

Avantajele magistralei VL Bus constau în simplitatea arhitecturii și costul redus. Dezavantajul principal este că la apariția unor procesoare mai performante, logica de interfață trebuie reproiectată pentru acestea. De asemenea, faptul că nu se definește formatul registrelor de configurație și adresele acestora nu permite realizarea unor interfețe software unitare pentru dispozitivele conectate la magistrală.

### 3.5.8. Magistrala PCI

#### 3.5.8.1. Prezentare generală

În anul 1992, firma *Intel* a anunțat prima versiune (1.0) a unei specificații pentru o magistrală locală, denumită PCI (*Peripheral Component Interconnect*). Intenția inițială era realizarea unui standard care să permită interconectarea circuitelor rapide pe placa de bază, deoarece cu fiecare generație de microprocesoare, *Intel* trebuia să modifice arhitectura magistralei locale pentru a mări performanțele sistemului, ceea ce implica și modificarea circuitelor de interfață pentru periferice.

Magistrala PCI urma să se conecteze la magistrala locală a procesorului prin intermediul unui circuit, special destinat acestui scop. Astfel, la fiecare schimbare a procesorului și a magistralei locale, trebuia schimbat doar circuitul de legătură, circuitele de interfață cu perifericele nefiind afectate.

Specificația PCI inițială nu prevedea conectori de extensie. *Intel* a actualizat specificațiile PCI pentru ca aceasta să admită și conectori de extensie. Astfel, magistrala PCI a fost definită detaliat din punct de vedere electric și funcțional, ajungând cea mai utilizată arhitectură de magistrală. Versiunea 2.0 a apărut în 1993, iar versiunea 2.1 în 1995.

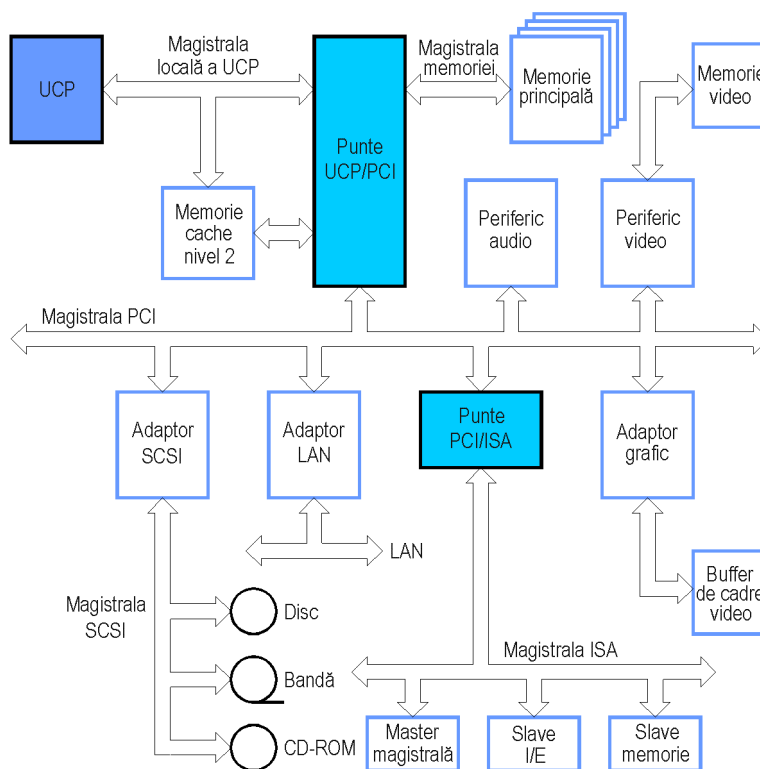
Sunt posibile transferuri de 32 sau de 64 biți. În versiunea 2.1, frecvența maximă a ceasului este de 66 MHz, ceea ce permite obținerea unei rate maxime de transfer de 528 MB/s.

Standardul de extensie a plăcii PCI definește o familie de conectori pentru adaptoare de 5 V sau 3,3 V, cu variante de 32 și 64 de biți. Este de așteptat ca în viitor majoritatea conectorilor de extensie să utilizeze surse de alimentare de 3,3 V în loc de 5 V [14].

Specificațiile PCI permit utilizarea a două din cele trei metode de conectare la magistrala locală a procesorului: conectarea printr-un buffer și conectarea de tip stație de lucru. Datorită avantajelor legate de performanțe și de flexibilitate, conectarea de tip stație de lucru este preferată. Se prezintă în Figura 3.20 schema bloc a unui sistem cu magistrală PCI, utilizând acest tip de conectare.

Magistrala PCI este conectată la magistrala sistem a procesorului prin intermediul unei punți UCP/PCI, având acces direct la memoria principală. În acest fel, transferurile între UCP și memoria cache, respectiv între dispozitivele de I/E și memoria principală pot avea loc simultan. Dispozitivele de I/E de viteză ridicată, ca adaptoarele grafice și adaptoarele de rețea, care solicită în măsură redusă procesorul, sunt conectate direct la magistrala PCI. Dispozitivele care trebuie să se conformeze altor standarde de magistrală, ca ISA sau SCSI, se interfațează cu magistrala PCI printr-o punte PCI/ISA, respectiv un adaptor SCSI.

Un sistem poate fi realizat și fără utilizarea punților de legătură. În acest caz, toate componentele, inclusiv procesorul și memoria principală, se interfațează direct cu magistrala PCI.

