

8. DISCURI COMPACT. INTERFAȚA ATAPI

În prima parte, această lucrare de laborator prezintă suportul fizic al discurilor compact, înregistrarea și codificarea datelor pe suport, nivelele de corecție a erorilor, formatul sectoarelor și organizarea discurilor. În partea a doua, lucrarea prezintă interfața *ATA Packet Interface* (ATAPI), registrele interfeței, protocoalele comenzilor, o listă de comenzi pentru unitățile CD/DVD și exemple de comenzi.

8.1. Suportul fizic al discurilor compact

Un disc compact are un diametru de 12 cm și o grosime de 1,2 mm. Este format dintr-un substrat de polycarbonat 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 subțire de aluminiu reflectorizant, iar apoi cu un strat protector de plastic (figura 8.1).

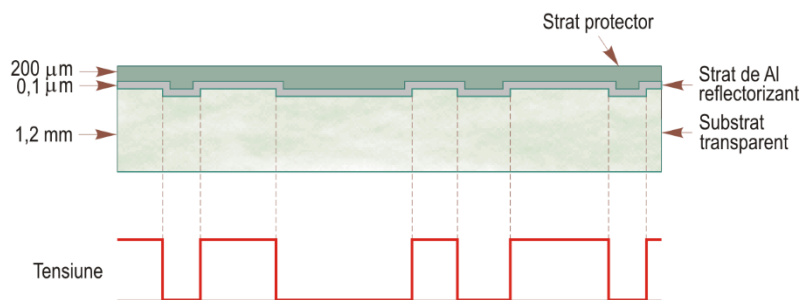


Figura 8.1. Straturile unui disc compact și semnalul generat de celula fotoelectrică.

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

Datele sunt înregistrate pe o singură spirală, începând de la centrul discului. Distanța între două piste consecutive ale spiralei este de 1,6 μm, ceea ce echivalează cu o densitate de aproximativ 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 μm și lățimea de 0,6 μm (figura 8.2). Lungimea cavităților și a suprafețelor este cuprinsă între 0,9 μm și 3,3 μm.

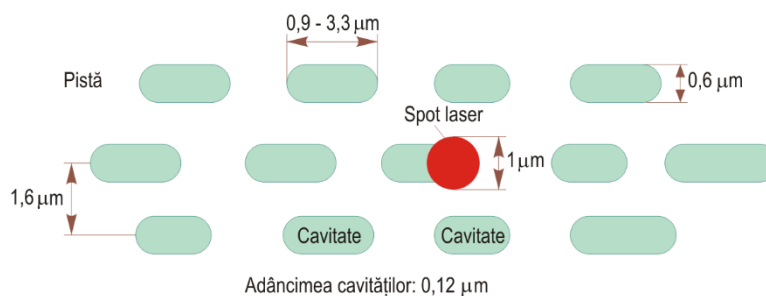


Figura 8.2. Dimensiunile cavităților și distanța între pistele unui disc compact.

8.2. 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 de fapt codificate prin diferite metode astfel încât să se asigure o densitate de înregistrare ridicată și recuperarea sigură a datelor.
- 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.

8.2.1. Înregistrarea și codificarea datelor pe suport

La nivelul cel mai scăzut, datele de pe disc sunt codificate astfel încât să se optimizeze procesul de conversie analog-digitală. Obiectivele codificării datelor la acest nivel cuprind 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. Deci, este necesară 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 nici cerința (1), nici cerințele (2) și (3). De exemplu, valoarea întregă 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, într-o zonă există un șir lung de biți cu valoarea 0 (sau 1), ansamblul de citire ar trebui să dispună de un ceas extrem de precis, care să comute exact la intervalul corespunzător lungimii unui bit (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 cavităților 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 acestora.

Biții sunt înregistrați pe suport utilizând codificarea NRZI (*Non Return to Zero Inverted*). Un bit de 1 este înregistrat printr-o schimbare a stării suportului, și anume, printr-o tranziție de la o cavitate la o suprafață (sau invers). O tranziție de la o cavitate la o suprafață corespunde unei tranziții de la o zonă cu reflectivitate redusă la o zonă cu reflectivitate ridicată (începutul unei noi suprafețe). O tranziție de la o suprafață la o cavitate corespunde unei tranziții de la o zonă cu reflectivitate ridicată la o zonă cu reflectivitate redusă (începutul unei noi cavități). Un bit de 0 este înregistrat prin lipsa de schimbare a stării suportului, ceea ce înseamnă continuarea unei suprafețe sau a unei cavități.

Înainte de înregistrare, biții de date sunt modulați (codificați) pentru a transforma secvențele binare arbitrare în secvențe care au anumite proprietăți avantajoase. Pentru suportul de înregistrare utilizat de discurile optice, o proprietate convenabilă ar fi ca șirurile înregistrate să nu conțină secvențe foarte scurte sau foarte lungi de biți de 0 sau 1. Secvențele foarte scurte conduc la frecvențe ridicate ale semnalului generat prin citirea informațiilor de pe disc, ceea ce poate cauza erori ale modului de detecție a biților. Secvențele foarte lungi conduc la inexactități de sincronizare ale circuitului PLL (*Phase-Locked Loop*) care regenerează semnalul

de ceas necesar pentru sincronizare și care ajustează semnalul de ceas la fiecare tranziție a semnalului de date.

Metoda de modulare utilizată constă în reprezentarea unui octet de date prin 14 biți și este numită codificare prin *modulare 8 la 14* (*Eight-to-Fourteen Modulation* – EFM). 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 satisfac cerința de a evita secvențe foarte scurte și foarte lungi de zerouri. Se utilizează o tabelă pentru a realiza corespondența între cele 256 de combinații valide pentru un octet și secvențele de 14 biți. Pentru unele valori zecimale, această corespondență este ilustrată în tabelul 8.1.

Tabelul 8.1. Exemple de codificare a valorilor de 8 biți prin secvențe de 14 biți (codificare 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 a evita existența a doi biți de 1 succesivi între două secvențe de cod EFM alăturate, după fiecare secvență de 14 biți sunt inserați 3 biți de legătură, care nu sunt luați în considerare la citirea datelor. În consecință, pentru reprezentarea unui octet de date sunt necesari 17 biți de canal: 14 biți EFM și 3 biți de legătură.

Codurile EFM satisfac cerințele unui cod RLL (*Run-Length Limited*). Lungimea secvenței (*“run-length”*) reprezintă numărul biților identici (de 1 sau de 0) pentru care semnalul nu are tranziții. În general, un cod RLL este specificat sub forma (d, k) RLL. Cei doi parametri, d și k , care specifică un cod RLL reprezintă, respectiv, numărul minim și numărul maxim de biți de 0 între doi biți consecutivi de 1. Terminologia codurilor RLL presupune o codificare NRZI, astfel încât biții de 1 indică tranziții și biții de 0 indică lipsa tranzițiilor; de aceea, se contorizează doar secvențele biților de 0. Codul RLL care a fost utilizat pentru alegerea codurilor EFM este $(2, 10)$ RLL. Ca o consecință, între doi biți de 1 se vor găsi cel puțin 2 și cel mult 10 biți de 0. Ca o altă consecință, lungimea minimă a cavităților și a suprafețelor este de 3 biți, iar lungimea maximă a acestora este de 11 biți.

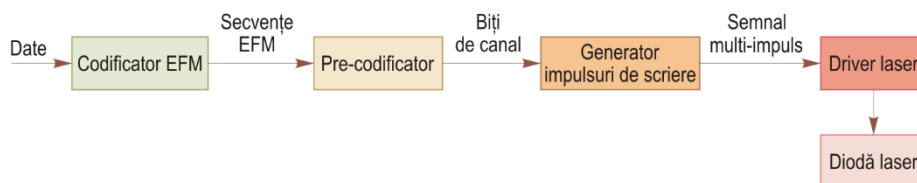


Figura 8.3. Structura unui sistem de înregistrare pentru discuri compact.

Pentru înregistrarea datelor pe discurile compact, mai întâi fiecare octet de date este codificat printr-un cod EFM de către un modul codificator EFM (figura 8.3). Apoi, secvențele binare rezultate sunt transformate cu ajutorul unui modul pre-codificator în așa-numiții biți de canal. Biții de canal sunt biții care sunt înscrisi efectiv pe disc. Un bit 1 de canal este asociat cu o anumită stare fizică a suportului (de exemplu, cu o cavitate), iar un bit 0 de canal este asociat cu cealaltă stare fizică (de exemplu, cu o suprafață). Biții de canal sunt transmiși apoi la un generator al impulsurilor de scriere unde aceștia sunt modulați într-un semnal cu impulsuri multiple. În sfârșit, impulsurile sunt transmise la un driver laser, care modulează puterea unui fascicul laser pentru a înregistra cavitățile pe disc.

Date binare (0x070A)	0000 0111	0000 1010
Secvențe EFM + biți de legătură	00 1001 0000 0000	100 10 0100 0100 0000 000
Biți de canal	00 1110 0000 0000	111 00 0111 1000 0000 000

Figura 8.4. Exemplu de codificare EFM pentru doi octeți de date și secvența corespunzătoare a biților de canal.

Figura 8.4 ilustrează un exemplu de codificare EFM pentru doi octeți de date și biții de canal generați din secvențele EFM și biții de legătură inserați. Șirul biților de canal poate fi partiționat în mai multe secvențe, fiecare secvență constând dintr-un număr de biți consecutivi identici.

