

2. PORTUL PARALEL ÎMBUNĂTĂȚIT

2.1. Scopul lucrării

Lucrarea prezintă portul paralel îmbunătățit al calculatoarelor IBM PC, pe baza standardului *IEEE 1284*. Sunt prezentate modurile de transfer specificate de acest standard, cu protocolul utilizat pentru fiecare. Se descriu metodele care pot fi utilizate pentru detectarea tipului de port paralel din calculator, și posibilitatea utilizării modului ECP pentru transferul datelor între două calculatoare.

2.2. Considerații teoretice

2.2.1. Prezentare generală a standardului *IEEE 1284*

La apariția perifericelor cu performanțe mai ridicate, portul paralel standard a fost utilizat și pentru conectarea acestor periferice. Problemele întâmpinate de proiectanții și utilizatorii acestor periferice se pot împărți în trei categorii. În primul rând, deși performanțele calculatoarelor au crescut considerabil, arhitectura și performanțele portului paralel au rămas practic neschimbate. Rata maximă de transfer care se poate obține cu această arhitectură este în jur de 150 KB/s, procesorul fiind solicitat în mod intens. În al doilea rând, nu exista un standard pentru interfața electrică, ceea ce conducea la numeroase probleme atunci când se încerca garantarea operațiilor de transfer pentru diferite platforme. În sfârșit, lipsa standardelor de proiectare a determinat o limitare a lungimii cablurilor externe sub 2 m.

În 1991 a avut loc o întâlnire a producătorilor de imprimante cu scopul inițierii unui nou standard pentru controlul inteligent al imprimantelor conectate într-o rețea. Acești producători, printre care *IBM*, *Lexmark*, *Texas Instruments*, au format o asociație numită "*Network Printing Alliance*" (*NPA*). *NPA* a definit un set de parametri care, dacă sunt implementați de sistemul de calcul și de imprimantă, asigură controlul complet al operațiilor cu imprimanta. Pentru implementarea acestui standard este necesară însă o conexiune bidirecțională cu performanțe

ridicate cu sistemul de calcul. Conexiunea obișnuită utilizată, portul paralel standard, nu avea performanțele necesare pentru a îndeplini cerințele acestui standard.

NPA a propus institutului *IEEE* crearea unui comitet pentru dezvoltarea unui nou standard al unui port paralel bidirecțional de viteză ridicată pentru calculatoarele personale. Cerința era ca noul standard să fie în totalitate compatibil cu portul paralel original, permițând însă creșterea ratei de transfer la peste 1 MB/s, în ambele sensuri. Acest comitet a elaborat standardul *IEEE 1284*, numit "*Standard Signaling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers*", care a fost aprobat în 1994.

Standardul *IEEE 1284* definește 5 moduri de transfer, care asigură crearea unui canal de conexiune în sens direct (PC la periferic), în sens invers (periferic la PC) sau bidirecțional între calculator și periferic. În ultimul caz, deoarece există un singur set de linii de date, conexiunea este semiduplex, datele fiind transferate într-un singur sens la un moment dat.

Modurile de transfer definite sunt următoarele [20]:

- Modul standard, numit și mod "*Centronics*" sau mod de compatibilitate;
- Modul de transfer pe 4 biți (*Nibble Mode*), utilizând liniile de stare pentru transferul datelor;
- Modul de transfer pe octet (*Byte Mode*), utilizând liniile de date;
- EPP (*Enhanced Parallel Port*), utilizat mai ales de periferice ca unități de bandă, CD-ROM, unități de disc, adaptoare de rețea;
- ECP (*Extended Capability Port*), utilizat în special de noile generații de imprimante și scannere.

Toate porturile paralele pot implementa o legătură bidirecțională prin utilizarea modurilor de compatibilitate și de transfer pe 4 biți. Modurile de compatibilitate și de transfer pe octet pot fi de asemenea utilizate pentru comunicația bidirecțională. Modul de transfer pe octet necesită ca un octet întreg de date să poată fi citit din exterior utilizând liniile de date. Aceasta se realizează de obicei prin adăugarea unui bit de direcție în registrul de control al portului. Acest tip de port este numit în general port paralel bidirecțional. Toate aceste moduri utilizează transferul programat. Procesorul este utilizat în mod intens, rata efectivă de transfer fiind limitată de obicei între 50 și 150 KB/s.

Modurile EPP și ECP permit transferuri bidirecționale ca parte a protocolurilor acestora. Aceste moduri necesită implementarea prin har-

dware a unei mașini de stare care generează automat semnalele de control necesare transferurilor de viteză ridicată. De exemplu, în modul EPP un octet poate fi transferat la periferic printr-o simplă instrucțiune OUT, controlerul de I/E gestionând comunicația cu perifericul.

În ansamblu, standardul *IEEE 1284* definește următoarele:

- 5 *moduri* pentru transferul datelor;
- o metodă prin care calculatorul și perifericul pot determina modurile de transfer disponibile și pot *negocia* modul care va fi utilizat;
- *interfața fizică*: cabluri și conectori;
- *interfața electrică*: drivere/receptoare, terminatoare și impedanțele cablurilor.

Fiecare din modurile de transfer, cu excepția modului de compatibilitate, redefinește semnalele de control și de stare. Pentru fiecare mod se vor utiliza denumirile specifice modului respectiv.

2.2.2. Moduri de transfer

2.2.2.1. Modul de compatibilitate

Acest mod definește protocolul utilizat de portul paralel standard pentru transferul datelor la imprimantă. Datele sunt plasate pe liniile de date, se testează starea imprimantei pentru a determina dacă aceasta este liberă sau ocupată, și apoi se generează prin program un strob de date. Acest transfer este prezentat în Figura 2.1.

Operațiile efectuate sunt descrise în continuare.

1. Se înscrie un octet de date în registrul de date.
2. Se citește registrul de stare pentru a testa semnalul Busy.
3. Dacă semnalul Busy nu este activ, se înscrie în registrul de control configurația corespunzătoare pentru activarea semnalului -Strobe.
4. Se înscrie în registrul de control valoarea necesară pentru dezactivarea semnalului -Strobe.

Pentru transferul unui octet, sunt necesare deci patru instrucțiuni de I/E și cel puțin un număr egal de instrucțiuni suplimentare. Rata de transfer rezultată este suficientă pentru comunicația cu imprimantele matriciale și imprimantele cu laser mai vechi, dar nu și pentru unitățile de disc portabile, adaptoare de rețea și noile generații de imprimante cu laser.

