

3. MAGISTRALE DE SISTEM

3.1. DEFINIȚIE, CLASIFICARE, STRUCTURĂ

O magistrală (bus) definește setul de reguli după care informația circulă în interiorul unui sistem de calcul, între mai multe sisteme de calcul sau între sisteme de calcul și echipamente specializate. Un standard de magistrală conține mai multe tipuri de specificații:

- mecanice - conectorii, cablurile, topologia de cuplare;
- electrice - nivelele de tensiune, curenții impedanțe;
- funcționale - descrierea semnalelor.

Unele calculatoare au impus un standard de magistrală prin popularitatea lor (PC AT au impus magistrala ISA, apoi PCI). Uneori standardele sunt create de grupuri de producători care au interese comune. Un astfel de standard este SCSI (Small Computer Interface System). Unele standarde sunt adoptate de organisme ca IEEE.

Magistralele realizează legătura electrică între procesor, memorie, dispozitive I/O și definesc cel mai jos nivel pentru un protocol de comunicație. Magistrala este o cale de comunicație folosită în comun de mai multe blocuri funcționale și este realizată fizic de un set de fire conductoare.

O magistrală trebuie să fie versatilă și să aibă un preț scăzut. Datorită versatilității, odată definită o schemă de conexiuni, este foarte ușor să se adauge noi dispozitive în sistem sau acestea să poată fi ușor mutate dintr-un calculator în altul, dacă cele două calculatoare sunt construite cu același tip de magistrală.

Pe de altă parte, folosirea unei magistrale poate fi dezavantajoasă datorită limitărilor de viteză de transfer a datelor (*throughput*) pe care aceasta le poate crea. Viteza la care poate lucra o magistrală este impusă în primul rând de factori fizici ca: lungimea traseelor și numărul dispozitivelor atașate (acestea din urmă aduc constrângeri atât ca număr efectiv cât și ca diversitate din punct de vedere al ratelor de transfer și a timpilor de întârziere).

O magistrală este organizată în grupe de semnale care au aceeași funcționalitate:

- un set de semnale de control;
- un set de semnale de date.

Semnalele de control sunt folosite pentru a indica cereri, acceptări sau pentru a indica tipul informației de pe liniile de date. Setul semnalelor de date este folosit pentru a transporta informația între blocurile funcționale ale calculatorului de la sursă la destinație (de la inițiator la țintă). Informația poate fi date, comenzi sau adrese. Adesea magistralele de date sunt despărțite în două seturi pentru a separa informația ce reprezintă adrese. Astfel există și o magistrală de adrese pentru ca datele și adresele să poată fi vehiculate într-o singură tranzacție de la sursă la destinație. După funcționalitatea semnalelor magistralele pot fi magistrale de control, magistrale de date și magistrale de adrese.

O clasificare a magistrelor poate fi făcută după blocurile care se conectează la aceasta. Conform acestui criteriu magistralele pot fi:

- magistrale procesor- memorie;
- magistrale I/O;
- magistrale de fundal.

Magistrala procesor memorie este scurtă și de mare viteză. După cum îi spune numele, este concepută pentru a lega procesoarele de memorie.

Magistralele I/O sunt în primul rând mai lungi și suportă o gamă largă de rate de transfer ca o necesitate a conectării unei game largi de dispozitive I/O la sistem. De obicei magistralele I/O nu sunt conectate direct la procesor, ci folosesc fie o magistrală procesor memorie, fie o magistrală de fundal.

Magistralele de fundal sunt proiectate astfel încât atât procesorul, memoria cât și dispozitivele I/O să folosească aceeași cale de comunicație. La acest tip de magistrală se face un compromis pentru a satisface atât necesitățile comunicației între procesor și memorie cât și ale comunicației dintre dispozitivele I/O, sacrificând desigur performanța sistemului.

Magistralele I/O și magistralele de fundal sunt definite prin specificații, deseori publice, iar magistralele procesor memorie sunt specifice (proprietatea producătorului).

O altă clasificare a magistrelor se poate face după metoda de comunicație folosită:

- magistrale sincrone;
- magistrale asincrone.

Magistrale sincrone dispun de o linie de tact în liniile de control, condusă de un oscilator cu cuarț, între 5-133MHz. Toate transferurile au loc după un protocol fix care este raportat la semnalul de tact, într-un număr întreg de cicluri, numite cicluri de bus. Pentru că protocolul este predeterminat și logica implicată este simplă, magistralele sincrone pot fi foarte rapide. Avantajul magistrelor sincrone este simplitatea deoarece nu există un dialog (procoloalele pot fi implementate prin automate finite simple). Dezavantajul este că dacă un transfer este mai scurt el trebuie totuși să dureze un număr întreg de cicluri de bus, ca urmare îmbunătățirile tehnologice ale plăcilor I/O nu duc la mărirea vitezei. Într-un cuvânt, dacă avem plăci lente și rapide pe o magistrală sincronă, transferurile au loc la viteza celei mai lente. Un alt dezavantaj este că, din cauza deformării semnalului de ceas, liniile magistralei nu pot fi foarte lungi și nici foarte încărcate.

La magistrale asincrone transferurile pot dura oricât, motiv pentru care acestea se poate adapta ușor la o mare varietate de dispozitive conectate. Coordonarea transferurilor de la sursă la destinație se face pe baza unui protocol de dialog conversațional (*handshaking*). Un astfel de protocol constă într-o serie de pași parcurși de sursă și destinație astfel încât trecerea la pasul următor se face numai cu acordul ambelor părți. Pentru implementarea unui astfel de protocol sunt necesare linii de control suplimentare (DATA REQUEST, DATA READY etc.).

Fie mai multe unități centrale și echipamente periferice EP cuplate la o magistrală, Fig. 3.1.

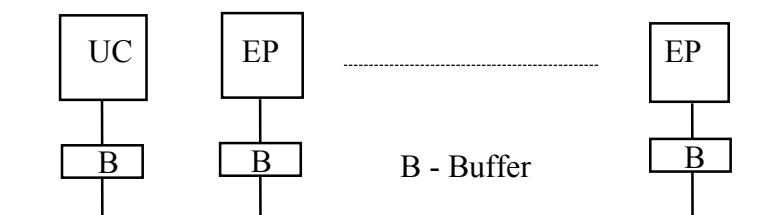


Fig. 3.1. Elemente interconectate prin magistrală

Fiecare sistem are o parte de emisie și una de recepție. Pentru ca magistrala să fie încărcată optim, fiecare sistem trebuie să conțină câte un buffer. Aceste sisteme trebuie să fie explorate ciclic sau transferul poate fi realizat printr-o logică de prioritate.

Pentru o magistrală asincronă, dacă fiecare sistem poate transfera date cu viteza medie de transfer v_i , atunci:

$$V_{\max} = \max V_i, \quad i=1-n$$

Viteza magistralei trebuie să fie mai mare decât cea mai mare viteză de transfer:

$$V_{\text{bus}} = k V_{\max}$$

La o magistrală sincronă, viteza de transfer este fixă, corespunzător celui mai lent echipament periferic. Pentru a urmări viteza magistralei, pentru echipamente periferice lente se introduc stări de WAIT. De exemplu la magistrala ISA există semnalul IOCHRDY care, atunci când este activat de un echipament periferic, introduce până la 8 stări de WAIT. Introducerea unui număr mai mare de stări de WAIT poate perturba reîmprospătarea memoriei sistemului.

Așa cum este firesc, cel puțin două blocuri sunt conectate la o magistrală. Pentru a evita haosul pe magistrală, controlul acesteia este făcut de un MASTER de magistrală. Un MASTER trebuie să poată să inițieze cereri pentru acces la magistrală și să poată să controleze toate tranzacțiile. Un procesor este capabil să fie MASTER pe când memoria poate fi numai SLAVE. Cel mai simplu sistem este un sistem cu un singur MASTER, acesta fiind procesorul. Implicarea procesorului în toate tranzacțiile este inefficientă, de aceea există alternativa de magistrală cu mai mulți MASTER. Într-o astfel de magistrală fiecare MASTER este capabil să inițieze un transfer. În aceste condiții trebuie să existe un mecanism de arbitrare a accesului la magistrală ce decide care MASTER va prelua controlul. Arbitrul primește cererea de administrare a magistralei (BUS REQUEST) pe care o onorează sau nu (GRANT) în urma unei analize. O magistrală cu mai mulți MASTER conține linii speciale pentru arbitrare. Există mai multe metode de arbitrare. Toate metodele respectă principiul priorității și al corectitudinii (*fairness*).

Pentru cuplarea diferitelor subsisteme la magistrale trebuie ținut cont de diafonie, de reflexii, de impedanțele liniilor etc. Pentru legarea semnalelor la 5V transmisie se folosesc circuite driver, așa cum sunt de exemplu 74HCxxx sau 74AHCxxx pentru CMOS și 74ABTxxx pentru TTL. Aceste circuite se pot folosi fără terminator de linie pentru că au o impedanță apropiată de impedanța caracteristică a liniei, dar se vor folosi cu o diodă pentru eliminarea vârfurilor negative sau/și cu un divizor rezistiv (180Ω, 220Ω la masă) (Fig. 3.2.).