Pentru a evita necesitatea unei piste separate de ceas, este necesară o sincronizare periodică. Din acest motiv, datele sunt divizate în blocuri numite *cadre*, la începutul fiecărui cadru existând un antet de sincronizare. Cadrul este unitatea de bază de sincronizare. Structura unui cadru este prezentată în figura 8.5.

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

Figura 8.5. Structura unui cadru al discurilor compact.

Cu excepția biților de sincronizare, toate celelalte câmpuri dintr-un cadru sunt codificate prin metoda EFM. Un cadru constă din următoarele:

- Un antet de sincronizare: 27 biți;
- Un octet de control: $14 + 3 = 17$ biți;
- 24 octeți de date, divizați în două câmpuri: $2 \times 12 \times (14 + 3) = 408$ biți;
- 8 octeți pentru codul de detecție și corecție a erorilor (CIRC – *Cross Interleaved Reed-Solomon Code*), divizați în două câmpuri: $2 \times 4 \times (14 + 3) = 136$ 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 6 eșantioane digitizate de 16 biți pentru fiecare canal audio. Ele sunt indicate în figura 8.5 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.

8.2.2. Primul nivel de corecție a erorilor

Erorile de date se pot datora prafului, amprentelor, defectelor de fabricație sau deteriorării discului. O caracteristică semnificativă a acestor erori este că ele apar adesea în secvențe lungi.

Sistemul de detecție și corecție a erorilor utilizat în interiorul cadrelor se numește *Cross Interleaved Reed-Solomon Code* (CIRC), după numele matematicienilor Reed și Solomon. Acest sistem este integrat la nivel hardware în unitățile de disc, având două componente. Componenta “*cross interleave*” divizează erorile lungi în mai multe erori scurte, iar componenta “*Reed-Solomon*” asigură corecția erorilor.

Se utilizează două decodificatoare *Reed-Solomon*. Pe măsură ce fiecare cadru este citit de pe disc, este decodificat mai întâi din 14 biți de canal (cei trei biți de legătură sunt ignorați) în octeți de date. Apoi, octeții din fiecare cadru (24 octeți de date și 8 octeți pentru corecția erorilor) sunt transmiși primului decodificator *Reed-Solomon*, care utilizează 4 din octeții pentru corecția erorilor și poate corecta un octet eronat din cei 32. Dacă nu sunt erori necorectabile, datele sunt transmise mai departe. Dacă există asemenea erori, datele se marchează ca fiind eronate în această etapă a decodificării.

Celor 24 de octeți de date și celor 4 octeți rămași pentru corecția erorilor li se aplică apoi *întârzieri inegale* înainte de a fi transmiși la cel de-al doilea decodificator *Reed-Solomon*. Aceste întârzieri inegale au ca rezultat o *intercalare* a datelor care divizează erorile lungi între mai multe treceri prin cel de-al doilea decodificator. Întârzierile sunt alese astfel încât erorile

lungi de până la 450 de octeți pot fi corectate în totalitate. După corecția erorilor din cei 24 octeți de date de către cel de-al doilea decodificator *Reed-Solomon*, se reface ordinea corectă a datelor.

Dacă este distrusă o zonă mai mare, în cazul discului audio recuperarea datelor se poate realiza prin interpolare. De exemplu, dacă pe disc s-a înregistrat un șir de valori, de exemplu 12, 16, 24, iar valoarea din mijloc nu poate fi citită (datorită impurităților sau a deteriorării suprafeței), această valoare se poate deduce prin interpolare ca fiind 18, care este media valorilor 12 și 24. Deși valoarea interpolată nu este exact cea corectă, diferențele pe plan acustic nu vor fi perceptibile.

8.2.3. Formatul unui sector

Un număr de 98 de cadre formează o unitate fizică de date numită *sector*. Deci, un sector conține 98 octeți de control, $24 \times 98 = 2352$ octeți de date și $8 \times 98 = 784$ octeți pentru detecția și corecția erorilor.

Tehnologia de bază utilizată la discurile de date CD-ROM este aceeași ca și la discurile audio. Fiecare sector trebuie să fie însă accesat direct, în acest scop fiind prevăzut un antet pentru fiecare sector. De asemenea, datele trebuie să aibă un grad mai ridicat de integritate, motiv pentru care este alocat un număr mai mare de octeți pentru detecția și corecția erorilor.

Pentru accesul direct la fiecare sector, se utilizează octeți de sincronizare și un antet care conține adresa sectorului. Cei 12 octeți de sincronizare au valori predefinite: primul și ultimul octet au valoarea 0, iar cei 10 octeți din mijloc au valoarea 0xFF. Această secvență permite unității de disc să identifice începutul unui nou sector.

În structura unui sector, după octeții de sincronizare urmează antetul de sector. Acesta constă din 4 câmpuri, fiecare câmp ocupând un octet:

- Câmpul M al adresei absolute;
- Câmpul S al adresei absolute;
- Câmpul F al adresei absolute;
- Câmpul de mod al datelor.

Adresa absolută a unui sector este codificată sub forma MSF (*Minute, Second, Frame*). Adresa este împărțită în trei câmpuri, și fiecare câmp conține două cifre BCD. Câmpul M conține partea cea mai semnificativă a adresei absolute (minutetele), având valori între 00 și 99. Câmpul S conține secunde și are valori între 00 și 59. Câmpul F conține partea cea mai puțin semnificativă a adresei absolute (numărul cadrului), cu valori între 00 și 74. Adresa MSF de început a pisteilor de date trebuie să fie exactă, dar pentru pisteile audio această adresă are o toleranță de ± 75 sectoare.

Câmpul de mod al datelor indică tipul informațiilor din sector (date sau audio/video), ca și modul de codificare audio.

Există două moduri de organizare a datelor pe discurile CD-ROM. În Modul 1, prevăzut pentru pisteile de date (programe), după octeții de sincronizare și antetul de sector urmează 2048 octeți de date. În continuare există un câmp auxiliar de date, care conține codul pentru detecția erorilor (*Error Detection Code* – EDC) și codul pentru corecția erorilor (*Error Correction Code* – ECC). Între cele două coduri există 8 octeți cu valoarea 0. Deoarece sunt necesare 75 de sectoare pentru o secundă, capacitatea totală a unui disc CD-ROM de 74 minute este de $2048 * 75 * 74 * 60 = 681.984.000$ octeți sau 650,39 MB. Structura unui sector în Modul 1 este prezentată în figura 8.6.

Sincronizare	Antet	Date	EDC	0	ECC
12 octeți	4 octeți	2048 octeți	4 octeți	8 octeți	276 octeți

Figura 8.6. Structura unui sector al discurilor CD-ROM în Modul 1.

În Modul 2, prevăzut pentru aplicații mai puțin critice (audio și video), câmpul auxiliar este de asemenea utilizat pentru datele utilizatorului, în locul codurilor pentru detecția și corecția erorilor. Aceasta deoarece, de obicei, nu este necesar să se asigure o integritate ridică-

tă pentru informațiile audio și video. În acest caz, un sector conține $2048 + 288 = 2336$ octeți de date. Structura unui sector în Modul 2 este prezentată în figura 8.7.

Sincronizare	Antet	Date
12 octeți	4 octeți	2336 octeți

Figura 8.7. Structura unui sector al discurilor CD-ROM în Modul 2.

Pe același disc pot exista atât date, cât și piste audio, imagini video etc. Asemenea discuri sunt *discuri în mod mixt*, care se conformează standardului ISO/IEC 10149. La aceste discuri, prima pistă conține date (în Modul 1), iar celelalte piste conțin informații audio sau video (în Modul 2). Problema care poate apare în acest caz este că unele aparate de redare CD nu efectuează automat saltul peste prima pistă, existând posibilitatea deteriorării echipamentului audio dacă prima pistă este redată cu un aparat CD audio la un volum ridicat.

O unitate CD-ROM obișnuită, care este capabilă să citească discuri în modul mixt nu poate citi date în același timp cu redarea audio, deoarece datele și informațiile audio se află pe piste diferite. De exemplu, o asemenea unitate nu permite redarea sincronizată a sunetului și a imaginilor. Pentru a rezolva această problemă, în specificațiile discurilor CD-I (CD *Interactive*) și CD-ROM/XA (CD-ROM *eXtended Architecture*) a fost adăugat un nou format. Acest format s-a obținut prin împărțirea Modulului 2 al sectoarelor în două forme: Forma 1 și Forma 2.

Forma 1 se utilizează pentru datele numerice și este similară cu Modul 1 al discurilor CD-ROM, însă există un identificator suplimentar în antetul de sector (figura 8.8).

Sincronizare	Antet	Sub-antet	Date	EDC	ECC
12 octeți	4 octeți	8 octeți	2048 octeți	4 octeți	276 octeți

Figura 8.8. Structura unui sector al discurilor CD-ROM/XA în Modul 2, Forma 1.

Cei 8 octeți cu valoarea 0 din cadrul sectorului CD-ROM, aflați între câmpurile EDC și ECC, sunt plasați imediat după octeții antetului, formând un *sub-antet* care se utilizează pentru identificarea tipului de sector (cu Forma 1 sau Forma 2).

Forma 2 se utilizează pentru informații audio și video, fiind similară cu Modul 2 al discurilor CD-ROM. Cei 276 octeți ai câmpului ECC sunt utilizați pentru păstrarea datelor, astfel încât un sector conține 2324 octeți cu date utilizabile (figura 8.9).

Sincronizare	Antet	Sub-antet	Date	EDC
12 octeți	4 octeți	8 octeți	2324 octeți	4 octeți

Figura 8.9. Structura unui sector al discurilor CD-ROM/XA în Modul 2, Forma 2.