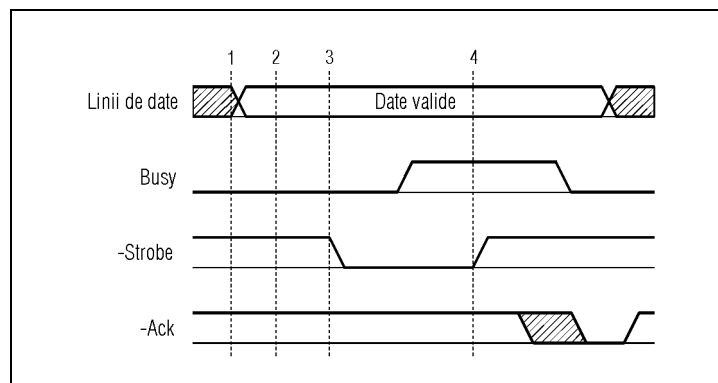


Figura 2.1. Ciclul de transfer în modul de compatibilitate.

Acest mod asigură doar realizarea unui canal în sens direct și trebuie combinat cu un mod care asigură un canal în sens invers pentru a realiza o comunicație bidirecțională. Includerea acestui mod în cadrul standardului s-a realizat pentru a asigura compatibilitatea cu numeroasele tipuri de imprimante și alte periferice existente.

Unele controlere de I/E integrate implementează un mod care utilizează un buffer FIFO pentru transferul datelor cu protocolul modului de compatibilitate. Acest mod este numit "*Fast Centronics*" sau "*Mod FIFO*". Atunci când acest mod este validat, datele înscrise în portul FIFO vor fi transferate la imprimantă utilizând stroburi de date generate prin hardware. Deoarece există o întârziere redusă între transferuri, iar protocolul nu este realizat prin program, cu anumite sisteme pot fi atinse rate de transfer de peste 500 KB/s. Acest mod nu este însă unul din modurile definite de standardul *IEEE 1284*.

2.2.2.2. Modul de transfer pe 4 biți ("Nibble")

Acest mod permite obținerea unui canal în sens invers de la un periferic, fiind combinat de obicei cu modul de compatibilitate pentru a crea un canal bidirecțional.

Toate porturile paralele standard dispun de 5 linii de la periferic la calculator, care pot fi utilizate pentru citirea stării perifericului. Prin utilizarea acestor linii un periferic poate transmite un octet de date în două cicluri de transfer de câte 4 biți. Deoarece linia *-Ack* este utilizată de obicei pentru generarea unei întreruperi, biții utilizați pentru transfer nu sunt împachetați în mod convenabil în cadrul octetului definit de registrul de stare, fiind necesare operații pe biți pentru a obține octetul corect.

În Tabelul 2.1 se prezintă numele semnalelor pentru acest mod.

Tabelul 2.1. *Semnalele modului "Nibble".*

Semnal SPP	Nume în Mod "Nibble"	← In → Out	Descriere
-Strobe	-Strobe	→	Neutilizat pentru transferul de date.
-AutoFeed	HostBusy	→	Setat la 0 atunci când calculatorul este pregătit pentru preluarea datelor (4 biți). Setat la 1 pentru a indica preluarea datelor.
-SelectIn	1284Active	→	Setat la 1 atunci când calculatorul se află într-un mod de transfer 1284.
-Init	-Init	→	Neutilizat pentru transferul de date.
-Ack	PtrClk	←	Setat la 0 pentru a indica date valide, setat la 1 ca răspuns la trecerea semnalului HostBusy la 1.
Busy	PtrBusy	←	Utilizat ca bit 3 de date, apoi ca bit 7 de date.
PaperEnd	AckDataReq	←	Utilizat ca bit 2 de date, apoi ca bit 6 de date.
SelectOut	XFlag	←	Utilizat ca bit 1 de date, apoi ca bit 5 de date.
-Error	-DataAvail	←	Utilizat ca bit 0 de date, apoi ca bit 4 de date.
Data [8:1]	Neutilizate		

Fazele în modul de transfer pe 4 biți sunt următoarele:

1. Calculatorul indică faptul că este pregătit pentru preluarea datelor prin trecerea semnalului HostBusy în 0.
2. Perifericul răspunde prin plasarea primilor 4 biți pe liniile de stare.
3. Perifericul semnalează datele valide prin semnalul PtrClk = 0.
4. Calculatorul setează la 1 semnalul HostBusy pentru a indica recepția datelor și faptul că nu este încă pregătit pentru următorii 4 biți.
5. Perifericul setează la 1 semnalul PtrClk ca semnal de achitare pentru calculator.
6. Stările 1-5 se repetă pentru următorii 4 biți ai octetului.

Figura 2.2 indică un exemplu de transfer de la periferic la calculator.

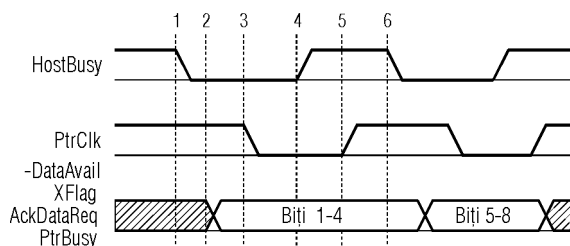


Figura 2.2. Ciclu de transfer în modul "Nibble".

Modul de transfer pe 4 biți, ca și modul de compatibilitate, necesită implementarea protocolului prin program. Acest mod este cel mai solicitant din punct de vedere software, motiv pentru care rata de transfer este limitată la aproximativ 50 KB/s. Această limitare nu are un efect vizibil în cazul perifericelor cu cerințe reduse pentru canalul în sens invers, ca imprimantele, dar nu este acceptabilă în cazul adaptoarelor de rețea, unităților de disc sau unităților CD-ROM.

Avantajul principal al combinării modului de compatibilitate și a modului de transfer pe 4 biți constă în posibilitatea utilizării lor pentru orice calculator cu un port paralel standard.

2.2.2.3. Modul de transfer pe octet

Prin implementările ulterioare ale portului paralel, anumiți producători, ca IBM cu portul paralel al calculatoarelor PS/2, au adăugat posibilitatea de a dezactiva driverule utilizate pentru liniile de date, permițând portului de date să devină un port de intrare. Aceasta permite transmiterea unui octet întreg de date de către un periferic într-un singur ciclu de transfer prin utilizarea celor 8 linii de date, spre deosebire de cele două cicluri necesare în modul "Nibble".

