

Lucrarea 9

Discuri compacte. Interfata ATAPI

1. Scopul lucrării

Lucrarea prezintă structura discurilor compact, formatele de înregistrare pentru acestea, modul de reprezentare a datelor pe suport. De asemenea, se prezintă principiile interfeței ATAPI, utilizarea registrelor, protocolul comenzilor, modul de execuție al unor comenzi, lista comenzilor, și se prezintă în detaliu unele comenzi.

2. Considerații teoretice

2.1. Standarde pentru discurile compact

Discurile compact CD (*Compact Disc*) reprezintă o categorie a discurilor optice. Primul tip de disc optic, care a utilizat un fascicul laser pentru citirea și înregistrarea informațiilor, a fost discul video, introdus în anul 1976. Acesta avea un diametru de 30 cm, datele fiind înregistrate sub formă analogică. Primul tip de disc compact a fost discul compact audio CD-DA (*Compact Disc Digital Audio*), care a apărut în anul 1983. Standardul pentru discul compact audio a fost elaborat în anul 1980 de firmele *Sony* și *Philips*, acest standard fiind publicat sub numele de “*Red Book*”. Acest document conține specificațiile pentru înregistrarea și codificarea datelor pe suport și pentru dimensiunile sale fizice: diametrul de 4,72 inci (120 mm) și grosimea de 1,2 mm. Ulterior, a fost elaborată o extensie a acestui standard, numită “*Red Book Plus*”, care descrie discul mixt. Acest disc conține atât date, cât și informații audio sau video, prima pistă fiind întotdeauna o pistă de date.

În anul 1985, firmele *Sony* și *Philips* au publicat standardul discului compact utilizat pentru înregistrarea datelor, CD-ROM (*Compact Disc Read Only Memory*), în documentul numit “*Yellow Book*”. Acest document se bazează în mare parte pe standardul pentru discul compact audio, suportul fizic fiind același. Firmele *Sony* și *Philips* au anunțat în anul 1986 standardul pentru discul CD-I (*CD Interactive*), care a fost publicat în documentul “*Green Book*”. În anul 1989, firmele *Sony* și *Philips* au extins funcționalitatea discului CD-ROM cu formatul CD-ROM/XA (*CD-ROM Extended Architecture*), specificat în documentul “*Extended Yellow Book*”.

Standardele pentru principalele discuri compact inscriptibile și reinscriptibile au fost publicate în documentul “*Orange Book*”. Astfel, partea I a acestui document descrie discul magneto-optic CD-MO (*CD-Magneto Optical*), care nu s-a impus însă pe scară largă. În anul 1990, au fost publicate specificațiile discului compact inscriptibil CD-R (*CD-Recordable*), în partea a II-a a documentului “*Orange Book*”. În anul 1995, în partea a III-a a aceluiași document, au fost publicate specificațiile pentru discul reinscriptibil CD-RW (*CD-Read/Write*), numit și CD-E (*CD-Erasable*).

Pe baza formatului CD-ROM/XA, firmele *Philips* și *Kodak* au elaborat în anul 1992 specificațiile discului *Photo CD*, primul format care a implementat înregistrarea multi-sesiune.

Specificațiile au fost publicate în standardul “*Beige Book*”. Un alt format, cel al discului *Video CD*, a fost elaborat de firmele *JVC* și *Philips*. Standardul pentru acest format a fost publicat în anul 1993, în documentul “*White Book*”. În anul 1996 a fost aprobat standardul “*Blue Book*” referitor la formatul CD EXTRA, care este o extensie a formatului pentru discurile compact audio. CD EXTRA este un disc mixt multi-sesiune, care conține o parte audio, conform specificațiilor “*Red Book*”, urmată de o parte CD-ROM care conține date, conform specificațiilor “*Yellow Book*”. Formatul acestui disc a fost elaborat de firmele *Sony* și *Philips*.

2.2. Suportul fizic al discurilor compact

Un disc compact are un diametru de 12 cm și o grosime de aproximativ 1,2 mm. Este format dintr-un substrat de policarbonat transparent, datele fiind înregistrate sub forma unor *cavități* (“*pits*”) microscopice de mărimi variabile, plasate între spații plate, numite *suprafețe* (“*lands*”). Întregul disc este acoperit cu un strat de aluminiu reflectorizant, peste care se suprapune un strat protector de plastic (Figura 10.1).

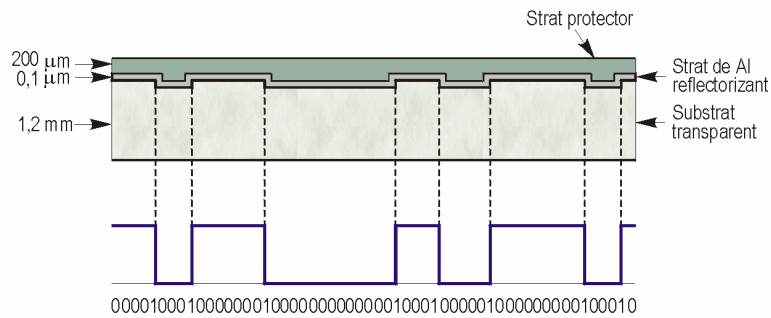


Figura 10.1. Diferitele straturi ale unui disc compact și semnalul obținut la citire.

Citirea informațiilor de pe disc se bazează pe gradul diferit în care este reflectată o rază laser de către cavități și suprafețe. Suprafețele au un grad de reflexie mai mare decât cavitățile. Raza reflectată este detectată de o celulă fotoelectrică, care va genera o succesiune de impulsuri electrice.

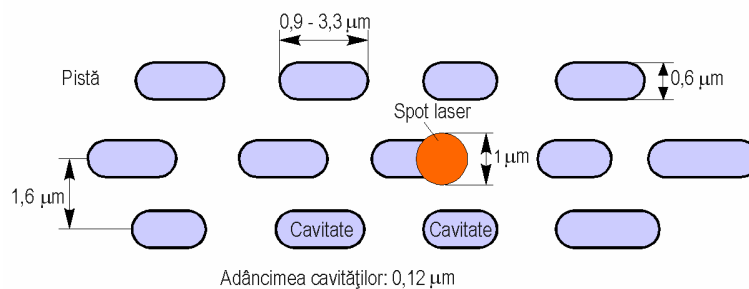


Figura 10.2. Dimensiunile cavităților și distanța între pistele unui disc compact.