Sub-antetul conține informații ca numărul fișierului, numărul canalului, un octet de sub-mod și informații de control.

- *Numărul fișierului* poate fi cuprins între 0 și 255; 0 indică un fișier care va fi citit în mod continuu (neîntrerăsurat).
- *Numărul canalului* este prevăzut pentru a permite citirea separată sau combinată a mai multor sectoare și selecția lor în timp real. Pentru datele audio sunt rezervate canalele între 0 și 15. Pentru datele video și datele numerice sunt utilizate canalele între 16 și 31.
- *Octetul de sub-mod* permite identificarea tipului de sector (date sau audio/video) și memorarea unor parametri.
- *Informațiile de control* se referă la modul de înregistrare (mono/stereo) și la nivelul sunetului.

Cele două tipuri de sectoare, de date și audio/video, pot fi amplasate alternativ pe o singură pistă în Modul 2. Datele sunt înscrise în Forma 1, iar informațiile audio/video sunt înscrise în Forma 2. Această tehnică de întrețesere permite redarea sincronizată a imaginilor și a sunetelor.

8.2.4. Al doilea nivel de corecție a erorilor

În cazul discurilor audio, o eroare necorectată la nivelul cadrelor provoacă un zgomot, deseori imperceptibil, în timpul audiției. În cazul discurilor utilizate pentru date, o eroare necorectată poate determina ca discul să devină inutilizabil. Rata de eroare de 10^{-9} asigurată de metoda CIRC nu este acceptabilă pentru discurile care conțin date. Pentru îmbunătățirea acestei rate, la discurile CD-ROM este prevăzut un al doilea nivel de detecție și corecție a erorilor (EDC/ECC) pentru datele înregistrate în Modul 1.

Pentru al doilea nivel de corecție a erorilor sunt alocați 280 de octeți suplimentari, 4 octeți pentru detecția erorilor (EDC) și 276 octeți pentru corecția erorilor (ECC). Tehnica de corecție a erorilor utilizată se numește L-EC (*Layered Error Correction*); rata de eroare asigurată este de 10^{-12} .

Codul pentru detecția erorilor este un cod ciclic redundant (*Cyclic Redundancy Check* - CRC). Acesta ocupă primii 4 octeți ai câmpului auxiliar de date și asigură o foarte mare probabilitate de detecție a erorilor necorectate. Codul de corecție a erorilor se bazează pe același principiu ca și primul nivel de corecție a erorilor, utilizând intercalarea și codificarea *Reed-Solomon*. Acest cod ocupă ultimii 276 de octeți ai câmpului auxiliar de date.

8.2.5. Formatele informațiilor din sub-canale

Fiecare bit al unui octet de control este identificat printr-o literă: P, Q, R, S, T, U, V și W. Biții cu același rang din toți octeții de control (și care au asignată aceeași literă) formează un *sub-canal*. Astfel, toți biții care ocupă prima poziție în toți cei 98 octeți de control formează *sub-canalul P*. Biții acestui sub-canal sunt utilizați separat și se pot folosi pentru redarea audio în surdină sau ca separator al pistelor audio. Biții din a doua poziție formează *sub-canalul Q*. Ultimii 6 biți sunt combinați în *sub-canalul R-W*. Acest sub-canal este utilizat de diferite variante de discuri compact audio pentru păstrarea unor informații cum sunt textele melodiilor sau anumite date grafice.

Sub-canalul Q are o structură mai complexă. Pentru un sector, biții sub-canalului Q sunt organizați într-un bloc de informații. Acest bloc constă din 98 de biți, câte un bit din fiecare cadru al sectorului. Structura acestui bloc este prezentată în tabelul 8.2.

Tabelul 8.2. Blocul de informații al sub-canalului Q.

Câmp	Semnificație
S0, S1	2 biți de sincronizare a sub-canalului
CONTROL	4 biți pentru definirea tipului informațiilor din pista curentă: x0xx = pistă audio 01xx = pistă de date 11xx = rezervat xx0x = copierea digitală interzisă xx1x = copierea digitală permisă
ADR	4 biți pentru definirea formatului utilizat de câmpul DATA-Q
DATA-Q	72 biți de date
CRC	16 biți pentru codul CRC al câmpurilor CONTROL, ADR și DATA-Q

Există trei formate posibile pentru câmpul DATA-Q. Acestea sunt descrise în continuare.

Formatul 1

Primul format trebuie să fie prezent în cel puțin 9 din 10 sectoare consecutive. Sunt posibile două structuri diferite ale acestui format. Pentru zona *lead-in*, structura câmpului DATA-Q este ilustrată în figura 8.10.

ADR	DATA-Q								
0001	TNO	POINT	MIN	SEC	FRAME	ZERO	PMIN	PSEC	PFRAME

Figura 8.10. Formatul 1 al câmpului DATA-Q pentru zona *lead-in*.

Pentru celelalte piste (date, audio și zona *lead-out*) structura câmpului DATA-Q este ilustrată în figura 8.11.

ADR	DATA-Q								
0001	TNO	INDEX	MIN	SEC	FRAME	ZERO	AMIN	ASEC	AFRAME

Figura 8.11. Formatul 1 al câmpului DATA-Q pentru pistele de date, piste audio și zona *lead-out*.

TNO (*Track Number*) reprezintă numărul pistei și se exprimă prin două cifre BCD.

INDEX (*Index to TNO*) reprezintă subdiviziunea în cadrul pistei TNO. Valoarea 00 indică o zonă de pauză sau *pre-gap*. În zona *lead-out* valoarea indexului este 01. Într-o pistă audio, prima valoare a indexului este 01 și această valoare se poate incrementa numai cu 1. Într-o pistă de date indexul are valoarea 01.

Câmpurile MIN, SEC, FRAME conțin timpul relativ la începutul pistei în minute, secunde și cadre, fiecare fiind exprimat sub forma a două cifre BCD (adresa relativă a sectorului codificată sub forma MSF). În zonele de pauză și *pre-gap*, adresele logice sunt negative și descresc în valoare absolută, ajungând la valoarea 0 la sfârșitul zonei respective. În celelalte zone, inclusiv în zonele *lead-in* și *lead-out*, adresele cresc. O secundă este împărțită în 75 de cadre.

Câmpul ZERO conține valoarea 00.

Câmpurile AMIN, ASEC, AFRAME conțin adresa absolută a sectorului, sub forma MSF.

Câmpurile POINT, PMIN, PSEC, PFRAME conțin tabela de conținut pe durata zonei *lead-in* (TNO = 0). Această tabelă de conținut este repetată în mod continuu în zona *lead-in*. În fiecare tabelă de conținut, câmpurile individuale sunt repetate de trei ori.

Valoarea dată de PMIN, PSEC și PFRAME reprezintă punctul de început al pistei indicate de POINT, ca o valoare absolută pe scala timpului, cu o precizie de ± 1 secundă. Poziția de început a unei piste este prima poziție cu noua valoare a pistei și indexul diferit de 00.

- Dacă POINT = 0xA0, PMIN indică numărul pistei pentru prima piesă audio de pe disc; PSEC și PFRAME sunt 0.
- Dacă POINT = 0xA1, PMIN indică numărul ultimei piste de pe disc; PSEC și PFRAME sunt 0.
- Dacă POINT = 0xA2, PMIN, PSEC și PFRAME conțin punctul de început al zonei *lead-out*.

Formatul 2

Al doilea format al câmpului DATA-Q este opțional, având structura din figura 8.12.

ADR	DATA-Q														
0010	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	ZERO	AFRAME

Figura 8.12. Formatul 2 al câmpului DATA-Q.

N1..N13 reprezintă *numărul de catalog* al discului exprimat prin 13 cifre BCD, conform valorilor codului uniform al produsului (cod de bare UPC/EAN). Aceste valori sunt controlate de *Uniform Product Code Council* și de *European Article Number Council*. Numărul de catalog nu se modifică pe un disc. Dacă nu s-a codificat nici un număr de catalog, N1..N13 conțin cifre de 0.

Câmpul ZERO conține 12 biți de 0.

Câmpul AFRAME are aceeași semnificație ca și la primul format (două cifre BCD între 00 și 74). În zona *lead-in* (TNO = 00), toți cei 8 biți ai câmpului au valoarea 0.

Formatul 3

Al treilea format al câmpului DATA-Q este, de asemenea, opțional, fiind utilizat pentru a asigna un număr unic unei piste audio prin codul ISRC (*International Standard Recording Code*). Codul ISRC înregistrat pe suport este definit în figura 8.13. În zonele *lead-in* și *lead-out*, acest format nu este prezent pe disc. Codul ISRC se modifică imediat ce s-a modificat numărul pistei (TNO).

ADR																
0011	I1	I2	I3	I4	I5	0	0	I6	I7	I8	I9	I10	I11	I12	ZERO	AFRAME

Figura 8.13. Formatul 3 al câmpului DATA-Q (codul ISRC).

Câmpurile I1 - I12 definesc codul ISRC, astfel:

- I1..I2: Codul țării;
- I3..I5: Codul proprietarului;
- I6..I7: Anul înregistrării;
- I8..I12: Numărul de serie al înregistrării.

Fiecare caracter din câmpurile I1..I5 este reprezentat printr-o codificare pe 6 biți. Caracterele I6..I12 sunt codificate în BCD. Câmpul ZERO conține 4 biți de 0. Câmpul AFRAME are aceeași semnificație ca și la formatele 1 și 2.

