**TEORIE FUNDAMENTALA**

INTERFATA = punctul de intalnire unitatii centrale cu dispozitivele periferice cu scopul transferului de date. Acestea pot fi bidirectionale si hardware sau software.

ECHIPAMENTUL PERIFERIC = dispozitivul conectat la un sistem de calcul gazda ce extinde functionalitatea acestuia.

PROTOCOLUL = set de reguli ce stabilesc structura mesajului si sincronizarea comunicatiilor.

INTERFATA PARALELA = in transferul paralel, informatia este transmisa pe mai multe linii, cuvintele fiind transmise succesiv.  
INTERFATA SERIALA = informatia este transmisa bit dupa bit, pe mai putine fire. Esantionare cu frecventa diferita rezulta in valori diferite ale datelor.

VERIFICAREA CORECTITUDINII = atasarea bitului de paritate.

BUFFER DE DATE = utilizat pentru marirea eficientei transferului de date permitand procesorului sa fie liber de sarcina de transfer pe perioade mai mari.

**TRANSFERUL DE DATE**

TRANSFERUL DE DATE = Programat, prin intreruperi, prin acces direct la memorie (DMA)

TRANSFER PROGRAMAT = citirea valorilor de la porturi prin cod

TRANSFER PRIN INTRERUPERI = dispozitivul care cere o intrerupere va lansa un semnal de intrerupere INT. Daca acesta este validat cu semnalul INTE, instructiunea curenta se termina si se salveaza continutul registrului PC si a registrului de stare. Procesorul generaza semnalul de acceptare INTA si asteapta ca dispozitivul IO sa puna datele pe magistrala. Daca nu se valideaza intreruperea, aceasta se ignora.

TRANSFER PRIN DMA = dispozitivul IO solicita transfer DMA prin semnalul HOLD, care suspenda activitatea procesorului prin trecerea magistralelor in inalta impedanta. Cand procesorul accepta aceasta suspendare, genereaza un semnal HOLDA (HOLD ack) si trece magistralele in inalta impendata.

SISTEME DE INTRERUPERI LA MICROCONTROLLERE = acestea pot fi externe, prin pini, sau interne prin interfetele integrate ( convertorul ADC, timer, interfata seriala). Cererile de intrerupere pot fi mascate de un registru de masti stabilit anterior in microcontroler.

**MAGISTRALE**

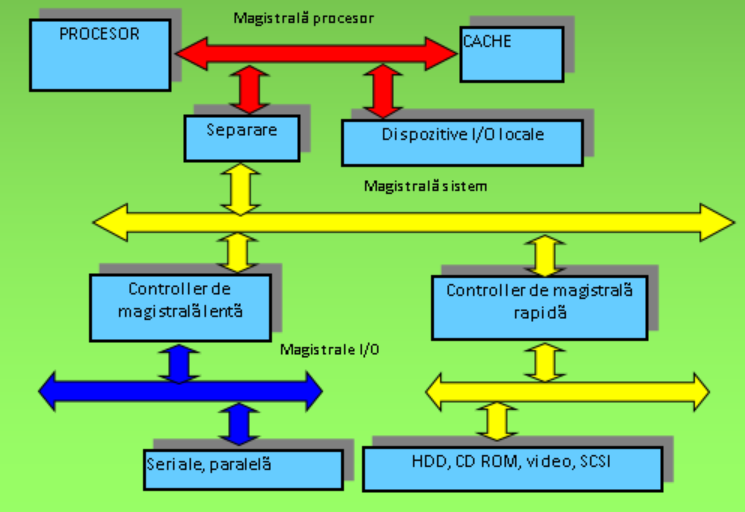
MAGISTRALA = subsistem cu functia de comutator universal bidirectional utilizat pentru transferul de date in interiorul sistemului.