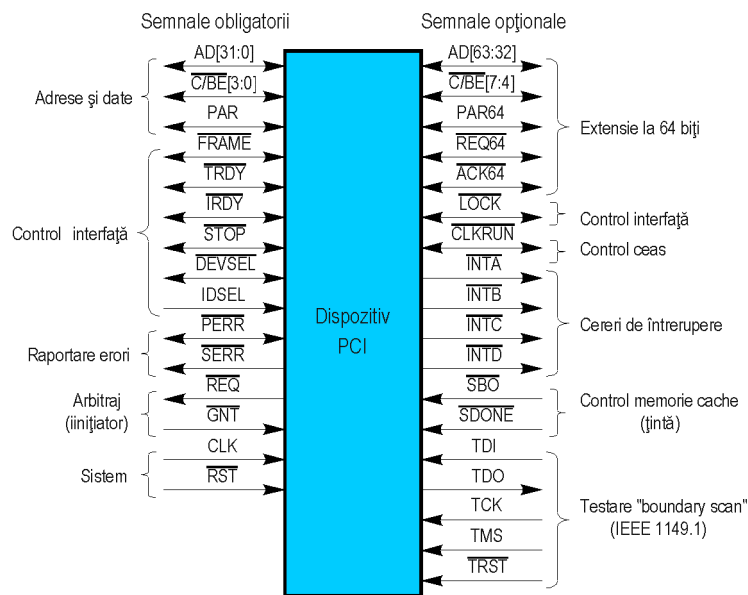


**Figura 3.20.** Schema bloc a unui sistem cu magistrală PCI.

### 3.5.8.2. Semnalele magistralei PCI

Fiecare dispozitiv conectat la magistrala PCI poate funcționa fie ca *slave*, fie ca *master/slave*. Un dispozitiv *master* se mai numește *inițiator*, iar unul *slave* se mai numește *țintă (target)*. Un dispozitiv țintă nu poate iniția transferuri pe magistrală.

Pentru a reduce numărul pinilor și dimensiunea conectorilor necesari pentru unitățile compatibile PCI, adresele și datele sunt multiplexate, setul de linii pentru linii și date fiind notat cu *AD*. O tranzacție tipică pe magistrală implică două faze. În prima fază se transmite o adresă pe liniile *AD*, iar în faza a doua se transmit una sau mai multe cuvinte de date pe aceleași linii. Toate operațiile magistralei sunt sincronizate cu ajutorul unui semnal de ceas, astfel că magistrala este sincronă. Sunt prevăzute însă semnale care permit inserarea stărilor de așteptare de către dispozitivele mai lente.



**Figura 3.21.** Semnalele unui dispozitiv PCI.

În Figura 3.21 se prezintă principalele semnale ale unui dispozitiv *master/slave*. În partea stângă se află semnalele obligatorii, necesare pentru transferurile de bază utilizând cuvinte de maxim 32 de biți. În partea dreaptă se află semnalele opționale, care permit transferuri de date utilizând cuvinte de 64

de biți, semnale care permit controlul întreruperilor și executarea altor funcții mai puțin uzuale.

Se descriu în continuare principalele semnale [16].

- $AD[31:0]$  (*Address/Data*)  
Reprezintă magistrala multiplexată de adrese și de date. În timpul fazei de adrese, pe această magistrală se transmite adresa de start a tranzacției. În timpul fazei de date, pe liniile  $AD[31:0]$  se transmit date provenite de la inițiator (la o scriere) sau de la ținta adresată (la o citire).
- $\overline{C/BE}$  [3:0] (*Command or Byte Enable*)  
În timpul fazei de adrese aceste linii definesc o comandă pe care inițiatorul o utilizează pentru a indica tipul tranzacției necesare. Dintre comenzile posibile se amintesc citirea din memorie, scrierea în memorie, citirea de la un dispozitiv de I/E, scrierea la un dispozitiv de I/E, achitarea unei întreruperi etc. În timpul fazei de date aceste linii sunt utilizate de inițiator pentru a indica octeții care trebuie transferați din cadrul cuvântului dublu adresat și grupele de linii ale magistralei  $AD$  care trebuie utilizate pentru transferul datelor.
- $PAR$  (*Parity*)  
Reprezintă semnalul de paritate pară pentru liniile  $AD[31:0]$  și  $\overline{C/BE}$  [3:0]. Paritatea este generată de inițiator după terminarea fazei de adrese și după terminarea fiecărei faze de date la o tranzacție de scriere. Paritatea este generată de ținta adresată după terminarea fiecărei faze de date la o tranzacție de citire.
- $\overline{FRAME}$  (*Cycle Frame*)  
Este activat de inițiator și indică începutul și durata unei tranzacții pe magistrală. O tranzacție poate consta din una sau mai multe transferuri de date între inițiatorul curent și ținta adresată. Semnalul  $\overline{FRAME}$  este dezactivat atunci când inițiatorul este pregătit să încheie faza finală de date.
- $\overline{TRDY}$  (*Target Ready*)  
Este activat de dispozitivul țintă adresat atunci când acest dispozitiv este pregătit pentru un transfer de date (poate executa faza curentă de date). Faza de date este terminată atunci când ținta activează  $\overline{TRDY}$  și inițiatorul activează  $\overline{IRDY}$  la frontul crescător al semnalului de ceas. În timpul unei operații de citire, semnalul  $\overline{TRDY}$  activat indică faptul că ținta a depus date valide pe magistrala de date. În timpul unei operații de scriere, semnalul  $\overline{TRDY}$  activat indică faptul că ținta este pre-

gătită să accepte datele de la inițiator. Sunt inserate stări de așteptare în faza curentă de date până când ambele semnale  $\overline{TRDY}$  și  $\overline{IRDY}$  sunt activate.