8.2.6. Organizarea discului

O *pistă* poate fi considerată ca o partiție a spațiului de adresare a discului. Un disc CD poate conține între 1 și 99 de piste, numerotate consecutiv. Totuși, prima pistă de informație poate avea un număr mai mare de 1. Pistele au o lungime minimă de 300 sectoare. Toate sectoarele unei piste trebuie să fie de același tip (să conțină fie date, fie informații audio/video), și trebuie să fie înregistrate în același mod. Fiecare schimbare a tipului de informații de pe disc necesită o trecere la o altă pistă. Un disc care conține atât date, cât și informații audio/video, va avea cel puțin două piste, una pentru date și una pentru audio/video.

Între pistele înregistrate cu tipuri diferite de informații trebuie să existe *zone de tranziție*. În plus, pot exista zone de tranziție la începutul și la sfârșitul oricărei piste. Zonele de tranziție conțin valori nule și conținutul lor nu poate fi citit. Pentru pistele audio zonele de tranziție se numesc *zone de pauză*. Pentru pistele de date zonele de tranziție se numesc *zone pre-gap* și *post-gap*. Standardele IEC 908 și ISO/IEC 10149 specifică duratele minime ale acestor zone. Nu sunt specificate duratele maxime. Zonele de tranziție sunt formate și adresa logică a sectoarelor continuă să se incrementeze pe parcursul acestor zone.

Un disc CD este format din trei zone: zona *lead-in*, care ocupă primii 4 mm de la centrul discului; zona de date; zona *lead-out*, care ocupă ultimul mm al discului. Zona *lead-in* a discului este identificată ca pista 0, dar nu este adresabilă prin setul de comenzi al unității. Sub-canalul Q al acestei zone conține tabela de conținut TOC (*Table Of Contents*) a discului. Zona *lead-out* este identificată ca pista 0xAA, dar nu este adresabilă prin setul de comenzi al unității.

Tabela de conținut indică adresa absolută a primului sector de informații din fiecare pistă. Această tabelă păstrează de asemenea informații de control pentru fiecare pistă (date/audio, metoda de codificare audio etc.). Tabela de conținut nu păstrează însă informații despre diferitele moduri ale pistelor de date (de exemplu, Modul 1 sau Modul 2). Adresa absolută a unui sector este codificată sub forma MSF (*Minute, Second, Frame*) și indică timpul față de începutul discului în minute, secunde și cadre.

Adresarea logică a informațiilor de pe discul CD poate utiliza orice dimensiune a blocului logic. Dimensiunea blocului logic este de obicei un divizor exact sau un multiplu întreg al numărului de octeți dintr-un sector. La discurile care conțin date și la cele mixte (date și audio/video), adresa logică 0 se asignează blocului cu adresa MSF 00/02/00. La discurile audio, adresa logică 0 se asignează blocului de început al pistei 1. Se poate utiliza adresa de

început a pistei 1 din tabela de conținut, sau se poate asigna adresa logică 0 blocului cu adresa MSF 00/02/00.

Un *index* este o partiție a unei piste. Zonele *pre-gap* sunt codificate cu o valoare a indexului egală cu 00. Zonele de pauză de la începutul pistelor audio sunt codificate de asemenea cu o valoare a indexului egală cu 00. Primul sector de informații a unui sector are un index egal cu 01. Sunt permise valori consecutive până la 99. Informațiile de index nu sunt memorate în tabela de conținut. Indexul este codificat în sub-canalul Q. Nu toate sectoarele au informații de index (cerința este ca 9 din 10 sectoare să aibă codificată o valoare de index). Pentru un sector fără o valoare de index se presupune aceeași valoare cu cea a sectorului precedent.

Informațiile din sub-canalele fiecărui sector (mai precis, din sub-canalul Q) conțin adresa relativă a sectorului, indicând distanța față de primul sector de informație al pistei. Această valoare este negativă în zonele *pre-gap* și crește spre 0 în aceste zone.

8.3. 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 atât modul de operare *master*, cât și cel *slave*. Aceste echipamente utilizează aceleași semnale care sunt specificate în standardele interfeței ATA.

Deși pentru conectarea unei unități de discuri optice prin interfața ATAPI se utilizează aceeași interfață fizică ATA, interfața logică diferă și permite facilități suplimentare. Echipamentele conectate la interfața ATA utilizează opt registre pentru comunicația cu calculatorul, 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 unor structuri numite *pachete de comandă*. 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 de registre necesare. 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, astfel încât să fie necesare modificări minime ale driverelor existente. 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 programul BIOS și driverele mai vechi să ignore aceste echipamente și să nu confunde datele de la acestea cu datele de la echipamentele ATA. Comenzile care nu sunt recunoscute nu vor fi executate și se va genera o cerere de întrerupere pentru a semnală terminarea comenzii cu o stare de eroare (*ABORTED*).

Pentru identificarea unui periferic ATAPI s-a prevăzut comanda IDENTIFY PACKET DEVICE; informațiile returnate de această comandă sunt utilizate de driverele de nivel inferior pentru a executa 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 definite în standardele SCSI, cele mai multe facilități ale protocolului 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 operații de deconectare / reconectare.
- Pachetele de comenzi au o lungime de 12 octeți, spre deosebire de cele ale interfeței SCSI, care pot avea o lungime de 6, 10, 12 sau 16 octeți. Sunt definite însă și pachete de 16 octeți pentru compatibilitate cu echipamentele viitoare. Dimensiunea pachetelor acceptate de un echipament este definită în cuvântul 0 al informațiilor returnate de 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 programe BIOS existente și drivere ale sistemelor de operare.

8.4. Registrele interfeței ATAPI

Comunicația cu dispozitivele periferice se realizează prin registre de I/E. 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.

Tabelul 8.3 prezintă registrele interfeței ATAPI. Aceste registre sunt aceleași ca și registrele interfeței ATA, deși numele unor registre este diferit. Aceste registre sunt descrise în lucrarea de laborator *Interfața ATA*, secțiunile 7.6.1-7.6.12.

Tabelul 8.3. Registrele interfeței ATAPI.

Registre ATAPI		
Deplasament	Registre din blocul de comandă	
	La citire	La scriere
0	Data	Data
1	Error	Features
2	Interrupt Reason	Sector Count
3	LBA Low	LBA Low
4	Byte Count Low	Byte Count Low
5	Byte Count High	Byte Count High
6	Device	Device
7	Status	Command
Registre din blocul de control		
6	Alternate Status	Device Control

8.5. Execuția comenzilor ATAPI

Controlul 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 calculatorul gazdă pentru execuția comenzilor și returnarea stării către calculatorul gazdă 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 *Device* 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. Comanda care trebuie executată de echipament va fi transmisă ca un pachet prin registrul de date *Data* și nu prin celelalte registre ale interfeței.
2. Parametrii comenzii sunt transmiși prin pachetul de comandă și prin registrele interfeței.
3. Se utilizează un contor de octeți pentru a determina cantitatea datelor pe care trebuie să le transfere calculatorul gazdă la fiecare întrerupere DRQ.
4. Pentru a indica faptul că se va utiliza modul de transfer prin DMA, se utilizează registrul *Features* în locul utilizării diferitelor coduri de operație.

5. Starea finală este returnată calculatorului gazdă ca o nouă întrerupere după transferul ultimei date și nu împreună cu ultimul bloc de date.

Aceste noi reguli se aplică până când starea care indică terminarea comenzii a fost citită de calculatorul gazdă. După aceasta, definițiile registrelor și protocolul utilizat vor fi cele definite de standardul ATA.

Comanda PACKET este transmisă ca orice comandă normală ATA, prin inițializarea registrelor interfeței, setarea bitului de selecție al unității din registrul *Device* 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 sau 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 calculatorul gazdă, s-a creat posibilitatea transferului unui număr variabil de octeți prin utilizarea registrelor pentru contorul de octeți *Byte Count*. Echipamentul indică pentru calculatorul gazdă numărul de octeți care trebuie transferați la fiecare întrerupere DRQ. Înaintea transferului de date, calculatorul gazdă trebuie să citească registrele *Byte Count Low* și *Byte Count High* și să țină cont de lungimea cerută. Echipamentul ATAPI și calculatorul gazdă vor avea contoare proprii de octeți și vor transfera date până când aceste contoare ajung la zero. Pentru anumite comenzi, ca *MODE SENSE*, calculatorul gazdă 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.

Prin utilizarea registrelor pentru contorul de octeți *Byte Count*, calculatorul gazdă are și posibilitatea de a indica echipamentului numărul maxim de octeți pe care le poate prelua într-un singur pachet DRQ sau dimensiunea preferată a pachetului. Pentru toate comenzile care necesită transferul unor date, calculatorul gazdă trebuie să încarce registrele *Byte Count* cu lungimea dorită înaintea transmiterii comenzii PACKET. 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 calculatorul gazdă.

Dacă echipamentul solicită transferul unui număr mai mare de octeți decât cel cerut de protocolul comenzii, calculatorul gazdă trebuie să extindă datele transmise echipamentului și să preia datele suplimentare la citirea acestora de la echipament. Echipamentul nu este responsabil cu extinderea datelor, transferând numai numărul de octeți specificat de calculatorul gazdă.

