

## 5a. Interfețe seriale. Codificarea datelor



### Cuprins modul

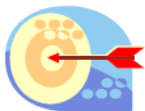
- 5a.1. Tactul în transmisiile seriale
- 5a.2. Codarea datelor
- 5a.3. Transmisii seriale asincrone și sincrone

### Cuprins



### Introducere

Modulul “**Interfețe seriale. Codificarea datelor**“ continuă prezentarea interfețelor seriale care în modulul 1 au fost descrise sumar, cu accent pe importanța tactului. Sunt prezentate câteva moduri de codificare mai des întâlnite și se compară din punctul de vedere a eficienței codării. Se introduce noțiunea de cod autosincronizabil și se explică importanța lui la codarea în transmisiile de date actuale. Se descriu apoi transmisiile seriale sincrone și asincrone prin explicarea structurii unui cuvânt serial asincron și a unui cadru de date cu transmisie sincronă.



### Obiective

După parcurgerea acestui modul studenții vor cunoaște cele mai uzuale codificări binare, vor înțelege domeniul lor de aplicabilitate și le vor putea compara din punctul de vedere al eficienței codării și a caracteristicii de autosincronizare. După descrierea transmisiei asincrone și sincrone studenții vor cunoaște:

- Structura unui cuvânt serial asincron;
- Structura unui cadru de date serial sincron.

Obiective specifice:

1. Învățarea principiilor fundamentale de transfer de date;
2. Învățarea unor tipuri de codare digitală. Înțelegerea caracteristicilor cerute codării;
3. Cunoașterea unor tipuri de transmisiile seriale și interfețe seriale ca structură.



### Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

### 5a.1.Tactul în transmisiile seriale

În primul modul de studiu a fost demonstrată importanța cunoașterii tactului cu care s-au emis date la receptor. Pentru a se asigura la receptor tactul corect de recepție există mai multe variante, prezentate în figura 5.1.

1. Tactul de transmisie se transmite de la emițător la receptor pentru citire, figura 5.1 a. Se asigură viteză mare, dar distanțele sunt mici. Costurile unui fir suplimentar sunt mari. Exemple sunt interfața SPI (Serial Peripheral Interface), IEEE1394a (1995), JTAG.