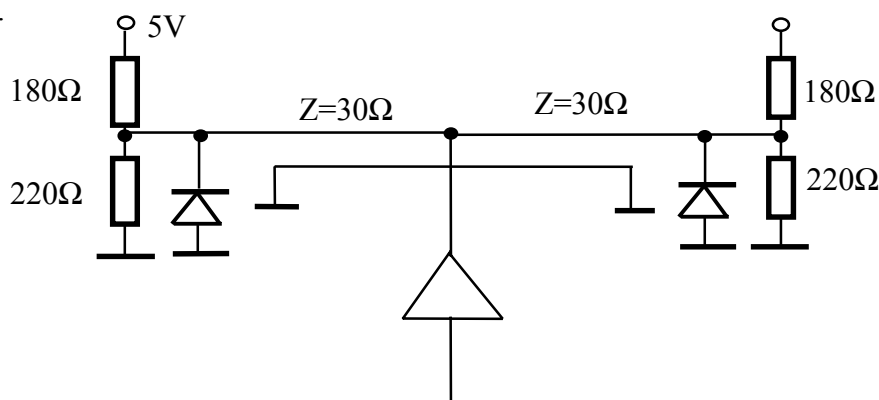


Fig. 3.2. Cuplarea la mijloc a unui driver de magistrală

Circuitele driver la capetele magistralei sau în mijloc. (Fig. 3.2. ilustrează conectare la mijloc) Se poate observa că circuitul driver din Fig. 3.2. are ca sarcină 2 linii de 30Ω în paralel, ceea ce este o sarcină mare. Este posibil ca circuitul să nu poată asigura nivelele logice corecte. La familia IWS (Incident Wave Switching), de exemplu SN74ABT25xxx, este asigurată o impedanță de ieșire de sub 2Ω și curenți $I_{OH} = -80\text{mA}$, $I_{OL} = 188\text{mA}$, ceea ce asigură nivelele corecte și în cazul Fig.2.2.

Unele magistrale moderne, cum ar fi noua magistrală VME64 trebuie să permită cuplarea subansamblelor pe magistrală în timpul funcționării. Capacitățile care apar la fiecare pin în momentul conectării modifică nivelele logice. Familia ETL (*Enhanced Transceiver Logic*), de exemplu SN74ABTExxxxx, remediază acest inconvenient, asigurând un potențial determinat și o capacitate fixă fiecărui pin neconectat.

O altă soluție pentru transmisia pe linii lungi este folosirea circuitelor driver cu colectorul în gol, de exemplu SN74BCTxxx. Această soluție nu asigură viteze mari de transfer.

Structura tipică a unei unități centrale (Fig. 3.3.) arată că magistrala este ierarhizată, în funcție de viteză, pe două nivele; unul de viteză mare (PCI) și unul de viteză mică (ISA). La magistrala de viteză mare se conectează echipamente periferice rapide (hard disc, rețea Ethernet, video, SCSI etc.) iar la cea de viteză mică se conectează echipamente periferice lente (unitatea de disc flexibil, canalele seriale, canalul paralel, etc.)

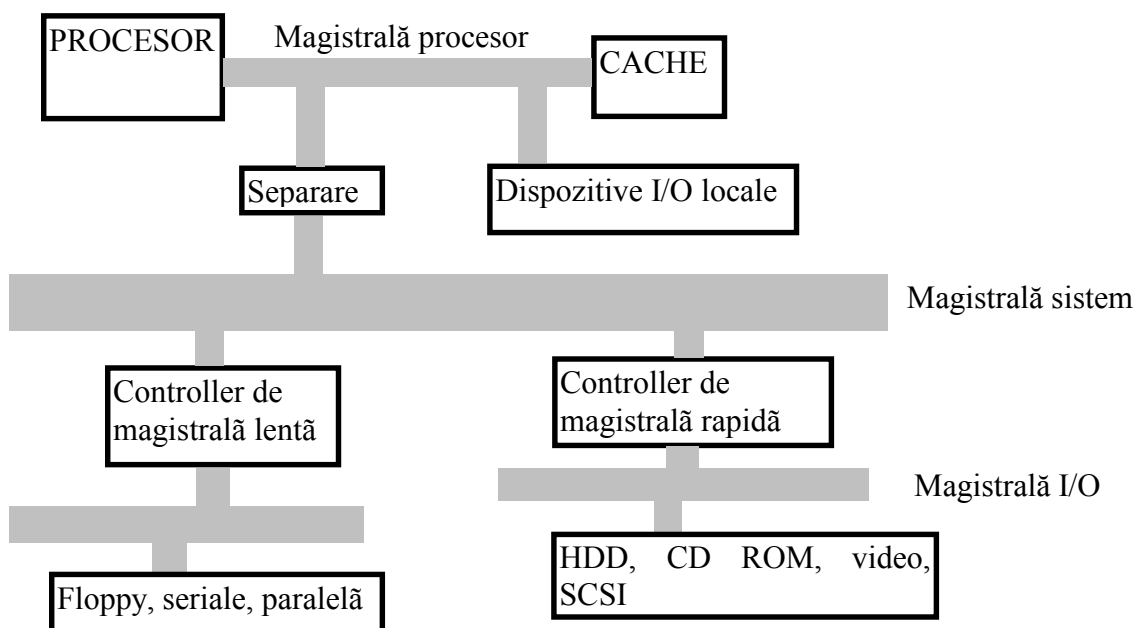


Fig. 3.3. Structura tipică a unei unități centrale

3.2. CANALE I/O

Legătura între procesor și EP (Echipamente Periferice) se realizează prin canale I/O (de intrare/ieșire) prin intermediul magistralei. Evoluția în timp a canalelor I/O este în același timp o evoluție a creșterii complexității și performanțelor. Pot fi enumerate următoarele etape:

1. CPU controlează direct EP;
2. Este adăugat un modul I/O (o interfață serială sau paralelă, programabile). CPU comandă EP prin transfer programat (direct sau prin interogare);
3. Aceeași configurație ca la 2, dar transferul are loc prin întreruperi;
4. Modulul I/O are acces direct la memorie prin DMA. Modulul poate muta informația direct în memorie, accesul CPU fiind necesar doar la începutul și sfârșitul transferului;
5. Modulul I/O folosește un microcalculator sau un microcontroller, cu instrucțiuni proprii. CPU programează procesorul I/O pentru un transfer, pe care acesta îl execută folosind instrucțiunile proprii. Când transferul se termină, procesorul I/O întrerupe CPU pentru a comunica rezultatele transferului;
6. Microcontrollerul are memorie locală. El poate controla astfel mai multe EP cu o intervenție minimă din partea CPU. Memoria locală poate fi folosită și ca buffer de date, realizând astfel o rată de transfer mare.

Evoluția microcontrollerelor integrate a fost paralelă cu cea a procesoarelor. Dacă la procesoare s-a urmărit o creștere a vitezei de prelucrare prin creșterea tactului (în jurul a 500MHz), creșterea mărimii memoriei CACHE (uzual 256K integrat), lărgimea busului de date și adrese, la microcontrolere s-a urmărit integrarea de cât mai multe subsisteme utile (memorie EPROM, RAM, convertoare AD și DA). Dacă prețul unui procesor nu a scăzut sub câteva zeci de dolari (PII/450MHz), prețul unui microcontroller poate ajunge la câțiva dolari, ceea ce încurajează eforturile proiectanților de a realiza module I/O inteligente cu microcontroller, sau alte aplicații cu microcontroller.

Conexiunea între un modul I/O și unitatea centrală poate fi:

1. Punct-la-punct; linii dedicate pentru transferul de date (de exemplu la un PC AT tastatura, RS232 etc.);
2. Multipunct; la liniile de interfață se pot lega mai multe EP (module I/O; ca la SCSI)

Dacă o legătură punct cu punct se numește indiscutabil interfață, o legătură multipunct se poate numi atât interfață cât și magistrală. Interfața serială multipunct USB (Universal Serial Bus) conține în numele ei denumirea de magistrală.

3.3. MAGISTRALĂ PC

3.3.1. Magistrala ISA (*Industry Standard Architecture*)

3.3.1.1. *Prezentare generală*

ISA este un standard de magistrală care s-a impus prin produs; a fost magistrala primelor calculatoare PC construite de IBM. Inițial magistrala de date era de 8 biți iar cea de adrese de 20 de biți. Ulterior, când a apărut calculatorul AT (*Advanced Technology*), magistrala de date s-a extins la 16 biți iar cea de adrese la 24 de biți. Sistemul AT a păstrat compatibilitatea cu prima variantă; când se face referire la magistrala ISA se subînțelege ISA-16. Destul de târziu, în anul 1990, IEEE a prevăzut reglementări pentru magistrală prin standardul IEEE P996.

În sistem sunt prevăzute 8 canale DMA și 16 nivele de întreruperi. Parte din canalele DMA și din întreruperile disponibile sunt alocate prin standard unor echipamente ce fac parte din

configurația calculatorului PC, celelalte sunt libere, la dispoziția utilizatorului pentru aplicații particulare. Magistrala poate fi controlată de procesor sau de controllerul DMA.

Magistrala este concepută pentru tranzacții I/O unice cu periferia. Tranzacțiile cu memoria se fac prin magistrală. Nu este prevăzut modul de exploatare multiutilizator real.

