

## 2. Interfețe

---

### 2.1. Aspecte teoretice privind cuplarea prin CENTRONICS și pe magistrală. Programe "DEBUGGER"

Echipamentele periferice standard au un principiu bine determinat de legare la calculator (exemplu hard discul prin cuplorul MIO pe magistrala de 16 biți, imprimanta prin interfața CENTRONICS). Proiectanții sunt puși în situația în care echipamentele concepute de ei (sisteme de achiziții de date, convertoare A/D, D/A în diferite aplicații) trebuie legate la un calculator. Ei optează pentru un calculator compatibil IBM PC deoarece este cel mai răspândit și asigură optimul raportului performanță preț pentru aplicațiile obișnuite. Proiectantul are de ales în principal între 3 soluții constructive:

- legarea la interfața CENTRONICS;
- legarea pe magistrală;
- legarea prin interfața serială RS232;

Unde este cel mai favorabil să se cupleze fiecare aplicație?

Acest subcapitol încearcă să răspundă acestei întrebări, arătând câteva soluții constructive. Se dezbate avantajele și dezavantajele fiecărei soluții, mai ales din p.d.v. al siguranței în funcționare.

#### 2.1.1. Cuplarea prin interfața CENTRONICS

Interfața CENTRONICS este o interfață paralelă de 8 biți, aflată pe placa MIO (Multi I/O), împreună cu cuplorul de hard disc, disc flexibil și 2 interfețe seriale V24. Ea a fost concepută pentru cuplarea unei imprimante. La tipurile mai vechi de calculatoare interfața CENTRONICS se află pe aceeași placă cu interfața pentru monitorul monocrom HERCULES. Semnalele interfeței CENTRONICS sunt disponibile la o cuplă DB de 25 pini mamă. Semnalele, numărul pinului și sensul sunt date în continuare:

- DATA0-D7, pini 2,3,4,5,6,7,8,9 , ieșire din calculator;
- STB, pin 1, ieșire;
- ACK, pin 10, intrare;
- BUSY, pin 11, intrare;
- PE, pin 12, intrare;
- SELECT, pin 13, intrare;
- ERROR, pin 15, intrare;
- AUTO FEED, pin 14, ieșire;
- INIT, pin 16, ieșire;
- SELECT IN, pin 17, ieșire;
- GND, pin 18-25, masa.

Se constată că există asociații celor 8 biți de date DATA0-7 un număr de 4 biți de ieșire (CONTROL0-3) și 5 biți de intrare în calculator (STATUS0-4). Adresele celor două porturi paralele (LPT1 și LPT2) sunt 378h și 278h. Majoritatea cuploarelor MIO permit configurarea adresei interfeței CENTRONICS ca LPT1 sau LPT2. Pentru LPT1, la următoarele adrese se află:

**378h-** latch-ul de date;

**37Ah-** octetul de control (CONTROL0-3) B0-B7, în care B7, B6, B5 sunt 0, B0 este STB, B1 este AUTO FEED, B2 este INIT, B3 este SELECT IN, B4 este IRQ ENABLE, adică întrerupere hard validată (se generează de căderea în 0 a lui ACK și este conectată la nivelul IRQ 7);

**379h-** octetul de stare (STATUS0-4), este READ ONLY, de forma B0- B7, în care B2, B1, B0 sunt 0, B3 este ERROR, B4 este SELECT, B5 este PE, B6 este ACK și B7 este BUSY.

Ca port de ieșire, interfața CENTRONICS poate livra spre ieșire 8 biți +4 biți în doi pași, dar ca port de intrare poate achiziționa doar 5 biți, ceea ce constituie un dezavantaj serios. Pentru achiziția mai multor biți este nevoie de mai mulți pași, deci apare un consum de timp suplimentar.

Pentru a realiza o interfață pe 16 biți de date, 8 biți de adresă, 2 biți de comandă, care să poată lucra atât prin transfer programat cât și prin întreruperi, se poate folosi o schemă ca cea din figurile următoare.

Pentru achiziția a 16 biți de date este nevoie de 4 pași, obținuți prin multiplexarea datelor, comandată prin decodificarea semnalelor CSMUX0 și CSMUX1, figura 2.1.