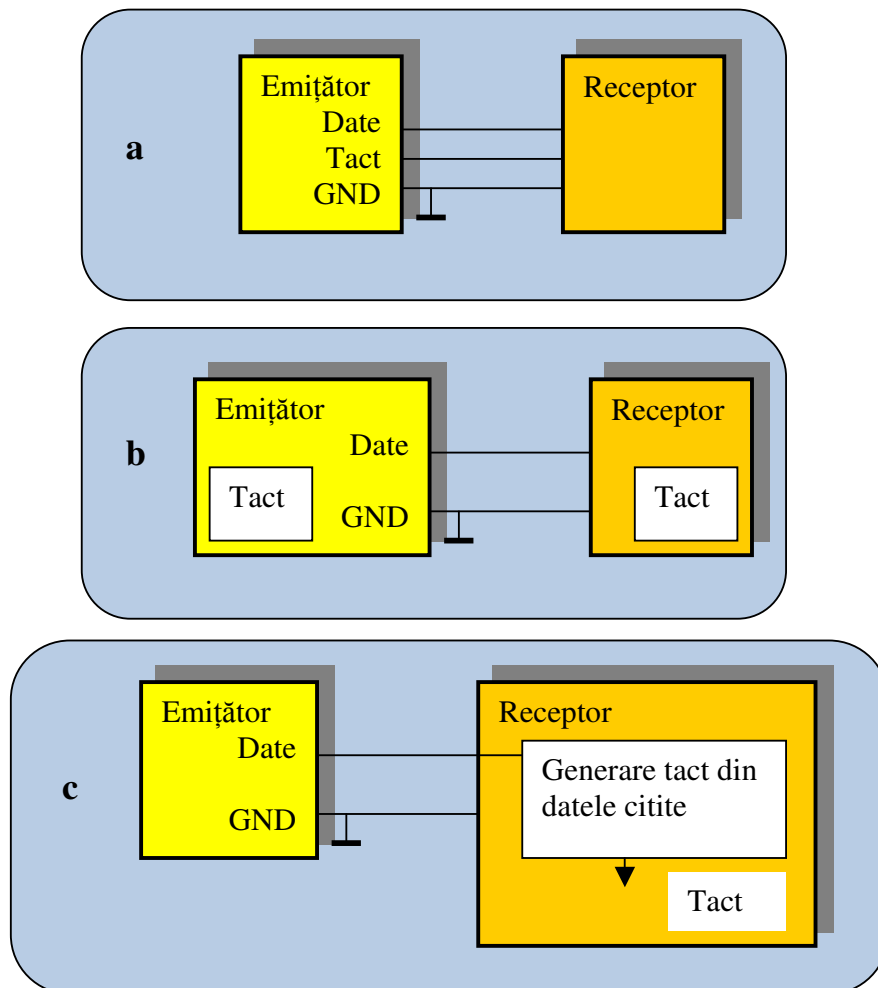


Figura 5.1. Variante de asigurare a tactului la recepție

1. Emisia și recepția se fac cu același tact, standard, cunoscut și de emițător și de receptor. Distanțele pot fi mari, nu apar costurile unui fir suplimentar, figura 5.1 b. Viteza nu poate fi mare datorită faptului că între tacele standard există diferențe. Câteva exemple sunt interfața serială asincronă RS232 și LIN.
2. Refacerea tactului din datele emise este cea mai modernă metodă folosită la cele mai multe aplicații actuale. Se pot asigura distanțe mari, costuri mici, viteze mari. Circuitul care reface tactul din date se numește buclă PLL și nu poate reface datele decât dacă există variații permanente de nivel, figura 5.1 c. Exemple sunt USB, CAN, FlexRay, IEEE1394b (2002), dar și Ethernet, SATA etc. Transmisiile wireless, de exemplu Bluetooth și ZigBee pot fi asociate cu transmisiile seriale pe un singur fir cu refacerea tactului.

## 5a.2. Codarea datelor

Primul pas în transmiterea datelor digitale pe o linie de transmisie este asocierea fiecărui bit cu variația unui semnal electric printr-un proces de codare. În continuare vor fi prezentate câteva tipuri de codări și exemple de codare.

### A.Codarea NRZ (Non Return to Zero)

Cea mai simplă și naturală codare este cea în care fiecare valoare logică se codează cu un nivel de tensiune.

Regula de codare este:

1. Zero se codifică cu 0V;
2. Unu se codifică cu +5V.

În figura 5.2. 16 biți într-o succesiune aleasă la întâmplare în care datele au multe schimbări de nivel logic au fost codificați cu 7 tranziții (2,28 biți/tranziție). Dacă apare un șir de 0-uri sau 1-uri semnalul electric nu are variații.

Intervalul de timp în care se codează un bit se numește celulă bit (interval cu roșu)

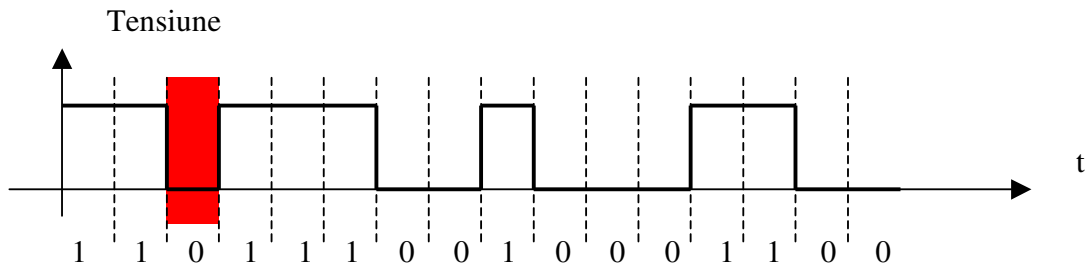


Figura 5.2. Codarea NRZ

### B.Codarea NRZI (Non Return to Zero Inverted)

Este o codare asemănătoare cu NRZ, cu regula de codare:

1. Zero se codifică cu lipsa unui front.
2. Unu se codifică cu un front.

În figura 5.3. 16 biți într-o succesiune aleasă la întâmplare au fost codificați cu 7 tranziții (2,28 biți/tranziție). Dacă apare un șir de 0-uri sau 1-uri semnalul electric nu are variații.

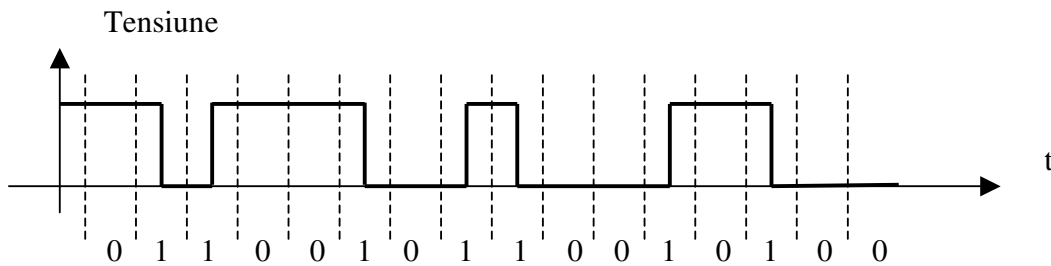


Figura 5.3. Codarea NRZI

### C.Codarea Manchester

Codarea Manchester a fost realizată prima oară la Universitatea din Manchester cu ocazia construirii calculatorului Mark 1 (numit și Baby) în 1948.

Regula de codare este:

1. Zero se codifică cu un front descrescător;
2. Unu se codifică cu un front crescător.

În figura 5.4. 8 biți într-o succesiune aleasă la întâmplare au fost codificați 8 biți cu 12 tranziții (1,5 tranziții pe bit). Dacă apare un șir de 0-uri sau 1-uri semnalul electric are variații. Codarea Manchester se folosește la rețeaua Ethernet.

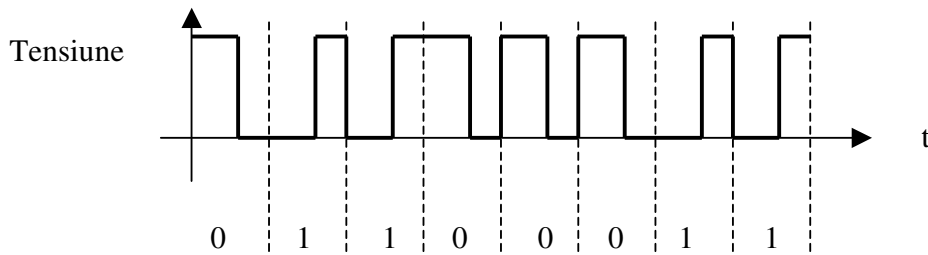


Figura 5.4. Codarea Manchester

#### D. Codarea FM (Frequency Modulation)

Codarea FM și MFM (Modified FM) au fost introduse de IBM în anii 1970 pentru codificarea datelor la scrierea pe hard discuri. La aceste codări fiecare celulă bit începe cu un impuls de tact, iar regula de codare este:

1. Zero se codifică cu lipsa unui impuls.
2. Unu se codifică cu un impuls.

În figura 5.5. 8 biți într-o succesiune aleasă la întâmplare au fost codificați cu 24 tranziții (3 tranziții pe bit). Codarea FM s-a folosit la înscrierea datelor pe suporturi magnetice, în prezent fiind înlocuită cu codări mai eficiente.