Datele sunt înregistrate pe o singură spirală, începând de la centrul discului. Distanța între două piste spiralete alăturate este de $1,6 \mu\text{m}$, ceea ce corespunde unei densități de 16.000 piste/inch (*tracks per inch - tpi*). Fiecare bit de informație necesită o lungime de 300 nm pe pista spiralată.

Cavitățile au adâncimea de $0,12 \mu\text{m}$ și lățimea de $0,6 \mu\text{m}$ (Figura 10.2). Lungimea cavităților și a suprafețelor este cuprinsă între $0,9 \mu\text{m}$ și $3,3 \mu\text{m}$.

2.3. Organizarea și codificarea datelor pe discurile compact

Există mai multe nivele de organizare a datelor pe discurile compact, fiecare nivel fiind construit pe nivelul precedent.

- La nivelul cel mai scăzut, datele sunt înregistrate pe disc sub formă de *cavități și suprafețe*. Ele sunt codificate prin diferite metode astfel încât să se asigure o densitate de înregistrare ridicată și recuperarea sigură a erorilor.
- La următorul nivel, datele sunt organizate în sectoare și piste.
- Specificațiile *High Sierra*, care au fost adoptate ulterior ca standardul ISO 9660, definesc un sistem de fișiere bazat pe sectoarele și pistele discului.

2.3.1. Înregistrarea și codificarea datelor pe suport

La nivelul cel mai scăzut, datele sunt codificate astfel încât să se optimizeze procesul de conversie analog-digitală. Aspectele care s-au avut în vedere la codificarea datelor la acest nivel sunt următoarele:

1. *Densitatea ridicată a informațiilor.* Aceasta necesită o codificare prin care să se utilizeze în mod optim rezoluția ridicată, dar limitată, a razei laser și a ansamblului optic de citire.

2. *Interferența minimă între simboluri.* Aceasta necesită ca numărul minim de biți consecutivi de 0 sau de 1 să fie cât mai mare posibil.

3. *Evitarea unei piste separate pentru sincronizare.* Pentru aceasta, datele trebuie codificate astfel încât semnalul de ceas să poată fi regenerat din semnalul de date. Este necesară deci limitarea numărului de biți consecutivi de 0 sau de 1, astfel încât tranzițiile de date să regenereze semnalul de ceas.

O codificare simplă ar fi reprezentarea biților de 0 prin suprafețe și a biților de 1 prin cavități.

O astfel de codificare nu respectă însă cerința (1), dar nici cerințele (2) și (3). De exemplu, valoarea

întreagă 0 exprimată prin 32 de biți de 0 ar avea un număr prea mare de biți consecutivi de 0 și nu ar respecta cerința (3). Motivul acestei cerințe este de natură tehnologică. Dacă, de exemplu, pe o porțiune există un șir lung de biți cu valoarea 0 (sau 1), unitatea de citire ar trebui să dispună de un ceas extrem de precis, care să se declanșeze exact la intervalul corespunzător parcurgerii celor 300 nm, pentru ca bitul următor să fie citit corect. Acest lucru nu este posibil în stadiul actual al tehnologiei. Din acest motiv, s-a limitat lungimea cavitațiilor și a suprafețelor. Prin această limitare, trecerea de la o stare la alta se va realiza suficient de frecvent pentru ca precizia ceasului să fie suficientă. Ceasul se utilizează la numărarea biților reprezentați într-o cavitate sau pe o suprafață, în funcție de lungimea ocupată.

Biții sunt înregistrați pe suport după metoda RLL (*Run Length Limited*), metodă utilizată și la discurile fixe. Fiecare tranziție de la o cavitate la o suprafață (sau invers) corespunde cifrei binare 1, și fiecare lipsă de tranziție corespunde cifrei binare 0. Atât cavitațiile cât și suprafețele conțin deci cifre de 0. Lungimea minimă a acestora este de 3 biți, iar lungimea maximă este de 11 biți. De aici rezultă că:

- Nu se pot înregistra doi biți succesivi cu valoarea 1.
- Între doi biți cu valoarea 1 se vor găsi cel puțin 2 și cel mult 10 biți cu valoarea 0.

Deoarece aceste restricții nu pot fi respectate pentru un octet oarecare (nu se pot reprezenta toate combinațiile posibile), se utilizează o codificare prin care un octet de date este reprezentat prin 14 *biți de canal*. Există un număr mult mai mare de combinații de 14 biți (16.384) față de combinații de 8 biți (256). Pentru codificarea combinațiilor de 8 biți, s-au ales 256 de combinații de 14 biți care respectă cerințele enunțate. Metoda se numește *modulare 8 la 14* (*Eight to Fourteen Modulation - EFM*). Se utilizează o tabelă pentru a realiza corespondența între cele 256 de combinații valide pentru un octet și secvențele de 14 biți. Pentru valorile zecimale, această corespondență este prezentată în Tabelul 10.2.

Tabelul 10.2. Exemple de corespondențe între valori de 8 biți și secvențele de 14 biți pentru modularea EFM.

Zecimal	8 biți	14 biți
0	0000 0000	01 0010 0010 0000
1	0000 0001	10 0001 0000 0000
2	0000 0010	10 0100 0010 0000
3	0000 0011	10 0010 0010 0000
4	0000 0100	01 0001 0000 0000
5	0000 0101	00 0001 0001 0000
6	0000 0110	00 0100 0010 0000
7	0000 0111	00 1001 0000 0000
8	0000 1000	01 0010 0100 0000
9	0000 1001	10 0000 0100 0000
10	0000 1010	10 0100 0100 0000

Pentru ca între două secvențe de cod EFM alăturate să nu existe doi biți de 1 succesivi, după fiecare secvență de 14 biți sunt inserați 3 biți suplimentari, care nu sunt luați în considerare la prelucrarea datelor. Pentru reprezentarea unui octet de date sunt necesari deci 17 biți de canal: 14 biți EFM și 3 biți suplimentari.

Pentru evitarea unei piste separate de ceas, este necesară o sincronizare periodică. De aceea, datele sunt divizate în blocuri numite *cadre*, la începutul fiecărui cadru existând un antet desincronizare. Cadrul este deci unitatea de bază de sincronizare. Structura unui cadru este prezentată în Figura 10.3.

Sincronizare	Control	Date (Eșantioane L)	CIRC	Date (Eșantioane R)	CIRC
27 biți	1 octet	12 octeți	4 octeți	12 octeți	4 octeți

Figura 10.3. Structura unui cadru al discurilor compact.