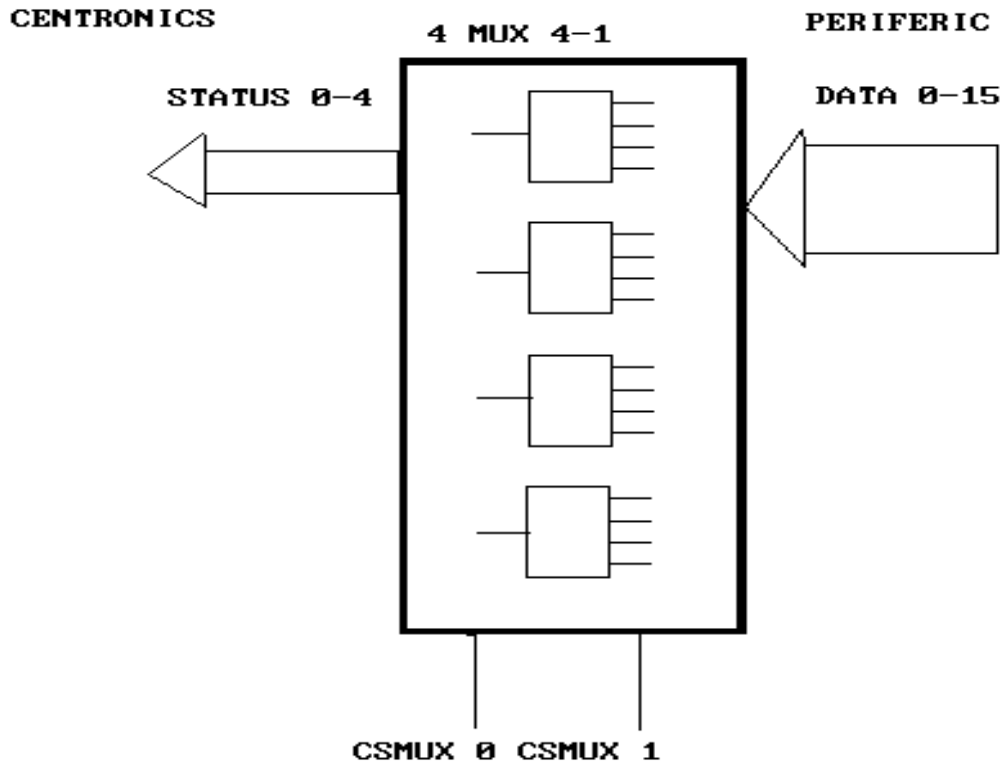


Figura 2.1

Pentru ieșirea datelor pe 16 biți, ele se înscriu în doi pași în LATCH-uri, comanda fiind realizată cu CS LATCH 0 și CS LATCH 1. Cele două LATCH-uri sunt cu ieșirile cu 3 stări, figura 2.2.

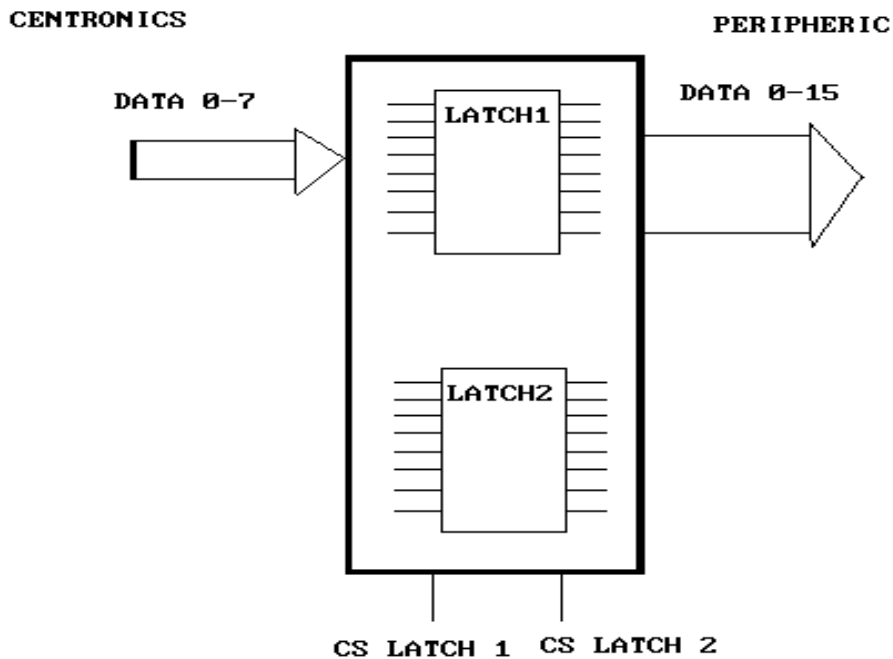


Figura 2.2

Cu ajutorul unui LATCH se formează cei 8 biți de adresă (Figura 2.3).

În ipoteza că nu se fac operații simultan, se poate folosi un decodificator pentru decodificarea comenzilor din semnalele CONTROL 0-3, din aceste comenzi deducându-se semnalele care informează perifericul că se face o citire (RD) sau scriere (WR). Pentru a se lucra cu sistemul de

întreruperi, cererea de întrerupere de la periferic se conectează la una din liniile STATUS, și anume ACK.

