BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DERS 4

Veri Sıkıştırma

Veri İşleme

*Ders sunumları ticari herhangi bir amaç içermez

Veri sıkıştırma

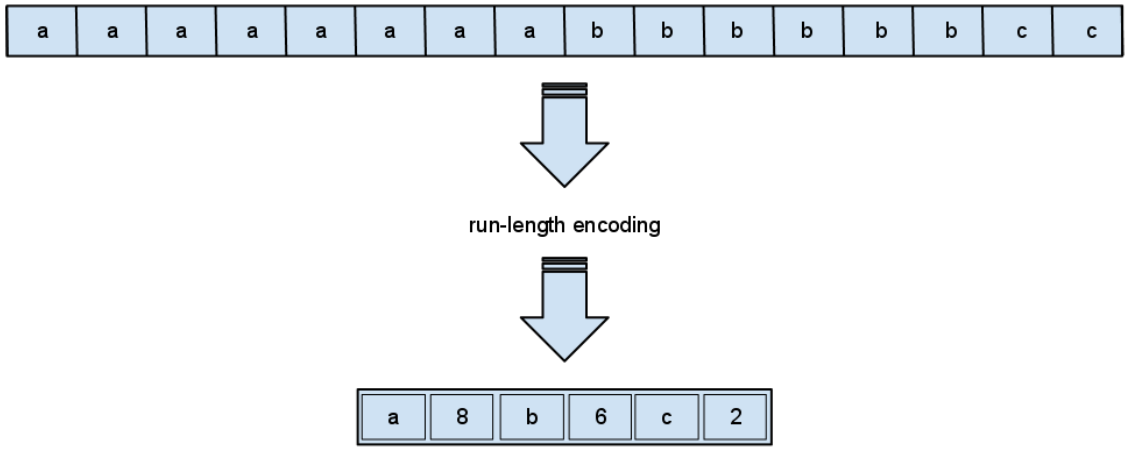
- Veri depolama veya taşıma amacıyla verinin boyutunu küçültme işlemine veri sıkıştırma denir.

Genel veri sıkıştırma yöntemleri

- Kayıpsız
- Kayıplı (Daha çok sıkıştırma sağlar, resim ses gibi tolere edilebilir verilerde kullanılır)

Sayı-uzunluk şifreleme

- Kayıpsız veri sıkıştırma yöntemidir.
- Aynı değerin uzun dizilerini içeren verilerde kullanılır.

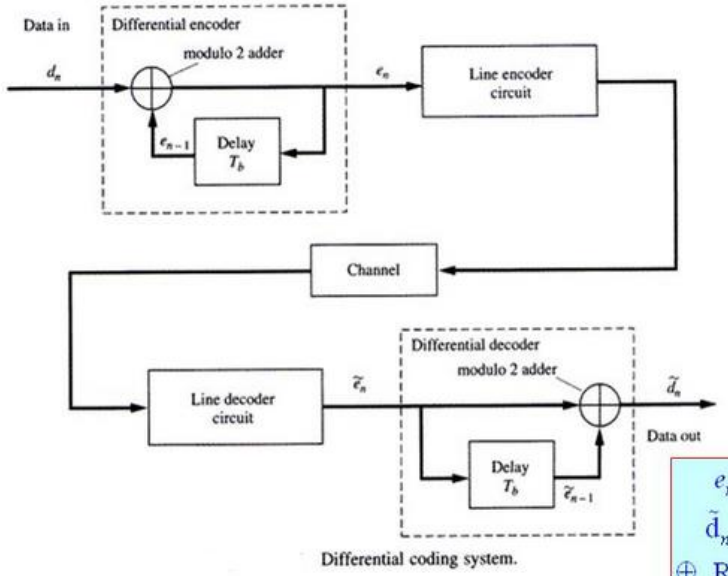


Frekans bağımlı kodlama

- Kayıpsız veri sıkıştırma yöntemidir.
- Bir veri parçasını göstermek için kullanılan bit deseninin uzunluğunu o öğenin kullanım sıklığıyla ters orantılı ilişkilendirildiği sistemdir.
- Örnek olarak İngilizce metinlerde e,t,i,a harfleri z,q,x harflerinden sık kullanılmaktadır. Bu nedenle çok kullanılan harfler daha kısa bit desenleriyle daha az kullanılan harfler daha uzun bit desenleriyle gösterilebilir.

Göreceli (Diferansiyel) kodlama

- Birbirini izleyen veri birimleri arasındaki farkları kaydeder.
- Kayıplı yada kayıpsız formda olabilir.



Differential coding system.

