

## 6. INTERFAȚA SCSI

Această lucrare de laborator prezintă diferite tipuri de interfețe electrice SCSI (*Small Computer System Interface*), standardele SCSI care au fost elaborate, magistrala SCSI, structura comenzilor SCSI, configurarea echipamentelor SCSI și una din interfețele de programare SCSI.

### 6.1. Prezentarea interfeței SCSI

Interfața SCSI (*Small Computer System Interface*) provine din interfața SASI (*Shugart Associates System Interface*), care a fost elaborată de firma producătoare de unități de discuri magnetice *Shugart Associates* și a fost destinată conectării unităților de discuri la calculator. Această interfață utilizează adrese logice în locul adreselor fizice și comenzi de câte 6 octeți. Comitetul de lucru *X3T9* al institutului american de standarde *ANSI* a utilizat specificațiile SASI ca bază pentru elaborarea unui standard al unei interfețe paralele, standard care a fost numit ulterior SCSI-1 și a fost publicat în anul 1986. Același comitet *X3T9* a elaborat standardul SCSI-2, care a fost publicat în forma sa finală în anul 1994. În anul 1993, un alt comitet de lucru, *T10*, a început activitatea pentru elaborarea versiunii SCSI-3 a standardului. Documentele acestei versiuni au fost publicate separat, pe parcursul mai multor ani, începând din anul 1996. Aceste documente continuă să fie actualizate și în prezent prin versiuni noi.

SCSI nu este o interfață de disc, deci un anumit tip de controler, ci o interfață de sistem formată dintr-o magistrală la care se pot conecta mai multe echipamente. Unul din aceste echipamente, adaptorul sistemului gazdă, funcționează ca o punte între magistrala SCSI și magistrala sistemului. Magistrala SCSI nu comunică direct cu echipamentele periferice, cum sunt unitățile de discuri, ci cu controlerul care este inclus în aceste unități.

O singură magistrală SCSI poate accepta până la 8 sau 16 unități fizice, numite unități SCSI, dintre care una este adaptorul SCSI. Unitățile fizice pot fi unități de discuri magnetice, unități optice, scanere, imprimante. Majoritatea sistemelor pot accepta până la patru adaptoare SCSI.

Una din cauzele care au întârziat acceptarea interfeței SCSI pe piața calculatoarelor personale a fost lipsa unui standard pentru adaptorul sistemului gazdă, driverele de interfață și programul BIOS. Din cauza lipsei unui standard de interfață, au apărut mai multe probleme, ca imposibilitatea de a utiliza unitățile de discuri în afara magistralei SCSI, imposibilitatea de a încărca sistemul de operare de pe aceste unități și de a folosi unele sisteme de operare. Aceste probleme au fost rezolvate prin elaborarea standardelor SCSI.

SCSI este o interfață destinată în special stațiilor de lucru și serverelor cu performanțe ridicate. La prima versiune a interfeței SCSI, frecvența de ceas a magistralei era de 5 MHz, iar rata maximă de transfer era de 5 MB/s. La versiunile ulterioare ale acestei interfețe, frecvența de ceas a magistralei a fost de 80 MHz sau 160 MHz, iar ratele maxime de transfer de 320 MB/s, respectiv 640 MB/s.

Pe lângă aceste versiuni paralele ale interfeței SCSI, a fost elaborată și o versiune serială a interfeței. Această interfață SCSI serială, numită *Serial Attached SCSI*, a înlocuit treptat interfața SCSI paralelă. Atât tehnologia SCSI paralelă, cât și cea serială, sunt promovate de organizația SCSI *Trade Association* ([www.scsita.org](http://www.scsita.org)).

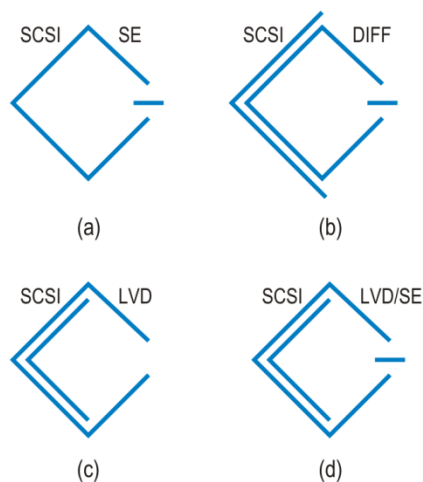
## 6.2. Tipuri de interfețe electrice SCSI

Există două tipuri principale de interfețe electrice SCSI: normale și diferențiale. În cazul unei interfețe SCSI *normale* (*Single Ended – SE*), pentru fiecare semnal transmis pe magistrală există un singur fir. Circuitele receptoare de la celălalt capăt al cablului detectează tensiunile electrice de pe liniile magistralei. Costul unei asemenea interfețe este redus, dar semnalele transmise sunt afectate de zgomote și de interferențe electromagnetice.

La interfețele SCSI *diferențiale*, pentru fiecare semnal există câte o pereche de fire. Pe unul din fire se transmite un semnal de același tip cu cel transmis de interfețele SCSI normale. Pe al doilea fir se transmite un semnal care este obținut prin inversarea logică a semnalului original. Circuitele de recepție nu trebuie să detecteze valoarea absolută a tensiunii, ci numai diferența dintre semnalele care sunt recepționate pe cele două fire. Interfețele diferențiale au mai multe avantaje: imunitatea crescută la zgomote, posibilitatea unor transferuri cu viteze mai ridicate și o lungime mai mare a cablului. La interfețele SCSI normale, lungimea cablului poate fi de până la 6 m pentru frecvențe reduse ale magistralei SCSI, sau de până la 3 m pentru frecvențe mai ridicate. La interfețele SCSI diferențiale, lungimea cablului poate fi de până la 25 m sau 12 m, în funcție de frecvența magistralei.

La primele interfețe SCSI diferențiale, denumite *High Voltage Differential (HVD)*, pentru transmiterea semnalelor se utilizau tensiuni relativ ridicate. Din această cauză, nu se puteau realiza circuite de interfață cu putere redusă și de dimensiuni mici. Pentru implementarea interfeței erau necesare circuite care utilizau mai multe capsule, ceea ce creștea costul interfeței. O altă problemă a acestei interfețe diferențiale era faptul că echipamentele cu interfață diferențială nu se puteau conecta la aceeași magistrală la care erau conectate echipamente cu interfață normală, deoarece tensiunile mai ridicate ale interfeței diferențiale puteau distruge circuitele de recepție ale echipamentelor cu interfață normală. Din cauza costului ridicat și a incompatibilității cu echipamentele având interfață normală, interfața diferențială HVD a fost utilizată foarte rar, iar specificațiile acesteia au fost eliminate din ultima versiune a standardului SCSI.

În locul interfeței electrice HVD, a fost elaborată o nouă interfață diferențială, denumită *LVD (Low Voltage Differential)*, care utilizează tensiuni reduse. Această interfață poate fi implementată cu circuite de putere redusă și costuri mici. Un alt avantaj al acestei interfețe este că permite conectarea echipamentelor LVD la o magistrală SCSI normală SE, fără riscul defectării circuitelor de interfață ale echipamentelor. În mod opțional, echipamentele cu interfață LVD pot fi proiectate astfel încât să se poată conecta la ambele magistrale LVD și SE. Aceste echipamente multi-mod detectează dacă sunt conectate la o magistrală SE și vor funcționa în modul SE compatibil cu această magistrală. Atunci când există chiar și un singur echipament SE conectat la o magistrală SCSI, toate echipamentele LVD conectate la aceeași magistrală vor funcționa în modul SE, la o frecvență redusă a magistralei (până la 20 MHz).



**Figura 6.1.** Simboluri pentru interfețele electrice SCSI: (a) interfața normală SE; (b) interfața diferențială HVD; (c) interfața diferențială LVD; (d) interfața multi-mod LVD/SE.

Deoarece conectorii SCSI sunt aceiași pentru diferite tipuri ale interfeței electrice, iar conectarea unui echipament cu interfață HVD la o magistrală cu echipamente SE sau LVD poate cauza defectarea acestor echipamente, este necesară diferențierea între diferitele tipuri de interfețe. Echipamentele cu interfață normală se pot deosebi de cele cu interfață diferențială HVD sau LVD prin simbolurile marcate pe acestea. În industrie au fost adoptate simboluri diferite pentru interfețele SCSI normale și cele diferențiale (figura 6.1).

### 6.3. Standarde SCSI

Standardele interfeței SCSI au fost elaborate de către mai multe grupuri de lucru din cadrul institutului *ANSI*, care au funcționat sau funcționează ca și comitete acreditate pentru standardizare. Standardele SCSI definesc parametri fizici și electrici ai unei magistrale de I/E, care este utilizată pentru conectarea în lanț a echipamentelor periferice.

#### 6.3.1. SCSI-1

Primul standard al interfeței SCSI, SCSI-1 (*ANSI X3.131-1986*), a fost aprobat de institutul *ANSI* în anul 1986. Încă înaintea aprobării acestui standard, producătorii unităților de discuri erau îngrijorați de faptul că multe din comenzile și caracteristicile specificate de standard erau opționale. De aceea, nu exista nici o garanție că un anumit periferic va recunoaște toate aceste comenzi. Pentru a rezolva această problemă, industria de profil a solicitat comitetului *X3T9* care a elaborat specificațiile SCSI-1 extinderea comenzilor și caracteristicilor obligatorii ale acestor specificații. Deoarece specificațiile erau aproape terminate, institutul *ANSI* a aprobat standardul în forma existentă și a constituit un nou grup de lucru pentru elaborarea unui set de 18 comenzi SCSI de bază. Acest set a fost numit *CCS (Common Command Set)* și va deveni setul minim de comenzi recunoscute de toate perifericele. Acest set de comenzi a stat la baza standardului SCSI-2.

#### 6.3.2. SCSI-2

Standardul SCSI-2 este o versiune îmbunătățită a standardului SCSI-1, la care au fost adăugate caracteristici și opțiuni noi. În mod normal, echipamentele SCSI-1 și cele SCSI-2 sunt compatibile, dar echipamentele conforme cu standardul SCSI-1 nu recunosc opțiunile suplimentare introduse de standardul SCSI-2.

