

12. PORTUL SERIAL

12.1. CONSIDERAȚII GENERALE

Interfața serială RS 232 este simplă, universală ușor de înțeles și utilizat.

Portul serial permite conectarea calculatoarelor cu dispozitive terminale de date (Data Terminal Equipment) seriale (calculatoare, display, imprimante) sau cu echipamente de comunicație (Data Communication Equipment) seriale (modem).

Specificațiile electrice ale portului serial sunt conținute în standardul EIA (Electronics Industry Association) RS232C. În acest standard sunt definiți mai mulți parametri electrici, printre care:

- “Space” (0 logic) între +3 și +25 V
- “Mark” (1 logic) între -3 și -25 V.
- Regiunea între +3 și -3 V este nedefinită.
- Tensiunea unui circuit deschis nu trebuie să depășească 25V
- Curentul unui scurt circuit nu trebuie să depășească 500 mA
- Lungimea maximă a cablului nu trebuie să depășească 15m.

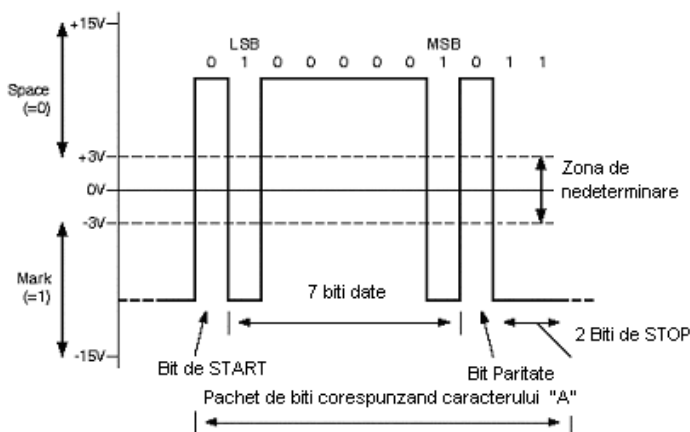


Fig.12.1 Semnalul pe linie la transmisia unui caracter

Semnalele și pinii asociați conectoarelor interfeței RS 232 sunt prezentate în tabelul de mai jos.

Semnal Pin (DB25/DB9)	Denumire	Funcție
TxD (2/3)	Transmit Data (Transmisie date)	Iesirea de date a portului serial (TXD)
RxD (3/2)	Receive Data (Recepție date)	Intrarea de date a portului serial (RXD)
CTS (5/8)	Clear to Send (Modem gata de emisie)	Aceasta linie indică faptul că modemul este pregătit pentru schimbul de date

DCD (8/1)	Data Carrier Detect (Detectare purtătoare de semnal)	Când modemul detectează o putătoare de date la celălalt capăt al liniei telefonice, acest semnal devine activ
DSR (6/6)	Data Set Ready (Modem operational)	Acest pin comunica circuitului UART faptul ca modemul este pregatit să stabileasca o legatura
DTR (20/4)	Data Terminal Ready (Terminal de date gata)	Se comunică modemului că circuitul UART este pregătit de comunicație
RTS (4/7)	Request To Send (Cerere de emisie)	Această linie informează modemul ca circ UART este pregatit să facă schimbul de date.
RI (22/9)	Ring Indicator (Indicator apel)	Devine activ când modemul detectează un semnal de apel de la PTSN.

Tabel 12.1 Funcțiile semnalelor interfeței seriale

Modul de conectare al dispozitivelor este prezentat în fig. 12.1.

12.2 CIRCUITELE UART

Elementul central al unei interfețe seriale este circuitul UART (Universal Asynchronous Receiver/Transmitter) sau USART (Universal Synchronous Asynchronous Receiver/Transmitter) având funcția de a converti octeții veniți de pe bus-ul paralel al PC-ului într-un șir de biți, câte unul o dată sau de a împacheta șirul de biți recepționați în octeți. Așa precum magistralele paralele au un debit de transfer în octeți/s debitul interfeței seriale, în biți/s (sau bauds) este mult mai redus. UART-ul are un set de debite (viteze) fixe pentru transmisia serială. La PC-uri se folosesc numai circuite UART.

Primele calculatoare personale au utilizat la implementarea interfeței seriale circuite UART 18250, care pe parcurs au fost înlocuite cu alte circuite mai performante. UART-urile folosite în diferitele generații de PC-uri se pot împărți în două categorii : *cele originale* (cu buffer de 1 octet) 8250, 16450 și primele versiuni ale 16550 și *UART-uri FIFO* : 16550A, 16650, 16750 și 16950, care sunt cele mai întâlnite tipuri.

- **8250** UART-ul original cu buffer de un octet
- **8250A** versiune a circuitului 8250 cu un registru suplimentar
- **16450** versiune îmbunătățită a UART-urilor anterioare, mai rapid
- **16550** UART cu 16-byte FIFO (dar FIFO nefuncțional)
- **16550A** UART cu aceleași caracteristici ca anteriorul, dar defectele eliminate
- **16650** UART cu 32-byte FIFO, debite până la 460.8 Kbps
- **16750** UART cu 64-byte FIFO, debite până la 921.6 Kbps
- **16950** UART cu 128-byte FIFO
- **Hayes ESP (Enhances Serial Ports)** - cartelă pe 8 biți cu două UART-uri 16550A și un procesor capabil de transfer DMA între portul serial și memorie.