Modulo-2 addition
Exclusive OR

I_1	I_2	Out
0	0	0
0	1	1
1	0	1
1	1	0

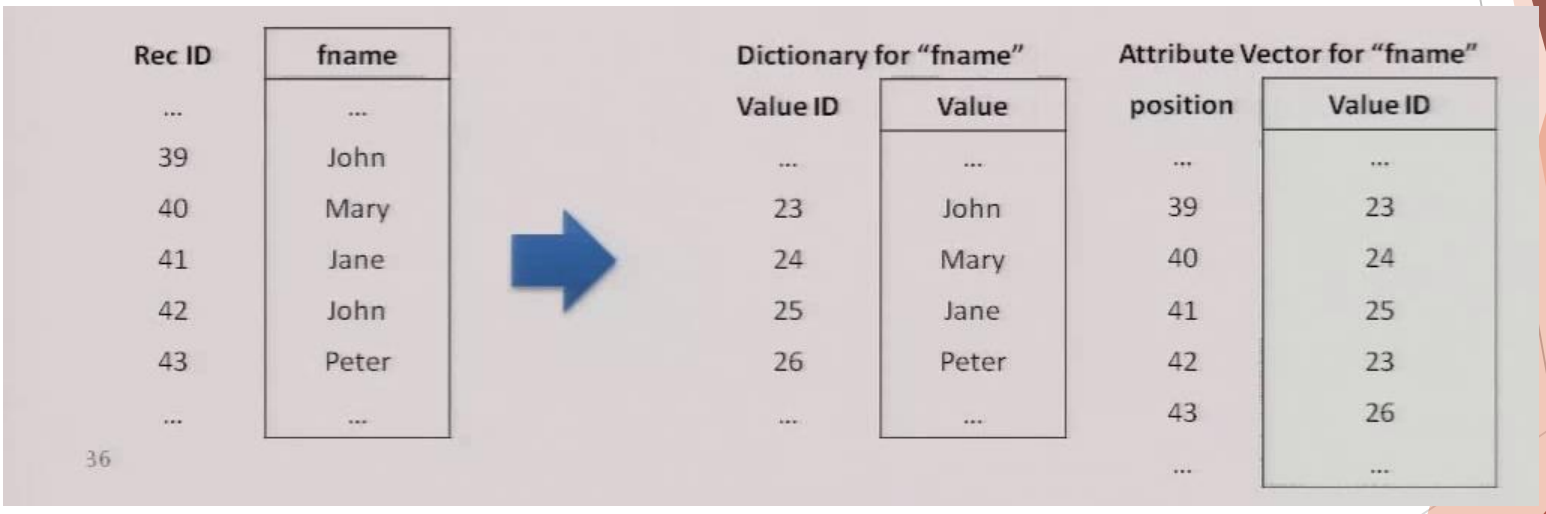
$e_n = d_n \oplus e_{n-1}$

$\tilde{d}_n = \tilde{e}_n \oplus \tilde{e}_{n-1}$

$\oplus$ Represents Modulo-2 adder (XOR)

Sözlük kodlama

- Metin sıkıştırma için örnek:



Rec ID	fname
...	...
39	John
40	Mary
41	Jane
42	John
43	Peter
...	...

Dictionary for "fname"

Value ID	Value
...	...
23	John
24	Mary
25	Jane
26	Peter
...	...

Attribute Vector for "fname"

position	Value ID
...	...
39	23
40	24
41	25
42	23
43	26
...	...

Adaptif sözlük kodlama

- Örnek (LZW):
- Kodlama:
- xyx xyx xyx xyx
- 1=x; 2=y; 3=[boşluk];
- 1213=xyx[boşluk]
- 4=xyx
- 12134343434

- LZW:Lempel-Ziv-Welch Kayıpsız veri sıkıştırma Algoritması

- Çözme:
- 12134343434
- xyx[boşluk]
- xyx xyx xyx xyx

Görüntü Sıkıştırma

- GIF
- JPEG
- TIFF

GIF (Graphic Interchange Format)

- Sözlük kodlama sistemini kullanır.
- Tüm renkleri sözlüğe eklediği 256 farklı renge göre kodlar
- Bu durumda kayıplı sıkıştırma yapar.
- Fakat görüntüleri bölgelere ayırarak kodlama yaparsa kayıpsız kodlama da yapabilir.

JPEG

- Joint Photographic Experts Group tarafından geliştirilen bir ISO standardıdır.
- JPEG temel standardı; insan gözünün sınırlarını avantaj olarak kullanır. İnsan gözü renklerdeki değişimden çok parlaklıktaki değişimlere daha duyarlıdır.
- Görüntüdeki orijinal parlaklık bilgisi aynı kalırken renk bilgileri dörtte bir oranında azaltılmaktadır.

TIFF

- Veri sıkıştırma alanından daha çok; tarih, zaman, kamera ayarları gibi bilgileri fotoğraflara depolanması için kullanılmaktadır.
- Genellikle faks metinleri gibi verileri kodlamak için dizayn edilmiştir.
- Bu nedenle sayı-uzunluk algoritmasına benzer bir algoritmayla çalışır.

Ses ve video sıkıştırma

- MPEG: Motion Picture Experts Group tarafından geliştirilen bir ISO standardıdır.
- MPEG videodan belli aralıklarala ana çerçeveler seçer. Bu çerçeveler JPEG'e benzer şekilde sıkıştırılır.
- Bir ana çerçeveden sonra her bir çerçevenin kendinden önceki çerçeveyle olan farkını kaydeder.

MP3

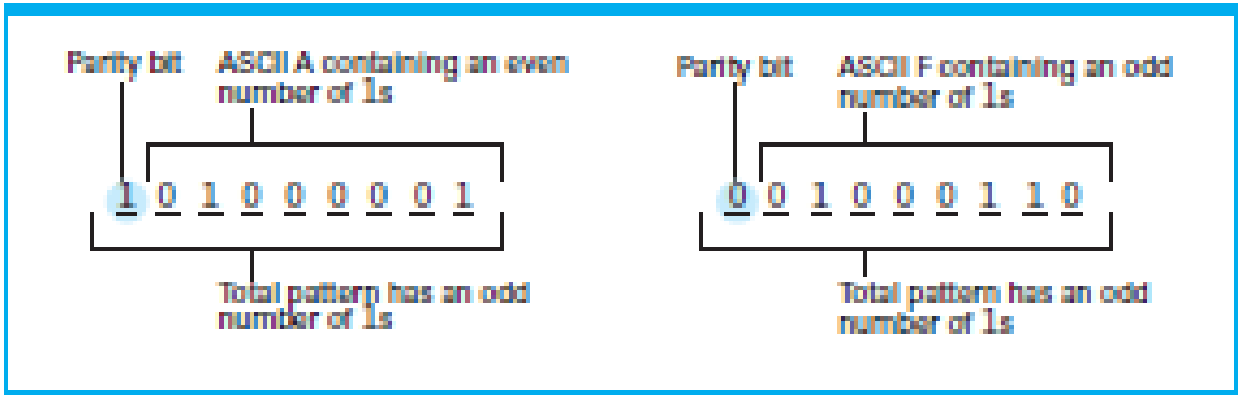
- MPEG Layer 3
- Ses sıkıştırması için yaygın olarak kullanılır.
- İnsan kulağının fark edemeyeceği detayları kaldırarak sıkıştırma yapar.

İletişim hatları

- Bir bilgi (bit deseni); bilgisayar içinde veya başka bir yere gönderildiğinde orijinalliğinin bozulmuş olma ihtimali vardır.
- Manyetik kayıt yüzeyindeki kir veya arızalı bir devre verinin yanlış okunmasına neden olabilir.
- Bu tür sorunları gidermek için farklı hata belirleme hatta düzeltme kodları kullanılmaktadır.

Eşlik (Parity) Bitleri

- Tek eşlikli sistem
  tek sayıda 1 biti içerir.
- Çift eşlikli: çift sayıda 1 biti içerir.
- Ana belleklerde de eşlik biti mevcuttur. Her bir hücre 8-bit olarak bilinmesine rağmen aslında 1 eşlik bitiyle beraber 9 bittir.
- Veri iki hataya birden uğradıysa eşlik biti işe yaramamaktadır.
- Bu amaçla uzun bit deseninden belirli bir düzene göre seçilen bitlerle de sağlama byte'ı (checkbyte) oluşturur.