- $\overline{IRDY}$  (*Initiator Ready*)  
Este activat de inițiatorul tranzacției și semnalează momentul în care inițiatorul este pregătit pentru un transfer de date. În timpul unei operații de scriere, semnalul  $\overline{IRDY}$  activat indică faptul că inițiatorul a depus date valide pe magistrala de date. În timpul unei operații de citire, semnalul  $\overline{IRDY}$  activat indică faptul că inițiatorul este pregătit să accepte datele de la ținta adresată.
- $\overline{STOP}$   
Prin activarea acestui semnal dispozitivul țintă solicită inițiatorului oprirea tranzacției în curs în timpul fazei curente de date.
- $\overline{DEVSEL}$  (*Device Select*)  
Este activat de dispozitivul țintă atunci când acesta și-a decodificat adresa. Dacă un dispozitiv *master* a inițiat un transfer și nu detectează semnalul  $\overline{DEVSEL}$  activ în cursul a șase perioade de ceas, va presupune că ținta nu poate răspunde sau că adresa respectivă nu este utilizată. Va rezulta un abandon din partea dispozitivului *master*.
- $\overline{IDSEL}$  (*Initialization Device Select*)  
Este o intrare pentru un dispozitiv PCI și este utilizat pentru selecția circuitului în timpul accesului la unul din registrele de configurare ale dispozitivului.
- $\overline{PERR}$  (*Parity Error*)  
Indică o eroare de paritate. Generarea informației de paritate este obligatorie pentru toate dispozitivele PCI care transmit adrese și date pe magistrala *AD*. Detectarea și raportarea erorilor de paritate de către dispozitivele PCI este necesară în general (în anumite cazuri, erorile de paritate pot fi ignorate). Dacă se detectează o eroare de paritate de către un dispozitiv țintă în timpul unei faze de date la o operație de scriere, ținta trebuie să activeze semnalul  $\overline{PERR}$ . Ținta poate continua tranzacția sau poate activa semnalul  $\overline{STOP}$  pentru terminarea prematură a tranzacției. Dacă se detectează o eroare de paritate de către un dispozitiv inițiator în timpul unei faze de date la o operație de citire, inițiatorul trebuie să activeze semnalul  $\overline{PERR}$ . Inițiatorul unei tranzacții are responsabilitatea de a raporta detectarea unei erori de paritate. În acest scop, inițiatorul trebuie să monitorizeze semnalul  $\overline{PERR}$

în timpul fazelor de date la operațiile de scriere pentru a determina dacă ținta a detectat o eroare de paritate a datelor.

- $\overline{SERR}$  (*System Error*)  
Acest semnal poate fi activat de orice dispozitiv PCI pentru a raporta erori de paritate a adreselor, erori de paritate a datelor în timpul unui ciclu special, sau erori critice diferite de cele de paritate. Acest semnal este considerat ca ultima modalitate de raportare a erorilor serioase. Erorile care nu sunt catastrofale și cele corectabile trebuie semnalate în alte moduri. La calculatoarele PC,  $\overline{SERR}$  determină în general o întrerupere NMI către procesorul sistemului.
- $\overline{REQ}$  (*Request*)  
Este activat de inițiator pentru a indica o cerere de magistrală. Această linie este conectată la arbitrul de magistrală. Metoda de arbitrare a magistralei nu este descrisă în specificațiile PCI; se pot implementa diferite metode. Este specificat doar faptul că arbitrul de magistrală trebuie să utilizeze un algoritm prin care să se evite blocajele; fiecărui dispozitiv *master* potențial trebuie să i se permită accesul la magistrală.
- $\overline{GNT}$  (*Grant*)  
Este activat de arbitrul de magistrală pentru a indica acordarea magistralei pentru inițiator. Atunci când detectează acest semnal, inițiatorul trebuie să aștepte terminarea tranzacției în curs de către inițiatorul curent.
- $CLK$  (*Clock*)  
Reprezintă semnalul de ceas utilizat pentru sincronizarea tuturor tranzacțiilor, inclusiv a arbitrajului de magistrală. Toți parametrii de sincronizare ai magistralei sunt specificați relativ la frontul crescător al semnalului de ceas. Frecvența semnalului de ceas se poate modifica în orice moment, cu condiția să nu existe cereri de magistrală și semnalul  $\overline{LOCK}$  să nu fie activ. De asemenea, ceasul poate fi oprit în starea *low* (pentru reducerea puterii consumate).
- $\overline{RST}$  (*Reset*)  
Reprezintă semnalul de resetare. Atunci când este activat, acest semnal forțează trecerea tuturor registrelor de configurare PCI, a dispozitivelor inițiator și țintă într-o stare de inițializare. În general, semnalele de ieșire ale dispozitivelor PCI vor fi trecute în starea de înaltă impedanță. Semnalul  $\overline{RST}$  poate fi activat sau dezactivat asincron relativ la frontul semnalului  $CLK$ .