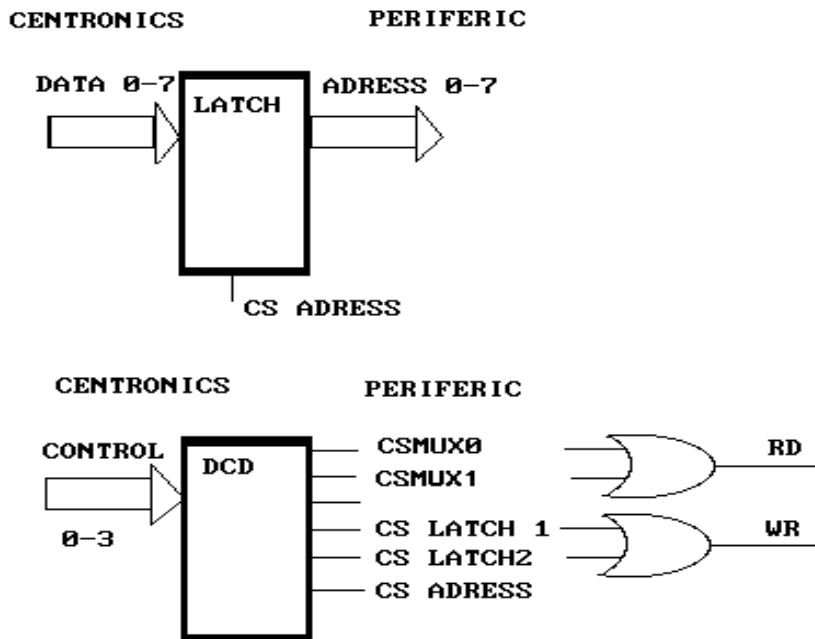


Figura 2. 3

Pentru a aprecia viteza maximă de transfer se consideră un calculator cu frecvența de tact de 50MHz (cu procesor 80486 DX sau DX2). Un program de transfer include câteva instrucțiuni, astfel:

```
IN AX, (port)
MOV M, AX
INC (DEC) BX
```

Durata acestui program este de 440 ns. Comenzile pentru selectarea grupurilor de 4 biți (4 seturi de comenzi) :

```
MOV AX, octet
OUT (port), AX
```

Pe magistralele cunoscute (ISA, EISA, MCA), transferul nu se face cu tactul unității centrale ci cu tactul de magistrală BCLK (SYSCLK) care poate fi 6, 8, 10, 12.5 MHz, programabil. O operație de I/O durează 3 cicluri (0WS), deci 240 ns. Ținând cont de timpul necesar tuturor comenzilor de selecție și achiziție se estimează că viteza de transfer nu poate depăși 250 K cuvinte de 16 biți/s. Oricum această viteză de achiziție este mult superioară vitezei maxime de transfer prin V24.

Pentru achizițiile de date care necesită o viteză de transfer mai mare se poate folosi o memorie locală. Atunci achiziția va avea loc în timp real, iar transferul de date în timp amânat. O astfel de achiziție este descrisă în (DEM94).

Universalitatea legării la interfața CENTRONICS se apreciază prin numărul tipurilor de calculatoare la care există acest tip de interfață. Toate calculatoarele IBM de tip PS1 și PS2 și compatibile, construite de oricare firmă, laptop-urile și notebook-urile sunt echipate cu acest tip de interfață. Se poate spune că orice calculator are o interfață CENTRONICS, iar în cazul în care aceasta întâmplător nu a fost cumpărată, prețul ei 12-18 USD o face accesibilă oricărui utilizator.

Observație: interfețele construite pentru IBM XT erau prevăzute cu un jumper prin care se putea selecta ca portul cu cei 8 biți de date să fie definit de intrare, ceea ce modifică esențial datele problemei, ușurând folosirea interfeței pentru achiziția de date. Modelele mai noi de MIO cu bus local VESA, așa cum este WINBOND au de asemenea această posibilitate.

### Siguranța în funcționare.

Din punct de vedere tehnologic interfața se realizează în echipamentul periferic extern, fiind cuplat la calculator prin intermediul unui cablu cu conector DB 25 pini tată.

Un prim dezavantaj care se reliefează este faptul că interfața nu beneficiază de calitățile antiperturbative ale carcasei calculatorului, nefiind în interiorul carcasei. Carcasa calculatoarelor

compatibile IBM PC a fost atent concepută și proiectată pentru a minimiza efectele perturbațiilor electromagnetice.

Avantajul acestei soluții este că legătura cu calculatorul se face printr-un conector cu 25 de pini, care se fabrică industrial, este de bună calitate, ieftin și se obține ușor. De asemenea pentru montarea acestei interfețe nu se desface calculatorul. Conectarea prin intermediul interfeței CENTRONICS face ca un defect la interfața aplicației, în cel mai defavorabil caz poate afecta doar interfața CENTRONICS.