Hata düzeltme kodları

- Eşlik bitleri, hatayı yakalamasına rağmen düzeltemez.
- Düzeltme kodları ise verinin düzeltilmesine yardımcı olur.
- Örnek: Hamming Mesafesi
- Her kod arasında en az 3 bitlik fark vardır.

Symbol	Code
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

Hamming Mesafesi

- Hatalı okunan bit deseni kod tablosunda bulunmayan 010100 deseninin kodunun çözülmesi kendine en yakın olan D karakteri olarak bulunur.

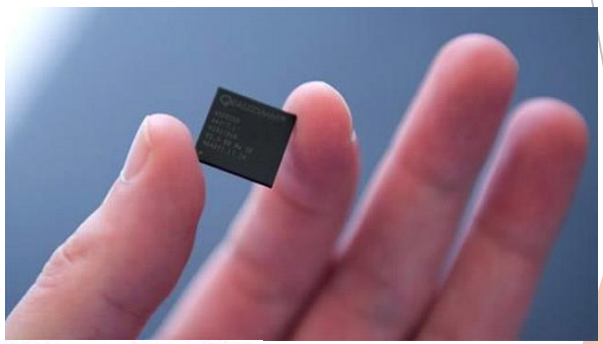
Character	Code	Pattern received	Distance between received pattern and code
A	0 0 0 0 0 0	0 **1** 0 **1** 0 0	2
B	0 0 1 1 1 1	0 **1 0** 1 **0 0**	4
C	0 1 0 0 1 1	0 1 0 **1 0 0**	3
D	0 1 1 1 0 0	0 1 **0** 1 0 0	**1** — Smallest distance
E	1 0 0 1 1 0	**0** 1 0 1 **0** 0	3
F	1 0 1 0 0 1	**0 1 0** 1 0 **0**	5
G	1 1 0 1 0 1	**0** 1 0 1 0 **0**	2
H	1 1 1 0 1 0	**0** 1 **0** 1 **0** 0	4

Veri İşleme

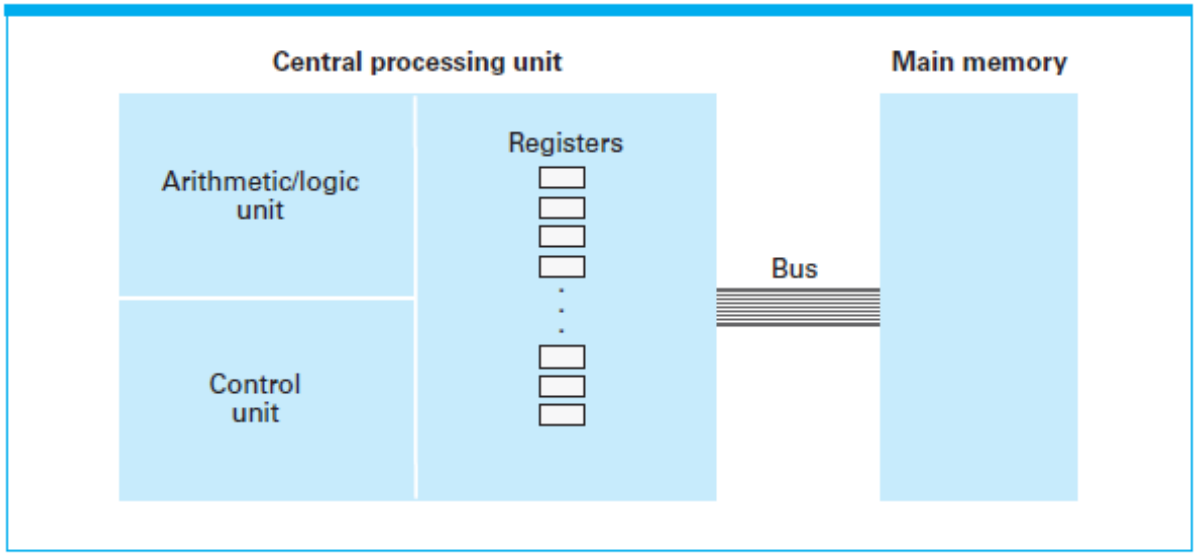
- Verinin taşınması
- Aritmetik hesaplamalar
- Metin veya resim düzenleme gibi işlemlerdir.

Bilgisayar Mimarisi - İşlemciler

- Bir bilgisayarda verinin işlenmesini kontrol eden devreye merkezi işlem birimi (CPU – Central Processing Unit) denir. Kısaca işlemci denir.



CPU Temelleri



- Registerlar (Yazmaçlar) CPU tarafından işlenen verinin geçici olarak saklanması için kullanılırlar.
- Aritmetik/lojik birimin giriş ve sonuç verilerini tutarlar.
- Kontrol birimi; ana bellekteki verileri, registerlara taşır ve ALU'ya bildirir.

Saklı Program Kavramı

- İlk üretilen bilgisayarlarda programlar üretim esnasında makinenin bir parçası olarak tanımlanırdı ve sonradan değiştirilemezlerdi.

- İlk nesil elektronik bilgisayarlar ise CPU'nun yeniden düzenlenebileceği şekilde geliştirildiler. Bu yeniden düzenleme işlemi santraldeki hatlar arasındaki bağlantıların elle sökülüp takılması gibi yapılıyordu.

- Bir programın da bir veri gibi saklanabileceği fikri ile saklı program kavramı doğdu.

Makine dili

- Bit desenleri şeklinde kodlanmış komutlardır.
- Makine komutları listesi oldukça kısadır.
- Yeni özellikler (komutlar) eklemek sistemin bazı işlemleri daha kolay gerçekleştirebilmesini sağlasa da bilgisayarın temel kapasitesini artırmaz.
- Yürütebileceği komut sayısına göre işlemciler 2 sınıftır. RISC ve CISC

RISC (Reduced instruction set computer)

- İndirgenmiş komut setli bilgisayarın avantajları
  - Daha verimli ve hızlı.
  - Üretim maliyetinin düşük olması.

CISC (complex instruction set computer)

- Karmaşık komut takımlı bilgisayarın avantajları:
  - Çok sayıda karmaşık komutu yürütebilir.
  - Günümüzün sürekli gelişmekte olan yazılımlarıyla daha iyi baş edebileceği savunulur.