Comitetul de lucru *X3T9* a finalizat standardul SCSI-2 în anul 1990, dar documentul a fost retras la sfârșitul aceluiași an pentru unele modificări, care urmau a fi efectuate înainte de publicarea definitivă de către institutul *ANSI*. Versiunea finală a fost aprobată doar la începutul anului 1994, deși acest document (*ANSI X3.131-1994*) conține foarte puține modificări față de versiunea inițială din 1990. Practic, toate prevederile standardului SCSI-1 se pot regăsi și în standardul SCSI-2.

În plus față de acceptarea setului de 18 comenzi SCSI de bază, standardul SCSI-2 conține și specificații noi, cum sunt comenzi noi pentru unități CD-ROM (inclusiv posibilitatea utilizării discurilor CD audio), unități de bandă și alte periferice. De asemenea, a fost definită o versiune mai rapidă a interfeței, numită *Fast SCSI-2*, și o versiune pe 16 biți, numită *Wide SCSI-2*. O altă îmbunătățire definită de standardul SCSI-2 este posibilitatea de a scrie comenzile într-o coadă de comenzi, ceea ce permite unui periferic să accepte mai multe comenzi și să le execute într-o ordine care este considerată cea mai eficientă. Această posibilitate este importantă pentru sistemele de operare *multitasking*.

Unele din modificările specificate de standardul SCSI-2 sunt minore. De exemplu, în standardul SCSI-1 paritatea pe magistrala SCSI era opțională, pe când în standardul SCSI-2 este obligatorie utilizarea unui bit de paritate. O altă cerință este aceea că dispozitivele care inițiază transferul, cum sunt adaptoarele sistemelor gazdă, trebuie să alimenteze circuitele terminatoare ale interfeței. Însă, majoritatea echipamentelor îndeplineau deja această cerință.

Standardul SCSI-2 definește următoarele caracteristici opționale:

- *Fast SCSI*;
- *Wide SCSI*;

- Coada de comenzi;
- Comenzi noi;
- Terminatoare îmbunătățite.

*Fast SCSI* se referă la posibilitatea de a efectua transferuri sincrone la viteze mai ridicate. Cu această versiune se pot obține rate de transfer de 10 MB/s pe magistrala SCSI standard, de 8 biți. Atunci când se combină cu o interfață *Wide SCSI* de 16 biți, această configurație permite rate de transfer de 20 MB/s.

*Wide SCSI* permite transferuri de date pe magistrale de 16 sau 32 de biți. Pentru aceste variante sunt necesare alte cabluri decât pentru varianta de 8 biți. Cablul standard cu 50 de fire utilizat pentru transferuri de 8 biți se numește cablu *A*. Pentru varianta *Wide SCSI* de 16 biți este necesar un cablu *P* cu 68 de fire. Pentru varianta *Wide SCSI* de 32 de biți, care nu a fost implementată niciodată, sunt necesare două cabluri: cablul *P* cu 68 de fire și cablul *Q* cu 68 de fire.

Conform standardului SCSI-1, un echipament inițiator, cum este adaptorul calculatorului gazdă, poate transmite câte o singură comandă unui periferic. Conform standardului SCSI-2, adaptorul calculatorului gazdă poate transmite până la 256 de comenzi unui singur periferic, care va memora comenzile într-o coadă de comenzi și le va executa înainte de a se transmite un răspuns pe magistrala SCSI. Perifericul poate chiar modifica ordinea comenzilor pentru a permite execuția cea mai eficientă posibilă.

Comenzile din setul de comenzi comune (*CCS*), care erau utilizate deja în industrie, au fost incluse oficial în standardul SCSI-2. Setul de comenzi comune a fost definit mai ales pentru unitățile de discuri și nu includea comenzi specifice pentru alte tipuri de echipamente. Au fost modificate multe din comenzile mai vechi și au fost adăugate comenzi noi. De exemplu, au fost adăugate noi seturi de comenzi pentru unitățile CD-ROM, alte discuri optice, scannere, echipamente de comunicație etc.

Pentru funcționarea corectă a magistralei SCSI cu interfață normală, sunt necesare rezistențe terminatoare cu toleranțe strânse. Terminatoarele pasive de 132 Ω, definite în standardul SCSI-1, nu sunt adecvate pentru vitezele mai mari de transfer permise de standardul SCSI-2. Aceste terminatoare pasive pot cauza reflexii de semnal și pot apare erori atunci când ratele de transfer cresc sau sunt conectate mai multe echipamente la magistrală. Conform standardului SCSI-2, trebuie folosite componente active ca și terminatori, care asigură o impedanță de 110 Ω și îmbunătățesc integritatea sistemului.

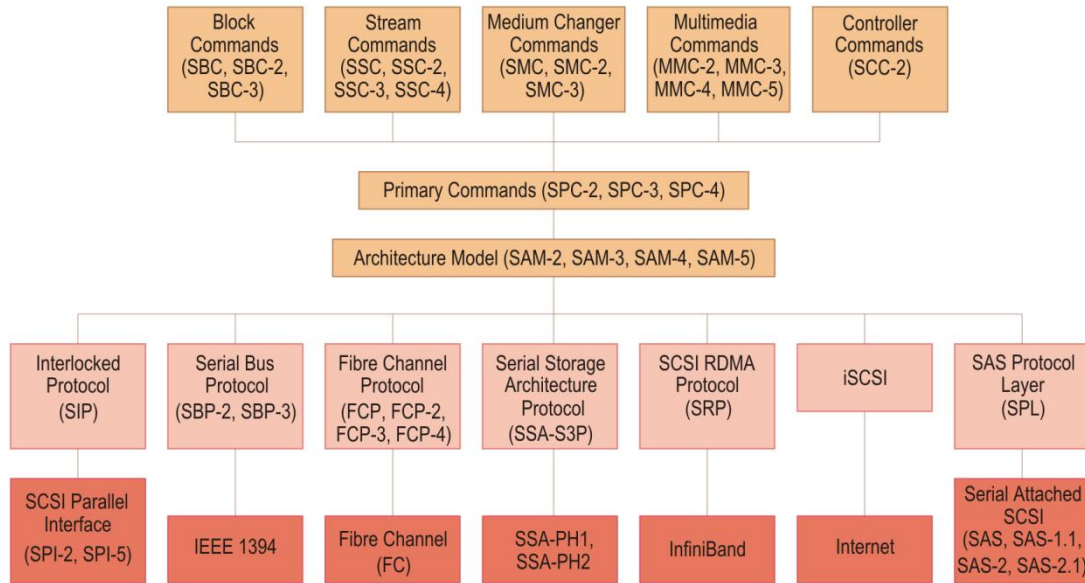
### 6.3.3. SCSI-3

SCSI-3 reprezintă o colecție de standarde care au fost elaborate și publicate separat. Aceste standarde au fost împărțite în categorii care cuprind: standarde pentru comenzile primare, standarde pentru comenzile care sunt specifice diferitelor clase de echipamente, standarde pentru protocoalele de comunicație și standarde pentru interconexiunile fizice. În plus, există un model arhitectural SCSI (*SAM – SCSI Architectural Model*) pentru interfețele fizice și electrice. Standardele SCSI-3 au fost elaborate și sunt actualizate de comitetul tehnic *T10* din cadrul comitetului internațional de standarde în tehnologia informației *INCITS (International Committee on Information Technology Standards)*, care este acreditat de institutul *ANSI*. Documentele de lucru ale comitetului *T10* sunt disponibile la adresa [www.t10.org](http://www.t10.org).

Principalele îmbunătățiri introduse de standardul SCSI-3 cuprind următoarele:

- Ultra2 (Fast-40) SCSI;
- Ultra3 (Fast-80DT) SCSI;
- Ultra4 (Fast-160DT) SCSI;
- Ultra5 (Fast-320DT) SCSI;
- Noua interfață electrică *Low Voltage Differential*;
- Eliminarea interfeței electrice *High Voltage Differential*.

Figura 6.2 prezintă principalele componente ale colecției de standarde SCSI-3. Majoritatea standardelor individuale au mai multe versiuni, care sunt indicate în figură.



**Figura 6.2.** Principalele componente ale colecției de standarde SCSI-3.

Componentele principale ale familiei de standarde SCSI-3 sunt descrise în continuare. SCSI *Architecture Model* (*SAM*) definește modelul sistemelor SCSI, partiționarea funcțională a setului de standarde SCSI-3 și cerințele aplicabile pentru toate implementările SCSI-3.

*Comenzile* reprezintă specificații care definesc clasele de echipamente și un model de echipament pentru fiecare clasă. Aceste specificații definesc comenzile care trebuie implementate de toate echipamentele sau cele care sunt specifice diferitelor clase de echipamente, și descriu regulile care trebuie urmate de un echipament inițiator atunci când transmite comenzi unui alt echipament. Principalele standarde care se referă la comenzi sunt următoarele:

- *Primary Commands (SPC)*: comenzi de bază pentru toate echipamentele SCSI;
- *Block Commands (SBC)*: comenzi pentru echipamente cu acces direct cum sunt unitățile de discuri magnetice;
- *Stream Commands (SSC)*: comenzi pentru echipamente cu acces secvențial cum sunt unitățile de benzi magnetice;
- *Medium Changer Commands (SMC)*: comenzi pentru echipamente de schimbare a volumelor cum sunt tonomatele pentru discuri audio;
- *Multimedia Commands (MMC)*: comenzi pentru unități de discuri optice cum sunt CD-ROM, CD-R/E (*Recordable/Erasedable*), DVD;
- *Controller Commands (SCC)*: comenzi pentru controlere de I/E, de exemplu, pentru seturi de unități de discuri RAID (*Redundant Array of Independent Disks*).

*Protocoalele* reprezintă specificații care definesc regulile de comunicație între diferite echipamente SCSI.