Circuitul UART 16550 este compatibil cu 8250 și 16450 și reprezintă la ora actuală varianta standard de UART folosită în PC-uri. Apar doar două diferențe la pinii 24 și 29 față de I 8250. În cazul circuitului 8250 pinul 24 avea

funcția de "chip select out" și indica dacă chip-ul era activ sau nu, iar pinul 29 nu era conectat. La circuitul 16550 se introduc 2 pini noi în locul lor. Aceștia sunt TXRDY (Transmit Ready) și RXRDY (Receive Ready) care pot fi utilizați pentru transferuri cu acces direct la memorie. Pinii au două moduri diferite de funcționare. Modul 0 suportă un singur transfer pe DMA, în timp ce Modul 1 suportă Multi-transfer pe DMA.

Modul 0 este selectat atunci când buffer-ele FIFO sunt dezactivate prin bitul 0 al registrului de control FIFO sau atunci când buffer-ele FIFO sunt activate, dar DMA Mode Select este 0. În acest mod RXRDY este activ în starea "low" când cel puțin un caracter (Byte) este prezent în Buffer-ul de recepție. RXRDY va deveni inactiv în starea "high" atunci când nu mai rămâne nici un caracter în buffer-ul de recepție. TXRDY va deveni activ, "low", atunci când nu există caractere în Transmit Buffer. Va trece în starea inactivă, "high", după ce primul caracter (byte) va fi încărcat în buffer-ul de transmisie.

În modul 1 buffer-ele FIFO sunt active, iar DMA Mode Select este 1. În acest mod RXRDY va deveni activ, "low", atunci când este atins nivelul de basculare și va trece în starea inactivă, când nu se mai află caractere în FIFO. TXRDY va fi activ în momentul în care nu vor mai fi caractere în buffer-ul de transmisie și va deveni inactiv, atunci când bufferul este plin.

Semnalele pinilor circuitelor UART sunt compatibile TTL, inclusiv TxD, RxD, RI, DCD, CTS, DTR, și RTS care se găsesc în conectorul interfeței seriale. Trecerea de la nivel TTL la RS232 se face folosind circuite de linie sau convertoare de nivel : 1489 la recepție și 1488 la transmisie.

Importanța buffer-ului FIFO în creșterea performanțelor de prelucrare este prezentată în cele ce urmează. La variantele standard procesorul primea întreruperi de la UART la fiecare caracter recepționat de acesta, deoarece are un buffer doar de 1 octet. La debite reduse acest mod de tratare este convenabil, însă la debite mari timpul acordat de procesor UART-ului este prea mare ducând la scăderea performanțelor de prelucrare sau chiar la erori de depășire. Prin introducerea buffer-elor FIFO rata întreruperilor se reduce, la un prag programabil. De exemplu la 16550A, cu un FIFO de 16 octeți, declanșarea întreruperii se poate face la 14 (1/4/8) octeți la recepție și 16 la transmisie.

12.3 CIRCUITELE UART 8250/16550

12.3.1 SCHEMA BLOC. SEMNALE.

Elementele componente ale schemei bloc sunt prezentate în fig.12.1, astfel :

- Blocul de interfață și control cu sistemul realizează legătura între circuit și magistralele sistemului
- Blocul LSR și LCR conține registrele de Control Linie și Stare Linie
- Blocul Logic de Control Modem încorporează registrele de Control Modem Stare Modem și registrul Scratch Pad
- Blocul de Control Întreruperi generează semnalul de întrerupere spre procesor
- Blocul Control FIFO include registrul de control a stivei FIFO