Cu excepția biților de sincronizare, toate celelalte câmpuri dintr-un cadru sunt codificate după metoda EFM. Un cadru conține:

- Un antet de sincronizare: 27 biți;

- Un octet de control: $14 + 3 = 17$ biți;
- De două ori câte 12 de octeți de date: $2 \times 12 \times (14 + 3) = 408$ biți;
- De două ori câte 4 octeți pentru detecția și corecția erorilor (CIRC - *Cross Interleaved Reed-Solomon Code*): $2 \times 4 \times (14 + 3) = 136$ biți.

În total, pentru reprezentarea celor 24 octeți de date dintr-un cadru (192 biți) se utilizează 588 biți de canal.

În cazul discurilor audio, fiecare cadru conține câte 6 eșantioane digitizate de 16 biți pentru fiecare canal audio. Ele sunt indicate în Figura 10.3 prin *Eșantioane L* și *Eșantioane R*. Aceste canale audio sunt de obicei componentele din stânga și din dreapta ale unei perechi stereo. Fiecare cadru necesită pentru redare un timp de aproximativ 136,05 μ s. Rezultă astfel o frecvență de eșantionare de 44,1 KHz pentru fiecare canal.

2.4. Prezentare generală a interfeței ATAPI

Scopul realizării interfeței ATAPI (*ATA Packet Interface*) a fost de a se pune la dispoziție o interfață mai extensibilă și mai generală decât interfața ATA. Echipamentele ATAPI trebuie să permită modurile de operare *master* și *slave*. Aceste echipamente utilizează aceleași semnale care sunt specificate în standardul interfeței ATA.

Deși pentru conectarea unei unități de discuri optice sau a unei unități de bandă prin interfața ATAPI se utilizează aceeași interfață fizică ATA, interfața logică diferă într-o oarecare măsură și are posibilități suplimentare. Echipamentele conectate prin interfața ATA utilizează pentru comunicația cu calculatorul opt registre, care conțin comanda și toți parametrii necesari pentru o operație. Aceste registre nu sunt însă suficiente pentru transmiterea tuturor informațiilor necesare controlului diferitelor tipuri de periferice. Pentru a elimina acest dezavantaj, comenzile pentru echipamentele ATAPI sunt transmise sub forma unei structuri numite *pachet de comandă* (*Command Packet*). Aceste comenzi completează comenzile ATA existente.

A fost adăugată o nouă comandă, numită *Packet*, care permite transmiterea unui pachet de comandă la un echipament ATAPI. Pachetul este transmis prin scrieri repetate în registrul de date.

Această tehnică reduce numărul adreselor necesare pentru registre. Deși toate comenzile pentru echipamentele ATAPI pot fi transmise în acest mod, aceste echipamente trebuie să accepte și unele din comenzile ATA existente, ca și principalele protocoale de comenzi ATA, pentru ca modificările driverelor existente să fie minime. Acest set minimal de comenzi ATA este mai redus decât cel minimal definit pentru interfața ATA, dar este suficient pentru funcționarea normală.

Pe de altă parte, echipamentele ATAPI nu vor răspunde la comenzile *Identify Device* sau *Read* ale interfeței ATA, ceea ce permite ca sistemul BIOS și unele drivere mai vechi să nu recunoască existența acestor echipamente și să nu confunde datele de la acestea cu cele de la unitățile de discuri magnetice. Comenzile care nu sunt recunoscute nu vor fi executate, generându-se o întrerupere pentru a indica terminarea comenzii respective cu o stare de eroare (*ABORTED*).

Pentru identificarea unui echipament ATAPI s-a prevăzut comanda *Identify Packet Device*, informațiile returnate de această comandă fiind utilizate de driverele de nivel inferior pentru configurarea hardware a interfeței ATA. Aceste informații au o structură similară cu cea a informațiilor returnate de comanda *Identify Device* a interfeței ATA, din motive de compatibilitate. Deoarece informațiile returnate de comanda *Inquiry* a interfeței SCSI nu pot fi returnate de comanda *Identify Packet Device* a interfeței ATAPI, comanda *Inquiry* este recunoscută de interfața ATAPI pentru a fi utilizată de driverele de nivel mai înalt.

Deși interfața ATAPI utilizează structuri ale pachetelor care sunt preluate din standardele SCSI, cele mai multe facilități ale interfeței SCSI nu sunt utilizate. Diferențele principale față de interfața SCSI sunt următoarele:

- Perifericul ATAPI are rol de *slave* în timpul unei operații, spre deosebire de rolul de *master* al unui periferic SCSI.
- Nu există faze de funcționare și nu sunt transmise mesaje.
- Nu sunt posibile operațiile de deconectare / reconectare.
- Pachetele de comenzi au o lungime de 12 octeți, spre deosebire de interfața SCSI, unde lungimea pachetelor poate fi de 6, 10 sau 12 octeți. Sunt definite și pachete de 16 octeți pentru compatibilitate cu echipamentele viitoare. Dimensiunea pachetelor acceptate de un

echipament este definită în cuvântul 0 al informațiilor returnate la comanda *Identify Packet Device*.

Pentru configurarea echipamentelor ATAPI, sunt utilizate comenzile *Mode Select* și *Mode Sense* din standardele SCSI. Combinația comenzii SCSI *Mode Select* și a comenzii ATA *Set Features* asigură realizarea funcțiilor necesare și este compatibilă cu cele mai multe sisteme BIOS existente și drivere ale sistemelor de operare.

2.5. Registrele interfeței ATAPI

Comunicația cu dispozitivele periferice se realizează prin registrele de I/E selectate de sistem prin semnalele $-CS0$, $-CS1$, $DA2$, $DA1$, $DA0$, $-IOR$ și $-IOW$. Cu excepția registrului de date, care este de 16 biți, toate celelalte registre ale interfeței ATAPI sunt accesate prin operații de intrare/ieșire pe 8 biți. În Tabelul 10.4 se prezintă funcțiile registrelor interfeței ATAPI și adresele lor pentru interfața ATA/ATAPI primară. Pentru interfața ATA/ATAPI secundară, adresele sunt 170h..177h și 376h.

Tabelul 10.4. Registrele interfeței ATAPI și adresele acestora pentru interfața ATA/ATAPI primară.