*Interconexiunile* conțin specificații care definesc diferite interfețe fizice. Colecția de standarde SCSI-3 definește mai multe tipuri de interfețe. Pentru o lungă perioadă de timp, cea mai utilizată a fost interfața paralelă SCSI (*SCSI Parallel Interface* – SPI), până când a început să fie înlocuită cu interfața serială *Serial Attached SCSI* (SAS). O altă interfață serială este *IEEE 1394*, utilizată în special pentru aplicații video. *Fibre Channel* este o interfață serială cu performanțe ridicate, care permite comunicația prin fibră optică. Interfața serială *Serial Storage Architecture* (SSA) este destinată conectării unităților de discuri sau a matricilor de discuri RAID la servere. *InfiniBand* este o altă interfață serială cu performanțe ridicate, destinată conectării procesoarelor cu periferice de viteză ridicată cum sunt echipamentele de memorare.



*iSCSI (Internet SCSI)* permite calculatoarelor server să acceseze volume de discuri aflate la distanță utilizând infrastructura de rețea existentă și protocolul TCP/IP.

Considerăm interfața SCSI paralelă. Au fost elaborate mai multe versiuni ale acestei interfețe, pe măsură ce protocolul electric a fost îmbunătățit. Diferitele tipuri ale interfeței paralele SCSI sunt prezentate în tabelul 6.1.

**Tabelul 6.1.** Tipuri de interfețe paralele SCSI specificate în standardele SCSI-3.

Standard	Tehnologie	Nume alternativ	Frecvența de ceas (MHz)	Nr. biți	Rata de transfer maximă (MB/s)
SPI	Fast-20	Ultra	20	8	20
SPI	Fast-20/Wide	Ultra/Wide	20	16	40
SPI-2	Fast-40	Ultra2	40	8	40
SPI-2	Fast-40/Wide	Ultra2/Wide	40	16	80
SPI-3	Fast-80DT	Ultra3 (Ultra160)	80	16	160
SPI-4	Fast-160DT	Ultra4 (Ultra320)	80	16	320
SPI-5	Fast-320DT	Ultra5 (Ultra640)	160	16	640

În tabelul 6.1, DT (*Double Transition*) indică efectuarea a două transferuri în fiecare ciclu de ceas, câte unul la fiecare front al semnalului de ceas. La interfețele SPI-3, SPI-4 și SPI-5, lățimea magistralei SCSI este de 16 biți. Începând cu interfața SPI-2, se utilizează numai interfața electrică diferențială LVD.

## 6.4. Magistrala SCSI

### 6.4.1. Comunicația pe magistrala SCSI

Comunicația pe magistrala SCSI are loc între un echipament care inițiază transferul și un echipament destinație. La un moment dat, comunicația se realizează doar între două echipamente, un *inițiator* care selectează și comandă echipamentul *destinație* care efectuează operația cerută. De obicei, un echipament SCSI are un rol fix de inițiator sau destinație, dar unele echipamente pot îndeplini ambele roluri.

Un echipament inițiator poate adresa până la opt unități logice conectate la un echipament destinație. Pentru toate blocurile de date se utilizează adrese logice și nu fizice. Pentru echipamentele cu adresare directă, fiecare unitate logică poate fi interogată pentru a determina numărul blocurilor de date pe care le conține. O unitate logică poate coincide cu un echipament periferic sau poate fi o parte a acestuia.

Standardele SCSI definesc nivelul semnalelor de pe magistrală, funcția lor logică, protocolul de comunicație și secvențele de comenzi. Toate echipamentele trebuie să permită utilizarea protocolului asincron definit de standarde pentru transferurile de date. În plus, este definit un protocol opțional pentru transferuri sincrone. De asemenea, este specificat un protocol pentru transmiterea mesajelor în scopul controlului interfeței.

Magistrala SCSI utilizează un sistem de arbitraj distribuit pentru a permite existența a mai multor inițiatori și execuția concurentă a operațiilor de I/E. Un sistem de priorități permite acordarea magistralei pentru echipamentul SCSI cu prioritatea cea mai mare dintre cele care solicită magistrala. Timpul necesar pentru efectuarea arbitrajului este independent de numărul echipamentelor care solicită magistrala și este mai mic de 10  $\mu$ s.

Inițiatorul poate solicita magistrala SCSI și poate selecta un anumit echipament destinație. Destinația poate solicita transferul informațiilor de date, de comandă sau de stare pe magistrala de date, iar în unele cazuri poate solicita magistrala și poate reselecta inițiatorul în scopul continuării unei operații.

### 6.4.2. Protocolul SCSI asincron și sincron

Interfața SCSI permite utilizarea fie a protocolului asincron, fie a protocolului sincron pentru transferurile de date. Prin utilizarea protocolului sincron se pot obține transferuri de date mai rapide. Implementarea protocolului asincron este obligatorie pentru toate echipamentele, în timp ce implementarea protocolului sincron este opțională.

În cazul protocolului SCSI *asincron*, un echipament transmite mai întâi o cerere (prin activarea semnalului *REQ*), iar apoi așteaptă o confirmare de la inițiator. Inițiatorul depune un octet sau cuvânt pe magistrala de date și activează semnalul *ACK*. Echipamentul citește octetul sau cuvântul, dezactivează semnalul *REQ* și așteaptă ca inițiatorul să dezactiveze semnalul *ACK*. Aceste operații sunt repetate apoi pentru următorul octet sau cuvânt.

Acest protocol *REQ/ACK* cu confirmare necesită ca semnalele să se propage prin cablul SCSI de două ori pentru fiecare transfer de date. Aceasta necesită un anumit timp; în mod tipic, întârzierea de propagare este în jur de 5,25 ns pentru fiecare metru. Întârzierea de propagare reprezintă principala limitare a vitezei transferurilor SCSI asincrone atunci când se utilizează cabluri lungi.

În cazul protocolului SCSI *sincron*, o confirmare este de asemenea necesară pentru fiecare cerere transmisă, dar confirmarea poate fi întârziată. În consecință, un echipament poate transmite pachete de date unul după altul fără întârzierile de propagare necesare în cazul protocolului asincron. Viteza este determinată de durata ciclului de magistrală indiferent de întârzierea de propagare, și astfel este independentă de lungimea cablului.

Prima frecvență de ceas specificată pentru transferurile sincrone a fost de 5 MHz. Această frecvență a fost crescută treptat în noile versiuni ale interfeței, fiecare versiune dublând frecvența versiunii sale precedente. Începând cu interfața SPI-3, se utilizează tehnologia DT (*Double Transition*), astfel încât transferurile de date au loc atât pe frontul crescător, cât și pe frontul descrescător al fiecărui ciclu de ceas.

### 6.4.3. Semnalele magistralei SCSI

Magistrala SCSI de 8 biți care utilizează cablul A conține 18 semnale, dintre care 9 semnale de date și 9 semnale de control. Pentru variantele de 16 și 32 de biți există extensii ale magistralei. Semnalele magistralei sunt descrise în continuare.

- *BSY (Busy)*: Un semnal SAU cablat care indică starea ocupată a magistralei.
- *SEL (Select)*: Un semnal SAU cablat utilizat de un echipament inițiator pentru a selecta un echipament destinație sau de un echipament destinație pentru a reselecta un echipament inițiator. Identificatorul echipamentului selectat va apare pe liniile de date.
- *C/D (Control/Data)*: Este utilizat de echipamentul destinație pentru a specifica dacă se transmit informații de control sau de date pe magistrala de date. Starea activă a acestui semnal specifică faptul că se transmit informații de control.
- *I/O (Input/Output)*: Echipamentul destinație controlează cu acest semnal direcția de transfer a datelor. Direcția se consideră din punctul de vedere al echipamentului inițiator. Valoarea activă indică o operație de intrare pentru inițiator. Acest semnal este utilizat și pentru a deosebi între ele fazele de *Selecție* și *Reselecție*.
- *MSG (Message)*: Destinația indică prin acest semnal faptul că pe magistrală se transmite un mesaj (în faza de *Mesaje*).
- *REQ (Request)*: Este generat de echipamentul destinație pentru a specifica o cerere de transfer utilizând protocolul asincron.
- *ACK (Acknowledge)*: Este generat de echipamentul inițiator pentru a confirma o cerere de transfer asincron efectuată de un echipament destinație prin activarea semnalului *REQ*.
- *ATN (Attention)*: Este utilizat de un echipament inițiator pentru a indica o condiție de atenționare pentru echipamentul destinație.
- *RST (Reset)*: Un semnal SAU cablat care inițializează magistrala SCSI și resetează toate echipamentele conectate la magistrală.
- *DB (7..0, P) (Data Bus)*: Reprezintă semnalele bidirecționale de date și semnalul pentru bitul de paritate, care formează o magistrală de date. *DB (7)* este bitul cel mai

semnificativ și are prioritatea cea mai mare în timpul fazei de arbitraj. *DB (P)* este bitul de paritate impară a datelor. Paritatea este nedefinită în timpul fazei de arbitraj.

- *DB (31..8, P1, P2, P3) (Data Bus)*: Reprezintă extensia magistralei de date. *DB (P1, P2, P3)* sunt biți de paritate impară pentru *DB (15..8)*, *DB (23..16)*, respectiv *DB (31..24)*.
- *TERMPWR (Terminator Power)*: Reprezintă semnalul de alimentare pentru terminatorii de magistrală.

#### 6.4.4. Fazele magistralei SCSI

În timpul funcționării, magistrala SCSI trece prin mai multe stări distincte, numite *faze*. O fază descrie sensul transferului și conținutul informațiilor transferate. Se pot distinge următoarele faze:

- Magistrală liberă (*Bus Free*);
- Arbitrare (*Arbitration*);
- Selecție (*Selection*);
- Reselecție (*Reselection*);
- Comandă (*Command*);
- Date (*Data*);
- Stare (*Status*);
- Mesaj (*Message*).

