

1.Noțiuni introductive



Cuprins

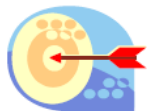
Cuprins modul

- 1.1. Definiții. Istorie: prima comunicație cu protocol
- 1.2. Interfețe paralele și seriale
- 1.3. Verificarea corectitudinii datelor transmise cu bit de paritate
- 1.4. Rolul unui buffer în transferul de date
- 1.5. Structura cursului



Introducere

Modulul “**Noțiuni introductive**“ prezintă definițiile noțiunilor principale din acest curs, apoi descrie sumar interfețele paralele și seriale, cu accent pe importanța tactului. Este prezentată pe scurt și verificarea corectitudinii datelor la transfer și rolul unui buffer de date în fluidizarea transferului. Scopul acestui modul este să împrăspăteze cunoștințele anterioare ale studenților și să ofere o imagine generală a cursului.



Obiective

După parcurgerea acestui modul studenții vor cunoaște definițiile noțiunilor de bază din curs, vor înțelege ce își propune acest curs și își vor verifica cunoștințele anterioare din domeniu, de exemplu:

- Noțiuni generale de interfețe paralele și seriale;
- Importanța tactului la transmisia datelor;
- Verificarea corectitudinii transferului de date;
- Rolul unui buffer în transferul de date

Obiective specifice:

- 1.Învățarea principiilor fundamentale de transfer de date;
- 2.Învățarea noțiunii de protocol cu aplicare la transferul de date prin interfețe;
- 3.Cunoasterea unor tipuri de transmisii seriale si paralele.



Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

1.1. Definiții. Istorie: prima comunicație cu protocol.

În domeniul calculatoarelor **interfața** este punctul de întâlnire a unității centrale cu dispozitivele periferice cu scopul transferului de date. Interfețele pot fi hardware sau software, iar în cazul interfețelor software situarea punctului de întâlnire este între două programe.

Interfețele pot fi bidirecționale în care datele circulă în ambele direcții (ex. interfața cu hard discul) și unidirecționale în care datele circulă într-o singură direcție (ex. interfața cu mouse-ul). Interfețele pot fi punct la punct ceea ce înseamnă că pot fi conectate pe o linie doar un sistem gazdă și un periferic (ex. interfața PS2 cu tastatura) sau multipunct în care la un sistem gazdă se pot lega mai multe periferice (ex. USB).

Pentru ca două sisteme de calcul să comunice între ele este nevoie ca formatul mesajului să fie bine stabilit ca să fie recunoscut de ambele sisteme. **Protocolul** conține un set de reguli care stabilesc structura mesajului și asigură sincronizarea comunicației. De regulă protocoalele sunt standardizate de autorități în domeniul respectiv.

Echipamentul periferic este un dispozitiv conectat la un sistem de calcul gazdă care extinde funcționalitatea sistemului gazdă dar nu face parte din nucleul de calcul. Echipamentul periferic este deseori, dar nu întotdeauna dependent de sistemul gazdă.

Primul sistem de comunicații cu protocol a fost telegraful hidraulic al lui Aeneas, construit în secolul 4 BC, figura 1.1:

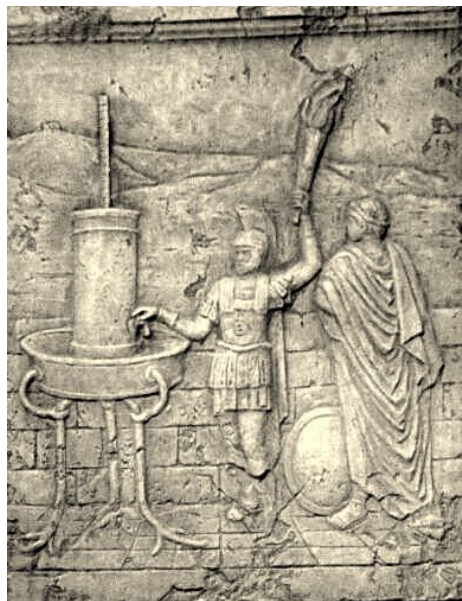


Fig. 1.1. Telegraful hidraulic al lui Aeneas (sursa http://en.wikipedia.org/wiki/Hydraulic_telegraph)