MAGISTRALELE POT FI

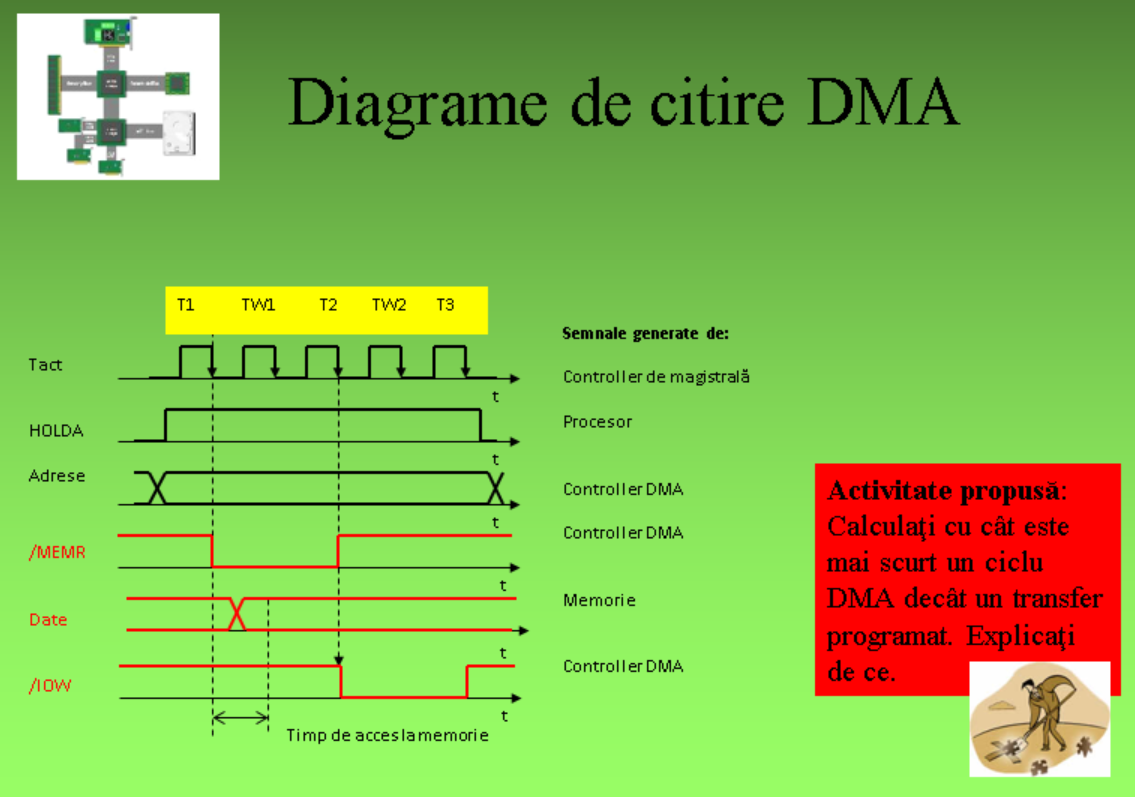
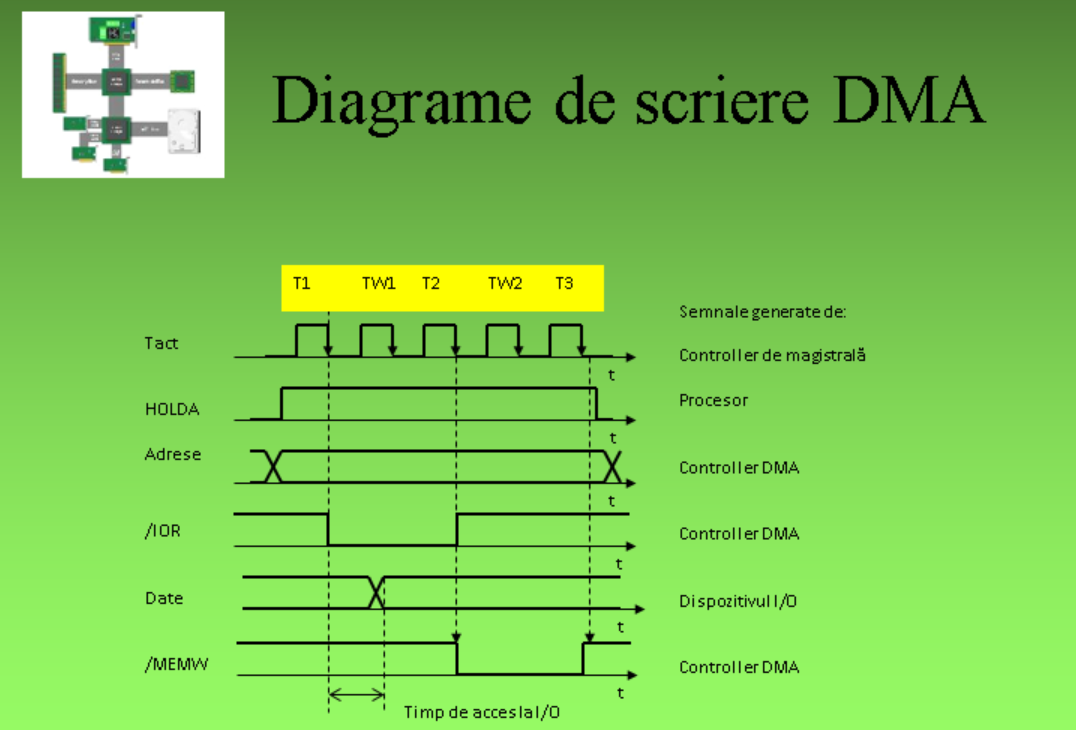
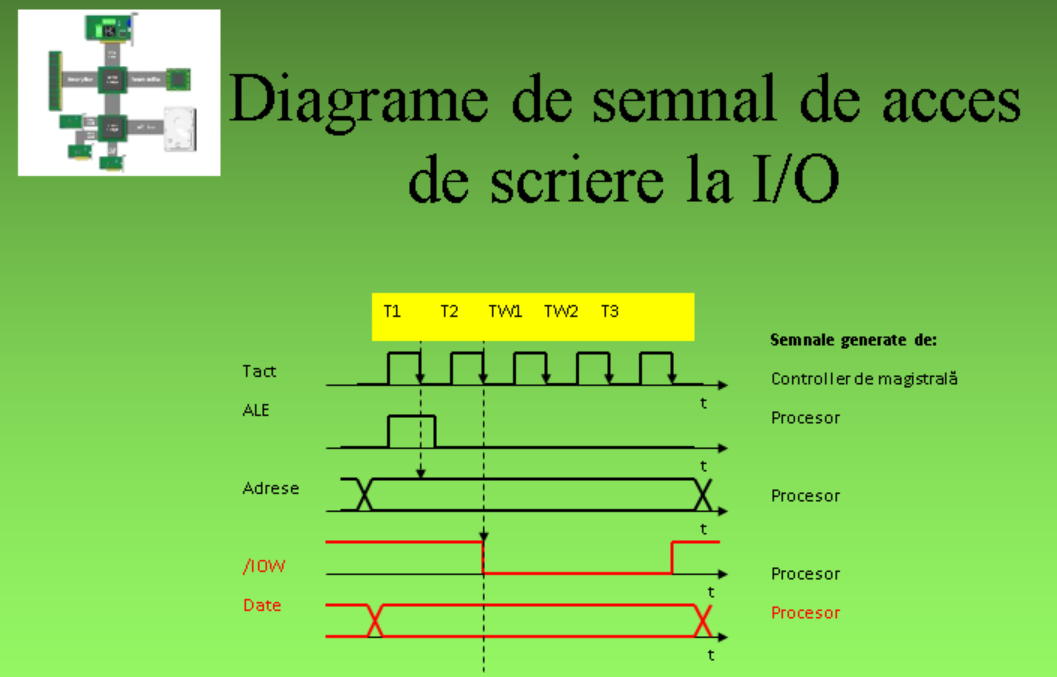