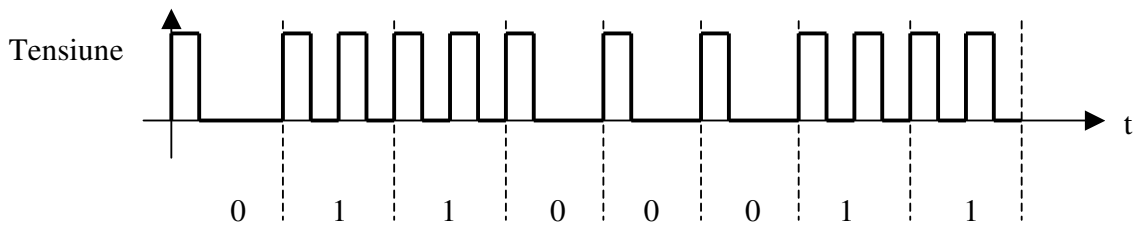


Figura 5.5. Codarea FM

Un tabel centralizator care cuprinde exemplele prezentate este dat în continuare:

Tabelul 5.1.

Codare	Eficiență (biți/tranziție)
NRZ	2.28
NRZI	2.28
Manchester	0.66
FM	0.33

**Concluzie în această etapă:**

Limita unui canal de transmisii de date seriale este dată de numărul de tranziții pe secundă. Interesul este să se trimită cât mai mulți biți cu cât mai puține tranziții. Deci la codurile NRZ și NRZI în exemplele alese au fost codificați în medie 2,28 biți cu o tranziție, iar la codurile FM și Manchester eficiența a fost mult redusă, doar 0,33 respectiv 0,66 biți cu o tranziție. Concluzia este că NRZ și NRZI sunt cele mai eficiente din acest punct de vedere.



Codarea Manchester este folosită la transmisia Ethernet iar codurile NRZ și NRZI nu mai sunt folosite în forma aceasta. Nu este aici o contradicție?



În compararea codurilor nu am luat în considerare o proprietate importantă, și anume capacitatea de autosincronizare.

În figura 5.1 c se poate vedea că tactul de citire este refăcut din datele citite cu o buclă PLL. Dacă o codare permite un număr mare de valori logice de același fel succesive care nu produc nicio tranziție, bucla PLL pierde sincronizarea și transmisia nu poate fi efectuată. Prin urmare, la codurile NRZ și NRZI tactul nu se poate reface din datele citite și codurile nu pot fi utilizate în modul de transfer cu refacerea tactului din date. Aceste coduri se numesc din acest motiv coduri neautosincronizabile. Codurile FM și Manchester, chiar dacă asigură o codare mai puțin eficientă sunt coduri autosincronizabile. Tabelul 5.1 poate fi completat astfel:

Tabelul 5.2

Codare	Eficiență (biți/tranziție)	Autosincronizabilitate
NRZ	2.28	NU
NRZI	2.28	NU
Manchester	0.66	DA
FM	0.33	DA

Cercetările orientate spre găsirea unor noi metode de codificare eficientă și autosincronizabile au fost îndreptate către modificarea NRZ sau NRZI care au o eficiență ridicată pentru a deveni autosincronizabile. Astfel au apărut codificările numite de grup, sau cu adăugare de biți (engl. bit stuffing). Aceste codificări impun ca la transmisia codată NRZ sau NRZI dacă apare un semnal codat cu mai multe celule bit fără tranziții decât o anumită limită, emițătorul forțează o tranziție, deci se adaugă un bit suplimentar.

În figura 5.6. se poate observa sus un semnal codat NRZ cu o succesiune de 6 valori de 1 logic succesive. Dacă codarea de grup admite doar 4 valori succesive de 1 logic, atunci emițătorul inserează un bit (cu roșu) suplimentar care generează 2 tranziții.