Această magistrală este întâlnită în toate calculatoarele compatibile PC IBM-PC/AT și în calculatoarele construite cu microprocesor 386SX, 386DX, 486SX, 486DX.

În Fig. 3.4. este prezentată structura unei plăci de bază cu magistrală ISA.

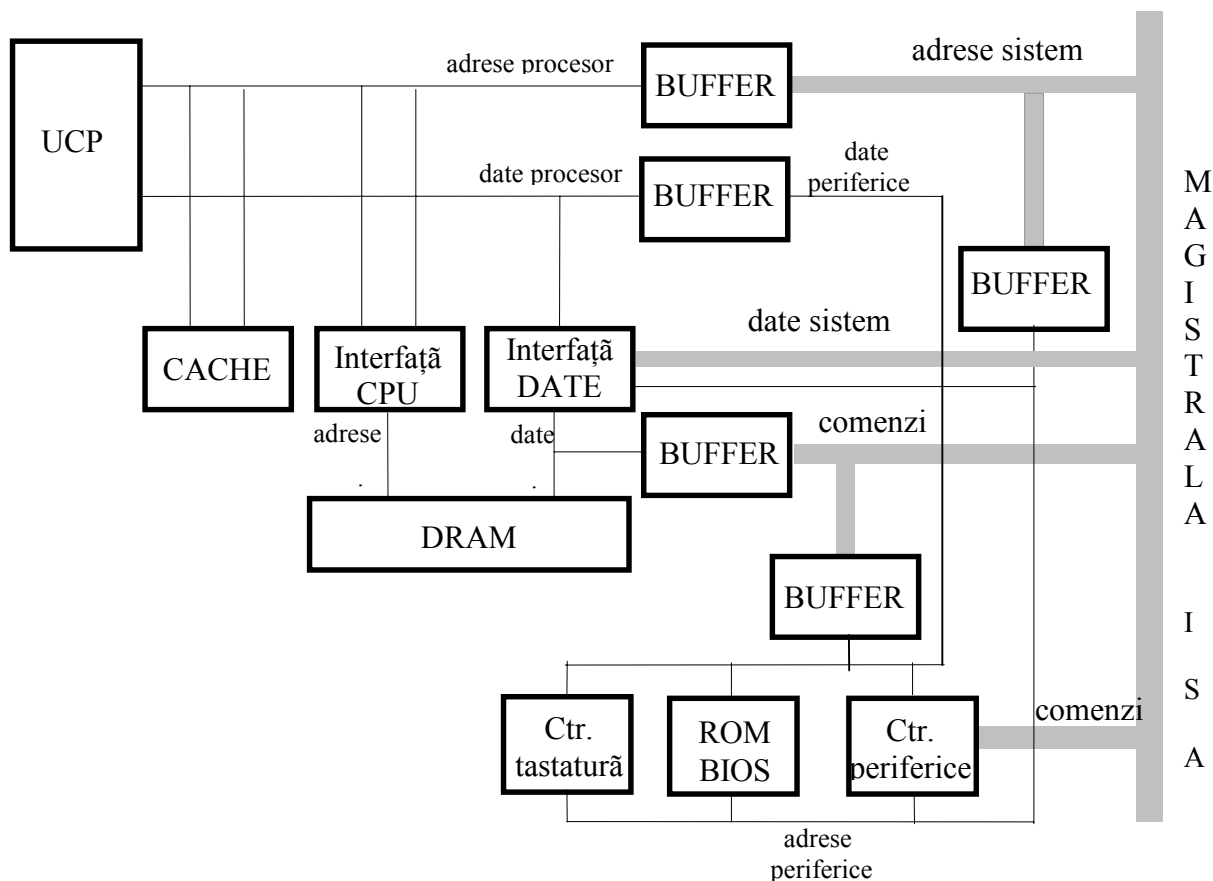


Fig. 3.4. Sistem cu magistrală ISA

3.3.1.2. Semnalele magistralei ISA

RESET DRV - *Reset Driver* - ieșire

Semnalul apare la pornirea generală sau la acționarea Resetului logic.

BCLK - ieșire

Inițial acest semnal a fost tactul tuturor operațiilor de pe cartelele de extensie. Este un semnal între 6 MHz și 12.5 MHz.

OSC - ieșire

Semnal cu frecvența 14.31818 MHz disponibil cartelei CGA. Poate fi ignorat în dezvoltarea altor cartele.

IRQ3 - IRQ15 - *Interrupt Request* - intrări

Linii conectate la intrările controlerului de întreruperi. Transportă cererile de întrerupere venite de la periferice.

REFRESH# - intrare/ieșire

Marchează un ciclu refresh activ. Ciclul refresh se poate prelungi cu semnalul IOCHRDY, nu se poate scurta. Semnalul este produs prin canalul 1 al timerului la fiecare 15.6 μs.

DRQ0-DRQ7 - *DMA request* - intrări

Cerere de inițiere a unui ciclu DMA trimisă prin magistrală controlerului DMA.

DACK0-DACK7 - *DMA Acknowledge* - ieșiri

Onorarea unei cereri de transfer DMA. Semnalul este generat de controlerul DMA.

AEN - *Address Enable* - ieșire

Semnal generat de controlerul DMA, activ (HIGH) pe toată durata unui ciclu DMA. Servește la diferențierea acceselor DMA de accesesele I/O. Din acest motiv, toate modulele de extensie trebuie să folosească validarea adreselor cu AEN pentru ca să nu apară decodificare falsă în timpul unui ciclu DMA.

TC - *Terminal Count* - ieșire

Indică sfârșitul unui transfer DMA.

SMEMW# - *Small Address Memory Write* - ieșire

Indică accesul procesorului sau al controlerului DMA pentru scriere în memorie.

SMEMR# - *Small Address Memory Read* - ieșire

Indică accesul procesorului sau al controlerului DMA pentru citire din memorie.

MEMW# - *Memory Write* - intrare/ieșire

Indică accesul procesorului sau al controlerului DMA pentru scriere în memorie. Condiționează apariția semnalului SMEMW#.

MEMR# - *Memory Read* - intrare/ieșire

Indică accesul procesorului sau al controlerului DMA pentru citire din memorie. Condiționează apariția semnalului SMEMR#.

IOW# - *I/O Write* - intrare/ieșire

Indică accesul procesorului sau al controlerului DMA pentru scriere la periferic.

IOR# - *I/O Read* - intrare/ieșire

Indică accesul procesorului sau al controlerului DMA pentru citire de la periferic.

BALE - *Bus Address Latch Enable* - ieșire

Validează cu frontul său negativ semnalele SA0-SA19, BHE, AEN.

SD0-SD15 - *System Data* - intrare/ieșire

Reprezintă liniile magistralei de date.

SA1-SA19 - *Small Address* - intrare/ieșire

Semnalele magistralei de adrese compatibile cu prima variantă ISA.

BHE# - *Bus High Enable* - intrare/ieșire

Participă la generarea selecției de Byte pentru a se ști cu care jumătate de byte se lucrează.

SA0 - *System Address bit 0* - intrare/ieșire

Împreună cu semnalul BHE realizează semnalul de selecție Byte.

SA0	BHE	
0	0	SD0-SD15
1	0	SD0-SD15
0	1	SD0-SD7
1	1	nepermis

LA17-LA23 - *Large Address* - intrare/ieșire

Reprezintă adresele superioare. Împreună cu Small Address pot adresa spațiul 000000h-FFFFFFh.

IOCHRDY - *I/O Channel Ready* - intrare

Servește la prelungirea ciclurilor I/O la transferuri cu periferice lente. Ciclul se poate prelungi până la 15.6 μ s.

IOCHCK# - *I/O Channel Check* - intrare

Dă posibilitatea tuturor modulelor de extensie să fie verificate la erori interne de către unitatea centrală gazdă.

MEMCS16# - *Memory Chip Select 16 bit* - intrare/ieșire

Modulele de extensie de memorie pot activa acest semnal pentru a indica desfășurarea unui transfer pe 16 biți.

IOCS16# - *I/O Chip Select 16 bit* - intrare

Modulele I/O activează acest semnal atunci când fac transferuri pe 16 biți.

OWS# - *Zero Wait State* - intrare

Un modul I/O sau un modul de extensie de memorie poate scurta durata unui acces prin înlăturarea stărilor Wait activând acest semnal.

MASTER# - *Master Request* - intrare

Semnal cu care controlul magistralei poate fi preluat de un procesor extern.

3.3.1.3. Interfața I/O

O interfață I/O pentru magistrala ISA are o structură destul de simplă. Spațiul de adresare este 000h-3FFh. În acest spațiu există poziții rezervate echipamentelor din configurația standard a unui calculator PC. La adresele lăsate libere de standardului PC utilizatorul poate configura sistemul după necesitățile particulare.

Ciclurile I/O se desfășoară conform diagramelor de semnal prezentate în Fig. 3.5.