RISC-CISC karşılaştırması

- CISC mimarisinde yazılan bir komut RISC mimarisinde komut kümesine karşılık gelebilir.
- Intel işlemciler CISC, PowerPC işlemciler (Machintosh bilgisayarların işlemcisi) RISC mimarisini kullandılar.
- CISC işlemcilerin günümüzde maliyeti oldukça düştüğünden nerdeyse tüm bilgisayarlarda CISC kullanılmaktadır.
- CISC işlemcilerin yüksek güç tüketimi nedeniyle mobil cihazlar için ARM (Advanced RISC Machine) firması RISC mimaride işlemci tasarladı.

İşlemci komut türleri

- Veri aktarma komutları
- Aritmetik/lojik komutları
- Kontrol komutları

Veri aktarma komutları

- Verinin bir yerden bir yere taşınmasını (kopyalanmasını) sağlayan komutlardır.
- CPU ve anabellek arasında;
- Anabellekten registera kopyalama işlemine YÜKLEME (LOAD) komutu
- Registerdan anabelleğe kopyalama işlemine KAYIT(STORE) komutu denir.
- CPU ile anabellek dışındaki cihazların (yazıcı, klavye, harddisk gibi) haberleşmesini ise Giriş/Çıkış (I/O) komutları sağlar.

Aritmetik/Lojik komutları

- AND, OR, XOR gibi boolean işlemlerin yapıldığı birimdir.

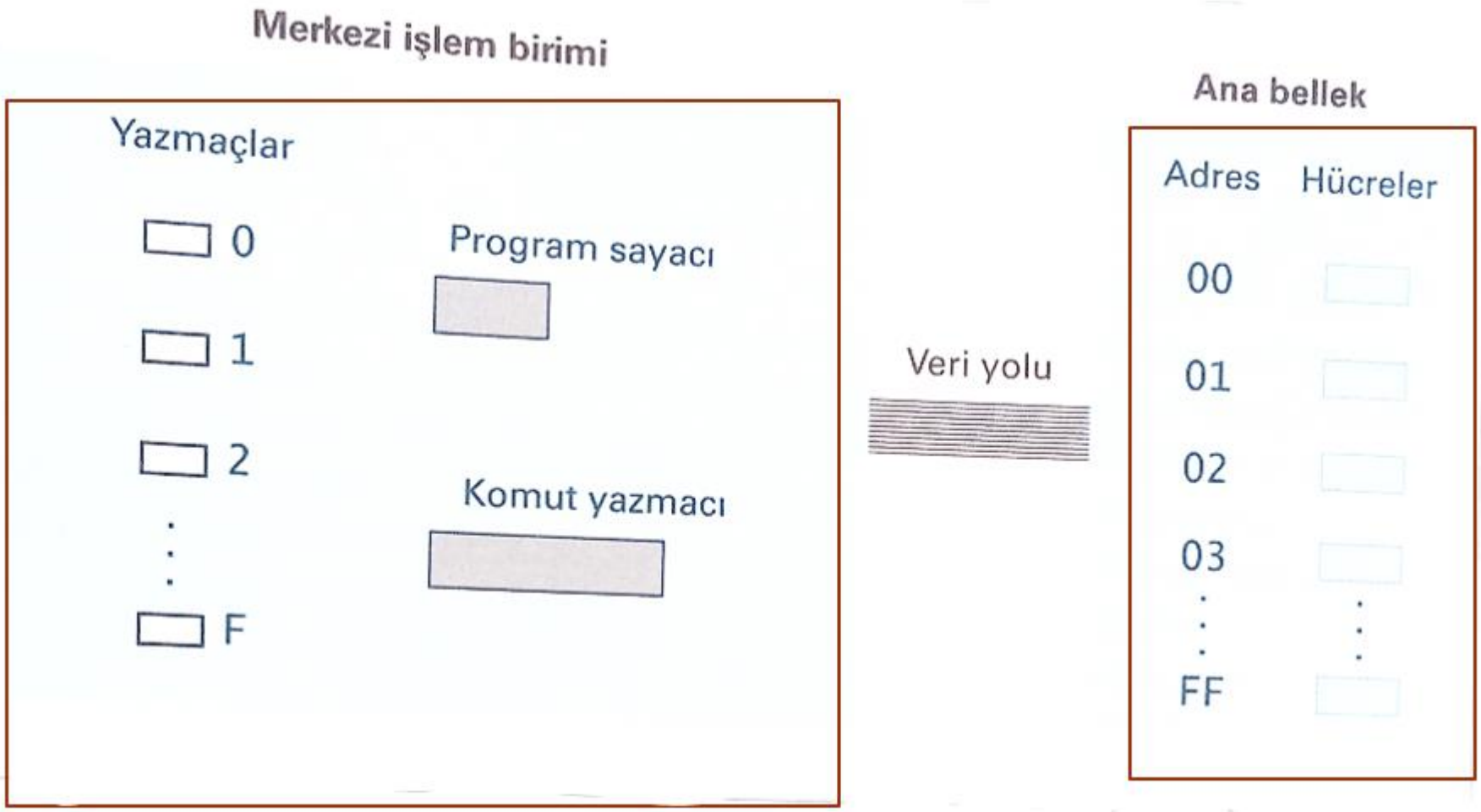
- Register içerisindeki bitler üzerinde Kaydırma (Shift) ve Döndürme (Rotate) komutları da gerçekleştirir.

Kontrol Komutları

- Programın çalışmasını yönetirler
- CPU'ya bellekteki bir sonraki komut yerine başka bir komutu işlemesini söyleyen ATLAMA (JUMP) komutlarıdır.
- Koşulsuz atlama ve koşullu atlama olarak iki türdür.
- Koşulsuz atlama: «Adım 5'e atla»
- Koşullu atlama: «Eğer elde edilen sonuç 0 ise adım 5'e atla»