### 2.1.2. Cuplarea pe magistrală

Semnificația semnalelor și configurația pinilor pentru câteva din magistralele actuale (ISA și MCA) sunt date în bibliografie. La magistrală sunt disponibile toate liniile de date (16 sau 32), liniile de adresă și liniile de comenzi.

#### Cuplarea la magistrala ISA de 8 biți

În primul rând trebuie stabilit spațiul de adrese libere pentru calculatoarele compatibile IBM PC. Decodificarea adresei se poate face cu un comparator binar sau cu un decodicator, ca în figura 2.4 a și b:

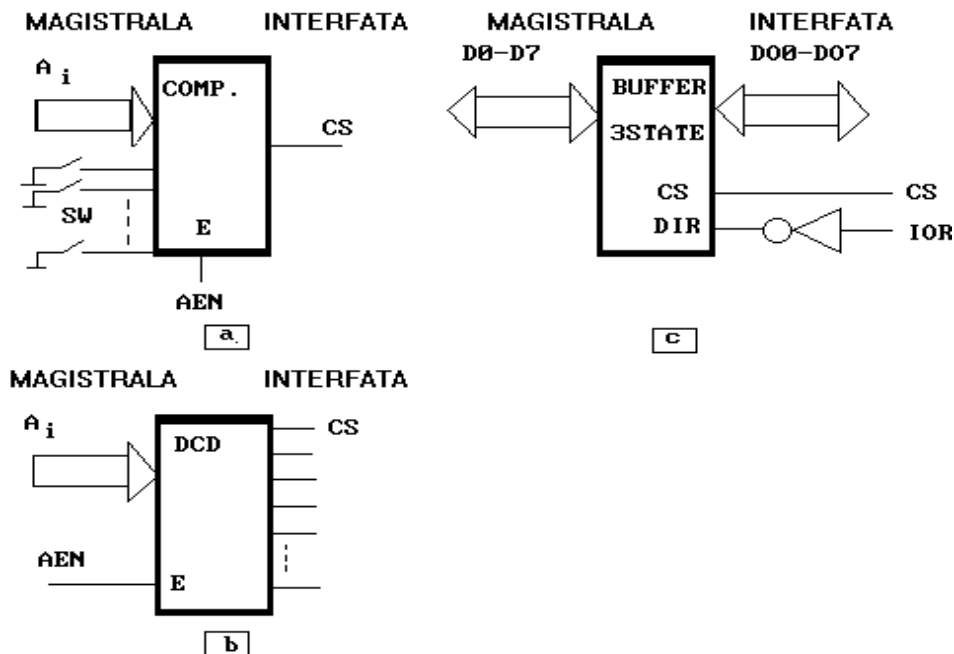


Figura 2.4

Comparatorul identifică configurația liniilor de adresă  $A_i$  identică cu cea selectată din blocul de comutatoare  $SW$  și generează  $CS$  activ. Adresa la care răspunde interfața este configurabilă. În al doilea caz, la o anumită configurație a liniilor de adresă se activează  $CS$ , adresa nefiind configurabilă. În ambele cazuri validarea selecției se face cu  $AEN$  (Address Enable). Liniile de date se pot trece prin buffere bidirecționale, comandate ca în figura 24 c. Bufferul este validat de semnalul de selecție interfață  $CS$ , iar sensul transferului este comandat de semnalul  $IOR$ . Dacă se dorește buffer pe liniile de adresă se pot folosi buffere unidirecționale (dacă nu se folosește un controller DMA extern).

În spațiul de adrese pot fi folosite adresele 0210-021Fh, 0300- 031Fh care sunt libere întotdeauna sau 02B8-02BFh (LPT3), 02E8- 02EFh (COM4), 03E8-03EF (COM3), 0380-038Fh (SDLC/BISYNC2), 03A0- 03AFh (BISYNC1). Interfața poate lucra în întreruperi și poate folosi o linie neutilizată (IRQ2, 3, 4, 7) sau prin DMA folosind semnalele de protocol DMA pentru un canal neutilizat- DMA1 (SDLC).