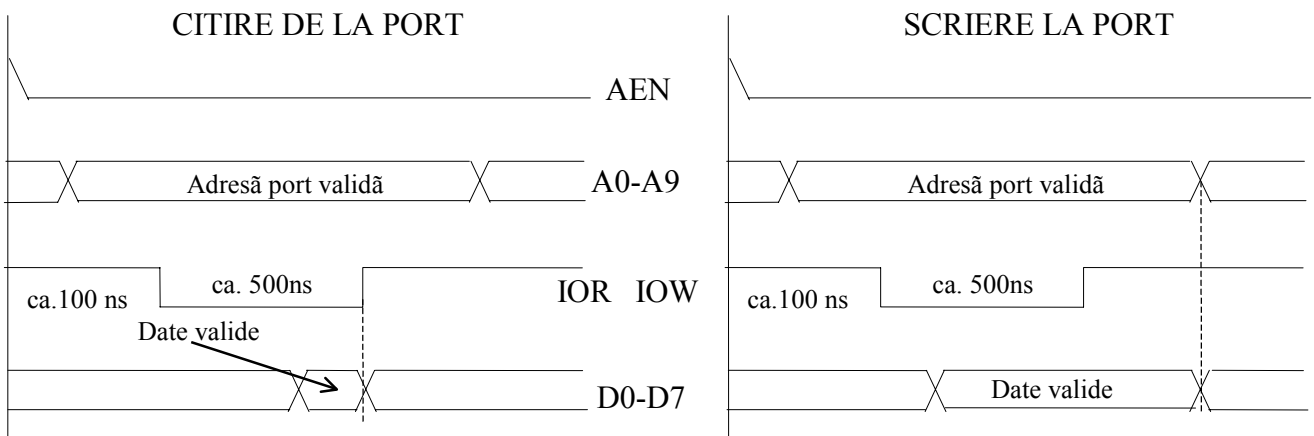


Fig. 3.5. Cicluri I/O

3.3.2. Magistrala PCI (Peripheral Component Interconnect)

3.3.2.1. Structura magistralei PCI

Magistrala PCI a fost realizată în 1990 de INTEL pentru procesoarele PENTIUM. INTEL a cedat specificațiile domeniului public, ceea ce explică răspândirea acestei magistrale. PCI este o magistrală sincronă care poate lucra cu 32 sau 64 de biți de date, la o viteză de maximum 2,112 GBps. Structura unui sistem cu magistralei PCI este dată în Fig. 3.6.

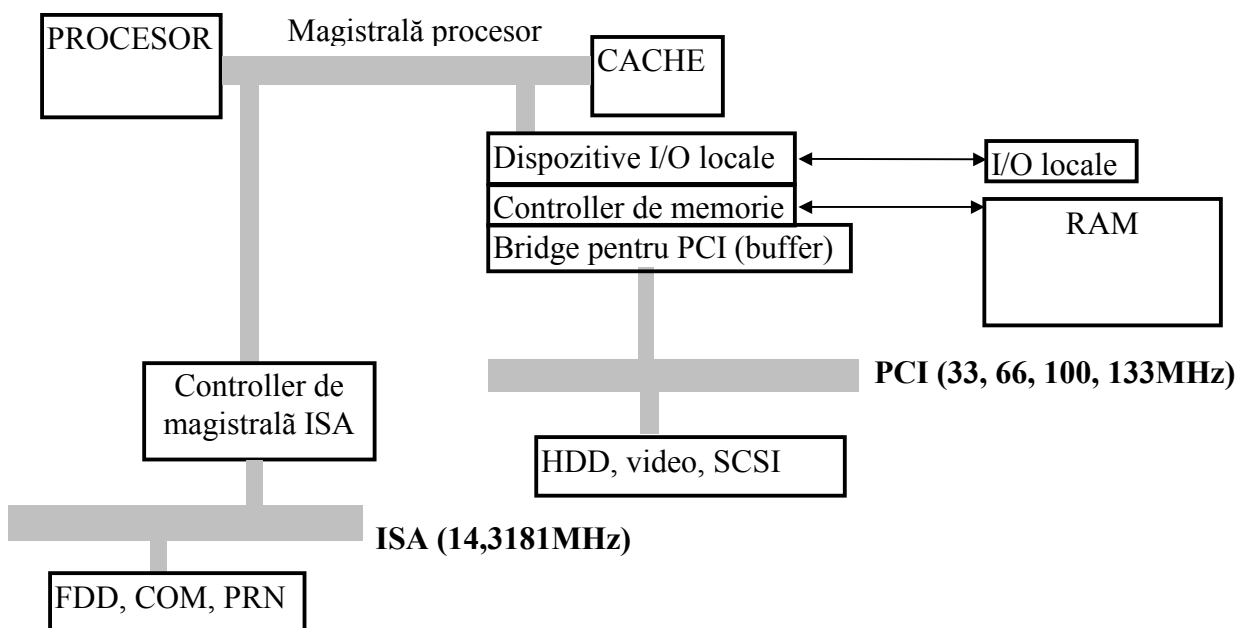


Fig. 3.6. Structura unui sistem cu magistrală PCI

Magistrala PCI are o arhitectură de tip "mezanin", din cauza apropierii de procesor. În Fig. 3.6. au fost notate și frecvențele de tact la care funcționează cele două magistrale, pentru a se putea observa creșterea de viteză la PCI. Magistrala ISA a fost păstrată în configurație din considerente de compatibilitate cu echipamentele existente încă în utilizare.

3.3.2.2. Semnalele magistralei PCI

Semnalele magistralei PCI sunt:

Linii generale

CLK - tactul de magistrală PCI, 33MHz-133MHz;

RST - semnal de RESET, forțează toate modulele de I/O și semnalele la o stare inițială;

AD 0-31 - 32 linii multiplexate pentru adrese și date;

C/BE 0-3 - linii multiplexate pentru comenzi de bus sau Byte Enable. În timpul fazei de date arată care din cei 4 octeți de date poartă date semnificative;

PAR - conține bitul de paritate pentru liniile AD și C/BE la un un tact de magistrală după ce datele sau adresele au fost puse pe magistrală. Dispozitivul MASTER generează PAR pentru fazele de adresare și scriere date și dispozitivul țintă pentru faza de citire date;

FRAME - generat de MASTER-ul curent (inițiator) ca să indice startul unui transfer. Este dezactivat când inițiatorul începe faza finală de date;

IRDY - inițiator pregătit, generat de MASTER-ul curent. La citire indică faptul că MASTER-ul este gata să accepte date, iar la scriere, pe AD sunt prezente date valide;

TRDY - țintă pregătită, generat de modulul selectat (țintă). La citire, pe AD sunt prezente date valide, la scriere indică faptul că ținta este gata să accepte date;

STOP - ținta curentă dorește ca inițiatorul să termine transferul curent;

LOCK - indică o operație care poate consta în mai multe transferuri;

IDSEL - se folosește ca Chip Select pe durata inițializărilor pentru citire/scriere;

DEVSEL - activat de țintă când își recunoaște adresa;

REQ - indică arbitrilor că un dispozitiv solicită utilizarea magistralei. Este o linie punct cu punct specifică fiecărui slot PCI;

GNT - indică dispozitivului care a cerut magistrala că a primit accesul. Este o linie punct cu punct specifică fiecărui slot PCI;

PERR - eroare de paritate detectată de țintă la o scriere de date sau de inițiator la o citire de date;

SERR - eroare de sistem, poate fi generat de orice dispozitiv pentru a raporta o eroare de paritate la adresare sau un alt tip de eroare;

INTA - cerere de întrerupere;

INTB, INTC, INTD - cereri de întrerupere de la un dispozitiv multifuncțional;

SBO - indică o modificare în CACHE;

SDONE - indică terminarea unui transfer cu CACHE-ul.

Linii pentru extensia la 64 de biți:

AD 32-63 - 32 de linii de date și adrese multiplexate;

C/BE 4-7 - la adresare liniile conțin date despre tipul de ciclu de bus. În faza de date indică care din cei 4 octeți de date poartă date semnificative;

REQ64 - indică o cerere de transfer pe 64 de biți;

ACK64 - indică faptul că ținta va face un transfer pe 64 de biți;

PAR64 - bitul de paritate pe liniile AD 32-64 și C/BE un tact mai târziu.

Linii pentru testarea JTAG conform standardului IEEE 149.1:

TCK - *Test Clock*, ceas pentru testare cu date seriale de intrare sau de ieșire în/din dispozitiv;

TDI - *Test Input*, date de intrare seriale în dispozitiv;

TDO - *Test Output*, date seriale de ieșire din dispozitiv;

TMS - *Test Mode Select*, folosit pentru controlul testării;

TRST - *Test Reset*, se inițializează portul de test.

3.3.2.3. Cicluri de magistrală

Activitatea magistralei se manifestă sub forma unor transferuri între un inițiator sau MASTER și o țintă. După ce un MASTER a obținut controlul magistralei, determină tipul transferului care va urma. În timpul fazei de adresare liniile C/BE informează despre tipul ciclului de magistrală.

Ciclurile de magistrală sunt:

1. Acceptare întrerupere; este un ciclu de citire conceput pentru dispozitivele care funcționează ca și controler de întreruperi pe PCI. Liniile de adresă nu sunt folosite în faza de adresare, iar în faza de date este indicat vectorul de întrerupere.

2.Ciclu special; este folosit de inițiator pentru a emite mesaje către una sau mai multe ținte.