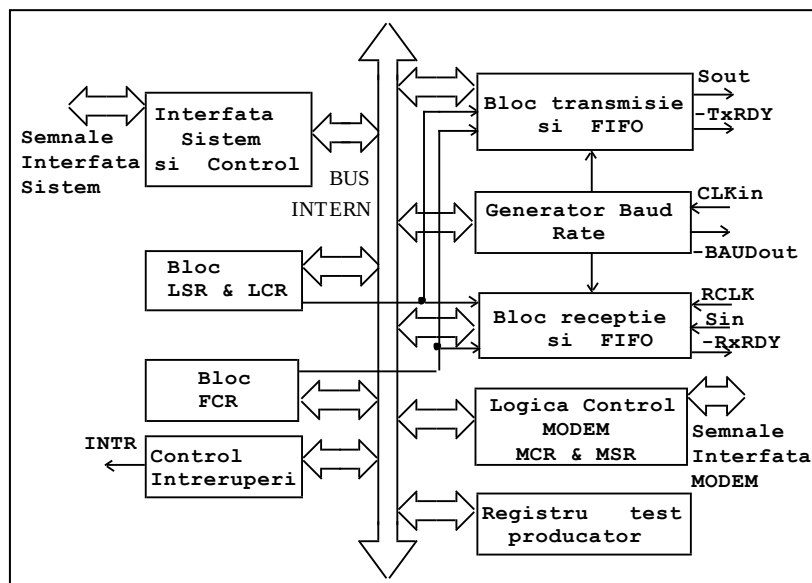


Fig.12.2 Schema bloc a circuitului UART 16550

- Blocul Generare Baud Rate conține registrele de divizare a frecvenței de ceas CLKIN
- Blocurile de Transmisie+FIFO și Recepție+FIFO conțin registrele de transmisie respectiv recepție date și stivele aferente
- Registrul testare producător permite testarea generatorului de baud rate și testarea erorilor de paritate și depășire.

În tabelul 12.2 se prezintă semnificațiile pinilor circuitului UART.

Pin	NUME	DESCRIERE
1-8	D0-D7	Magistrala de date
9	RCLK	Intrarea de ceas pentru Receiver
10	RD	Date receptionate
11	TD	Date transmise
12	CS0	Chip Select 0 - Activ pe 1 logic
13	CS1	Chip Select 1 - Activ pe 1 logic
14	-CS2	Chip Select 2 - Activ pe 0 logic
15	-BAUDOUT	Iesirea de simbol - Iesirea de la Generatorul de frecventa de simbol cu rata programabilă. Frecventa = (Baud Rate x 16)
16	XIN	Intrare externa pentru quart- Utilizată pentru oscilatorul Generatorului de frecventa de simbol.
17	XOUT	Iesire externa pentru quartz.
18	-WR	Linie de scriere - Inversoare
19	WR	Linie de scriere - Neinversoare
20	VSS	GND
21	RD	Linie de citire - Neinversoare
22	-RD	Linie de citire - Inversoare
23	DDIS	Dezactivare Driver. Acest pin devine zero când Unitatea Centrală citește de la circuitul UART. Poate fi conectat la emițătorul/receptorul de

		magistrală în cazul unei magistrale de date de capacitate mare.
24	-TXRDY	Transmisie terminată (8250-CSout)
25	-ADS	Strobare adresa. Utilizată dacă semnalele nu sunt stabile pe durata ciclului de citire sau scriere.
26-28	A2-A0	Biți de adresa pentru selecția regiștrilor
29	-RXRDY	Recepție terminată (8250 -Nefolosit)
30	INT	Iesire de întrerupere
31	-OUT2	Iesire de utilizator nr. 2
32	-RTS	Cerere de transmisie
33	-DTR	Terminal de date pregătit
34	-OUT1	Iesire de utilizator nr. 1
35	MR	Master Reset
36	-CTS	Modem gata de emisie
37	-DSR	Modem operational
38	-DCD	Dectecție de purtătoare
39	-RI	Indicator de apel
40	VDD	+ 5 V

Tabel 12.2. Funcțiile pinilor pentru circuitul UART 16550A

12.3.2 REGISTRELE CIRCUITELOR 8250/16550

Registrele circuitelor sunt accesibile prin instrucțiuni de I/O. Pentru primul adaptor serial, adresele porturilor sunt cuprinse între 3F8h și 3FFh, iar pentru al doilea între 2F8h și 2FFh. Primele două porturi permit accesul la mai mulți regiștri ai circuitului. Există și regiștri care nu sunt accesibili prin program utilizatorului. Tabelul următor prezintă registrele circuitului și adresele lor.

Adresa	R/W	Abr.	Numele Registrului
3F8h (2F8h)	Write	THR	Transmit Holding Buffer (Buffer de transmisie)
	Read	RBR	Receive Buffer (Buffer de recepție)
	Read/ Write	-	Divisor Latch Low Byte (Registru divizor, octetul „low”, dacă b7 al registrului de control al liniei este 1)
3F9h (2F9h)	Read/ Write	IER	Interrupt Enable Register (Registru de validare a întreruperilor)
	Read/ Write	-	Divisor Latch High Byte (Registru divizor, octetul „high”, dacă b7 al registrului de control al liniei este 1)
3FAh (2FAh)	Read	IIR	Interrupt Identification Register (Registru de identificare a întreruperilor)
	Write	FCR	FIFO Control Register (Registru de control FIFO)-numai la 16550
3FBh (2FBh)	Read/ Write	LCR	Line Control Register (Registru de control linie)
3FCh (2FCh)	Read/ Write	MCR	Modem Control Register (Registru de control modem)
3FDh (2FDh)	Read	LSR	Line Status Register (Registru de stare linie)
3FEh (2FEh)	Read	MSR	Modem Status Register (Registru de stare modem)
3FFh (2FEh)	Read/ Write	SPR	Scratchpad Register - numai la 16550

Tabel 12.3 Registrele circuitului 8250/16550

THR – Transmit Holding Register (R/W)

Este selectat dacă bitul 7 al registrului LCR este 0. Caracterul care trebuie transmis va fi înscris în acest registru. Dacă registrul THR se golește (deci poate fi înscris cu un nou caracter) circuitul generează o întrerupere, dacă este validată generarea întreruperilor. Starea acestui registru se poate determina prin testarea bitului 5 al registrului LSR.

