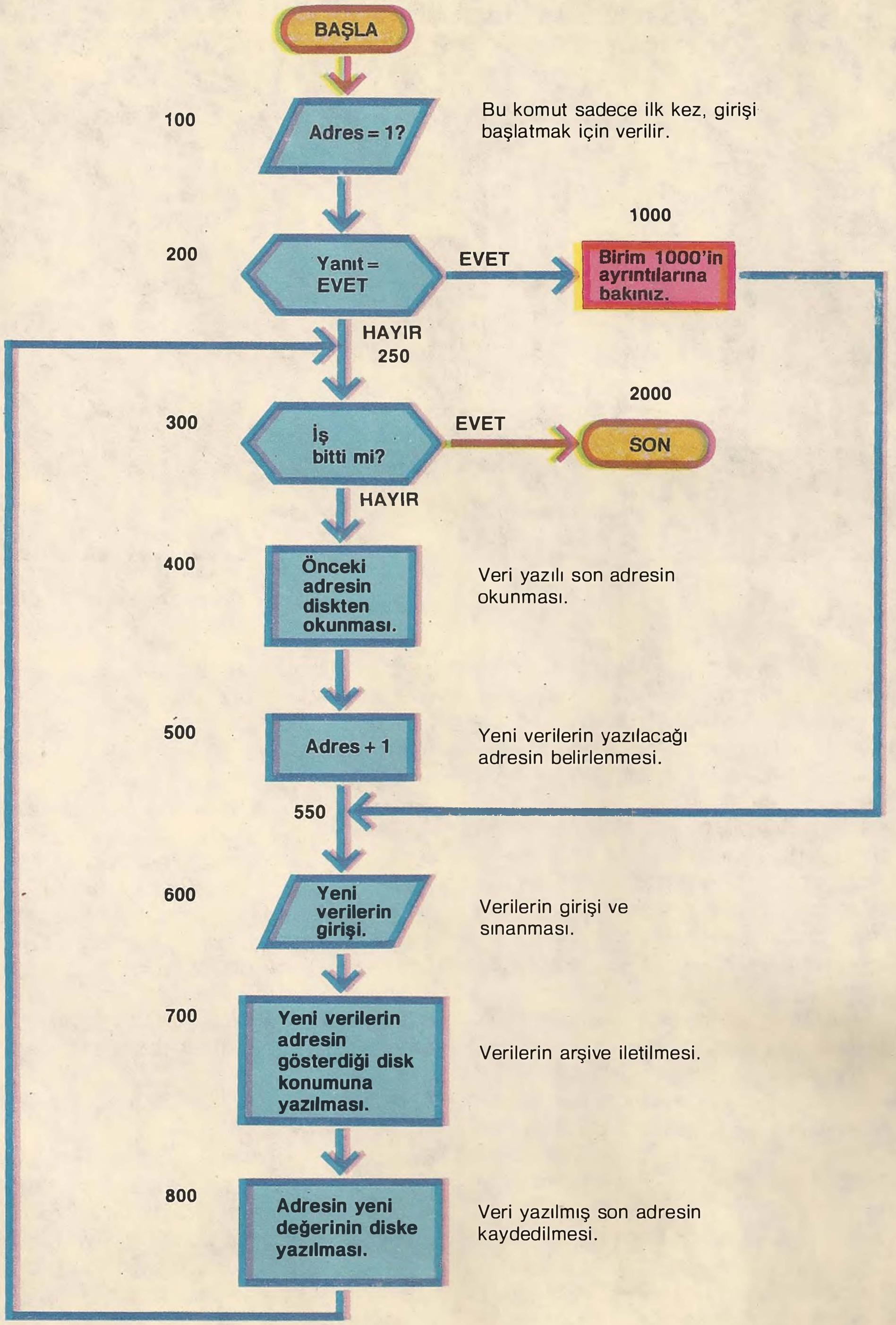
1000 numaralı koruma biriminin gerekli koruma işlemlerini içeren düzeltilmiş şekli sayfa 264'te yer alıyor.

İkinci sakıncayı bir örnek üzerinde göstererek açıklayacağız. 121. veriyi yüklemekte olduğumuzu varsayalım, bu değer henüz diske kaydedilmemiş durumdadır (Bu işlem ancak işin sonunda yapılır. Bkz. akış şeması s.263). Tam bu anda beslenme gerilimi kesilebilir. Sistem tekrar açıldığında adres artık 121 değildir; gerilimin kesilmesinden önce verilmiş tüm bilgiler veri arşivinde olduğu halde, kaydedilmiş oldukları bilinemez, çünkü sistem adresini kaybetmiştir. Bu sorunun çözümü basittir: Adres, program sonu yerine her veri girişinde kaydedilir.

Akış şemasının geliştirilmiş son durumunu sayfa 266'da görüyoruz.

Burada şu soru ortaya çıkıyor: Adresi manyetik diskin neresine vereceğiz? Sorunun iki yanıtı olabilir. İlk olarak, kütüğün, adres dışında hiçbir şey içermeyen boş bir kaydını kullanabiliriz. Bu fiziksel kayıt, kütüğün bir parçası olmakla birlikte, yalnız yardımcı bir işlev taşır. ''Yardımcı kayıt'' kütüğün başında yer alabilir; adresi oraya kaydedip, oradan alabiliriz. Bu

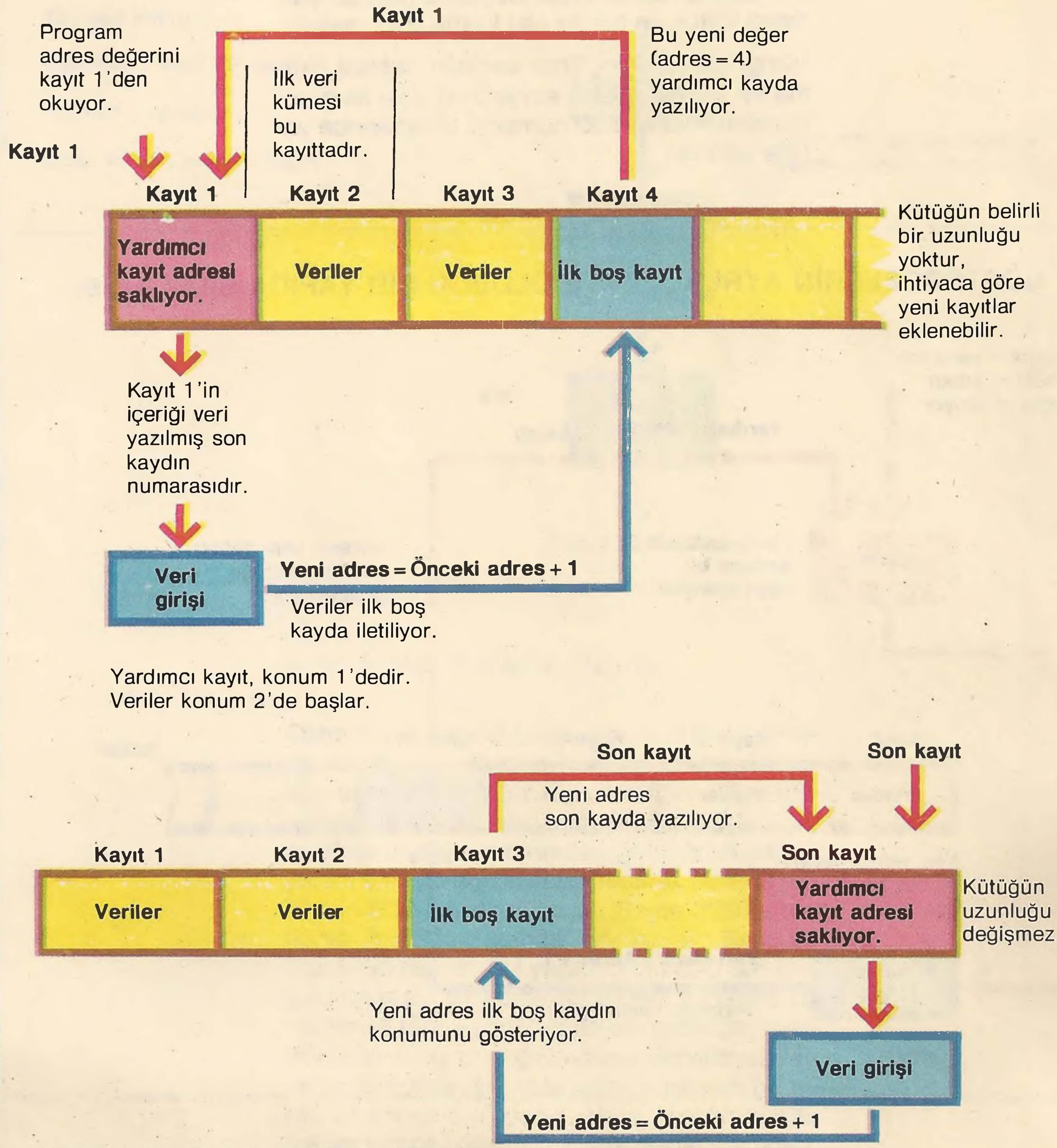
HER VERİ GİRİŞİNDE ADRES KAYDETMEK İÇİN AKIŞ ŞEMASI



durumda verilerin kaydedileceği ilk kayıt, 1 değil 2 numaralı konumdadır. Kütük istendiği kadar uzun olabilir; kullanım sırasında da uzatılabilir, çünkü uzunluğunu önceden tam olarak belirlemek gerekmez.

İkinci bir seçenek olarak, adresler yardımcı kayıt kütüğün sonunda da yer alabilir. Bu yön-

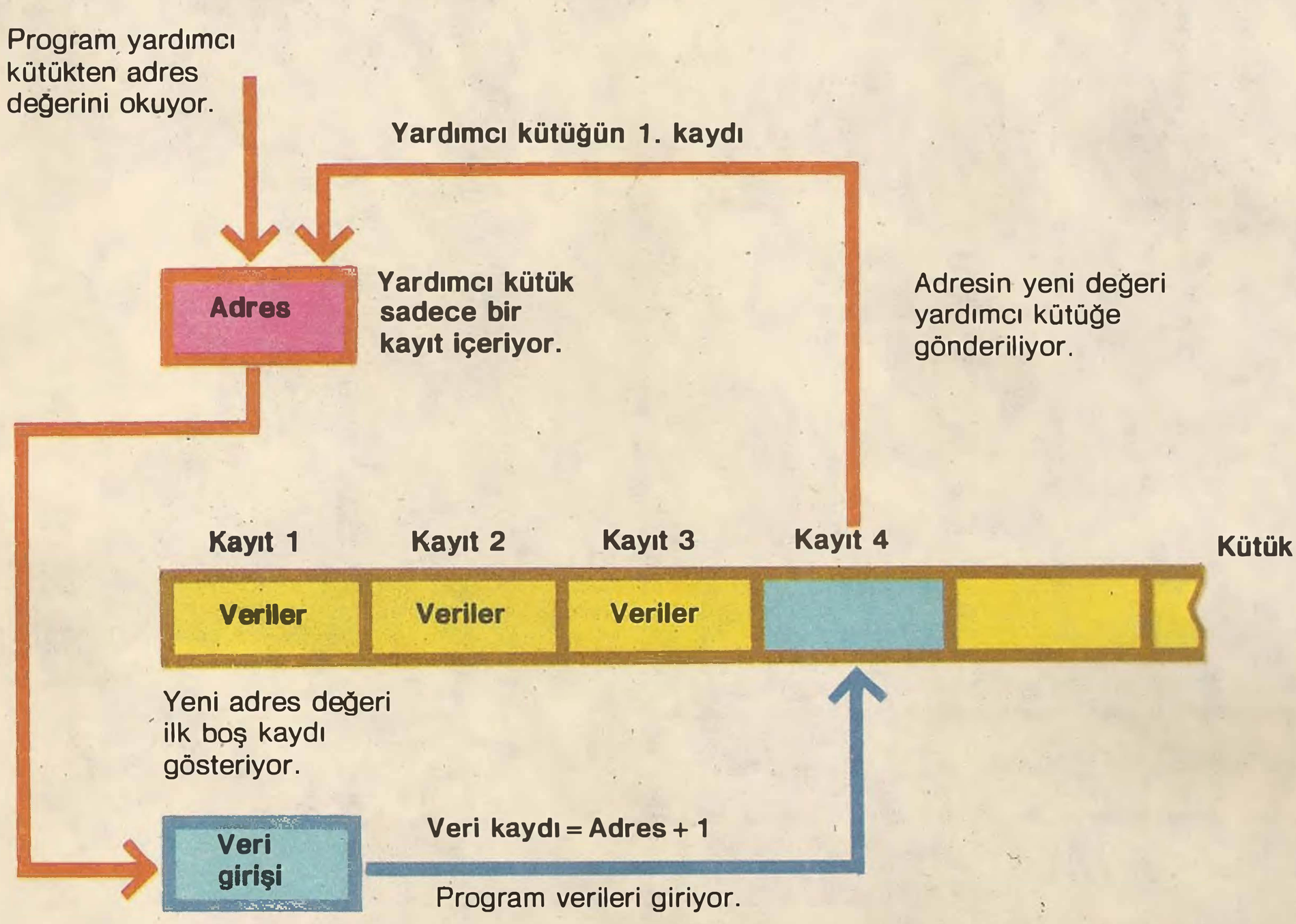
YARDIMCI KAYITLI KÜTÜKLERİN YAPISI



temle verilerin girişi 1 numara ile başlar (ilk kayıt artık yardımcı kayıt değildir), ancak bu yöntemin, daha öncekinde olmayan bir sakıncası da böylece doğar; adresin bulunduğu yardımcı kayıt, kütüğün son kaydı olduğundan, kütüğün ne uzunlukta olacağı önceden belirlenmelidir (bkz. s.267). Bu sakıncayı ortadan kaldırmanın bir yolu, yalnız adreslerin saklandığı tek bir kayıttan oluşan yeni bir kütük hazırlamaktır. Böylece hem veri yazmaya 1 numaralı kütükten başlayabiliriz, hem de kütüğün uzunluğunu önceden belirlemeye gerek kalmaz. Bu durumda tek olumsuzluk, iki kütük hazırlamak zorunda oluşumuzdur: Bir yardımcı kütük ve bir de asıl kütük (bkz. şekil).

