

„Nu e de părerea ta cel ce te aprobă, ci cel ce te imită.” G. Moasil

C7. PORTUL SERIAL – COM:

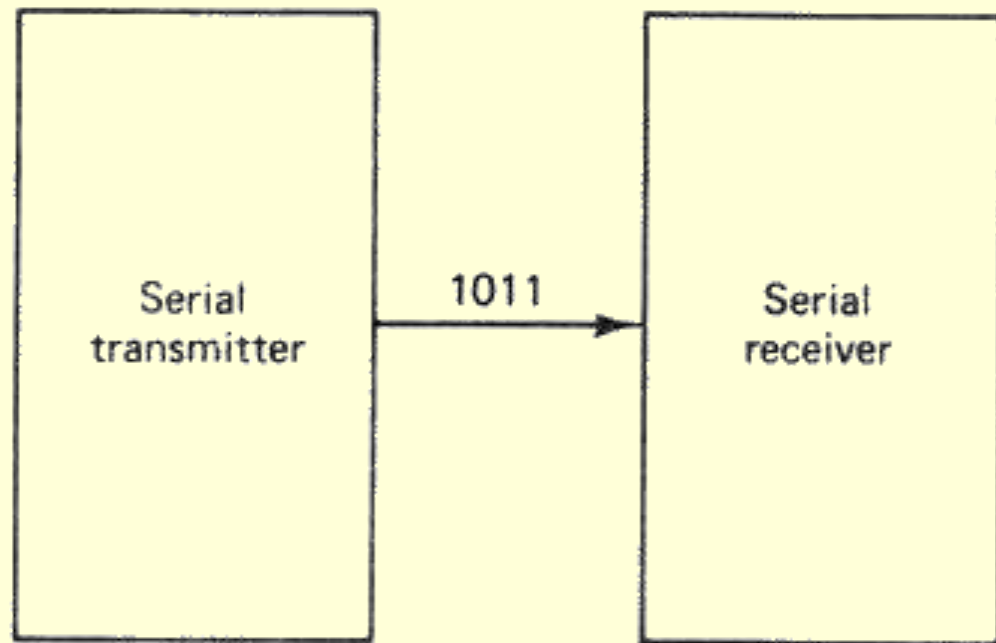
INTRODUCERE – COMUNICATII SERIALE

- 1. SEMNALE INTERFETEI RS232**
- 2. EVOLUTIA UART-URILOR**
- 3. ARHITECTURA. PINI**
- 4. PROGRAMARE. REGISTRE**
- 5. FUNCTII BIOS PENTRU PORTUL SERIAL (INT 14h)**
- 6. APLICATII**

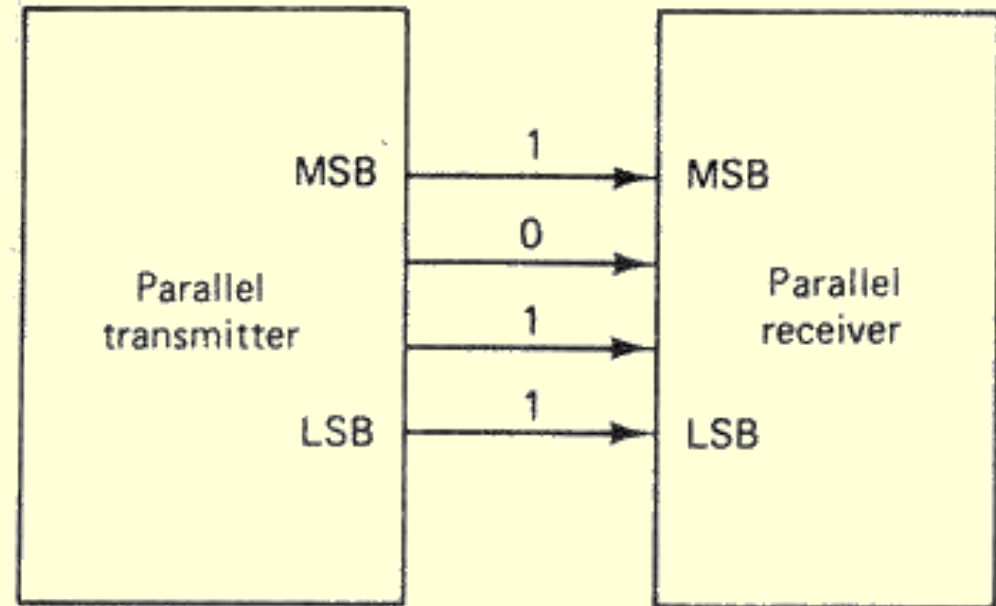
<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/uart.html>

Parallel vs. Serial

- 4 clock / 4 biti transfer



- 1 clock / 4 biti transfer



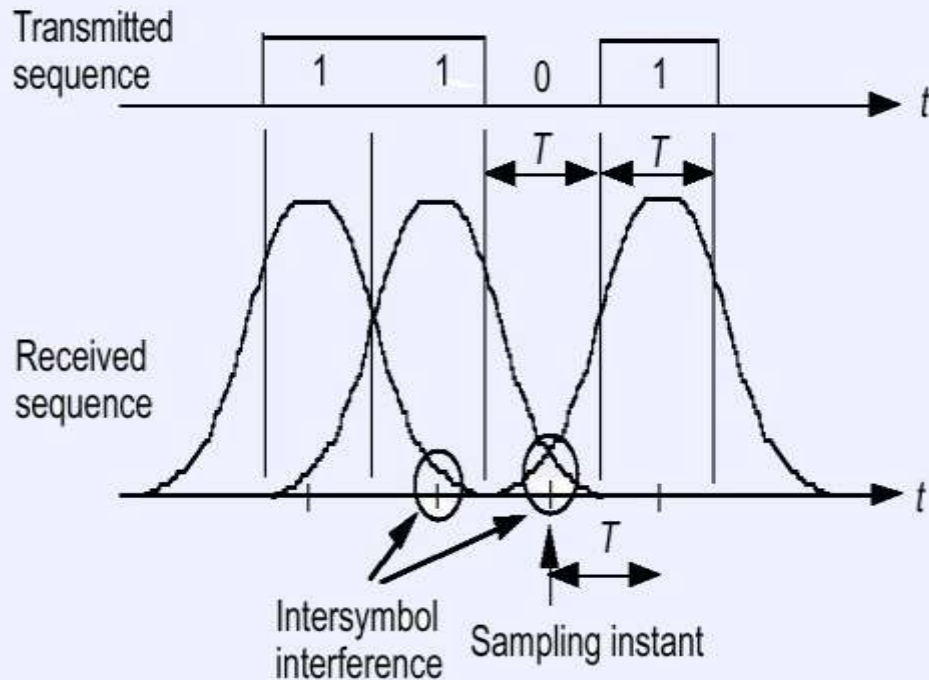
- La prima vedere, s-ar părea că porturile paralele ar trebui să poată să trimită date mult mai rapid decât porturi seriale.

- Transmisia în serie este mai lentă decât transmisia paralelă, la aceeași frecvență a semnalului.
- Cu o transmisie paralelă puteți transfera un cuvânt pe ciclu (ex. 1 octet = 8 biți), dar cu o transmisie în serie doar o fracțiune din acesta (de exemplu, 1 bit).

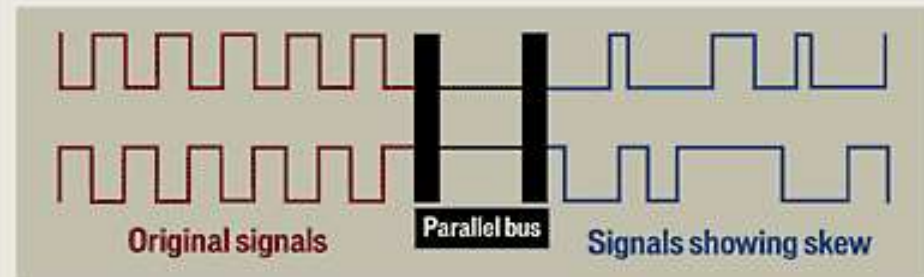
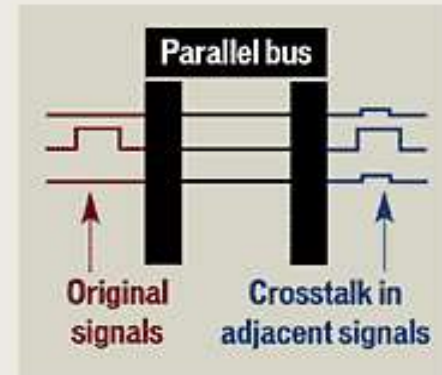
Motivul pentru care dispozitivele moderne folosesc transmisia în serie este următorul:

- Nu putem crește frecvența semnalului pentru o transmisie paralelă fără limită, deoarece, prin proiectare, toate semnalele de la transmițător trebuie să ajungă la receptor în același timp. Acest lucru nu poate fi garantat pentru frecvențele înalte, deoarece nu puteți garanta că **timpul de tranziție** al semnalului este egal pentru toate liniile de semnal (sunt diferite trasee pe placa de bază).
- Cu cât frecvența este mai mare, cu atât diferențele mici de timp contează mai mult. Prin urmare, receptorul trebuie să aștepte până când toate liniile de semnal sunt stabilizate -- evident, așteptarea scade rata de transfer.
- Un alt aspect este că trebuie să luați în considerare **diafonia** la liniile de semnal paralele. Cu cât frecvența este mai mare, cu atât diafonia devine mai pronunțată și, odată cu aceasta, cu atât este mai mare probabilitatea unui semnal corupt și nevoia de a-l retransmite.
- Deci, chiar dacă transferam mai puține date pe ciclu cu o transmisie în serie, putem merge la frecvențe mult mai mari, ceea ce duce la o rată de transfer netă mai mare.

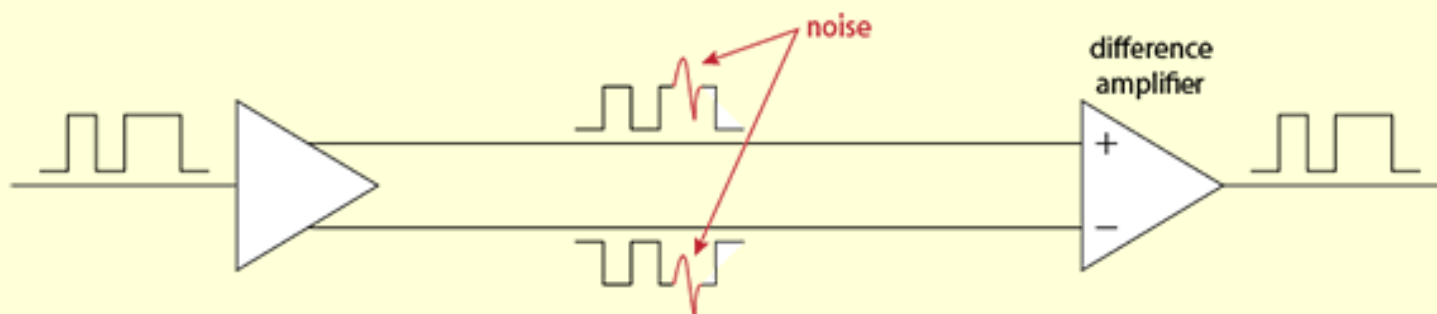
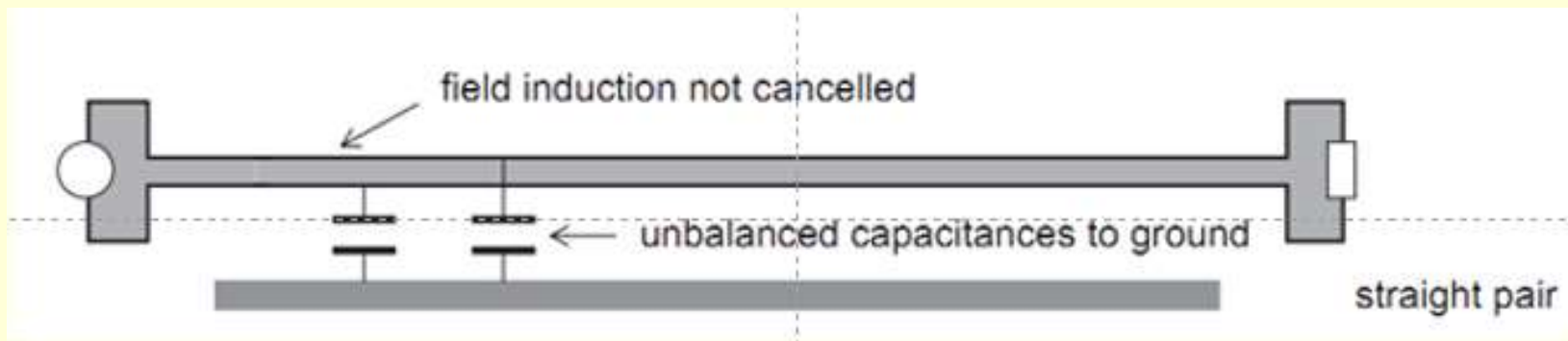
- Conexiunile paralele sufera si datorita *interferenței inter-simbol (ISI)*, distorsiunii (*skew*) și *diafoniei* (zgomotului), și, prin urmare, datele pot fi corupte pe distanțe mari.
- O lățime de bandă mare duce la un debit binar mai bun și mai puțin zgomot în canal, înseamnă că pot transmite cu succes date, sigur cu un raport semnal-zgomot (SNR) mai scazut.



Crosstalk (right) occurs when the signal on one wire in a parallel bundle imprints itself on an adjacent wire. Skew (below) is the result of random imperfections in the wires and connections of the parallel bundle.