#### Cuplarea la magistrala ISA de 16 biți

La această magistrală există posibilitatea de a se efectua transferuri pe 8 sau pe 16 biți. Semnalele care comandă acest lucru sunt A0 (linia de adresă A0) și BHE (Bus High Enable), astfel:

A0	BHE	
0	0	D0-D15 transfer pe 16 biți;
1	0	D8-D15 transfer pe octetul superior;
0	1	D0-D7 transfer pe octetul inferior;

O schemă simplificată a transferului de date este dată în figura 2.5:

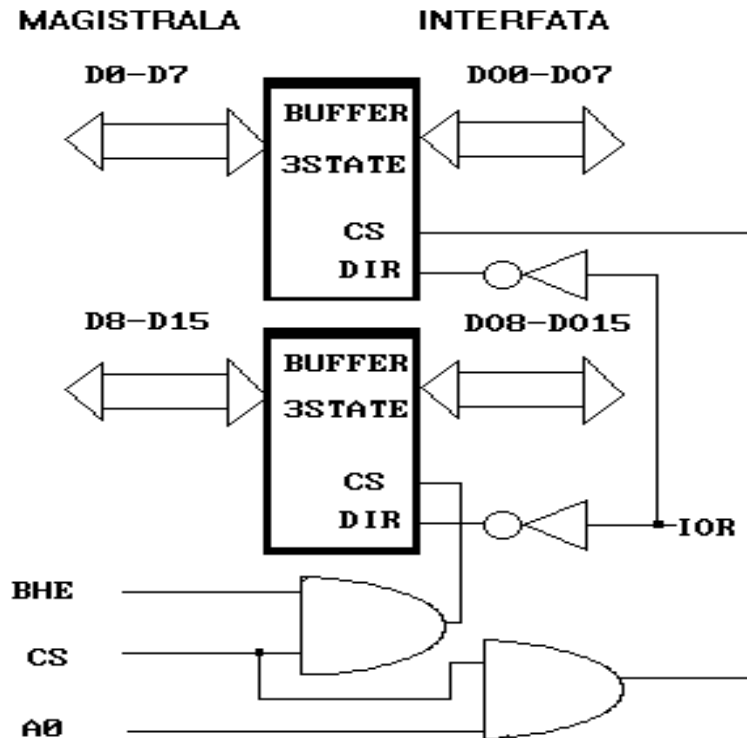


Figura 2.5

Se poate face transfer prin intermediul întreruperilor pe nivelele libere IRQ 9, 10, 11, 15, sau prin DMA prin canalele libere DMA5, 6, 7.

Pentru a se aprecia viteza maximă de transfer pe magistrala ISA de 16 biți se observă că se pot face transferuri în cuvinte de 16 biți într-un singur pas, atât prin transfer programat cât și prin întreruperi sau DMA. Pentru transfer programat programul necesită o secvență ca și cea de la CENTRONICS:

```
IN AX, (port)
MOV M, AX
(INC)DEC BX
```

Secvența durează la un BCLK de 12.5MHz circa 1040 ns, ceea ce înseamnă o viteză de transfer de 960 Kcuvinte de 16 biți/ s. Transferul prin întreruperi nu este mai rapid, dar are avantajul că cererea de transfer poate veni în orice moment.

O asemenea placă cu 2 circuite I8255 a fost realizată experimental. Putem estima o dublare a vitezei de transfer deoarece datele sunt transferate în cuvinte de 16 biți și nu de 8 biți, pe un hardware identic cu cel de la transferul de 8 biți.

Viteza de transfer se poate mări în dauna flexibilității aplicației folosind în loc de I8255, care este o interfață lentă o interfață bidirecțională, cu trei stări, rapidă, dar neprogramabilă. Selecția ei se face la fel prin decodificare de adrese, iar sensul transferului este comandat de /IOR, figura 2.6:

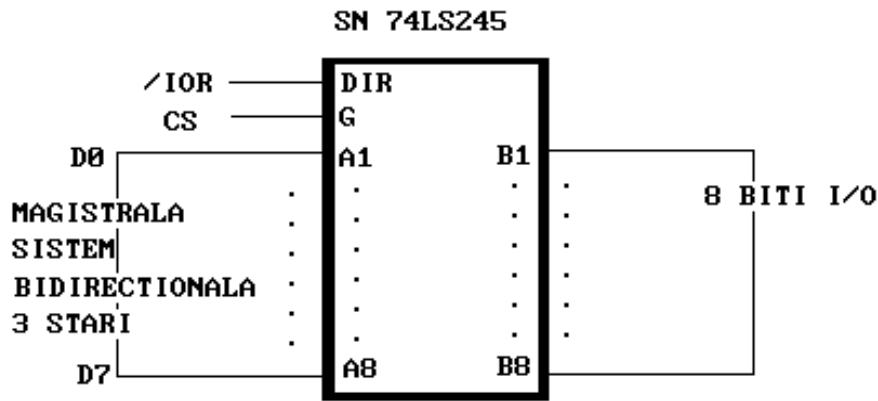


Figura 2.6

Un transfer mai rapid este posibil prin DMA. Un cuvânt se transferă în 8 tacti BCLK, ceea ce înseamnă o rată maximă de transfer de 1562 Kcuvinte/ s. În cel mai simplu mod se poate folosi unul dintre canalele DMA libere.