Hangi yöntem seçilirse seçilsin, adresi okuma ve yazma işlemi sayfa 266'daki akış şemasının 400 ve 800 numaralı birimlerince yerine getirilir.

GÖSTERGELERİN AYRI KÜTÜKTE OLDUĞU BİR YAPIDA BİLGİ AKIŞI



Bu birimler, seçilen yönteme göre geliştirilebilir. Sayfa 269'daki şekil, DIR yardımcı kütüğünün kullanılması durumunda okuma biri-

mini gösteriyor. Yazma birimi 800 de aynı biçimde geliştirilebilir (doğal olarak bu durumda, 420 numaralı kutudaki ''okuma''nın yerini ''yazma''nın alması gerekecektir.''

DIR KÜTÜĞÜNDEN (YARDIMCI KÜTÜK) ADRES OKUMA

Birim: 400

Amaç: Yardımcı kütükten adres okuma.

Kütüğün adı: DIR

Uzunluk: 1 kayıt (60 bayt)

Adresin konumu: 1-3

Çıktı: Adresin sayısal değeri.

Birim 800, benzer biçimde yazma işlemini yürütür.

Veri Girişi Sınama Birimi

Daha önce değindiğimiz gibi, veri girişi evresinde ''tarih'', ''hesap no'' ve ''miktar'' alanlarının biçimsel doğruluğunun denetlenmesi gerekir. Bu işlev, veri girişi birimi içinde, yalnız doğru verilerin birimden geçmesini sağlayacak şekilde gerçekleşmelidir (s.266'da şekildeki 600 numaralı kutu). Sayfa 270'te bununla ilgili akış şeması görülüyor. Şemada ''tarih''in denetimini yapan birim 620'dir. Uygulamada sık sık kullanıldığından, bu işlevi daha sonra tüm ayrıntıları ile ele alacağız; şimdilik gün ve ay'ın doğruluğunu denetleyebilen ve gerektiğinde bir hata iletisi verebilen bir birim üzerinde duracağız. Hata numarası sıfır ise veri (tarih) doğrudur; sıfır değilse yüklenmiş tarih değeri yanlış demektir.

Hata iletisinin aldığı sayısal değer, tarihin ve-

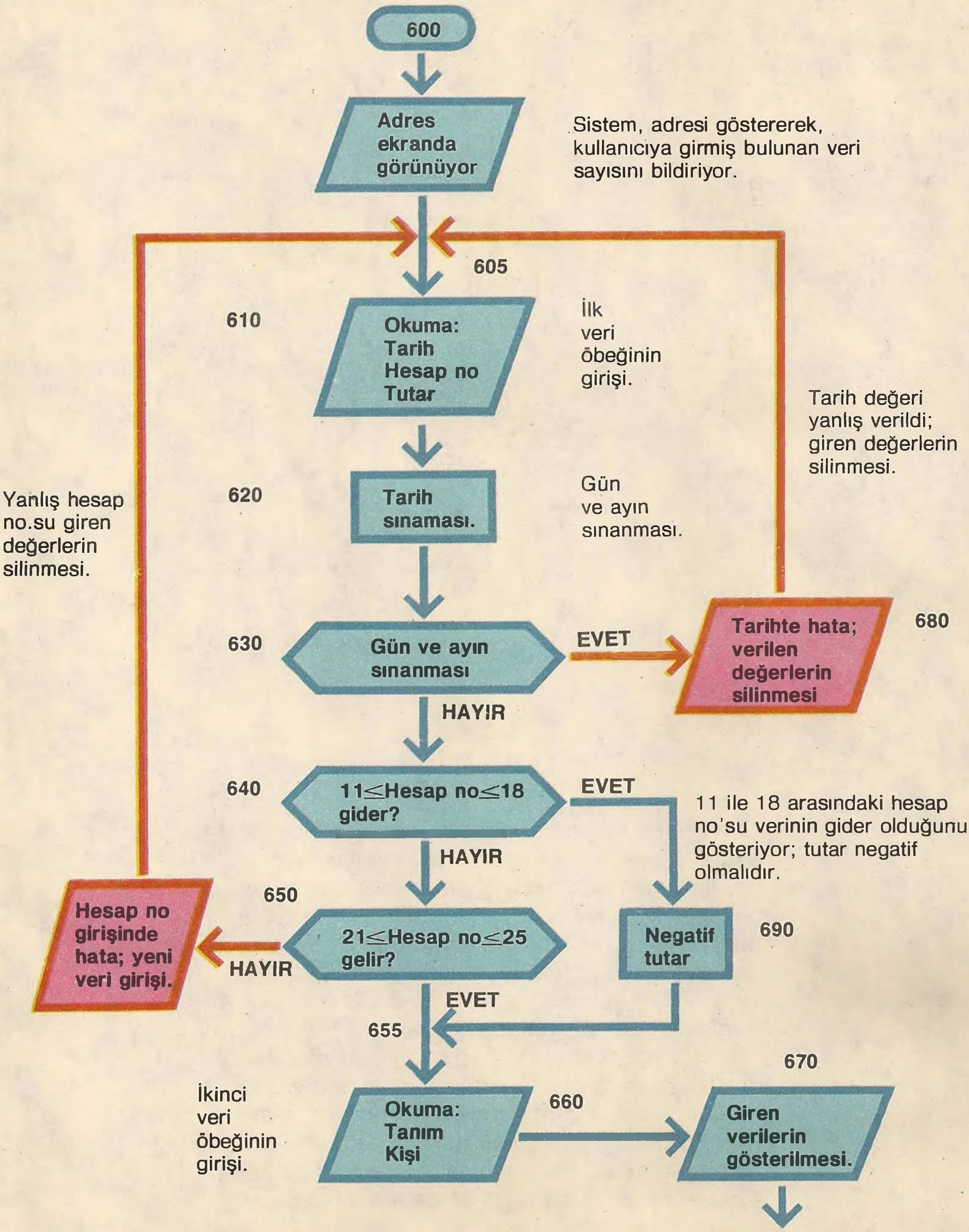
VERİ SINAMASI DA YAPAN GİRİŞ BİRİMİNİN AYRINTILI AKIŞ ŞEMASI

Birim: 600

Amaç: Klavyeden giren verileri okuma ve biçimsel sınamalarını yapma.

Girdi: Adres.

Çıktı: Diske iletilecek alan verileri.



rilişi sırasında yapılan hatanın cinsinin belirlenmesini sağlar. Bu birim kullanılarak programlara bir ''geçerlik süresi'' de verilebilir. Örnek olarak, 1995'ten sonraki yılların geçerli olmayacağı koşulunu verirsek, program 1996'yı girdi olarak kabul etmez. Kullanıcı, elinde yalnızca programın derlenmiş biçimi varsa bu sınırlayıcı değeri değiştiremez; düzeltme ancak ham program üzerinde yapılabilir. Elbette düzeltmeler bittikten sonra programın yeniden derlenmesi gerekir. Böyle durumlarda kullanıcının sisteme rasgele bir sayı verme olanağı da vardır. Örneğin 1966 yerine 1995 gibi geleceğe ait bir tarih yazdırılabilir. Ancak bu program, yalnız özel amaçlarla kullanılacaksa mümkündür. Program, örneğin veri belgeleri (bilançolar, hesaplar vb.) gibi genel bir kullanım alanı bulmuşsa, yanlış tarih değerleri verilemez.

Geçerlik süresi tekniğiyle, programları kiraya vermek mümkün olur; ''tarih'' birimi, programcının belirlediği bir güne bağlı olarak, programın kullanılmakta bulunduğu tarihi kabul etmiyorsa, programın işleyişi durur. Böylece programı kiralayanın kira ücretini ödemesi sağlama bağlanır. 600 birimi, veri girişini sağlar.

Özetleyelim:

- **S.262'de kayıt yapısını ve alan uzunluklarını görüyoruz.**
- **S.266'da, veri sınama birimleriyle birlikte tüm giriş programı için akış şeması yer alıyor.**
- **Yürütme hazırlığı işlevi gören birim 1000'in akış şeması s.264'tedir.**
- **S.268 ve 269'daki şekiller, bir yardımcı kütükten veri adreslerinin alınmasını gösteriyor.**
- **Veri girişi ve sınanması işlevlerini s.270'te görüyoruz. S.264, 266, 269 ve 270'teki akış şemaları programın bütününü oluşturuyor.**

Bu işlemleri geliştirerek artık yöntemimizi tamamlayabiliriz. Yeni programın işlevi, yalnız, tüm kütüğü okumak, gelir ve giderlerin ayrı ayrı genel toplamını hesaplamaktır. Böylelikle parasal işlemlerin tümü bir bakışta görülebilir ve gelirlerle giderler arasındaki toplam farkını bildiren bir rapor hazırlanabilir. Böyle bir bütçede, yalnız, önceden belirlenmiş iki tarih arasındaki miktarları gözönünde bulundurmak tercih edilir. Böylelikle hem belirli bir zaman aralığına ilişkin bilgileri alabilir, hem de gelecekteki parasal işlemlerle ilgili tahminlerde bulunabiliriz. Programın akış şeması sayfa 272'dedir.

Hesap numarası, tanım, kişi gibi verilere özel bir seçme işlemi uygulayan ''ek'' programların, özelliklerine değinmekle yetineceğiz. Sayfa 272'deki birimleri ayrıntıları ile birlikte ilerideki bir bölümde, gerekli BASIC komutlarını açıkladıktan sonra bir programın nasıl yazılacağını da göstererek yeniden ele alacağız.

Kütüklerde Veri Arama

Verilerin girişi evresinin sonunda kütük hazırlanmış olur. Bundan sonra ortaya çıkan değişikliklere uygun olarak kütüğe yeni veriler işlenir.