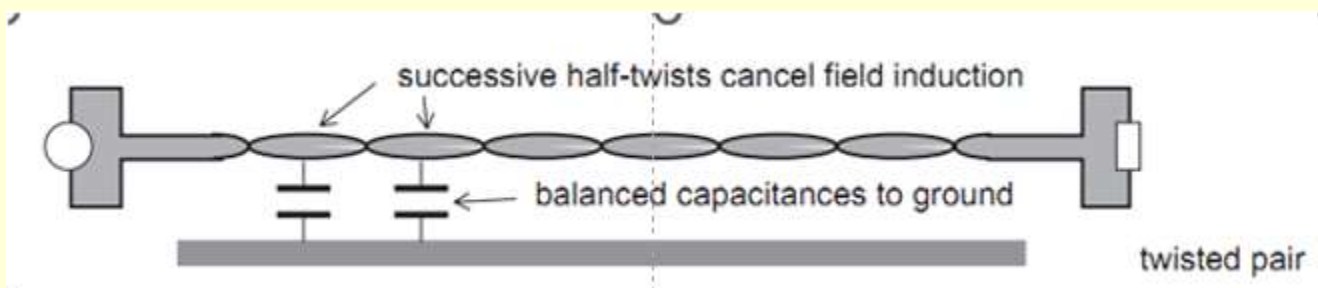


- De asemenea, pentru că firele într-un sistem paralel au mici capacitati și inductanțe mutuale, *lățimea de bandă* la fire/conexiuni paralele este mult mai mica decât lățimea de bandă la firele serie.



- Cand datele sunt trimise pe o linie diferențială, înseamnă că D- este o imagine oglindă a lui D +, astfel încât ambele linii de date poartă semnalul. Receptorul scade D- din D +. Dacă un semnal de zgomot ar fi preluat de ambele fire, scăderea îl va anula.

- Deci, semnalizarea diferențială ajută la *suprimarea zgomotului*, la fel și tipul de cablare, și anume perechea torsadată. Dacă firele ar funcționa doar paralel, acestea ar forma o buclă (îngustă) care ar putea detecta interferențe magnetice. Dar, datorită răsucirilor, orientarea firelor față de câmp se schimbă continuu. Un curent indus va fi anulat de un curent cu semnul opus o jumătate de buclă mai departe.

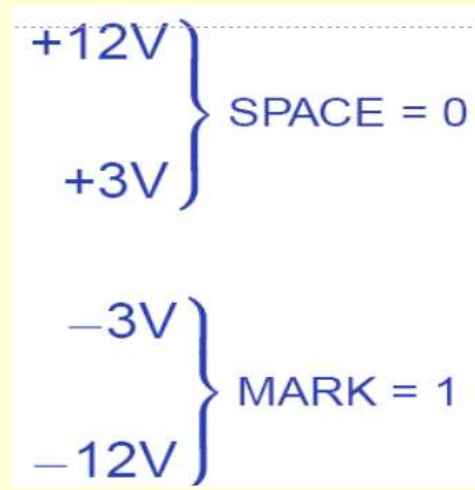
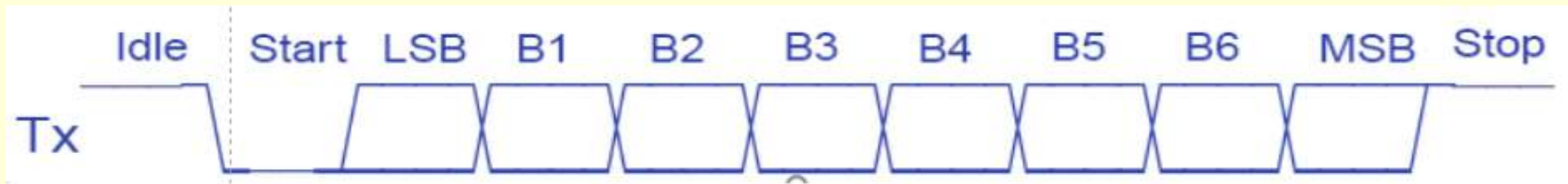


- Să presupunem că avem o perturbatie care lucrează vertical pe firul răsucit. Putem privi fiecare jumătate de răsucire ca o buclă mică care preia perturbatia. Apoi, este ușor de văzut că următoarea buclă mică vede câmpul opus (cu susul în jos), astfel încât anulează primul câmp. Acest lucru se întâmplă pentru fiecare pereche de jumătăți de răsucire.
- Un efect similar de echilibrare are loc pentru capacitatea fata de masa. La o pereche dreaptă de fire, un conductor arată o capacitate mai mare față de masa decât celălalt, în timp ce la o pereche răsucită fiecare fir va arăta aceeași capacitate.

Interface	Format	Number of Devices (maximum)	Distance (maximum, ft)	Speed (maximum, bps)	Typical Use
<u>RS-232 (TIA-232)</u>	asynchronous serial	2	50-100	20k (faster with some hardware)	Modem, basic communications
RS-485 (TIA-485)	asynchronous serial	32 unit loads (up to 256 devices with some hardware)	4000	10M	Data acquisition and control systems
Ethernet	serial	1024	1600	10G	PC network communications
IEEE-1394b (FireWire 800)	serial	64	300	3.2G	Video, mass storage
IEEE-488 (GPIB)	<u>parallel</u>	15	60	8M	Instrumentation
I ² C	synchronous serial	40	18	3.4M	Microcontroller communications
Microwire	synchronous serial	8	10	2M	Microcontroller communications
MIDI	serial current loop	2 (more with flow-through mode)	50	31.5k	Music, show control
Parallel Printer Port	<u>parallel</u>	2 (8 with daisy-chain support)	10–30	8M	Printer
SPI	synchronous serial	8	10	2.1M	Microcontroller communications
USB	asynchronous serial	127	16 (up to 98 ft with 5 hubs)	1.5M, 12M, 480M	PC peripherals

Interfata RS 232

- Standardul de transmisie seriala a fost definit la inceputul anilor '60 ca fiind o conexiune seriala, asincrona, full-duplex, p2p, bazata pe nivele de tensiune, la distante $>30m$



- **Moment elementar (T_s)** – timpul in care caracteristicile semnalului transmis nu variaza (poate fi 1bit/combinatii de biti) - *perioada simbol*

- **Viteza modulatiei (V_m)** – nr. momente elementare/ sec.

$$V_m = 1/T_s \text{ (baud)}$$

- **Debit binar (D)** – nr. de simboluri elementare/sec

$$D=1/T \text{ (bps)} ; T=\text{durata bit}$$

- prin artificii se pot caracteriza m.m. biti fixand m.m. caracteristici si m.m. nivele ptr. caracteristicile semnalului, astfel:

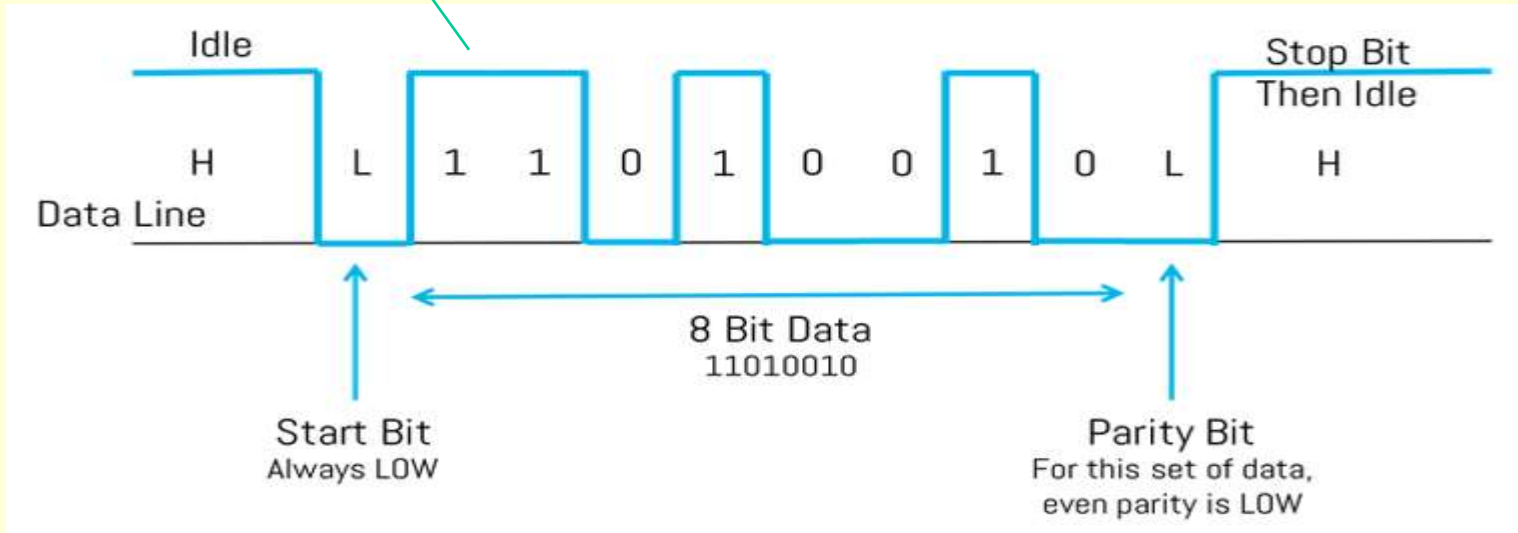
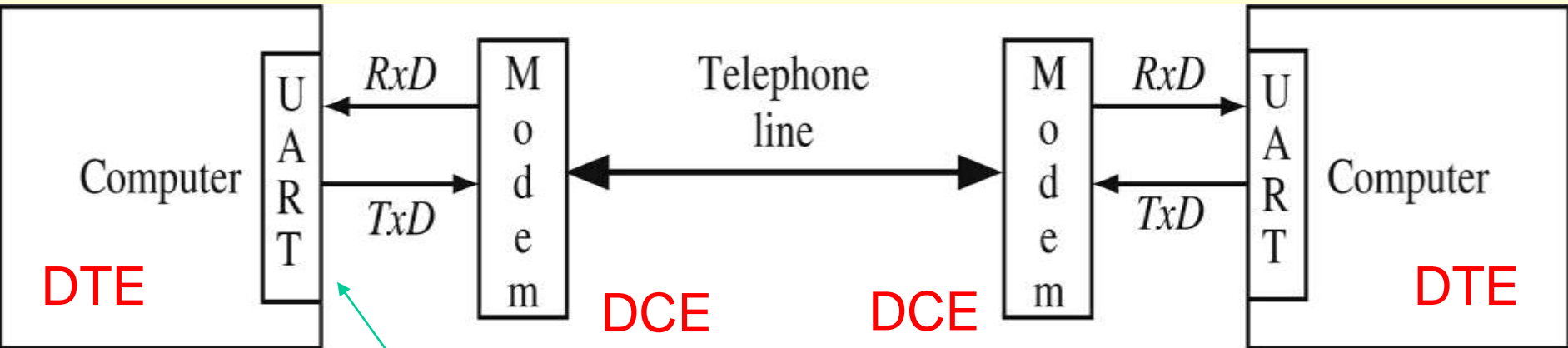
-daca avem n – stari ale modemului :

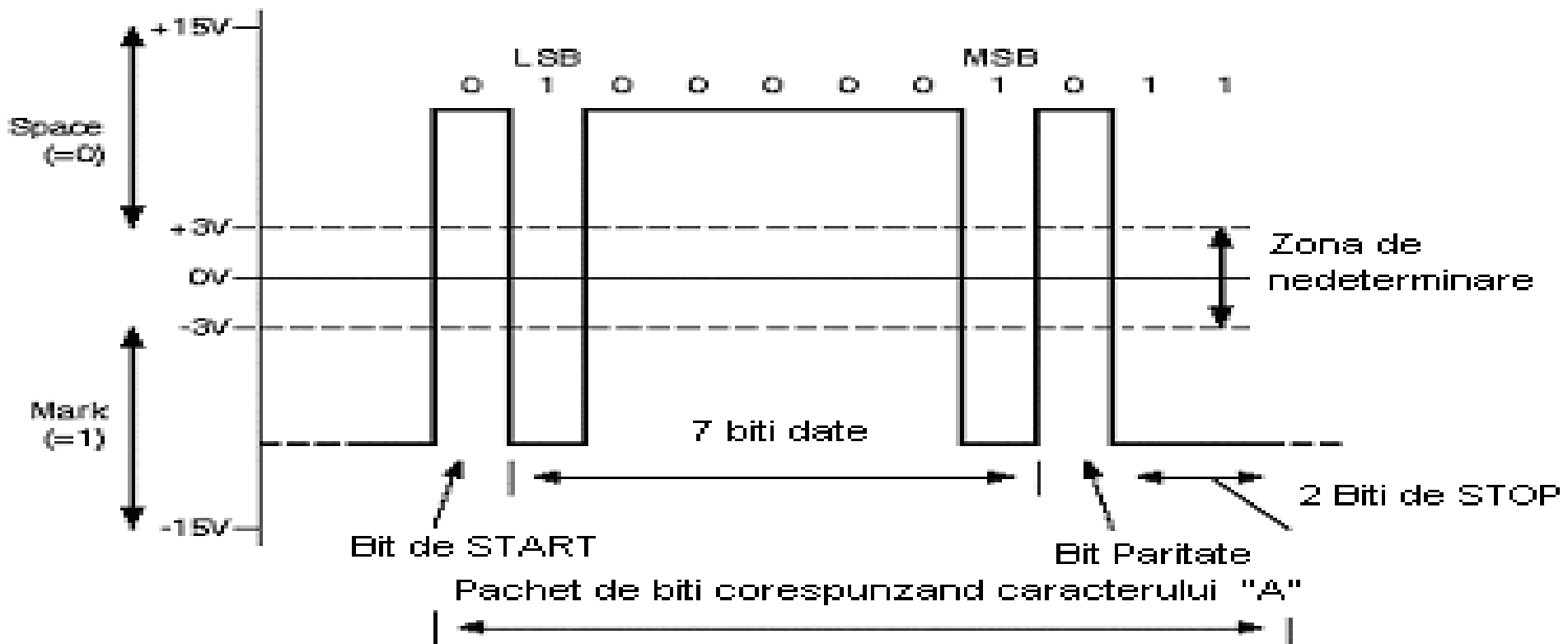
$$D= V_m \log_2 n \text{ [bps]}, n=2 \Rightarrow D = V_m$$