Anumite comenzi ATAPI sunt imediate. Pentru aceste comenzi, echipamentul returnează starea de terminare a comenzii imediat, continuând execuția propriu-zisă a 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 în coada de comenzi.
- Dacă se înscrie o nouă comandă ATA în registrul de comandă înaintea terminării unei comenzi, comanda în curs va fi oprită din execuție, iar noua comandă va fi abandonată cu setarea bitului *ABRT* (*Aborted Command*) din registrul de eroare *Error*.
- Noile comenzi ATAPI recepționate în timp ce o comandă ATAPI precedentă este în curs de execuție vor cauza abandonarea ambelor comenzi cu setarea bitului *CHK* (*Check Condition*) din registrul de stare *Status*.

Există echipamente care permit suprapunerea comenzilor, îmbunătățind performanțele sistemului. Aceste echipamente eliberează magistrala înaintea terminării comenzii în curs, permițând utilizarea magistralei de celălalt echipament.

8.6. Protocoale utilizate pentru comenzile ATAPI

8.6.1. Protocolul ATAPI 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. Presupunând că generarea întreruperilor nu este validată, din punctul de vedere al calculatorului gazdă protocolul pentru comenzile care nu transferă date este următorul:

1. Programul citește registrul de stare *Status* și așteaptă până când biții BSY și DRQ devin 0. Dacă acești biți nu devin 0 într-un anumit timp, programul trebuie să abandoneze execuția protocolului.
2. Programul inițializează registrul *Device*. În registrul *Device*, bitul DEV trebuie resetat la 0 pentru accesul la unitatea 0 sau trebuie setat la 1 pentru accesul la unitatea 1.
3. Programul scrie codul comenzii PACKET (0xA0) în registrul de comandă.
4. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0.
5. Programul citește registrul de stare *Status* și testează bitul DRQ. Dacă bitul DRQ este 0, comanda nu a fost recunoscută de unitate și execuția protocolului este terminată. Dacă bitul DRQ este setat, programul continuă cu pasul 6.
6. Programul scrie pachetul de comandă în registrul de date *Data*, câte un cuvânt la un moment dat. Dacă pachetul de comandă este definit ca un tablou de octeți, doi octeți consecutivi trebuie împachetați într-un cuvânt înaintea scrierii cuvântului în registrul de date. Primul octet trebuie plasat în partea inferioară a cuvântului, iar următorul octet trebuie plasat în partea superioară a cuvântului.
7. Programul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, acesta poate citi registrul *Alternate Status*, ignorând rezultatul citirii.
8. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0. Dacă bitul BSY nu devine 0 într-un anumit timp, programul trebuie să abandoneze execuția protocolului.
9. Programul citește registrul de stare *Status* și așteaptă până când bitul DRQ devine 0, moment în care execuția comenzii este terminată.

8.6.2. Protocolul ATAPI pentru intrare în modul PIO

Acest protocol este utilizat de comenzi cum sunt INQUIRY sau READ. Execuția cuprinde transferul unui număr necunoscut de octeți de date de la echipament la calculatorul gazdă. Presupunând că generarea întreruperilor nu este validată, din punctul de vedere al calculatorului gazdă, protocolul pentru intrare în modul PIO este următorul:

1. Programul citește registrul de stare *Status* și așteaptă până când biții BSY și DRQ devin 0. Dacă acești biți nu devin 0 într-un anumit timp, programul trebuie să abandoneze execuția protocolului.
2. Programul inițializează registrul *Device*, registrul *Byte Count Low* și registrul *Byte Count High*. În registrul *Device*, bitul DEV trebuie resetat la 0 pentru accesul la unitatea 0 sau trebuie setat la 1 pentru accesul la unitatea 1. Registrele *Byte Count* trebuie încărcate cu numărul maxim de octeți care trebuie transferați de unitate la fiecare setare a bitului DRQ. Pentru comenzile simple utilizate în această lucrare de laborator, aceste registre trebuie încărcate cu numărul total de octeți care trebuie transferați de unitate pentru comanda particulară care va fi lansată.
3. Programul înscrie codul comenzii PACKET în registrul de comandă.
4. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0.

5. Programul citește registrul de stare *Status* și testează bitul DRQ. Dacă bitul DRQ este 0, comanda nu a fost recunoscută de unitate și execuția protocolului este terminată. Dacă bitul DRQ este setat, programul continuă cu pasul 6.
6. Programul scrie pachetul de comandă în registrul de date *Data*, câte un cuvânt la un moment dat. Dacă pachetul de comandă este definit ca un tablou de octeți, doi octeți consecutivi trebuie împachetați într-un cuvânt înaintea scrierii cuvântului în registrul de date. Primul octet trebuie plasat în partea inferioară a cuvântului, iar următorul octet trebuie plasat în partea superioară a cuvântului.
7. Programul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, acesta poate citi registrul *Alternate Status*, ignorând rezultatul citirii.
8. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0. Dacă bitul BSY nu devine 0 într-un anumit timp, programul trebuie să abandoneze execuția protocolului.
9. Programul citește registrul de stare *Status* și testează bitul DRQ. Dacă bitul DRQ este 0, unitatea a terminat comanda cu o eroare și execuția protocolului este terminată. Dacă bitul DRQ este setat, programul continuă cu pasul 10.
10. Programul citește registrele *Byte Count Low* și *Byte Count High* și inițializează o variabilă de tip `WORD` cu conținutul acestor registre. Programul transferă un bloc de date prin citirea într-o buclă a registrului de date *Data* și memorarea acestuia într-un buffer, utilizând ca și contor al buclei variabila inițializată anterior. Dacă bufferul este definit ca un tablou de octeți, cuvântul citit din registrul de date trebuie memorat în doi octeți consecutivi din buffer. Partea inferioară a cuvântului trebuie memorată în primul octet, iar partea superioară a cuvântului trebuie memorată în următorul octet.
11. Programul așteaptă un timp corespunzător unui ciclu de transfer PIO. De exemplu, acesta poate citi registrul *Alternate Status*, ignorând rezultatul citirii.
12. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0. Dacă bitul BSY nu devine 0 într-un anumit timp, programul trebuie să abandoneze execuția protocolului.
13. Programul citește registrul de stare *Status* și dacă bitul DRQ este 1, continuă cu transferul unui nou bloc de date (pasul 10). Dacă bitul DRQ este 0, execuția comenzii este terminată.

8.7. Comenzi ATAPI

8.7.1. Identificarea perifericelor ATAPI

Atunci când perifericele ATAPI sunt puse sub tensiune, se execută o resetare hardware sau software a acestora, sau se execută una din comenzile ATA `DEVICE RESET` sau `EXECUTE DEVICE DIAGNOSTIC`, aceste periferice încarcă în unele registre ale interfeței ATAPI o semnătură care permite identificarea perifericelor ATAPI. De asemenea, comanda `ATA IDENTIFY DEVICE` nu va fi executată de perifericele ATAPI, iar acestea vor seta bitul `ABRT` al registrului de eroare *Error* și vor încărca o parte din registrele interfeței cu aceeași semnătură ca și la execuția comenzii `DEVICE RESET`.

Semnătura care identifică perifericele ATAPI și registrele ATAPI în care este plasată această semnătură sunt indicate în tabelul 8.4 pentru interfața ATAPI paralelă și pentru interfața ATAPI serială.

Tabelul 8.4. Semnătura caracteristică perifericelor ATAPI.

Registru ATAPI	Interfața ATAPI paralelă	Interfața ATAPI serială
Interrupt Reason	0x01	0x01
LBA Low	0x01	0x01
Byte Count Low	0x14	0x69
Byte Count High	0xEB	0x96

Pentru a detecta dacă un anumit periferic conectat la o interfață ATA/ATAPI este un periferic ATAPI, se poate transmite comanda IDENTIFY DEVICE perifericului respectiv, testând dacă această comandă este recunoscută de periferic. Dacă execuția comenzii este abandonată cu starea de eroare *ABORTED*, indicată prin setarea bitului ABRT din registrul de eroare *Error*, registrele *Byte Count Low* și *Byte Count High* vor conține semnătura caracteristică perifericelor ATAPI. Dacă aceste registre conțin valorile indicate în tabelul 8.4, perifericul este unul ATAPI și recunoaște comanda PACKET. În continuare, se poate transmite comanda IDENTIFY PACKET DEVICE (cu codul 0xA1) pentru a afla informații detaliate despre acel periferic ATAPI, utilizând același protocol ca și cel pentru execuția comenzii IDENTIFY DEVICE.

Pentru transmiterea comenzii IDENTIFY DEVICE, se poate utiliza protocolul ATA pentru intrare în modul PIO, descris în lucrarea de laborator *Interfața ATA*, secțiunea 7.7.2. Pentru mai multă claritate, mai jos se prezintă operațiile necesare pentru transmiterea și execuția acestei comenzi, completate cu operațiile necesare pentru detectarea abandonării comenzii de către un periferic ATAPI.

1. Programul citește registrul de stare *Status* și așteaptă până când biții BSY și DRQ devin 0. Dacă acești biți nu devin 0 într-un anumit timp, programul trebuie să abandoneze execuția operației.
2. Programul setează bitul DEV din registrul *Device* în funcție de unitatea căreia i se transmite comanda (DEV = 0 pentru unitatea 0, DEV = 1 pentru unitatea 1).
3. Programul înscrie codul comenzii IDENTIFY DEVICE în registrul de comandă.
4. Programul citește registrul de stare *Status* și așteaptă până când bitul BSY devine 0.
5. Programul citește registrul de stare *Status* și testează bitul DRQ. Dacă bitul DRQ este setat, ceea ce înseamnă că unitatea este o unitate ATA deoarece aceasta a recunoscut comanda, programul continuă cu pasul 6. Dacă bitul DRQ este 0, comanda nu a fost recunoscută de unitate. În acest caz, programul testează dacă bitul ABRT din registrul de eroare *Error* este setat.
 - Dacă bitul ABRT este setat, programul citește conținutul registrelor *Byte Count Low* și *Byte Count High*. Dacă aceste registre conțin valorile indicate în tabelul 8.4, unitatea este identificată ca o unitate ATAPI; în caz contrar, unitatea nu este o unitate ATAPI. În ambele cazuri, secvența de operații este terminată.
 - Dacă bitul ABRT este 0, nu există nicio unitate prezentă cu numărul specificat în registrul de selecție al dispozitivului, iar secvența de operații este terminată.
6. Programul transferă 256 cuvinte de date prin citirea repetată a registrului de date. Atunci când s-au citit toate cele 256 cuvinte, secvența de operații este terminată.