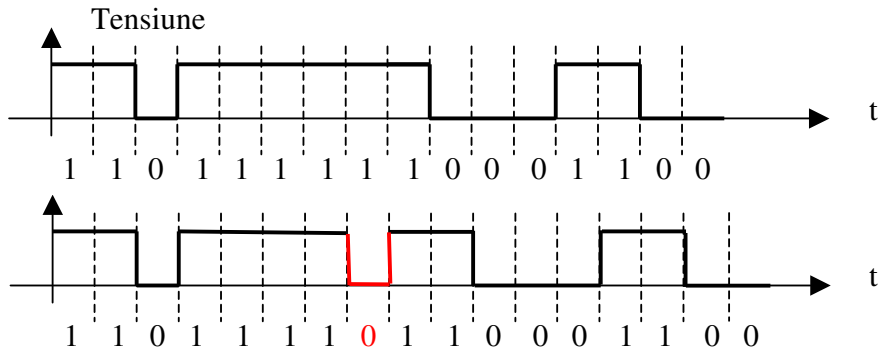


Figura 5.6. Codarea de grup

În această situație 16 biți de date au fost codificați NRZ cu 5 tranziții (3,2 biți cu o tranziție) iar după inserarea bitului suplimentar cei 16 biți sunt codificați cu 7 tranziții, adică 2,28 biți codificați cu o tranziție. Astfel se păstrează aproape la fel de mare eficiența codului NRZ și codarea a devenit autosincronizabilă.

Vor fi descrise în detaliu exemple de astfel de codări în modulele următoare. Codarea 8B10B va fi descrisă la interfața serial IEEE1394 (Modul 11), codarea EFM la înregistrarea pe suporturi optice și codarea RLL la scrierea pe suporturi magnetice (Modul 12).



Cum poate receptorul să diferențieze între situația în care cuvântul transmis are 4 biți de 1 urmați de un bit de 0 și situația în care cuvântul are 5 biți de 1 și emițătorul inserează un bit suplimentar de 0 după 4 biți de 1?



Se va înțelege după descrierea unui sistem de codare cu inserare de biți în modulele următoare. Ca exercițiu de gândire studenții pot propune variante de răspuns.

### 5a.3. Transmisii seriale asincrone și sincrone

Al doilea pas în transmisia informației este asocierea biunivocă a unui caracter cu o configurație binară. Cel mai utilizat este codul ASCII. În codul ASCII (American Standard Code for Information Interchange) fiecare literă este reprezentată de un număr. De exemplu, litera **A** este reprezentată prin numărul **65**, în timp ce pentru litera **z** este alocat numărul **122** în zecimal. În codul ASCII fiecare caracter este codificat prin 7 biți.

Semne speciale sunt de exemplu caracterul de comandă de revenire la începutul liniei CR (Carriage Return) și avans la o nouă linie (Line Feed) care împreună au efectul apăsării tastei Return.

Formatul, integritatea și dispunerea în timp a schimbului de date între procese care comunică între ele, fiecare desfășurându-se independent în sisteme de calcul interconectate, se stabilesc printr-un set de reguli și convenții specificate de un **protocol de comunicație**.

Un grup de cuvinte binare formează un cadru (bloc) de date. Protocoalele seriale pot fi orientate pe cuvânt sau pe cadru. Un cuvânt respectiv un bloc de date reprezintă entitățile de informație care se tratează unitar la receptor, adică sunt acceptate sau respinse în întregime. În transferul asincron (RS232, LIN) se transmit cuvinte și informația de sincronizare este atașată cuvântului dar în cele mai multe tipuri de transfer (USB, CAN, IEEE1394b, Ethernet, Bluetooth, ZigBee) informația de sincronizare este atașată cadrului, transmisiile fiind numite sincrone. La un cuvânt respectiv la un bloc de date se adaugă și informație de verificare a corectitudinii transferului (bit de paritate sau cuvinte de control).

În figura 5.7. este prezentată o diagramă simplificată a transmiterii cuvântului CURS.