Prin utilizarea modului de transfer pe octet pentru transferurile pe canalul invers se pot obține rate de transfer de la periferic la calculator apropiate de cele ale modului de compatibilitate, de la calculator la periferic. Acest tip de port este denumit uneori ca "port bidirecțional îmbunătățit", creând anumite confuzii cu portul paralel îmbunătățit EPP (*Enhanced Parallel Port*).

Figura 2.3 prezintă un ciclu de transfer, iar Tabelul 2.2 descrie semnalele acestui mod.

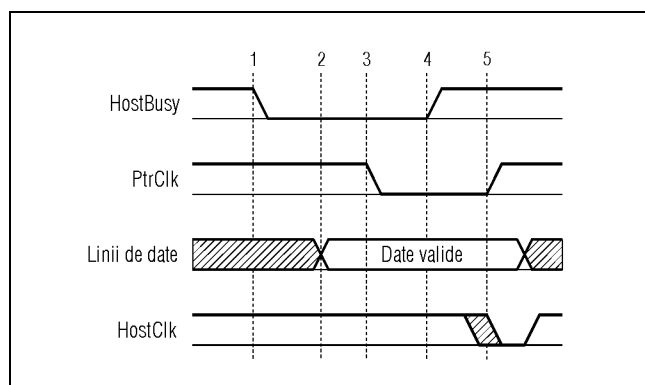


Figura 2.3. Ciclu de transfer în modul "Byte".

Tabelul 2.2. Semnalele modului de transfer pe octet.

Semnal SPP	Nume în Mod "Byte"	← In → Out	Descriere
-Strobe	HostClk	→	La sfârșitul fiecărui transfer de date se generează un impuls negativ al acestui semnal pentru a indica recepția octetului (semnal de achitare).
-AutoFeed	HostBusy	→	Setat la 0 atunci când calculatorul este pregătit pentru preluarea unui octet. Setat la 1 pentru a indica preluarea octetului.
-SelectIn	1284Active	→	Setat la 1 atunci când calculatorul se află într-un mod de transfer 1284.
-Init	-Init	→	Neutilizat; setat la 1.
-Ack	PtrClk	←	Setat la 0 pentru a indica date valide, setat la 1 ca răspuns la trecerea semnalului HostBusy la 1.
Busy	PtrBusy	←	Indică starea Busy a canalului direct.
PaperEnd	AckDataReq	←	Urmărește semnalul -DataAvail.
SelectOut	XFlag	←	Indicator de extensibilitate; nu este utilizat în acest mod.
-Error	-DataAvail	←	Setat la 0 de periferic pentru a indica disponibilitatea datelor pe canalul invers.
Data [8:1]	Data [8:1]	↔	Reprezintă datele transmise de la periferic la calculator.

Operațiile efectuate pentru un transfer sunt următoarele:

1. Calculatorul indică faptul că este pregătit pentru preluarea datelor prin setarea la 0 a semnalului *HostBusy*.
2. Perifericul răspunde prin plasarea primului octet pe liniile de date.
3. Perifericul indică un octet valid prin semnalul $\text{PtrClk} = 0$.
4. Calculatorul setează semnalul *HostBusy* la 1 pentru a indica preluarea unui octet și faptul că nu este pregătit pentru un nou octet.
5. Calculatorul generează un impuls negativ al semnalului *HostClk* ca un semnal de achitare pentru periferic.
6. Operațiile 1-5 sunt repetate pentru următorii octeți.

2.2.2.4. Modul EPP

Protocolul pentru portul EPP a fost dezvoltat inițial de firmele *Intel*, *Xircorn* și *Zenith Data Systems*, ca un mijloc de a asigura o legătură paralelă cu performanțe ridicate, fiind în același timp compatibilă cu portul paralel standard. Acest protocol a fost implementat de *Intel* în setul de cipuri 386SL (circuitul de I/E 82360), înainte de începerea elaborării standardului *IEEE 1284*. Protocolul EPP a oferit numeroase avantaje producătorilor de periferice și a fost adoptat ca o metodă opțională de transfer, fiind formată o asociație pentru promovarea acestui protocol. Asociația a promovat adoptarea acestui protocol ca unul din modurile avansate ale standardului *IEEE 1284*.

Protocolul EPP definește patru tipuri de cicluri de transfer:

1. Ciclu de scriere date.
2. Ciclu de citire date.
3. Ciclu de scriere adrese.
4. Ciclu de citire adrese.

Ciclurile de date sunt utilizate pentru transferul datelor între calculator și periferic. Ciclurile de adrese pot fi utilizate pentru transferul adreselor, a informațiilor de comandă și control. Tabelul 2.3 descrie semnalele EPP și semnalele SPP asociate acestora.

Tabelul 2.3. Definiția semnalelor EPP.

Semnal SPP	Nume în Mod EPP	← In → Out	Descriere
-Strobe	-Write	→	În starea activă indică o operație de scriere. În starea inactivă indică un ciclu de citire.
-AutoFeed	-DataStrobe	→	Indică o operație de citire sau scriere date în curs.
-SelectIn	-AddrStrobe	→	Indică o operație de citire sau scriere adrese în curs.
-InIt	-Reset	→	Reset pentru periferic.
-Ack	-Intr	←	Înterupere de la periferic.
Busy	-Wait	←	În starea activă indică posibilitatea începerii unui ciclu (activarea strobului). În starea inactivă indică posibilitatea încheierii ciclului (dezactivarea strobului).
Data [8:1]	AD [8:1]	↔	Linii bidirecționale de adrese/date.
PaperEnd	Definit de utilizator	←	Poate fi utilizat în mod diferit de fiecare periferic.
SelectOut	Definit de utilizator	←	Poate fi utilizat în mod diferit de fiecare periferic.
-Error	Definit de utilizator	←	Poate fi utilizat în mod diferit de fiecare periferic.

Figura 2.4 prezintă un exemplu de ciclu de scriere date. Semnalul -IOW al procesorului este figurat doar pentru a pune în evidență că toate operațiile au loc într-un singur ciclu de I/E.