După resetare sau punerea sub tensiune, magistrala trece în faza de *Magistrală liberă*. Aceasta este urmată de faza de *Arbitrare*, în care unul din echipamente câștigă controlul asupra magistralei. Dacă arbitrarea eșuează, magistrala revine în faza de *Magistrală liberă*. Dacă arbitrarea reușește, magistrala trece în faza de *Selecție* sau *Reselecție*, în care se selectează echipamentul destinație și cel inițiator care vor executa o comandă. După determinarea celor două echipamente, urmează una sau mai multe faze de transfer a informațiilor (*Comandă*, *Date*, *Stare*, *Mesaj*). Ultima fază de transfer a informațiilor este, în mod normal, faza de *Mesaj la inițiator*, în care se transferă un mesaj DISCONNECT sau COMMAND COMPLETE la inițiator, urmată de faza de *Magistrală liberă*.

fronturile anterioare ale impulsurilor *REQ* succesive și ale impulsurilor *ACK* succesive pentru transferul corect al datelor.

Câteva exemple de mesaje sunt descrise în continuare.

- **COMMAND COMPLETE**: Este transmis de echipamentul destinație la inițiator pentru a indica faptul că o comandă a fost terminată și s-a transmis o stare validă la inițiator.
- **DISCONNECT**: Este transmis de echipamentul destinație pentru a informa un inițiator că se va întrerupe conexiunea existentă și că va fi necesară o reconectare ulterioară pentru a termina operația curentă.
- **INITIATOR DETECTED ERROR**: Este transmis de inițiator pentru a informa un echipament destinație că a apărut o eroare (de exemplu, de paritate) care nu împiedică destinația de la reîncercarea operației.
- **ABORT**: Este transmis de inițiator la echipamentul destinație pentru a abandona operația curentă.
- **SYNCHRONOUS DATA TRANSFER**: Este transmis de inițiator la un echipament destinație pentru a selecta protocolul de transfer sincron.

#### 6.4.5. Execuția unei comenzi SCSI

Se va utiliza o comandă **READ** ca exemplu pentru a explica diferitele faze și semnale ale magistralei. Această comandă transferă date de la un echipament destinație la inițiator.



Comanda începe în faza de *Magistrală liberă*. Apoi, urmează o fază de *Arbitrare*, în care unul sau mai multe echipamente concurează pentru a prelua controlul asupra magistralei. Fiecare din echipamente activează semnalul *BSY* și una din liniile de date. Fiecare echipament are un identificator unic de la 0 la 7 (sau de la 0 la 15) și fiecare echipament activează una din liniile de date corespunzătoare propriului identificator. Fiecărui identificator  $i$  se asignează o prioritate, 7 (sau 15) având prioritatea cea mai înaltă, iar 0 prioritatea cea mai joasă. Dacă mai mult de un echipament activează identificatorul său în timpul fazei de *Arbitrare*, atunci echipamentul cu prioritatea cea mai înaltă preia controlul asupra magistralei.

Echipamentul care a câștigat arbitrarea devine inițiator. Acest echipament trece în faza de *Selecție* prin activarea semnalului *SEL*. În timpul acestei faze, inițiatorul activează ambele linii de date corespunzătoare propriului identificator și cel al echipamentului destinație. După o întârziere, inițiatorul dezactivează semnalul *BSY*. Atunci când echipamentul destinație detectează că semnalul *SEL* este activat, *BSY* și *I/O* sunt dezactivate, și își recunoaște identificatorul, va activa semnalul *BSY*. Atunci când inițiatorul detectează că semnalul *BSY* este activat, eliberează magistrala de date și dezactivează semnalul *SEL*.

În continuare, echipamentul destinație indică faptul că a trecut în faza de *Comandă* prin activarea semnalului *C/D*. Acest semnal va rămâne activat în timpul acestei faze. Echipamentul destinație activează apoi semnalul *REQ* pentru a solicita primul octet al comenzii de la inițiator. Inițiatorul depune primul octet al comenzii pe magistrala de date și activează semnalul *ACK*. Echipamentul destinație citește octetul și dezactivează semnalul *REQ*; inițiatorul dezactivează apoi semnalul *ACK*. Primul octet al comenzii conține codul operației, care indică numărul de octeți rămași de transferat. Acești octeți suplimentari sunt transferați cu același protocol *REQ/ACK*.

După ce echipamentul destinație a recepționat și interpretat comanda, trece magistrala în faza de *Date la inițiator* prin dezactivarea semnalului *C/D* (indicând informații de date) și activează semnalul *I/O* (indicând direcția de la destinație la inițiator). Echipamentul destinație depune primul octet sau cuvânt al datelor solicitate pe magistrala de date și activează semnalul *REQ*. Inițiatorul activează semnalul *ACK* după ce a citit datele. Octeții sau cuvintele suplimentare de date sunt transferate cu același protocol *REQ/ACK*.

După transferul tuturor datelor solicitate, echipamentul destinație trece magistrala în faza de *Stare* și transferă un octet de stare la inițiator, indicând faptul că a terminat cu succes transferul. Semnalul *C/D* este activat din nou și semnalul *I/O* rămâne activat. Se utilizează protocolul *REQ/ACK* pentru a transfera octetul de stare.

În sfârșit, echipamentul destinație trece magistrala în faza de *Mesaj la inițiator* prin activarea semnalului *MSG* și transferarea octetului mesajului *COMMAND COMPLETE*. Odată ce acest mesaj este recepționat de inițiator, echipamentul destinație eliberează toate semnalele magistralei pentru a trece magistrala în faza de *Magistrală liberă*.

## 6.5. Comenzi SCSI

Standardele SCSI specifică un set de comenzi de nivel înalt pe care trebuie să le recunoască echipamentele SCSI. Sunt definite comenzi obligatorii și comenzi opționale, dintre care unele sunt comune pentru toate tipurile de echipamente, în timp ce altele sunt specifice pentru diferite tipuri de echipamente.

### 6.5.1. Structura blocului descriptor al comenzii

O comandă este specificată sub forma unui *bloc descriptor al comenzii* (*Command Descriptor Block*), care se transmite la echipamentul destinație. Pentru unele comenzi, blocul descriptor al comenzii este urmat de o listă de parametri care se transmit în timpul fazei de *Date la destinație*. Blocul descriptor al comenzii începe cu un cod de operație în primul octet și se termină cu un octet de control.

Există structuri tipice ale blocului descriptor pentru comenzi de 6, 10 și 16 octeți. Figura 6.3 prezintă structura tipică a unui bloc descriptor al comenzii pentru comenzile de 10 octeți. În această figură și în cele următoare, Octet ms reprezintă octetul mai semnificativ, iar Octet mps reprezintă octetul mai puțin semnificativ.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație							
1	Număr unitate logică			Rezervat				
2	Octet ms							
3	Adresă bloc logic							
4								
5								
6	Rezervat							
7	Octet ms		Lungime transfer, Lungime listă					
8	de parametri, Lungime alocată							Octet mps
9	Control							

Figura 6.3. Structura unui bloc descriptor pentru comenzile SCSI de 10 octeți.

Codul de operație conține două câmpuri: *codul de grup* (biții 7..5) și *codul de comandă* (biții 4..0). Cei trei biți ai codului de grup permit 8 grupuri de coduri. Cei cinci biți ai codului de comandă permit 32 de coduri de comandă în fiecare grup. Astfel, există un număr total de 256 de coduri de operație posibile. Codul de grup definește grupuri separate pentru comenzile de 6, 10 sau 16 octeți, ca și pentru comenzi specifice producătorilor.

*Numărul unității logice* este definit în mesajul IDENTIFY. Destinația va ignora numărul unității logice specificat în blocul descriptor al comenzii dacă s-a recepționat un mesaj IDENTIFY. Se recomandă ca numărul unității logice din blocul descriptor să fie setat la zero. Acest câmp a fost inclus în blocul descriptor al comenzii pentru compatibilitate cu unele echipamente SCSI-1.

*Adresa blocului logic* din cadrul unei unități logice sau a unei partiții a unui volum începe cu blocul zero și trebuie să fie contiguă până la ultimul bloc logic al unității logice sau al partiției. Un bloc descriptor pentru comenzi de 6 octeți conține o adresă de bloc logic de 21 de biți. Blocurile descriptor pentru comenzile de 10 octeți și 16 octeți conțin adrese de blocuri logice de 32 de biți.

*Lungimea transferului* specifică lungimea datelor care trebuie transferate, de obicei, sub forma numărului de blocuri. Pentru unele comenzi, lungimea transferului specifică numărul de octeți care trebuie transferați. Pentru comenzile care utilizează un octet pentru lungimea transferului, o lungime a transferului între 1 și 255 indică numărul de blocuri care trebuie transferate printr-o singură comandă. O valoare 0 indică 256 de blocuri. Pentru comenzile care utilizează mai mulți octeți pentru lungimea transferului, o lungime 0 indică faptul că nu trebuie transferate date.

*Lungimea listei de parametri* se utilizează pentru a specifica numărul de octeți care se transferă în timpul fazei de *Date la destinație*, octeți care reprezintă parametrii transmiși la echipamentul destinație.

*Lungimea alocată* specifică numărul maxim de octeți alocați de echipamentul inițiator pentru datele transmise de la echipamentul destinație. Destinația trebuie să termine faza de *Date la inițiator* atunci când s-au transferat un număr de octeți indicați de lungimea alocată. Această lungime se utilizează pentru a limita numărul de octeți returnați la echipamentul inițiator.

*Câmpul de control* are structura următoare:

7	6	5	4	3	2	1	0
Specific producător		Rezervat				Flag	Link

Bitul *Link* este utilizat pentru a continua procesul de I/E după ce comanda curentă se termină cu succes. Dacă bitul *Link* este 1, destinația trece în faza de *Comandă* după terminarea comenzii curente. Implementarea acestui bit este opțională. Bitul *Flag* specifică mesajul care trebuie returnat de echipamentul destinație la inițiator dacă bitul *Link* este 1 și comanda se termină fără erori. De obicei, acest bit este utilizat pentru a genera o întrerupere a echipamentului inițiator între comenzile înlănțuite. Implementarea acestui bit este opțională.

Dacă bitul *Link* este 0 și bitul *Flag* este 1, echipamentul destinație va returna starea *Check Condition*. Această stare indică faptul că a apărut un eveniment neprevăzut în timpul

operației. Echipamentul inițiator trebuie să transmită o comandă suplimentară (*Request Sense*) pentru a determina evenimentul neprevăzut. Dacă bitul *Link* este 1, bitul *Flag* este 0 și comanda se termină cu succes, destinația transmite mesajul LINKED COMMAND COMPLETE. Dacă bitul *Link* este 1, bitul *Flag* este 1 și comanda se termină cu succes, destinația transmite mesajul LINKED COMMAND COMPLETE (WITH FLAG). Aceste mesaje indică terminarea unei comenzi înlănțuite.