Selecție					Adresa	Funcție	
-CS0	-CS1	DA2	DA1	DA0		Citare (-IOR)	Scriere (-IOW)
Registre bloc de comandă							
0	1	0	0	0	1F0h	Date (16 biți)	
0	1	0	0	1	1F1h	Eroare	Setare caracteristici
0	1	0	1	0	1F2h	Identificare întrerupere	Neutilizat
0	1	0	1	1	1F3h	Rezervat	
0	1	1	0	0	1F4h	Contor de octeți (biți 7..0)	
0	1	1	0	1	1F5h	Contor de octeți (biți 15..8)	
0	1	1	1	0	1F6h	Selecție dispozitiv	
0	1	1	1	1	1F7h	Stare ATAPI	Comandă ATA
Registre bloc de control							
1	0	1	1	0	3F6h	Stare alternativă ATAPI	Control dispozitiv

2.5.1. Registrul de stare ATAPI

Acest registru corespunde registrului de stare al interfeței ATA.

7 6 5 4 3 2 1 0
BSY DRDY DMRD SERV DRQ X X CHK

- Bitul 7 (BSY – *Busy*) este setat la 1 ori de câte ori echipamentul are acces la registrele blocului de comandă. Dacă bitul BSY este 1, o scriere într-un registru al blocului de comandă de către sistem va fi ignorată de echipament.
- Bitul 6 (DRDY – *Device Ready*) indică prin valoarea 1 faptul că echipamentul poate răspunde la o comandă.
- Bitul 5 (DMRD – *DMA Ready*) indică prin valoarea 1 faptul că echipamentul este gata pentru a lansa un transfer de date în modul DMA sau Ultra DMA în cazul execuției suprapuse a transferurilor DMA. Dacă execuția suprapusă a transferurilor DMA nu este validată, acest bit este utilizat ca bit de eroare a unității DF (*Device Fault*).
- Bitul 4 (SERV – *Service*) indică prin valoarea 1 solicitarea de către echipament a unui serviciu sau a unei întreruperi. Dacă un echipament permite suprapunerea comenzilor, la recepția unui pachet pentru o comandă a cărei execuție poate fi suprapusă, echipamentul eliberează magistrala ATA. Aceasta permite calculatorului să transmită o comandă celui alt echipament conectat la magistrală. Atunci când primul echipament poate continua comanda, acesta setează bitul SERV la 1 și, dacă bitul nIEN este 0, activează semnalul de întrerupere *INTRQ*. Calculatorul va transmite apoi comanda *Service* pentru a continua execuția comenzii. Dacă execuția suprapusă a comenzilor nu este validată, acest bit este utilizat pentru funcția DSC (*Device Seek Complete*), indicând poziționarea capului de citire deasupra unei piste.
- Bitul 3 (DRQ – *Data Request*) indică faptul că echipamentul este gata pentru transferul unui cuvânt de date cu calculatorul. Informațiile din registrul de identificare a întreruperii sunt valide în timpul unei comenzi atunci când bitul DRQ este setat la 1.
- Biții 2 și 1 sunt nedefiniți.

- Bitul 0 (CHK – *Check Condition*) indică apariția unei erori în timpul execuției comenzii precedente. Biții din registrul de eroare conțin codul erorii.

2.5.2. Registrul de eroare

Acest registru corespunde registrului de eroare al interfeței ATA.

7 6 5 4 3 2 1 0
Cod de eroare X ABRT EOM ILI

- Biții 7..4 conțin un cod de eroare (*Sense Key*) care indică natura erorii apărute.
- Bitul 3 este nedefinit.
- Bitul 2 (ABRT – *Aborted Command*) are semnificația definită în standardul ATA, indicând prin valoarea 1 abandonarea comenzii cerute deoarece codul comenzii este invalid sau a apărut o eroare de dispozitiv.
- Bitul 1 (EOM – *End Of Media*) indică prin valoarea 1 faptul că s-a detectat sfârșitul suportului. Acest bit poate avea o semnificație diferită pentru unele comenzi.
- Bitul 0 (ILI – *Illegal Length Indication*) indică prin valoarea 1 o lungime incorectă. Acest bit poate avea o semnificație diferită pentru unele comenzi.

2.5.3. Registrul pentru setarea caracteristicilor

Acest registru corespunde registrului pentru setarea caracteristicilor al interfeței ATA.

7 6 5 4 3 2 1 0
X X X X X X OVL DMA

- Biții 7..2 sunt nedefiniți.
- Bitul 1 (OVL – *Overlap*) trebuie setat la 1 pentru a indica echipamentului execuția suprapusă a comenzii. În acest caz, echipamentul poate elibera magistrala ATA înaintea terminării comenzii. Eliberarea magistralei rămâne la discreția echipamentului.
- Bitul 0 (DMA) trebuie setat la 1 pentru a indica echipamentului faptul că transferul datelor asociat comenzii trebuie realizat în modul DMA multicuvânt sau Ultra DMA. Pachetul de comandă este transferat întotdeauna în modul PIO.

2.5.4. Registrele pentru contorul de octeți

Registrul pentru biții 7..0 ai contorului de octeți corespunde registrului pentru adresa LBA mijlocie, iar registrul pentru biții 15..8 ai contorului de octeți corespunde registrului pentru adresa LBA superioară al interfeței ATA.

7 6 5 4 3 2 1 0

Contor de octeți (biți 7..0)

Contor de octeți (biți 15..8)

Contorul de octeți este utilizat numai pentru transferurile în modul PIO. Contorul trebuie setat înaintea transmiterii comenzii *Packet*. Contorul indică dimensiunea maximă a transferului pentru comenzile care transferă un singur bloc de date (de exemplu, *Mode Select*, *Inquiry*). Pentru comenzile care necesită întreruperi DRQ multiple (de exemplu, *Read*), contorul este setat la dimensiunea dorită.

Echipamentul ATAPI va seta contorul de octeți la numărul de octeți de date pe care calculatorul trebuie să le transfere, iar apoi va genera întreruperea DRQ. Conținutul acestor registre nu trebuie modificat până când nu s-a transferat primul cuvânt. După începerea transferului, conținutul poate fi modificat pentru următorul transfer.

2.5.5. Registrul de identificare a întreruperii

Acest registru corespunde registrului contor de sectoare al interfeței ATA.