8.7.2. Comanda IDENTIFY PACKET DEVICE

Această comandă este similară cu comanda ATA IDENTIFY DEVICE, dar este recunoscută doar de unitățile ATAPI. Atunci când recepționează această comandă, unitatea ATAPI pregătește un bloc de 256 cuvinte cu informații despre unitate: modelul, numărul reviziei, numărul de serie etc. Calculatorul gazdă poate transfera blocul de date prin citirea succesivă a registrului de date *Data*. Protocolul pentru această comandă este protocolul ATA pentru intrare în modul PIO (descris în lucrarea de laborator *Interfața ATA*, secțiunea 7.7.2). Cu excepția

bitului DEV care trebuie resetat sau setat în registrul *Device*, nu trebuie inițializate alte registre înaintea scrierii codului de comandă în registrul de comandă *Command*.

Unitățile ATAPI nu vor raporta o eroare după execuția acestei comenzi. Unitățile ATA vor seta bitul ABRT în registrul de eroare *Error*.

Tabelul 8.5 prezintă semnificația unei părți a cuvintelor de 16 biți care sunt returnate de comanda IDENTIFY PACKET DEVICE. Unii parametri ai unității sunt definiți ca șiruri de caractere ASCII. Fiecare cuvânt conține două caractere ASCII: primul caracter este conținut în octetul cel mai semnificativ al cuvântului, iar al doilea caracter este conținut în octetul cel mai puțin semnificativ.

Tabelul 8.5. Semnificația unor cuvinte returnate de comanda IDENTIFY PACKET DEVICE.

Cuvânt	Semnificație
0	Informații generale de configurație
10-19	Număr de serie (20 de caractere ASCII)
23-26	Revizie firmware (8 caractere ASCII)
27-46	Număr model (40 de caractere ASCII)

Biții 1..0 ai cuvântului 0 indică lungimea pachetelor de comandă ATAPI recunoscute de unitate. Dacă acești biți sunt 00, lungimea pachetelor de comandă este de 12 octeți, iar dacă acești biți sunt 01, lungimea pachetelor de comandă este de 16 octeți; alte valori sunt rezervate.

8.7.3. Structura pachetului de comandă ATAPI

O comandă ATAPI este transmisă unui dispozitiv ATAPI prin intermediul unui pachet. Pentru unele comenzi, pachetul este urmat de o listă de parametri care se transmite după generarea unei întreruperi ca urmare a transmiterii pachetului de comandă.

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

Figura 8.14. Structura tipică a unui pachet de comandă ATAPI de 12 octeți.

În general, pachetele de comandă ATAPI au lungimea de 12 octeți. Figura 8.14 prezintă structura tipică a unui pachet de comandă ATAPI. Câmpurile au aceeași semnificație cu câmpurile blocului descriptor de comenzi de la interfața SCSI. Octet ms reprezintă octetul mai semnificativ, iar Octet mps reprezintă octetul mai puțin semnificativ. Primul octet al pachetului conține întotdeauna un *cod de operație*, reprezentând codul comenzii care trebuie executată de dispozitiv. *Adresa blocului logic* din cadrul unei unități logice începe cu blocul zero și trebuie să fie contiguă până la ultimul bloc logic al unității logice. Dimensiunea adreselor blocurilor logice este de 32 de biți. *Lungimea transferului* specifică lungimea datelor care trebuie transferate, fie în număr de blocuri, fie în număr de octeți, în funcție de comandă. *Lungimea alocată* specifică numărul maxim de octeți alocați de calculatorul gazdă pentru datele transmise de către dispozitiv. Această lungime se utilizează pentru a limita numărul de octeți returnați de către dispozitiv.

Ultimul octet al pachetului de comandă conține un *cod de control*. Pentru comenzile obișnuite, acest cod de control trebuie să fie setat la 0. Câmpurile rezervate trebuie setate la 0.

8.7.4. Lista comenzilor ATAPI pentru unitățile CD/DVD

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

Tabelul 8.6 prezintă lista comenzilor ATAPI pentru unitățile CD/DVD.

Tabelul 8.6. Comenzi ATAPI pentru unitățile CD/DVD.

Comandă	Cod operație	Tip
BLANK	0xA1	E
CLOSE TRACK/SESSION	0x5B	R
FORMAT UNIT	0x04	E
INQUIRY	0x12	M
LOAD/UNLOAD CD	0xA6	C
MECHANISM STATUS	0xBD	M
MODE SELECT (10)	0x55	M
MODE SENSE (10)	0x5A	M
PAUSE/RESUME	0x4B	A
PLAY AUDIO (10)	0x45	A
PLAY AUDIO (12)	0xA5	A
PLAY AUDIO MSF	0x47	A
PLAY CD	0xBC	O
PREVENT/ALLOW MEDIUM REMOVAL	0x1E	M
READ (10)	0x28	M
READ (12)	0xA8	M
READ BUFFER CAPACITY	0x5C	O
READ CD	0xBE	M
READ CD MSF	0xB9	M
READ CD RECORDED CAPACITY	0x25	M
READ DISC INFORMATION	0x51	R
READ HEADER	0x44	M
READ MASTER CUE	0x59	O
READ SUB-CHANNEL	0x42	M
READ TOC/PMA/ATIP	0x43	M
READ TRACK INFORMATION	0x52	R
REPAIR TRACK	0x58	O
REQUEST SENSE	0x03	M
RESERVE TRACK	0x53	R
SCAN	0xBA	A
SEEK	0x2B	M
SEND CUE SHEET	0x5D	O
SEND OPC INFORMATION	0x54	O
SET CD SPEED	0xBB	R
START/STOP UNIT	0x1B	M
STOP PLAY/SCAN	0x4E	M
SYNCHRONIZE CACHE	0x35	R
TEST UNIT READY	0x00	M
WRITE (10)	0x2A	R
WRITE (12)	0xAA	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		

8.7.5. Comanda PLAY AUDIO MSF

Comanda PLAY AUDIO MSF solicită unei unități care conține un disc CD audio să înceapă o operație de redare audio. Aceasta este o comandă imediată, permițând suprapunerea comenzilor. La terminarea efectivă a operației, unitatea setează bitul SERV (bitul 4) din registrul de stare. Protocolul utilizat este protocolul ATAPI pentru comenzile fără transferuri de date.

Structura pachetului de comandă este ilustrată în figura 8.15.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x47)							
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	Cod de control							
10..11	Rezervați							

Figura 8.15. Structura pachetului comenzii PLAY AUDIO MSF.

Câmpurile M, S și F de început specifică adresa MSF absolută de la care trebuie să înceapă operația, iar câmpurile M, S și F de sfârșit specifică adresa MSF absolută unde operația trebuie terminată. Dacă valorile câmpurilor M, S și F de început sunt setate la 0xFF, adresa de început va fi poziția capului optic. 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 nicio operație. Dacă adresa MSF de început este mai mare decât adresa MSF de sfârșit, comanda se va termina cu starea *CHECK CONDITION*, iar bitul ERR/CHK din registrul de stare va fi setat.

8.7.6. Comanda READ (12)

Comanda READ (12) solicită unității CD/DVD transferul unor date de la unitate la calculatorul gazdă. Pachetul de comandă are o structură tipică unui pachet de comandă de 12 octeți (figura 8.14). Octeții 6..9 conțin lungimea transferului. Protocolul utilizat este protocolul ATAPI pentru intrare în modul PIO.

Lungimea transferului specifică numărul blocurilor logice contigue care trebuie transferate. Deși unitatea CD/DVD poate returna diferite tipuri de 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*, bitul 0) din registrul de eroare *Error* dacă se încearcă citirea acestora prin comanda READ.

8.7.7. Comanda READ CD RECORDED CAPACITY

Comanda READ CD RECORDED CAPACITY permite calculatorului gazdă solicitarea unor informații despre capacitatea înregistrată a unui disc CD/DVD. Protocolul utilizat pentru această comandă este protocolul ATAPI pentru intrare în modul PIO. Comanda returnează adresa ultimului bloc logic, pe baza datelor tabelului de conținut. Dacă ultima pistă este o pistă audio, valoarea returnată poate fi inexactă, deoarece există 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/DVD, această implementare permite un răspuns mai rapid.

Structura pachetului de comandă este ilustrată în figura 8.16. Câmpurile Rezervat și Cod de control trebuie setate la 0.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x25)							
1	Rezervat							
...								
8								
9	Cod de control							

Figura 8.16. Structura pachetului comenzii READ CD RECORDED CAPACITY.

Datele returnate de această comandă au structura din figura 8.17. Comanda raportează o lungime a blocului de 2048 de octeți.