### 6.5.2. Exemple de comenzi

Standardul SCSI-3 definește comenzi care se pot utiliza pentru toate tipurile de echipamente și comenzi care sunt specifice pentru diferite tipuri de echipamente. Principalele tipuri de echipamente pentru care sunt definite comenzi sunt următoarele:

- Echipamente cu acces direct (discuri magnetice);
- Echipamente cu acces secvențial (benzi magnetice);
- Imprimante;
- Procesoare (echipamente inteligente);
- Discuri WORM (*Write Once, Read Multiple*);
- Discuri CD-ROM (inclusiv discuri audio);
- Discuri DVD;
- Scanere;
- Memorii optice (unele discuri optice, de exemplu CD-R);
- Echipamente de comunicație (noduri de rețea).

În tabelul 6.2 se prezintă unele comenzi SCSI pentru echipamentele cu acces direct.

**Tabelul 6.2.** Comenzi SCSI pentru echipamente cu acces direct.

Nume comandă	Cod	Nume comandă	Cod
ATA PASS-THROUGH (12)	0xA1	READ MEDIA SERIAL NUMBER	0xAB/0x01
ATA PASS-THROUGH (16)	0x85	RECEIVE DIAGNOSTIC RESULTS	0x1C
FORMAT UNIT	0x04	REQUEST SENSE	0x03
INQUIRY	0x12	SEND DIAGNOSTIC	0x1D
READ (10)	0x28	START STOP UNIT	0x1B
READ (12)	0xA8	TEST UNIT READY	0x00
READ (16)	0x88	VERIFY (10)	0x2F
READ BUFFER (10)	0x3C	VERIFY (16)	0x8F
READ BUFFER (16)	0x9B	WRITE (10)	0x2A
READ CAPACITY (10)	0x25	WRITE (12)	0xAA
READ CAPACITY (16)	0x9E/0x10	WRITE (16)	0x8A
READ DEFECT DATA (10)	0x37	WRITE AND VERIFY (10)	0x2E
READ DEFECT DATA (12)	0xB7	WRITE BUFFER	0x3B

### 6.5.3. Comanda INQUIRY

Comanda INQUIRY solicită informații despre unitatea logică și dispozitivul SCSI. Blocul descriptor al acestei comenzi este prezentat în figura 6.4.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x12)							
1	Rezervat							EVPD
2	Cod pagină							
3	Octet ms	Lungime alocată						Octet mps
4								
5	Control							

**Figura 6.4.** Structura blocului descriptor pentru comanda INQUIRY.

Atunci când bitul EVPD (*Enable Vital Product Data*) este setat la 1, controlerul dispozitivului va returna datele specificate de câmpul Cod pagină (codurile de pagină nu sunt

descrie în acest document). Dacă pagina solicitată nu este implementată, comanda este terminată cu codul de stare al dispozitivului setat la CHECK CONDITION, un cod al cheii (*sense key*) setat la ILLEGAL REQUEST (0x05) și un cod suplimentar (*additional sense code*) setat la INVALID FIELD IN CDB (0x24). În cazul interfeței de programare ASPI, codul de stare al dispozitivului se află în octetul SRB\_TargStat al structurii SRB (*SCSI Request Block*). Codul cheii se află în octetul SenseArea[2] al structurii SRB, iar codul suplimentar se află în octetul SenseArea[12] al structurii SRB. Atunci când bitul EVPD este setat la 0, controlerul dispozitivului va returna datele standard pentru comanda INQUIRY. În acest caz, câmpul Cod pagină trebuie setat la 0; în caz contrar, comanda este terminată cu codul de stare al dispozitivului setat la CHECK CONDITION.

Câmpul Lungime alocată specifică numărul maxim de octeți alocați pentru datele returnate de la dispozitivul SCSI (lungimea bufferului alocat de utilizator). Octetul 3 al blocului descriptor trebuie să conțină octetul mai semnificativ (Octet ms) al lungimii bufferului, iar octetul 4 trebuie să conțină octetul mai puțin semnificativ (Octet mps) al lungimii bufferului. Câmpul Control trebuie setat la 0.

Datele standard ale comenzii INQUIRY conțin cel puțin 36 octeți. Un număr de octeți suplimentari pot conține date specifice producătorului. Formatul primilor 36 octeți ai datelor standard ale comenzii INQUIRY este prezentat în figura 6.5. O parte din câmpurile acestei structuri de date sunt descrise în continuare. Alte câmpuri (reprezentate cu o culoare, dar fără nume de câmpuri) nu sunt descrise în acest document.

Bit Octet	7	6	5	4	3	2	1	0
0	Calificator periferic			Tip dispozitiv periferic				
1	RMB				Rezervat			
2	Versiune							
3	Rezervat			Formatul datelor de răspuns (0x02)				
4	Lungime suplimentară							
5	Rez.					Rezervat		
6	Rez.				Rezervat			
7	Rezervat				Rezervat		CmdQue	
8	Octet ms							
...	Identificare producător							
15	Octet mps							
16	Octet ms							
...	Identificare produs							
31	Octet mps							
32	Octet ms							
...	Revizie produs							
35	Octet mps							

Figura 6.5. Primii 36 octeți ai datelor standard returnate de comanda INQUIRY.

Câmpul Calificator periferic specifică dacă unitatea logică adresată este accesibilă (biții câmpului sunt 000) sau nu este accesibilă de controlerul din dispozitivul SCSI. Câmpul Tip dispozitiv periferic identifică tipul dispozitivului (de exemplu, periferic cu acces direct la nivel de bloc, unitate CD/DVD, controler RAID, etc.). Bitul RMB (*Removable Medium*) indică dacă suportul nu este amovibil (atunci când este 0) sau este amovibil (atunci când este 1).

Câmpul Versiune specifică versiunea standardului *SCSI Primary Commands* (SPC) care este implementată. Acest câmp conține 0x04 pentru versiunea SPC-2, 0x05 pentru versiunea SPC-3, 0x06 pentru versiunea SPC-4 și 0x07 pentru versiunea SPC-6. Câmpul Lungime suplimentară indică lungimea rămasă (în octeți) a datelor standard ale comenzii INQUIRY (numărul de octeți rămași, excluzând octeții din câmpurile anterioare). Conținutul acestui câmp nu depinde de lungimea alocată specificată în blocul descriptor al comenzii. Atunci când bitul CmdQue este setat la 1, indică faptul că unitatea logică permite gestiunea comenzilor printr-o coadă de comenzi.

Câmpul Identificare producător conține opt caractere ASCII care identifică producătorul unității logice. Șirurile de identificare ale producătorilor sunt asigurate de comitetul T10.

Câmpul Identificare produs conține 16 caractere ASCII care identifică produsul. Șirurile de identificare ale produselor sunt definite de producător. Câmpul Revizie produs conține patru caractere ASCII care identifică revizia produsului. Șirurile cu reviziile produselor sunt definite de producător.

#### 6.5.4. Comanda READ CAPACITY (10)

Comanda READ CAPACITY (10) solicită controlerului de dispozitiv să transfere opt octeți de date într-un buffer alocat de utilizator. Datele transferate descriu capacitatea și lungimea blocului logic al unui dispozitiv cu acces direct la nivel de bloc. Blocul descriptor al acestei comenzi este prezentat în figura 6.6. Câmpurile Rezervat și Control trebuie setate la 0.

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

Figura 6.6. Structura blocului descriptor pentru comanda READ CAPACITY (10).

Formatul datelor returnate de comanda READ CAPACITY (10) este ilustrat în figura 6.7. Câmpul de patru octeți Adresa returnată a blocului logic conține adresa ultimului bloc logic al dispozitivului. Acest câmp conține 0xFFFFFFFF dacă adresa ultimului bloc logic este mai mare decât valoarea maximă care poate fi specificată în acest câmp. În acest caz, aplicația trebuie să transmită o comandă READ CAPACITY (16) controlerului de dispozitiv pentru a transfera opt octeți de date cu capacitatea dispozitivului. Câmpul de patru octeți Lungimea blocului logic conține numărul octeților de date ale utilizatorului dintr-un bloc logic.

Bit Octet	7	6	5	4	3	2	1	0
0	Octet ms							
...	Adresa returnată a blocului logic							
3								
4	Octet ms							
...	Lungimea blocului logic							
7								

Figura 6.7. Structura datelor returnate de comanda READ CAPACITY (10).

#### 6.5.5. Comanda READ CAPACITY (16)

Comanda READ CAPACITY (16) solicită controlerului de dispozitiv să transfere într-un buffer alocat de utilizator date care descriu capacitatea și formatul suportului unui dispozitiv cu acces direct la nivel de bloc. Această comandă utilizează un format al structurii CDB numit SERVICE ACTION IN (16), ilustrat în figura 6.8.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0x9E)							
1	Rezervat				Acțiune serviciu (0x10)			
2	Rezervat							
...								
9	Lungime alocată							
10								
...	Octet mps							
13								
14	Rezervat							
15	Control							

Figura 6.8. Structura blocului descriptor pentru comanda READ CAPACITY (16).



Câmpurile Cod de operație și Acțiune serviciu trebuie setate la valorile indicate în figura 6.8. Câmpurile Rezervat și Control trebuie setate la 0. Câmpul Lungime alocată trebuie setat la numărul maxim de octeți alocați pentru datele returnate de dispozitiv (lungimea bufferului alocat de utilizator).

Bit Octet	7	6	5	4	3	2	1	0
0	Octet ms							
...	Adresa returnată a blocului logic							
7	Octet mps							
8	Octet ms							
...	Lungimea blocului logic							
11	Octet mps							

Figura 6.9. Primii 12 octeți ai datelor returnate de comanda READ CAPACITY (16).