Ex. FAX (amplitudine, faza) – m.m. valori ptr. fiecare caracteristica

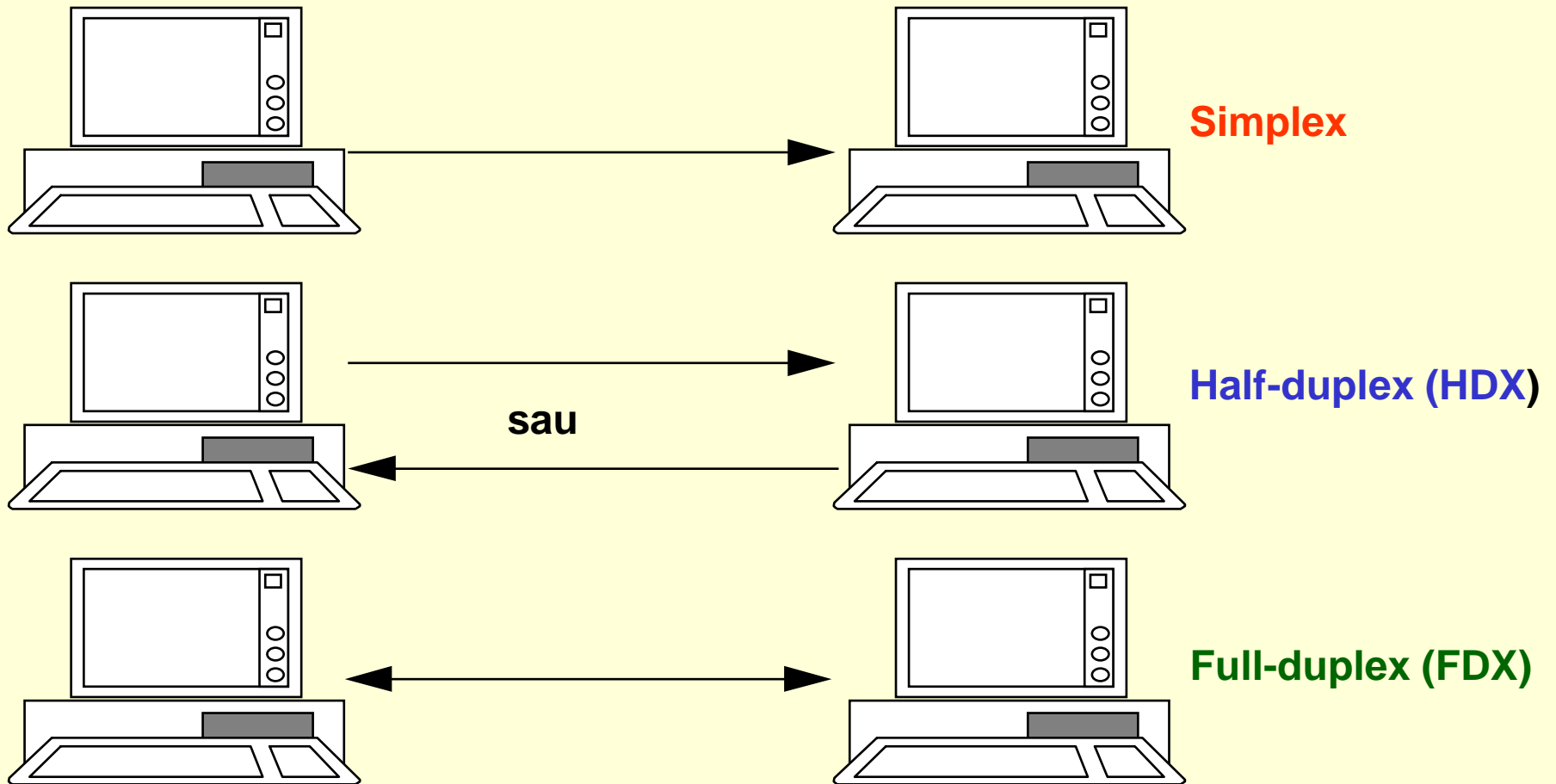
Se pot coda m.m. biti intr-un moment elementar :

$$D=9600 \text{ bps}, V_m = 1200 \text{ baud} \Rightarrow n = 2^8$$



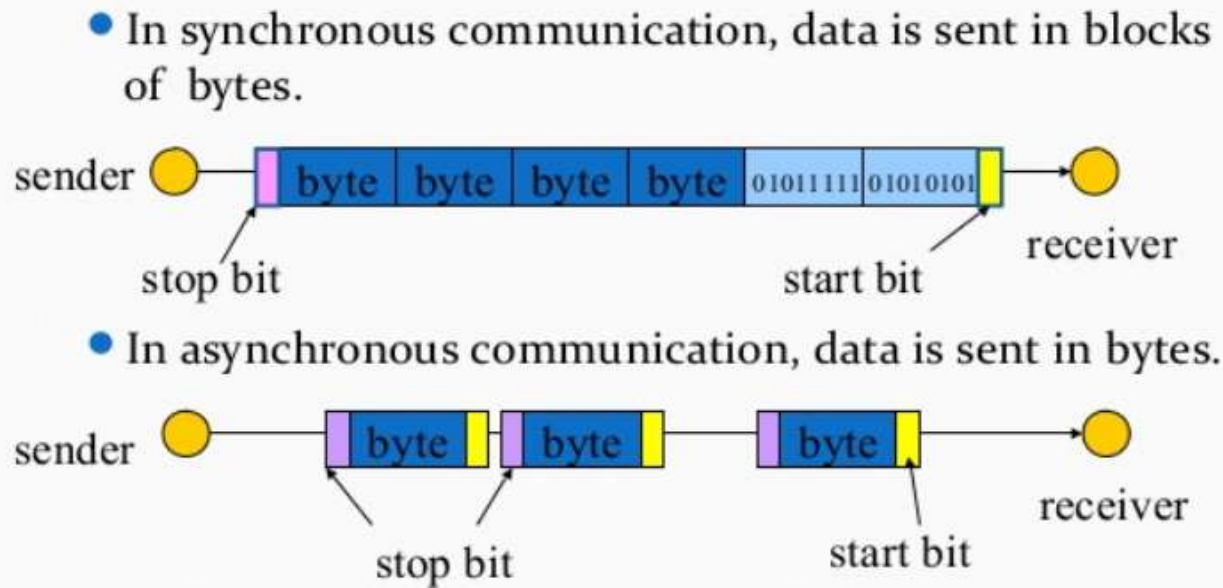


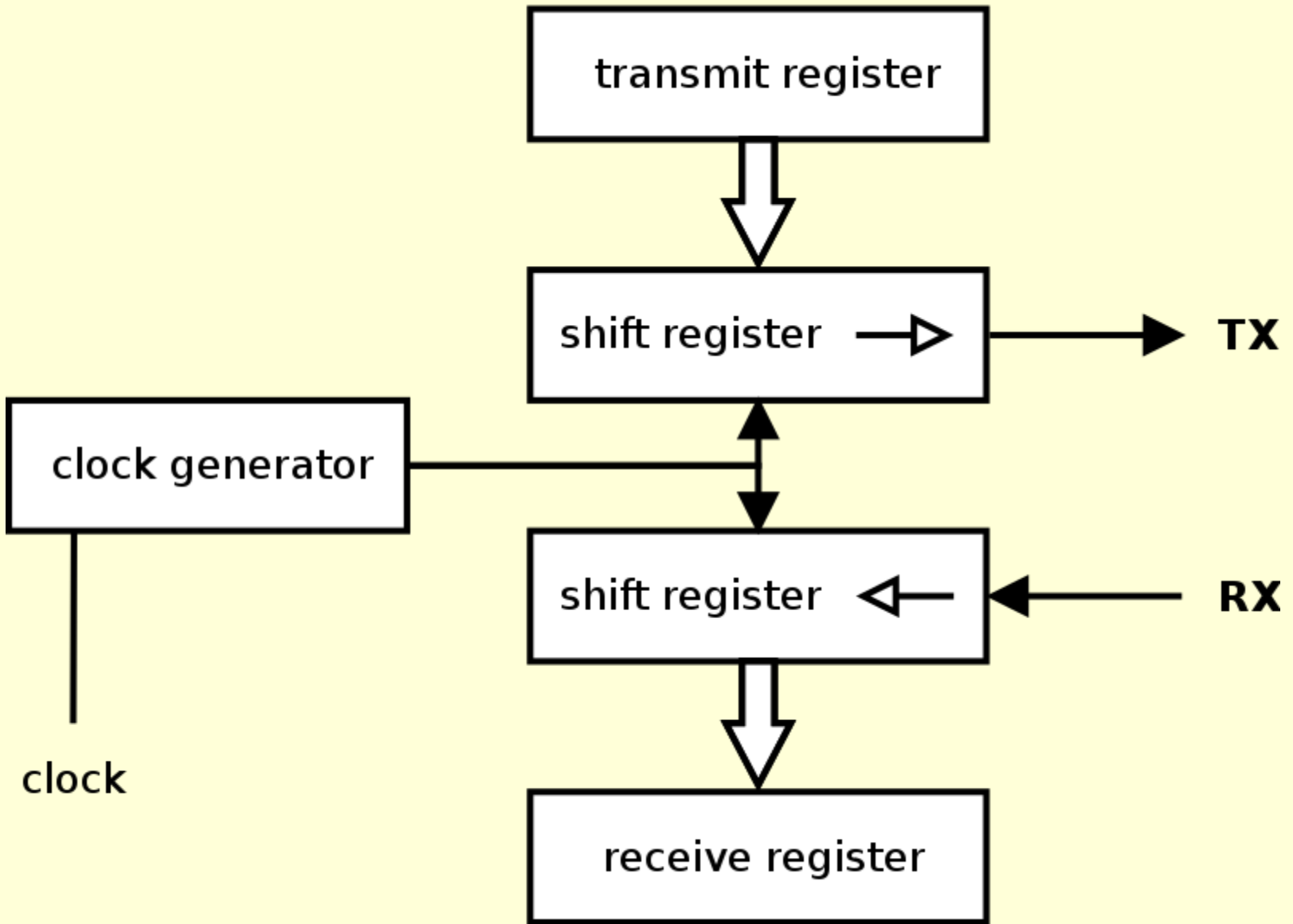
Moduri de Comunicatie



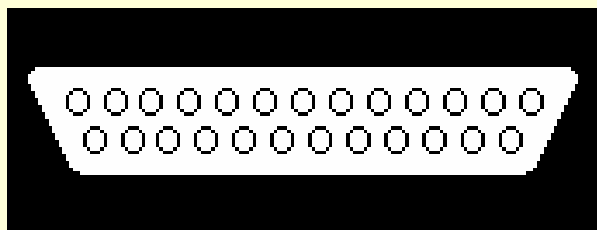
Diferențe cheie între transmisia serială sincronă și asincronă

1. În *transmisia sincronă* datele sunt transferate sub formă de cadre, pe de altă parte în *transmisia asincronă* datele sunt transmise 1 octet la un moment dat.
2. *Transmisia sincronă necesită un semnal de ceas* între transmitator și receptor, pentru a informa receptorul despre noul octet. Întrucât, în *transmisia asincronă* transmitatorul și receptorul *nu necesita un semnal de ceas*, deoarece datelor trimise îi este atașat un *bit de start* care indică începutul noului octet.
3. *Rata de transfer de date a transmisiei asincrone este mai redusă* decât cea a transmisiei sincrone.
4. *Transmisia asincronă este simplă și economică*, în timp ce, transmisia sincronă este complexă și costisitoare.
5. Transmisia sincronă este eficientă și are mai puțin *overhead/redundanta* în comparație cu transmisia asincronă.