Sanal bir makine dili örneği

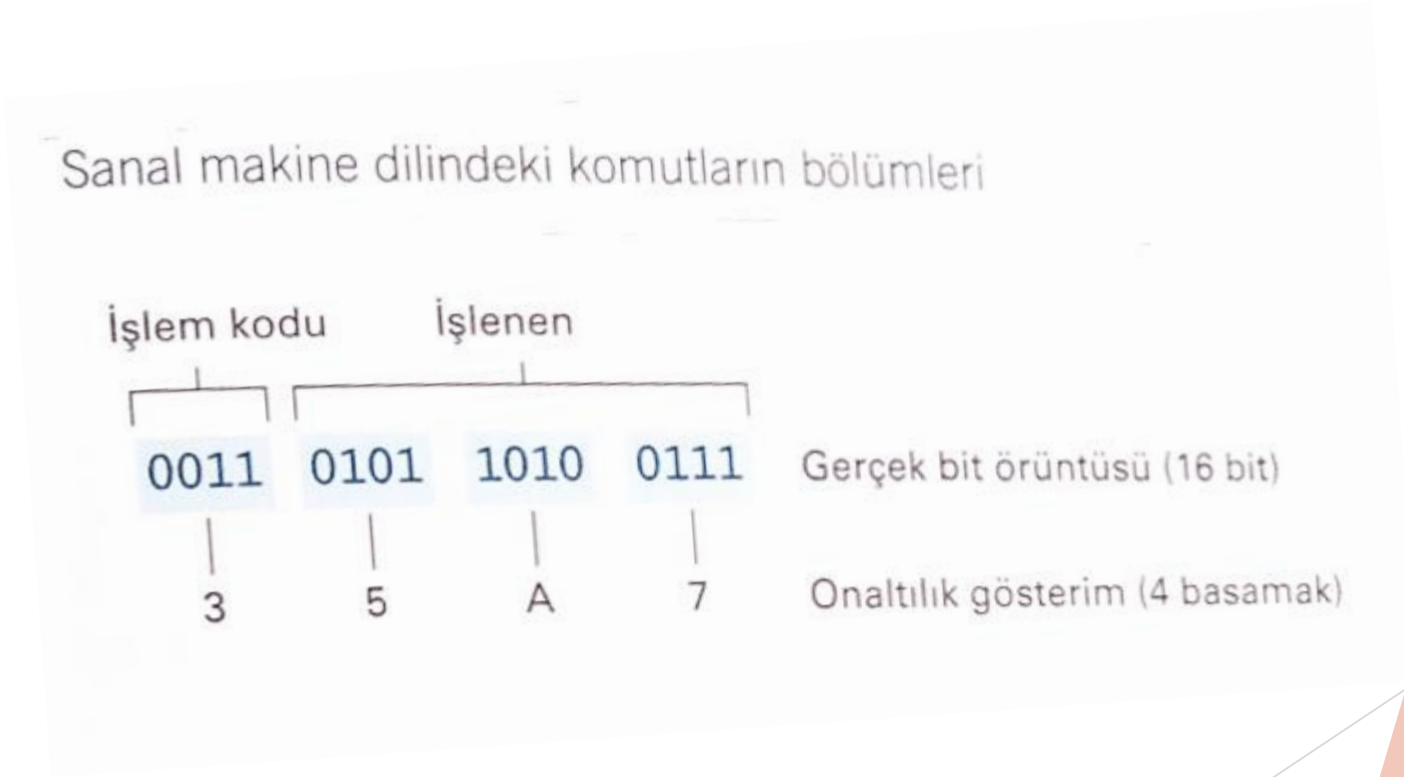
- 16 adet genel amaçlı register (0-F)
- 8bit 256 hücreli anabellek (00-FF)
- Bir makine kodu:işlem+işlenen kod
- (op-code+operand)
- Op-code:kaydet;atla vb.
- Operand:veri(register adres)
- 16 bitlik kodlama
- 12 temel komut


Merkezi işlem birimi
Yazmaçlar
0
1
2
F
Program sayacı
Komut yazmacı
Veri yolu
Ana bellek
Adres
Hücreler
00
01
02
03
FF