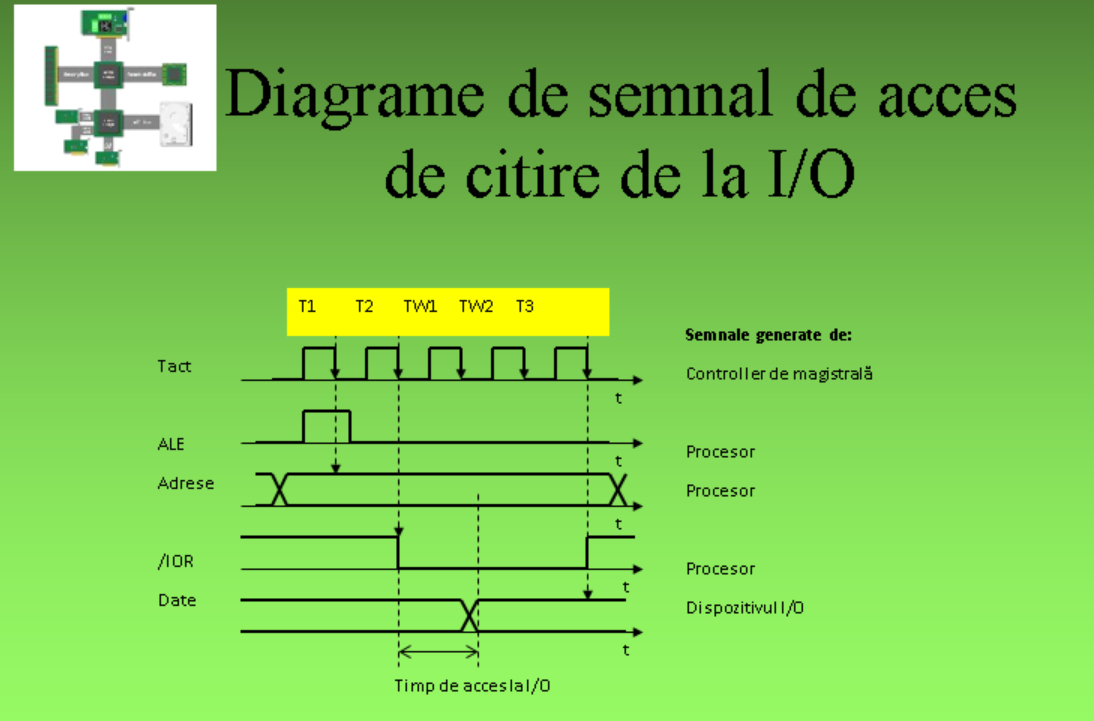
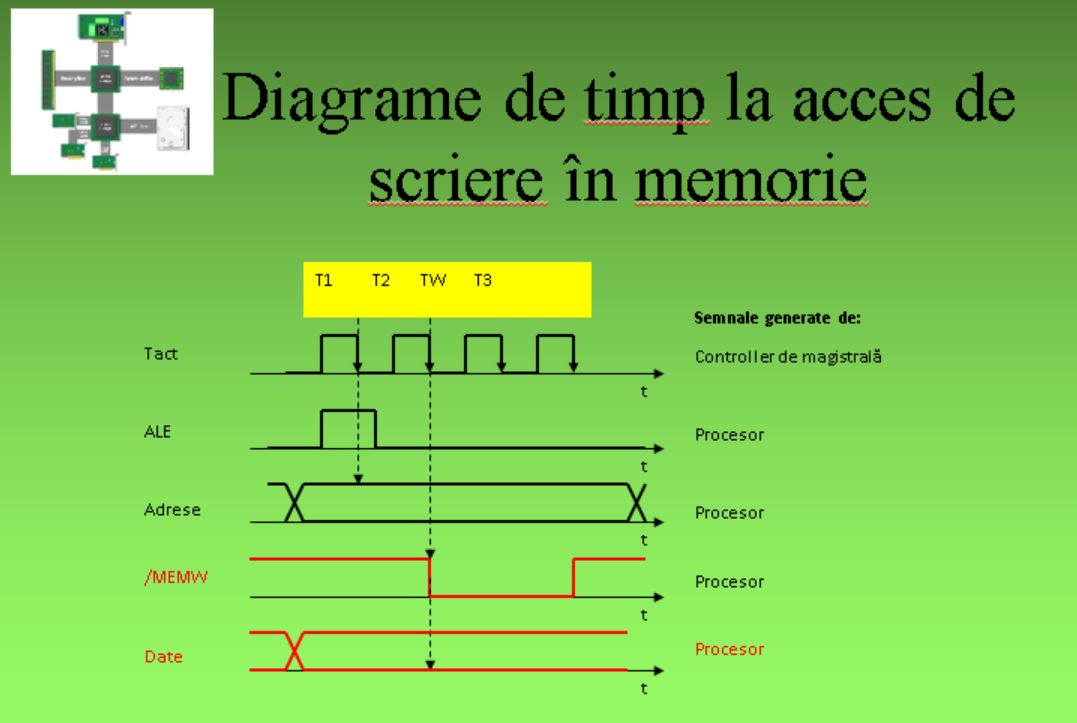
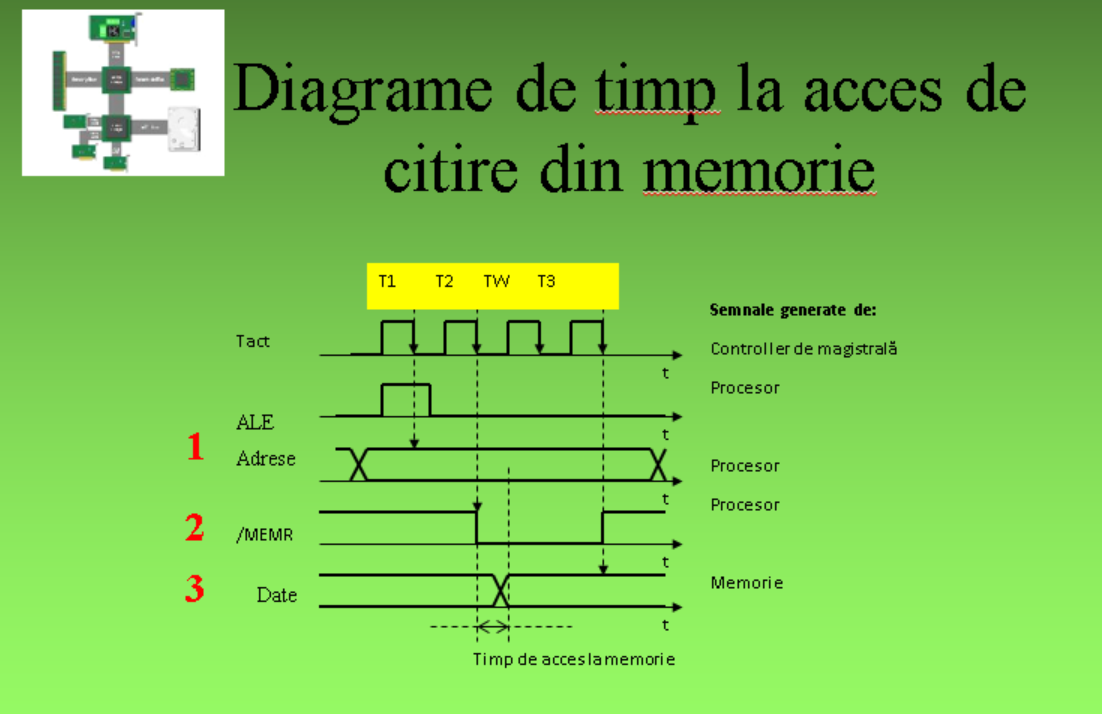
* SINCRONE, unde exista o linie de tact, unde datele se transmit in nr intreg de tacte, fiind cea mai simpla metoda si cea mai raspandita la microcontrolere
* ASINCRONE, transferul poate dura oricat insa este nevoie de un protocol de dialog si un mecanism de supraveghere pentru a nu bloca sistemul la erori.