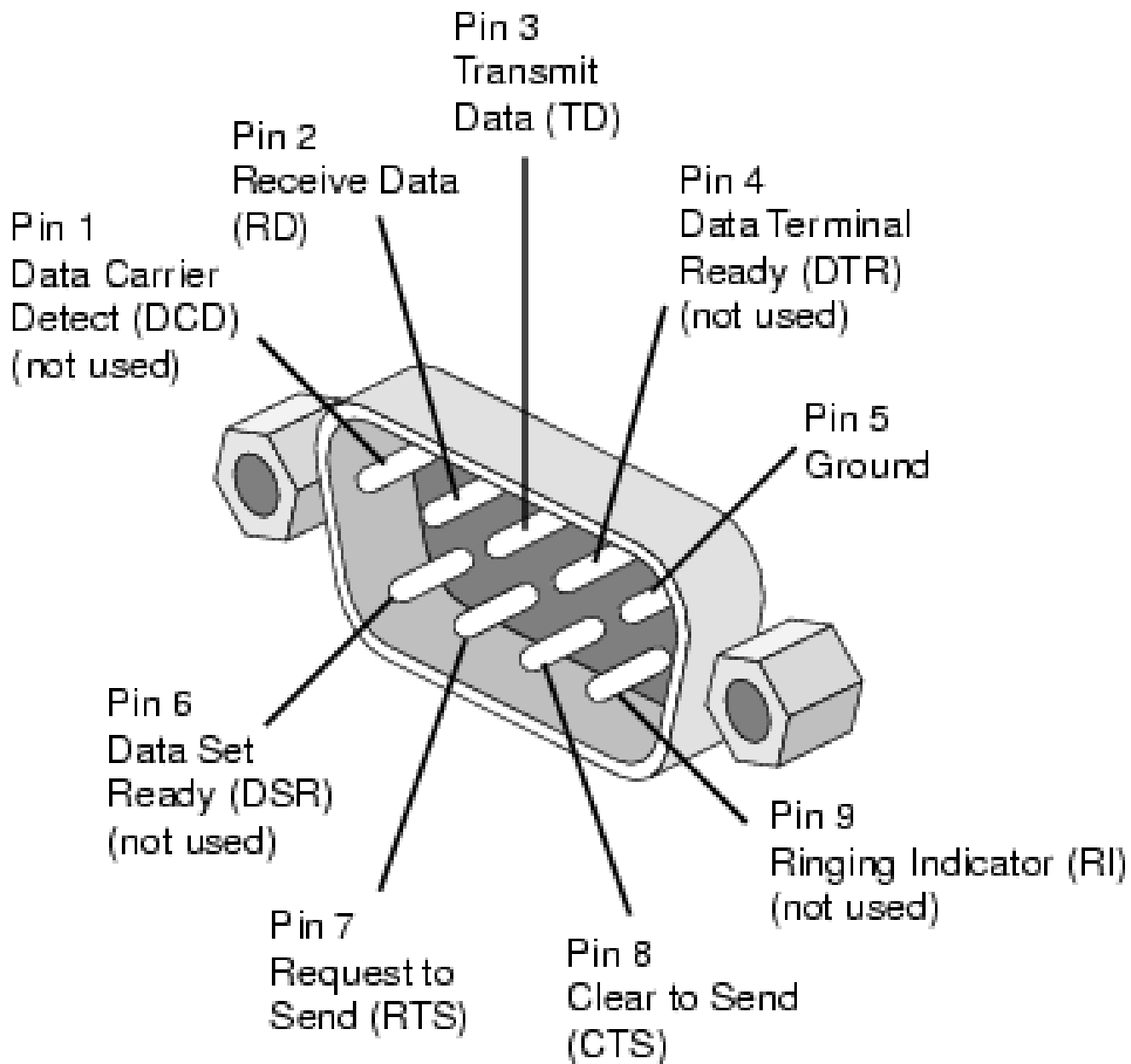




Semnal Pin (DB25/DB9)	Denumire	Funcție
TxD (2/3)	Transmit Data (Transmisie date)	Iesirea de date a portului serial (TXD)
RxD (3/2)	Receive Data (Receptie date)	Intrarea de date a portului serial(RXD)
CTS (5/8)	Clear to Send (Modem gata de emisie)	Aceasta linie indica faptul ca <i>modemul este pregatit pentru schimbul de date</i>
DCD (8/1)	Data Carrier Detect (Detectare purtatoare)	Când <i>modemul detecteaza un putător</i> de date la celălalt capăt al liniei telefonice, această linie devine activă
DSR (6/6)	Data Set Ready (Modem operational)	Acest pin comunica circuitului <i>UART</i> faptul ca <i>modemul este pregatit să stabileasca o legatura</i>
DTR (20/4)	Data Terminal Ready (Terminal de date gata)	Se comunica modemului ca <i>circuitul UART este pregătit de legătura</i>
RTS (4/7)	Request To Send (Cerere de emisie)	Aceasta linie informeaza modemul ca <i>UART este pregatit sa faca schimbul de date.</i>
RI (22/9)	Ring Indicator (Indicator apel)	Devine activ când modemul detecteaza un semnal de apel de la PTSN.



1. Functiile semnalelor interfetei seriale

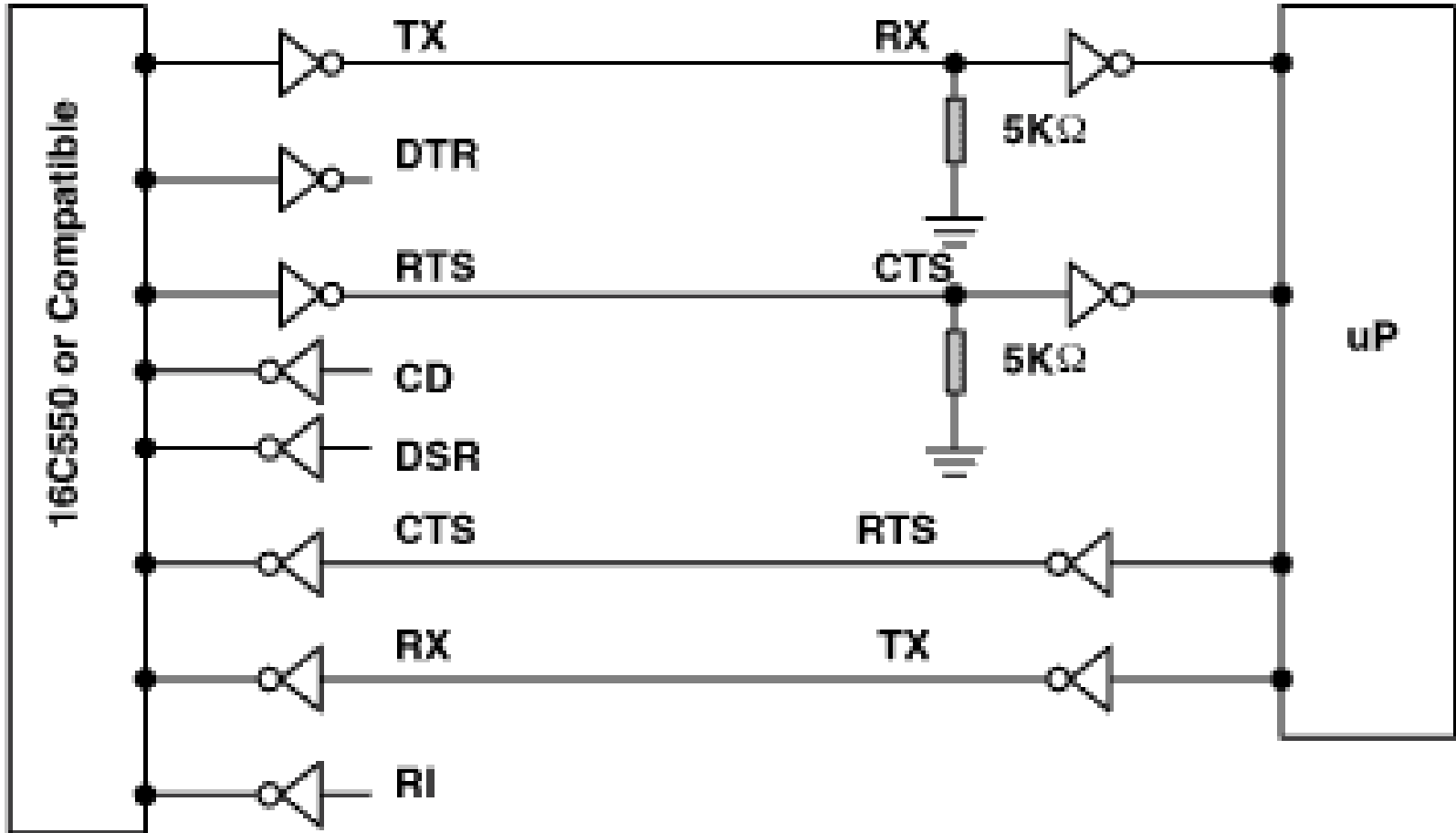


IBM PC

Peripheral Unit

MAX3209

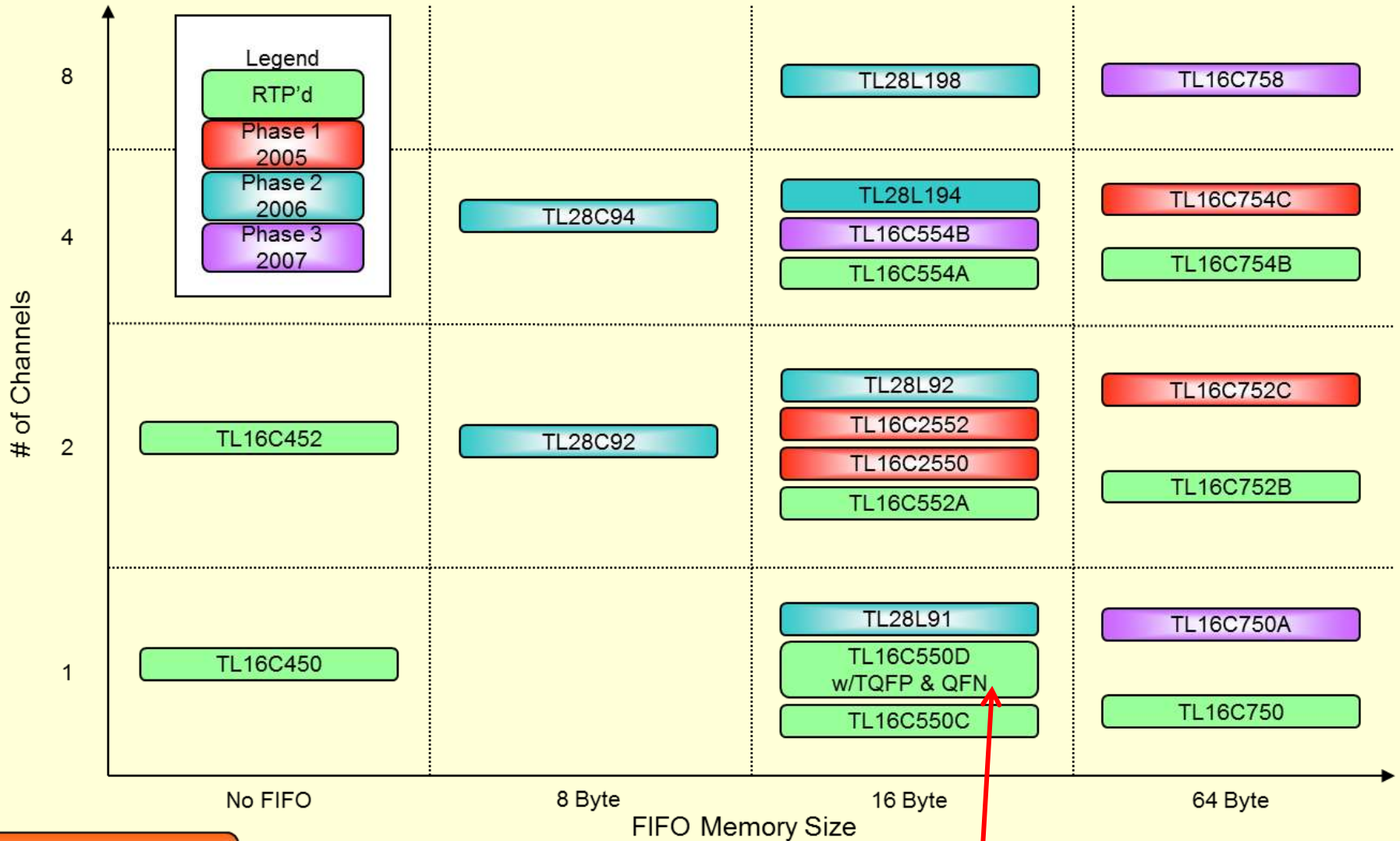
MAX3232E



2. EVOLUTIA UART-urilor IN PC

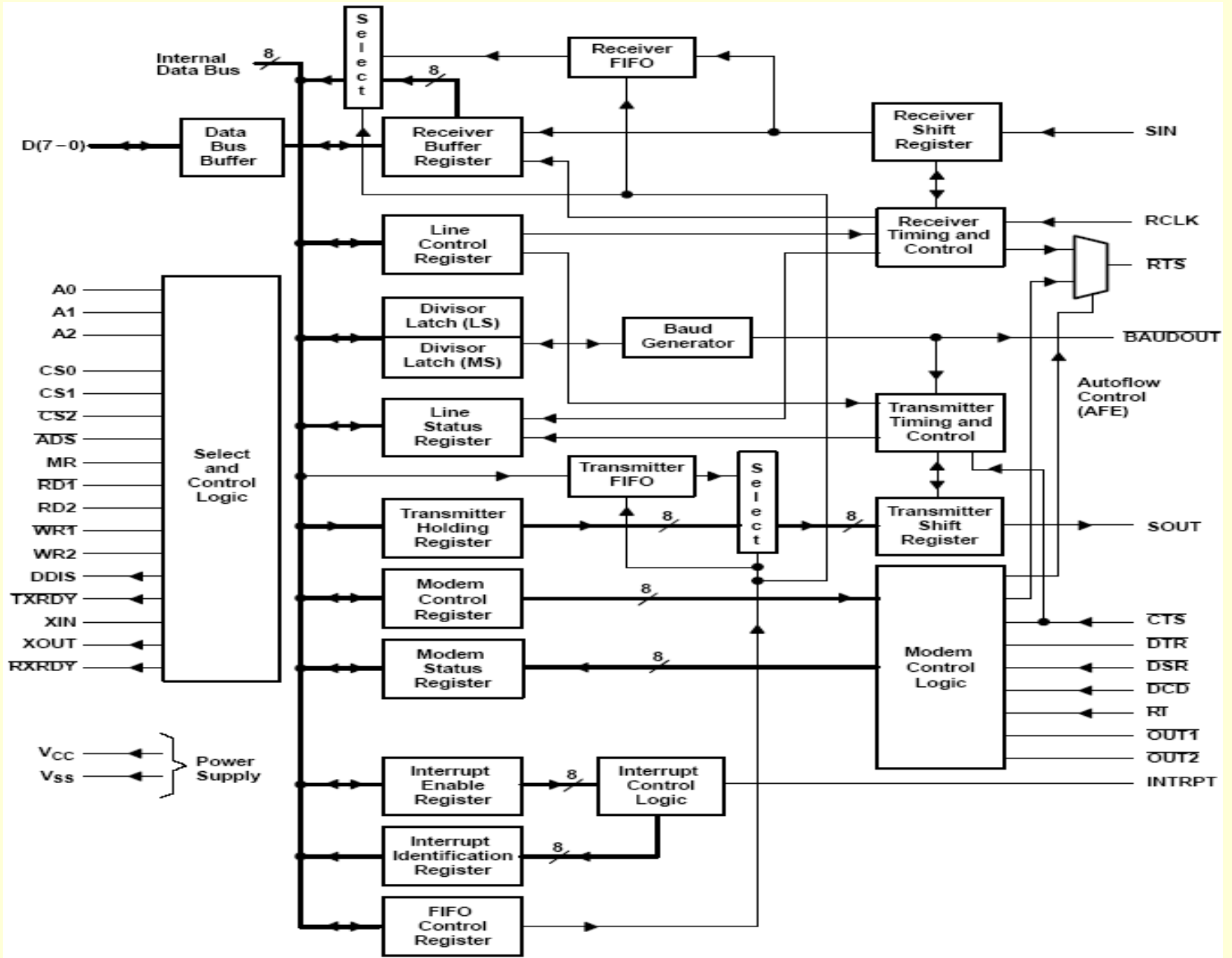
- **8250** UART-ul original cu 1-byte FIFO (USART 8251)
- **8250A** versiune a circuitului 8250 cu un registru suplimentar
- **16450** versiune îmbunătățită a UART-urilor anterioare, mai rapid
- **16550** UART cu 16-byte FIFO (dar FIFO nefuncțional)
- **16550A UART** cu aceleași caracteristici ca anteriorul, dar defectele eliminate
- **16650** UART cu 32-byte FIFO, debit până la 460.8 Kbps
- **16750** UART cu 64-byte FIFO, debit 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