7 6 5 4 3 2 1 0

Marcaj REL I/O C/D

- Biții 7..3 conțin marcajul comenzii în cazul în care echipamentul implementează o coadă de comenzi. Marcajul identifică în mod unic comanda și poate avea o valoare cuprinsă între 0 și 31. Coada de comenzi permite calculatorului să transmită comenzi concurente la același echipament. Echipamentul va depune în coada de comenzi doar acele comenzi a căror execuție poate fi suprapusă, exemple de asemenea comenzi fiind *Packet*, *Read DMA Queued*, *Service* sau *Write DMA Queued*. Atunci când echipamentul reface parametrii din registre în urma execuției comenzii *Service*, reface și valoarea marcajului, astfel încât calculatorul poate identifica comanda la care se referă starea registrelor.

- Bitul 2 (REL – *Release*) indică eliberarea magistralei ATA de către echipament înaintea terminării comenzii în curs.
- Bitul 1 (I/O – *Input / Output*) specifică sensul transferului, valoarea 0 indicând o operație de ieșire (la periferic), iar valoarea 1 o operație de intrare (la calculator).
- Bitul 0 (C/D – *Command / Data*) indică tipul informațiilor transferate. Dacă acest bit este 0, informațiile transferate reprezintă date, iar dacă bitul este 1, informațiile transferate reprezintă comenzi.

2.5.6. Registrul de selecție al dispozitivului

Acest registru corespunde registrului de selecție al dispozitivului de la interfața ATA.

```
7 6 5 4 3 2 1 0
X X X DEV X X X X
```

- Biții 7..5 sunt nedefiniți.
- Bitul 4 (DEV – *Device Select*) selectează unitatea 0 (DEV = 0) sau unitatea 1 (DEV = 1).
- Biții 3..0 sunt nedefiniți.

2.5.7. Registrul de control al dispozitivului

Acest registru corespunde registrului de control al dispozitivului de la interfața ATA.

```
7 6 5 4 3 2 1 0
X X X X X SRST nIEN 0
```

- Biții 7..3 sunt rezervați.
- Bitul 2 (SRST – *Software Reset*) permite resetarea software a echipamentului, printr-un protocol specific pentru interfețele ATA și ATAPI. Există în plus și posibilitatea utilizării comenzii ATA *Device Reset*.
- Bitul 1 (nIEN – *INTRQ Enable*) validează prin valoarea 0 generarea întreruperilor către calculator.
- Bitul 0 trebuie să fie setat la 0.

2.6. Execuția comenzilor ATAPI

Comanda unui echipament ATAPI se poate realiza prin două metode, prin comenzile ATA obișnuite sau prin comanda ATAPI *Packet*. Pentru ambele metode, echipamentele care sunt conectate la interfață sunt programate de calculator pentru execuția comenzilor, iar aceste echipamente vor returna starea la terminarea comenzilor. Dacă sunt conectate în lanț două echipamente la interfață, comenzile sunt transmise în paralel ambelor echipamente, dar numai cel selectat prin bitul DEV din registrul de selecție al dispozitivului va executa comanda.

Pentru transmiterea comenzii *Packet*, se aplică regulile și protocolul obișnuit al interfeței ATA, dar după transmiterea comenzii se aplică un nou set de reguli:

1. Trebuie să se interpreteze bitul DRQ din registrul de stare împreună cu biții I/O și C/D din registrul de identificare a întreruperii pentru a determina tipul întreruperii.
2. Comanda care va fi executată de echipament va fi transmisă ca un pachet prin registrul de date și nu prin celelalte registre ale interfeței.
3. Parametrii comenzii sunt transmiși prin pachetul de comandă și prin registrele interfeței.
4. Se utilizează un contor de octeți pentru a determina cantitatea datelor care vor fi transferate de calculator la fiecare întrerupere DRQ.
5. Pentru a indica utilizarea modului de transfer prin DMA se utilizează registrul pentru setarea caracteristicilor în locul utilizării diferitelor coduri de operație.
6. Starea finală este returnată calculatorului ca o nouă întrerupere după transferul ultimei date și nu împreună cu ultimul bloc de date.

Aceste reguli se aplică până la citirea de către calculator a stării care indică terminarea operației. După aceasta, definițiile registrelor și protocolul utilizat vor fi cele din standardul ATA.

Comanda *Packet* este transmisă ca orice comandă normală ATA, prin inițializarea registrelor interfeței, setarea bitului de selecție al unității și scrierea octetului de comandă în registrul de comandă. La comenzile ATA normale, este setat bitul DRQ din registrul de stare (și, în mod opțional, se generează o întrerupere), pentru a indica faptul că parametrii comenzii pot fi transferați la / de la echipament. La comanda *Packet*, prima setare a bitului DRQ indică faptul că datele pachetului de comandă trebuie transmise echipamentului. După transmiterea pachetului, comanda continuă ca o comandă ATA normală.

Dacă în timpul testării bitului BSY echipamentul rămâne într-o stare în care nu poate accepta o comandă pentru mai mult de 5 secunde, calculatorul trebuie să reseteze echipamentul.

Deoarece lungimea datelor transferate în modul PIO la și de la un echipament ATAPI este controlat de calculator, s-a creat posibilitatea transferului unui număr variabil de octeți prin utilizarea registrelor pentru contorul de octeți. Echipamentul indică numărul de octeți care trebuie transferați la fiecare întrerupere DRQ. Înaintea transferului de date, calculatorul trebuie să citească registrele pentru contorul de octeți și să țină cont de lungimea cerută. Echipamentul ATAPI și calculatorul vor avea contoare proprii de octeți și vor transfera date până când aceste contoare ajung la zero. Pentru anumite comenzi, ca *Mode Sense*, calculatorul nu cunoaște cantitatea datelor care vor fi transferate și trebuie să se bazeze pe contorul de octeți furnizat de echipament pentru a transfera numărul corect de octeți.

O altă utilizare a registrelor pentru contorul de octeți constă în indicarea de calculator a numărului maxim de octeți pe care le poate prelua într-un singur pachet DRQ sau a dimensiunii preferate a pachetului. Pentru toate comenzile care necesită transferuri de date, calculatorul trebuie să încarce registrele pentru contorul de octeți cu lungimea dorită înaintea transmiterii pachetului de comandă. Această lungime este utilizată de echipamentul ATAPI ca dimensiune maximă a fiecărui pachet de date PIO sau DMA. Echipamentul poate transfera pachete de dimensiune mai mică decât cea indicată de calculator.

Dacă echipamentul solicită transferul unui număr mai mare de octeți decât cel cerut de protocolul comenzii, calculatorul trebuie să completeze datele transmise echipamentului și să preia datele suplimentare la citirea acestora de la echipament. Echipamentul nu trebuie să realizeze completarea datelor, transferând numai numărul de octeți specificat de calculator.