3.Citire I/O; este un ciclu de transfer de date între inițiator și un dispozitiv I/O. Fiecare dispozitiv I/O are alocat un spațiu propriu de adresare.

4.Scriere I/O

5.Citire din memorie; citirile din memorie se pot executa în salvă, pe blocuri, funcție de modul în care controlerul de memorie gestionează transferurile între memorie și CACHE. Se citește o jumătate de linie de CACHE sau mai puțin, în salvă sunt unul sau două cicluri de citire.

6.Citire o linie din memorie; se citește mai mult de o jumătate de linie de CACHE dar mai puțin de trei linii, în salvă sunt între 3 și 12 cicluri de citire.

7.Citire multiplă din memorie; se citesc mai mult de 3 linii de CACHE, în salvă sunt mai mult de 12 cicluri de citire.

8.Scriere în memorie

9.Scriere în memorie și invalidare; scrierea se poate face de asemenea în salvă. Se respectă modul WRITE BACK pentru scrierea din CACHE în memorie;

10.Citire configurație; fiecare dispozitiv PCI include un set de 256 de registre interne care se folosesc în faza de configurare. În acest tip de ciclu, dispozitivul MASTER poate citi registrele de configurare.

11.Scriere configurație; în acest tip de ciclu, dispozitivul MASTER poate modifica registrele de configurare.

12.Ciclu de adresare dual; este folosit de inițiator pentru a arăta că este o adresare pe 64 de biți.

Orice transfer de date pe PCI constă într-o fază de adresare și una sau mai multe faze de date. În Fig. 3.7. este arătat un transfer de citire.

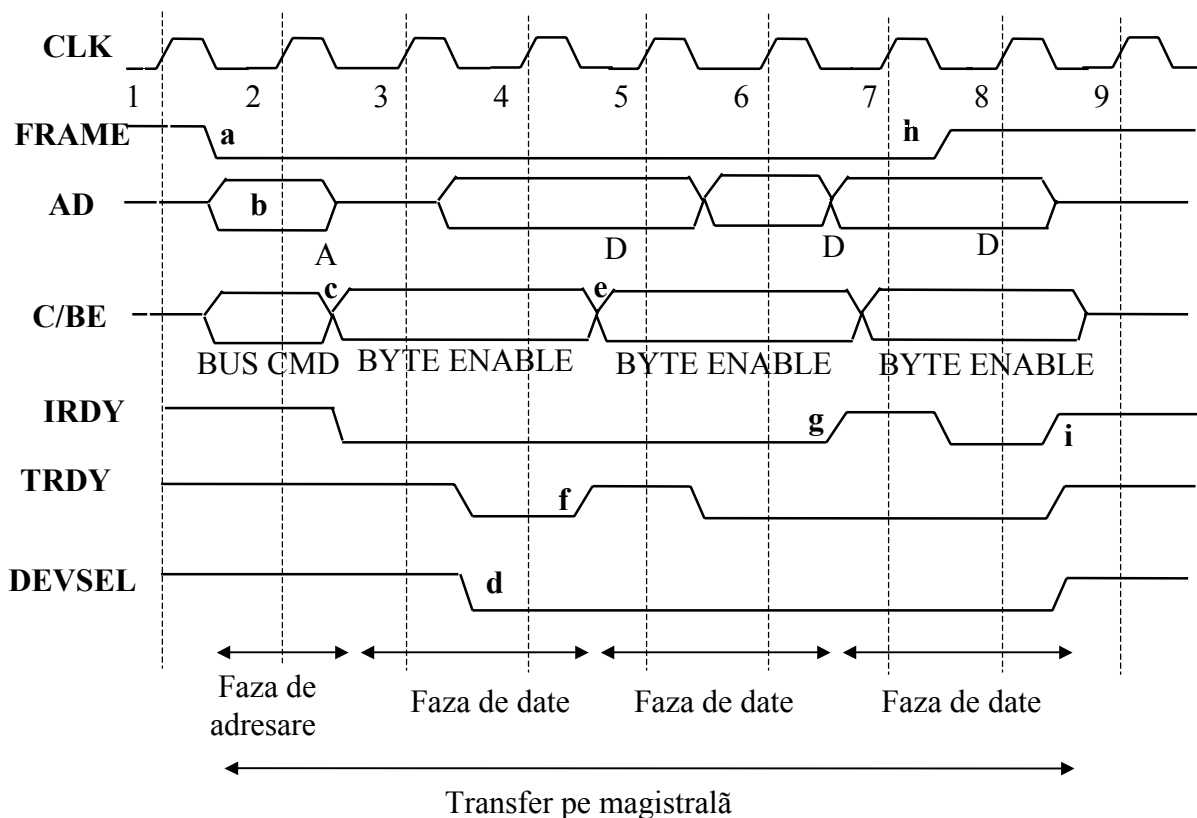


Fig. 3.7. Ciclu de citire PCI

a De îndată ce MASTER-ul are controlul magistralei el poate începe transferul activând FRAME. Linia rămâne activă până inițiatorul termină ultima fază de date. Inițiatorul pune adresa pe AD și comanda de citire pe liniile C/BE.

b Ținta își recunoaște adresa pe liniile AD.

c Inițiatorul cedează bus-ul țintei (începând cu tactul 3). Inițiatorul schimbă informația pe liniile C/BE pentru a indica pe care dintre liniile AD se vor transmite datele de la țintă (1 până la 4 octeți)

d Ținta selectată activează DEVSEL pentru a semnaliza că și-a recunoscut adresa, apoi pune datele pe liniile AD (primul octet) și activează TRDY.

e În acest exemplu ținta are nevoie de timp pentru a pregăti următorul octet de date.

f Ținta dezactivează TRDY pentru a informa inițiatorul că pe următorul tact nu vor exista date valide. Ca urmare, inițiatorul nu va citi datele la începutul tactului 5 ci la începutul tactului 6. S-a inserat astfel o stare de WAIT.

g După tactul 6 ținta pune pe bus al treilea octet de date. Presupunem de data aceasta că inițiatorul nu este gata să preia datele (are de exemplu bufferul plin). Inițiatorul dezactivează IRDY, ceea ce are ca efect menținerea datelor de către țintă un tact suplimentar (datele nu se citesc la începutul tactului 7 ci la începutul tactului 8).

h inițiatorul dezactivează semnalul FRAME și activează semnalul IRDY, pentru a semnaliza că este gata să finalizeze transferul.

I Inițiatorul dezactivează IRDY, permițând busului să treacă în starea inițială. Ținta a detectat dezactivarea semnalului FRAME și dezactivează TRDY și DEVSEL.

3.3.2.4. Arbitrarea magistralei

Pentru a arbitra gestionarea magistralei, PCI folosește un arbitru central într-o configurație asincronă, cu cereri (REQ) și confirmări (GNT) dedicate, ca în Fig. 3.8.

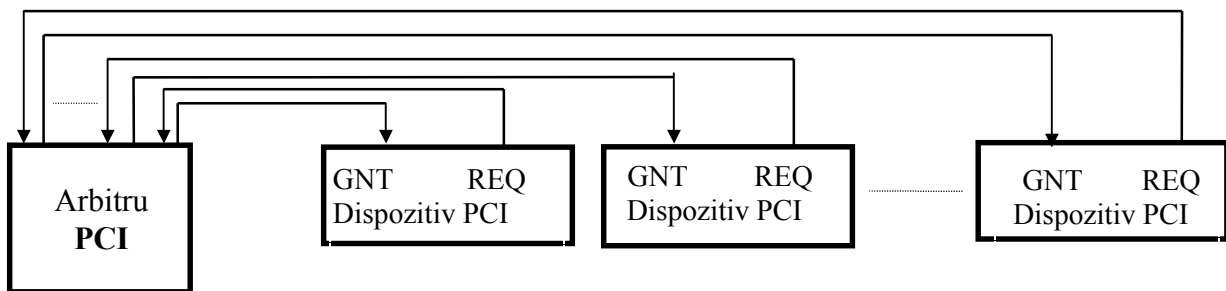


Fig. 3.8. Arbitrarea magistralei PCI

Observație: din cauza liniilor dedicate pentru arbitrarea accesului la magistrală, poziția plăcilor în sloturile PCI nu este indiferentă, așa cum era la magistrala ISA. În funcție de poziția slotului, plăcii i se atașează un nivel de prioritate.

În Fig. 3.9. se exemplifică mecanismul de arbitrare a două dispozitive A și B de către un arbitru.