- $AD[63:32]$   
Reprezintă extensia la 64 de biți a magistralei de date  $AD[31:0]$ . Aceste linii nu sunt utilizate în timpul fazei de adresare a unui transfer (cu excepția cazului în care se utilizează adresarea pe 64 de biți).
- $\overline{C/BE}$  [7:4]  
Reprezintă extensia semnalelor  $\overline{C/BE}$  [3:0]. Se utilizează numai în timpul fazei de date a unui transfer (cu excepția cazului în care se utilizează adresarea pe 64 de biți).
- $PAR64$  (*Parity for the Upper Doubleword*)  
Este bitul de paritate pară asociat cu  $AD[63:32]$  și  $\overline{C/BE}$  [7:4].
- $\overline{REQ64}$  (*Request 64-bit Transfer*)  
Semnal generat de inițiatorul curent pentru a indica faptul că dorește executarea transferurilor utilizând extensia la 64 de biți a liniilor de date. Semnalul  $\overline{REQ64}$  are aceeași sincronizare ca și semnalul  $\overline{FRAME}$ .
- $\overline{ACK64}$  (*Acknowledge 64-bit Transfer*)  
Semnal generat de ținta curent adresată (dacă aceasta permite transferuri pe 64 de biți) ca răspuns la activarea semnalului  $\overline{REQ64}$  de către inițiator. Semnalul  $\overline{ACK64}$  are aceeași sincronizare ca și semnalul  $\overline{DEVSEL}$ .
- $\overline{LOCK}$   
Acest semnal este utilizat de către un inițiator PCI care solicită accesul exclusiv la un dispozitiv țintă de memorie în timpul a două sau mai multe tranzacții separate. Acest semnal este prevăzut cu scopul de a permite operații de citire/modificare/scriere la operațiile cu semafoare. Dacă un dispozitiv PCI conține o memorie executabilă sau o memorie care conține date de sistem (gestionate de sistemul de operare), trebuie să implementeze această funcție de blocare a resurselor.
- $\overline{CLKRUN}$  (*Clock Run*)  
Acest semnal este opțional și este definit pentru sistemele portabile. Un sistem portabil cuprinde o resursă centrală, care include logica de generare a ceasului pentru magistrala PCI. Atunci când ceasul funcționează normal, această logică menține semnalul  $\overline{CLKRUN}$  activat. În perioadele în care ceasul este oprit sau frecvența acestuia este micșorată, logica de generare a ceasului monitorizează semnalul  $\overline{CLKRUN}$

pentru a recunoaște cereri de la dispozitivele PCI care urmăresc modificarea stării semnalului de ceas. Ceasul nu poate fi oprit dacă magistrala nu este liberă. Înaintea opririi (sau micșorării frecvenței) ceasului, logica de generare a ceasului dezactivează semnalul  $\overline{CLKRUN}$  pentru o perioadă de ceas pentru a informa dispozitivele PCI că ceasul urmează să fie oprit (sau frecvența acestuia urmează să fie micșorată). După aceasta, logica trece driverul de ieșire al semnalului  $\overline{CLKRUN}$  în starea de înaltă impedanță. Rezistența conectată la linia  $\overline{CLKRUN}$  va menține acest semnal în starea dezactivată în timpul în care ceasul este oprit. Ulterior, un dispozitiv *master* poate solicita repornirea ceasului în scopul utilizării magistralei prin activarea semnalului  $\overline{CLKRUN}$ .

- $\overline{INTA}$ ,  $\overline{INTB}$ ,  $\overline{INTC}$ ,  $\overline{INTD}$   
Reprezintă semnalele pentru cererile de întrerupere.
- $\overline{SBO}$  (*Snoop Back Off*)  
Acest semnal este o ieșire de la puntea UCP/PCI și o intrare pentru subsistemele de memorie conectate la magistrala PCI care pot utiliza o memorie *cache*. Semnalul  $\overline{SBO}$  este activat de puntea UCP/PCI pentru a indica faptul că prin accesul curent la memorie urmează să se citească sau să se actualizeze informații perimate din memorie. Semnalul  $\overline{SBO}$  are semnificație numai atunci când semnalul  $\overline{SDONE}$  este de asemenea activat de către puntea UCP/PCI. Atunci când ambele semnale sunt activate, subsistemul de memorie PCI curent adresat trebuie să răspundă prin semnalarea unei retransmisii la inițiatorul curent.
- $\overline{SDONE}$  (*Snoop Done*)  
Acest semnal este o ieșire de la puntea UCP/PCI și o intrare pentru subsistemele de memorie conectate la magistrala PCI care pot utiliza o memorie *cache*. Semnalul  $\overline{SDONE}$  este dezactivat de puntea UCP/PCI în timp ce memoria *cache* a procesorului intervine într-o cerere de acces la memorie executată de inițiatorul curent. Puntea activează semnalul  $\overline{SDONE}$  atunci când intervenția s-a terminat. Rezultatele intervenției sunt indicate de către semnalul  $\overline{SBO}$ . Semnalul  $\overline{SBO}$  dezactivat indică faptul că inițiatorul PCI accesează date valide din memorie și că memoria poate accepta sau furniza datele respective. Semnalul  $\overline{SBO}$  activat indică faptul că inițiatorul PCI accesează date perimate din memorie și nu trebuie să termine accesul curent. Memoria va trebui să termine accesul prin semnalarea faptului că operația trebuie repetată.



- *TCK (Test Clock)*  
Utilizat ca semnal de ceas pentru transferul informațiilor de stare și a datelor în timpul testării “*boundary scan*”.
- *TDI (Test Data Input)*  
Utilizat împreună cu semnalul *TCK* pentru transferul datelor și a instrucțiunilor către dispozitivul PCI prin portul TAP (*Test Access Port*) sub forma unui șir de biți, în mod serial.
- *TDO (Test Data Out)*  
Utilizat împreună cu semnalul *TCK* pentru transferul datelor de la dispozitivul PCI prin portul TAP sub forma unui șir de biți, în mod serial.
- *TMS (Test Mode Select)*  
Utilizat pentru controlul stării controlerului portului TAP.
- $\overline{TRST}$  (*Test Reset*)  
Utilizat pentru a forța controlerul portului TAP în starea inițială.