Fazele ciclului de scriere date sunt următoarele:

1. Programul execută un ciclu de I/E de scriere în registrul 4 (port de date EPP).
2. Se activează semnalul -Write și datele se înscriu în portul paralel.
3. Strobul de date este activat, deoarece -Wait este activ.
4. Portul așteaptă pentru achitarea de la periferic (-Wait dezactivat).
5. Strobul de date este dezactivat și ciclul EPP se termină.
6. Ciclul de I/E al magistralei ISA se termină.
7. Semnalul -Wait este activat pentru a indica posibilitatea începerii următorului ciclu.

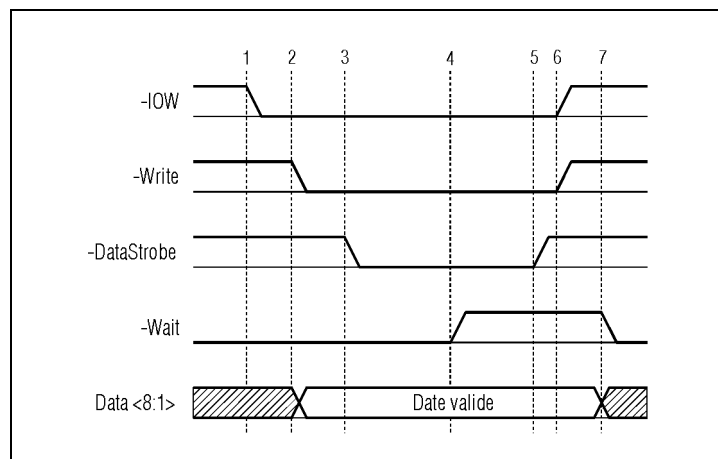


Figura 2.4. Ciclu de scriere date în modul EPP.

Se subliniază faptul că întregul transfer de date are loc într-un singur ciclu ISA. Efectul este că prin utilizarea protocolului EPP pentru transferul datelor se pot atinge rate de transfer între 500 KB/s și 2 MB/s. Astfel, perifericele care utilizează portul paralel pot atinge nivele de performanță apropiate de cele ale plăcilor de extensie ISA.

Modurile de transfer 1284 sunt implementate printr-un protocol cu interblocare (confirmare). Cu un asemenea protocol, transferul de date are loc la viteza interfeței celei mai lente. Interblocarea se referă la faptul că fiecare tranziție a unui semnal de control este confirmată de cealaltă parte a interfeței. În diagrama precedentă, -DataStrobe poate fi activat deoarece -Wait este activ, -Wait se dezactivează ca răspuns la activarea -DataStrobe, -DataStrobe se dezactivează ca răspuns la dezactivarea -Wait, și în final -Wait se dezactivează ca răspuns la dezactivarea -DataStrobe. În acest fel perifericul poate controla timpul de setup necesar pentru operația respectivă. Acest timp este timpul de la activarea -DataStrobe până la dezactivarea -Wait. Interblocarea are de asemenea avantajul că ciclul de transfer este independent de lungimea cablului.

Setul de registre EPP reprezintă o extensie a registrelor portului paralel standard. Registrele EPP sunt prezentate în Tabelul 2.4.

Tabelul 2.4. *Registrele EPP.*

Nume port	Offset	Mod	R/W	Descriere
Port de date SPP	+0	SPP/EPP	W	Port de date standard.
Port de stare SPP	+1	SPP/EPP	R	Citește liniile de stare ale interfeței.
Port de control SPP	+2	SPP/EPP	W	Setează liniile de control.
Port de adresă EPP	+3	EPP	R/W	Generează un ciclu de citire sau scriere adrese.
Port de date EPP	+4	EPP	R/W	Generează un ciclu de citire sau scriere date.
Nedefinite	+5..+7	EPP	n/a	Utilizate în mod diferit de diferitele implementări. Pot fi utilizate pentru I/E de 16 sau 32 biți.

Prin generarea unei singure instrucțiuni de scriere în portul de date EPP, controlerul EPP va genera semnalele de protocol pentru transferul datelor într-un ciclu de scriere date EPP. Instrucțiunile de I/E la porturile cu adrese +0..+2 față de adresa de bază au același efect cu cele la portul paralel standard. Astfel se garantează compatibilitatea cu perifericele portului paralel standard. Ciclurile de adrese sunt generate atunci când operațiile de citire sau scriere se efectuează la portul cu adresa +3 față de adresa de bază.

Porturile +5..+7 sunt utilizate în mod diferit de diversele implementări hardware. Acestea pot fi utilizate pentru implementarea interfețelor pe 16 sau 32 de biți, sau ca registre de configurație. De exemplu, există porturi paralele care pot fi accesate utilizând instrucțiuni de I/E pe 32 de biți, chiar dacă au interfețe de 8 biți. Controlerul ISA va intercepta operația pe 32 de biți și va genera 4 cicluri rapide pe 8 biți. Aceste cicluri suplimentare sunt generate prin hardware și sunt transparente pentru programe. Timpul total pentru aceste cicluri va fi mai mic decât cel pentru 4 cicluri independente de 8 biți.

Posibilitatea de a transfera datele prin utilizarea unei singure instrucțiuni permite porturilor paralele în modul EPP să transfere datele la viteza magistralei ISA. În funcție de implementarea portului calculatorului și de performanțele perifericului, un port EPP poate atinge o rată de transfer între 500 KB/s și 2 MB/s. Această rată de transfer este mai mult decât suficientă pentru a permite adaptoarelor de rețea, unităților CD-ROM, unităților de bandă și altor periferice să funcționeze la o viteză apropiată de cea a magistralei ISA.

2.2.2.5. Modul ECP

Protocolul ECP a fost propus de firmele *Hewlett Packard* și *Microsoft* ca un mod avansat pentru comunicația cu periferice ca imprimante și scannere. Ca și modul EPP, modul ECP asigură o comunicație bidirecțională între calculator și periferic.

Protocolul ECP pune la dispoziție următoarele tipuri de cicluri atât în sens direct, cât și invers:

1. Cicluri de date
2. Cicluri de comenzi

Spre deosebire de modul EPP, atunci când a fost propus protocolul ECP, a fost propusă de asemenea o implementare standard a registrelor. Această propunere definește caracteristici care sunt dependente de implementare și care nu sunt specificate de standardul *IEEE 1284*. Aceste caracteristici cuprind compresia de date *Run Length Encoding* (RLE) pentru adaptoarele din calculator, adresarea canalelor, memorii FIFO pentru canalul direct și cel invers, DMA și I/E programate pentru interfața calculatorului.