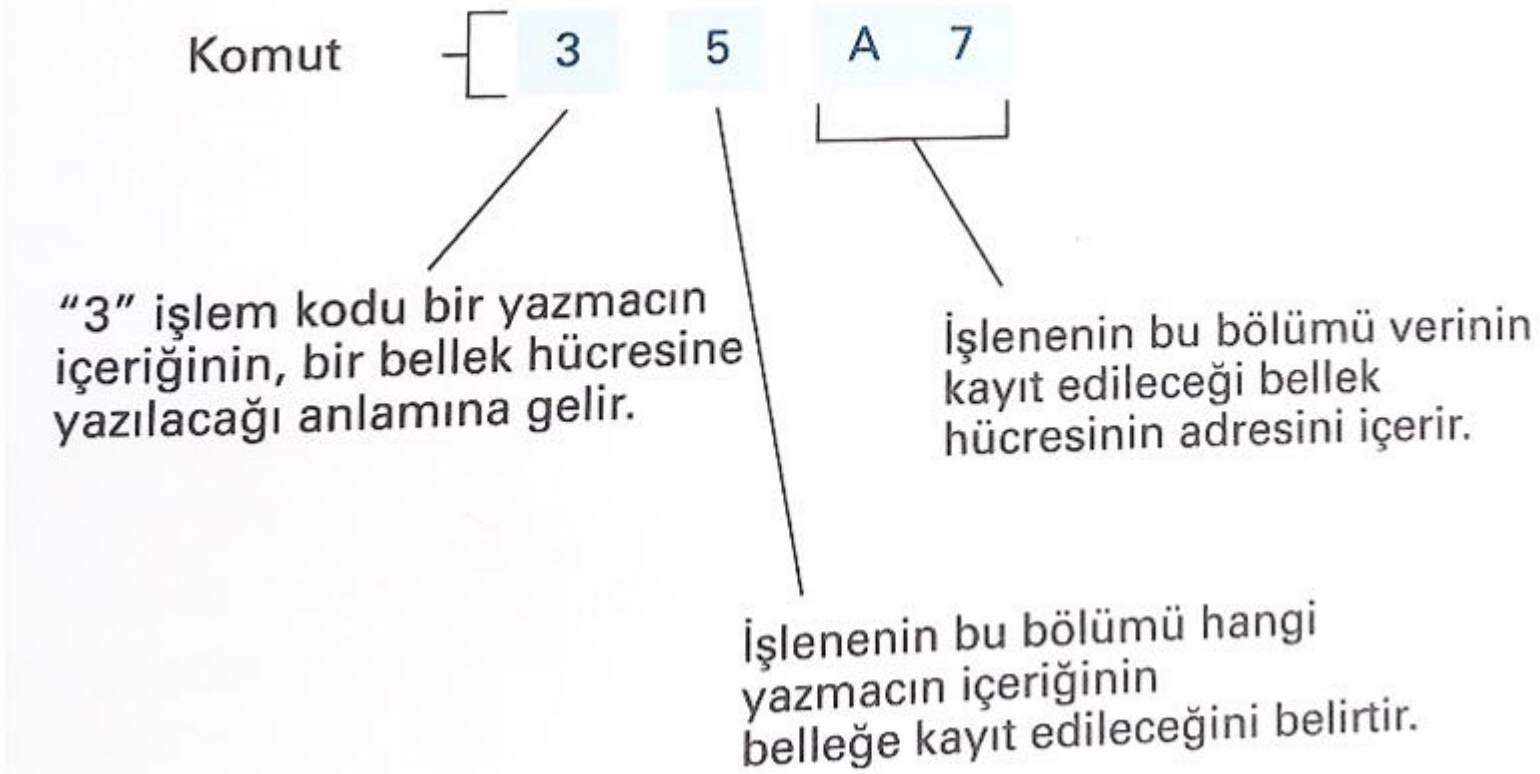
SANAL MAKİNE DİLİ

İşlem-kodu (Op-code)	İşlenen (Ope-rand)	Açıklama
1	RXY	Adresi XY olan hafıza hücresinde bulunan bit deseni ile R'yi YÜKLE (LOAD). *Örnek:* 14A3, A3 adresindeki hafıza hücresinin içeriğinin 4 numaralı yazmaca yerleştirilmesini sağlayacaktır.
2	RXY	R yazmacını XY bit deseni ile YÜKLE. *Örnek:* 20A3, A3 değerini yazmaç 0'a yerleştirmeye sebep olur.
3	RXY	R yazmacında bulunan bit desenini adresi XY olan hafıza hücresine KAYDET (STORE). *Örnek:* 35B1, 5 numaralı yazmacın içeriğinin belleğin B1 adresine yerleştirilmesine sebep olur.
4	0RS	R yazmacındaki bit desenini S yazmacına TAŞI (MOVE). *Örnek:* 40A4, A yazmacının içeriğinin yazmaç 4'e kopyalanmasını sağlar.
5	RST	Yazmaç S ve T'deki bit örüntülerini ikinin tümleyeni gösterimindeymiş gibi TOPLA (ADD) ve sonucu R yazmacına yaz. *Örnek:* 5726, 2 ve 6 numaralı yazmaçlardaki ikili sayıları toplar ve sonucun 7 numaralı yazmaca yerleştirilmesini sağlar.
6	RST	S ve T yazmaçlarındaki bit örüntülerini sanki değerleri kayan-nokta gösteriminde temsil ediyorlar gibi TOPLA (ADD) ve kayan-nokta sonucu R yazmacına yaz. *Örnek:* 634E, 4 numaralı yazmaç ve E yazmacındaki değerlerin kayan-nokta değerleri olarak toplanmasına ve sonucun 3 numaralı yazmaca yerleştirilmesine neden olur.
7	RST	S ve T yazmaçlarındaki bit desenlerine OR işlemini uygula ve sonucu R yazmacında tut. *Örnek:* 7CB4, yazmaç B ve 4'teki içeriğin OR işlemine alınmasını ve sonucun C yazmacında tutulmasını sağlar.
8	RST	S ve T yazmaçlarındaki bit desenlerini TOPLA ve sonucu R yazmacına yerleştir. *Örnek:* 8045, 4 ve 5 numaralı yazmaçlardaki içeriklerin AND işlemine sokulmasıyla elde edilen sonucun 0 numaralı yazmaca yerleştirilmesini sağlar.
9	RST	S ve T yazmaçlarındaki bit desenlerini XOR işlemine alarak sonucu R yazmacına yerleştirir. *Örnek:* 95F3 F ve 3 numaralı yazmaçlardaki içeriğin XOR işlemine alınarak sonuç 5 numaralı yazmaca yerleştirilir.
A	R0X	R yazmacındaki bit desenini X kere 1 bit sağa kaydır. Her seferinde, düşük değer ucunda başlayan biti, yüksek seviye uca yerleştir. *Örnek:* A403 4 numaralı yazmacın içeriğinin döngüsel biçimde 3 bit sağa kaydırılmasını sağlar.
B	RXY	Eğer R yazmacındaki bit deseni 0 numaralı yazmaçtaki bit desenine eşit ise, XY adresindeki hafıza hücresinde bulunan komuta ATLA (JUMP). Aksi durumda, yürütmeye normal sırada devam et. (atlama, yürütme sırasında program sayacına XY'yi kopyalayarak gerçekleşir) *Örnek:* B43C , önce 4 numaralı yazmaç ile 0 numaralı yazmaçtaki içerikleri karşılaştırır. Eşit ise, 3C deseni program sayacına yüklenir ve böylece çalıştırılan sonraki komut hafıza adresinde tutulan olacaktır. Aksi durumda, hiçbir şey yapılmaz ve programın yürütülmesi normal seyrinde devam eder.
C	000	Çalışmayı DURDURMA (HALT). *Örnek:* C000 programın yürütülmesinin durdurulmasına neden olur.

5 numaralı registerin içeriğini belleğin A7 adresine kaydet.



"35A7" komutunun çözümlenmesi



Bellekte yer alan iki sayının toplanması

Adım 1. Toplanacak sayılardan birini bellekten al ve bir yazmaca yerleştir.

Adım 2. Bellekten diğer sayıyı al ve başka bir yazmaca yerleştir.

Adım 3. Adım 1 ve 2'deki yazmaçları giriş olarak kullanacak ve sonucu başka bir yazmaca yazacak şekilde toplama devresini çalıştır.

Adım 4. Sonucu belleğe kaydet.

Adım 5. Dur.

Kodlanmış Komut	Anlamı
156C	Yazmaç 5'e 6C adresli bellek hücresinin içeriğini yükle
166D	Yazmaç 6'ya 6D adresli bellek hücresinin içeriğini yükle
5056	Yazmaç 5 ve 6'nin içeriklerini (2'ye tümleyen gösterimine göre) topla ve sonucu Yazmaç 0'da tut.
306E	Yazmaç 0'ın içeriğini 6E adresli bellek hücresine kaydet
C000	Dur.