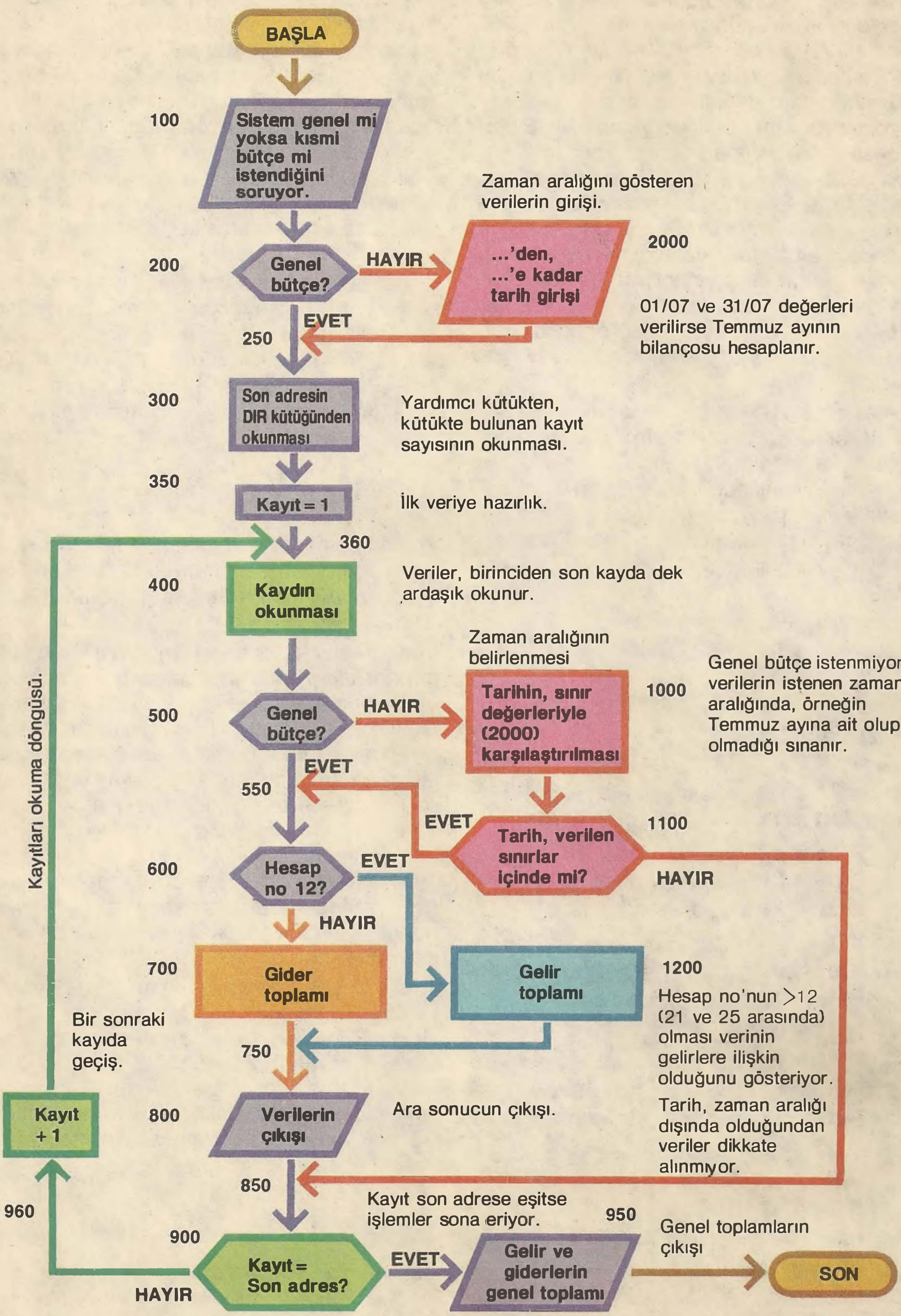
Kütükteki bir kaydı en yeni duruma uygun hale getirmek için önce kayıt okunur, sonra da gerekli ekleme ve değişiklikler yapılır. Veriler yenilenirken en önemli işlem, aramadır. Kullanıcı, kayıtları okumak için anahtarlardan yararlanır ve hangi verilerde değişiklik yapılacağına karar vererek ilgili kayıtları ayıklar. Kütük sürekli kullanılıyorsa (örneğin tablo çıkışı için), çoğu zaman verileri belirli bir mantığa göre ayıklamak gerekir. Bu amaçla, örneğin, bir adres arşivinden belirli bir şehirdeki adreslerin hepsini veya bir kütükten belirli bir bölümde çalışanlara ilişkin tüm kayıtlar seçilebilir.

Tablo çıkışı isteniyorsa, veri arşivinde bulunan verilerin bir bölümünün ayıklanması gerekir. Dolayısıyla, önceden belirlenmiş özelliklere uyan verilerin bulunduğu kayıtları okumak üzere arama programlarının kullanılması kaçınılmaz hale gelir. Başlıca iki arama yöntemi, **''Seri Arama''** ile **''Yarılamalı Arama''** (ya da **çatallı arama**)'dır.

Seri Arama

Seri aramada verilere erişmek için veri arşivindeki kayıtlar birbiri ardından okunarak anahtar değeri ile karşılaştırılır. Eşitlik sağlan-

AİLE BÜTÇESİ PROGRAMININ AKIŞ ŞEMASI



dığında aranan veri bulunmuş demektir ve bilgiyazara verilebilir; yoksa kayıtların hepsinin okunması bitinceye dek arama sürer.

Veri bulununcaya dek yapılması gereken okuma işlemlerinin ortalama sayısı şu denklemle bulunur:

$$\text{Okuma işlemlerinin ortalama sayısı} = \frac{\text{Kütük uzunluğu} + 1}{2}$$

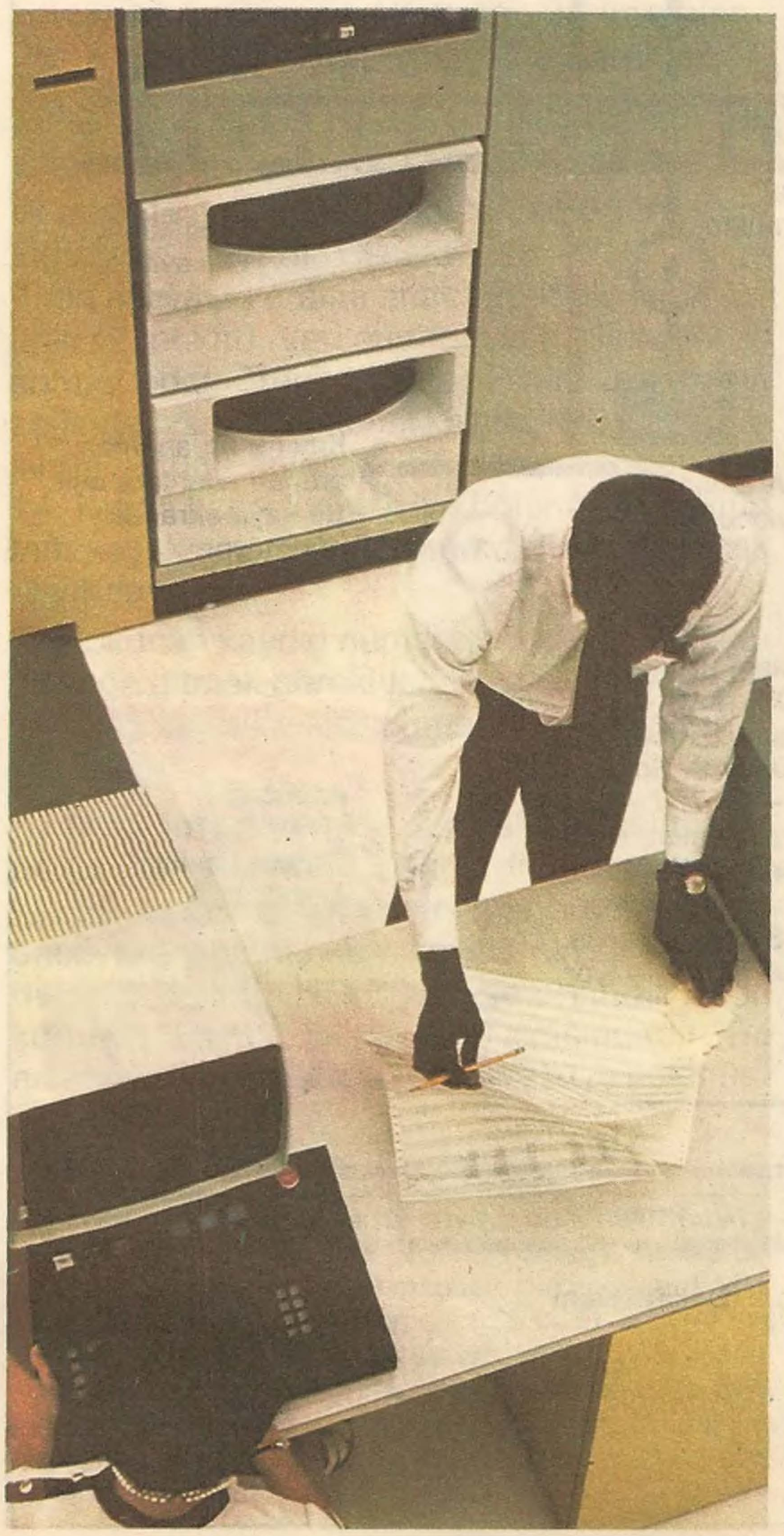
Buna göre 1000 kayıtlık bir kütükte herhangi bir kaydı bulmak için ortalama 500 kaydın okuma işleminden geçmesi gerekir.

Aranan bir veriden kütükte yalnız bir adet bulunduğu varsayılmaktadır. Aynı veri birçok kayıtta yer alabilir; bu durumda aramanın, verilerin sonuna ulaşılasıya yürütülmesi gerekir.

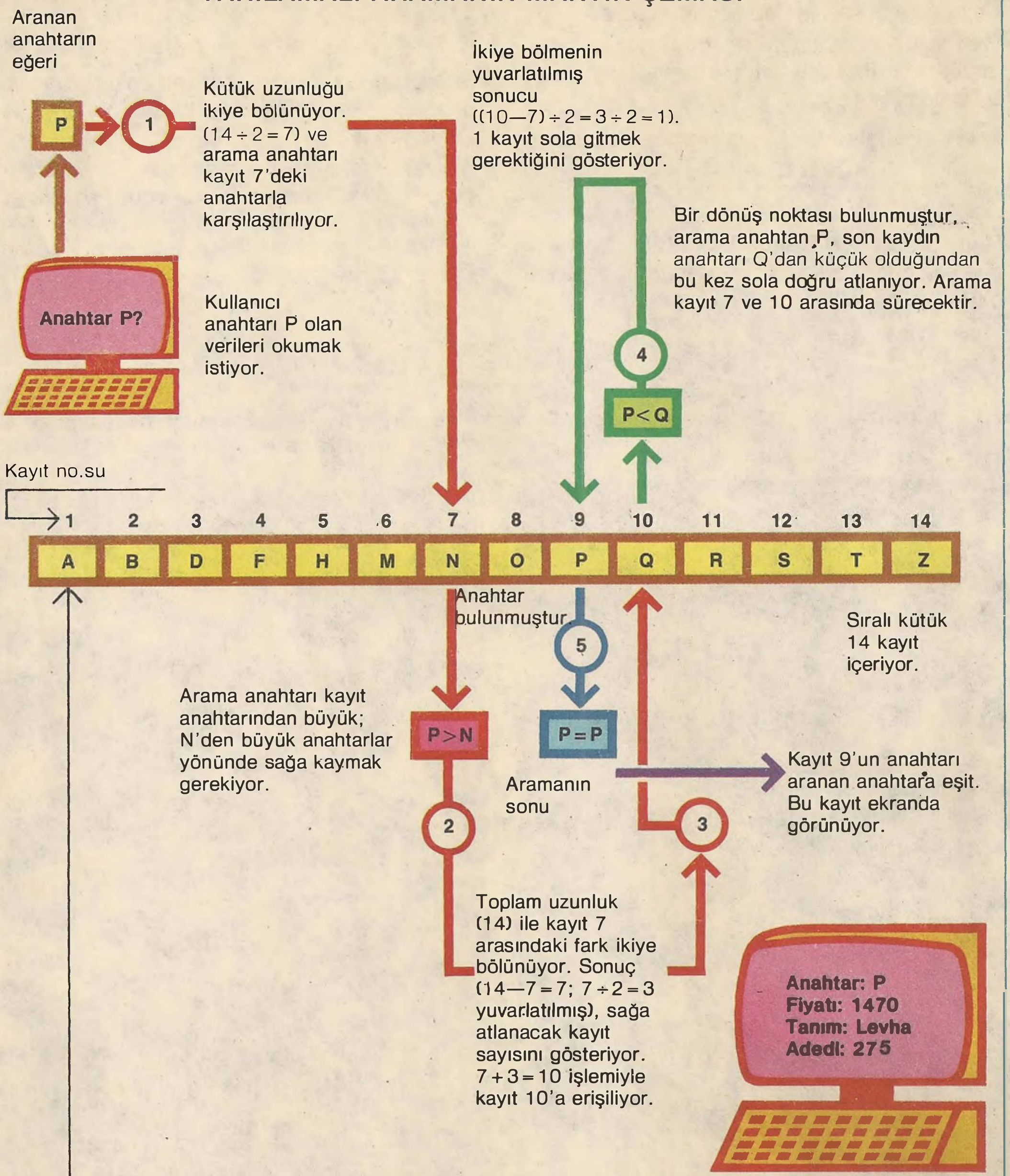
Bu süreci hızlandırmanın tek yolu, bir arama programını işletmeye koymadan önce verileri sıralamaktır.

Seri arama için uzun arama süreleri gerektiğinden, bu yöntem yalnızca küçük boyutlu veri arşivlerinde kullanılır. Küçük arşivlerde başka arama yöntemlerinin sağlayacağı zaman kazancı önemsizdir.

Bir veri bankasında çalışmanın iki evresi. Planlama ve verilerin yapılandırılması, bellekten en iyi biçimde yararlanmanın ön koşullarıdır.



YARILAMALI ARAMANIN MANTIK ŞEMASI



	Anahtar	Fiyatı	Tanımı	Adedi
İçeriği	A	12.560	Dolmakalem	106
Karakter sayısı	1	6	20	3

Kayıt uzunluğu = 1 + 6 + 20 + 3 = 30 bayt

Yarılamalı Arama

Yarılamalı (ya da çatallı) arama yöntemi sadece sıralanmış kütüklere uygulanabilir. Bu yöntemle kayıtlar atlamalı olarak okunur. Toplam veri kaydı sayısı her atlama sırasında yarıya bölünür, gelinen noktada kayıt anahtarı, arama anahtarı ile karşılaştırılır, bunun sonucuna bağlı olarak bir sonraki atlamanın yönü belirlenir. Arama anahtarı veri anahtarından büyükse, daha büyük anahtarlı verilere doğru atlama yapılır, değilse, daha küçük anahtarlı veriler atlanır. Bu işlemi bir örnekle açıklayalım (bkz. şekil, s.274).