Facilitatea RLE permite decompresia și (opțional) compresia în timp real a datelor, cu rate de compresie de până la 64:1 (4:1 în mod tipic). Acest lucru este util în special în cazul imprimantelor și scannerele care transferă imagini de dimensiuni mari, cu șiruri identice de date. Compresia este realizată prin contorizarea octeților identici și transmiterea unui octet RLE care indică numărul de repetări ale următorului octet. Decompresia este realizată prin interceptarea octetului RLE și repetarea corespunzătoare a următorului octet. Un contor RLE egal cu 0 specifică faptul că următorul octet reprezintă un singur octet de date, iar un contor egal cu 127 indică necesitatea repetării următorului octet de 128 de ori.

Adresarea canalelor este diferită conceptual de adresarea în cazul modului EPP, fiind utilizată pentru adresarea mai multor dispozitive logice ale unui singur dispozitiv fizic. De exemplu, în cadrul unui singur dispozitiv multifuncțional poate exista un fax, o imprimantă și un modem, cu un singur port paralel. Fiecare din aceste funcții separate sunt realizate de câte un dispozitiv logic separat. Utilizând adresarea canalelor pentru accesul la fiecare din aceste dispozitive, se pot recepționa date de la modem în timp ce canalul de date al imprimantei este ocupat cu prelucrarea unei imagini. În cazul protocolului modului de compatibilitate, dacă imprimanta devine ocupată, nu se mai pot realiza alte operații de comunicație până la eliberarea canalului de date al imprimantei. În cazul modului ECP, driverul software adresează un alt canal și comunicația poate continua.

Tabelul 2.5 descrie semnalele modului ECP.

Tabelul 2.5. Semnalele modului ECP.

Semnal SPP	Nume în Mod ECP	← In → Out	Descriere
-Strobe	HostClk	→	Utilizat cu PeriphAck pentru transferul datelor sau adreselor în sens direct.
-AutoFeed	HostAck	→	Furnizează informații de stare pentru sensul direct. Utilizat cu PeriphClk pentru transferul datelor în sens invers.
-SelectIn	1284Active	→	Setat la 1 atunci când calculatorul se află într-un mod de transfer 1284.
-Init	-ReverseRequest	→	Activat pentru selectarea sensului invers al canalului.
-Ack	PeriphClk	←	Utilizat cu HostAck pentru transferul datelor în sens invers.
Busy	PeriphAck	←	Utilizat cu HostClk pentru transferul datelor sau adreselor în sens direct. Furnizează informații de stare pentru sensul invers.
PaperEnd	-AckReverse	←	Activat pentru achitarea semnalului -ReverseRequest.
SelectOut	XFlag	←	Indicator de extensibilitate.
-Error	-PeriphRequest	←	Setat la 0 de periferic pentru a indica disponibilitatea datelor pe canalul invers.
Data [8:1]	Data [8:1]	↔	Reprezintă datele transmise între periferic și calculator.

Figura 2.5 prezintă două cicluri de transfer în sens direct. Atunci când semnalul HostAck este activ, are loc un ciclu de date. Dacă semnalul HostAck este inactiv, are loc un ciclu de comenzi, iar datele reprezintă fie un contor RLE, fie o adresă de canal, în funcție de bitul 7 al octetului de date. Dacă acest bit este 0, biții 0-6 reprezintă un contor RLE (0-127). Dacă bitul 7 este 1, biții 0-6 reprezintă o adresă de canal (0-127).

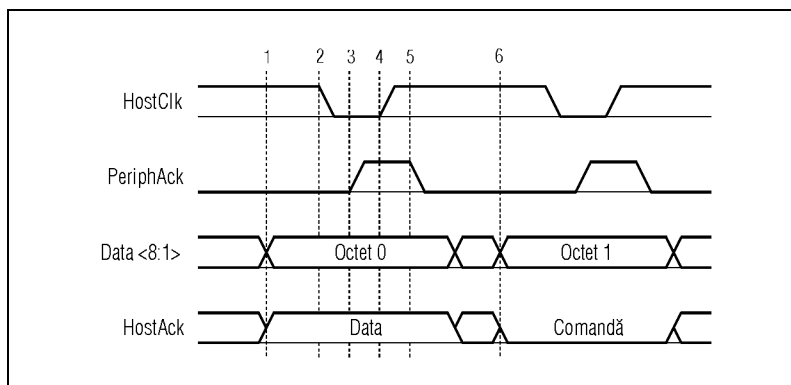


Figura 2.5. Ciclu de date și ciclu de comenzi în sens direct.

Fazele transferului în sens direct sunt următoarele:

1. Calculatorul plasează datele pe liniile de date și indică un ciclu de date prin setarea HostAck la 1.
2. Calculatorul setează HostClk la 0 pentru a indica existența unei date valide.
3. Perifericul răspunde prin setarea PeriphAck la 1.
4. Calculatorul setează HostClk la 1. Acest front al semnalului HostClk trebuie utilizat pentru înscrierea datei la periferic.
5. Perifericul setează PeriphAck la 0, indicând faptul că este pregătit pentru următorul octet.
6. Ciclul se repetă, dar de data aceasta este un ciclu de comenzi deoarece HostAck este 0.

Deoarece există de obicei memorii FIFO în ambele părți ale interfeței, care decuplează transferurile de date pe magistrala ISA de transferurile efective între calculator și periferic, este important de precizat momentul în care data se consideră transferată în cadrul unui ciclu. Acest moment apare în faza 4, când semnalul HostClk devine 1. În acest moment, data trebuie înscrisă în registrul perifericului, și trebuie actualizate contoarele de date. În cadrul protocolului ECP există o condiție care poate determina abandonarea transferului între fazele 3 și 4. În acest caz data nu trebuie considerată ca fiind transferată.

Figura 2.6 prezintă un ciclu de comenzi urmat de un ciclu de date, ambele pentru un canal în sens invers. În același timp, figura pune în evidență și una din diferențele între protocolul ECP și EPP. În cazul protocolului EPP, driverul software poate alterna operații de citire și de scriere fără a fi necesară activarea unor semnale de protocol. În cazul protoco-

lului ECP, schimbarea sensului de transfer a datelor trebuie negociată. Calculatorul trebuie să solicite un transfer în sens invers prin activarea semnalului `-AckReverse`. În plus, deoarece transferul precedent poate fi efectuat prin DMA, programul trebuie fie să aștepte terminarea transferului DMA, fie să întrerupă acest transfer, să golească memoria FIFO pentru a determina numărul de octeți transferați, iar apoi să solicite canalul invers. Aceste operații suplimentare reprezintă un dezavantaj în cazul perifericelor care necesită numeroase operații alternative de scriere și citire ale registrelor sau bufferelor de dimensiuni mici.