Program Yürütme

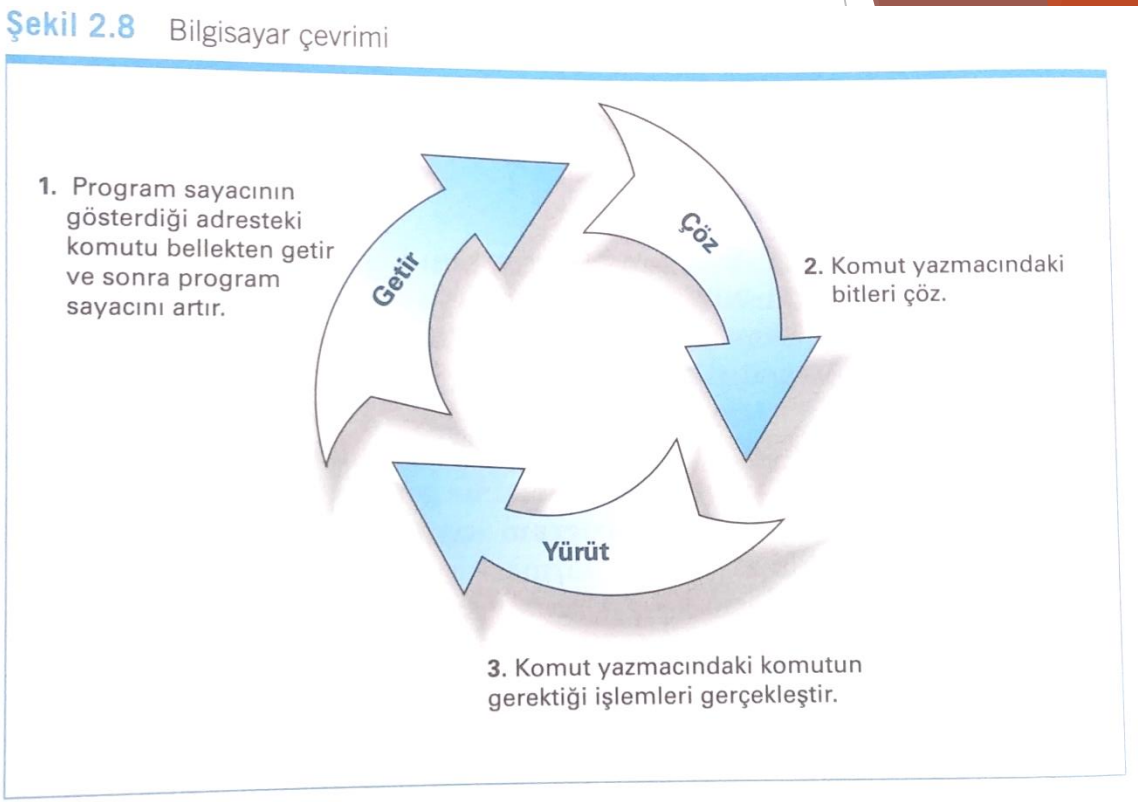
Program yürütülürken komutlar belli bir sırayla CPU'ya kopyalanarak çalıştırılır.

Komut register'ı: işlemci tarafından
işlenmekte olan komutu tutar.

Program sayacı: bir sonraki işlenecek komutun bellekteki adresini tutar.

CPU görevini bilgisayar çevrimi adı verilen 3 adımlık bir işlemi sürekli tekrar ederek gerçekleştirir.