Recipientul se umple cu apă, peste care se așează un plutitor care are fixată o tijă verticală. Pe tijă există gradații cu însemnări relevante pentru purtarea războiului (vreau cavalerie, artilerie, etc.). Dacă s-a dorit începerea unei transmisiuni, soldatul operator ridică torța. Operatorul de la recepție, observând cererea, indică că este pregătit pentru recepție ridicând și el torța și rămâne foarte atent la punctul de transmisie. Operatorul de la transmisie coboară torța de semnalizare și deschide dopul de pe recipient, ceea ce făcea (sincron) și cel de la recepție. Apa începe să coboare identic în ambele recipiente cu viteză egală. La nivelul superior al vasului apăreau rând pe rând inscripțiile mesaj de pe tijă. În momentul în care apărea cea dorită a fi transmisă, operatorul ridică torța, și pune dopul. La fel făcea și cel de la recepție. În acest moment putea citi și el mesajul. Pentru alt mesaj, recipientele trebuiau reumplute. Se pare că în istorie acesta a fost primul sistem sincron.

Într-o schemă bloc în care este figurat un sistem de calcul, figura 1.2. se pot observa interfețele, ca puncte de întâlnire a datelor de intrare și ieșire, care pot fi analogice sau digitale.

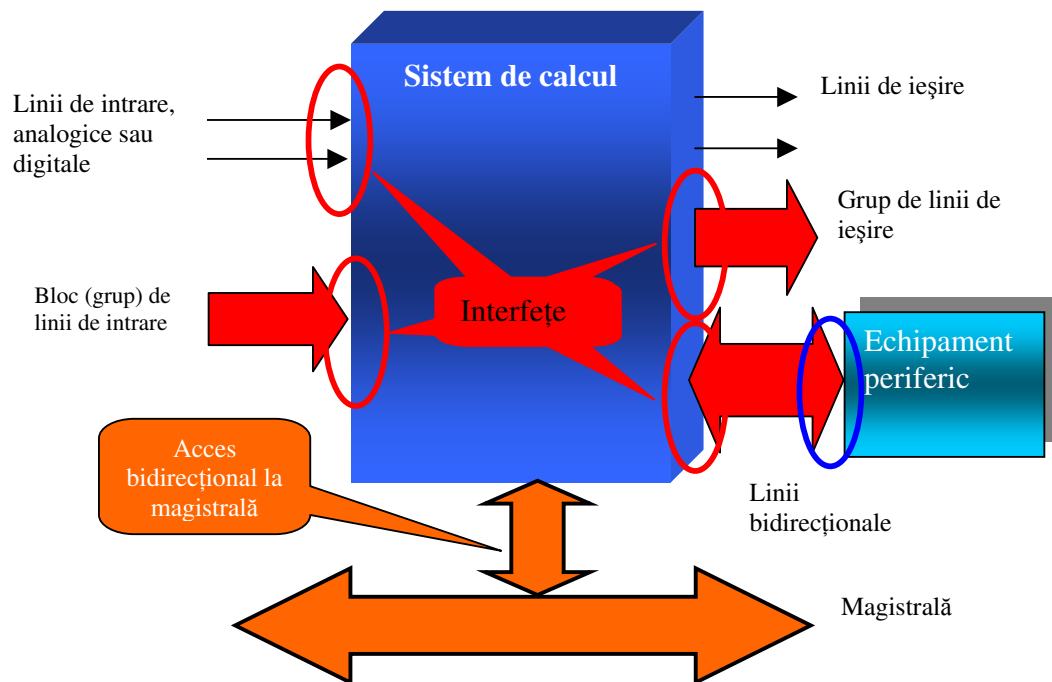


Fig. 1.2. Interfețe marcate pe o schemă bloc

Mărimile de intrare și de ieșire pot fi analogice sau digitale. Un semnal analogic are o variație continuă în timp. Exemple: temperatura, umiditatea, iluminarea etc. Un semnal digital are o variație discretă în timp, el poate lua doar 2 valori. Exemplu: comanda dată de un întrerupător către un bec.