### 3.5.8.3. Caracteristici ale magistralei PCI

#### Dispozitiv inițiator și țintă

La fiecare transfer participă două dispozitive: inițiatorul și ținta. *Inițiatorul*, sau dispozitivul *master*, este cel care inițiază un transfer. *Ținta*, sau dispozitivul *slave*, este adresat de inițiator în scopul executării transferului. Dispozitivele inițiator și țintă sunt numite *agenți PCI*.

#### Transferuri în mod exploziv

Un asemenea transfer constă dintr-o singură fază de adrese urmată de mai multe faze de date. Arbitrajul de magistrală trebuie executat o singură dată. În timpul fazei de adrese se transmite adresa de început și tipul tranzacției care urmează. Dispozitivul țintă memorează adresa de început într-un contor de adrese, și va incrementa adresa în fiecare fază de date.

La magistralele EISA și Micro Channel, posibilitatea de a utiliza modul exploziv este stabilită printr-o negociere între dispozitivul *master* și cel *slave*. Dacă unul sau niciunul din acestea nu permite transferuri în mod exploziv, blocul de date se va transmite utilizând o serie de tranzacții separate. Este necesar arbitrajul de magistrală pentru fiecare tranzacție individuală. Un alt dispozitiv *master* poate obține magistrala între două tranzacții. Aceasta poate limita în mod sever rata de transfer.

În cazul magistralei PCI, cele mai multe transferuri se execută în mod exploziv, cele mai multe dispozitive fiind proiectate astfel încât să permită asemenea transferuri. Dacă un dispozitiv țintă poate efectua numai tranzacții singulare, la încercarea dispozitivului *master* de a executa o tranzacție în mod exploziv, ținta termină tranzacția după prima fază de date. Astfel dispozitivul *master* va fi obligat să solicite din nou magistrala printr-un proces de arbitraj. În acest mod performanțele vor fi mai reduse, dar tranzacțiile singulare se pot utiliza pentru dispozitivele care nu necesită rate ridicate de transfer.

Presupunând că nici inițiatorul și nici dispozitivul țintă nu inserează stări de așteptare, poate fi transferat câte un cuvânt la fiecare front crescător al semnalului de ceas. La o frecvență de 33 MHz, se poate obține o rată de transfer de 132 MB/s în cazul implementării pe 32 de biți, sau 264 MB/s pentru 64 de biți. Cu o frecvență a ceasului de 66 MHz, se pot obține rate de transfer de 264 MB/s, respectiv 528 MB/s.

### Dispozitive multifuncționale

Un dispozitiv fizic PCI poate fi o componentă integrată pe placa sistem sau o placă de extensie PCI. Fiecare dispozitiv poate încorpora până la opt funcții separate, un asemenea dispozitiv fiind numit multifuncțional.

### Ceasul magistralei PCI

Frecvența semnalului de ceas poate fi în intervalul de la 0 la 33 MHz. Specificațiile versiunii 1.0 indică faptul că toate dispozitivele trebuie să funcționeze între 16 și 33 MHz, recomandând funcționarea la 0 MHz. Specificațiile versiunilor 2.x indică faptul că toate dispozitivele trebuie să funcționeze în intervalul de la 0 la 33 MHz. Funcționarea la 0 MHz asigură un consum redus de putere și posibilitatea depanării statice. Frecvența ceasului se poate modifica în orice moment și ceasul poate fi oprit (în starea *low*).

Versiunea 2.1 definește de asemenea funcționarea la frecvențe de până la 66 MHz.

#### 3.5.8.4. Tranzacții

##### Faza de adrese

Toate tranzacțiile pe magistrala PCI constau dintr-o *fază de adrese* urmată de una sau mai multe *faze de date*. Excepția o reprezintă tranzacțiile în care se utilizează adresarea pe 64 de biți, unde adresa este fur-

nizată în două faze de adrese. O fază de adrese are durata unei perioade de ceas. În această fază, inițiatorul identifică dispozitivul țintă și tipul tranzacției.

Ținta este identificată prin transmiterea unei adrese de start pe magistrala de adrese/date în cadrul domeniului de adrese asignat dispozitivului. Tipul tranzacției este identificat prin depunerea tipului comenzii pe liniile  $C/BE$ . Inițiatorul activează semnalul  $\overline{FRAME}$  pentru a indica prezența unei adrese de start și tip de tranzacție valide. Deoarece adresa de start este prezentă numai pe durata unei perioade de ceas, fiecare dispozitiv țintă trebuie să memoreze adresa pentru a putea fi decodificată ulterior.

Atunci când un dispozitiv țintă determină faptul că este adresat, trebuie să activeze semnalul  $\overline{DEVSEL}$ , pentru a revendica tranzacția. Dacă acest semnal nu va fi activat într-un timp predeterminat, inițiatorul va abandona tranzacția.

După terminarea fazei de adrese, magistrala de adrese/date este utilizată pentru transferul datelor în cadrul unei sau mai multor faze de date.

### Faza de date

Pe durata fazei sau fazelor de date are loc transferul datelor între inițiator și țintă. Atât inițiatorul cât și ținta trebuie să indice faptul că sunt pregătite pentru terminarea unei faze de date prin activarea semnalelor  $\overline{IRDY}$ , respectiv  $\overline{TRDY}$ . În cazul în care nu sunt activate ambele semnale la începutul unei perioade de ceas, se va insera o stare de așteptare cu durata unei perioade.