	7	6	5	4	3	2	1	0
D	D6	D5	D4	D3	D2	D1	D0	

TSR – Transmit Shift Register

Este un registru intern care nu este accesibil prin program. La terminarea transmisiei unui caracter, conținutul registrului THR este transferat automat în registrul TSR și începe transmisia acestuia. Dacă registrul TSR se golește (deci ar putea începe transmisia unui nou caracter), iar registrul THR este gol, circuitul generează o întrerupere. Starea acestui registru se poate determina prin testarea bitului 6 al registrului LSR.

RBR – Receiver Buffer Register (R/W)

Este selectat dacă bitul 7 al registrului LCR este 0. Un caracter recepționat este depus în acest registru, de unde trebuie preluat de programul de recepție, înaintea terminării recepției unui nou caracter. În caz contrar apare o eroare de suprapunere. Recepția unui caracter determină generarea unei întreruperi. Starea acestui registru se poate determina prin testarea bitului 0 al registrului LSR.

	7	6	5	4	3	2	1	0
D	D6	D5	D4	D3	D2	D1	D0	

IER – Interrupt Enable Register (R/W)

Este selectat dacă bitul 7 al registrului LCR este 0. Circuitul poate genera patru tipuri de întreruperi, cu nivele de prioritate diferite. Registrul IER permite validarea independentă a generării acestor întreruperi.

	7	6	5	4	3	2	1	0
0	0	LPM*	SM*	MODEM STATUS CHANGE	RECEIVE LINE STATUS CHANGE	TRANSMIT HOLDING REG. EMPTY	RECEIVE DATA AVAILABLE	

Bit	Descriere
7	-
6	-
5	Validează Low Power Mode (16750)
4	Validează Sleep Mode (16750)
3	Generarea unei întreruperi la modificarea registrului de stare al modemului.
2	Validează generarea unei întreruperi la modificarea registrului de stare a liniei (LSR)
1	Validează generarea întreruperii la golirea registrului THR
0	Validează generarea întreruperii la recepția unui caracter

Divisor Latch Register LSB, MSB (R/W)

Cele două registre sunt accesibile prin porturile cu adresele 3F8h (2F8h), respectiv 3F9h (2F9h), dacă bitul 7 al registrului de control al liniei (LCR) este 1. Registrele conțin valoarea cu care trebuie divizată frecvența ceasului UART (1.8432 MHz) pentru a se obține debitul binar cerut. Pentru calculul divizorului trebuie să se țină cont de factorul de multiplicare a ceasului circuitului UART, de obicei 16. Se poate utiliza formula :

$$\text{Divizor} = 1843200 / (\text{DebitBinar} * 16)$$

Debit binar (bps)	Divizor	Debit binar (bps)	Divizor
150	300h	4800	18h
300	180h	7200	10h
600	C0h	9600	0Ch
1200	60h	19200	06h
2400	30h	38400	03h
3600	20h	115200	01h

Tabel 12.4. Divizorii frecvenței de 1.8432 MHz pentru diferite debite binare

IIR – Interrupt Identification Register (R)

Deși întreruperile generate de circuitul UART apar pe un singur nivel (IRQ4 pentru primul adaptor și IRQ3 pentru adaptorul al doilea), aceste întreruperi pot avea diferite cauze cu prioritățile LSI>DAI>TREI>MSI. Identificarea cauzei întreruperii se poate realiza prin testarea biților 3–0 ai registrului IIR. Ceilalți biți ai registrului nu sunt utilizați la circuitul 8250. La 16550 dacă b7=b6=1 este validat buffer-ul FIFO.

7	6	5	4	3	2	1	0
FIFO ENABLED (16550)	FIFO ENABLED (16550)	0	0	INTERRUPT ID 2 (16550)	INTERRUPT ID 1	INTERRUPT ID 0	PEND

Biți 3–0	Tip întrerupere	Cauza întreruperii	Resetarea întreruperii
xxx1	-	Nu exista întrerupere	-
0000	Modificare stare modem	Modificare CTS, DSR, RI, DCD	Citire registru de stare modem (MSR)
0010	Terminare transmisie caracter	Registru THR gol	Citire IIR sau scrierea unui caracter în THR
0100	Receptie caracter	Registru RBR conține un caracter recepționat	Citire registru RBR
0110	Modificare stare linie	Eroare de sprapunere, de incadrare sau de paritate sau de transmisie spatii (BREAK)	Citire registru de stare linie (LSR)

Tabel 12.5 Biții de identificare a întreruperilor din registrul IIR

FCR - FIFO Control Register (W)

Registru FCR este un registru care poate fi doar înscris. Acest registru este folosit pentru a controla buffer-ele FIFO care se găsesc începând cu circuitele UART 16550 și următoarele.