Anumite comenzi ATAPI sunt imediate. Pentru o asemenea comandă, echipamentul returnează starea de terminare a execuției imediat, continuând execuția comenzii. Trebuie să se țină cont de următoarele:

- Dacă se transmite o nouă comandă ATAPI după o comandă imediată care raportează terminarea înaintea terminării efective (*Seek, Play Audio* etc.), noua comandă este memorată de echipament.
- Dacă se înscrie în registrul de comandă o nouă comandă ATA înaintea terminării unei comenzi, comanda în curs va fi oprită din execuție, iar noua comandă va fi abandonată, setându-se bitul ABRT (*Aborted Command*) din registrul de eroare.
- Noile comenzi ATAPI recepționate în timpul execuției unei comenzi ATAPI precedente determină abandonarea ambelor comenzi și setarea bitului CHK (*Check Condition*) din registrul de stare. Există echipamente care permit suprapunerea comenzilor, îmbunătățind performanțele sistemului. Aceste echipamente eliberează magistrala ATA înaintea terminării comenzii în curs, permițând utilizarea magistralei de celălalt echipament.

2.7. Protocoale utilizate pentru comenzile ATAPI

2.7.1. Protocolul pentru comenzile fără transferuri de date

Acest protocol este utilizat pentru comenzi ca *Seek, Play Audio* sau *Start/Stop Unit*, a căror execuție nu implică transferuri de date. Din punctul de vedere al calculatorului, protocolul utilizat este următorul:

1. Calculatorul așteaptă până când biții BSY și DRQ din registrul de stare devin egali cu 0.
2. Calculatorul inițializează registrul de selecție al dispozitivului, registrul de control al dispozitivului și registrul pentru setarea caracteristicilor. În registrul de selecție al dispozitivului, bitul DEV se resetează la 0 pentru accesul la unitatea 0 sau se setează la 1 pentru accesul la unitatea 1. În registrul de control al dispozitivului, bitul nIEN se resetează la 0 dacă generarea întreruperilor de către echipament trebuie validată sau se setează la 1 dacă generarea întreruperilor trebuie invalidată.
3. Calculatorul scrie codul comenzii *Packet* (A0h) în registrul de comandă.
4. Calculatorul așteaptă un timp de minim 400 ns. De exemplu, se poate citi registrul de stare alternativă, ignorând rezultatul citirii.
5. Calculatorul citește registrul de stare și așteaptă până când bitul BSY devine 0.
6. Calculatorul citește registrul de stare și așteaptă până când bitul DRQ devine 1.
7. Calculatorul scrie cuvintele comenzii în registrul de date, câte un cuvânt la un moment dat.

8. Dacă generarea întreruperilor de către echipament a fost validată ($nIEN = 0$), calculatorul așteaptă până când semnalul *INTRQ* devine activ, după care continuă cu pasul 9. Dacă generarea întreruperilor de către echipament nu a fost validată ($nIEN = 1$), calculatorul continuă cu pasul 9.

9. Calculatorul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, se poate citi registrul de stare alternativă, ignorând rezultatul citirii.

10. Calculatorul citește registrul de stare și așteaptă până când bitul BSY devine 0.

11. Calculatorul citește registrul de stare și așteaptă până când bitul DRQ devine 0, moment în care execuția comenzii este terminată.

2.7.2. Protocolul pentru intrare în modul PIO

Acest protocol este utilizat de comenzi ca *Inquiry* sau *Read*. Execuția cuprinde transferul unui număr necunoscut de octeți de la echipament la calculator. Din punctul de vedere al calculatorului, protocolul utilizat este următorul:

1. Calculatorul așteaptă până când biții BSY și DRQ din registrul de stare devin egali cu 0.

2. Calculatorul inițializează registrul de selecție al dispozitivului, registrul de control al dispozitivului, registrul pentru setarea caracteristicilor și registrele pentru contorul de octeți. În registrul de selecție al dispozitivului, bitul DEV se resetează la 0 pentru accesul la unitatea 0 sau se setează la 1 pentru accesul la unitatea 1. În registrul de control al dispozitivului, bitul $nIEN$ se resetează la 0 dacă generarea întreruperilor de către echipament trebuie validată sau se setează la 1 dacă generarea întreruperilor trebuie invalidată.

3. Calculatorul înscrie codul comenzii *Packet* (A0h) în registrul de comandă.

4. Calculatorul așteaptă un timp de minim 400 ns. De exemplu, se poate citi registrul de stare alternativă, ignorând rezultatul citirii.

5. Calculatorul citește registrul de stare și așteaptă până când bitul BSY devine 0.

6. Calculatorul citește registrul de stare și așteaptă până când bitul DRQ devine 1.

7. Calculatorul scrie cuvintele comenzii în registrul de date, câte un cuvânt la un moment dat.

8. Dacă generarea întreruperilor de către echipament a fost validată ($nIEN = 0$), calculatorul așteaptă până când semnalul *INTRQ* devine activ, după care continuă cu pasul 9. Dacă generarea întreruperilor de către echipament nu a fost validată ($nIEN = 1$), calculatorul continuă cu pasul 9.

9. Calculatorul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, se poate citi registrul de stare alternativă, ignorând rezultatul citirii.

10. Calculatorul citește registrul de stare și așteaptă până când bitul BSY devine 0.

11. Calculatorul citește registrul de stare și așteaptă până când bitul DRQ devine 1.

12. Calculatorul transferă un bloc de date, citind câte un cuvânt din registrul de date. Numărul de octeți care trebuie transferați este specificat de registrele pentru contorul de octeți.

13. Calculatorul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, se poate citi registrul de stare alternativă, ignorând rezultatul citirii.

14. Calculatorul citește registrul de stare și așteaptă până când bitul BSY devine 0.

15. Calculatorul citește registrul de stare și dacă bitul DRQ este 1, continuă cu transferul unui nou bloc de date (pasul 12). Dacă bitul DRQ este 0, execuția comenzii este terminată.

2.8. Comenzi ATAPI

2.8.1. Identificarea echipamentelor ATAPI

La punerea sub tensiune a echipamentelor ATAPI, la resetarea hardware sau software a acestora, sau la execuția comenzilor *ATA Device Reset* și *Execute Device Diagnostic*, aceste echipamente încarcă în unele registre ale interfeței ATA o semnătură care permite identificarea echipamentelor ATAPI. De asemenea, comanda *ATA Identify Device* nu va fi executată de echipamentele ATAPI, care vor seta bitul ABRT din registrul de eroare al interfeței și vor încărca o parte din registrele interfeței cu aceeași semnătură ca și la execuția comenzii *Device Reset*.