### Siguranța în funcționare.

O interfață pe magistrală trebuie să fie astfel construită încât să poată fi introdusă într-un conector mamă, al magistralei. Contactele pe placa interfeței trebuie să fie depuse pe cablaj. Calitatea acestor contacte este determinantă pentru siguranța în funcționare a interfeței.

Pentru ISA de 8 biți sunt 62 de contacte, pentru ISA de 16 biți 98 de contacte, ca în figura 8.7.a, iar pentru EISA sunt 124+72 de contacte ca în figura 2.7 b.

Aceste dimensiuni arată că precizia de execuție a cablajului trebuie să fie mare. Adăugând că este necesară aurirea contactelor pentru a se obține un contact sigur, se reliefează primul dezavantaj, și anume dificultatea de realizare tehnologică. O siguranță bună în funcționare se obține numai cu plăci executate prin procedee industriale. Numărul mare de contacte diminuează siguranța în funcționare.

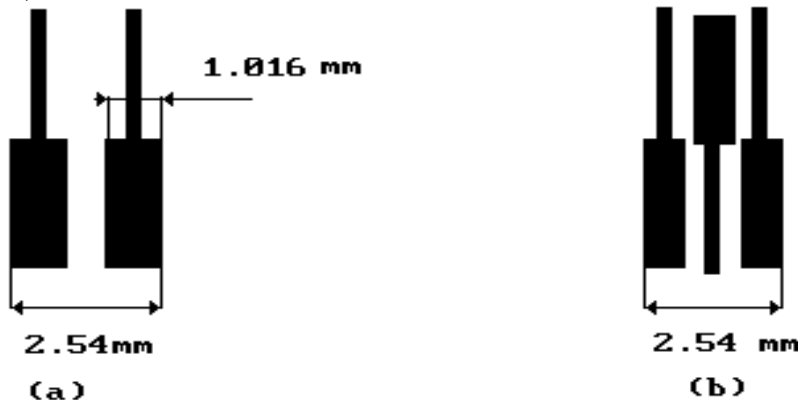


Figura 2.7

Situarea interfeței în interiorul carcasei constituie un avantaj, din cauza măsurilor de ecranare și protecție EM luate la conceperea carcasei. Acest avantaj este umbrat de necesitatea de a aranja traseele pe cablaj pe considerente EMC, pentru ca circuitul să funcționeze corect și să nu periclitizeze funcționarea celorlalte plăci ale calculatorului.

Ca dezavantaj poate fi menționat faptul că un defect în sistemul de achiziție poate afecta oricare placă din calculator. O suprapunere pe liniile de întrerupere sau în zonele adresate poate deasemenea produce neplăceri, chiar sub forma unor defecte intermitente.

Gradul de universalitate este redus, deoarece interfațele nu se pot cupla la calculatoare de tip laptop sau notebook, sau cu un alt tip de magistrală decât cel pentru care a fost concepută interfața (cu excepția unei interfețe concepute pentru ISA de 8 biți care funcționează pe ISA de 16 biți).

**Comparații:**

Avantajele și dezavantajele legării la CENTRONICS sau magistrală sunt trecute în următorul tabel:

	<b>CENTRONICS</b>	<b>ISA 8</b>	<b>ISA 16</b>	<b>EISA</b>
<b>Viteza max. de transfer (programat sau prin întreruperi)</b>	250K cuvinte 16 bit/s	480Kcuvinte 16 bit /s	960Kcuvinte16 bit/s	1380Kcuvinte 16 bit/s
<b>Viteza max. de transfer (DMA)</b>	-----	780Kcuvinte 16 bit /s	1560Kcuvinte 16 bit /s	2700Kcuvinte 16 bit/s
<b>Complexitate (12 bit ieșire 4 bit intrare)</b>	simplă (0 cipuri)	complexă (7 cipuri)	complexă (10 cipuri)	complexă (12 cipuri)
<b>Complexitate (16bit intrare 16 bit ieșire)</b>	complexă (12 cipuri)	complexă (7 cipuri)	complexă (10 cipuri)	complexă (12 cipuri)
<b>Universalitate</b>	totală	parțială	parțială	parțială
<b>Tehnologie de realizare</b>	simplă	complexă	complexă	complexă
<b>Probleme EMC</b>	cablul-carcasa-cablajul	cablajul	cablajul	cablajul
<b>Siguranța în funcționare</b>	bună	satisfăcătoare	satisfăcătoare	satisfăcătoare
<b>Riscuri la defectare</b>	numai pt. interfață	pt. întreg calculatorul	pt. întreg calculatorul	pt. întreg calculatorul