Bir kırtasiye deposu için hazırlanan kütüğün 30'ar bayt uzunluğundaki kayıtları aşağıdaki alanlara bölünmüştür:

- Anahtar alanı: Alfabetik, 1 karakter.
- Malın fiyatı: Sayısal,* 6 karakter.
- Malın tanımı: Alfabetik, 20 karakter.
- Stok düzeyi: Sayısal,* 3 karakter.

Kullanıcı, anahtarı P harfi olan kayda erişmek istiyor. Bu noktada yarılamalı arama başlar.

İlk adımda (şekildeki 1. nokta), kütüğün uzunluğu ikiye bölünür: Örnekte, 14 ÷ 2 = 7. Kayıt 7'nin anahtarı, arama anahtarı (P) ile karşılaştırılır (2. nokta). Veri anahtarı N harfidir, P> N sonucundan ötürü anahtar P'nin, bu kaydın sağında, yani 7 numaralı kayıtla, 14. kayıt arasında olduğu anlaşılır. Bu aralık (7. kayıttan 14. kayda kadar) yine ikiye bölünür, bölümün tam sayı kısmı olan 3, sağa doğru atlanacak kayıt sayısıdır.

Son okunan kaydın numarası 7 idi, şimdi 3 kayıt sağa gitmek gerekiyor, dolayısıyla 7 + 3 işlemi sonucunda 10 numaralı kayıt okunur (3. nokta).

Kayıt 10'un anahtarı P'den daha büyüktür (4. nokta) demek aranan anahtarın bu kaydın solunda olması gerektiği anlamına gelir. Daha önce, aranan anahtarın kayıt 7'nin sağında olduğunu belirlemiştik, öyleyse anahtar P'nin konumu, kayıt 7 ile kayıt 10 arasındadır. Bu aralıktaki kayıt sayısı da (10—7) ikiye bölününce sonuçtaki tam sayı olan 1, bulunulan noktadan bir kayıt sola gidilmesi gerektiğini gösterir. Son karşılaştırmayı 10 numaralı kayıtla yaptığımızdan sola doğru bir kayıtlık atlamayla kayıt 9'a varılır. Bu kaydın anahtarı P'dir (5. nokta); arama sona ermiştir ve kayıt 9'un verileri ekranda gösterilir.

Bu arama yöntemiyle 14 kayıtlık bir kütükte 3 okuma işlemi yaptık (Kayıt 7, 10 ve 9); seri aramada 9 okuma işlemi gerekirdi (anahtar P'yi buluncaya dek ilk 9 kayıt). Yarılamalı aramanın, seri aramaya oranla üstünlüğü açıkça görülüyor.

Burada birkaç noktaya değinmek gerekiyor. Yarılamalı arama ancak sıralanmış kütüklerde mümkündür, kütükteki veriler bir düzene göre kaydedilmemişse arama işleminden önce sıralı olarak yeniden yazılmaları gerekir. Bu durumda, yalnız, asıl arama için gereken vakit değil, kütüğün yeniden düzenlenmesi için harcanan vakit de hesaba katılmalıdır. Bundan ötürü az sayıda kayıt içeren kütükler için yarılamalı aramanın fazla yararı yoktur, böyle bir kütük için seri arama daha uygundur.

Verilerin Sıralanması

Kütük hazırlamanın başlıca sorunlarından biri, verileri sıralamaktır. Belirli teknikler (örneğin yarılamalı arama) ve düzenli tablolar için sıralama işleminin mutlaka yapılması gerekir.

Bir veri arşivinden (küçük boyutta da olsa) verileri düzenli olarak gösteren bir tablo elde edilemiyorsa, istenen verilerin bulunması çok güçleşir. Bu durum bir telefon rehberinde abonelerin alfabetik sırayla değil de, telefon başvurusu sırasına göre yer almasına benzer. Böyle bir rehberde, telefonunu daha önce alan Bay Seren'in adı, Bay Baran'dan daha önceki bir konumda olurdu. Herhangi bir arama yöntemini sınırlı sürede uygulayamayacağımızdan, belirli bir telefon numarasını bulmak da olanaksızlaşırdı.

Aynı şey veri arşivi için de geçerlidir; veriler giriş sırasına göre yerleştirilmişse arşivden yararlanmak mümkün olmaz. Sonuç olarak, kütükteki verileri sıralayan özel programlara her zaman ihtiyaç vardır. Bu programlara genel olarak **"Sıralama Programları"** adı verilir.

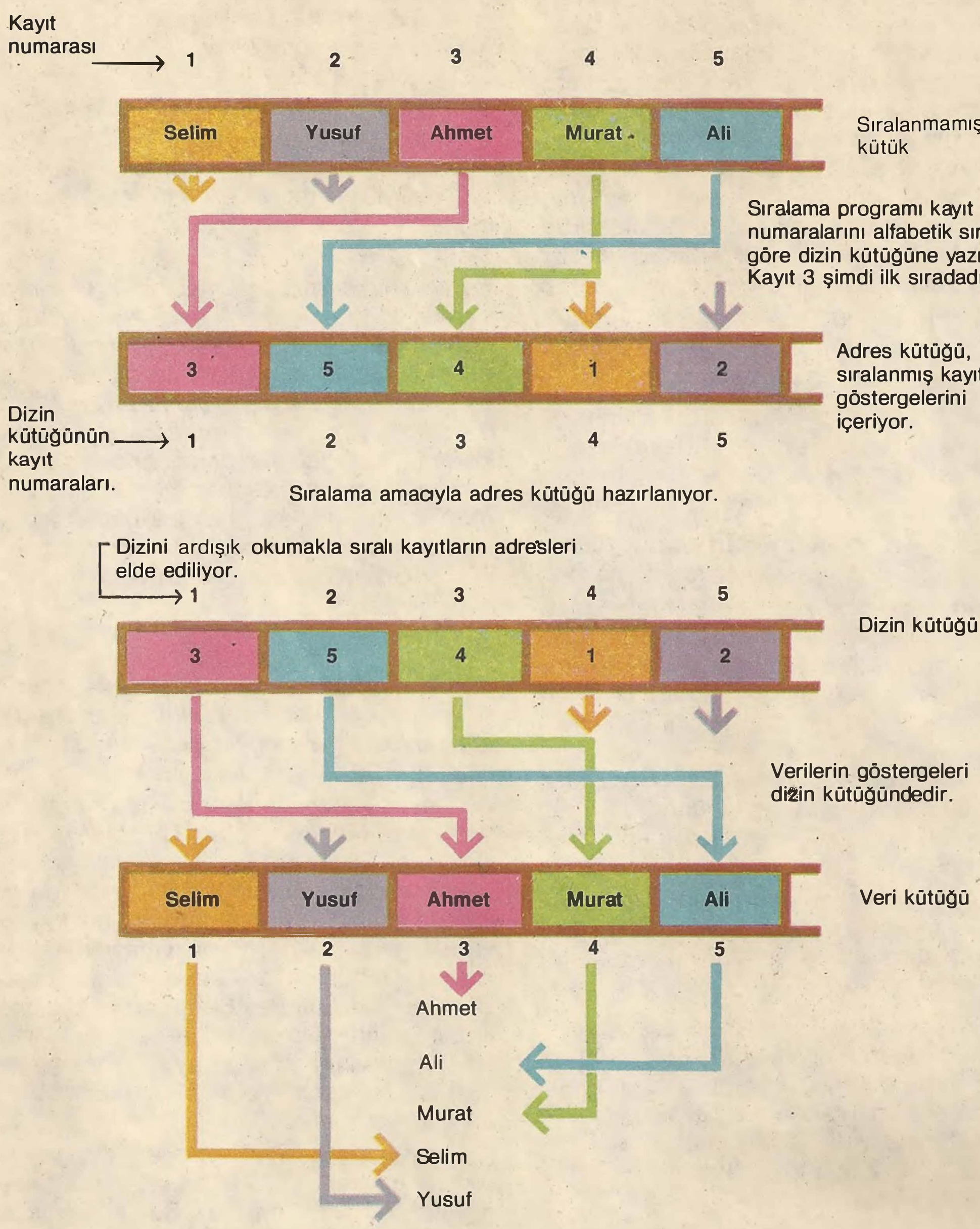
Sıralama programları, önemli olduklarından, işletim sistemlerinin hemen hemen hepsinde vardır; hızlı bir uygulama sağlamak için Assembler (çevirici) dilinde yazılırlar.

Bu programlar, çok çeşitli amaçlarla kullanılabilen **"Hizmet Programları"** sınıfına girer.

---

* Birçok kişisel bilgisayarda sayısal değerler, karakter kodları biçimleriyle diskete yazılırlar; örneğin sayısal değer 579, sırasıyla 5, 7 ve 9 rakamlarının karakter kodlarının onaltı tabanlı karşılıkları olan 35 (=5), 37 (=7) ve 39 (=9)'un yazıldığı 3 baytta saklanırlar (bkz. ASCII kodları). Bazı başka bilgisayarlarda sayıların doğrudan doğruya iki tabanlı karşılıkları diskete kaydedilir: Bu durumda, kullanılan bayt sayısı daha azdır. Bu konuya daha sonra yine döneceğiz.

VERİLERİN DİZİN KÜTÜĞÜ İLE SIRALANMASI

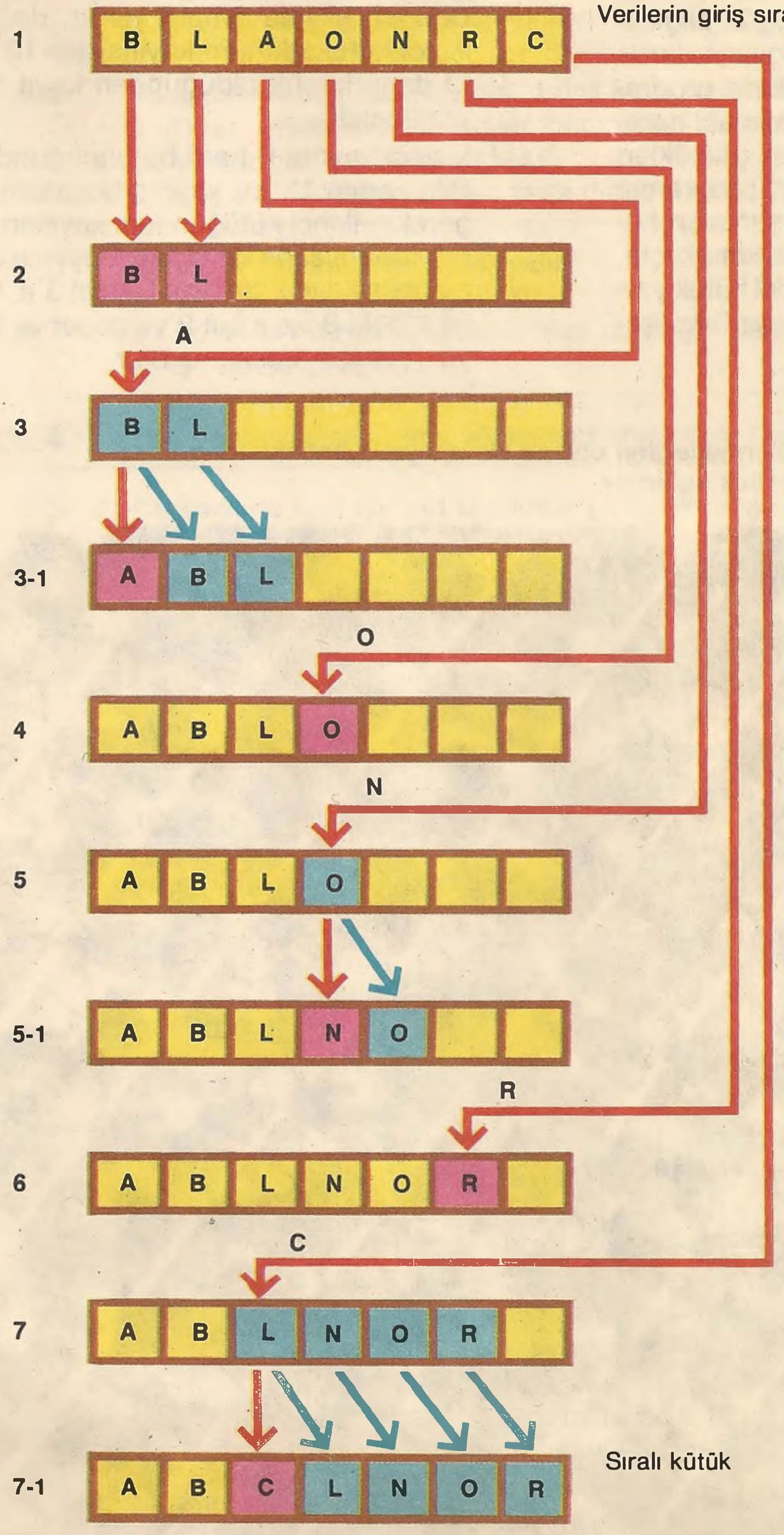


Verileri ekran ya da bilgiyazar çıkışında sıralı göstermek için dizin kütüğü ardışık okunur, bu kütüğün kayıtlarındaki göstergelerle, asıl kütükteki verilere doğru sırayla erişilir.