Comanda READ CAPACITY (16) returnează 32 octeți de date. Formatul primilor 12 octeți ai datelor returnate este ilustrat în figura 6.9. Câmpul de opt octeți Adresa returnată a blocului logic conține adresa ultimului bloc logic al dispozitivului. Câmpul de patru octeți Lungimea blocului logic conține numărul octeților de date ale utilizatorului dintr-un bloc logic.

### 6.5.6. Comanda READ CD RECORDED CAPACITY

Comanda READ CD RECORDED CAPACITY permite solicitarea informațiilor despre capacitatea înregistrată a unui disc CD/DVD prezent într-o unitate logică. Formatul blocului descriptor și structura datelor returnate de această comandă sunt aceleași ca și cele ale comenzii READ CAPACITY (10) (secțiunea 6.5.4). Adresa din câmpul Adresa returnată a blocului logic este cea a ultimului sector din ultima sesiune completă de înregistrare. Lungimea blocului logic este raportată ca fiind 2048.

### 6.5.7. Comanda READ MEDIA SERIAL NUMBER

Comanda READ MEDIA SERIAL NUMBER solicită controlerului de dispozitiv să returneze numărul de serie al suportului prezent într-o unitate logică. Această comandă utilizează un format al structurii CDB numit SERVICE ACTION IN (12), ilustrat în figura 6.10.

Bit Octet	7	6	5	4	3	2	1	0
0	Cod de operație (0xAB)							
1	Rezervat				Acțiune serviciu (0x01)			
2	Rezervat							
...	Rezervat							
5	Rezervat							
6	Octet ms							
...	Lungime alocată							
9	Octet mps							
10	Rezervat							
11	Control							

Figura 6.10. Structura blocului descriptor pentru comanda READ MEDIA SERIAL NUMBER.

Câmpurile Cod de operație și Acțiune serviciu trebuie setate la valorile indicate în figura 6.10. Câmpurile Rezervat și Control trebuie setate la 0. Câmpul Lungime alocată trebuie setat la numărul maxim de octeți alocați pentru numărul de serie (lungimea bufferului alocat de utilizator).

Formatul datelor returnate de comanda READ MEDIA SERIAL NUMBER este ilustrat în figura 6.11. Câmpul Lungimea numărului de serie al suportului conține numărul de octeți din câmpul Numărul de serie al suportului. Numărul de octeți este un multiplu de patru. Conținutul câmpului Lungimea numărului de serie al suportului nu depinde de lungimea alocată specificată în blocul descriptor al comenzii. Câmpul Numărul de serie al suportului con-

ține numărul de serie specific producătorului al suportului prezent în unitatea logică. Dacă numărul de serie nu este disponibil (de exemplu, suportul curent nu are un număr de serie valid), atunci câmpul Lungimea numărului de serie al suportului va fi setat la 0.

Bit Octet	7	6	5	4	3	2	1	0
0	Octet ms							
...	Lungimea numărului de serie al suportului (4n-4)							
3	Octet mps							
4	Octet ms							
...	Numărul de serie al suportului							
4n-1	Octet mps							

**Figura 6.11.** Structura datelor returnate de comanda READ MEDIA SERIAL NUMBER.

Dacă nu există niciun suport prezent în unitate, comanda este terminată cu codul de stare al dispozitivului setat la CHECK CONDITION, codul cheii setat la NOT READY (0x02) și codul suplimentar setat la MEDIUM NOT PRESENT (0x3A). În cazul interfeței de programare ASPI, locațiile acestor coduri din structura SRB au fost prezentate în secțiunea 6.5.3.

## 6.6. Configurarea echipamentelor SCSI

Pentru a configura un echipament SCSI trebuie efectuate două operații: stabilirea identificadorului SCSI și instalarea terminatoarelor.

### 6.6.1. Stabilirea identificadorului SCSI

La o magistrală SCSI se pot conecta până la 8 sau 16 echipamente SCSI și fiecare dintre ele trebuie să aibă un identificador SCSI unic pentru a evita conflictele. Unul din acești identificatori, de obicei 7 sau 15, care are prioritatea cea mai mare, i se asignează adaptorului din sistemul gazdă. Există adaptoare care permit încărcarea sistemului de operare doar de pe o unitate de discuri cu un anumit identificador. De exemplu, adaptoarele Adaptec mai vechi necesitau ca unitatea de discuri de pe care se încarcă sistemul de operare să aibă identificadorul 0. Adaptoarele mai noi permit încărcarea sistemului de pe oricare unitate, indiferent de identificadorul său.

De obicei, stabilirea identificadorului necesită poziționarea a trei sau patru comutatoare în unitate. Configurația comutatoarelor rezultă din reprezentarea binară a valorii identicatorilor de la 0 la 7 sau de la 0 la 15. În alte cazuri, identificadorul poate fi setat cu un comutator rotativ.

Sistemele SCSI actuale sunt de tip “*Plug and Play*” și asignarea identicatorului se realizează în mod automat de către sistemul de operare și adaptorul SCSI. Se utilizează un protocol numit SCAM (SCSI *Configured Automatically*), care interoghează echipamentele SCSI și asignează identificatori unici pentru fiecare, astfel încât să nu existe conflicte.

### 6.6.2. Instalarea terminatoarelor

Magistrala SCSI, ca și alte magistrale, are nevoie de terminatoare la ambele capete ale magistralei. Terminarea incorectă a liniilor magistralei reprezintă una din problemele care pot apare la utilizarea echipamentelor SCSI. Dacă adaptorul din sistemul gazdă este la unul din capetele magistralei, trebuie să aibă terminatorul său activat. Dacă adaptorul este la mijlocul magistralei și există legături atât spre magistrala internă, cât și spre cea externă, terminatorul din adaptor trebuie dezactivat, iar echipamentele de la cele două capete ale magistralei trebuie să aibă terminatoarele lor instalate.

Există mai multe tipuri de terminatoare pentru magistrala SCSI:

- Terminatoare pasive;
- Terminatoare active;
- Terminatoare FPT (*Forced Perfect Termination*).

*Terminatoarele pasive*, formate din rezistențe, permit fluctuații ale semnalelor magistralei. Nivelul semnalelor depinde de căderea de tensiune pe aceste rezistențe. De obicei, terminatoarele pasive sunt adecvate pentru distanțe scurte, de până la 1 m, dar pentru distanțe mai lungi sunt necesare terminatoare active.

*Terminatoarele active* folosesc, în locul divizoarelor de tensiune formate din rezistențe, unul sau mai multe regulatoare de tensiune care asigură o tensiune constantă. Regulatele de tensiune asigură terminarea semnalelor de pe magistrala SCSI la un nivel de tensiune corect. Standardele SCSI-2 și SCSI-3 recomandă folosirea terminatoarelor active la ambele capete ale magistralei.

*Terminatoarele FPT* reprezintă o variantă a terminatoarelor active care utilizează nivele de tensiune stabile obținute cu diode. Acestea elimină fluctuațiile nivelului semnalelor, mai ales la viteze de transfer mari sau lungimi mari ale cablurilor.

De obicei, echipamentele SCSI externe au un conector SCSI de intrare și unul de ieșire, astfel încât mai multe echipamente pot fi conectate în lanț. Atunci când echipamentul se află la unul din capetele magistralei SCSI, în conectorul de ieșire trebuie instalat un terminator.

Unele echipamente au terminatoare încorporate care pot fi activate sau dezactivate printr-un comutator sau prin îndepărtarea lor. Alte echipamente nu au terminatoare încorporate și pentru acestea trebuie utilizate module terminatoare externe. Modulele terminatoare sunt disponibile în diferite configurații ale conectorilor, care includ și terminatoare de trecere. Acestea sunt necesare pentru echipamentele instalate la capătul magistralei și care au un singur conector SCSI. Terminatoarele de trecere sunt utilizate în mod curent și în montajele interne în care echipamentele nu au rezistențe terminatoare încorporate. Multe din unitățile de discuri utilizează astfel de terminatoare în montajele interne pentru a economisi spațiul de pe placa logică.

### 6.6.3. Alte configurări

Există și opțiuni suplimentare care pot fi configurate pentru unitățile SCSI. Cele mai obișnuite setări suplimentare sunt următoarele:

- Pornirea la comandă;
- Paritatea SCSI;
- Tensiunea pentru terminatoare;
- Negocierea transferului sincron.

#### 6.6.3.1. Pornirea la comandă

Dacă există mai multe unități de discuri conectate la un sistem, acestea se pot configura astfel încât nu toate unitățile să pornească simultan, imediat după punerea sub tensiune a sistemului. În cazul pornirii simultane, consumul de putere ar crește semnificativ, sursa de alimentare poate fi supraîncărcată, ceea ce poate cauza blocarea sistemului sau alte probleme la pornire.

De obicei, unitățile SCSI permit întârzierea pornirii motorului de antrenare. Atunci când adaptoarele SCSI inițializează magistrala, majoritatea dintre ele transmit pe rând o comandă numită *Start Unit* la fiecare din unitățile de discuri. Prin poziționarea unui comutator din unitatea de discuri, motorul de antrenare va porni doar atunci când unitatea recepționează comanda *Start Unit* de la adaptorul calculatorului gazdă. Deoarece adaptorul transmite această comandă către toate unitățile de discuri, pe rând, începând cu unitatea cea mai prioritară până la unitatea cea mai puțin prioritară, unitățile cele mai prioritare pot fi configurate astfel încât să pornească primele, iar apoi unitățile mai puțin prioritare să pornească secvențial. Există unele adaptoare care nu transmit comanda *Start Unit*. În acest caz, unitățile de discuri pot fi configurate astfel încât să-și întârzie pornirea cu un număr oarecare de secunde, în loc de a aștepta comanda *Start Unit*.

Nu este necesară activarea funcției de pornire întârziată pentru unitățile care au o sursă de alimentare separată. Această funcție este necesară pentru unitățile interne care sunt alimentate de la aceeași sursă care alimentează sistemul.

### 6.6.3.2. Paritatea SCSI

Utilizarea unui bit de paritate permite o detectare limitată a erorilor. Majoritatea adaptoarelor permit controlul parității, deci această opțiune ar trebui validată la fiecare echipament. Această funcție este opțională, deoarece există adaptoare mai vechi care nu utilizează bitul de paritate SCSI, astfel încât pentru aceste adaptoare controlul parității trebuie dezactivat.