Semnătura care identifică echipamentele ATAPI și registrele interfeței ATA în care este plasată această semnătură sunt indicate în Tabelul 10.5.

Tabelul 10.5. Semnătura caracteristică echipamentelor ATAPI.

Registru ATA	Valoare
Contor de sectoare	01h
Adresă LBA inferioară	01h
Adresă LBA mijlocie	14h
Adresă LBA superioară	EBh

Pentru a detecta dacă un anumit echipament conectat la o interfață ATA este un echipament ATAPI, se poate transmite comanda *Identify Device* echipamentului respectiv, urmărind dacă această comandă este recunoscută de echipament. Dacă execuția comenzii este abandonată cu starea de eroare *ABORTED*, indicată prin setarea bitului ABRT din registrul de eroare al interfeței, se citește conținutul registrelor pentru adresa LBA mijlocie și adresa LBA superioară. Dacă aceste registre conțin valorile 14h, respectiv EBh, echipamentul respectiv este un echipament ATAPI și recunoaște comanda *Packet*. În continuare, se poate transmite comanda *Identify Packet Device* (cu codul A1h) pentru a afla informații detaliate despre echipamentul ATAPI, utilizând același protocol ca și cel pentru execuția comenzii *Identify Device*.

Pentru execuția comenzii *Identify Device*, se poate utiliza protocolul descris în Lucrarea de laborator Nr. 7, secțiunea 2.9.4. Pentru mai multă claritate, se prezintă mai jos operațiile necesare pentru transmiterea și execuția acestei comenzi, completate cu operațiile necesare pentru detectarea abandonării comenzii în cazul unui echipament ATAPI.

1. Se așteaptă până când biții BSY și DRQ din registrul de stare devin egali cu 0.
2. Se setează bitul DEV din registrul de selecție al dispozitivului în funcție de unitatea căreia i se transmite comanda (DEV = 0 pentru unitatea 0, DEV = 1 pentru unitatea 1). Se setează la 1 bitul nIEN din registrul de control al dispozitivului pentru a invalida generarea întreruperilor de către unitate.
3. Se înscrie codul comenzii *Identify Device* (ECh) în registrul de comandă.
4. Se citește registrul de stare și se așteaptă până când bitul BSY devine 0.
5. Se citește registrul de stare și se testează bitul DRQ. Dacă acest bit este 1, se continuă cu pasul 6. Dacă bitul DRQ este 0, comanda nu a fost recunoscută de unitate. În acest caz, se testează dacă bitul ABRT din registrul de eroare este 1. În caz afirmativ, se citește conținutul registrelor pentru adresa LBA mijlocie și adresa LBA superioară. Dacă aceste registre conțin valorile 14h, respectiv EBh, unitatea este o unitate ATAPI și secvența de operații este terminată.
6. Se așteaptă până când bitul BSY din registrul de stare devine 0.
7. Se așteaptă până când bitul DRQ din registrul de stare devine 1.
8. Se citește un cuvânt din registrul de date și se memorează.
9. Dacă nu s-au citit toate cele 256 de cuvinte, se continuă cu operația de la punctul 6. În caz contrar, secvența de operații este terminată.

2.8.2. Descrierea pachetului de comandă

O comandă ATAPI este transmisă unui echipament ATAPI prin intermediul unui pachet. Pentru unele comenzi, pachetul este urmat de o listă de parametri transmiși după generarea unei întreruperi în urma transmiterii pachetului de comandă.

În Figura 10.12 se prezintă structura tipică a unui pachet de comandă de 12 octeți. Câmpurile au aceeași semnificație cu câmpurile blocului descriptor de comenzi de la interfața SCSI (Lucrarea de laborator Nr. 9). Primul octet al pachetului conține întotdeauna un cod de operație. Ultimul octet conține un cod de control. Pentru comenzile obișnuite, acest cod de control trebuie să fie 0.

Bit Octet	7	6	5	4	3	2	1	0	
0	Cod de operație								
1	Rezervat								
2	Bit cms	Adresă bloc logic						Bit cms	
3									
4									
5									
6	Bit cms	Lungime transfer,						Bit cms	
7	Lungime listă de parametri,								
8	Lungime alocată								
9								Bit cms	
10	Rezervat								
11	Control								

Figura 10.12. Structura tipică a unui pachet de comandă ATAPI de 12 octeți.

2.8.3. Lista comenzilor ATAPI pentru unitățile CD

Comenzile ATAPI pentru unitățile CD sunt derivate din setul de comenzi SCSI. Cu excepția tehnicii de adresare MSF, interfața utilizează adresarea logică pentru toate blocurile de date. Fiecare echipament poate fi interogat pentru a determina numărul de blocuri pe care le conține.

În Tabelul 10.6 se prezintă lista comenzilor ATAPI pentru unitățile CD.

Comandă	Cod operație	Tip
BLANK	A1h	E
CLOSE TRACK/SESSION	5Bh	R
FORMAT UNIT	04h	E
INQUIRY	12h	M
LOAD/UNLOAD CD	A6h	C
MECHANISM STATUS	BDh	M
MODE SELECT (10)	55h	M
MODE SENSE (10)	5Ah	M
PAUSE/RESUME	4Bh	A
PLAY AUDIO (10)	45h	A
PLAY AUDIO (12)	A5h	A
PLAY AUDIO MSF	47h	A
PLAY CD	BCh	O
PREVENT/ALLOW MEDIUM REMOVAL	1Eh	M
READ (10)	28h	M
READ (12)	A8h	M
READ BUFFER CAPACITY	5Ch	O
READ CD	BEh	M
READ CD MSF	B9h	M
READ CD RECORDED CAPACITY	25h	M
READ DISC INFORMATION	51h	R
READ HEADER	44h	M
READ MASTER CUE	59h	O
READ SUB-CHANNEL	42h	M
Comandă	Cod operație	Tip
READ TOC/PMA/ATIP	43h	M
READ TRACK INFORMATION	52h	R
REPAIR TRACK	58h	O
REQUEST SENSE	03h	M
RESERVE TRACK	53h	R
SCAN	BAh	A
SEEK	2Bh	M
SEND CUE SHEET	5Dh	O
SEND OPC INFORMATION	54h	O
SET CD SPEED	BBh	R
START/STOP UNIT	1Bh	M
STOP PLAY/SCAN	4Eh	M
SYNCHRONIZE CACHE	35h	R
TEST UNIT READY	00h	M
WRITE (10)	2Ah	R
WRITE (12)	AAh	R
M: Comandă obligatorie O: Comandă opțională A: Comandă obligatorie pentru unități audio R: Comandă obligatorie pentru unități CD-R/RW E: Comandă obligatorie pentru unități CD-RW C: Comandă obligatorie pentru schimbătoare de discuri		