Kullanıcı, sıralama programını yürütmek için, kütüğün adını, kayıt yapısını (kayıtların hangi uzunlukta kaç alana bölündüğünü) ve hangi anahtara bağlı olarak sıralanacağını bildirir. Sıralama programları, başlıca iki mantıktan bi-

rine göre işler. Bunlardan birinde kütük, manyetik diskin başka bir bölgesine sıralanmış biçimiyle yazılır, böylece yeni bir adla, sıralan-

VERİLERİN BİR PARALEL KÜTÜKTE SIRALANMASI



Verilerin giriş sırasıyla yer aldığı kütük.

Verilerin sıraya uygun olarak yazıldığı paralel kütük.

Sıralı kütük

mış veriler içeren yeni bir kütük elde edilir. Ötekinde, kütükteki kayıtların adreslerini sıralanmış olarak saklayan bir adres kütüğü hazırlanır; verileri sıralı okumak için önce adres kütüğü, sonra adreslerin gösterdiği kayıtlar okunur (bkz. şekil, s.276).

Kütük kopya edilmeyip yalnız adresler yazıldığından ikinci yöntem daha hızlıdır, yine aynı nedenle manyetik diskte daha az yer kaplar. Sıralama programları tüm küçük bilgisayarlarda vardır; ancak kişisel bilgisayarlarda bulunmayabilir. Bu durumda sıralama programını, programcının kendisinin yazması gerekir. Sıralama programı yazmanın güçlükleri yöntemli bir çalışmayla aşılabilir; programın açık bir dökümü ve ayrıntılı akış şemaları hazırlanmalıdır. Böyle bir yöntemi tanıtmak için, örnek olarak, bir kütüğün bir paralel kütük yardımıyla düzenlenmesini ele alacağız (bkz. şekil, s.277).

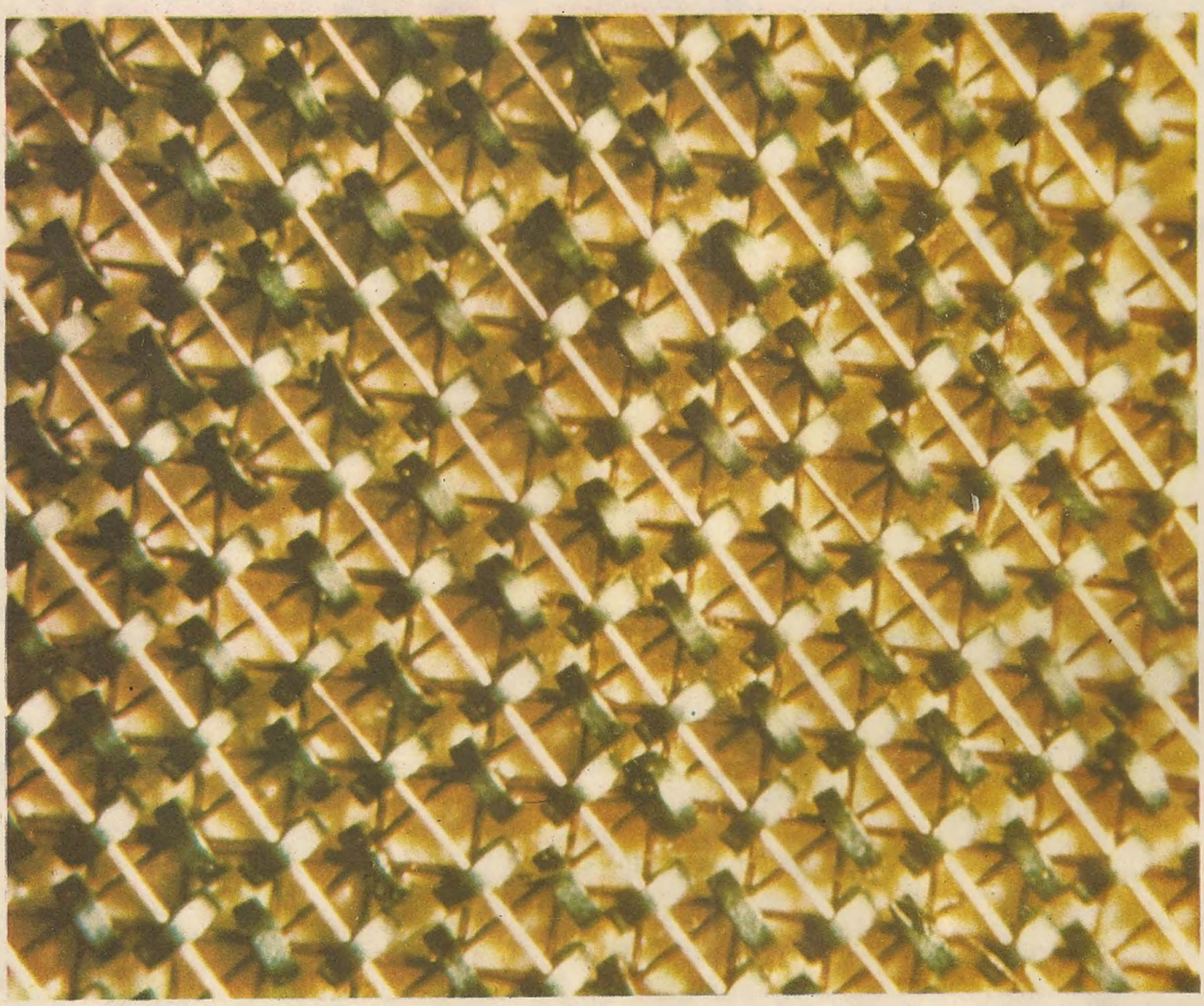
1 / Birinci kayıt (B) okunur ve ikinci kütüğün ilk kaydına yazılır. (Bu ''paralel'' kütükte sıralanmış veriler yer alacaktır.)

2 / İkinci kaydın değeri (L), birinci kaydın değerinden daha büyük olduğundan (L > B: L alfabede B'den daha sonradır), paralel kütükte ikinci kayda yazılır.

3 / Üçüncü kayıtta A harfi vardır, değeri ikinci kütükteki tüm kayıtlardan (B ve L) daha küçük olduğundan kayıt 1'e yazılmalıdır.

3-1 / Birinci kayıtta B harfi bulunduğundan (bkz. adım 1), bu kaydın boşaltılması gerekir. İkinci kütüğün tüm kayıtları birer konum sağa kaydırılır. Böylece daha önce kayıt 2'de olan L kayıt 3'e, kayıt 1'deki B ise kayıt 2'ye geçer ve kayıt 1, A için boşalmış olur.

Modern veri bankalarında, ferit çekirdeklerden oluşan ilk manyetik belleklerden çok daha gelişkin dev kapasiteli bellek kullanılır.



TEST 7

1 / Aşağıdaki ifadelerden hangisi sıralı kütüğe uyar?
a - Kayıtlara herhangi bir sırayla erişilebilir.
b - Veriler anahtar alanındaki değerin küçükten büyüğe doğru arttığı bir sıra içinde saklanır.
c - Kayıtla ardışık okunur ve yazılır.

2 / Aşağıdaki ifadelerden hangisi dizinli kütüğün bir özelliğidir?
a - Dizin, kayıtlarla ilgili bilgiler içeren özel bir alandır.
b - Veriler çok çabuk sıralanabilir.
c - Böyle bir kütüğün birden fazla dizini olabilir.

3 / Sıralama programının işlevi nedir?
a - Veriler doğru sıraya sokulur.
b - Kütükteki veriler arasında belirli bir değer aranır.
c - Bir kütük çiftlenir.

4 / Seri ve yarılamalı arama yöntemleri arasındaki başlıca farkları sayınız.

5 / Çalışanlarla ilgili bir veri arşivinde şu bilgilerin saklanması öngörülüyor: Adı, soyadı, işi, derecesi, yaşı (ya da işletmede çalışma süresi), maaşı, gelmediği günler. Alanların yaklaşık uzunluğunu tahmin ederek kayıt yapısını belirleyiniz.

6 / Depo yönetiminde bir kaydın yapısını belirleyiniz. Kayıt şu alanlardan oluşuyor: Malın cinsi, stok miktarı, en az miktar, son siparişin tarihi, gönderenin adı ve soyadı.

7 / 6. sorudaki yapıda, bazı alanlarda kodlama yaparak yer kazanmak mümkündür: Son sipariş tarihini (gün, ay, yıl) ve gönderenin adını kodlamayı deneyiniz. (En çok 100 göndericinin bulunduğu varsayılıyor.)

Testin çözümleri sayfa 286 ve 287'dedir.

4 / O harfi, değeri diğerlerinin tümünden yüksek olduğundan onların en sağına yazılır.

5 / Bir sonraki kayıtta gelen N harfi, O'dan önce yer almalıdır. Bu amaçla O harfi bir kayıt sağa kaydırılır, önceki yerine N harfi yazılır (Adım **5-1**).

6 / R harfi (kayıt 6) öbürlerinden daha yüksek bir değer taşır, dolayısıyla sıranın en sonuna yerleştirilir.

7 / Kayıt 7'deki C harfinin, B ve L arasına yazılabilmesi için, ikinci kütükteki son dört kayıt (L, N, O, R) sağa kaydırılır.

7-1 / Son kayıt olan C'nin doğru konuma yerleşmesiyle sıralama işlemi tamamlanmış olur.

Özel Amaçlı Veri Bankaları

Sürekli değişen dünyamızda özel ve kamu kesiminde çalışan iş uzmanlarının ve bilim adamlarının bilgi talebi de sürekli artmaktadır.

Zaman kazanmak ya da maliyet düşürmek gibi amaçlarla herhangi bir konudaki mevcut kaynakların son gelişmelerle sürekli olarak yenilenmesi büyük önem taşır. Bu gereklilik bilişim pazarını doğurmuştur. Veri bankası şirketleri önemli konulara ilişkin büyük miktarda bilgileri toplamakta, sıralamakta ve saklamaktadırlar.

Belirli konularda uzmanlaşan veri bankaları, belli bir konudaki kaynakların tümünü saklayabiliyor. Çoğunlukta olan genel amaçlı veri bankaları ise çeşitli alanda hizmet veriyorlar. Çeşitli istatistik ve ekonomik verileri toplayan veri bankaları bu ikinci gruptandır. Veri bankalarının giderek artan önemi, verileri gönderenlere kendi kütüklerini doğrudan yenileme olanağı sağlamasından ileri geliyor.

Küçük bir araştırma merkezinde çalışan bir arşivcinin, kendisini ilgilendiren yayınlar konusunda bilgi toplama olanakları sınırlıdır; ancak bir veri bankasından, güncel kütüphane bilgilerini içeren bir listeyi çabucak alabilir ve kendisi için olanları seçebilir.

Patent konusunda uzmanlaşmış veri bankaları da vardır. Bu bankalar sayesinde küçük ölçekli bir kuruluş bile, kendi ilgi alanındaki, yeni ürünler yöntemlerin ne gibi zarar ya da olanaklar getirebileceğini öğrenebilir.

Aynı biçimde tıp, uluslararası ticaret ve hukuk alanlarındaki son gelişmeleri de elde etmek mümkündür.

Bilişim endüstrisinin olanaklarını artırdığı ülkeler daha çok gelişmiş ve sürekli değişim halindeki bir üretim yapısına sahip olanlardır; bu ülke ekonomilerine dışardaki gelişmelerden sürekli uyarı gelmekte, yeni üretim yöntemleri üretici pazarında, yeni ürünler ise hazır ürün pazarında sürekli değişmelere yol açmaktadır. Bir bilişim sistemi, teknoloji, organizasyon, ekonomi konularındaki yeniliklere ilişkin karar verme yetkisini artırır, böylelikle gelişme düzeyini korumak ve hatta artırmak mümkün olur.

''Doğrudan bağlı'' (on-line) sistemlerde uzmanlaşmış ilk işletmeler Amerika Birleşik Devletleri'nde ortaya çıktı. Oradaki çok gelişmiş veri işleme sistemleri, iletişimde ulaşılmış son gelişmelerin sağladığı olanaklardan yararlanabiliyorlardı. Daha sonra bu teknik yerleşip, uygulama alanlarının sayısı artınca, Amerikalı üreticiler kütüklerini okyanusun ötesine de sundular. Yılların geçmesiyle Amerikan modeli kabul edildi, bugün artık gelişmiş teknolojiye sahip tüm ülkelerde bilgi sunucu kuruluşlar bulunuyor. Bunlardan bazıları Amerikan standartlarına uyarken, diğerleri belirli ulusal özellikler göstermekte.