UART Roadmap

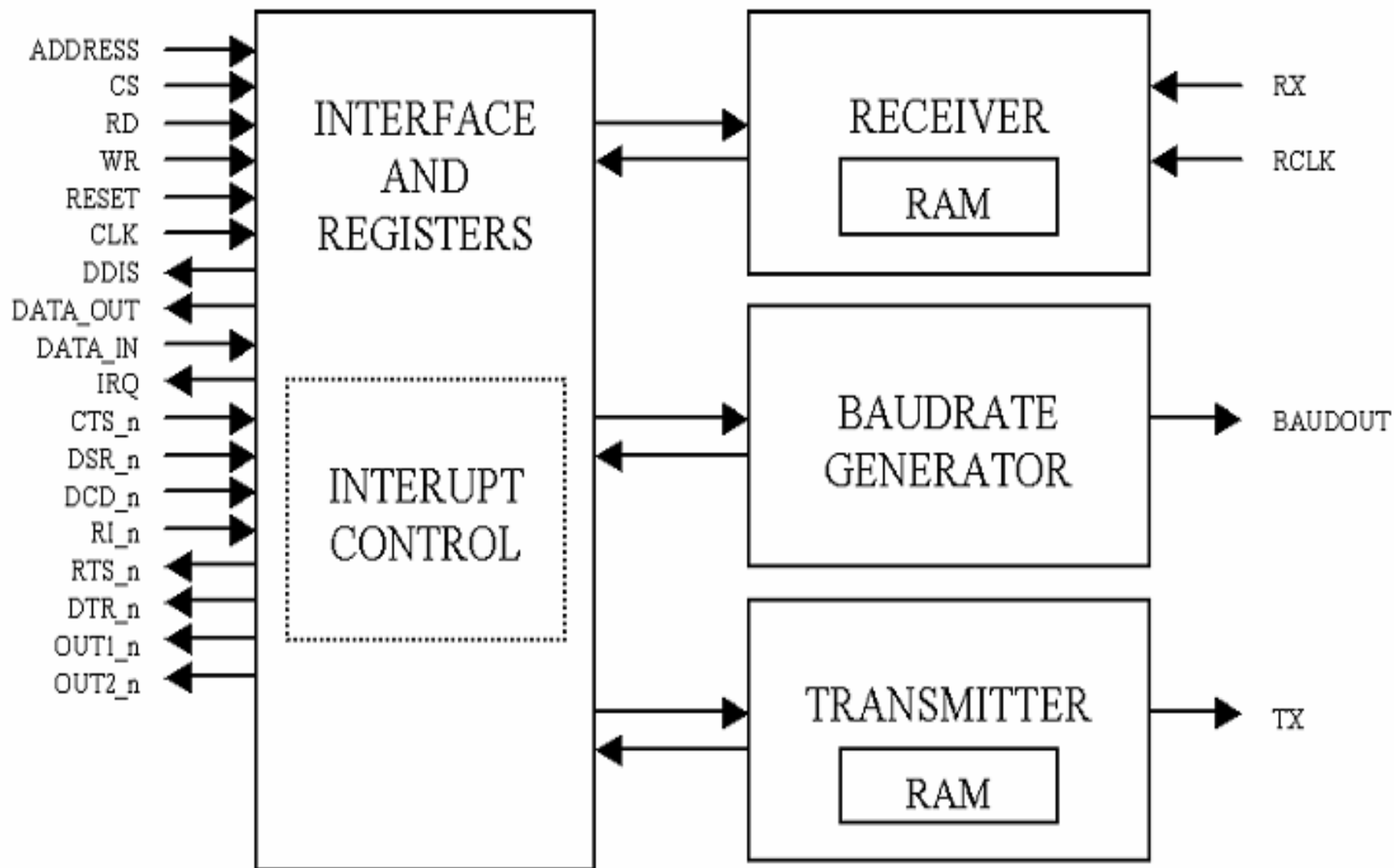


[More Information](#)

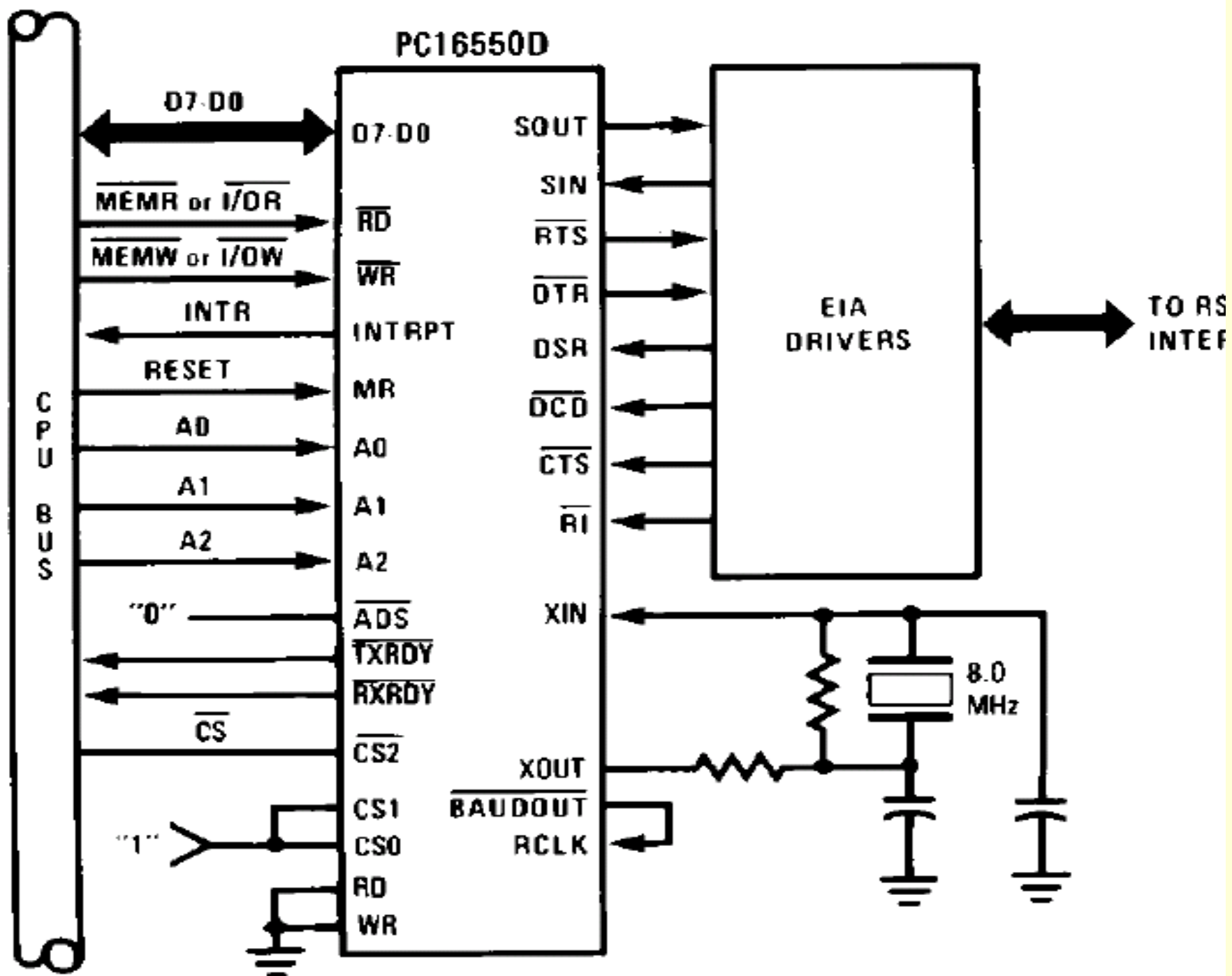
3. ARCHITECTURA UART 16550



UART



1. **Blocul de interfață și control cu sistemul** > realizează legătura între circuit și magistralele sistemului
2. **Blocul LSR și LCR** conține registrele de Control Linie și Stare Linie
3. **Blocul de Logica Control Modem** încorporează registrele de Control Modem, Stare Modem și registrul Scratch Pad
4. **Blocul de Control Întreruperi** > generează semnalul de întrerupere spre procesor
5. **Blocul Control FIFO** include registrul de control a stivei FIFO
6. **Blocul Generare Baud Rate** conține registrele de divizare a frecvenței de ceas CLKIN
7. **Blocurile de Transmisie + FIFO și Recepție+FIFO** conțin registrele de transmisie respectiv recepție date și stivele aferente
8. **Registrul testare** producător permite testarea generatorului de baud rate și testarea erorilor de paritate și depășire.



ADRESELE si ÎNTRERUPERILE REZERVATE PORTULUI SERIAL

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

```
#include <dos.h>
#include <stdio.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++;
    }
}
```

4. PROGRAMARE UART. Registrele circuitului 8250/16550A

Adresa COM1/COM2	R/W	Abr.	Numele Registrului
3F8h (2F8h)	Write	THR	Transmit Holding Buffer (Buffer de transmisie)
	Read	RBR	Receive Buffer (Buffer de receptie)
	Read/ Write	DLLB	Divisor Latch Low Byte (Registru divizor, octetul „low”, dacă b7=1 din LCR) -DLLB
3F9h (2F9h)	Read/ Write	IER	Interrupt Enable Register (Registru de validare a întreruperilor)
	Read/ Write	DLHB	Divisor Latch High Byte (Registru divizor, octetul “high”, dacă b7=1 din LCR) DLHB
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 (2FFh)	Read/ Write	SPR	Scratchpad Register - numai la 16550

THR – Transmit Holding Register (R/W)

TSR – Transmit Shift Register

RBR – Receiver Buffer Register (R/W)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

IER – Interrupt Enable Register (R/W)

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 (MSR). |
| 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)** **Divizor = 1843200 / (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

Divizorii frecventei de 1.8432 MHz pentru diferite debite binare

- **IIR – Interrupt Identification Register (Read)**

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 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)

FCR - FIFO Control Register (W)

TRIGGER 1	TRIGGER 0	0	0	DMA MODE	TXFIFO RESET	RXFIFO RESET	ENABLE
-----------	-----------	---	---	----------	--------------	--------------	--------

Bit	Descriere		
-----	-----------	--	--

7-6	B7	B6	Nivelul pragului de întrerupere 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. Arhitectura PC nu acceptă acest mod.

2 Șterge stiva FIFO de transmisie

1 Șterge stiva FIFO de receptie

0 Activează FIFO

Advanced Port Settings
✕

Use FIFO buffers (requires 16550 compatible UART)

Select lower settings to correct connection problems.
 Select higher settings for faster performance.

Receive Buffer: Low (1)

 High (14)

Transmit Buffer: Low (1)

 High (16)

OK

Cancel

Defaults

LCR – Line Control Register (R/W)

DIVISOR LATCH ACCESS BIT	BREAK	STICK	EVEN PARITY SELECT	PARITY ENABLE	STOP BIT SELECT	WORD LENGTH SELECT 1	WORD LENGTH 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					
	1	2 biti de stop pentru caractere de 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			

- MCR – Modem Control Register (R/W)**

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)**

RXFIFO HOLDS ERROR (16550)	TRANSMIT EMPTY	TRANSMIT HOLDING REG. EMPTY	BREAK DETECTED	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	RECEIVE DATA AVAILABLE
-------------------------------------	-------------------	-----------------------------------	-------------------	------------------	-----------------	------------------	------------------------------

Bit	Description
0	Data available
1	Overrun error
2	Parity error
3	Framing error
4	Break signal received
5	THR is empty
6	THR is empty, and line is idle
7	Error data in FIFO

- MSR – Modem Status Register (R)**

DATA CARRIER DETECT	DATA SET READY	RING INDICATE	CLEAR TO SEND	DELTA DCD	DELTA DSR	DELTA RI	DELTA CTS
---------------------------	----------------------	------------------	---------------------	--------------	--------------	-------------	--------------

- Registrul Scratchpad (16550) (R/W)**

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

BAUD RATE GENERATOR PROGRAMMING TABLE

All boards will typically have the 1.8432 MHz crystal. Some boards will also allow selection of **the 7.3728 MHz clock.**

BAUD RATE	1.8432 MHZ	7.3728 MHZ
50	2304	9216
75	1536	6144
110	1047	4188
134.5	857	3428
150	768	3072
300	384	1536
600	192	768
1200	96	384
2400	48	192
3600	32	128
4800	24	96
7200	16	64
9600	12	48
19.2K	6	24
38.4K	3	12
57.6K	2	8
115.2K	1	4
230.4K	x	2
460.8K	x	1

5. FUNCTII BIOS PENTRU PORTUL SERIAL (INT 14h)

AH (Stare linie) =LSR

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 transmisie;
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 transiterii unui caracter;
3	(Framing Error): eroare de cadrare (nu s-a receptionat 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

Funcția	Operația	Intrare	Ieșire
AH=00	Inițializare	AL = parametrii comunicatiei # 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=car. citit
AH=03	Citire port stare	DX = nr. port serial (0-3)	AH = stare linie AL = stare modem