### Terminarea tranzacției

Inițiatorul identifică durata totală a unui transfer cu ajutorul semnalului  $\overline{FRAME}$ . Acest semnal este activat la începutul fazei de adrese și rămâne activat până când inițiatorul este pregătit pentru a termina faza finală de date (activează  $\overline{IRDY}$ ).

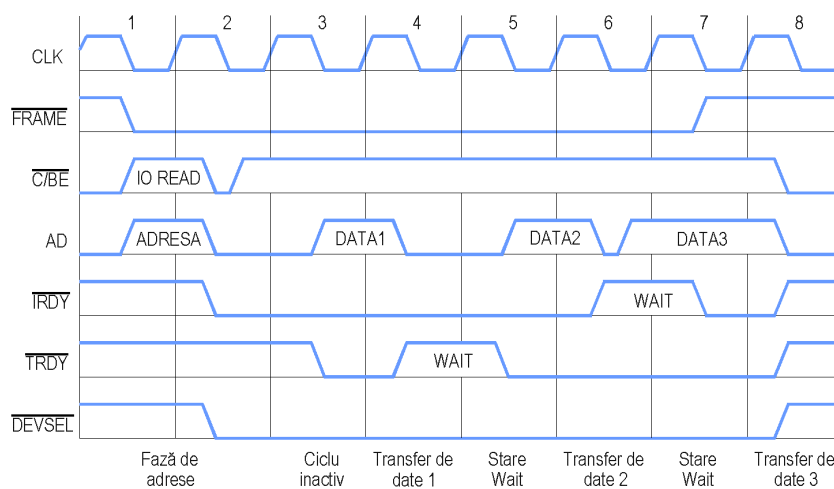
Executarea ultimei faze de date este indicată de inițiator prin dezactivarea semnalului  $\overline{FRAME}$  și activarea semnalului  $\overline{IRDY}$ . După terminarea ultimului transfer, inițiatorul readuce magistrala în starea inactivă prin dezactivarea semnalului  $\overline{IRDY}$ .

Arbitrajul de magistrală are loc în timp ce un dispozitiv *master* utilizează magistrala. Dacă arbitrul de magistrală a acordat în prealabil magistrala unui alt dispozitiv *master*, acest dispozitiv poate detecta sta-

rea inactivă a magistralei prin faptul că semnalele  $\overline{FRAME}$  și  $\overline{IRDY}$  sunt ambele dezactivate.

### Exemplu de tranzacție

Figura 3.22 prezintă un transfer de date reprezentativ de la un dispozitiv *slave* la un dispozitiv *master*, de exemplu o operație de citire de la un dispozitiv de I/E [6].



**Figura 3.22.** Tranzacție de citire pe magistrala PCI.

Tranzacția începe în momentul în care inițiatorul (care se presupune că are în permanență controlul asupra magistralei) activează semnalul  $\overline{FRAME}$  în ciclul de ceas 1. Inițiatorul plasează apoi o adresă și un cuvânt de comandă (IO READ în acest exemplu) pe liniile  $AD$ , respectiv  $\overline{C/BE}$ ; aceste informații trebuie să fie valide în momentul în care începe ciclul de ceas 2. În timpul ciclului 2 toate unitățile conectate la magistrală încearcă să decodifice adresa și comanda. În acest exemplu, o unitate de I/E își va recunoaște adresa. În continuare inițiatorul va plasa informații valide pe liniile  $\overline{C/BE}$  pe durata restului tranzacției. Pentru evitarea conflictelor în momentul în care inițiatorul nu mai are controlul asupra liniilor  $AD$  și acest control este preluat de dispozitivul țintă, faza de adrese este urmată de un ciclu inactiv (ciclul 3 în Figura 3.22). Dispozitivul țintă poate transmite o secvență de cuvinte de date începând din ciclul 4 cu rata maximă de un cuvânt pe ciclu de ceas. Cele două dispozitive controlează

rata de transfer reală prin liniile  $\overline{IRDY}$  și  $\overline{TRDY}$ , care permit inserarea unui număr oarecare de stări de așteptare după fiecare ciclu de transfer de date.

Transferul de date nu poate începe până când inițiatorul nu activează semnalul  $\overline{IRDY}$  pentru a indica faptul că este pregătit pentru recepționarea datelor (ciclul 2). Dispozitivul țintă furnizează cuvântul 1 de date și semnalează acest lucru prin activarea semnalului  $\overline{TRDY}$  în ciclul 3. Transferul de date are loc în ciclul 4. În acest exemplu dispozitivul țintă dezactivează imediat linia  $\overline{TRDY}$ , ciclul 5 devenind o stare de așteptare; apoi reactivează  $\overline{TRDY}$  și plasează cuvântul 2 de date pe liniile  $AD$  pentru a fi transmis în ciclul 6. Dispozitivul țintă plasează apoi cuvântul 3 de date pe liniile  $AD$ , cuvânt care va fi transmis în ciclul 7. De această dată, inițiatorul va insera o stare de așteptare prin dezactivarea semnalului  $\overline{IRDY}$  pe durata unui ciclu de ceas. În consecință, transferul cuvântului 3 de date este amânat până în ciclul 8. Inițiatorul dezactivează semnalul  $\overline{FRAME}$  în ciclul 7 pentru a indica faptul că următorul ciclu marchează sfârșitul tranzacției pe magistrală. Linia de control  $\overline{DEVSEL}$  este activată de dispozitivul țintă în ciclul 2 pentru a indica faptul că acest dispozitiv a decodificat cu succes adresa și este ținta tranzacției curente. Nici un transfer de date nu poate avea loc până în momentul în care semnalul  $\overline{DEVSEL}$  nu este activ.