Sistemul de calcul procesează digital, prin urmare există interfețe de conversie a semnalelor analogice în semnale digitale numite convertoare analog digitale și circuite de conversie inversă numite convertoare digital analogice. Exemplu: placa de sunet a unui calculator conține un convertor analog digital la intrarea de la microfon și un convertor digital analogic la ieșirea de boxe. Aceste convertoare sunt importante pentru că în natură predomină mărimile analogice și pentru ca acestea să fie măsurate și prelucrate cu calculatorul digital este nevoie de conversia analog digitală și digital analogică.

Dacă la portul cu linii bidirecționale a sistemului de calcul din figura 1.2 se conectează un echipament periferic EP acesta conține o interfață de conectare care trebuie să fie compatibilă cu cea a sistemului gazdă.

1.2. Interfețe paralele și seriale

Un proces de comunicații de date necesită cel puțin 5 elemente, figura 1.3:

- Transmițător;
- Mesaj;
- Interfață binară (digitală);
- Canal de comunicație;
- Receptor.

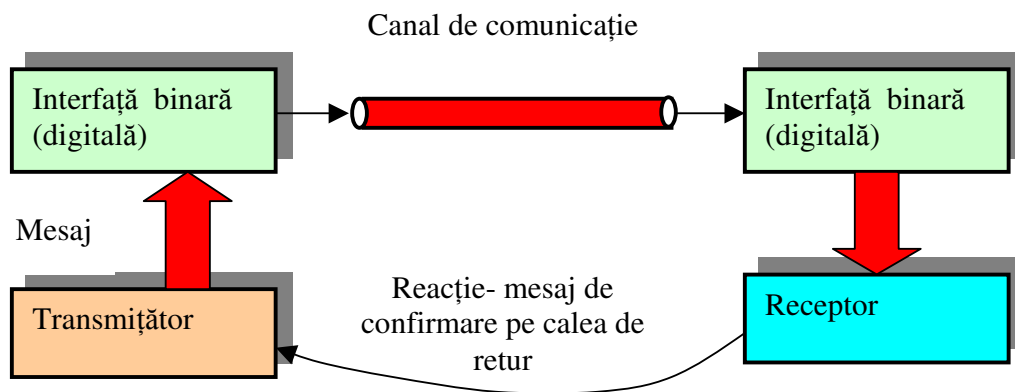


Figura 1.3. Procesul de comunicații de date

Pentru transmisia unui mesaj, fiecărui grup de biți trebuie să îi corespundă un caracter (literă, cifră, semn special). Cel mai cunoscut este codul ASCII (American Standard Code for Information Interchange).

La transferul paralel informația este transmisă pe mai multe linii (8, 16, 32, 64, 128 ...), cu un număr de biți transmis la un impuls de tact egal cu numărul de linii, cuvintele fiind transmise succesiv, figura 1.4.

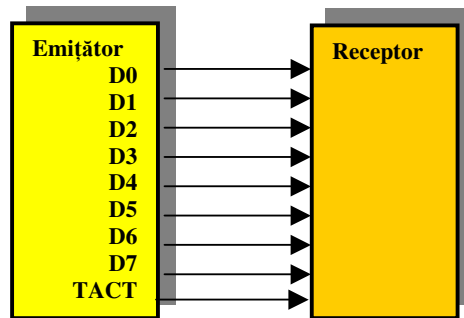


Figura 1.4. Schema bloc generală a interfeței paralele pe 8 biți

Diagrama de timp comprimată a transferului este dată în figura 1.5. Semnalul de tact are rolul de stabili momentul exact al citirii datelor, pentru a evita situațiile în care datele sunt citite când încă nu sunt stabile pe linie. În anumite interfețe acest semnal poate lipsi dar viteza de transfer este mai mică în acest caz.