7	6	5	4	3	2	1	0
TRIGGER	TRIGGER	0	0	DMA	TXFIFO	RXFIFO	ENABLE
1	0			MODE	RESET	RESET	

Bit	Descriere		
7-6	B7	B6	Nivelul pragului de intrerupere la
	0	0	1 Byte
	0	1	4 Bytes
	1	0	8 Bytes
	1	1	14 Bytes
5	Activeaza FIFO pe 64 de octeti (doar la 16750)		
4	Rezervat		
3	Selecteaza modul DMA. Schimabă starea pinilor RXRDY și TXRDY din modul 1 în modul 2.		
2	OBS. Arhitectura PC nu acceptă acest mod.		
1	Șterge stiva FIFO de transmisie		
1	Șterge stiva FIFO de receptie		
0	Activează FIFO		

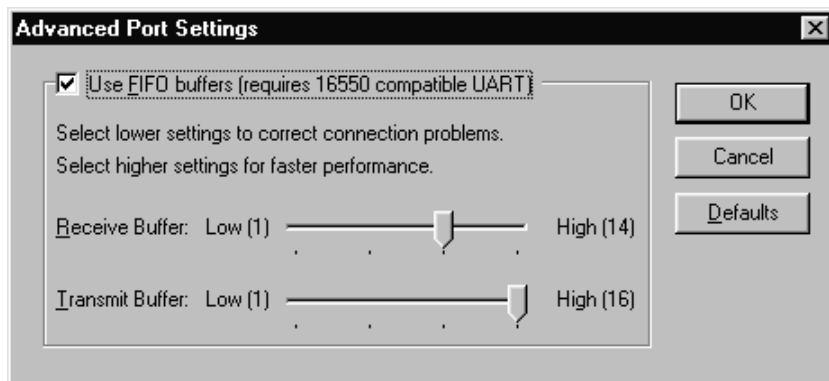


Fig.12.3 Setarea dimensiunilor buffer-elor FIFO

LCR – Line Control Register (R/W)

Prin inscrierea registrului LCR se pot stabili parametrii transmisiei.

7	6	5	4	3	2	1	0
DIVISOR	BREAK	STICK	EVEN	PARITY	STOP	WORD	WORD
LATCH			PARITY	ENABLE	BIT	LENGTH	LENGTH
ACCESS BIT			SELECT		SELECT	SELECT 1	SELECT 0

Bit	Descriere			
7	1	Bitul de acces la regiștrii divizorului		
	0	Acces la bufferul de receptie,transmisie și la registrul IER		
6	Validare Break			
5-3	B5	B4	B3	Selectarea paritatii
	X	X	0	fără Paritate
	0	0	1	Paritate impară
	0	1	1	Paritate Pară
	1	0	1	Paritate high (Sticky)
	1	1	1	Paritate low (Sticky)
2	Lungimea "bitului" de stop			
	0	1 bit de stop		
		2 biti de stop pentru caractere de		
	1	lungime 6/7/8 biți sau 1.5 biți de stop pentru caractere de 5 biți		
1-0	B1	B0		Lungimea caracterului
	0	0		5 Biti
	0	1		6 Biti
	1	0		7 Biti
	1	1		8 Biti

- Bitul 5 (Stick Parity) permite transmiterea sau așteptarea unor biți de paritate cu valoare fixă
- Bitul 6 (Set Break) dacă este 1, circuitul forțează linia la nivelul "0" logic atunci când nu exista caractere de transmis (linia este în starea "break")
- Bitul 7 (Divisor Latch Access Bit) modifică rolul registrelor accesibile prin adresele 3F8h (2F8h) și 3F9h (2F9h). Dacă este "0" aceste registre sunt Transmitter/Receiver Buffer Register, respectiv Interrupt Enable Register, iar dacă este 1 registrele sunt cele utilizate pentru înscrierea divizorului care stabilește viteza de transmisie.

MCR – Modem Control Register (R/W)

Se utilizează pentru comanda comunicației cu modemul.

7	6	5	4	3	2	1	0
0	0	0	LOOPBACK TEST	OUT1	OUT2	REQUEST TO SEND	DATA TERMINAL READY

Bit	Descriere
7	Rezervat
6	Rezervat
5	Rezervat
4	Mod LoopBack (DTR, RTS, RI și DCD deconectate)
3	Iesire auxiliara 2
2	Iesire auxiliara 1
1	Cerere de transmisie
0	Terminal de Date Ready

LSR – Line Status Register (R)

Indică starea liniei de comunicație. Bitii 0–4 se referă la receptie, iar biții 6–7 se referă la transmisie

7	6	5	4	3	2	1	0
RXFIFO HOLDS ERROR (16550)	TRANSMIT EMPTY	TRANSMIT HOLDING REG. EMPTY	BREAK DETECTED	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	RECEIVE DATA AVAILABLE