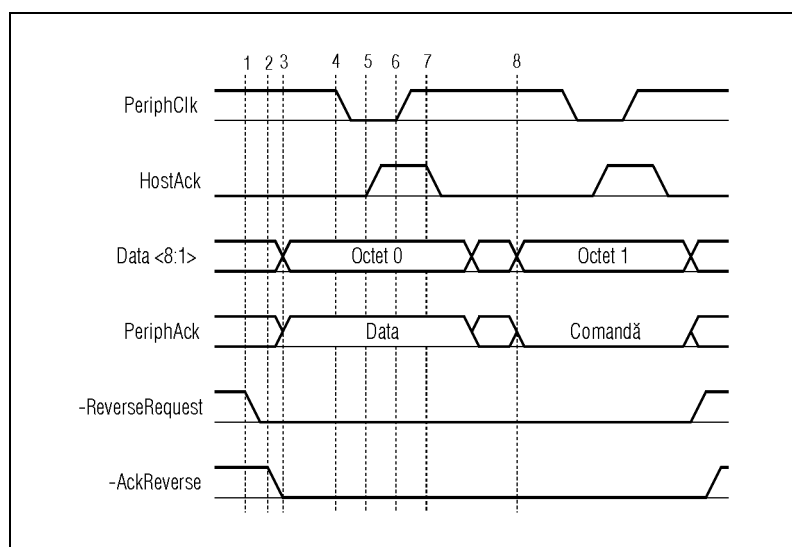


Figura 2.6. Ciclul de date și ciclul de comenzi în sens invers.

Există mai multe moduri de operare ale registrelor ECP. Tabelul 2.6 descrie aceste moduri.

Tabelul 2.6. Modurile registrelor ECP.

Mod	Descriere
000	Mod SPP
001	Mod bidirecțional PS/2 (Byte)
010	Fast Centronics (FIFO)
011	Mod ECP
100	Mod EPP
101	(Rezervat)

Mod	Descriere
110	Mod de test
111	Mod de configurare

Observații.

1. Modul EPP nu este definit de specificațiile ECP. Controlerele 1284, ca de exemplu SMC FDC37C665/666, implementează de obicei acest mod.
2. În modul de test memoria FIFO poate fi citită și înscrisă, dar datele nu vor fi transmise la portul paralel.
3. În modul de configurare sunt accesibile registrele de configurație (cnfgA și cnfgB).

În cazul arhitecturii IBM PC, sunt utilizate numai primele 1024 de adrese de I/E, spațiul de adrese fiind cuprins între 000h și 3FFh. Pentru adresarea acestui spațiu, sunt necesari numai 10 biți de adresă (AD0..AD9). La calculatoarele mai vechi se decodificau doar acești biți de adresă ai magistralei ISA, limitându-se astfel spațiul de I/E disponibil la 1024 de registre. La calculatoarele mai noi se decodifică un număr mai mare de biți, creându-se mai multe "pagini" ale primului 1 KB al spațiului de I/E. Un driver poate accesa aceste pagini prin adunarea valorii 400h la o adresă de bază. De exemplu, prin adresele 378h și 778h se pot accesa două registre din două pagini separate, fără interferență cu alte dispozitive ISA. Avantajul este că noile plăci de extensie pot avea registre "ascunse", extinzându-se numărul de registre disponibile, în același timp păstrându-se compatibilitatea cu plăcile mai vechi.

Modelul registrelor ECP utilizează acest avantaj, și definește 6 registre care necesită numai 3 porturi de I/E. Aceste registre sunt descrise în Tabelul 2.7.

Tabelul 2.7. Descrierea registrelor ECP.

Offset	Nume	R/W	Mod ECP	Funcție
000	Data	R/W	000-001	Data Register
000	ecpAFifo	R/W	011	ECP Address FIFO
001	dsr	R0	Toate modurile	Status Register
002	dcr	R/W	Toate modurile	Control Register
400	cFifo	R/W	010	Parallel Port Data FIFO
400	ecpDFifo	R/W	011	ECP Data FIFO

Offset	Nume	R/W	Mod ECP	Funcție
400	tFifo	R/W	110	Test FIFO
400	cnfgA	R	111	Configuration Register A
401	cnfgB	R/W	111	Configuration Register B
402	ecr	R/W	Toate modurile	Extended Control Register

Dacă portul ECP este în modul standard sau în modul bidirecțional, primele 3 registre operează identic cu cele ale unui port standard, fiind menținută compatibilitatea cu dispozitivele și driverele mai vechi.

Se descriu în continuare o parte din registrele portului ECP [13]. Funcționarea registrelor depinde de câmpul de mod din registrul ecr. Pentru alte moduri decât cele specificate, operarea registrelor este nedefinită.

ecpAFifo (mod 011)

Un octet de date scris în acest registru este plasat în memoria FIFO și este marcat ca o adresă ECP. Acest octet va fi transmis la periferic în mod automat (prin hardware). Utilizarea acestui registru este definită numai pentru sensul direct.

cFifo (mod 010)

Cuvintele înscrise în acest registru, sau transferate prin DMA de la sistem, sunt transmise prin hardware la periferic prin protocolul portului paralel standard. Dacă nu trebuie transferate cuvinte complete, operația trebuie efectuată în modul 000. Acest mod este definit numai pentru sensul direct.

ecpDFifo (mod 011)

Pentru sensul direct, cuvintele înscrise în acest registru, sau transferate prin DMA de la sistem, sunt transmise prin hardware la periferic prin protocolul portului ECP. Dacă nu trebuie transferate cuvinte complete, operația trebuie efectuată în modul 000. Pentru sensul invers, se citesc cuvintele de la periferic în memoria FIFO de date.

tFifo (mod 110)

Se pot citi, scrie sau transfera prin DMA cuvinte între sistem și acest registru, în ambele sensuri. Datele înscrise în acest registru nu vor fi transmise pe liniile portului paralel prin utilizarea protocolului hardware. Datele vor fi transferate la rata maximă a magistralei ISA, astfel că se pot efectua măsurători de performanță prin program.

cnfgA (mod 111)

Acest registru permite obținerea unor informații specifice implementării portului paralel.