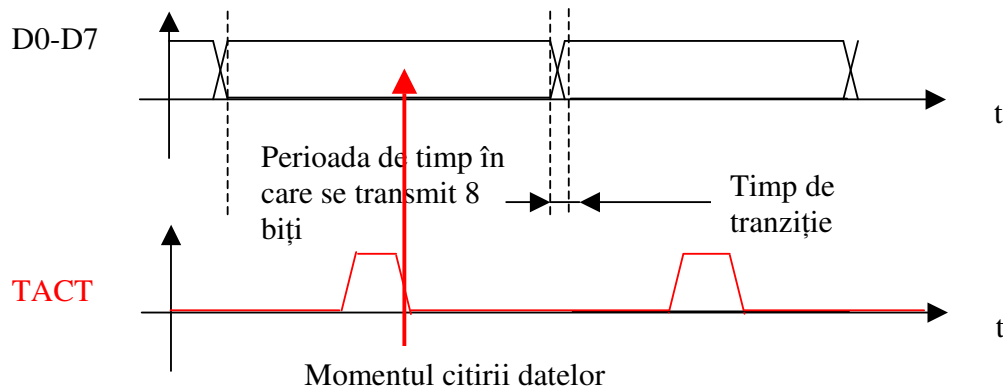


Figura 1.5. Diagrama de timp a transferului

În acest curs în alte module vor fi prezentate transferurile paralele și exemple de protocoale prin porturile paralele de I/O de uz general, dar și alte câteva aplicații cum ar fi de exemplu circuitul de conversie USB- paralel FTDI. Magistralele și interfețele

paralele sunt în scădere de piață, fiind înlocuite de cele seriale, de aceea și ponderea lor în acest curs este mai scăzută.

La transferul serial informația este transmisă bit după bit, pe mai puține fire (minimum 2 fire, dintre care unul de referință, masa electrică), figura 1.6. Pentru transmisia datelor acestea trebuie codificate. În alt mod vor vor fi descrise codificările NRZ, Manchester, 8B10B etc. Prețul mic al interfețelor seriale au dus la răspândirea acestor interfețe în majoritatea aplicațiilor.

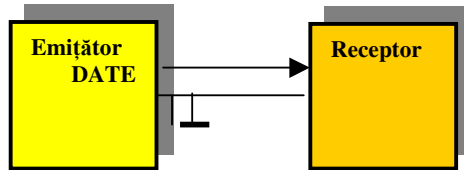


Figura 1.6. Schema bloc generală a interfeței seriale pe un fir (linia de masă nu este considerată)

În perioada actuală numărul, viteza și calitatea interfețelor seriale cresc, cucerind din ce în ce mai multe segmente de piață.

Tactul în transmisiile seriale are o importanță majoră. Dacă datele seriale vin la receptor pe un singur fir receptorul le poate citi / eșantiona cu o anumită frecvență și să obțină datele recepționate ca în figura 1.7:

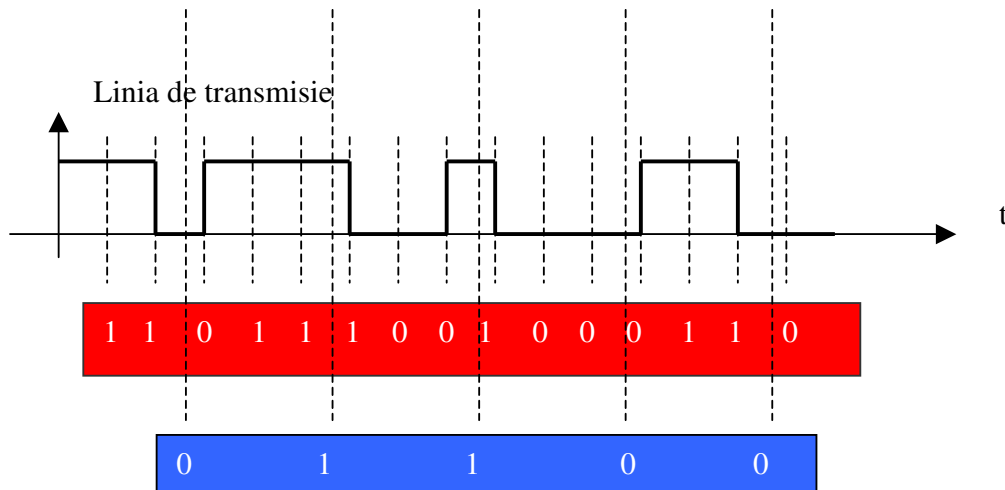


Figura 1.7. importanța eșantionării corecte

Dacă datele sunt eșantionate cu o anumită frecvență (pe fond roșu) se obține un anumit șir de date iar dacă datele sunt eșantionate cu altă frecvență (pe fond albastru) se obține un șir de date diferit. Acest lucru arată importanța ca datele să fie recepționate cu același tact cu care au fost trimise. Este nevoie **ca tactul de transmisie să fie cunoscut de receptor.**