Se poate observa că informația de sincronizare (marcată cu roșu) este atașată blocului de date la transmisia sincronă și fiecărui cuvânt la transmisia asincronă.

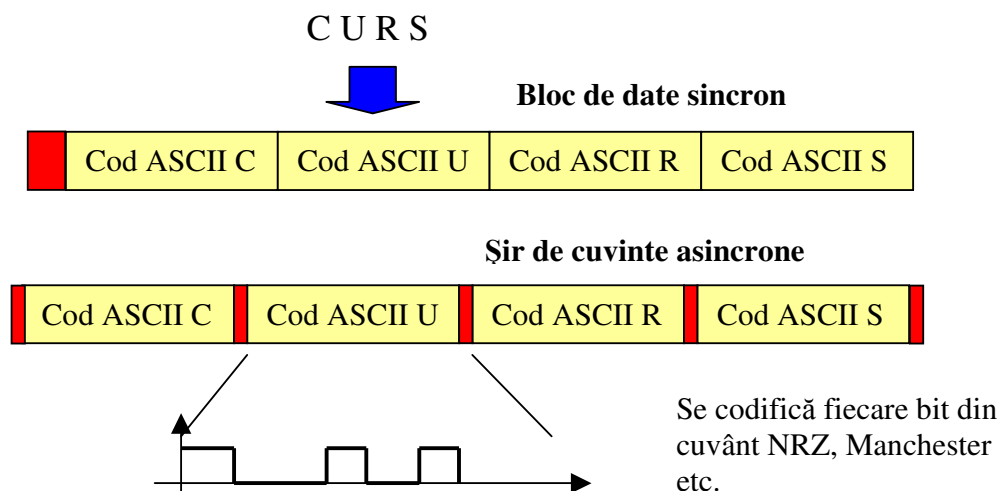


Figura 5.7. Cuvântul CURS transmis ca bloc de date sincron (sus) și asincron (jos)



### A. Structura unui cuvânt serial asincron

Linia de transmisie este în repaus în starea MARK (1 logic). Emisiș unui caracter este precedată de trecerea liniei în starea SPACE (0 logic) pe durata unui bit numit de START, figura 5.8.

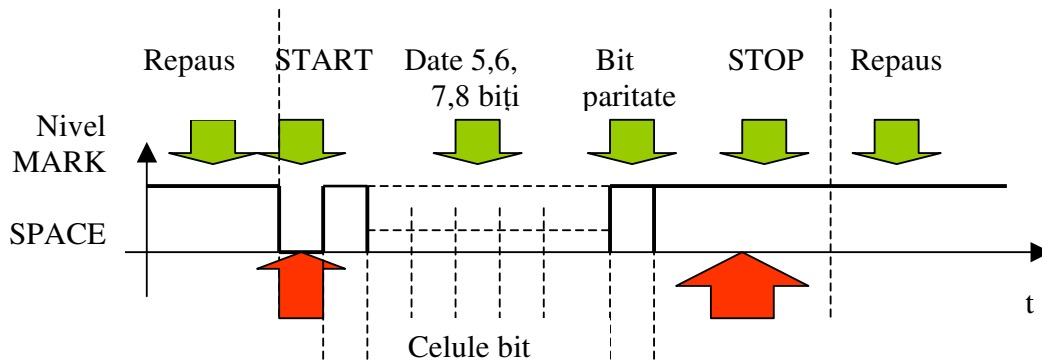


Figura 5.8. Structura unui cuvânt serial asincron

După bitul de START urmează 5, 6, 7 sau 8 biți de date cu codare NRZ. Codarea NRZ nu este autosincronizabilă, prin urmare transmisia serială asincronă poate fi cu tact standard sau cu transmisia tactului. Transmisia se numește asincronă pentru că emisia unui caracter poate avea loc în orice moment, dacă linia este în repaus, în stare MARK. Biții în cuvânt sunt transmiși sincron, cu un tact standard sau cu un tact transmis, fiecare bit fiind codificat cu un nivel logic într-o celulă bit. După biții de date urmează un bit de paritate pentru verificarea corectitudinii mesajului la receptor. După bitul de paritate urmează un bit, un bit și jumătate sau doi biți de STOP, în stare MARK, după care linia reintră în repaus.