2.8.4. Comanda PLAY AUDIO MSF

Comanda specifică începerea unei operații de redare audio de către unitatea CD conținând un disc CD audio. Aceasta este o comandă imediată, permițând suprapunerea comenzilor. La terminarea efectivă a operației, este setat bitul SERV (DSC) din registrul de stare. Protocolul utilizat este cel pentru comenzile fără transferuri de date.

Structura pachetului de comandă este ilustrată în Figura 10.13.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (47h)							
1..2	Rezervați							
3	Câmp M de început							
4	Câmp S de început							
5	Câmp F de început							
6	Câmp M de sfârșit							
7	Câmp S de sfârșit							
8	Câmp F de sfârșit							
9	Control							

Figura 10.13. Structura pachetului comenzii *Play Audio MSF*.

Câmpurile M, S și F de început specifică adresa MSF absolută de la care va începe operația, iar câmpurile M, S și F de sfârșit specifică adresa MSF absolută la care operația va fi terminată. Dacă valorile câmpurilor M, S și F de început sunt setate la FFh, adresa de început va fi cea corespunzătoare poziției ansamblului optic de citire. Aceasta permite schimbarea adresei de sfârșit fără întreruperea operației curente.

Dacă adresa MSF de început este egală cu adresa MSF de sfârșit, nu se execută nici o operație.

Dacă adresa MSF de început este mai mare decât adresa MSF de sfârșit, comanda se va termina cu starea *CHECK CONDITION*.

2.8.5. Comanda READ (12)

Această comandă solicită unității CD transferul datelor de la unitate la calculator. Structura pachetului de comandă este cea tipică a unei comenzi (Figura 10.12). Octeții 6..9 conțin lungimea transferului. Lungimea transferului specifică numărul blocurilor logice contigue care trebuie transferate. Deși unitatea CD poate returna diferite informații, această comandă va transfera numai porțiunea de date din cadrul sectorului. Acest câmp de date are întotdeauna lungimea de 2048 de octeți pentru sectoarele în Modul 1 și Modul 2, Forma 1, care sunt singurele tipuri de sectoare permise. Pentru alte tipuri de sectoare, va fi setat bitul ILI (*Illegal Length Indication*) din registrul de eroare, dacă se încearcă citirea acestora prin comanda *Read*.

2.8.6. Comanda READ CD RECORDED CAPACITY

Această comandă permite citirea capacității înregistrate a unui disc CD. Comanda returnează adresa ultimului bloc logic, pe baza tabelii de conținut a volumului. Dacă ultima pistă este o pistă audio, valoarea returnată poate fi inexactă, existând o toleranță de ± 75 cadre la adresarea datelor audio, conform specificațiilor suportului. Ultimul bloc poate fi deci la o distanță de ± 75 cadre față de sfârșitul efectiv al pistei. Pentru unitățile CD, această implementare permite un răspuns mai rapid.

Structura pachetului de comandă este ilustrată în Figura 10.14.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (25h)							
1	Rezervat							RELAD
2	Bit cms							
3	Adresă bloc logic							
4								
5	Bit cmpls							
6	Rezervat							
7	Rezervat							
8	Rezervat							PMI
9	Control							

Figura 10.14. Structura pachetului comenzii *Read CD Recorded Capacity*.

Biții RELAD (*Relative Address*) și PMI (*Partial Medium Indicator*) trebuie setați la 0. Adresa blocului logic trebuie setată la 0.

Protocolul utilizat pentru această comandă este cel pentru intrare în modul PIO. Datele returnate (4 cuvinte) au structura din Figura 10.15. Comanda returnează o lungime a blocului de 2048 de octeți.

Bit Octet	7	6	5	4	3	2	1	0
0..3	Adresa blocului logic (octet 0 - c.m.s.)							
4..7	Lungimea blocului în octeți (octet 4 - c.m.s.)							

Figura 10.15. Datele returnate de comanda *Read CD Recorded Capacity*.

3. Desfășurarea lucrării

3.1. Răspundeți la următoarele întrebări:

- Cum sunt înregistrați biții de 0 și de 1 pe suportul fizic al discurilor compact ?
- Ce înseamnă modularea 8 la 14 ?
- Cum se realizează corecția erorilor în interiorul cadrelor ?
- Cum este posibilă citirea simultană a datelor și a informațiilor audio/video la discurile CD-ROM/XA și CD-I ?
- Ce reprezintă subcanalele discurilor compact și la ce se utilizează acestea ?

3.2. Scrieți un program care determină dacă o unitate conectată la o interfață ATA este o unitate ATAPI. Verificați funcționarea programului pentru:

- Unitatea 0, prima interfață ATA (cu adresa de bază 1F0h);
- Unitatea 0, a doua interfață ATA (cu adresa de bază 170h).

3.3. Scrieți un program pentru citirea tabelii de conținut a unui disc CD audio dintr-o unitate ATAPI. Programul va afișa adresa MSF de început a fiecărei piste audio. Introduceți un disc CD audio în unitate și verificați programul, comparând adresele MSF afișate cu cele inscripționate pe disc.

3.4. Scrieți un program care transmite unei unități ATAPI comanda *Play Audio MSF*. Definiți adresele MSF de început și de sfârșit prin valori constante. Introduceți un disc CD audio în unitate și verificați programul, conectând la unitate o cască audio sau difuzoare.

3.5. Modificați programul de la punctul 3.4 astfel încât să permită introducerea de la tastatură a numărului pistei audio pentru care va fi executată comanda *Play Audio MSF*. Programul va determina adresele MSF de început ale pistelor audio prin citirea tabelii de conținut a discului CD audio.

3.6. Scrieți un program pentru citirea capacității înregistrate a unui disc CD-ROM de date. Verificați rezultatul prin comparare cu capacitatea indicată de programul *Total Commander*.