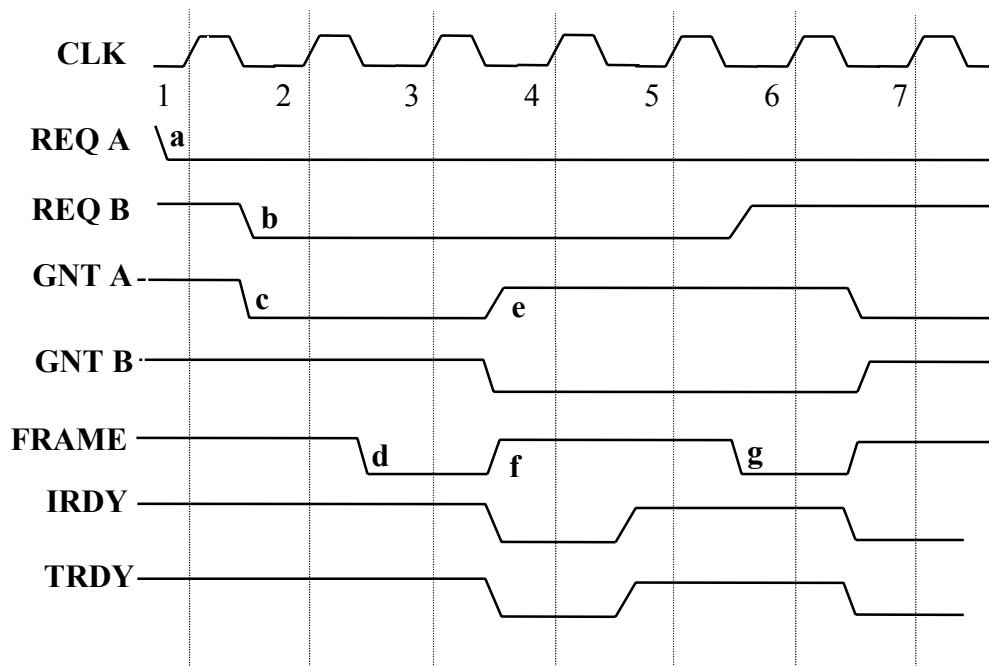


Fig. 3.9. Mecanism de arbitrare PCI

a Înainte de tactul 1 A a activat cererea de bus REQ A.

b B solicită busul cu REQ B.

c În același timp arbitrul cedează busul dispozitivului A (GNT A activ).

d MASTER-ul A, odată ce îi este indicat prin GNT A că are acces la bus, verifică IRDY și TRDY. Acestea fiind inactive, deci nu există un transfer pe bus, activează semnalul FRAME și începe un transfer prin stabilirea adresei etc.

e În tactul 3 arbitrul de bus găsește 2 cereri active, de la A și de la B. și decide cedarea busului dispozitivului B pentru următorul transfer. Activează GNT B și dezactivează GNT A. B nu poate folosi busul până acesta nu trece în stare inactivă, IRDY și TRDY inactive.

f A dezactivează FRAME indicând că transferul este ultimul transfer. Pune datele pe bus și semnalizează cu IRDY. Ținta citește datele.

g La începutul tactului 5, B găsește IRDY, FRAME și TRDY inactive și preia controlul busului generând semnalul FRAME. B dezactivează linia REQ B pentru că avea nevoie de un singur transfer. Controlul busului este din nou cedat dispozitivului A, care și-a menținut cererea REQ A.

Ca urmare a necesității arbitrării magistralei și a stabilirii pe liniile C/BE a unor coduri în diferitele faze ale transferului, o placă ce se conectează pe PCI nu poate fi atât de simplă ca și o placă ISA.

O schemă bloc a unui circuit de interfață PCI echipat cu controllerul de magistrală PCI, PCI9060 (furnizor PLX Technology, <http://www.plx-tech.com>) este dată în Fig. 3.10. O schemă completă este dată în bibliografie.

PCI9060 are rolul de a rezolva problema accesului la magistrală cu REQ și GNT, de a asigura semnalele de dialog la transfer C/BE, FRAME, IRDY, TRDY etc., poate lucra în mod DMA cu DRQ și DACK, de a demultiplexa cele 32 de linii de adrese și date AD 0-31. Datele de configurare sunt înscrise în controller de un EEPROM serial în faza de RESET a calculatorului. PCI 9060 furnizează tactul de citire EECLK pentru linia de date citite EEDI.

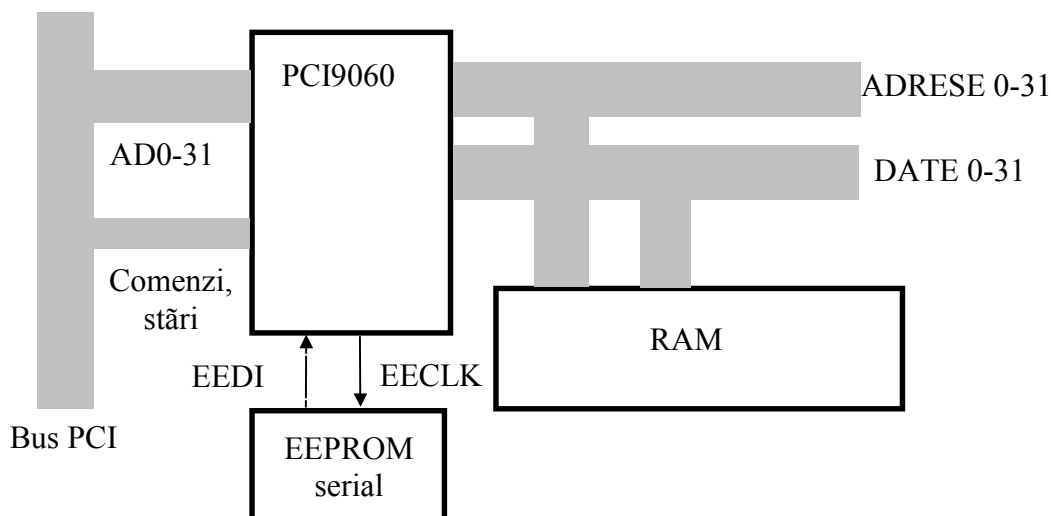


Fig. 3.10. Schema bloc a unui circuit de interfață PCI

PCI9060 poate gestiona o memorie proprie, cu rolul de memorie tampon între EP care se conectează pe magistrala PCI și UC.

3.3.3. Magistrala VME

VME este o magistrală asincronă de 32 (64) biți de date care permite lucrul multiprocesor. Adresarea poate fi de asemenea pe 32 de biți. Pe această magistrală se pot conecta mai multe subsisteme cu procesoare și memorii proprii (cu lungime diferită a cuvântului de date). VME s-a răspândit mai puțin la noi dar în societățile dezvoltate industrial există multe calculatoare cu VME în domeniul măsurărilor și automatizărilor sau în domeniul aviației. VME a fost realizată de MOTOROLA în 1981 și a fost standardizată ca IEC 821 BUS și IEEE 1014-1987.

Dintre caracteristicile generale ale acestei magistrale se pot aminti:

- transferul datelor pe 8, 16, 24, 32, sau 32 biți;
- magistrala de adrese de 16, 24, 32, 40 sau 64 biți;
- arhitectură master/slave, multiprocesor cu 1-21 procesoare;
- mecanism de transfer asincron (multiplexat și nemultiplexat);
- 7 nivele de întrerupere;
- rata de transfer de la 40 MBps la 500 MBps, în funcție de vesiunea magistralei;
- 21 sloturi.

3.3.3.1. Structura magistralei VME

Magistrala VME lucrează cu 4 grupări de semnale:

- Data Transfer Bus (magistrala de date) DTB - facilitează schimbul de date dintre stăpânul și sclavii sistemului;
- Arbitration Bus (magistrala de arbitraj) - realizează împărțirea magistralei între mai mulți stăpâni;
- Interrupt Bus (magistrala de întrerupere);

- Utility Bus - (magistrală utilitară) distribuie elementelor din sistem alimentarea, semnale de tact, resetarea, indicarea scăderii unei tensiuni de alimentare sub valoarea minimă, indicarea erorilor de subsistem.

Fig. 3.11. prezintă structura unui sistem cu magistrală VME.