O tranzacție de scriere (în care inițiatorul este sursa datelor) este similară cu cea din Figura 3.22. În acest caz ciclul inactiv după faza de adrese nu este necesar, deoarece inițiatorul menține în continuare controlul asupra liniilor  $AD$  pe durata tranzacției.

### 3.5.8.5. Întreruperi

Magistrala PCI are patru linii pentru cererile de întrerupere,  $\overline{INTA}$ , ...,  $\overline{INTD}$ ; acestea fac parte din liniile opționale ale magistralei. Un dispozitiv de I/E cu o singură funcție trebuie să utilizeze  $\overline{INTA}$  ca linie pentru cererea de întrerupere; dispozitivele de I/E multifuncționale pot utiliza toate cele patru linii [6]. O configurație particulară pe liniile de comandă  $\overline{C/BE}$  [3:0] indică achitarea întreruperii. Liniile cererilor de întrerupere  $\overline{INTx}$  și comanda de achitare a întreruperii pot implementa împreună semnalele necesare în timpul unei tranzacții de întrerupere pe magistrala PCI.

Fiecare dispozitiv compatibil PCI trebuie să dispună de un set standard de registre de configurație adresabile CR care identifică dispozitivul. La inițializare, sistemul de operare citește registrele CR pentru a determina, printre altele, conexiunile de întrerupere ale dispozitivului de I/E. Registrul "pinului de între-

rupere” de 8 biți din CR indică linia cererii de întrerupere  $\overline{INTx}$  care este utilizată de dispozitiv. Un alt registru de 8 biți din CR, numit registrul “liniei de întrerupere”, specifică linia de intrare a controlerului de sistem care este conectată la linia  $\overline{INTx}$ , astfel încât rutarea liniilor pentru cererile de întrerupere este programabilă. Controlerul de sistem poate utiliza această informație pentru a determina prioritatea cererii de întrerupere a dispozitivului de I/E și pentru accesarea vectorilor săi de întrerupere. Registrele CR formează un mic spațiu de adresă care este separat de spațiul de adresă al memoriei și cel de I/E.

### 3.5.8.6. Registre de configurație

Fiecare dispozitiv funcțional PCI dispune de un bloc de 64 cuvinte duble rezervate pentru implementarea registrelor de configurație. Formatul primelor 16 cuvinte duble este predefinit de specificațiile PCI. Această zonă reprezintă *antetul de configurație* al dispozitivului. Specificațiile definesc două formate ale antetului, numite tipul 0 și 1. Tipul 1 este definit pentru punțile de legătură între două magistrale PCI, la sistemele cu mai multe magistrale PCI. Tipul 0 este utilizat de alte dispozitive, fiind descris în continuare.

Formatul antetului de configurație al unui dispozitiv funcțional PCI (diferit de o punte PCI-PCI) este prezentat în Figura 3.23. Registrele din acest antet sunt utilizate pentru identificarea dispozitivului, controlul funcțiilor sale și detectarea stării acestuia. Utilizarea celorlalte 48 de cuvinte duble a spațiului de configurație este specifică dispozitivului.

În continuare se descriu registrele obligatorii. Aceste registre trebuie implementate în fiecare dispozitiv PCI, inclusiv în punțile de legătură [16].

#### Registrul de identificare a producătorului

Acest registru de 16 biți, de tip R/O, identifică producătorul dispozitivului printr-un cod atribuit de o autoritate care controlează atribuirea acestor coduri. Valoarea FFFFh a codului este rezervată și trebuie returnată de puntea dintre UCP și magistrala PCI atunci când se încearcă citirea registrului de identificare al producătorului dintr-un dispozitiv inexistent.

#### Registrul de identificare a dispozitivului

Valoarea de 16 biți din acest registru este asignată de producător și identifică tipul dispozitivului.

Octet				Adresă de cuvânt dublu
3	2	1	0	
ID dispozitiv		ID producător		00
Registru de stare		Registru de comandă		01
Cod clasă			ID revizie	02
BIST	Tip antet	Contor întârziere	Dimens. linie cache	03
Adresa de bază 0				04
Adresa de bază 1				05
Adresa de bază 2				06
Adresa de bază 3				07
Adresa de bază 4				08
Adresa de bază 5				09
Rezervat				0A
ID subsistem		ID producător subsist.		0B
Adresă de bază memorie ROM de extensie				0C
Rezervat				0D
Rezervat				0E
		Pin într.	Linie într.	0F

**Figura 3.23.** Antetul de configurație al unui dispozitiv funcțional PCI.

### Registru de comandă

Este un registru de 16 biți, fiind definiți numai biții 9-0, ceilalți biți fiind rezervați pentru utilizări viitoare. Se descriu funcțiile principale care pot fi specificate de biții acestui registru.

- *Validarea dispozitivului.*
- *Validarea accesului la memoria dispozitivului*, dacă acesta implementează o memorie.
- *Validarea funcției de master a dispozitivului*, dacă acesta are posibilitatea de a deveni master.
- *Validarea monitorizării ciclurilor speciale PCI.* Aceste cicluri se utilizează pentru a transmite diferite mesaje dispozitivelor țintă, care pot monitoriza ciclurile pentru a determina dacă mesajele sunt destinate acestora.