**2.1.3. Cuplarea pe magistrale locale**

După cum am amintit la punctul anterior magistrala ISA nu mai face față la vitezele tot mai mari de transfer și la lărgimea bus-ului de date care este acum de cel puțin 32 de biți la majoritatea procesoarelor de pe piață (486 sau P5). Magistralele EISA și MCA nu s-au răspândit atât de mult cum s-au răspândit magistralele locale VESA (Video Equipment Standards Association) și PCI (Peripheral Connect Interface, dezvoltată de INTEL). Schema bloc a unui sistem care este echipat atât cu magistrală ISA pentru periferice lente cât și cu magistrală locală este dată în figura 2.8:

Magistralele locale sunt conectate prin controlere specializare. Controllerul VESA permite lucrul sincronizat cu semnale de tip READY și se poate selecta pe placă lucrul cu frecvențe mai mari sau mai mici decât 33MHz. Controllerul PCI poate lucra cu frecvențe de 33, 40, 66 MHz, în funcție de tipul echipamentului periferic.

Iată câteva din rezultatele obținute prin măsurarea timpului de transfer de date cu un hard disc la 2 plăci echipate cu procesor INTEL 486 DX/ 66MHz, una cu magistrală locală VESA și una cu PCI:

1. VESA, -viteza de transfer prin ISA (16 biți) - 1233Kocteți/s;  
-viteza de transfer prin VESA (32 biți) - 1495Kocteți/s;
2. PCI, -viteza de transfer prin ISA (16 biți) - 1233Kocteți/s;  
-viteza de transfer prin PCI (32 biți) - 1804Kocteți/s.

Măsurătorile s-au efectuat în următoarele condiții:

- hard disc de tip QUANTUM 420Mocteți, 12,7ms timp mediu de acces și 4,0ms timp de acces de la o pistă la alta;
- pentru VESA driver soft VG4;
- pentru PCI driver soft ATC;
- atât placa de bază cât și cuploarele sunt echipate cu setul de chip-uri UMC, plăcile de bază fiind echipate cu 256Kocteți memorie CACHE;
- nu a fost definit CACHE de hard disc soft.

Construirea unei interfețe pentru cuplarea pe magistrale locale este dificilă atât din punct de vedere constructiv cât și din cauza lipsei documentației.