1. Nivelul de acces la magistrala de fundal 2. Nivelul de transfer de date.

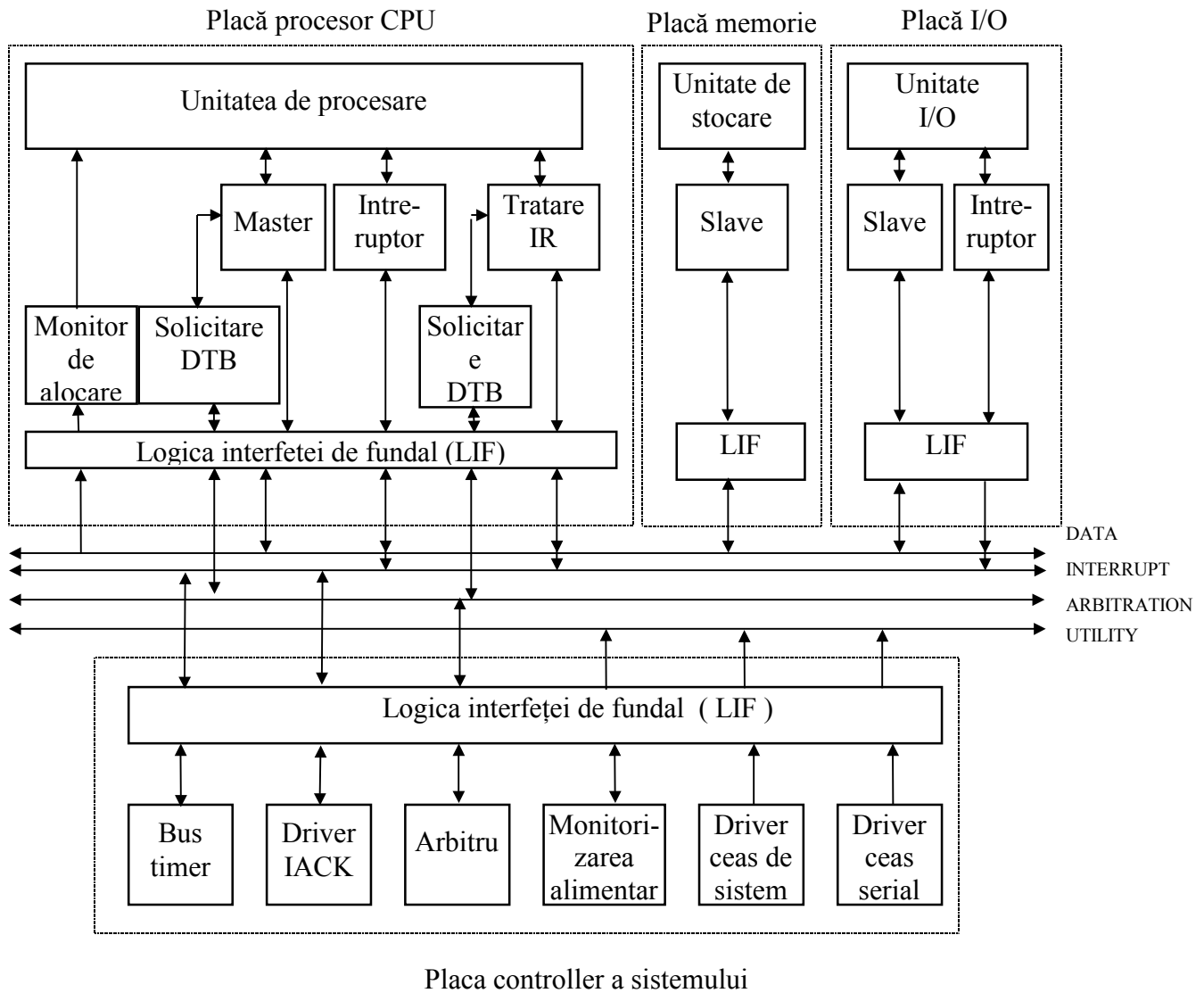


Fig. 3.11. Structura unui sistem cu magistrală VME .

3.3.3.2. Magistrala de date

Semnalele de transfer de date realizează interfața între un MASTER și un SLAVE. Controlerul de sistem, care este o placă situată în slotul 1 al unui sistem VME, comandă transferurile și arbitrează subsistemele MASTER pentru controlul magistralei. La un moment dat, pe magistrală poate avea loc o singură tranzacție MASTER-SLAVE.

Magistrala admite conectarea procesoarelor pe 8, 16, 32 și 64 de biți (revizia D). Linia de adresă A0 formează strob-urile de date DS0 și DS1 care indică un transfer pe 8 sau pe 16 biți. Pentru transferul pe 32 de biți se activează semnalul LWORD (Long Word).

Când adresa este pusă pe bus se activează AS (Adress Strobe) (Fig. 3.12.). Dispozitivul SLAVE răspunde cu DTACK că a preluat datele. Dacă apare o eroare, ciclul este abandonat prin activarea semnalului BERR. Dacă BERR sau DTACK nu sunt activate, sistemul fiind asinron se blochează. Din acest motiv se adaugă un ceas de gardă care activează BERR după un anumit timp.

După adresă se trimite un cod pe liniile AM0-AM4 care specifică tipul tranzacției, câți biți se folosesc din adresă etc. Semnalul WRITE specifică dacă este un ciclu de scriere sau de citire.

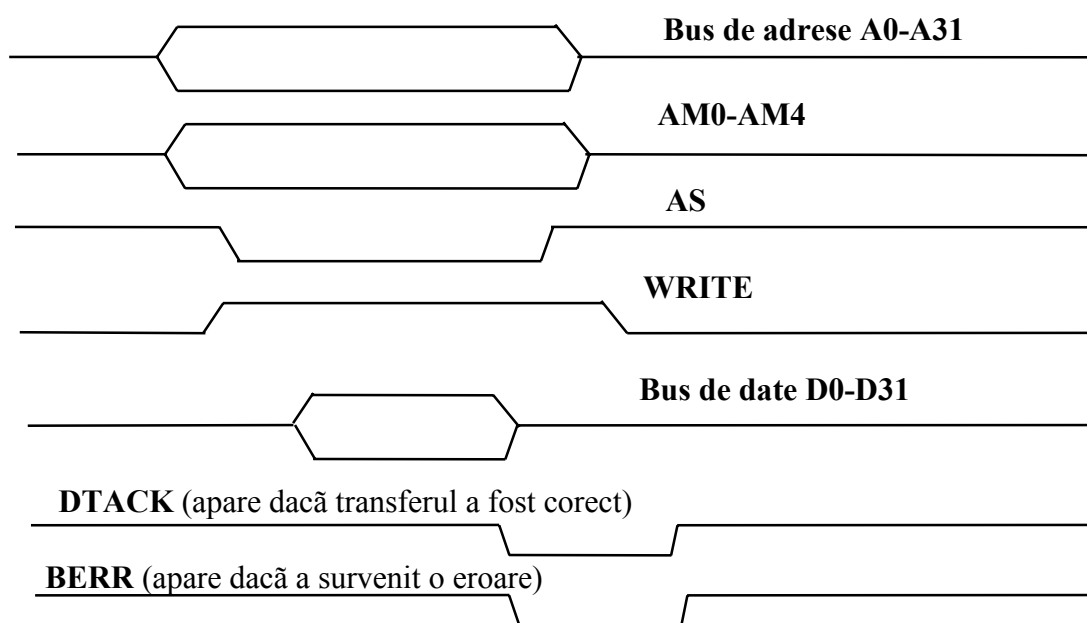


Fig. 3.12. Transfer de date pe magistrala VME

Există posibilitatea unui ciclu de citire/modificare/scriere (ciclu indivizibil). Semnalul AS rămâne valid în timp ce WRITE trece în 0 pentru a permite o citire., iar un alt set de date este poziționat pe bus. Ciclul se termină prin apariția DTACK și BERR.

Stabilind un anumit cod pe liniile AM0-AM4 se poate face transfer pe blocuri, de maximum 256 de octeți. Primul octet se transferă în mod obișnuit, dar apoi magistrala de adrese este lăsată la dispoziția dispozitivului SLAVE care o incrementează (cu numărul de octeți transferați într-un ciclu).

VME permite un pipeline al adreselor, adică adresa următoare poate fi plasată în timp ce pe liniile de date mai sunt datele precedentului transfer.

Există posibilitatea realizării unui ciclu de adresare, la care nu se face nici un transfer de date și nu se verifică DTACK sau BERR folosit pentru a iniția diferite comenzi într-un subsistem VME.

3.3.3.3. Magistrala de arbitraj

Semnalele de arbitrare a bus-ului de date sunt necesare pentru că bus-ul fiind multiprocesor, trebuie să existe garanția că numai un MASTER accesează bus-ul la un anumit moment. Un dispozitiv MASTER solicită magistrala cu unul din semnalele BR0-BR3 și așteaptă ca arbitrul de bus situat în slotul 1 să răspundă. Dacă bus-ul este vacant, arbitrul răspunde cu semnalul BGIN_i ($i=0-3$) corespunzător. Semnalele BGIN_i și BGOUT_i sunt cuplate în topologie daisy chain. Aceasta înseamnă că ierarhia este bazată pe poziția în slot. MASTER-ul poziționează linia BBSY care indică faptul că bus-ul este ocupat și dezactivează cererea de bus. Arbitrul dezactivează permisiunea de preluare a bus-ului. După ce MASTER-ul termină transferul dezactivează linia BBSY, magistrala devenind liberă. Este important ca lanțul de priorități să fie neîntrerupt, de aceea se folosesc *jumperi* pentru a lega BGIN cu BGOUT acolo unde nu sunt plăci.

3.3.3.4. Magistrala de întreruperi

Gestionarea întreruperilor se poate realiza pentru 7 nivele de întrerupere. Se solicită o întrerupere pe liniile IRQ1-IRQ7 (IRQ7 are cea mai mare prioritate). Se face o citire a liniei care a cerut întreruperea (biții 1-3 de adresă). Ca și la arbitrarea magistralei se face un daisy chain prin semnalele IACKIN și IACKOUT (Fig. 3.13.).

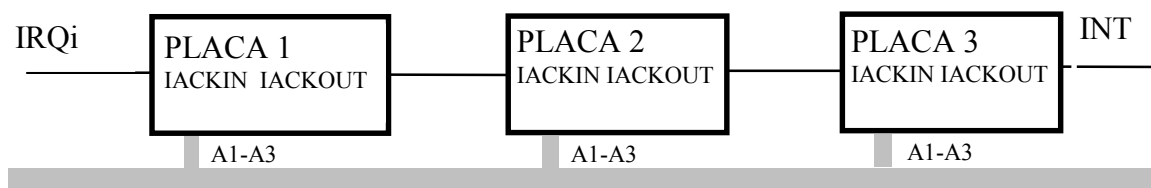


Fig. 3.13. Gestionarea întreruperilor în magistrala VME

Placa care a cerut întreruperea recunoaște codul de 3 biți din adresă și blochează prin IACKOUT calea de întrerupere. MASTER-ul execută transferul și termină prin activarea semnalului DTACK care dezactivează IACKOUT.