Bu mirası Avrupa'ya özgü biçimde geliştirmek ve bağımsız olarak yeni teknolojiler bulmak amacıyla Avrupa Ekonomik Topluluğu'nun posta, telgraf, telefon hizmeti veren kuruluşları, EURONET adlı ilk Avrupa iletişim ağını kurdular. EURONET'in kuruluşu sırasında, Avrupa'daki veri bankası üreticilerinin serbest rekabet içinde çalışması ve böylece herkesin herhangi bir kısıtlama olmaksızın bilgilere erişebilmesi üzerinde anlaşmaya varıldı. Ticari mallar konusunda gerçekleştirilmesi güç olan bir karar ile, iletişim alanındaki tüm sınırlar kaldırılmıştı.

EURONET 1980'den bu yana çalışmaktadır ve halen iletişim alanında, DIANE (Direct Information Access Network in Aurope = Avrupa Doğrudan Bilgi Erişim Ağı) adı altında birleşmiş olan AET veri bankalarına hizmet veriyor.

Bu iletişim ağının özelliği uluslararası olmasıdır. Başka bir ülkedeki veri bankasından yararlanmak isteyenler EURONET ile bir anlaşma yapar. Kullanıcının, telefon hattına bağlanabilen bir terminali olması gerekir. Ağın düğüm noktalarını oluşturan bilgisayarlar bağlantıyı otomatik olarak gerçekleştirir.

İtalya'daki bir kullanıcı yalnızca Roma veya Milano'daki düğüm noktasını arar ve ülke içi bir bağlantıyla Büyük Britanya, Hollanda, Almanya, Fransa, Belçika ve Lüksemburg'daki veri bankalarına erişebilir. Uzaklığın bir rolü yoktur, ancak kullanıcının masrafları karşılaması gerekir. EURONET ücretlerinin Amerika'dakilerden daha düşük olduğu da bilinmektedir.

Amerika'daki veri bankalarına bağlanmak da daha karmaşık bir yöntemi gerektirmez. Bunun için de İtalya'dan çok sayıdaki Amerikan ağlarından birine telefon bağlantısı kurmak üzere bir anlaşma yapılması yeterlidir.

EURONET'de, veri bankalarının çoğunluğunda olduğu gibi sadece gerçek kullanım hesap-

lanır. Çünkü geleneksel telefon bağlantılarının aksine veri bankalarına ulaşımda maliyeti arttıran, kullanıcının bankadan olan uzaklığı değil, veri bankasından istenen toplam veri miktarıdır.

İletişim ağlarıyla erişilebilen veri bankaları, içerik ve işlev bakımından belirgin farklarla birbirlerinden ayrılır. İçeriğe bağlı temel bir farklılık, ''Genel Veri Bankaları'' ve ''Özel Veri Bankaları'' adlarıyla ortaya çıkar. Genel veri bankaları, ''İkincil Yayınlar''ın saklandığı kütüklerden oluşur ve terminal aracılığı ile erişilebilir. İkincil yayınlar, dergi, monograf, kitap, rapor ve konferans gibi yayınların yorumlarıdır; çeşitli gruplardan kişilere, belirli bir konu üzerine kısaltılmış biçimleriyle bilgi sunarlar.

Üretici kuruluş çeşitli yayınlardan geleneksel biçimde kağıt üzerinde kısa raporlar hazırlamış bir kurum, belirli bir üretim kesimiyle bağlantısı olan ve uygun mesleki yayınları tarayan bir dernek, bir üniversite veya veri işleme alanında deneyimli özel bir şirket olabilir. Bu durumda kütüklerin içeriği çeşitli meslekleri kapsayan niteliktedir. Her makale, monograf, ya da alıntı için şu bilgiler verilir: ad; yazar; kaynak (dergi, monograf, makale,..) ve kaynağa ilişkin bilgiler; içeriği doğru değerlendirmeye yardımcı, konunun kapsamını belirten, metinden alınmış tanımlayıcı veya anahtar sözcükler; metin hakkında fikir sahibi olmaya yetebilecek uzunlukta, kısa bir rapor.

Bu veri arşivlerine uygulanan bir seçme yöntemi ile arşivcinin bulabildiği tüm bibliyografik bilgiler elde edilir. Arşivci bu sonucun elde edilebilmesi için aranan konuyu tanımlayan belirli sayıda özgün ögeyi eklemiştir.

Bu bilişim olanakları, özellikle bilgi kaynağından uzak, bu nedenle işinde gereksindiği yayınları elde etmesi güç olan kullanıcılara yararlıdır. Kullanıcı, bir terminal yardımıyla kendisine gerekli yazıları seçer. Bir yerleşim merkezinde kütüphane yoksa, ya da düzenli çıkan dergilere bulunmayan çok özel yazılara ihtiyaç varsa, bu tekniği kullanmak zorunlu hale gelir.

Avrupa'da bibliyografik veri bankaları alanındaki bu gelişmeler oladursun, Amerika'da veri bankası mantığı bir adım daha da ileri götürüldü:

1980'de çeşitli bibliyografik bilgiler sunan veri bankalarından elde edilen gelir, bu alanda çalışan yayınevleri ve diğer kuruluşların toplam gelirlerinin ancak % 10'unu oluşturuyordu. Bu tür hizmetler veren kuruluşlar ise daha çok ya resmî kuruluşlar, ya da kâr amacı gütme-

Veri bankalarının özellikle bilim alanında büyük yararı vardır. Fotoğraf, Greenbank radyoastronomi gözlemevinin veri toplama ve işleme sistemini gösteriyor (Virginia, ABD).

yen bilimsel kuruluşlar idi. % 10 gibi düşük bir oran veri bankası işine kâr amacı için girecek özel şirketlere çekici gelmiyordu. Bu yüzden özel şirketler bibliyografik bilgi değil, doğrudan birinci elden kaynak bilgi sunan veri bankalarına yöneldiler.

Ancak bu tür kaynak bilgi sunan veri bankaları bibliyografik bankalara kıyasla çok daha uzman bir ekip gerektirdiğinden, maliyetleri de daha yüksek oluyordu. Kuşkusuz bu maliyet doğrudan doğruya piyasa fiyatlarına yansıyor. O kadar ki, bu veri bankalarını kullanmanın bedeli saati 100 dolara kadar çıkabiliyor.

Veri bankası mantığının en uç noktasına ulaşmış olduğu sistemler ise büyük bir olasılıkla New York Times gibi gazetelerin ve Times gibi dergilerin kurmuş oldukları elektronik haber merkezleri. Burada bankadaki verilerin sürekli olarak yenilenmesi ve güncel tutulması gerekiyor. Merkeze abone olan kullanıcı ise bazı anahtar kelimeler ya da bir fihrist aracılığıyla istediği konulardaki haberlere ulaşabiliyor.

Ancak bibliyografik olsun, kaynak bilgiler ya da haberler içersin, herhangi bir veri bankasındaki verilere nasıl ulaşılacağı, bu ulaşım sırasında kullanılacak yöntemler sorunu bu endüstrinin gelişmesini sınırlandıran engellerin başında geliyor. Çünkü henüz bu alanda ciddi bir standartlaşmaya gidilmiş değil.

Kuşkusuz başka bazı kısıtlar da mevcut. Örneğin burada insan faktöründen kaynaklanan bir kısıtlamadan söz edebiliriz. Eğer veri bankaları daha da yaygınlaşacaksa, bu işi gerçekleştirecek bilgi, beceri ve birikime sahip kurumlar yayınevleridir. Ancak geleneksel olarak yayıncılık sermaye-yoğun olmaktan çok emek-yoğun bir iştir. Genellikle, yayıncılığı seçen insanlar da emek-yoğun bir işte çalışmayı tercih ettikleri için bu işi seçerler.

Oysa veri bankacılığına girişmek büyük sermaye yatırımları gerektiren bir iştir. Dolayısıyla bu işe kalkışacak bir yayıncının hazır sermayeye duyduğu ihtiyaç bir yana, işletmesinin ve çalışma tarzının temel bazı özelliklerini değiştirmeyi de göze alması gerekir.

Ayrıca yayıncılık geniş okur kitleleri ile belli bir tarzda etkileşimi öngören bir iş. Birinci elden bilgiler içeren veri bankalarının (ki geleneksel yayıncıları özellikle ilgilendirecek veri bankaları bunlardır) yüksek maliyeti düşünülürse, elektronik yayıncılığın yayınevi-okur ilişkilerinde de önemli değişikliklere yolaçması gerektiği anlaşılır. Bu koşullarda elektronik yayıncılığa girişmeye karar veren bir yayınevinin önünde iki seçenek vardır. Yalnızca abonman ücretleri ile işi döndürmek istiyorsa, paralı, çok özel niteliklere sahip ve sınırlı bir okur kitlesine yönelmesi gerekir. Örneğin elektronik olarak borsa haberleri yayınlayan Dow Jones Şirketi'nin çözümü budur. Veri bankasının borsa haberleri ile sınırlandırılması hem maliyeti düşürür, hem de abonmanların yüksek kullanım fiyatları ödeyebilecek bir grup olmasını temin eder.

Diğer yandan geniş bir okur kitlesi ile ilişki sürdürülmek isteniyorsa, bu sefer de reklam gelirleri yayınevi için daha da önemli bir yer tutmaya başlayacaktır. Çünkü geniş kapsamlı bir veri bankasının çalışma maliyetini yalnızca abonman gelirlerinin karşılaması beklenemez.

Bütün bu tür gerekçelerden ötürü bugüne kadar elektronik yayıncılıktan önemli kârlar sağlayan şirketler, asıl işleri yayıncılık olmayan şirketler oldu. Örneğin Citibank'ın mali haberler içeren veri bankası, herhangi bir yayınevinin rekabet edebileceği boyutları aşmış durumda. Bu koşullar karşısında bir yayıncı, endişesini şöyle dile getiriyor: ''1990'da elektronik yayıncılık 60 milyar dolarlık bir iş haline gelecek. Ancak bu 60 milyar doların masraf mı gelir mi olacağından emin değilim.''

Sonuç olarak önümüzdeki yıllarda veri bankalarının bilimsel ve diğer özel amaçlı uygulamalarının artacağı, fakat elektronik yayıncılığın geleneksel yayıncılığın ve elektrik sinyalinin kağıdın yerini almayacağı söylenebilir. Bu alanda olası olan ''melez'' diyebileceğimiz elektronik destekli geleneksel yayınların artmasıdır. Eğitim malzemeleri yayıncılığında bu melez uygulamalar zaten yaygınlaşmış durumda. Ders kitapları yayınlayan birçok yayıncı, kitapların yanında ek olarak eğitim programları, elektronik olarak saklanmış görsel malzemeler, vb. sunuyor. Gelecekte daha ayrıntılı bilgi için ilgili veri bankalarına gönderme yapan kitapların yayınlanacağı da düşünülebilir.

Sayılan işlemler, bir sıralama programının başlıca işlevleri olan, veri karşılaştırma ile ikinci kütükte bunun karşılığı olan kaydırmaları açıklıyor. Başlıca BASIC komutlarının açıklandığı bölümden sonra, bu konuya döneceğiz.

Asıl kütüğün bir kopyasının ikinci kütük olarak kullanılması iki sakınca doğurur: Yığın belleğin yapması gereken iş artar (sıralama programı ile, kütük tek iken çift olur) ve bu yüzden çalışma hızı belirgin bir biçimde düşer, çünkü yazılacak her kayıt için, ikinci kütükteki kayıtların kaydırılması gerekebilir. Dizin kütüğü kullanılırsa yığın bellek fazla yüklenmez ve gerekli kaydırmalar bilgisayarın kendi belleğinde yapılabilir. (Dizin kütüğü, veri içermediğinden bilgisayarın belleğinde yer alabilecek kadar küçüktür.) Böylece sıralama programıyla oldukça hızlı çalışabilir.

Veri Sıkıştırma

Bazı uygulamalarda, verileri, yığın belleklere yazmadan önce ''sıkıştırmak'' gerekir. Büyük miktarlarda veri işleyen büyük sistemlerde birkaç karakterlik yer kazanmak bile önemlidir. Bu durumda, yığın belleklerden daha fazla yararlanabilmek için, verileri sıkıştıran ve daha sonra yine asıl biçimine dönüştüren karmaşık programlar, kullanmaya değer. Küçük ve kişisel bilgisayarlar için, aşağıda açıklanacak olan bazı özel durumlar dışında, böyle karmaşık yöntemler pek salık verilmez.

Başlıca sıkıştırma yöntemleri boş alan ve karakterlerin atlanması ve bazı karakter bileşimlerinin kodlanmasına dayanır.