IERARHIZAREA MAGISTRALELOR

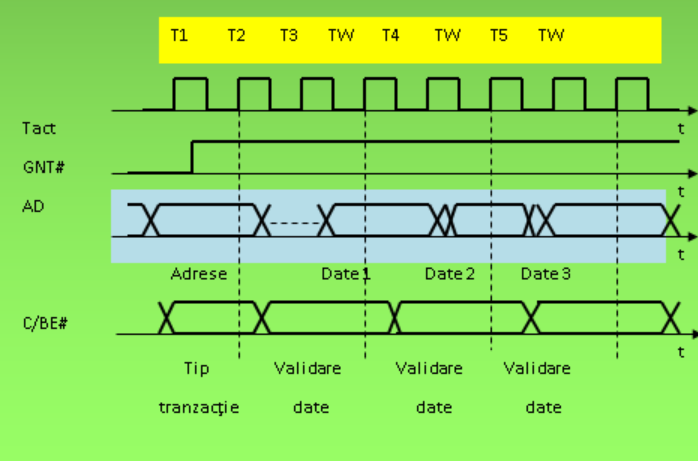
* MAGISTRALA PROCESOR
* MAGISTRALA SISTEM
* MAGISTRALA IO



SEMNALUL ALE SE FOLOSESTE DOAR IN CAZUL MAGISTRALELOR MULTIPLEXATE



MAGISTRALA PCI = admite 12 tipuri de tranzactii, de ex cu memoria, cu IO, de configurare etc. Liniile de date si adrese sunt multiplexate (AD) pt a reduce nr de linii. Un dispozitiv MASTER cuplat poate solicita semnalul REQ# si cand magistrala este libera, controllerul va raspunde cu GNT#. La primul tact pe liniile AD se pune adresa si pe C/BE# tipul tranzactiei. Apoi pe AD urmeaza datele, cuvant dupa cuvant. Dupa fiecare tact, urmeaza unul de wait pentru a permite sicronizarea datelor de pe magistrala si pe C/BE# se pun cuvintele de validare pt date.



LIMITARI PCI = apar efecte perturbatoare cum ar fi reflexia si diafonia in transmisii de date la viteze mari pe mai multe linii apropiate intre ele. Ca solutie exista latirea traseelor, introducerea traseelor de masa intre cele de semnal nu este posibila deoarece se doreste ca marimea componentelor sa fie din ce in ce mai mica. O alta solutie ar fi codificarea datelor pe liniile adiacente pentru a elimina diafonia.

PCI EXPRESS = transmisie punct la punct, sincrona, cu cadre de date si refacerea tactului din date, cu codare de grup 8b/10b. Prezinta un controller de magistrala ce are functia de switch intre master si slave. Acesta prezinta gruprui de linii de comunicatii seriale numite canale ( 2 linii de transmisie seriala unidirectionala, fiecare fiind cu transmisie diferentiala )