Bit Octet	7	6	5	4	3	2	1	0
0..3	Adresa ultimului bloc logic (octet 0 - ms)							
4..7	Lungimea blocului în octeți (octet 4 - ms)							

Figura 8.17. Formatul datelor returnate de comanda READ CD RECORDED CAPACITY.

8.7.8. Comanda READ TOC/PMA/ATIP

Comanda READ TOC/PMA/ATIP solicită unității CD/DVD să transfere date din tabela de conținut (TOC), zona de memorie a programului PMA (*Program Memory Area*) sau zona ATIP (*Absolute Time in Pre-Groove*). PMA este o zonă suplimentară prezentă la discurile CD-R/DVD-R și CD-RW/DVD-RW, care conține numerele pistelor pentru titlurile înregistrate și pozițiile lor de început și de sfârșit. Zona ATIP conține informații ca timpul absolut de început pentru zonele *lead-in* și *lead-out* în format MSF, puterea recomandată a laserului pentru scriere, sau viteza de înregistrare minimă și maximă care se poate utiliza. Structura pachetului de comandă este ilustrată în figura 8.18. Protocolul utilizat pentru această comandă este protocolul ATAPI pentru intrare în modul PIO.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x43)							
1	Rezervat						MSF	Rezerv.
2	Rezervat				Format			
3..5	Rezervat							
6	Număr pistă/sesiune (Hexa)							
7	Octet ms				Lungime alocată			
8					(octet 7 - ms)			
9					Cod de control			
10..11	Rezervat							
					Octet mps			

Figura 8.18. Structura pachetului comenzii READ TOC/PMA/ATIP.

Lungimea alocată indică numărul maxim de octeți care trebuie returnați de această comandă. Pentru a specifica tipul datelor care trebuie returnate, se utilizează cei patru biți mai puțin semnificativi ai octetului 2 (Format). Pentru discuri multi-sesiune și/sau discuri *Kodak Photo CD*, se poate utiliza formatul 0001b. Pentru unități care nu permit discuri multi-sesiune, numărul primei sesiuni trebuie să fie egal cu numărul ultimei sesiuni din informațiile TOC returnate. Definiția câmpului Format este următoarea:

0000b Pentru acest format, câmpul Număr pistă/sesiune specifică numărul primei piste pentru care trebuie returnate datele. Dacă această valoare este 0, tabela de conținut va începe cu datele primei piste. Pentru discurile multi-sesiune, comanda va returna datele tabelii de conținut pentru toate sesiunile. Datele sunt returnate în ordinea crescătoare a numerelor de pistă.

- 0001b Acest format este destinat discurilor multi-sesiune. Formatul returnează numărul primei sesiuni complete, numărul ultimei sesiuni complete și adresa de început a ultimei sesiuni complete. În acest format, câmpul Număr pistă/sesiune trebuie să fie setat la 0x00.
- 0010b Specifică returnarea datelor sub-canalului Q din zona *lead-in* (TOC) începând cu numărul sesiunii specificate în câmpul Număr pistă/sesiune.
- 0011b Specifică returnarea datelor sub-canalului Q din zona PMA. Câmpul Număr pistă/sesiune trebuie să fie setat la 0x00.
- 0100b Specifică returnarea datelor ATIP. Câmpul Număr pistă/sesiune trebuie să fie setat la 0x00.

Datele returnate pentru formatul 0000b au structura din figura 8.19. Blocul de date returnat conține un antet de patru octeți urmat de zero sau mai mulți descriptori de pistă.

Bit Octet	7	6	5	4	3	2	1	0
0..1	Lungime date TOC (octet 0 - ms)							
2	Numărul primei piste (hexa)							
3	Numărul ultimei piste (hexa)							
Descriptor(i) de pistă								
0	Rezervat							
1	ADR				CONTROL			
2	Număr pistă (hexa)							
3	Rezervat							
4..7	Adresa logică a blocului (octet 4 - ms)							

Figura 8.19. Datele returnate de comanda READ TOC/PMA/ATIP pentru formatul 0000b.

Lungime date TOC specifică lungimea în octeți a datelor TOC care urmează. Această lungime nu include câmpul de lungime a datelor TOC. Câmpul Numărul primei piste conține numărul primei piste din tabela de conținut a primei sesiuni complete. Câmpul Numărul ultimei piste conține numărul ultimei piste din tabela de conținut a ultimei sesiuni complete (înainte de zona *lead-out*). Numerele valide ale pistelor sunt cuprinse între 01 și 99 (0x63). Prima pistă poate avea orice număr valid. Câmpul ADR indică tipul informațiilor codificate în sub-canalul Q al blocului în care s-a găsit această intrare în tabela de conținut:

- 0x0: Nu sunt furnizate informații despre codificarea informațiilor din sub-canalul Q.
- 0x1: Sub-canalul Q codifică informații despre poziția curentă (pistă, index, adresa absolută, adresa relativă).
- 0x2: Sub-canalul Q codifică numărul de catalog al suportului.
- 0x3: Sub-canalul Q reprezintă codul ISRC (*International Standard Recording Code*).
- 0x4-0xF: Valori rezervate.

Câmpul CONTROL indică atributele pistei (date sau audio, copierea permisă sau interzisă). Câmpul Număr pistă indică numărul pistei pentru care datele din descriptorul de pistă sunt valide. Un număr de pistă egal cu 0xAA arată că descriptorul de pistă se referă la zona *lead-out*. Adresa logică a blocului conține adresa primului bloc cu informații pentru pista respectivă. Dacă bitul MSF din pachetul comenzii este 0, câmpul Adresa logică a blocului conține o adresă logică, iar dacă bitul MSF este 1 acest câmp conține o adresă MSF. Câmpurile M, S și F ale acestei adrese se află în octeții 1, 2, respectiv 3 ai adresei absolute (octetul 0 este rezervat).

Datele returnate pentru formatul 0001b au structura din figura 8.20.

Bit Octet	7	6	5	4	3	2	1	0
0..1	Lungime date TOC (octet 0 - ms)							
2	Numărul primei sesiuni complete (hexa)							
3	Numărul ultimei sesiuni complete (hexa)							
Descriptor de pistă								
0	Rezervat							
1	ADR				CONTROL			
2	Numărul primei piste din ultima sesiune completă (hexa)							
3	Rezervat							
4..7	Adresa logică a blocului pentru prima pistă din ultima sesiune (octet 4 - ms)							

Figura 8.20. Datele returnate de comanda READ TOC/PMA/ATIP pentru formatul 0001b.

Numărul primei sesiuni complete este setat la 1. Pentru discurile cu o singură sesiune, sau dacă unitatea nu recunoaște discuri multi-sesiune, numărul ultimei sesiuni complete este setat la 1.

Pentru formatul 0010b, datele returnate au structura din figura 8.21.

Bit Octet	7	6	5	4	3	2	1	0
0..1	Lungime date TOC (octet 0 - ms)							
2	Numărul primei sesiuni complete (hexa)							
3	Numărul ultimei sesiuni complete (hexa)							
Descriptor(i) de pistă								
0	Număr sesiune (hexa)							
1	ADR				CONTROL			
2	TNO							
3	POINT							
4	MIN							
5	SEC							
6	FRAME							
7	ZERO							
8	PMIN							
9	PSEC							
10	PFRAME							

Figura 8.21. Datele returnate de comanda READ TOC/PMA/ATIP pentru formatul 0010b.

Semnificația octeților 2..10 este aceeași cu cea descrisă în paragraful 8.2.5 pentru câmpul DATA-Q al sub-canalului Q. Datele returnate pentru discurile multi-sesiune sunt aranjate în ordinea crescătoare a numerelor de sesiune. În cadrul unei sesiuni, datele sunt aranjate în următoarea ordine a valorii câmpului POINT: 0xA0, 0xA1, 0xA2, numerele pistelor, 0xB0, 0xB1, 0xB2, 0xB3, 0xB4, 0xC0 și 0xC1.

8.7.9. Comanda START/STOP UNIT

Comanda START/STOP UNIT permite validarea sau invalidarea accesului la suport de către unitatea CD/DVD și plasarea unității într-un mod cu un consum redus de energie. Protocolul utilizat este protocolul ATAPI pentru comenzile fără transferuri de date. Structura pachetului de comandă este ilustrată în figura 8.22.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x1B)							
1	Rezervat							IMED
2..3	Rezervat							
4	Mod				Rezervat		LOEJ	START
5	Cod de control							
6..11	Rezervat							

Figura 8.22. Structura pachetului comenzii START/STOP UNIT.

Dacă bitul IMED este 1, se va returna starea imediat ce pachetul de comandă a fost transmis și validat. Dacă bitul IMED este 0, starea este returnată după terminarea operației. Câmpul Mod permite selectarea unui mod de funcționare cu un consum redus de energie. Dacă acest câmp conține o valoare diferită de 0x0, biții LOEJ și START vor fi ignorați. Tabelul 8.7 indică unele valori uzuale ale câmpului Mod.

Tabelul 8.7. Valori uzuale ale câmpului Mod din pachetul comenzii START/STOP UNIT.

Conținut Mod	Semnificație
0x0	Modul de funcționare rămâne neschimbat
0x1	Selectează modul activ
0x2	Selectează modul inactiv
0x3	Selectează modul "Standby"
0x5	Selectează modul "Sleep"

Dacă bitul START este 1, se validează utilizarea unității, iar dacă bitul START este 0, se invalidează utilizarea unității (suportul nu poate fi accesat de calculatorul gazdă).