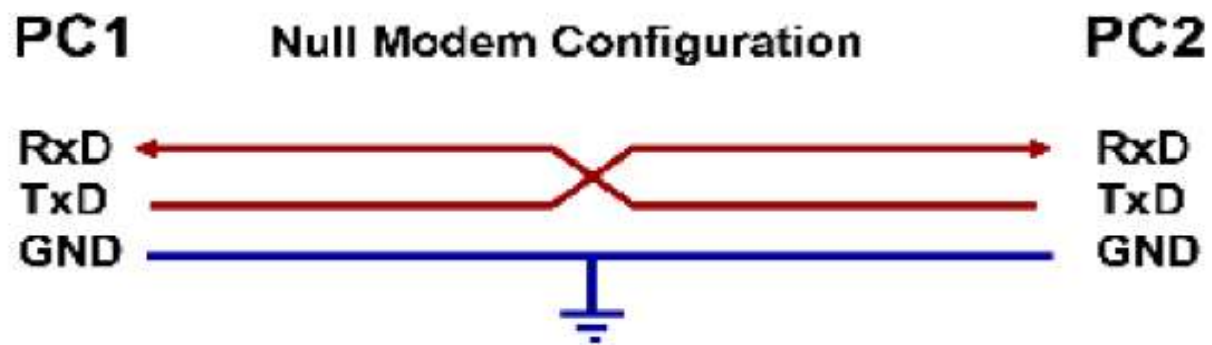
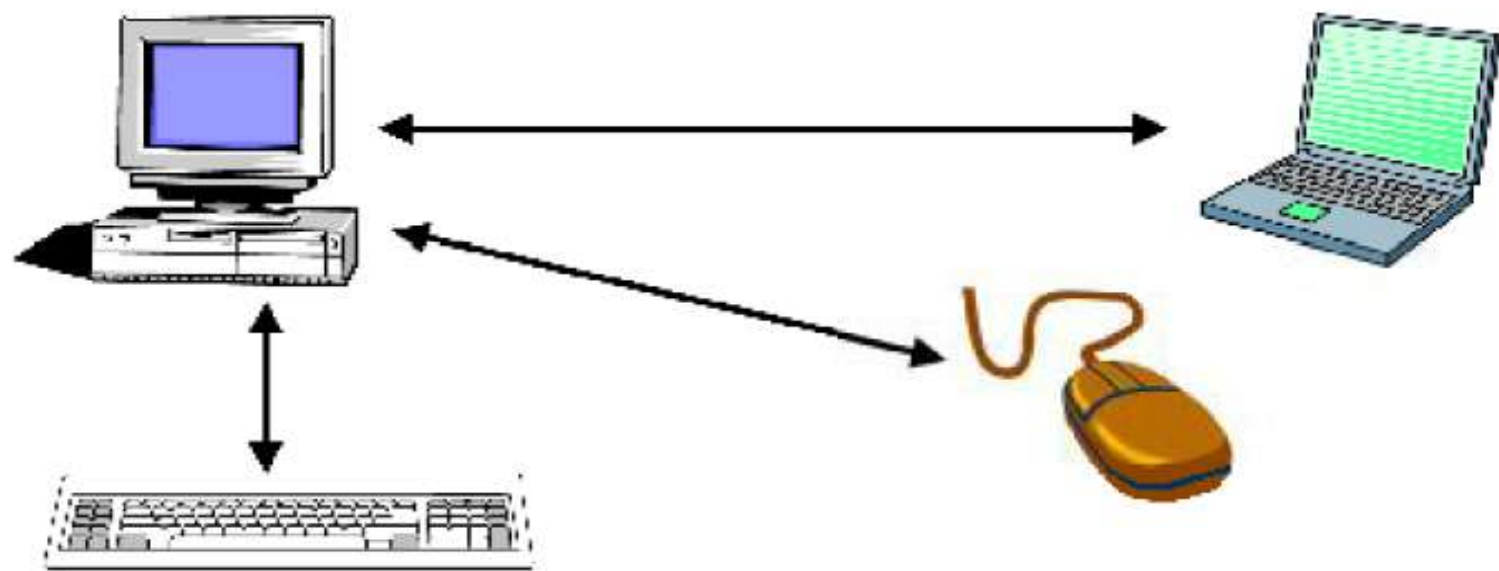
- Biții 1-0 : lungimea caracterului (10 - 7 biți; 11 - 8 biți)
- Bit 2 : numărul biților de stop (0- 1 bit de stop ; 1 - 2 biți de stop)
- Biți 4–3 : controlul parității (01- p. impară, 11 – p. pară, x0 fără paritate)
- Biți 7–5 : viteza de transmisie (000 -110 bps,111 -9600 bps)

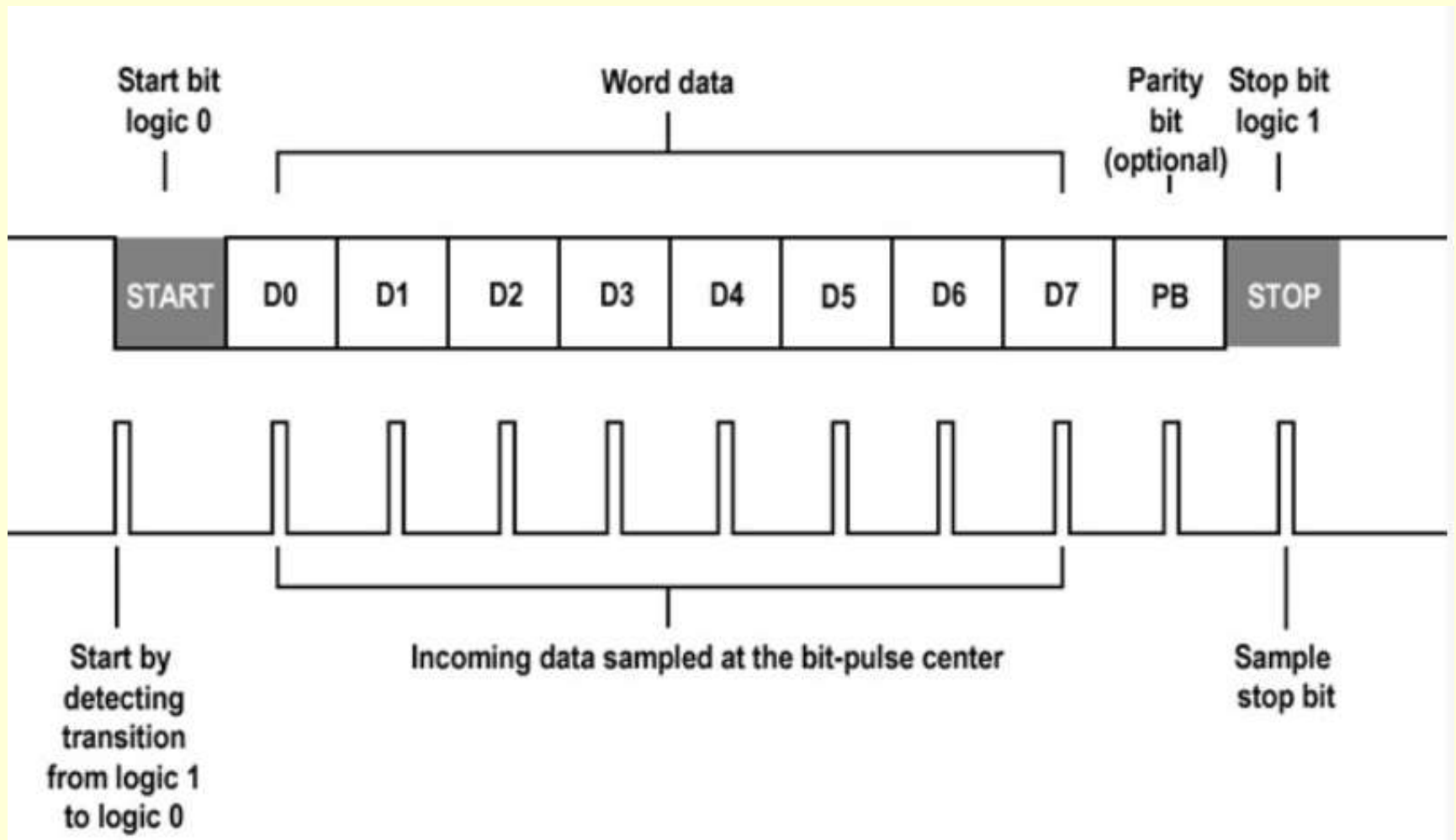
6. APLICATII

Realizati o conexiune seriala intre 2 PC-uri folosind interfata COM1. Scrieti o aplicatie care permite schimbarea mesajelor text intre 2 utilizatori (chat).

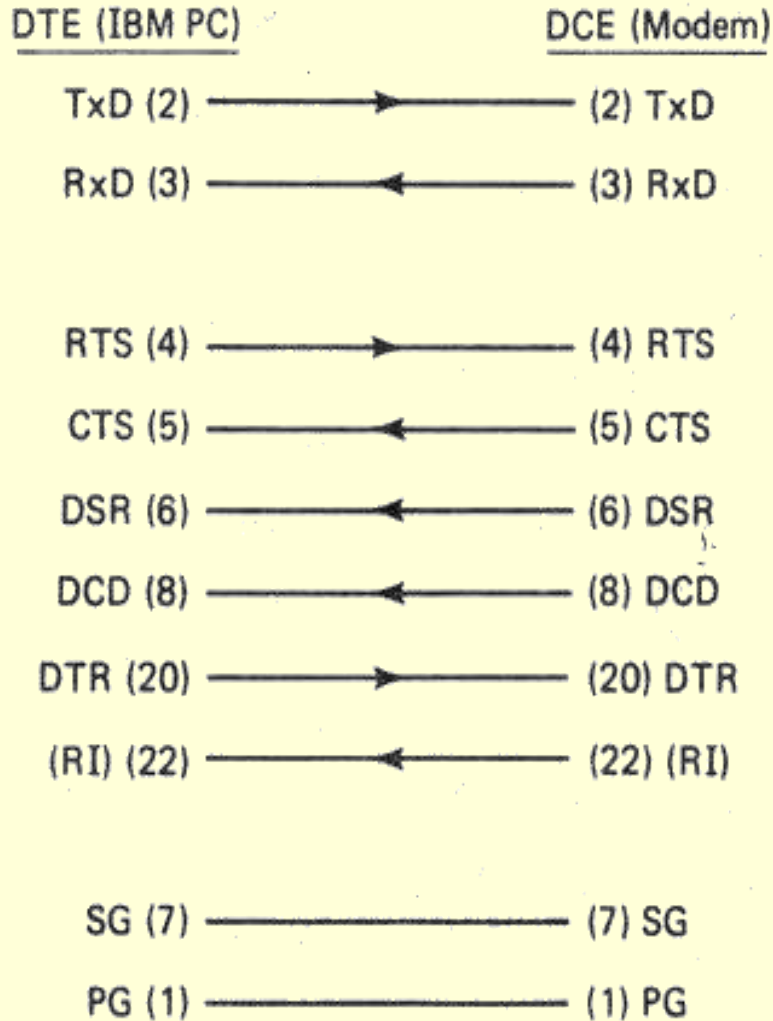
Date: COM1: 9600Baud, 2biti STOP, No parity; Folositi servicii INT 14h.

Tema. Rescrieti aplicatia folosind interuperile.

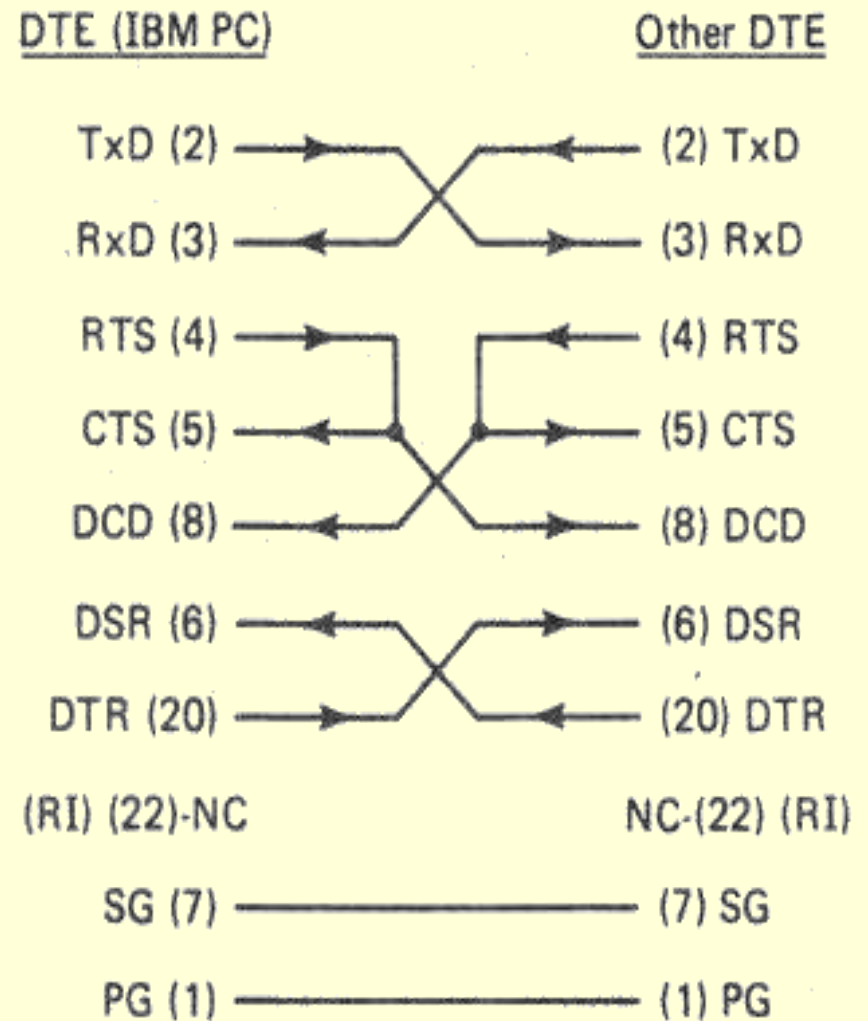




Ideal Case



Null Modem Adapter



References:

<http://www.beyondlogic.org/>

RXFIFO HOLDS ERROR (16550)	TRANSMIT EMPTY	TRANSMIT HOLDING REG. EMPTY	BREAK DETECTED	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	RECEIVE DATA AVAILABLE
----------------------------	----------------	-----------------------------	----------------	---------------	--------------	---------------	------------------------