### 6.6.3.3. Tensiunea pentru terminatoare

Terminatoarele active poziționate la fiecare capăt al magistralei SCSI trebuie să fie alimentate de la cel puțin unul din echipamentele conectate la magistrală. De obicei, tensiunea de alimentare a terminatoarelor este furnizată de adaptorul calculatorului gazdă, însă există și excepții.

Nu reprezintă o problemă faptul că tensiunea pentru terminatoare este furnizată de mai multe echipamente, deoarece fiecare sursă este protejată cu ajutorul unor diode. Pentru simplitate, adesea se configurează toate echipamentele pentru a furniza tensiune terminatoarelor. Dacă nici unul din echipamente nu alimentează cu tensiune terminatoarele, magistrala nu va fi terminată corect și nu va funcționa corespunzător.

### 6.6.3.4. Negocierea transferului sincron

În mod implicit, magistrala SCSI utilizează protocolul de transfer asincron. Este posibilă selectarea protocolului de transfer sincron, care este mai rapid, printr-o procedură numită negocierea transferului sincron. Înaintea transferului de informații pe magistrală, echipamentul inițiator și echipamentul destinație negociază modul în care va avea loc transferul. Dacă ambele echipamente permit utilizarea protocolului sincron, transferul se va realiza cu acest protocol.

Unele echipamente mai vechi nu răspund cererii de transfer sincron și se pot chiar dezactiva atunci când recepționează asemenea cereri. Din acest motiv, adaptoarele și perifericele care permit negocierea transferului sincron au adesea un comutator care se poate utiliza pentru a dezactiva această negociere, astfel încât ele să poată funcționa cu echipamente mai vechi.

## 6.7. Adaptoare SCSI

Cei mai importanți producători de adaptoare SCSI sunt firmele Adaptec și Future Domain. Adaptoarele SCSI pot fi instalate cu ușurință, deoarece toate funcțiile lor pot fi configurate prin program. Informațiile de configurație sunt înscrise într-o memorie de pe placa logică adaptorului. Cele mai importante caracteristici ale acestor adaptoare sunt următoarele:

- Memoria ROM a adaptoarelor conține programe utilitare care permit configurarea completă a adaptoarelor;
- Nivelele de întrerupere, adresele memoriei ROM, adresele porturilor DMA și de I/E, paritatea magistralei SCSI, adresele fizice ale perifericelor pot fi configurate prin program;
- Terminatoarele pot fi validate și invalidate prin program;
- Nu sunt necesare drivere suplimentare pentru mai mult de două unități de discuri;
- Se permite pornirea la comandă a unităților;
- Sistemul de operare se poate încărca de pe oricare echipament SCSI.

## 6.8. Drivere SCSI

Fiecare din echipamentele SCSI conectate la magistrală, cu excepția unităților de discuri, are nevoie de un driver SCSI extern. De obicei, unitățile de discuri au driverele incluse în programul BIOS de pe adaptorul SCSI. Driverele externe sunt proiectate pentru un anumit tip de echipament și sunt specifice unui anumit adaptor SCSI.

S-au răspândit două tipuri de drivere standard pentru interfața cu adaptorul calculatorului gazdă. Dacă este disponibil un driver standard pentru adaptor, producătorii pot crea mai simplu noi drivere pentru echipamentele periferice, care pot dialoga cu driverul universal din adaptor. Astfel se elimină dependența de un tip particular de adaptor. Driverele universale efectuează legătura între adaptor și sistemul de operare.

Unul din aceste drivere este interfața de programare avansată ASPI (*Advanced SCSI Programming Interface*), driver care a fost inițiat de firma Adaptec. Ulterior, și alți furnizori de echipamente SCSI au utilizat driverul ASPI pentru produsele lor. Multe sisteme de operare includ un driver ASPI pentru diferite adaptoare SCSI.

Al doilea driver este numit CAM (*Common Access Method*) și a fost creat de firmele Future Domain și NCR. Driverul CAM este un protocol aprobat de organizația ANSI și permite comanda mai multor adaptoare SCSI. Firma Future Domain pune la dispoziție și un convertor CAM-ASPI pentru adaptoarele sale.

Sistemele de operare *Windows* actuale utilizează interfața de programare *SCSI Pass Through Interface* (SPTI) pentru accesul la dispozitivele SCSI. Această interfață de programare a fost dezvoltată de firma Microsoft și este accesibilă programelor prin intermediul funcției *DeviceIoControl()*, care permite transmiterea unui cod de control direct unui driver de dispozitiv.

## 6.9. Interfața Serial Attached SCSI

*Serial Attached SCSI* (SAS) reprezintă versiunea serială a interfeței SCSI. Aceasta utilizează un protocol serial punct la punct și setul standard de comenzi SCSI. Interfața serială SAS oferă compatibilitate cu unitățile de discuri SATA (*Serial ATA – Serial Advanced Technology Attachment*) de generația a doua, care se pot conecta la plăci de bază SAS.

Standardele SAS au fost elaborate de comitetul tehnic *T10* al *INCITS* (*InterNational Committee for Information Technology Standards*). Prima versiune a standardului interfeței SAS a fost publicată în anul 2003, iar o versiune îmbunătățită (SAS-1.1) a fost publicată în anul 2005. Ambele versiuni specifică a interfață serială cu o viteză maximă de 3 Gbiți/s. Standardul SAS-2, care a fost publicat de comitetul *INCITS* în anul 2009, definește generația a doua a interfeței SAS. Această versiune a standardului introduce o legătură serială cu o viteză maximă de 6 Gbiți/s, un nivel fizic care este compatibil cu interfața SATA, și protocoale pentru transferul comenzilor SCSI la echipamentele SAS și a comenzilor ATA la echipamentele SATA. Versiunea SAS-2.1 a standardului, care a fost aprobată în anul 2010, definește un număr de îmbunătățiri ale standardului SAS-2, printre care conectori suplimentari și facilități de gestiune a energiei consumate. Versiunea SAS-3 a standardului, care a fost publicată în anul 2013, definește generația a treia a interfeței SAS, cu o viteză maximă de 12 Gbiți/s. Versiunea SAS-4 a standardului, care a fost aprobată de comitetul tehnic *T10* în anul 2018, crește viteza maximă a interfeței la 24 Gbiți/s. Produse bazate pe versiunea SAS-4 (24G SAS) au început să apară în anul 2020.

Figura 6.12 ilustrează simbolul interfețelor SAS-2, SAS-3 și SAS-4.



Figura 6.12. Simboluri ale interfețelor SAS-2, SAS-3 și SAS-4.

Interfața SAS este o arhitectură punct la punct, cu fiecare echipament conectat direct la un port SCSI în loc de a fi conectat la o magistrală partajată. Deoarece lățimea de bandă nu



este partajată de mai multe echipamente, ca în cazul interfeței SCSI paralele, sunt posibile transferuri cu viteze mai ridicate. Prin utilizarea unei conexiuni punct la punct, siguranța datelor și posibilitatea de localizare a defectelor sunt îmbunătățite comparativ cu o arhitectură cu magistrală partajată.

Protocolul SAS specifică o comunicație duplex între controlerul SAS și o unitate de discuri. Astfel, operațiile de citire și scriere pot fi executate în același timp, ceea ce crește performanța. Comparativ, unitățile SATA utilizează o comunicație semiduplex, astfel încât datele sunt transmise la unitate pentru scriere, datele care trebuie citite de la unitate trebuie să aștepte terminarea comunicației precedente.

Se pot conecta un număr mare de unități de discuri SAS sau SATA la un port al unui controler SAS prin utilizarea unor *expandoare* SAS. Un expandor permite ca un singur inițiator să comunice cu un număr de echipamente destinație SAS/SATA. Un expandor SAS este similar cu un comutator dintr-o rețea, care permite conectarea mai multor sisteme utilizând un singur port al comutatorului. Costul unui sistem care conține un expandor este mult mai redus comparativ cu costul unui sistem care conține un controler SAS cu un număr mare de porturi sau mai multe controlere cu un număr mai mic de porturi. Prin utilizarea expandoarelor, sunt posibile până la 16.384 legături fizice.

Un domeniu SAS constă dintr-un set de echipamente SAS care comunică între ele cu ajutorul cablurilor și plăcilor de bază, cu sau fără expandoare. Fiecărui port SAS și echipament expandor dintr-un domeniu SAS i se asignează un identificator global unic de către producătorul echipamentului. Acest identificator de 64 biți care reprezintă adresa SAS este numit *World Wide Name* (WWN) și identifică în mod unic echipamentul în domeniul SAS, la fel cum un identificator SCSI identifică un echipament conectat la o magistrală SCSI paralelă. Din cei 64 de biți, 24 biți reprezintă identificatorul firmei producătoare și 40 de biți reprezintă identificatorul specific al producătorului. Spre deosebire de interfața SCSI paralelă, nu este necesară setarea manuală a adreselor într-un sistem SCSI serial, deoarece toate configurările se execută în mod automat. Interfața SAS nu necesită instalarea unor terminatoare ca în cazul interfeței SCSI paralele.

Interfața SAS utilizează semnale diferențiale și cifrarea datelor pentru reducerea interferențelor electromagnetice. Se pot combina până la patru porturi cu aceeași adresă într-un port de lățime mai mare, ceea ce permite creșterea ratei datelor.

Unitățile de discuri SAS sunt cu port dual, ceea ce înseamnă că acestea se pot conecta direct la și pot fi controlate de două controlere SAS în același timp. Această posibilitate permite realizarea unui sistem redundan. Atunci când unul din controlerele SAS se defectează, celălalt va fi în măsură să acceseze unitățile de discuri SAS și datele memorate pe aceste unități.