- Bitul 0 (Data Ready) este setat dacă în registrul RBR se afla un caracter recepționat. Bitul este șters automat în urma citirii caracterului.
- Bitul 1 (Overrun Error) indică o eroare de suprapunere. Este setat dacă se recepționează un caracter înaintea citirii caracterului din registrul RBR de către unitatea centrală în acest caz se pierd unul sau mai multe caractere. Bitul este resetat prin citirea registrului LSR.
- Bitul 2 (Parity Error) este setat dacă se recepționează un caracter cu paritatea diferită de cea așteptată. Bitul este resetat prin citirea registrului LSR.
- Bitul 3 (Framing Error) este setat dacă se recepționează un caracter fără biții de stop corespunzători. La detectarea acestei erori circuitul încearcă să se resincronizeze. Bitul este resetat prin citirea registrului LSR.
- Bitul 4 (Break Interrupt) este setat dacă este sesizat nivelul 0 (spatiu) pe linie pentru o perioadă mai mare decât este necesară pentru transmiterea unui caracter. Bitul este resetat prin citirea registrului LSR.
- Bitul 5 (Empty Transmitter Holding Register) este setat atunci când conținutul registrului THR este depus în registrul TSR și începe transmiterea caracterului. Bitul este resetat la înscrierea unui caracter în registrul THR.
- Bitul 6 (Transmitter Shift Register Empty) este setat dacă atât registrul THR cât și TSR s-au golit.
- Bitul 7 indică o eroare la un octet din bufferul FIFO de recepție

MSR – Modem Status Register (R)

Conține informații despre starea modemului.

7	6	5	4	3	2	1	0
DATA CARRIER DETECT	DATA SET READY	RING INDICATE	CLEAR TO SEND	DELTA DCD	DELTA DSR	DELTA RI	DELTA CTS

- Bitii 0–3 indică modificarea stării semnalelor CTS, DSR, RI și CD de la ultima citire a registrului MSR. Acești biți sunt reșetați la citirea registrului MSR.
- Bitii 4 –7 indică starea curentă a semnalelor CTS, DSR, RI și CD.

Registrul Scratchpad (16550) (R/W)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Acest registru nu se utilizează la comunicare, ci mai degrabă pentru a păstra un octet de date. Utilitatea lui reală este de a determina dacă tipul UART-ului este 8250/8250B sau 8250A/16450 (deși 8250/8250B nu a fost proiectat pentru a fi utilizat în PC-AT).

12.4 ADRESE ȘI ÎNTRERUPERI REZERVATE PORTULUI SERIAL

Nume	Adresa de bază	Întrerupere	Adr. COM în zona BIOS
COM 1	3F8	4	0000:0400
COM 2	2F8	3	0000:0402
COM 3	3E8	4	0000:0404
COM 4	2E8	3	0000:0406

Tabel 12.6 Adresele de bază standard ale portului serial

În tabelul 12.6 sunt prezentate adresele standard de port întâlnite în majoritatea calculatoarelor personale. La fel ca și pentru portul paralel de cuplare a imprimantei (LPT), adresele de baza pentru porturile COM pot fi citite din regiunea de date din BIOS. Fiecare adresă ocupă 2 octeți. Programul în C de mai jos se arată cum se pot citi aceste locații pentru a obține adresele porturilor COM.

```
#include <stdio.h>
#include <dos.h>

void main(void)
{
    unsigned int far *indadr; /* Indicator catre locatia
    adresei portului */
    unsigned int adr_port;    /* Adresa portului */
    int i;
    indadr=(unsigned int far *)0x00000400;
    for (i = 0; i < 4; i++)
    {
        adr_port = *indadr;
        if (adr_port == 0)
            printf("Nu s-a gasit nici un port pentru COM%d \n",i+1);
        else
            printf("Adresa alocata pentru COM%d is %Xh\n",i+1,adr_port);
        *indadr++;
    }
}
```

12.5 FUNCȚII BIOS PENTRU PORTUL SERIAL

Funcțiile BIOS pentru comunicația serială asincronă sunt accesibile prin întreruperea **14h**, iar pentru controlul prin întreruperi al interfeței seriale este necesară scrierea unor rutine proprii, deoarece funcțiile BIOS permit doar transmisia și recepția prin testarea stării.

În general, funcțiile returnează starea liniei în registrul AH și cea a modemului în registrul AL. Semnificația bitilor celor doi octeți de stare este următoarea:

AH (Stare linie)

Bit	Descriere
7	(Time-Out Error): s-a depășit timpul de așteptare pentru transmisie sau receptie
6	(Transmitter Shift Register Empty): circuitul a terminat de transmis toate caracterele care au fost înscrise în registrul de transmisii;.
5	(Transmitter Holding Register Empty): registrul de transmisie este gol;
4	(Break Detect): linia este la nivelul "0" pentru un timp mai lung decât cel necesar transmiterii unui caracter;
3	(Framing Error): eroare de cadrare (nu s-a recepționat nici un bit de stop);
2	(Parity Error): eroare de paritate;
1	(Overrun Error): eroare de suprapunere
0	(Data Ready): există un caracter recepționat în bufferul de receptie

AL (Stare modem)