Boş alanlar: Veri alanlarından biri kullanılmıyorsa, boş bir alan olarak kaydedilmek yerine tamamiyle iptal edilir, sonuçta diğerlerinden daha kısa bir kayıt ortaya çıkar. Gerçi bu yöntem yer kazandırır, ama değişen uzunlukta kayıtların işlenmesi, karmaşık programları gerekli kılar.

Gereksiz karakterlerin kaldırılması: ASCII koduyla kodlanmış alanlarda iki karakter arasındaki boşluk kaldırılabilir. Aynı şekilde sayısal alanlardaki sıfırları da göstermemek mümkündür. Ama bu işlemler yapılırsa verilerin sonradan yine ilk duruma getirilebilmesi, iptal edilen boşluk ve sıfırların sayı ve konumlarının saptanması için karmaşık programlara ihtiyaç vardır.

Karakter bileşimlerini kodlama: Bazı karakter bileşimleri veriler içinde tekrarlanıyorsa, bunlar tek bir simgeyle karşılanabilir. Örneğin işçilerle ilişkili bir kütükte Kutlu, Köksal ve Sezgin soyadları, utlu = 1, öksal = 2 kodlamasıyla K1, K2, Sezgin biçimine çevrilebilir. Doğal olarak bu yöntem, bu karakter bileşimleri (utku, öksal) kodlamayı gerektirecek kadar sık yineleniyorsa uygulanır.

Büyük Sistemlerde Kütük Yönetimi

Kütük işleme programlarının yazılmasında veya değiştirilmesinde ortaya çıkan sorunlarla büyük bilgisayarlarda karşılaşılmaz. Oysa bazı büyükçe hesap makinelerinde ve küçük bilgisayarlarda çok karmaşık kütük yönetimi programları vardır.

Bu programlar sayesinde kullanıcının, kütüğün fiziksel yapısı ile uğraşmasına gerek kalmaz, kullanıcı yalnız mantıksal yapıyı tanımlar ve verilerin nasıl dizileceğine karar verir. Bundan sonra mantıksal yapıyı fiziksel düzene sokmak ve işletilmesini yönetmek sistem programlarının işidir. Dolayısıyla bu programlara başvurulduğunda, kullanıcı verilerin fiziksel konumunu bilmez, verilerin yığın belleklere (manyetik diskler, manyetik tamburlar) paylaştırmak ve yazmak sistem programlarınca yürütülür. Sayfa 284'teki şekiller iki yöntem arasındaki nitel farkları gösteriyor. Kütük yönetimi için programların bulunmadığı birinci yöntemde, kullanıcı hem mantıksal yapıyı (kütüğün içeriğini; örnekte soyadı, ad ve adresten oluşan verileri), hem de kaydın fiziksel yapısını ve kayıtların kütük içindeki konumunu belirler.

İkinci yöntemde ise kütük yönetimi programlarından yararlanılır. Bu yöntemde kullanıcı sadece mantıksal yapıyı saptar, verilerin nasıl ve nereye yazılacağını belirlemek, sistemin görevidir. Daha gelişmiş sistemlerde kütük yönetimi özel bir dille gerçekleşir, bu dil normal programlamada kullanılan dilden farklı olmalıdır, bir ''yardımcı'' dil de kullanılabilir.

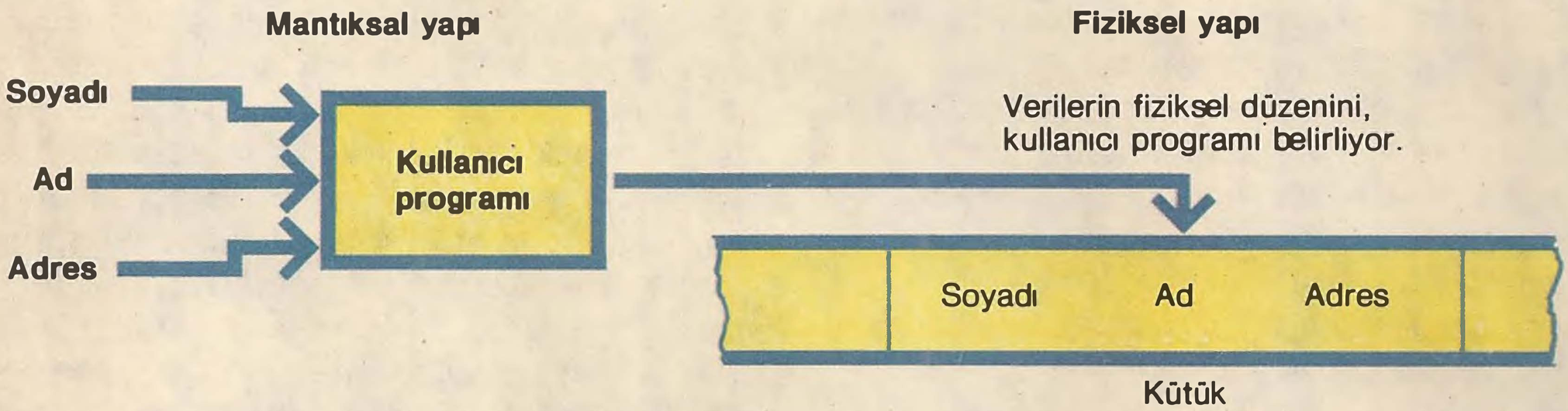
Bu son yöntemde küçük yönetiminin özel dili, genel bir programlama diline (Basic, Cobol vb.) bağlanmıştır. Uygulamada hem özel dilin, hem de genel dilin komutlarından yararlanılabilir. Konunun öneminden ötürü 1959'da Washington'da **CODASYL** (Conference on Data System Language - Veri Sistem Dili Konferansı) kuruldu. Bu kuruluşun görevi, tüm dünyada bu konuda yapılan araştırmaların sonuçlarını toplamak ve veri bankası sistemlerinde uyulması gereken normları belirlemek-

tir. (Veri tabanı (**database**) terimi, mantıksal bağıntılar kümesini ve alanlar arasındaki ilişkileri ifade eder.)

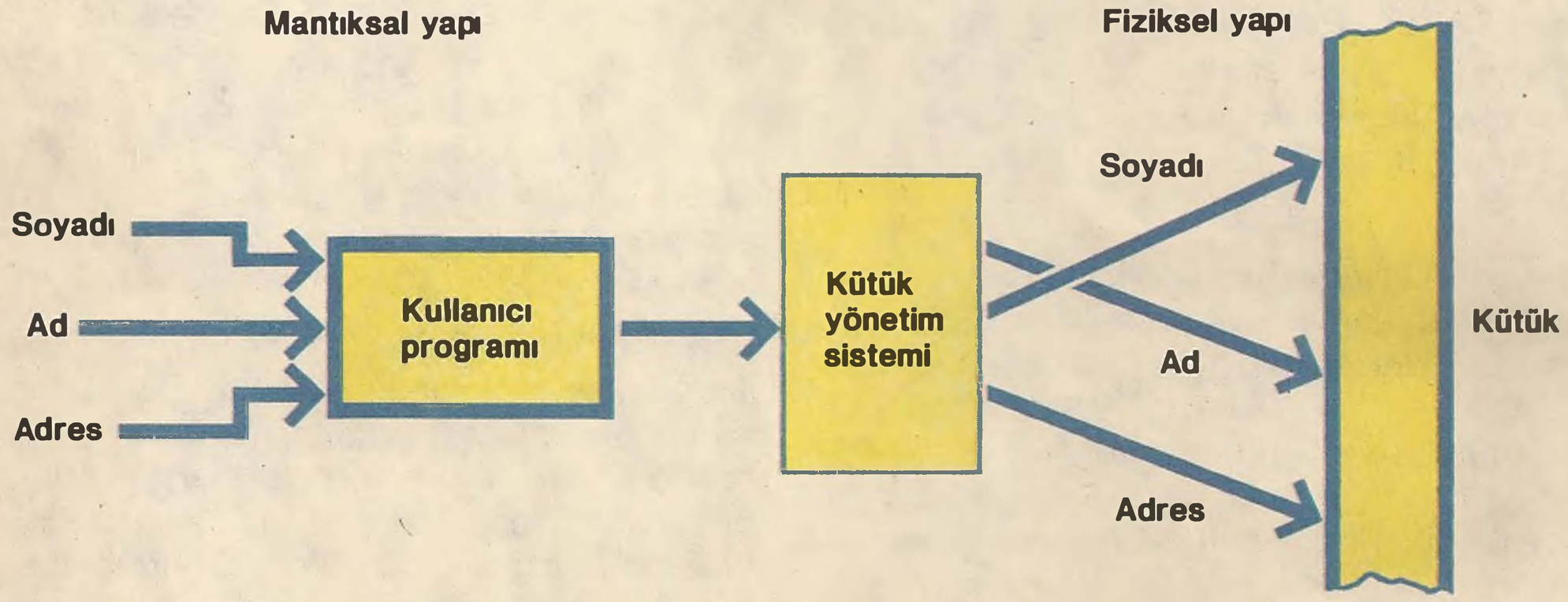
Karşılaştırmalı Veri Bankaları

Bu yöntemle verilerin mantıksal ve fiziksel yapılarından tamamiyle bağımsız çalışmak mümkündür. Karşılaştırmalı bir veri bankasında verileri bağıntılar yoluyla birbirine bağlayan mantıksal bir yapı vardır. Bağıntının anahtarını oluşturan ve ''dal'' adı da verilen belirli bir veriden çıkılarak, bu verinin diğerleriyle olan bağıntıları incelenir ve böylece birbiriyle ilişkili

VERİ İŞLEME VE KULLANICI PROGRAMLARI

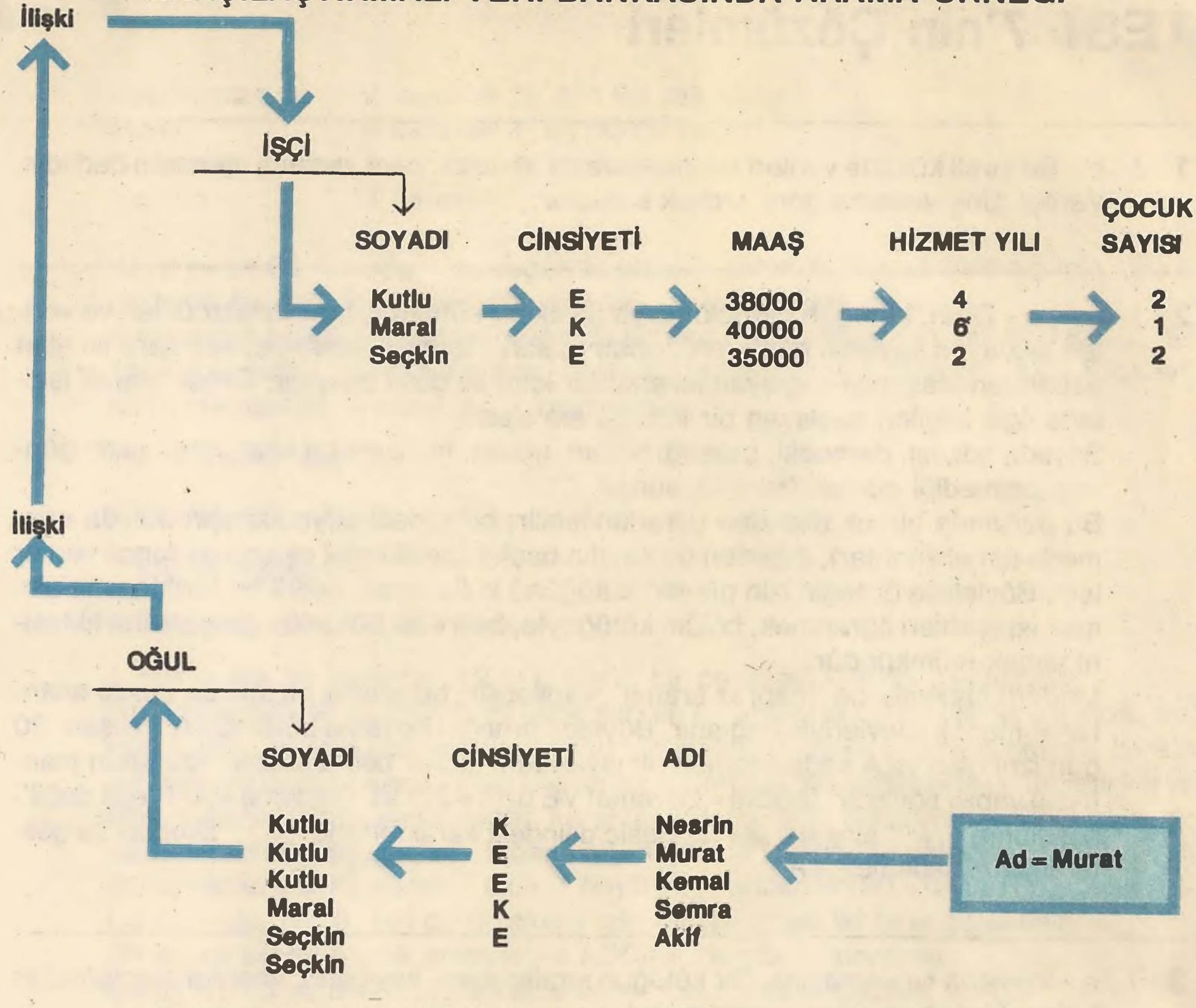


Veri işlemede modern sistemler kullanılmıyorsa, verilerin yığın bellekte fiziksel yerleşimi de dahil tüm işlevleri kullanıcı programı yerine getirir.