1.3. Verificarea corectitudinii datelor transmise cu bit de paritate

Teorema a 2-a a lui Shannon din teoria transmiterii informației pentru canale afectate de perturbații afirmă că: *Pentru o sursă de informație cu debitul de R bps (biți pe secundă) și un canal de capacitatea C bps, dacă $R < C$ există un cod având cuvintele de lungime n astfel încât probabilitatea de eroare la decodare este oricât de mică.* La un cuvânt binar de forma $A_2 = a_7a_6a_5a_4a_3a_2a_1a_0$ i se atașează la emițătorul de informație un bit numit de paritate a_p care poate fi egal cu 1 (paritate pară) dacă numărul de biți din cuvânt este par și 0 dacă numărul de biți din cuvânt este impar (sau invers, cu condiția ca la emițător și la receptor convenția să fie aceeași). Controlul de paritate poate detecta o eroare dar nu o poate corecta. Se poate solicita o repetare a mesajului prin linia de retur. Controlul de paritate se folosește la transmisii fără perturbații majore, de exemplu la debite mari de informații pe distanțe mici (lucrul procesorului cu memoria de exemplu) sau distanțe mari și debite mici (transmisia serială RS232).

La receptor se generează un bit de paritate după același algoritm și se compară cu bitul de paritate transmis. Dacă acești biți sunt egali transmisia a fost corectă, dacă nu transmisia a fost eronată. Schema bloc a transferului cu verificarea bitului de paritate este dată în figura 1.8.

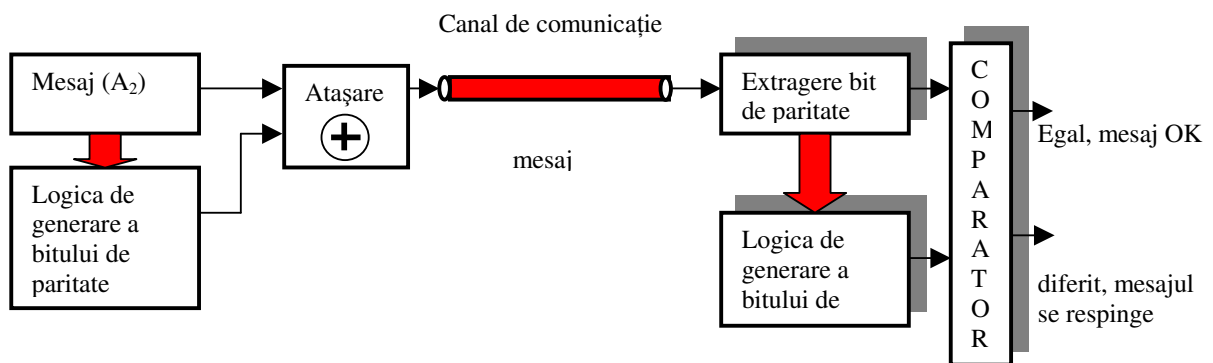


Figura 1.8. Schema bloc a transferului cu verificarea bitului de paritate

1.4. Rolul unui buffer în transferul de date

Viteza unui sistem de calcul depinde de 2 componente:

1. viteza cu care procesorul execută operațiile;
2. viteza cu care circulă datele în sistem (viteza de I/O, viteza de acces la memorie).

Dacă procesorul este forțat să rămână inactiv perioade lungi de timp deoarece sistemul de I/O nu poate transfera datele nu îi sunt furnizate, sistemul este limitat I/O. Ideal este ca cele două viteze să fie comparabile. În figura 1.9 se poate vedea o diagramă a fluxului de date.