**Figura 6.13.** Conectori ai unităților de discuri SATA și SAS.

Un sistem SAS poate utiliza fie unități de discuri SAS, fie unități de discuri SATA. O unitate SATA poate fi conectată la un sistem SAS printr-un expandor sau un circuit care im-

plementează protocolul SAS. Această compatibilitate este posibilă deoarece conectorii unităților SAS și SATA sunt similari. Acești conectori sunt ilustrați în figura 6.13. Ambii conectori au același număr de pini, iar pinii au aceleași dimensiuni și aceeași formă. Totuși, conectorul unității SATA are o creștătură, care lipsește la conectorul unității SAS. Această creștătură previne conectarea unei unități SAS într-un sistem SATA. Un sistem SAS utilizează un conector care permite instalarea unei unități de discuri cu o creștătură (SATA) sau a unei unități de discuri fără creștătură (SAS).

Unitățile de discuri SAS au performanțe superioare comparativ cu unitățile de discuri SATA. Viteza de rotație a unităților SAS este cuprinsă între 10.000 și 15.000 rotații pe minut (RPM), în timp ce viteza de rotație a unităților SATA este cuprinsă între 5.400 și 7.200 RPM. Viteza de rotație mai ridicată reduce timpul de acces. Comunicația duplex permisă de unitățile SAS contribuie de asemenea la performanța superioară a acestor unități. Unitățile SAS sunt cu port dual, iar aceasta le permite să comunice cu două adaptoare ale calculatorului gazdă sau controlere simultan, ceea ce îmbunătățește disponibilitatea datelor. De asemenea, unitățile SAS sunt mult mai fiabile decât unitățile SATA și sunt proiectate pentru o utilizare mult mai intensă.

## 6.10. Aplicații

### 6.10.1. Răspundeți la următoarele întrebări:

- Care sunt îmbunătățirile introduse de standardele SCSI-3 pentru interfața paralelă SCSI?
- Care este deosebirea dintre protocolul SCSI asincron și cel sincron?
- Care sunt operațiile necesare pentru configurarea unui sistem SCSI?
- Care sunt avantajele interfeței SCSI seriale față de interfața SCSI paralelă?

**6.10.2.** Creați o aplicație *Windows* pentru a testa dacă managerul ASPI este inițializat corect și pentru a afișa numărul adaptoarelor SCSI. Ca model pentru aplicația *Windows*, utilizați aplicația *AppScroll* disponibilă pe pagina laboratorului în arhiva *AppScroll.zip*. Executați următoarele operații pentru a crea proiectul aplicației:

- În mediul de programare *Visual Studio 2019*, creați un nou proiect gol de tip *Windows Desktop* cu utilitarul *Windows Desktop Wizard*. Bifați opțiunea *Place solution and project in the same directory* pentru a evita crearea unui alt director pentru soluția creată.
- Modificați proprietatea *Character Set* a proiectului prin selectarea opțiunii *Not Set* pentru această proprietate.
- Copiați în directorul proiectului fișierele din arhiva *AppScroll.zip* și adăugați la proiect toate fișierele copiate.
- Copiați în directorul proiectului fișierele din arhiva *WNASPI32.zip*, disponibilă pe pagina laboratorului. Adăugați la proiect fișierele antet *wnaspi32.h* și *scsidefs.h*.
- Deschideți fișierul sursă *AppScroll.cpp*, ștergeți directiva `#include "Hw.h"` și adăugați directive `#include` pentru a include fișierele antet *wnaspi32.h* și *scsidefs.h*.
- În funcția `AppScroll()`, ștergeți secvențele pentru inițializarea bibliotecii HW cu funcția `HwOpen()` și pentru închiderea bibliotecii HW cu funcția `HwClose()`.
- Selectați *Build* → *Build Solution* și verificați ca aplicația să fie construită fără erori.

Utilizați specificațiile interfeței de programare ASPI pentru Win32 din documentul *ASPI32.pdf* (disponibil pe pagina laboratorului sau la referința bibliografică [1]) și biblioteca *wnaspi32.dll* pentru a scrie o funcție fără parametri de intrare, care execută următoarele operații:

- Încarcă biblioteca `wnaspi32.dll` în memorie cu funcția `LoadLibrary()`;
- Determină adresa funcției `GetASPI32SupportInfo()` și a funcției `SendASPI32Command()` cu funcția `GetProcAddress()`;
- Apelează funcția `GetASPI32SupportInfo()` și afișează dacă managerul ASPI este inițializat corect;
- Dacă managerul ASPI este inițializat corect, afișează numărul adaptoarelor SCSI din sistem.

După scrierea funcției, adăugați apelul acestei funcții în funcția `AppScroll()` și verificați execuția funcției.

**6.10.3.** Extindeți aplicația 6.10.2 prin scrierea unei funcții care execută următoarele operații pentru fiecare adaptor SCSI, începând cu adaptorul cu numărul 0:

- Apelează funcția `SendASPI32Command()` cu codul de comandă `SC_HA_INQUIRY` și afișează următoarele informații din structura `SRB_HAInquiry`: șirul care descrie managerul ASPI (membrul `HA_ManagerId` al structurii); identificatorul SCSI al adaptorului (membrul `HA_SCSE_ID` al structurii); șirul care descrie adaptorul (membrul `HA_Identifier` al structurii).
- Determină numărul maxim al dispozitivelor SCSI; acest număr este conținut în octetul `HA_Unique[3]` al structurii `SRB_HAInquiry`.
- Pentru fiecare dispozitiv SCSI și pentru fiecare unitate logică (de la 0 la 7) a unui dispozitiv, apelează funcția `SendASPI32Command()` cu codul de comandă `SC_GET_DEV_TYPE` și afișează numărul dispozitivului SCSI, numărul unității logice și tipul dispozitivului (membrul `SRB_DeviceType` al structurii `SRB_GDEVBlock`).

După scrierea funcției, adăugați apelul acestei funcții în funcția `AppScroll()` și verificați execuția funcției.

### Observație

- Pentru obținerea informațiilor despre unele dispozitive SCSI utilizând biblioteca `wnaspi32.dll`, sunt necesare privilegiile de administrator.

**6.10.4.** Extindeți aplicația 6.10.3 prin scrierea unei funcții care transmite comanda `READ CAPACITY (10)` unui dispozitiv cu acces direct (unitate de discuri). Parametrii acestei funcții sunt numărul adaptorului SCSI, numărul dispozitivului SCSI și numărul unității logice. În această funcție, creați un eveniment cu resetare manuală și fără nume utilizând funcția `CreateEvent()` și resetați evenimentul utilizând funcția `ResetEvent()`. Inițializați structura `SRB` pentru funcția `SendASPI32Command()`, specificând `SC_EXEC_SCSE_CMD` în câmpul de comandă, direcția de intrare și notificarea printr-un eveniment în câmpul `SRB_Flags`, și lungimea `SENSE_LEN` în câmpul `SRB_SenseLen`. Inițializați structura `CDB` pentru comanda `READ CAPACITY (10)`, iar apoi apelați funcția `SendASPI32Command()`. Dacă funcția revine cu codul `SS_PENDING`, apelați funcția `WaitForSingleObject()` cu indicatorul returnat de funcția `CreateEvent()` pentru a aștepta până când evenimentul va fi semnalat. Apoi, examinați câmpul `SRB_Status` pentru a testa modul de terminare al comenzii. În cazul în care comanda se termină cu codul `SS_ERR`, afișați un mesaj de eroare conținând octetul de stare al dispozitivului, codul cheii și codul suplimentar (prezentate în secțiunea 6.5.3), închiideți indicatorul evenimentului și reveniți din funcție cu un cod de eroare. În cazul în care comanda se termină cu codul `SS_COMP`, calculați capacitatea utilizând datele returnate, afișați capacitatea în KB și GB, închiideți indicatorul evenimentului și reveniți din funcție cu codul 0.

Includeți apelul acestei funcții în funcția scrisă pentru aplicația 6.10.3, atunci când se detectează o unitate de discuri, și verificați funcționarea rulând fișierul `.exe` ca administrator. Apoi, includeți apelul funcției și atunci când se detectează o unitate optică (CD-ROM). În

mașina virtuală, montați una din imaginile disponibile pentru discurile optice utilizând fișierul .iso sau .cue disponibil și verificați funcționarea funcției.

**6.10.5.** Extindeți aplicația 6.10.4 cu o funcție care transmite comanda INQUIRY unei unități de discuri sau unei unități optice. Parametrii acestei funcții sunt numărul adaptorului SCSI, numărul dispozitivului SCSI și numărul unității logice. În structura CDB a comenzii, bitul EVPD și câmpul Cod pagină trebuie să fie 0. În cazul în care comanda se execută cu succes, afișați dacă suportul este amovibil sau nu este amovibil, versiunea implementată a standardului SPC, șirul de identificare al producătorului, șirul de identificare al produsului și șirul care indică revizia produsului. Includeți apelul acestei funcții în funcția scrisă pentru aplicația 6.10.3, atunci când se detectează o unitate de discuri sau o unitate optică, și verificați funcționarea sa rulând fișierul.exe ca administrator.

## Bibliografie

- [1] Adaptec, Inc., “ASPI for Win32 Technical Reference”, 2001, <http://www.zianet.com/jgray/dat/files/ASPI32.pdf>.
- [2] Hoffman, C., Sawyer, S., “6Gb/s SAS – An Evolutionary Step for SAS Technology”, White Paper, SCSI Trade Association, September 2, 2008.
- [3] Mueller, S., *Upgrading and Repairing PCs*, 22nd Edition, Que Publishing, 2015.
- [4] Rosch, W. L., *Hardware Bible*, Sixth Edition, Que Publishing, 2003.
- [5] Stallings, W., *Computer Organization and Architecture. Designing for Performance*, Ninth Edition, Pearson Education, 2013.
- [6] InterNational Committee for Information Technology Standards, “Information Technology – SCSI Architecture Model – 6 (SAM-6)”, Project T10/BSR INCITS 546, Revision 07, February 17, 2021.
- [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 – MultiMedia Command Set – 6 (MMC-6)”, Project T10/BSR INCITS 468, Revision 02g, December 11, 2009.
- [9] InterNational Committee for Information Technology Standards, “Information Technology – SCSI Primary Commands – 6 (SPC-6)”, Project T10/BSR INCITS 566, Revision 5, March 8, 2021.