Dacă bitul LOEJ (*Load / Eject*) este 0, nu vor fi executate acțiuni pentru încărcarea sau eliminarea suportului. Dacă bitul LOEJ este 1, comanda solicită eliminarea suportului dacă bitul START este 0 sau încărcarea suportului dacă bitul START este 1.

8.7.10. Comanda STOP PLAY/SCAN

Comanda STOP PLAY/SCAN oprește execuția comenzilor pentru discurile CD audio. Protocolul utilizat este protocolul ATAPI pentru comenzile fără transferuri de date. Structura pachetului de comandă este ilustrată în figura 8.23.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x4E)							
1..8	Rezervat							
9	Cod de control							
10..11	Rezervat							

Figura 8.23. Structura pachetului comenzii STOP PLAY/SCAN.

8.8. Aplicații

8.8.1. Răspundeți la următoarele întrebări:

- Care sunt etapele necesare pentru înregistrarea datelor pe discurile compact?
- Cum se realizează corecția erorilor în interiorul cadrelor?
- Cum este posibilă citirea simultană a datelor și a informațiilor audio/video de pe discurile CD-I și CD-ROM/XA?
- Ce reprezintă sub-canalele discurilor compact și la ce se utilizează acestea?

8.8.2. Deschideți proiectul creat pentru lucrarea de laborator *Interfața ATA* și extindeți acest proiect prin scrierea unei funcții care transmite comanda IDENTIFY PACKET DEVICE unei unități ATAPI. Parametrii de intrare ai funcției sunt adresa de bază (de tip WORD) a registrelor din blocul de comandă pentru canalul ATA la care este conectată unitatea și numărul unității (0 sau 1). Funcția nu returnează nicio valoare. Această comandă este descrisă în secțiunea 8.7.2; protocolul utilizat pentru această comandă este același cu cel utilizat pentru comanda ATA IDENTIFY DEVICE. În cazul în care comanda se termină cu succes, funcția afișează următoarele informații despre unitatea ATAPI: numărul modelului, numărul de serie, revizia *firmware* și lungimea pachetelor de comandă ATAPI.

După scrierea funcției, includeți un apel al acestei funcții în funcția `AppScroll()`, utilizând ca parametri adresa de bază a registrelor din blocul de comandă pentru canalul ATA

primar și numărul unității 1. Repetați apelul funcției pentru canalul ATA secundar cu numărul unității 1.

8.8.3. Deschideți proiectul creat pentru lucrarea de laborator *Interfața ATA* și extindeți acest proiect prin scrierea unei funcții care transmite comanda START/STOP UNIT unei unități ATAPI pentru a deschide ușa unității. Parametrii de intrare ai funcției sunt următorii: adresa de bază (de tip WORD) a registrelor din blocul de comandă pentru canalul ATA la care este conectată unitatea; adresa de bază (de tip WORD) a registrelor din blocul de control pentru canalul ATA la care este conectată unitatea; numărul unității (0 sau 1). Funcția nu returnează nicio valoare. Această comandă este descrisă în secțiunea 8.7.9. Mai întâi, definiți pachetul de comandă ca un tablou de 12 octeți și inițializați fiecare octet cu 0 utilizând funcția `memset()`. În continuare, inițializați octetul 0 al pachetului de comandă cu codul comenzii SCSI START/STOP UNIT. În octetul 4 al pachetului de comandă, setați bitul LOEJ la 1 și ceilalți biți la 0 (figura 8.22). Apoi, implementați protocolul ATAPI fără transferuri de date, descris în secțiunea 8.6.1.

După scrierea funcției, includeți un apel al acestei funcții în funcția `AppScroll()`, utilizând ca parametri adresele de bază ale registrelor din blocul de comandă și ale registrelor din blocul de control pentru canalul ATA primar și numărul unității 1. Repetați apelul funcției pentru canalul ATA secundar cu numărul unității 1.

8.8.4. Continuați aplicația 8.8.2 prin scrierea unei funcții care transmite comanda READ CD RECORDED CAPACITY unei unități ATAPI. Parametrii de intrare ai funcției sunt următorii: adresa de bază (de tip WORD) a registrelor din blocul de comandă pentru canalul ATA la care este conectată unitatea; adresa de bază (de tip WORD) a registrelor din blocul de control pentru canalul ATA la care este conectată unitatea; numărul unității (0 sau 1). Funcția nu returnează nicio valoare. Această comandă este descrisă în secțiunea 8.7.7. Mai întâi, definiți pachetul de comandă ca un tablou de 12 octeți și datele returnate de comandă ca un tablou de 8 octeți. Inițializați fiecare octet al pachetului de comandă cu 0 utilizând funcția `memset()`. În continuare, inițializați octetul 0 al pachetului de comandă cu codul comenzii SCSI READ CD RECORDED CAPACITY. Apoi, implementați protocolul ATAPI pentru intrare în modul PIO, descris în secțiunea 8.6.2. În cazul în care comanda se termină cu succes, definiți variabile de tip `DWORD` pentru adresa ultimului bloc logic și pentru lungimea blocului. Inițializați aceste variabile cu câmpurile corespunzătoare ale datelor returnate de comandă, ținând cont de ordinea corectă a octeților (adresa ultimului bloc logic trebuie incrementată pentru a obține numărul blocurilor logice). Afișați adresa ultimului bloc logic, lungimea blocului și capacitatea înregistrată a discului în MB.

După scrierea funcției, includeți un apel al acestei funcții în funcția `AppScroll()`, utilizând ca parametri adresele de bază ale registrelor din blocul de comandă și ale registrelor din blocul de control pentru canalul ATA primar și numărul unității 1. Repetați apelul funcției pentru canalul ATA secundar cu numărul unității 1. Inserați un disc CD sau DVD de date în unitate și verificați funcția comparând capacitatea înregistrată afișată cu cea indicată de sistemul de operare.

8.8.5. Continuați aplicația 8.8.4 prin scrierea unei funcții care transmite comanda READ TOC/PMA/ATIP unei unități ATAPI pentru citirea tabelului de conținut a unui disc CD audio. Parametrii de intrare ai funcției sunt aceiași ca și parametrii funcției scrise pentru aplicația 8.8.4. Funcția nu returnează nicio valoare. Această comandă este descrisă în secțiunea 8.7.8. Mai întâi, definiți pachetul de comandă ca un tablou de 12 octeți și datele returnate de comandă ca un tablou de 256 octeți. Inițializați fiecare octet al pachetului de comandă cu 0 utilizând funcția `memset()`. În continuare, inițializați octetul 0 al pachetului de comandă cu codul comenzii SCSI READ TOC/PMA/ATIP. În octetul 1 al pachetului de comandă, setați bitul MSF la 1 și ceilalți biți la 0 (figura 8.18). Specificați formatul 0000b pentru datele care trebuie returnate de unitate lăsând nemodificat octetul 2 al pachetului de comandă (octeții pachetului de comandă au fost inițializați cu 0). Inițializați câmpul pentru lungimea alocată al pachetului de comandă cu dimensiunea bufferului alocat pentru datele returnate de comandă (octetul 7 trebuie inițializat cu octetul superior al dimensiunii, iar octetul 8 cu octetul inferior

al dimensiunii). Apoi, implementați protocolul ATAPI pentru intrare în modul PIO, descris în secțiunea 8.6.2. În cazul în care comanda se termină cu succes, afișați numărul primei piste și numărul ultimei piste a discului CD audio (figura 8.19). Apoi, pentru fiecare pistă audio, afișați minutul, secunda și cadrul de început al pistei (câmpurile M, S și F ale adresei logice a blocului).

După scrierea funcției, includeți un apel al acestei funcții în funcția `AppScroll()`, utilizând ca parametri adresele de bază ale registrelor din blocul de comandă și ale registrelor din blocul de control pentru canalul ATA primar și numărul unității 1. Repetați apelul funcției pentru canalul ATA secundar cu numărul unității 1. Inserați un disc CD audio în unitate și verificați dacă informațiile afișate sunt corecte pe baza lungimilor pistelor inscripționate pe disc sau pe cutia discului.

Bibliografie

- [1] American National Standards Institute, Inc., “Information Technology - AT Attachment 8 - ATA/ATAPI Architecture Model (ATA8-AAM)”, T13/1700-D Revision 3, 2006.
- [2] American National Standards Institute, Inc., “Information Technology - AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)”, T13/1699-D Revision 4a, 2007.
- [3] American National Standards Institute, Inc., “Information Technology - AT Attachment 8 - ATA/ATAPI Parallel Transport (ATA8-APT)”, T13/1698-D Revision 2, 2007.
- [4] American National Standards Institute, Inc., “Information Technology - AT Attachment 8 - ATA/ATAPI Serial Transport (ATA8-AST)”, T13/1697-D Revision 1, 2007.
- [5] Intel Corporation, “Intel 8 Series/C220 Series Chipset Family Platform Controller Hub (PCH)”, Datasheet, May 2014.
- [6] InterNational Committee for Information Technology Standards, “Information Technology – MultiMedia Command Set – 6 (MMC-6)”, Project T10/BSR INCITS 468, Revision 02g, December 11, 2009.
- [7] InterNational Committee for Information Technology Standards, “Information Technology – SCSI Block Commands – 4 (SBC-4)”, Project T10/BSR INCITS 506, Revision 22, September 15, 2020.
- [8] InterNational Committee for Information Technology Standards, “Information Technology – SCSI Primary Commands – 6 (SPC-6)”, Project T10/BSR INCITS 566, Revision 5, March 8, 2021.
- [9] Rosch, W. L., *Hardware Bible*, Sixth Edition, Que Publishing, 2003.