Bit	Descriere
7	(Carrier Detect): starea semnalului CD
6	(Ring Indicator): starea semnalului RI;
5	(Data Set Ready): starea semnalului DSR;
4	(Clear To Send) : starea semnalului CTS;
3	Delta Carrier Detect): starea purtătoarei de date s-a modificat
2	(Trailing Edge Ring Indicator):semnalul RI a ajuns la nivel 1;
1	(Delta Data Set Ready): starea semnalului DSR s-a modificat
0	(Delta Clear To Send): starea semnalului CTS s-a modificat

Funcțiile BIOS apelabile cu INT 14h :

Funcția	Operația	Intrare	Ieșire
AH=00	Inițializare	AL = parametrii comunicației DX = nr. port serial (0-3)	AH = stare linie AL = stare modem
AH=01	Transmite caracter	AL = caracter DX = nr. port serial (0-3)	AH: b7=0 dacă e corect b7=1 dacă e eroare AH b6..0 = stare, dacă e corect AL=caracter
AH=02	Recepție caracter	DX = nr. port serial (0-3)	AH: b7=0 dacă e corect b7=1 dacă e eroare AH b6..0 = stare, dacă e corect AL=caracter citit
AH=03	Citire port stare	DX = nr. port serial (0-3)	AH = stare linie AL = stare modem

- Biții 1-0 : lungimea caracterului (10 - 7 biți; 11 - 8 biți)
- Bit 2 : numărul biților de stop (0- 1 bit de stop ; 1 - 2 biți de stop)
- Biți 4-3 : controlul parității (01- p. impară, 11 – p. pară, 0x fără paritate)
- Biți 7-5 : viteza de transmisie (000 -110 bps,111 -9600 bps)

12.6. APLICAȚII ȘI TEME

Aplicația următoare propune conectarea a două PC-uri prin intermediul portului serial.. Caracterele introduse la tastatura unui PC se transmit pe interfața serială și sunt recepționate (folosind întreruperile generate la recepția unui caracter) și afișate la celălalt PC.

Conectarea între PC-uri se face conform fig. 12.4, care reprezintă conexiunile unui cablu “Null Modem” utilizat la conectarea a două dispozitive DTE.

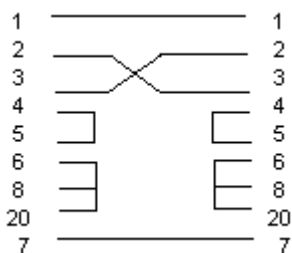


Fig.12.4 Conectarea PC-urilor

a. Analizați programul în limbaj de asamblare și executați-l.

```
;Programul trimite caracterele tastate pe interfata seriala si
;receptioneaza si afiseaza caractere de pe interfata seriala COM1
;PC-urile se conecteaza astfel: 2-3;3-2;7-7;
;la receptia unui caracter se genereaza intrerupere pe IRQ4
;-----
STIVA    SEGMENT PARA STACK 'STACK'
        DW      256 DUP (?)
STIVA    ENDS
DATA     SEGMENT PARA PUBLIC 'DATA'
MESAJ    DB 'Program de comunicatie seriala prin intreruperi',0dh,0ah
        DB      'PENTRU TERMINARE APASATI TASTA ESC !',0dh,0ah,'$'
IRQ4_old DW 2 DUP(?)      ;rezervat ptr. salvarea adresei rutinei
                        ;IRQ4 sistem
DATA     ENDS

CODE     SEGMENT PARA PUBLIC 'CODE'
        ASSUME  CS:CODE,DS:DATA,SS:STIVA,ES:NOTHING
        COM1    equ 3F8h

START:   PUSH    DS
        XOR     AX,AX                ;initializare DS
        PUSH    AX
        MOV     AX,DATA
        MOV     DS,AX

        MOV     AX,0600h             ;sterg ecranul
        MOV     DX,0
        MOV     CX,164Fh
        MOV     BH,7
        INT     10h

        MOV     DX,OFFSET MESAJ     ;afisare mesaj pe ecran
        MOV     AH,9
        INT     21h
```