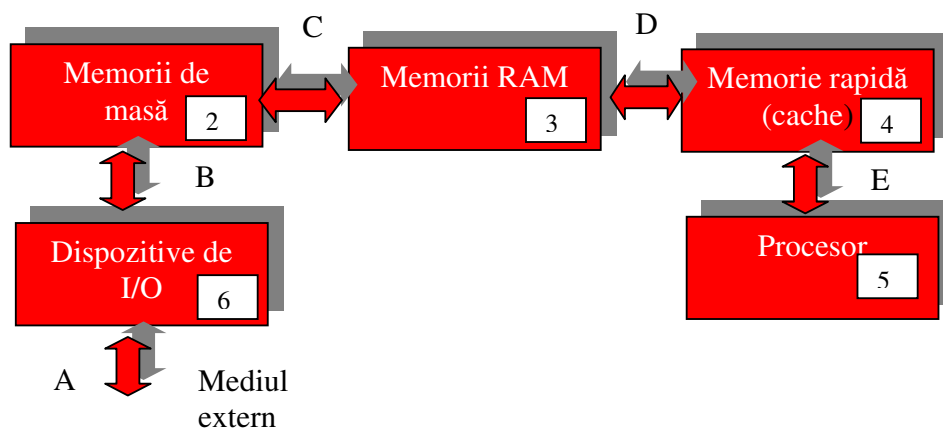


Figura 1.9. Fluxul de date într-un sistem de calcul

Pentru mărirea eficienței se iau anumite măsuri, ca de exemplu:

1. Pentru a elibera procesorul de sarcina supravegherii transferului, pe calea C se poate face transfer DMA;
2. Transferul pe calea A este lent când schimbul de date cade în sarcina operatorului. Lucrul în întreruperi (cu tastatura, cu imprimanta, cu rețeaua) duce la optimizarea încărcării sistemului;
4. Eficiența este maximă când capacitatea blocurilor 1,2,3 de a furniza date este egală cu capacitatea blocului 5 de a le prelucra.
5. Memoria rapidă conține date și secvențe de program. Lucrul între blocurile 4 și 5 este foarte rapid. Dar există momente în care este nevoie de informația din memoria de bază (mai ales când apar instrucțiuni de salt). În aceste momente este nevoie să se schimbe întregul conținut al memoriei rapide.

Un buffer de date situat între procesor și echipamentul periferic are un rol important în mărirea eficienței transferului de date, pentru că permite procesorului să fie liber de

sarcina de a transfera date perioade mai lungi de timp. Principiul transferului este dat în figura 1.10:

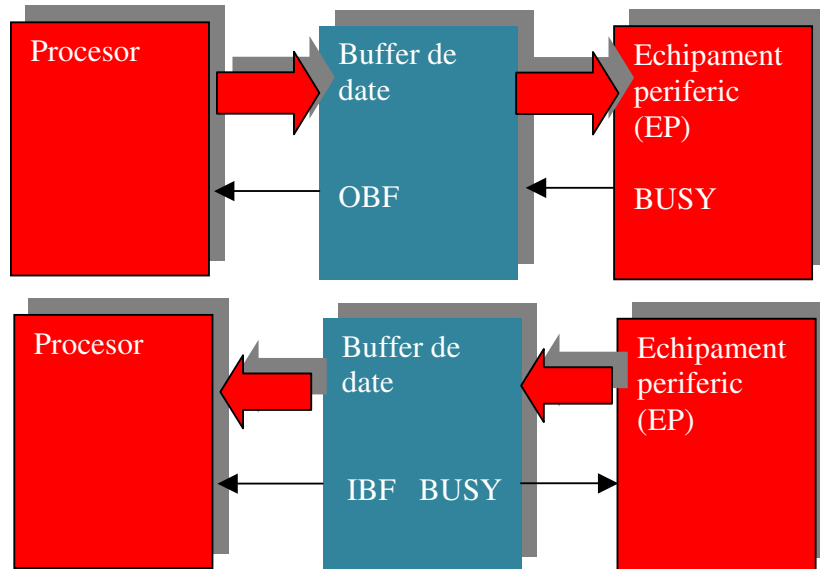


Figura 1.10. Rolul unui buffer de date în eficiența transferului

În cazul unui transfer de date de la procesor la EP (sus) datele se încarcă rapid în buffer (la viteza procesorului) până la umplerea bufferului, semnalizată cu semnalul OBF (Output Buffer Full). Datele sunt transferate lent la EP, primirea lor fiind acceptată dacă semnalul BUSY este inactiv. Invers, datele sunt încărcate în buffer până când acesta se umple, semnalizând cu BUSY. După umplerea lentă a bufferului procesorul este anunțat cu semnalul IBF (Input Buffer Full) și preia rapid datele. În anumite aplicații bufferul trebuie să asigure un transfer de date cu viteza constantă cu EP, cum este de exemplu la scrierea pe medii optice.