LSR

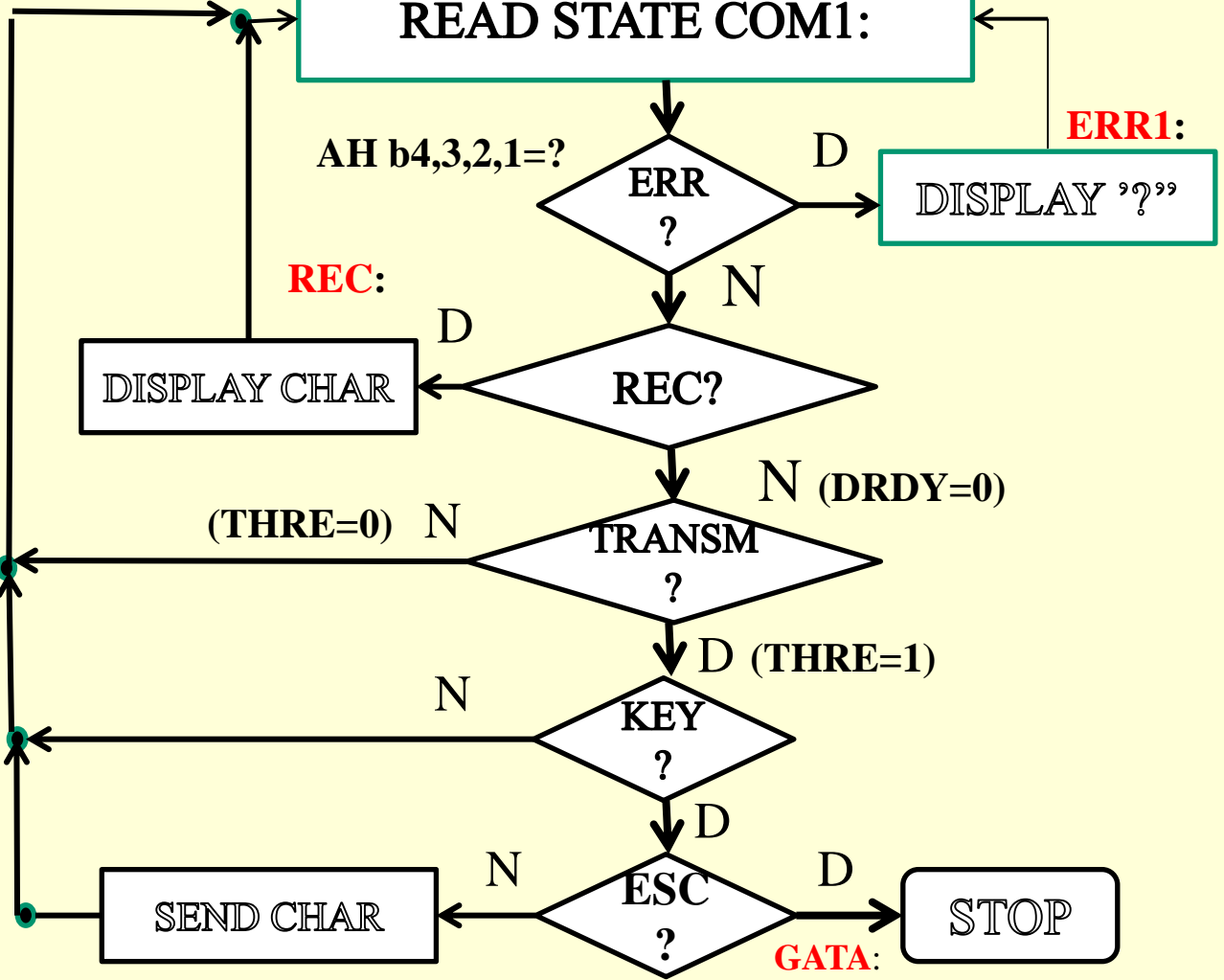
INCEP:



STARE:



AH=LSR, AL=MSR



REC:

ERR1:

GATA:

```

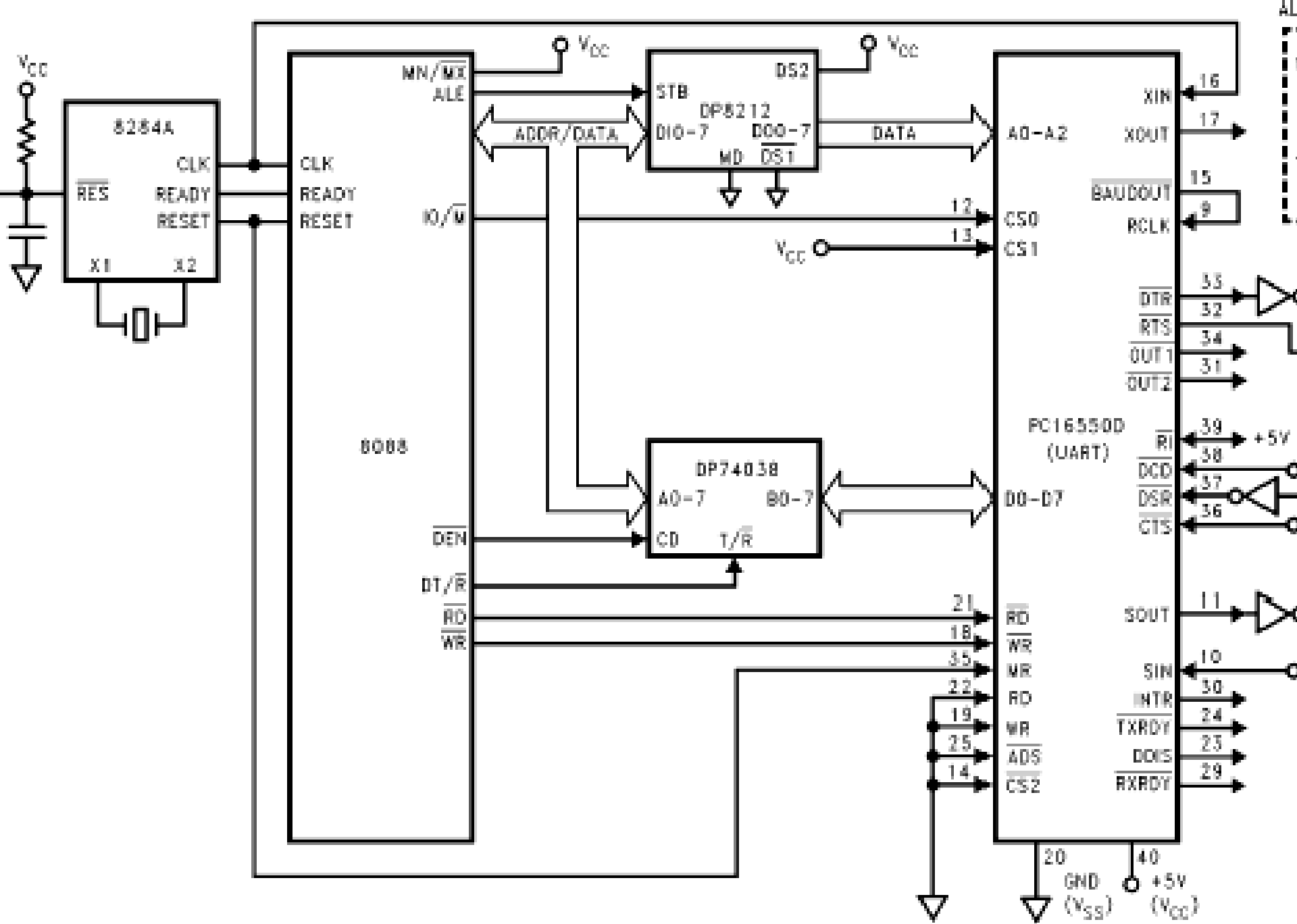
title program comunicatie seriala ;folosind serviciul INT ;14h
code segment para 'code'
    assume cs:code,ds:code,ss:nothing
    org 100h

incep:    mov     ah,0      ;init. UART
          mov     dx,0      ;COM1
          mov     al,11100111b
;9600 baud,fara parit.,8 biti/car,2STOP
          int     14h
stare:    mov     ah,3      ;servici stare UART
          mov     dx,0      ;com1
          int     14h
          test    ah,1eh    ;test biti eroare
          jnz    err1
          test    ah,01h    ;test receptie
          jnz    receptie
          test    ah,20h    ;test buffer tr.
          jz     stare ; THRE=1?
          mov     ah,1      ;astept tasta
          int     16h      ; z=0
          jz     stare
          mov     ah,0      ;citire tasta
          int     16h      ;al= car. tastat
          cmp    al,1bh    ;al= ESC ?
          jz     gata      ;transmit caracter
          mov     ah,1      ;emisie car. tastat
          mov     dx,0
          int     14h
          jmp    stare

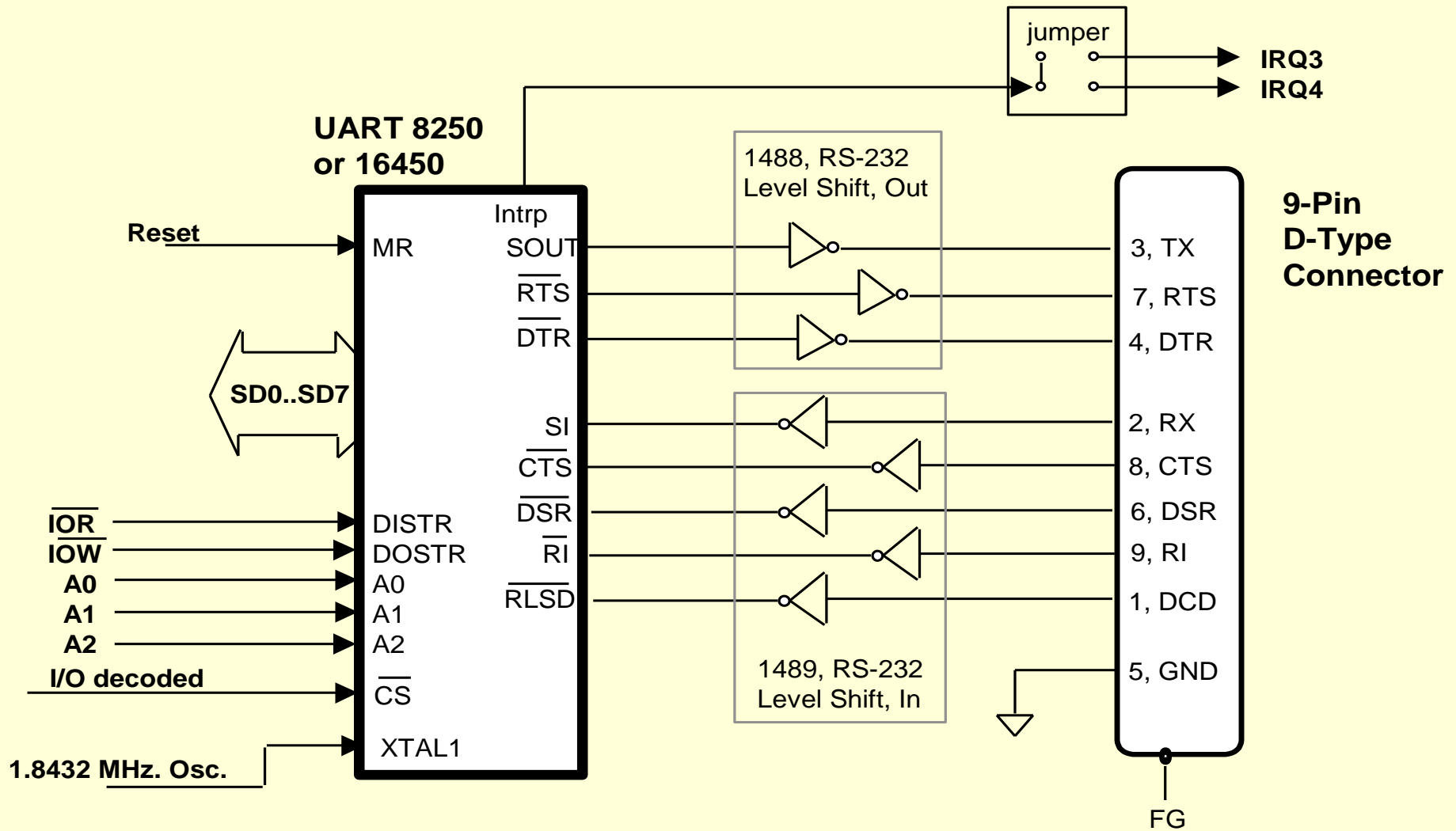
          gata:
code      ends

          mov     ah,2      ;serv. rec.caracter
          mov     dx,0
          int     14h
          push   ax        ;salvare caracter
          mov     bx,0     ;atrib. de afisare
          mov     ah,0eh
          int     10h      ;afisare
          pop    ax
          cmp    al,0dh    ;CR?
          jnz    stare
          mov     al,0ah   ;afisare LF
          mov     bx,0
          mov     ah,0eh
          int     10h
          jmp    stare
          mov     ah,2
          mov     dx,0
          int     14h
          mov     al,'?'
          mov     bx,0
          mov     ah,0eh
          int     10h
          jmp    stare
          int     20h
          end     incep

```



The Serial Port



DISTR, DOSTR = Data Input and Data Output Strobes.

Rezumat

- Frumusețea portului serial este simplitatea sa
- Nu se poate face o comunicare mai simplă și mai ieftină decât doar două fire pe care le folosește un port serial. Acest lucru o face să fie perfectă în multe cazuri.
- Un port serial este atât de omniprezent încât uneori chiar și cele mai noi dispozitive USB, cum ar fi modemurile 4G, îl emulează intern UART și este un protocol simplu, cu două fire, pentru schimbul de date seriale.
- Asincron, înseamnă că nu există un ceas partajat, astfel încât, pentru ca UART să funcționeze, trebuie să fie configurat cu aceeași rată binară de transmisie pe ambele sensuri ale conexiunii. Biții de start și stop sunt utilizați pentru a indica unde încep și unde se termină datele utilizatorului sau pentru a "încadra" datele. Un bit de paritate opțional poate fi utilizat pentru a detecta erorile pe un singur bit.
- Transmisia asincronă este încă un protocol de date seriale utilizat pe scară largă, dar în ultimii ani a fost înlocuit în unele aplicații de tehnologii precum SPI, I2C, USB și Ethernet.
- Chiar dacă nu vedeți niciun conector pe laptopul dvs. nou-nouț, există șanse ca placa sa de bază să aibă încă în interior circuitele unui UART 16550.