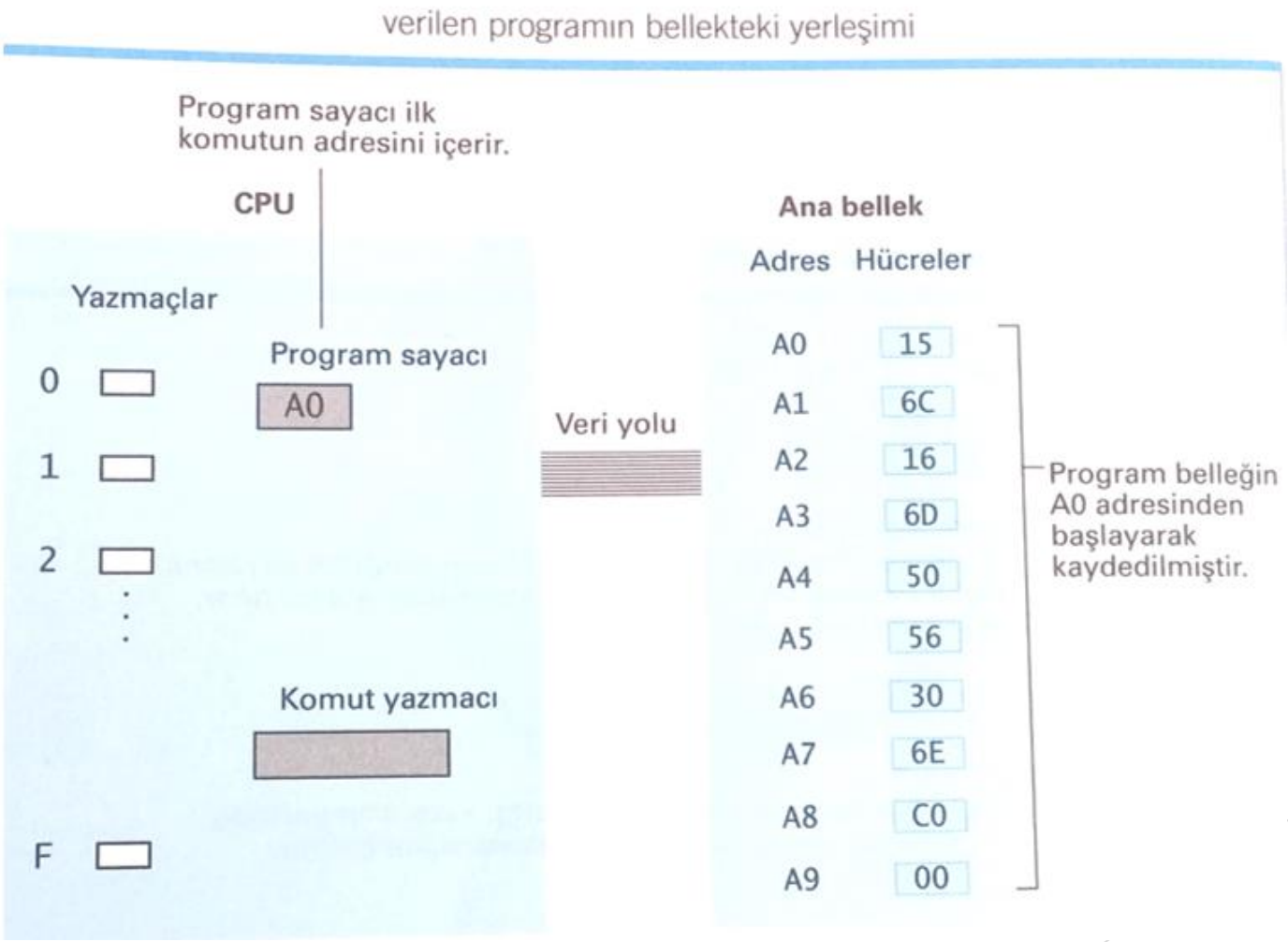
Şekil 2.8 Bilgisayar çevrimi



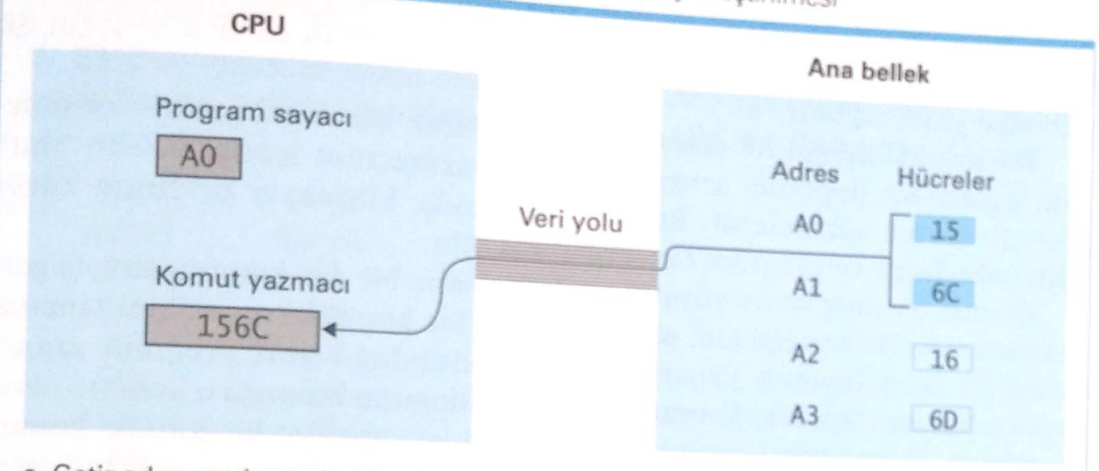
"B258" komutunun çözümlenmesi

- B058 olsaydı?

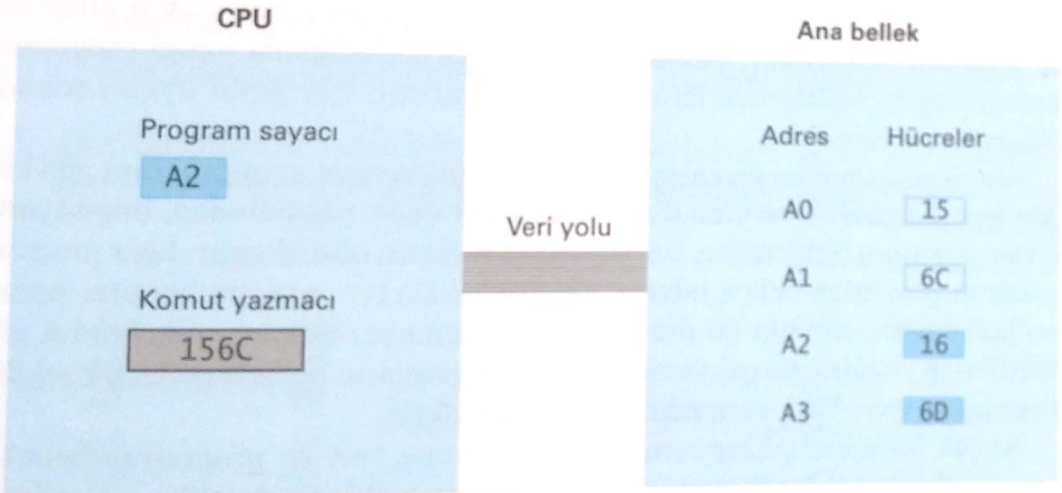
Bir program yürütme örneği



Şekil 2.11 Bilgisayar çevriminin Getir adımının gerçekleştirilmesi



a. Getir adımının başında A0 adresinde başlayan komut bellekten okunur ve komut yazmacına yerleştirilir.



b. Daha sonra program sayacı artırılarak bellekteki bir sonraki komutu

Aritmetik/lojik komutları Lojik komutlar:

```
    10011010
AND 11001001
    10001000
```

```
    10011010
OR  11001001
    11011011
```

```
    10011010
XOR 11001001
    01010011
```

- Maskeleme (and,or)
- tersini almak için (xor)

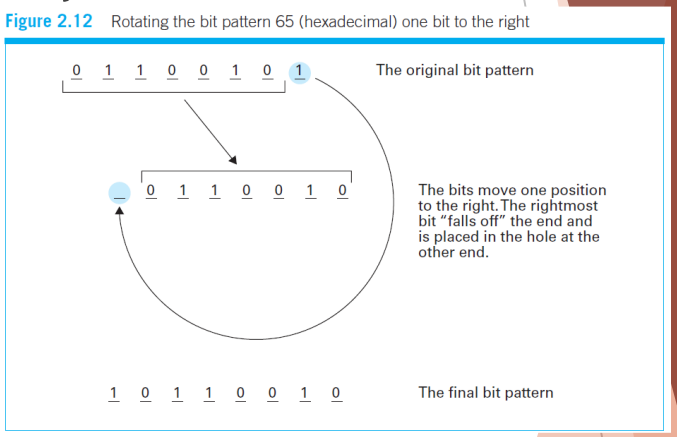
```
    00001111
AND 10101010
    00001010
```

```
    11110000
OR  10101010
    11111010
```

```
    11111111
XOR 10101010
    01010101
```

Döndürme ve Kaydırma işlemleri

- Sağa Döndürme: sağa kaydırılır ve en sondaki bit en başa gelir.
- Sola döndürme: sola kaydırılır ve en baştaki bit en sona gelir.



- Mantıksal Kaydırma:
- Sola kaydırma 2 ile çarpma
- Sağa kaydırma 2 ile bölme

- Aritmetik kaydırma: işaret biti korunarak yapılan kaydırmadır.

Aritmetik işlemler

- Çıkartma işlemi: negatifleme ve toplama işlemleriyle
- Çarpma işlemi: tekrarlanan toplama
- Bölme işlemi: tekrarlanan çıkartma

Ders bitti

Erciyes Üniversitesi
Selçuk Üniversitesi
Sakarya Üniversitesi
Kahramanmaraş Sütçü İmam Üniversitesi
ders notları kaynak ve içerik olarak kullanılmıştır.