Bitul 7 al acestui registru arată dacă întreruperile sunt generate prin impulsuri (bit 7 = 0) sau pe nivel (bit 7 = 1).

Biții 6-4 (implD) pot identifica implementarea portului (printr-un număr ID), și indică dimensiunea cuvântului:

- 000: 16 biți
- 001: 8 biți
- 010: 32 biți

cnfgB (mod 111)

Acest registru permite selectarea prin program a întreruperilor și a canalelor DMA. O implementare RW a acestui registru implică un dispozitiv configurabil prin program. Toate porturile ISA trebuie să implementeze acest registru cel puțin ca un registru R/O.

Bitul 7 (compress) este de tip R/O în cazul în care facilitatea de comprimare nu este implementată, la citire returnând în acest caz valoarea 0. Dacă bitul este setat la 1, datele vor fi comprimate înainte de transmiterea lor. Dacă bitul este 0, vor fi transmise numai date necomprimate.

Bitul 6 (intrValue) returnează valoarea de pe linia IRQ a magistralei ISA pentru a se putea determina conflictele posibile.

Biții 5-3 (intrLine) permit selectarea întreruperii. Dacă acești biți conțin valoarea 000, întreruperea trebuie selectată prin jumpere. Valorile 001..111 selectează, în ordine, IRQ 7 (implicit), IRQ 9, IRQ 10, IRQ 11, IRQ 14, IRQ 15, respectiv IRQ 5.

Biții 2-0 (dmaChannel) permit selectarea canalului DMA. Dacă acești biți conțin valoarea 000 sau 100, canalul DMA trebuie selectat prin jumpere, fiind de 8 biți, respectiv de 16 biți. Valorile 001, 010, 011 selectează canalul 1, 2, respectiv 3 (8 biți). Valorile 101, 110, 111 selectează canalul 5, 6, respectiv 7 (16 biți).

ecr

Registrul ecr (*Extended Control Register*) controlează funcțiile extinse ale portului ECP.

Biții 7-5 indică modul portului ECP (Tabelul 2.6).

Bitul 4 (-ErrIntrEn), valid numai în modul ECP, permite prin valoarea 0 generarea întreruperilor la frontul descrescător al semnalului

-PeriphRequest (-Error). Prin valoarea 1 generarea întreruperilor este invalidată.

Bitul 3 (dmaEn) validează prin valoarea 1 transferurile DMA. Un transfer DMA va începe atunci când bitul serviceIntr devine 0.

Bitul 2 (serviceIntr) dezactivează prin valoarea 1 transferurile DMA și toate cazurile de întrerupere.

Bitul 1 (full) indică prin valoarea 1 faptul că memoria FIFO nu poate accepta un alt cuvânt (dacă bitul Dir al registrului de control este 0) sau este complet plină (Dir = 1). Dacă bitul full este 0, memoria FIFO are cel puțin un cuvânt liber (Dir = 0) sau cel puțin un octet liber (Dir = 1). Acest bit este de tip R/O.

Bitul 0 (empty) indică prin valoarea 1 faptul că memoria FIFO este complet goală (Dir = 0) sau conține mai puțin de un cuvânt (Dir = 1). Dacă bitul empty este 0, memoria FIFO conține cel puțin un octet (Dir = 0) sau cel puțin un cuvânt (Dir = 1). Acest bit este de tip R/O.

Utilizarea portului ECP este similară cu cea a portului EPP. Se înscrie un mod de operare în registrul ecr, iar apoi transferul de date este realizat prin scrierea sau citirea portului de I/E corespunzător. Toate semnalele de protocol sunt generate automat de controlerul interfeței. Diferența principală este că în cazul portului ECP transferurile sunt realizate de obicei prin DMA și nu prin I/E programate.

2.2.3. Negocierea modului de transfer

Perifericele nu trebuie să implementeze toate modurile interfeței *IEEE 1284*. De aceea, este necesară o metodă prin care calculatorul poate determina posibilitățile perifericului și poate seta interfața într-unul din modurile disponibile. În acest scop s-a elaborat conceptul de *negociere*.

Negocierea este o secvență de operații ale interfeței portului paralel, care nu afectează un dispozitiv mai vechi, dar asigură identificarea unui periferic compatibil cu standardul *IEEE 1284*. Ideea este că un periferic mai vechi nu va răspunde la secvența de negociere, iar calculatorul va rămâne într-un mod compatibil cu perifericul, în timp ce un periferic *1284* va răspunde la această secvență și va putea fi setat în oricare mod disponibil.

În timpul fazei de negociere, calculatorul plasează un cod de cerere pe liniile de date și apoi inițiază secvența de negociere. Cererea poate fi de a plasa interfața într-un anumit mod, sau de a solicita un

identificator de dispozitiv (ID) de la periferic, care va permite calculatorului identificarea tipului de periferic atașat. Identificatorul poate fi returnat în oricare mod de transfer în sens invers, diferit de EPP. În timpul negocierii se utilizează un *octet de extensibilitate*. Semnalul XFlag este utilizat de periferic pentru a confirma faptul că modul cerut este disponibil. Acest semnal va fi setat la 1 ca o confirmare pozitivă pentru toate cererile, cu excepția modului de transfer pe 4 biți ("Nibble"). Toate dispozitivele compatibile cu standardul *IEEE 1284* trebuie să admită modul "Nibble" pentru transferurile prin canalul invers.

În Tabelul 2.8 se descriu biții octetului de extensibilitate. Bitul Extensibility Link este prevăzut pentru a asigura un mecanism de adăugare ulterioară a noi caracteristici și moduri de operare.

Tabelul 2.8. Valori ale octetului de extensibilitate.

Bit	Descriere	Valori valide
7	Cerere "Extensibility Link"	1000 0000
6	Cerere mod EPP	0100 0000
5	Cerere mod ECP cu RLE	0011 0000
4	Cerere mod ECP fără RLE	0001 0000
3	(Rezervat)	0000 1000
2	Cerere ID dispozitiv	Returnează datele utilizând modul: Mod "Nibble" 0000 0100 Mod "Byte" 0000 0101 Mod ECP cu RLE 0011 0100 Mod ECP fără RLE 0001 0100
1	(Rezervat)	0000 0010
0	Mod "Byte"	0000 0001
-	Mod "Nibble"	0000 0000

Negocierea și identificatorii de dispozitiv sunt elemente importante pentru posibilitatea viitoare a sistemelor de a determina perifericele conectate la portul paralel.