Intel® 400 Series Chipset On-Package Platform Controller Hub

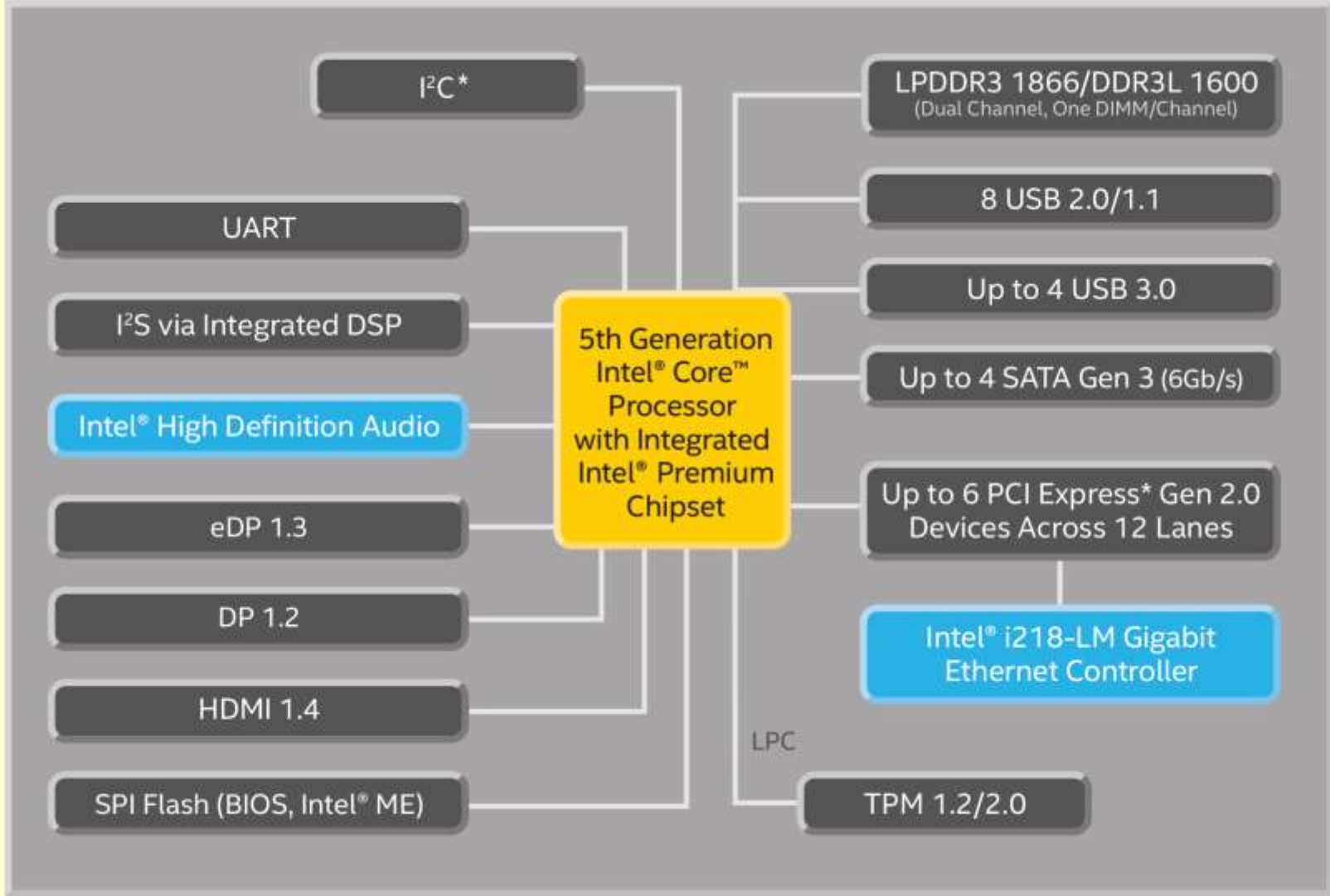
Online Register Database

UART Memory Mapped Registers

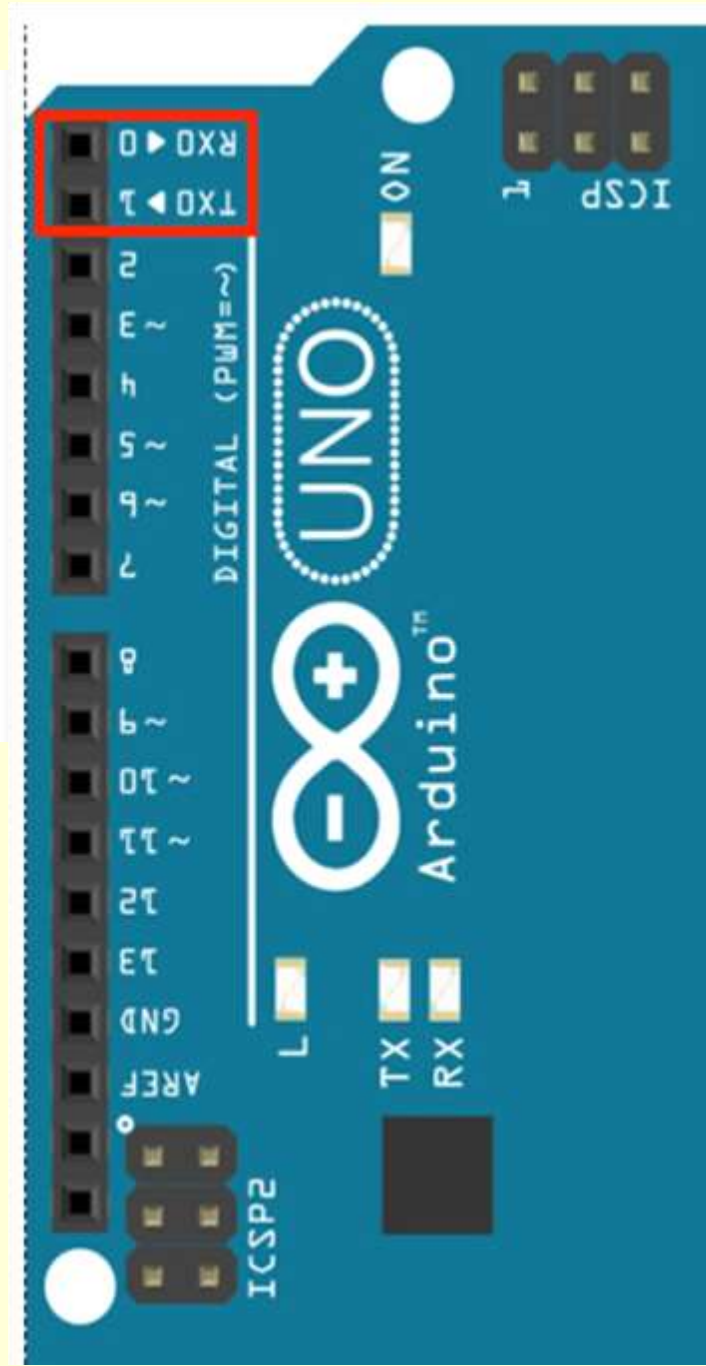
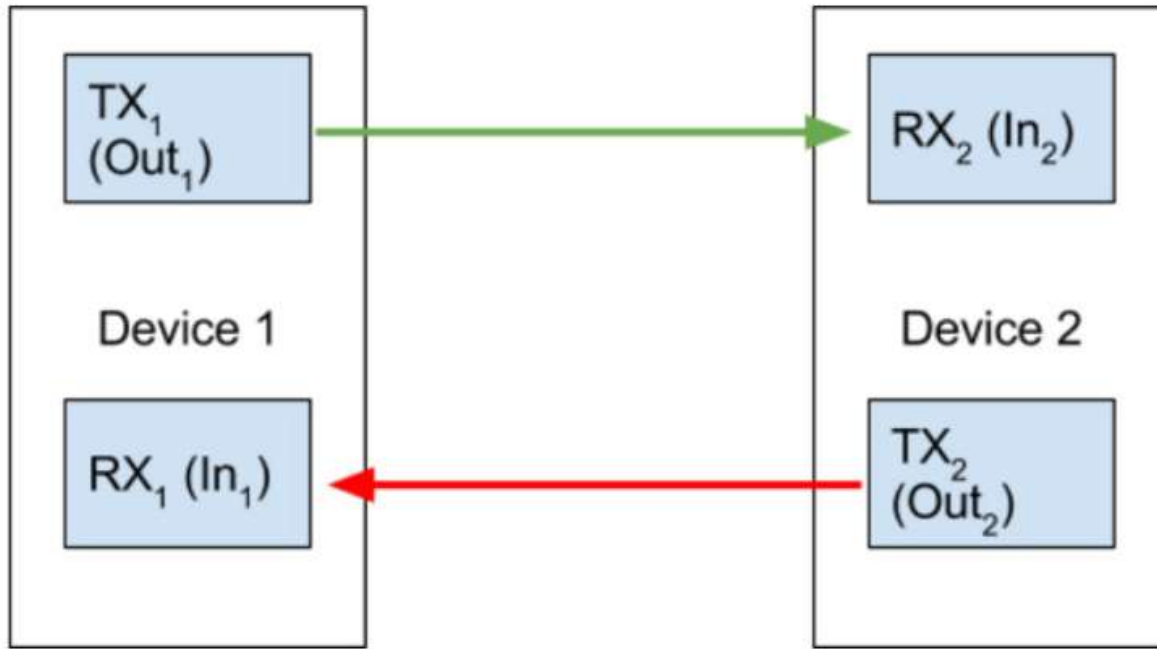
The registers in this section are memory-mapped registers based on the BAR defined in PCH Configuration space.

Summary of Bus:, Device:, Function: (MEM)

Offset	Size (Bytes)	Register Name (Register Symbol)	Default Value
0h	4	Receive Buffer Register (RBR)	0h
0h	4	Transmit Holding Register (THR)	0h
0h	4	Divisor Latch Low Register (DLL)	0h
4h	4	Interrupt Enable Register (IER)	0h
4h	4	Divisor Latch High (DLH)	0h
8h	4	Interrupt Identification (IIR)	1h
8h	4	FIFO Control (FCR)	1h
ch	4	Line Control Register (LCR)	0h
10h	4	MCR (MCR)	0h
14h	4	LSR (LSR)	60h
18h	4	MSR (MSR)	0h
1ch	4	SCR (SCR)	0h
30h	4	SRBR STHR0 (SRBR STHR0)	0h



Arduino Serial Communication



- Atunci când implementează comunicarea serială UART pe majoritatea platformelor embedded, cum ar fi Arduino, utilizatorul nu trebuie să se ocupe de comunicarea la nivel de biți.
- În schimb, platforma oferă *biblioteci software de nivel superior* care sunt singurul aspect al procesului de comunicare cu care utilizatorul trebuie să se ocupe.
- În platforma Arduino, utilizatorii pot utiliza bibliotecile Serial și SoftwareSerial pentru a implementa comunicarea pe UART pentru proiectele lor.
- O scurtă referință C ++ pentru Arduino Serial și SoftwareSerial
Inițializare și utilizare.

Serial and SoftwareSerial Method	Purpose	Code	Explanation
Constructor (SoftwareSerial only)	Define the GPIO pins that will serve as the UART RX and TX lines	SoftwareSerial comms (2 , 3);	Defines a serial connection with RX line on GPIO 2 and TX line on GPIO 3.
<i>begin</i>	Define the baud rate (transmission speed) for the serial connection in range 4800 to 115200	comms.begin(9600);	Communication on the “comms” serial port will occur at 9600 baud.
<i>print</i>	Write byte data converted into human-readable characters over serial connection	comms.println(“Hello World”);	Writes bytes equivalent to <i>Hello World</i> (human-readable characters)
<i>write</i>	Write raw byte data over the serial connection	comms.write(45);	Writes byte with value 45.
<i>available</i>	Evaluates to true when data is available over the serial connection	if (comms.available())	Enter <i>if</i> statement if there is data available to read over the serial connection
<i>read</i>	Read data from the serial connection	comms.read();	Reads from serial connection

```
YourDuinoStarter_SerialMonitor_SEND_RCVE | Arduino 1.0.3
File Edit Sketch Tools Help

Serial Monitor

YourDuinoStarter_SerialMonitor_SEND_RCVE
Serial.print(" ");
Serial.print(ByteReceived, HEX);
Serial.print(" ");
Serial.print(char(ByteReceived));

if(char(ByteReceived) == '1')
{
  digitalWrite(led,HIGH);
  Serial.print(" LED ON ");
}

if(char(ByteReceived) == '0')
{
  digitalWrite(led,LOW);
  Serial.print(" LED OFF");
}

Serial.println(); // End the line

} // END Serial Available
```

COM20

Send

--- Start Serial Monitor SEND_RCVE ---
(Decimal) (Hex) (Character)

65	41	A
66	42	B
67	43	C
49	31	1 LED ON
48	30	0 LED OFF
68	44	D
69	45	E
70	46	F

Autoscroll

No line ending (selected)
Newline
Carriage return
Both NL & CR

9600 baud (selected)
4800 baud
9600 baud
14400 baud
19200 baud
28800 baud
38400 baud
57600 baud
115200 baud

Device Manager

File Action View Help

- home-pc
 - Audio inputs and outputs
 - Batteries
 - Bluetooth
 - Computer
 - Disk drives
 - Display adapters
 - DVD/CD-ROM drives
 - Human Interface Devices
 - IDE ATA/ATAPI controllers
 - Imaging devices
 - Keyboards
 - Mice and other pointing devices
 - Monitors
 - Network adapters
 - Portable Devices
 - Ports (COM & LPT)
 - USB Serial Port (COM3)

Done uploading

Binary sketch size: 2,912 bytes (of a 30,720 byte maximum)

47 Arduino Duemilanove w/ ATmega328 on COM20

Tema 1. Gasiti si studiatii materiale despre VIRTUAL COMMUNICATIONS PORT (VCP).

Tema 2. Analizati aplicatia de mai jos.

```
#include <bios.h>
#include <conio.h>
#define COM1    0
#define DATA_READY 0x100
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)
int main(void)
{
    int in, out, status;
    bioscom(0, SETTINGS, COM1); /*initialize the port*/
    cprintf("Data sent to you: ");
    while (1)
    {
        status = bioscom(3, 0, COM1); /*wait until get a data*/
        if (status & DATA_READY)
            if ((out = bioscom(2, 0, COM1) & 0x7F) != 0) /*input a data*/
                putchar(out);
            if (kbhit())
            {
                if ((in = getch()) == 27) /* ASCII of Esc*/
                    break;
                bioscom(1, in, COM1); /*output a data*/
            }
    }
    return 0;
}
```