1.5. Structura cursului

Diversitatea de interfețe și protocoale este foarte mare. Apar multe tipuri de interfețe noi dintre care unele se răspândesc iar altele dispar în timp. Cursul de Interfațare și Protocoale se focalizează pe interfețe și protocoale la nivel fizic, apropiat de microprocesorul sau microcontrollerul gazdă. Cursul conține două părți, prima conține noțiunile de bază ale conectării pe magistrală, pe un port paralel și la o interfață asincronă simplă. A doua parte prezintă sumar câteva tipuri de interfețe și protocoalele specifice. Locul fiecărui modul de studiu în schema generală a interfațării este dat în figura 1.11.

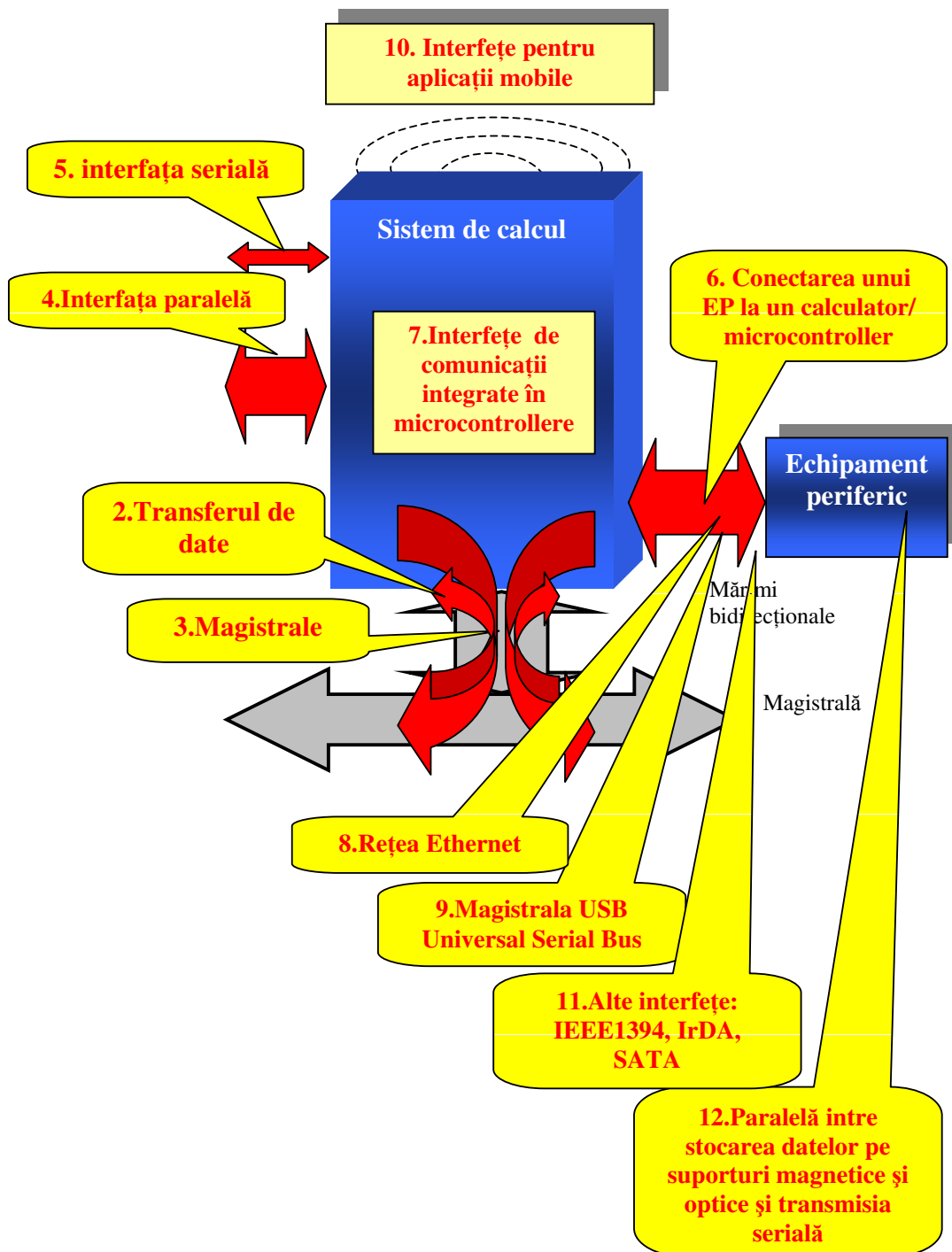


Figura 1.11. Modulele cursului de Interfațare și Protocoale și situarea lor în structura unui sistem de calcul



Rezumat

Modulul “**Noțiuni introductive**” prezintă definițiile noțiunilor principale din acest curs și prima realizare în istorie a unui transfer de date cu protocol. Se descriu pe scurt interfețele paralele și seriale, cu accent pe importanța tactului de transmisie. Este prezentată pe scurt verificarea corectitudinii datelor la transfer și o diagramă a fluxului de date într-un sistem de calcul, considerate cunoscute. Este descris rolul unui buffer de date situat între procesor și echipamentul periferic în fluidizarea transferului, în ambele sensuri de transfer. Ultima parte a modulului anticipează celelalte module, stabilind locul lor pe o schemă bloc sugestivă.



Bibliografie

1. M. Romanca, P. Ogrutan, *Sisteme cu calculator incorporat. Aplicații cu microcontrollere*, Editura Universitatii Transilvania Brașov, 2011, pag. 1-4
online la:
<http://vega.unitbv.ro/~ogrutan/Microcontrollere2011/1-introducere-interfete%20paralele%20si%20seriale.pdf>
2. C. Gerigan, P. Ogrutan, *Tehnici de interfațare*, Ed. Transilvania Brașov, 2000, 315p., ISBN 973-9474-94-2, pag. 2-11, online la:
<http://vega.unitbv.ro/~ogrutan/ti/cap1.pdf>
3. http://en.wikipedia.org/wiki/Interface_%28computing%29
4. http://en.wikipedia.org/wiki/Communications_protocol
5. <https://en.wikipedia.org/wiki/Peripheral>
6. http://en.wikipedia.org/wiki/Hydraulic_telegraph

Test de autoevaluare



1. Rolul transmisiei tactului la o interfață paralelă este:

R

- (a) să asigure o viteză cât mai mare de transfer
- (b) de a stabili momentul exact al citirii datelor
- (c) de a asigura citirea datelor când nu sunt stabile
- (d) asigurarea transmisiei de 2 octeți

I. vezi pagina 5

2. Verificarea bitului de paritate are următoarele avantaje:

R

- (a) asigură corectarea erorilor
- (b) asigură detectarea a două erori în același cuvânt
- (c) asigură verificarea corectitudinii transferului
- (d) asigură la debite mici distanțe mari

I. vezi pagina 7

3. Pentru mărirea vitezei fluxului de date într-un sistem de calcul se poate face:

R

- (a) verificarea corectitudinii datelor
- (b) transfer DMA între memorie și un hard disc
- (c) transfer prin întreruperi cu tastatura
- (d) alimentarea cu o tensiune mai mare

I. vezi pagina 8

4. În aplicațiile în care un buffer de date are rolul de a asigura un debit constant de date transferate cu EP, acesta trebuie să fie:

R

- (a) date spre EP- buffer cât mai plin
- (b) date de la EP- buffer cât mai plin
- (c) date spre EP- buffer cât mai gol
- (d) date de la EP- buffer cât mai gol

I. vezi pagina 9

R

Răspunsuri corecte:

1. b, vezi definiția rolului tactului de la pagina 5
2. c și d, vezi caracteristicile verificării cu bit de paritate de la pagina 7
3. b și c, vezi schema fluxului de date de la pagina 8. Răspunsul d este o glumă.
4. b și d. Pentru a micșora riscul ca să nu fie date disponibile pentru EP, la scriere în EP bufferul trebuie să fie cât mai plin, iar la citire cât mai gol.