3.3.3.5. Magistrala utilitară

Semnalele magistralei utilitare sunt:

- alimentare +5V, +12V, -12V;
- ceasul de magistrală SYSCLK;
- SYSFAIL indică o eroare de sistem;
- ACFAIL indică o cădere a sursei de alimentare;
- SYSRESET semnal de RESET pentru toate plăcile.

3.3.4. SCSI (Small Computer System Interface)

SCSI este o interfață paralelă multipunct cu 8, 16 sau 32 linii de date. Ea poate fi privită ca un bus, dar practic dispozitivele SCSI sunt legate prin daisy chain, fiecare având un conector de intrare și unul de ieșire. Din punctul de vedere al sincronizării, SCSI este un bus asincron. Orice dispozitiv poate transfera date cu orice alt dispozitiv sau cu dispozitivul MASTER. Standardul actual este SCSI 2 care permite transfer pe maximum 32 de linii de date la o viteză de transfer de maximum 20-40MBps. Orice transfer SCSI are un inițiator și o țintă. De obicei inițiatorul este dispozitivul MASTER. Fazele unui transfer sunt:

- Bus liber - indică faptul că bus-ul nu este ocupat și poate fi folosit;
- Arbitrare - validează dispozitivul SCSI să câștige controlul bus-ului, astfel încât să poată iniția un transfer;
- Selecție - validează dispozitivul inițiator pentru a selecta o țintă, spre care să facă un transfer I/O de READ sau WRITE;
- Reselecție - permite ca ținta să se reconecteze la inițiator pentru a continua o operație suspendată de țintă;
- Comenzi - validează dispozitivul inițiator să trimită comenzi țintei;
- Date - validează transferul spre inițiator (DATA IN) sau spre țintă (DATA OUT);
- Stări - validează comunicarea stării țintei către inițiator;
- Mesaje - validează transferul de mesaje spre inițiator (MESSAGE IN) sau spre țintă (MESSAGE OUT).

Semnalele interfeței SCSI sunt:

BSY - setat de orice dispozitiv pentru a indica că bus-ul este ocupat;

SEL - folosit de inițiator pentru a selecta o țintă, sau de țintă pentru a reselecta inițiatorul;

C/D - indică dacă este un transfer de comenzi, stări, mesaje sau de date;

MSG - indică un transfer de mesaje;

REQ - cerere de transfer de la țintă la inițiator;

ACK - inițiatorul răspunde cu ACK la cererea de transfer REQ a țintei și confirmă faptul că inițiatorul a pus informația pe liniile de date sau a citit-o de pe liniile de date;

ATN - folosit de inițiator pentru a informa că există un mesaj pentru a fi transferat;

RST - RESET;

D0-D31 - liniile de date;

P - linie (linii) de paritate.

Diagrama de timp pentru un transfer SCSI - arbitrare, selecție, transfer de comenzi, de date, de stări și de mesaje este dată în Fig. 3.14.

Fazele distincte 1, 2, 3, 4, 5 sunt explicate în continuare:

1. Inițiatorul pune BSY=1 și selectează ținta cu SEL=1. Pe linia de date se transmite codul operației de arbitrare apoi adresa țintei.

2. Linia C/D=1 indică un transfer de comenzi. În exemplul din figură se transmit 2 cuvinte pe liniile de date. Sincronizarea transferului se face cu semnalele REQ și ACK. Linia BSY este pusă în 1 de către țintă.

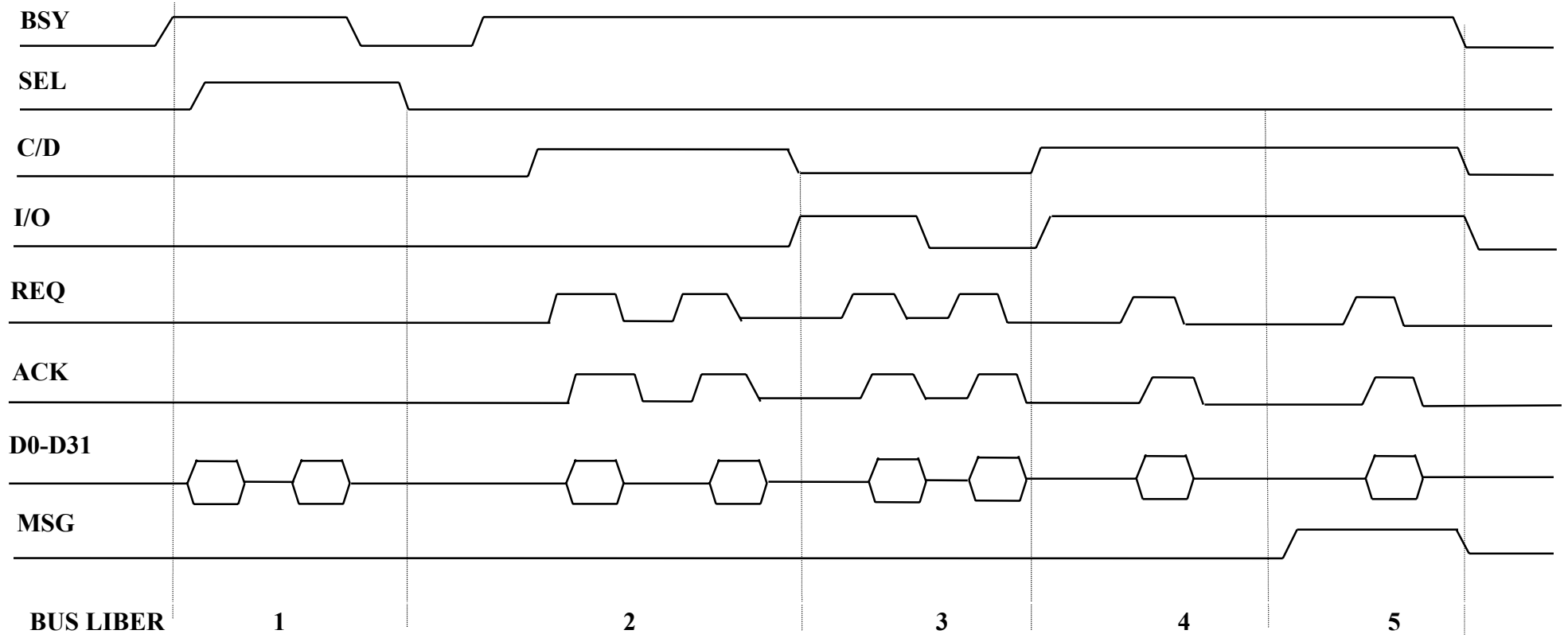


Fig. 3.14. Diagrama de timp a unui transfer SCSI

3. Transferul efectiv de date se face sincronizat cu REQ și ACK, linia C/D=0 indicând un transfer de date. Primul cuvânt este citit (linia I/O=1) iar al doilea este scris (linia C/D=1).
4. Se comunică starea (linia C/D=1) privind rezultatele transferului.
5. Se transferă un mesaj indicând linia MSG=1.

3.4. ALTE MAGISTRALE

FUTUREBUS+

Este un bus asincron de mare performanță. Suportă cuvinte de date de 32, 64, 128 și 256 biți. FUTUREBUS+ suportă două modele de arbitrare a bus-ului, unul centralizat și unul distribuit. Un grup de 3 linii permite fiecărui modul să anunțe modul de transfer suportat. Realizatorii bus-ului susțin că este independent de arhitectura sistemului de calcul, de tipul procesorului și de tehnologia de realizare a sistemului. La bus se pot conecta plăci cu memorie cache proprie, utilizată în comun. Oferă de asemenea și suport pentru toleranța la erori.

CEBus

Se bazează pe transmisia de date prin rețeaua industrială de curent alternativ (120V, 60Hz), (Standard Powerline Carrier Technology) pentru conectarea echipamentelor de electronică casnică între ele. În această tehnologie se folosește împrăștierea spectrului pentru a evita pericolul perturbărilor produse de rețea. Pe durata fiecărui bit de informație spectrul variază între 100Hz și 400Hz.

CAN (Controller Area Network) și J1850

Este o magistrală introdusă de Bosch și Intel pentru autovehicule. J1850 a fost introdusă în SUA de SAE (Society of Automotive Engineers) și este aplicată tot la autovehicule. Cele două standarde nu pot coexista, deoarece nivelele de tensiuni și protocoalele sunt diferite. Ambele variante sunt magistrale seriale, cu linii multiplexate cu care se poate lucra la fel la nivelul superior (software). Arbitrarea magistralei se realizează, la ambele, prin trimiterea unui identificator de 29 biți. Microcontrollerele construite pentru utilizarea în domeniul auto sunt echipate cu astfel de bus-uri.

MICROWIRE/PLUS

Este o magistrală serială, sincronă, bidirecțională, introdusă de National Semiconductor Corporation. Compania echipează cu acest bus microcontrollere, convertoare, display-uri, EEPROM etc.