PORTUL SERIAL

```

MOV     DX,COM1+3           ;initializare rata baud
MOV     AL,80H              ;selectie reg.divizori,b7=1
OUT     DX,AL               ;constanta de divizare
MOV     DX,COM1             ;0Ch, pentru 9600 baud
MOV     AL,0cH
OUT     DX,AL
MOV     DX,COM1+1
MOV     AL,0
OUT     DX,AL

MOV     DX,COM1+3           ;initializare control linie
MOV     AL,00010011B        ;8 biti/caracter,1 bit STOP,
OUT     DX,AL               ;fara paritate,B7=0 acces THR,RDR,IER

MOV     DX,COM1+4           ;initializare control modem
MOV     AL,00001011B        ;Activat DTR,RTS,OUT2
OUT     DX,AL

CLI
MOV     AX,350Ch            ;modifica adresa intreruperii IRQ4
INT     21h                 ;(tip 0ch) (COM1)
                                ;citesc adr.rutinei COM1 in ES:BX
                                ;apelez functia DOS 35h

MOV     IRQ4_old,BX
MOV     IRQ4_old+2,ES
PUSH    DS
MOV     DX,OFFSET INT_COM1
MOV     AX,seg INT_COM1
MOV     DS,AX
MOV     AX,250ch
INT     21h
STI

MOV     DX,COM1+1          ;validez intreruperea la buffer date plin
MOV     AL,1
OUT     DX,AL

CLI                                     ;demascheaza IRQ4
MOV     AL,0A0H
OUT     21H,AL
STI
WAIT:   MOV     AH,0
INT     16H                  ;citeste caracter de la tastatura
CMP     AL,1BH               ;testeaza daca s-a apasat ESC
JZ      GATA
MOV     DX,COM1              ;trimite caracterul pe serial
OUT     DX,AL
JMP     WAIT

GATA:   MOV     AL,0B0H       ;remascheaza IRQ4
OUT     21H,AL
CLI
PUSH    DS
MOV     AX,250ch
LDS     DX,dword ptr IRQ4_old
INT     21h
POP     DS
STI
MOV     AX,4C00H              ;iesire din program
INT     21h

INT_COM1 PROC FAR            ;intreruperea de IRQ4
MOV     DX,COM1

```

```

        IN     AL,DX           ;citeste caracter de la serial
        PUSH  AX
        MOV   AH,0EH
        MOV   BX,0
        INT   10H             ;afiseaza pe ecran caracter
        POP   AX
        CMP   AL,0DH           ;daca s-a tastat CR se mai adauga LF
        JNZ   EOI
        MOV   AL,0AH
        MOV   BX,0
        MOV   AH,0EH
        INT   10H
EOI:
        MOV   AL,20H           ;trimite EOI
        OUT   20h,AL
        IRET                   ;sfirsitul intreruperii
INT_COM1  ENDP

CODE      ENDS
END       START               ;sfirsit program

```

- b. Analizați și executați aplicația C care transmite și recepționează caracterele tastate pe interfața COM1. Recepția caracterelor se face prin program (pooling).

```

/*Programul trimite caracterele de la tastatura pe COM1 si le
recepționeaza tot pe COM1; se leaga pinii 2-3 la conectorul COM1*/
#define COM1_PORT 0x3F8
#define TXD COM1_PORT
#define LSR (COM1_PORT+5) //Registrul Starea Linie 0x3FD
#define LCR (COM1_PORT+3) //Registrul Control Linie 0x3FB
#define ESC 0x1B
#include <conio.h>
#include <stdio.h>
#include <dos.h>
/*Programul trimite si recepționeaza caracter pe COM1
TxD conectat la Rx D */

void init_com1 (void);
void txd_com1 (char);
char rxd_com1 (void);

int main (void)
{
    char car_in, car_out;

    init_com1();
    printf("\nIntroduceti caracter de la tastatura (ESC gata) : \n");
    do
    {
        printf("Transmis :");
        car_out=getche();
        txd_com1(car_out);
        car_in=rxd_com1();
        printf("\tReceptionat:%c \n",car_in);
    }
    while (car_out!=ESC);
    return(0);
}

```

```

}

void init_com1(void)
{
    outportb(LCR,0x80); //B7=1 setez bitul
                        //ptr.acces la reg.divizori
    outportb(TXD,0x0c); //Incarc reg. divozori ptr.
    outportb(TXD+1,0); //9600bps
    outportb(LCR,3);   //B7=0-acces TDR,RDR;8biti/car,
                        //2biti Stop, fara paritate
}

void txd_com1(char car)
{
    char stare;
    do
    {
        stare=inportb(LSR)&0x40;
    } while (stare!=0x40);
    outport(TXD, (char)car);
}

char rxd_com1(void)
{
    char stare;
    do
    {
        stare=inportb(LSR)&0x01;
    } while (stare!=0x01);
    return(inportb(TXD));
}

```

- c. Modificați aplicația C astfel ca recepția caracterelor să se facă prin întreruperi.
- d. Determinați tipul UART-ului din sistemul utilizat, prin program.

Obs. Determinarea tipului de UART folosit se poate face testând micile diferențe ce există între diferitele generații. Astfel , pentru a testa dacă UART-ul are sau nu FIFO se pune b0 din FCR pe "1", adică se validează buffer-ele FIFO. Apoi se citesc biții 6 și 7 din IIR și vor fi ambii "1" dacă dacă buffer-ele au fost validate, deci circuitul este 16550A sau următoarele. Dacă buffer-ele nu au fost validate cel mai probabil UART-ul este 16450, dar poate fi și 8250A/B în sistemele vechi. Deosebirea între 16450, 8250A și 8250, 8250B se poate face prin înscrierea și citirea registrului scratchpad care este implementat la primele două circuite. Pentru a verifica dacă circuitul nu este cumva 16750, se validează buffer-ul FIFO de 64 de octeți (b5=1 din FCR) și se testează bitul 5 din IIR pentru a verifica validarea sau nu a buffer-ului FIFO.