Büyük sistemlerde veri bankası işletimi için özel programlardan yararlanılması, programlama yöntemini temelden değiştirir. Kullanıcı verilerin yalnızca mantıksal yapısını tanımlar; verilerin nasıl ve nereye yazılacağını belirlemek sistem programlarının görevidir.

KARŞILAŞTIRMALI VERİ BANKASINDA ARAMA ÖRNEĞİ



Ad dalından (ad = Murat) çıkarak, oğul-işçi ilişkisi yoluyla işçi Kutlu'ya ilişkin tüm veriler ayrılır.

tüm veriler elde edilir. (Bkz. şekil) örnekte, kütüğün bağımlı ögeleri, yani veriler iki ilişkiye göre saklanmıştır; işçi olmak ve oğul olmak. Ad alanındaki belirli bir değerden (Murat) hareket edilirse işçiye ilişkin tüm veriler elde edilir. Karşılaştırmalı veri bankası en gelişmiş veri işleme biçimidir ve çok karmaşık ve gelişmiş veri tabanı yönetim sistemleri (data base mangement systems) gerektirir. Veri tabanı yönetim sistemi içerdiği bir şema sayesinde kullanıcının istediği bilgilerin hangi veri kayıtlarında saklandığını ve bu veri kayıtları arasındaki bağıntıları bulur. Ancak bu sistemlerin karmaşıklığından ötürü karşılaştırmalı veri bankaları yalnızca büyük sistemlerde bulunur (örneğin IBM S-38'de) ve belirli bir karmaşıklık düzeyinin üstündeki uygulamalarda kullanılır.

TEST 7'nin Çözümleri

1 / c - Bir sıralı kütükte verileri istenen sıraylą okumak veya yazmak mümkün değildir. Veriler giriş sırasına göre ardışık sıralanır.

---

2 / b ve c - Dizin, kütüğün bir bölümü ya da ek bir kütüktür, tanıtıcı sözcükleri ve verileri saklayan kayıtları gösteren ''göstergeleri'' içerir. Genellikle, kayıtlara iki alan üzerinden ulaşmayı sağlayan iki anahtar için, iki dizin bulunur. Örnek olarak işçilerle ilgili bilgileri saklayan bir kütüğü ele alalım:
Soyadı, adı, işi, derecesi, çalıştığı bölüm -maaşı, fazla mesai saat ücreti-, izin günleri, gelmediği günler, hastalık süresi.
Bu durumda birçok dizinden yararlanılabilir; bir tanesi soyadlar için (kayda erişmede temel anahtar), diğerleri de kaydın başka özelliklerini oluşturan ikincil veriler için. Böylelikle örneğin izin günleri kütüğünü kullanarak, belirli bir tarihte izne çıkmak isteyenleri öğrenmek, bölüm kütüğüyle, belirli bir bölümde çalışanların listesini almak mümkündür.
Modern sistemlerde ''çapraz arama'' yapılabilir; bu arama biçiminde çeşitli anahtarlar mantık işlevleriyle bağlanır. Böylece örneğin boyama bölümünde çalışan, 20 gün izni olan ve A kademesinde olmayan tüm işçiler belirlenebilir. Komutun mantıksal yapısı şöyledir: (bölüm = boyama) VE (izin = 20) VE (kademe = A) (''eşit değil'' koşulunu, '' ≠ '' simgesi yerine Basic dilindeki karakterlerle ''<>'' simgesi ile gösterebiliriz: kademe <> A.)

---

3 / a - Verilerin sıralamasına. Bir kütüğün sıralanması, kayıtların, anahtar alanlarından birindeki değere göre sıralanmasıdır.
Önceki örnekte (yanıt 2) çeşitli dizinleri (soyadı, izin, bölüm, ...) kullanarak değişik sıralama kurallarına göre, personel tabloları hazırlanabilir. (Her anahtar alanı için bir tane.) Bir sıralama programında da bir komutun içinde çeşitli koşullar birbirine bağlanabilir. Örneğin bölümlere göre bir sıralama yaparken, bir bölümdeki isimlerinde alfabetik bir sıraya uyması sağlanır. Sonuçta aynı bölümde çalışan Kutlu, tabloda Seçkin'den önce yer alır.

---

4 / Seri aramada, aranan kayıt bulununcaya dek, kütükte ondan önce yer alan tüm kayıtlar okunur.
Yarılamalı arama yönteminde kütük, birbirini izleyen her adımda ikiye bölünerek daha kısa sürede okunur. Bu yöntem yalnızca sıralı kütüklere uygulanabilir; dolayısıyla yarılamalı arama yapabilmek için önce bir sıralama programı yürütülür. Bir kütükte az sayıda veri varsa, sıralama ve yarılamalı arama işlemlerinin toplam süresi, seri arama için gereken süreye yakındır.

---

5 / Alanların yaklaşık uzunlukları şu değerler olabilir:

Adı ve soyadı	50 karakter
İşi	2 karakter (kısaltma)
Derecesi	2 karakter (kısaltma)

Yaş/Hizmet yılı	2	karakter (0'dan 99 yıla kadar)
Maaş	6	karakter (100.000'e kadar)
Gelmediği günler	2	karakter (99'a kadar)
Toplam	64	karakter

Kayıt en az 64 karakter uzunluğunda olabilir. Biraz da boşluk kalması için 16 karakterlik bir ekle 80 karakterlik bir kayıt uzunluğu elde edilir.
Bir disketin bellek kapasitesi hakkında bir fikir edinebilmek için, biri 250 KB, diğeri 1MB kapasiteli iki diskete kaç bayt sığdırılabileceğini bulalım (KB = Kilobayt, MB = Megabayt); 1 karakter, 1 bayt kaplar.

250 KB'lik disket için:

250.000 ÷ 80 = 3.125 kayıt

1 MB'lik disket için:

1.000.000 ÷ 80 = 12.500 kayıt

1 MB'lik tek bir diskette, 12.500 kişilik bir personel kütüğü saklanabilir.
Örneğimizde alan uzunluklarını hesaplarken rakamları karakter olarak ele aldık, bunun nedeni işletme sistemlerinin çoğunun manyetik diske yazma işleminde karakter biçimini kullanmasıdır. Bazı bilgisayar türleri sayısal değerlerin iki tabanlı karşılığını kullanır (yani sayısal değer, karakter katarı olarak ele alınmaz). Böylelikle yer kazanılır, çünkü sayıların iki tabanlı biçimi, ASCII kodundan daha kısadır. Örneğin 60 sayısı iki tabanlı olarak 8 bitle (1 bayt) ifade edilebilir (60 = 0111100), oysa ASCII biçimde, biri 6, biri de 0 rakamı için olmak üzere iki bayt gereklidir.
60 sayısı simge olarak anlaşılır ve ASCII kodunda gösterilirse:
6 = 0110110 ve 0 = 0110000 elde edilir. (Bkz. ASCII kodları, s.110).

---

6 / Bu örnek öncekinin bir benzeridir. Kayıt yapısı şöyle olabilir:

Malın cinsi	200	karakter (Malın tanımı)
Miktar	4	karakter (9999'a kadar)
En az miktarı	4	karakter
Tarih	8	karakter (gün = 2, ay = 2, yıl = 4, örnek: 21 11 1983)
Gönderen	20	karakter
Toplam	56	karakter, boş alanla birlikte 70 karakter.

---

7 / Bellekte yer kazanmak için kısaltma ve kodlama yapılabilir. Tarih, yalnızca son iki rakamla gösterilebilir, örneğin 1985 yılı 85 ile gösterilir. İkinci olarak gönderici alanında isimlerin karşılığı olarak sayılar kullanılır. 1 sayısı bir gönderici adını, 2 sayısı başka bir adı kodlar. Göndericilerin en büyük sayısı 100'dür, ancak 100'de 3 rakam bulunduğundan 3 rakamla 999'a kadar kodlama yapılabilir, dolayısıyla gönderici sayısının üst sınırı 99'a çıkar. Tarih alanında 2 bayt ve gönderici alanında 17 bayt olmak üzere toplam 19 baytlık yer kazanılmış olur, böylece 37 bayta inen kayda 13 baytlık bir boş alan eklersek 50 bayt uzunluğunda bir kayıt elde ederiz. 250.000 KB'lık bir diskette 5.000, 1 MB'lık diskette ise 20.000 mal türü ile ilgili bir kütük oluşturulabilir.

KÜÇÜK SÖZLÜK

Veri Arşivlerine İlişkin Sözcükler

Adres: Herhangi bir veri kaydının konumunu bildiren sayısal değer.

Ağaç: Bir kütükteki verilerin anahtar değerlerine göre sınıflandırılmasından oluşan yapı.

Alan: Veri kaydı birimleri. Örneğin stok bilgilerin içeren bir kütükte her veri kaydı malın tanımı için bir alan, fiyatı için bir alan vb.'den oluşur.

Algoritma: Belirli bir problemi çözerken kullanılacak matematiksel hesaplar ve işlemler bütünü.

Anahtar: Veri kayıtlarının aranmasında kullanılan özel bölüm.

Ara-Bellek: Veriler işlenirken kullanılan bellek bölümü. Kişisel bilgisayarlarda veriler yığın bellekte saklanmadan önce mutlaka ara-bellekten geçerler. Diskete yazılacak veriler de önce merkezi bellekteki ara-bellekten geçmeli. Bazı Türkçe kaynaklarda ara-bellek için ''tampon bellek'' deyimi de kullanılmaktadır.

Birleştirme: Birden fazla kütükten tek bir kütük yaratma yöntemi ve bu iş için kullanılan program.

Buffer: Bkz. Ara-bellek.

Data Base: Bkz. Veri tabanı.

Dolaylı Adresleme: Verilerin konum göstergelerinin bir dizin kütüğünde saklandığı adresleme biçimi. Böyle adreslenmiş veriler iki aşamada bulunur: Önce bir anahtar yardımıyla dizin kütüğündeki konum göstergesi bulunur; sonra konum göstergesi sayesinde aranan veri kaydı veri kütüğünde bulunur.

DBE: Doğrudan bellek erişimi.

DSEY: Doğrudan sıralı erişim yöntemi.

Erişim Süresi: Bir veriye ulaşmak için gereken süre. Okuma-yazma kafasının gerekli konuma yerleştirilmesi ve verinin yığın bellekten merkezi işleme aktarılması için gereken süre.

Erişim Yöntemleri: Okumak ya da yazmak için verilere ulaşma yöntemleri. Başlıca üç erişim yöntemi vardır: Doğrudan erişim, sıralı erişim ve doğrudan sıralı erişim (DSEY).

Konum Göstergesi: Veri kaydında konum gösteren sayısal bir değer içeren alan.

Kütük: Türdeş verileri içeren mantıksal ve fiziksel birim.

Öbek: Fiziksel kaydın bölümleri. Adreslemede kullanılan en küçük birimdir. Dolayısıyla okuma ve yazma işlemlerinde en az bir öbek kullanılmak zorundadır.

Sıkıştırma: Yerden tasarruf etmek amacıyla verileri saklarken kullanılan karakterleri azaltma yöntemi.

Veri Bankası: Belli birtakım konulardaki bilgileri içeren ve özel programlarla bu bilgilere ulaşma olanağı sağlayan kuruluşlar. Temel özellikleri buradaki verilerin onları kullanacak araştırmacıların kullanabileceği bir biçimde önceden düzenlenmiş olmalarıdır. Veri bankaları ikincil kaynaklar, kısa özetler, kaynakçalar içerebileceği gibi, orijinal belgelerin kendilerini de içerebilir.

Veri Tabanı: Belli bir mantıksal sıra içerisinde düzenlenmiş veri kütükleri. İşlevleri veri bankalarınınkine benzer. Ancak daha kısa notlar, özetler ve kaynakçalar içerirler. Verilerin standart bir yazı düzeni içerisinde girilmesi gerekir.

Yarılamalı Arama: Her adımda kütüğün uzunluğunu ikiye bölerek ortadaki verinin okunduğu veri arama yöntemi.

Zincirli Düzen: Her veri kaydının bir sonraki veri kaydını gösteren bir konum göstergesi içerecek şekilde girildiği veri saklama yöntemi. Özellikle veri tabanlarında kullanılır. Konum göstergesi, veri kaydında özel bir alan oluşturur.

Honos