- *Actualizarea registrelor de paletă a culorilor ale unei interfețe VGA fără confirmarea acceptării datelor (VGA Color Palette Snooping).* Această opțiune este utilă în cazul sistemelor cu două controlere grafice: o interfață compatibilă VGA și un alt controler grafic. Fiecare din acestea implementează un set de registre de paletă, care se află la aceleași adrese de I/E. Atunci când procesorul execută o scriere pentru actualizarea registrelor de paletă, ambele controlere vor recunoaște adresa și vor răspunde prin activarea semnalelor  $\overline{DEVSEL}$  și  $\overline{TRDY}$ , rezultând o coliziune pe aceste linii.

Programul de configurare trebuie să programeze unul din controlere ca fiind o destinație activă, care răspunde la comenzile de scriere în registrele de paletă, iar celălalt ca destinație pasivă. Controlerul VGA va răspunde întotdeauna la comenzile de citire a registrelor de paletă.

- *Validarea parității.* Dacă acest bit este setat, dispozitivul va raporta erorile de paritate. În caz contrar, erorile de paritate vor fi ignorate.
- *Validarea semnalării erorilor de sistem.* Dacă acest bit este setat, dispozitivul poate activa semnalul  $\overline{SERR}$  pentru a indica o eroare de sistem. Toate dispozitivele care utilizează semnalul  $\overline{SERR}$  trebuie să implementeze acest bit. Pentru raportarea erorilor de paritate, trebuie ca acest bit și bitul de validare a erorilor de paritate să fie setați.

După resetare, toți biții registrului de comandă sunt setați la 0, cu excepția unui dispozitiv care trebuie să fie validat în timpul pornirii sistemului, deoarece este utilizat în procesul de încărcare a sistemului. Prin aceasta dispozitivul este dezactivat, dar va fi accesibil pentru configurare, urmând să fie validat de programul de configurare. Dispozitivele care trebuie să fie accesibile la pornire prin adrese fixe trebuie să implementeze bitul de validare pentru a controla recunoașterea adreselor sale fixe. Aceasta permite programului de configurare să invalideze recunoașterea adreselor fixe după pornirea sistemului, și să reconfigureze gama de adrese la care răspunde dispozitivul.

### **Registrul de stare**

Păstrează starea evenimentelor legate de magistrala PCI. Acest registru poate fi citit, scrierile fiind tratate diferit de modul obișnuit. La scriere, un bit poate fi șters, dar nu poate fi poziționat. Un bit este șters prin scrierea valorii 1. Această metodă a fost aleasă pentru a simplifica programele. După citirea stării și determinarea biților de eroare care sunt setați, programatorul șterge biții prin înscrierea valorii citite anterior din registru.



Principalele informații conținute în registrul de stare sunt următoarele:

- *Frecvența de funcționare*  
Indică posibilitatea funcționării la 66 MHz, sau la 33 MHz.
- *Posibilitatea de finiri unor caracteristici* de către utilizator  
Deși programul de configurare poate detecta prezența unui dispozitiv și poate alocă resursele necesare acestuia (spațiul de memorie, spațiul de I/E, nivelul de întrerupere, nivelul de prioritate pentru arbitraj etc.), unele caracteristici ale anumitor subsisteme nu pot fi configurate automat atunci când subsistemul este detectat pentru prima dată. În acest caz se poate indica o listă de opțiuni care pot fi selectate. Un exemplu îl constituie un adaptor de rețea căruia trebuie să i se asigneze un anumit identificator de nod al rețelei. La prima instalare, opțiunile vor fi salvate în memoria nevolatilă și vor fi utilizate în continuare.
- *Raportarea parității*  
Acest bit este implementat numai de dispozitivele *master* și este setat în următoarele condiții: dispozitivul *master* a avut rol de inițiator și a setat el însuși semnalul  $\overline{PERR}$ , sau a detectat setarea acestui semnal de către țintă (la o scriere). De asemenea bitul de validare a parității din registrul de comandă trebuie să fie setat.
- *Abandon de către țintă*  
Este un bit setat de către dispozitivul țintă atunci când aceasta termină o tranzacție cu un abandon.
- *Recepționarea abandonului de către țintă*  
Acest bit este setat de un dispozitiv *master* atunci când tranzacția inițiată de acest dispozitiv este terminată printr-un abandon al țintei curent selectate. Toate dispozitivele *master* trebuie să implementeze acest bit.
- *Semnalarea unei erori de sistem*  
Acest bit trebuie setat ori de câte ori un dispozitiv activează semnalul de eroare  $\overline{SERR}$ .
- *Detectarea unei erori de paritate*  
Acest bit trebuie setat de un dispozitiv atunci când detectează o eroare de paritate (chiar dacă raportarea erorii de paritate este invalidată).

#### **Registrul de identificare al reviziei**

Conține o valoare de 8 biți setată de producător, care reprezintă numărul de revizie al dispozitivului.

### **Registrul codului de clasă**

Este un registru de 24 de biți, divizat în trei câmpuri, care conțin: clasa de bază, subclasa și interfața de programare. Acest registru identifică funcția de bază a dispozitivului (de exemplu, un controler pentru o memorie de masă), o subclasă mai precisă a dispozitivului (de exemplu, un controler IDE), și în anumite cazuri, o interfață de programare (ca setul de registre a unui controler VGA).

### **Registrul tipului de antet**

Este un registru de 8 biți care definește prin 7 biți formatul antetului de configurație. Pe lângă antetul de tip 0, indicat în figură, este definit un singur format, cel pentru punțile între două magistrale PCI. Alte formate vor fi definite în specificațiile viitoare. Bitul 7 al acestui registru indică dacă dispozitivul are o singură funcție sau este multifuncțional.