2.2.4. Detectarea tipului de port paralel din calculator

2.2.4.1. Detectarea unui port ECP

Un port ECP se poate detecta în general prin prezența registrului ecr. Detectarea trebuie realizată atunci când portul este într-o stare inactivă. Se poate utiliza următoarea metodă [13]:

1. Se citește registrul *ecr* (cu adresa de bază + 402h) și se verifică dacă bitul *empty* = 1, iar bitul *full* = 0. Valoarea acestor biți trebuie să fie diferită de cea a biților de pe pozițiile corespunzătoare din registrul de control *dcr* (cu adresa de bază + 2). Pentru verificare se poate inversa valoarea unuia din cei doi biți ai registrului de control, urmărindu-se ca bitul corespunzător din registrul *ecr* să nu se schimbe.
2. În acest moment se cunoaște că portul este probabil de tip ECP, și se poate efectua un test suplimentar. Se înscrie valoarea 34h în registrul *ecr* și se citește conținutul aceluiasi registru. Deoarece biții 0 și 1 ai registrului *ecr* sunt de tip R/O, în cazul existenței portului ECP valoarea citită trebuie să fie 35h.

2.2.4.2. Detectarea unui port EPP

Pe lângă cele trei registre ale unui port SPP, portul EPP are un număr de registre suplimentare. Aceste registre permit testarea prezenței unui port EPP, similar cu modul de testare a prezenței unui port SPP. Deoarece prin înscrierea registrelor și citirea lor protocolul EPP nu este terminat, nu există o garanție a conținutului acestor registre după ce transferul este abandonat prin time-out. De aceea se recomandă ștergerea bitului de time-out (bitul 0 al registrului de stare, cu adresa de bază +1), după fiecare citire sau scriere. Metoda de ștergere a acestui bit diferează de la un controler la altul. La unele controlere, bitul poate fi șters prin înscrierea valorii 1. În alte cazuri, simpla citire a registrului de stare va șterge acest bit.

Nu se va testa prezența unui port SPP la adresa de bază 3BCh. Registrele suplimentare ale portului EPP nu sunt disponibile la această adresă, și pot fi utilizate de adaptorul video.

Observație.

În modurile avansate de transfer (EPP și toate modurile ECP, cu excepția modului SPP), pentru performanțe superioare, tipul ieșirilor registrului de control se poate modifica de la "open-collector" la "totem-pole". Aceasta înseamnă că portul nu se poate utiliza pentru intrare prin setarea la 1 a unui bit din registrul de control și citirea bitului respectiv. De asemenea, la anumite porturi ieșirile registrului de control sunt de tip "totem-pole" în toate modurile, astfel încât dacă se dorește compatibilitatea cu toate porturile, nu se vor utiliza biții registrului de control pentru intrare.

2.2.5. Transferuri de date în modurile avansate

Porturile paralele care pot lucra în modurile PS/2, EPP sau ECP dispun de linii de date bidirecționale, astfel că acestea se pot utiliza pentru transferuri mai rapide între două calculatoare. Portul paralel a fost însă conceput pentru legătura dintre un calculator și un periferic. În cazul calculatorului, liniile de date și cele de control ale portului reprezintă ieșiri, iar cele de stare reprezintă intrări. În cazul perifericului, liniile de date și de control reprezintă intrări, iar liniile de stare reprezintă ieșiri. Dacă se dorește conectarea a două calculatoare prin porturile paralele, este necesar un cablu special care conectează intrările unui port cu ieșirile celui de-al doilea port.

O problemă care apare este că starea implicită a liniilor de date ale ambelor calculatoare este cea de ieșire. Dacă se conectează liniile de date a două porturi, se conectează ieșirile împreună, iar curentul rezultat poate distruge circuitele porturilor. O posibilitate este de a nu se conecta cablul până în momentul în care unul din porturi este configurat ca intrare. Se poate utiliza un comutator electronic care va conecta liniile de date numai la cererea programului.

Pentru interconectarea a două calculatoare se poate utiliza un cablu cu lista de conexiuni din Tabelul 2.9.

Tabelul 2.9. Lista de conexiuni pentru transferul între două calculatoare.

Port 1	Pin	Pin	Port 2
-Strobe	1	10	-Ack
-Ack	10	1	-Strobe
Busy	11	14	-AutoFeed
-AutoFeed	14	11	Busy
Data[8:1]	2-9	2-9	Data[8:1]
-Init	16	12	PaperEnd
PaperEnd	12	16	-Init
-Error	15	17	-SelectIn
-SelectIn	17	15	-Error

2.3. Desfășurarea lucrării

2.3.1. Se vor descrie operațiile efectuate într-un ciclu de date urmat de un ciclu de comenzi, în cazul unui transfer în sens invers (Figura 2.6).

2.3.2. Se va răspunde la următoarele întrebări:

- a. Care sunt condițiile ca un sistem de operare, de exemplu *Windows 95*, să utilizeze transferuri în modul ECP cu o imprimantă conectată la portul paralel?
- b. Care din modurile EPP și ECP este mai avantajos cu periferice ca imprimante, scannere, unități de disc, conectate la portul paralel?
- c. Cum se poate emula un port standard SPP cu un port PS/2, EPP și ECP?
- d. Care este ordinea în care trebuie testată existența unui anumit tip de port în scopul determinării tipului portului paralel din calculator?
- e. Cum se poate determina dimensiunea memoriei FIFO a unui port ECP?

2.3.3. Se va scrie un program pentru detectarea porturilor ECP ale unui calculator. Se va utiliza metoda descrisă la 2.2.4.1.

2.3.4. Se va scrie un program pentru determinarea caracteristicilor unui port ECP. Se va trece portul în modul de configurare, după care se citesc registrele `cnfgA` și `cnfgB` și se efectuează următoarele operații:

- Se citește dimensiunea cuvântului;
- Se determină dacă portul are facilitatea de compresie a datelor. Se va încerca setarea bitului `compress` la 1, după care se va citi valoarea bitului. Dacă portul nu are facilitatea de compresie, va returna valoarea 0 pentru bitul `compress`, iar în caz contrar va returna valoarea 1.
- Se determină dacă există un conflict pe linia de întrerupere (valoarea 0 a bitului `intrValue` din registrul `cnfgB` indică existența unui conflict).
- Se determină dimensiunea memoriei FIFO.

Programul va afișa pe ecran caracteristicile portului.