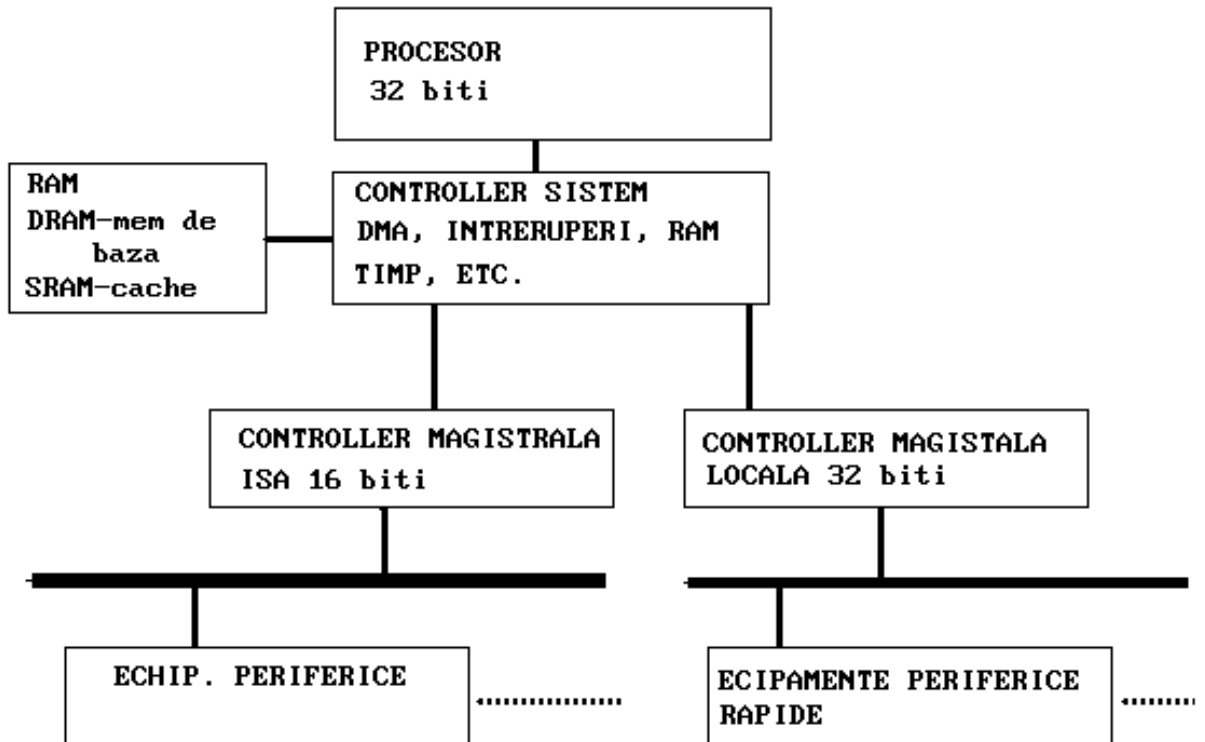


Figura 2.8

### 2.1.4. Programe "DEBUGGER"

Programele "DEBUGGER" sunt programe folosite pentru rularea unor programe foarte simple în limbaj de asamblare. Aceste programe simple, scrise în timpul laboratoarelor de Interfețe vor folosi pentru trimiterea unor comenzi către interfața CENTRONICS sau spre interfețele cuplate pe magistrală. Programele "DEBUGGER" mai răspândite sunt DEBUG livrat odată cu sistemul de operare MSDOS și SID, livrat în sistemul DRDOS.

#### Comenzile programului **DEBUG** (sub MSDOS)

**assemble** A [address]  
**compare** C range address  
**dump** D [range]  
**enter** E address [list]  
**fill** F range list  
**go** G [=address] [addresses]  
**hex** H value1 value2  
**input** I port  
**load** L [address] [drive] [firstsector]  
 [number]  
**move** M range address  
**name** N [pathname] [arglist]  
**output** O port byte  
**proceed** P [=address] [number]  
**quit** Q  
**register** R [register]  
**search** S range list  
**trace** T [=address] [value]  
**unassemble** U [range]  
**write** W [address] [drive] [firstsector]  
 [number]

**allocate expanded memory** XA [#pages]  
**deallocate expanded memory** XD [handle]  
**map expanded memory pages** XM [Lpage]  
 [Ppage] [handle]  
**display expanded memory status** XS

#### Comenzile programului **SID** (sub DRDOS)

? Help ?? Command formats  
 :name Define a macro (You're prompted to enter the macro body)  
 =name Invoke a macro = List all defined macros.  
**A** Assemble into memory  
**B** Block compare  
**D** Display memory -D Set default nr of bytes to display  
**E** Load (for Execution) program & symbol file[s]  
**F** Fill memory  
**G** Go (with optional temporary breakpoints)  
**H** Hexadecimal arithmetic  
**I** Set up program arguments  
**L** List memory (disassemble)



**M** Move (copy) memory block  
**P** Pass points (i.e., breakpoints)  
**Q** Direct I/O request - Or QUIT  
**R** Read disk file  
**S** Set memory SR Search  
**T** Trace (single-step execution)  
**U** Untraced single-step execution  
**V** Verify values of last file loaded  
**W** Write disk file  
**X** Examine [or modify] CPU state  
**Z** Display 8087 Math Co-processor registers  
 #  
**A s** Assemble into memory, at address s  
**B s1,f,s2** Block compare memory at s1  
 (through f) vs block at s2  
**D[W]s,f** Display memory (-D sets default nr  
 of bytes to display)  
**E file1 [-file2]** Load files and symbols; E  
 alone frees memory  
 try '-' or '+' preceding file2 for 'large'  
 symbol addresses  
**F[W]s,f,n** Fill memory from s to f with n  
**[-]G[s1],[s2],[s2]** Go at s1; s2 & s3 are  
 breakpoints; '-' quiets it.  
**H[n1],[n2]** Show symbols; show n1; show  
 n1+n2, n1-n2, n1\*n2 and n1/n2  
 #

**Istring** Set up command tail (command line  
 arguments)  
**[-]L[s,f]** List memory (disassemble) from s to  
 f; '-' won't show symbols  
**Ms1,f,s2** Move (copy) memory block from  
 s1 (through f) to s2  
**[-]P[s,n]** Set/clear/list breakpoints (called  
 "pass points" here)  
**QI[W]n1 or QO[W]n1,n2** Input from (or  
 output n2 to) port n1 ...or...  
**QRs,drive,sector,count** or QW... Read/Write  
 count sectors from drive to s  
**Rfile** Read disk file  
**[-]S[W][s]** Store into memory or  
 enable/disable segment register display  
**SRs,f,"string"** Search through memory  
**[-]T[W][n]** Trace/step execution  
**[-]U[W][n]** Untraced step execution  
**V** Values of last file read  
**Wfile[s,f]** Write memory (block from s to f) to  
 disk file  
**X[R|S]** Examine CPU state [alter: R ==  
 registers; S == a state flag]  
**Z** Display 8087 Math Co-processor registers