**INTERFATA PARALELA**

INTERFETE PARALELE

* PROGRAMABILE
* NEPROGRAMABILE

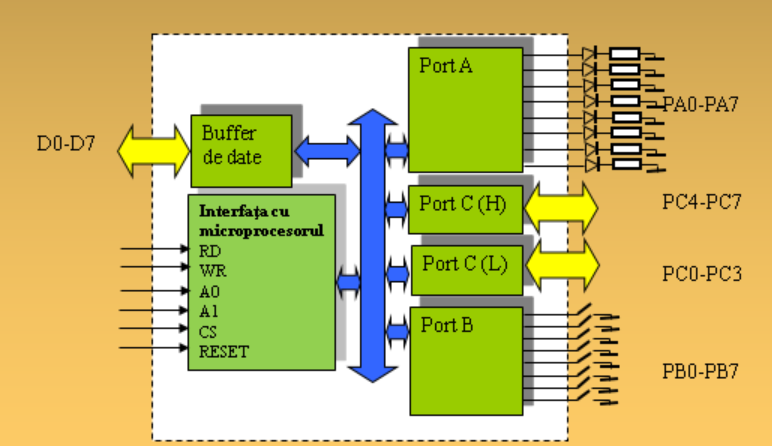
INTERFETE PARALELE PROGRAMABILE

SEMNALE :

* + RD = executa un ciclu de citire de la port/memorie
  + WR = executa un ciclu de scriere la port/memorie
  + A0/A1 = selectarea registrelor interne
  + RESET = resetarea circuitului
  + CS = selectarea circuitului
  + D0-7 = magistrala de date a gazdei, 8 linii bidirectionale
  + PORT A, B, C low, C high

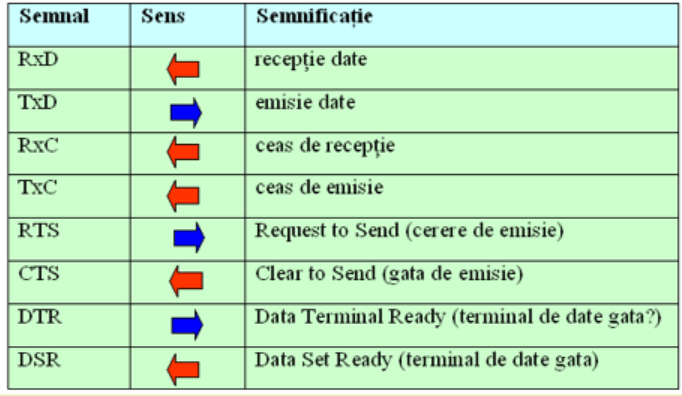
MOD DE LUCRU:

* + MODUL 0 = mod de intrare, iesire. Asigura functionarea porturilor A, B ca porturi de 8 biti. Acestea pot fi programate ca porturi de intrare sau iesire si mentin sensul pana la reprogramare.
  + MODUL 1 = asigura implementarea unui protocol de transfer prin portul A si B asistat de portul C pentru semnale de comanda.
  + MODUL 2 = portul A este port de date bidirectionale, astat de portul C. Portul B este in modul 0.



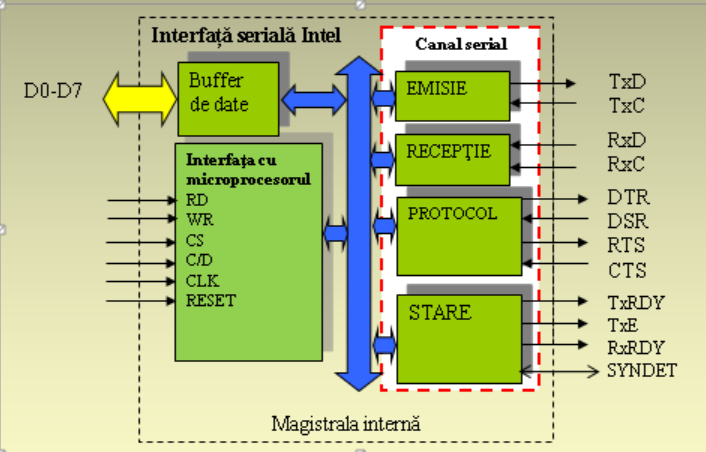
**INTERFATA SERIALA**

STANDARDUL RS232



LEGATURI RS232

* 7 FIRE = PROTOCOL HARDWARE
* 3 FIRE = PROTOCOL SOFTWARE



SEMNALE

* RD = citire pin/memorie, conectat la IOR
* WR = scriere pin/memorie, conectat la IOW
* C/D = 0 = trasfer date, 1 = transfer comenzi
* CLK = ceas magistrala
* RESET
* CS = selectarea circuitului
* D0-7 = magistrala de date
* TxRDY = pregatit sa preia datele de la magistrala de date
* RxRDY = receptorul a receptionat date care vor fi tramise pe magistrala de date
* TxE = transmitator gol
* SYNDET = semnal bidirectional, detectarea sincronizarii

FUNCTIONARE

* SINCRON/ASINCRON
* SERIALIZARE/DESERIALIZARE

Circuitul face conversia paralel-serial si serial-paralel. Un cuvant transmis paralel este receptionat pe magistrala de date si apoi este convertit asincron, adaugandu-i-se semnalele de START, STOP si PARITATE prin tactul semnalului TxC, fiind apoi transmis pe linia TxD prin protocol hardware sau software. Dupa transmisie, semnalul TxRDY este activat pentru ca procesorul sa transmita alt cuvant.

La receptie asicrona, informatia seriala de pe linia RxD este citita folosind tactul RxC incepand cu semnalul de Start, verificand paritatea si eliminand semnalele de Start si Stop si face conversia in paralel. Apoi se activeaza semnalul RxRDY pentru ca procesorul sa citeasca datele de pe magistrala de date.

In mod sincron, circuitul lucreaza cu sincronizare interna sau externa.

Sincronizarea interna reprezinta compararea datelor cu un cuvant de 8 biti standardizat pentru sincronizare, fiind folosit la cadrele de date codate ASCII marcand inceputul unui cadru de date.

Sincronizarea externa reprezinta recunoasterea cuvantului de sincronizare printr-un circuit extern. Recunoasterea acestui cuvant rezulta in activarea semnalului SYNDET si circuitul incepe sa citeasca date de pe RxD care urmaresc procesul de deserializare.

Cuvantul de sincronizare poate avea un octet sau 2.

PROGRAMARE

ASINCRONA:

* + CUVANT PENTRU MOD ASINCRON
  + CUVANT PORNIRE OPERARE

SINCRONA:

* + CUVANT PENTRU MOD SINCRON
  + CUVANT PENTRU SINCRONIZARE
  + CUVANT PENTRU PORNIRE OPERARE

Nivelele de tensiune RS232 sunt intre 3 si 15V, iar circuitul de interfata paralela suporta tensiuni intre 0 si 5V, fiind necesar un circuit de conversie MAX232.

**CODIFICARI**

NRZ

* 0 SE CODIFICA CU 0V
* 1 SE CODIFICA CU 5V

NRZI

* 1 SE CODIFICA CU O TRANZITIE DE FRONT

MANCHESTER

* 1 SE CODIFICA CU FRONT CRESCATOR
* 0 SE CODIFICA CU FRONT DESCRESCATOR

FM

* 1 SE CODIFICA CU UN IMPULS
* 0 SE CODIFICA CU LIPSA UNUI IMPULS

INSERARE DE BITI

* ESTE NECESAR LA NRZ SAU NRZI CAND APARE UN SEMNAL FARA TRANZITII PENTRU O ANUMITA PERIOADA DE TIMP, SE ADAUGA UN BIT SUPLIMENTAR
* BITII SE INSEREAZA INAINTE SI DUPA FIECARE CUVANT TRANSMIS

STRUCTURA UNU CUVANT SERIAL ASINCRON

* START 1 BIT CU VALOAREA 0
* 5,6,7,8 BITI DE DATE CODATI NRZ
* BIT DE PARITATE
* 1, 1+1/2, 2 BITI DE STOP CU VALOARE 1

STRUCTURA UNUI CADRU DE DATE SINCRON

* SYNC
* ANTET
  + IDENTIFICATOR
  + ADRESA DESTINATIE
  + ADRESA SURSA
* DATE
* CRC