Ce înseamnă că se transmit un bit și jumătate de STOP? Bitul este divizibil?



Un bit și jumătate înseamnă o perioadă de timp corespunzătoare unei celule bit și jumătate.

În figura 5.8. a fost marcată cu săgeți roșii informația de sincronizare, biții de START și STOP. Această informație este în cel mai favorabil caz 2 biți, care la 8 biți transmiși în cuvânt înseamnă 25%. Informația suplimentară transmisă în cazul acesta este 25%.

## B. Structura unui cadru de date sincron

Un cadru de date este format din mai multe cuvinte transmise serial. Informația de sincronizare este transmisă la începutul cadrului (blocului) de date, figura 5.9.

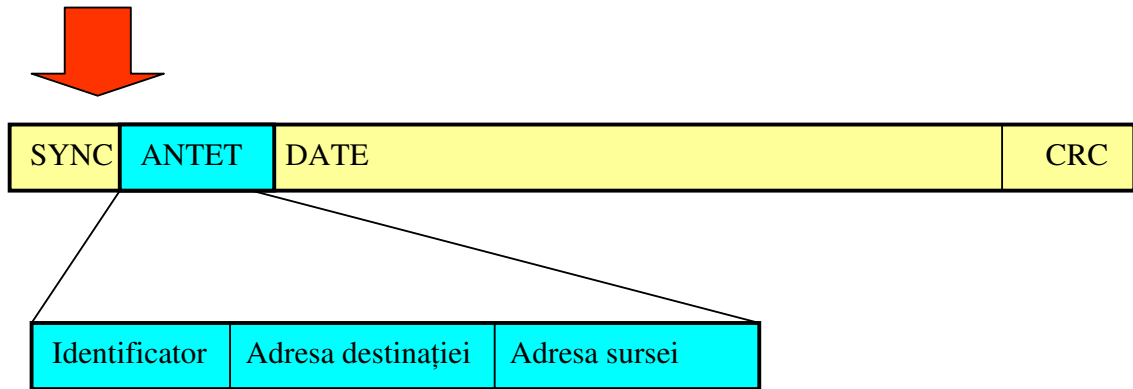


Figura 5.9. Structura unui cadru de date tipic

La transmisia sincronă este posibilă transmisia datelor cu refacerea tactului la receptor din datele transmise. De aceea, un cadru de date începe cu o zonă de sincronizare care conține tranziții dese, pentru a permite buclei PLL din receptor să se sincronizeze. Urmează apoi de regulă un antet format dintr-un identificator care depinde de tipul transmisiei, adresa destinației și adresa sursei. Zona de date are cea mai mare dimensiune și este urmată de cuvinte CRC (Cyclic Redundancy Code) pentru verificarea corectitudinii transmisiei. Presupunând că zona de date este formată din 100 de cuvinte iar zona de sincronizare și antetul au 10 cuvinte se poate vedea că surplusul de informație (overhead bits) este doar 10%, mai mic decât în transmisia asincronă.

Structuri particulare de cadre de date vor fi prezentate la transmisia Ethernet (în modulul 8), la USB (Modulul 9), la IEEE 1394 și SATA (modulul 11) etc.



Există transmisii serială sincrone în care informația să nu fie precedată de biți de sincronizare?



Da, de exemplu la interfața serială SPI cu transmiterea tactului la care tactul nu se refacă la receptor. SPI este des întâlnită la microcontrollere și va fi descrisă în Modulul 7.



### Rezumat

Modulul “**Interfețe seriale**” a fost împărțit în două. Prima parte, “**Codificarea datelor**” începe cu o clasificare a metodelor de realizare a tactului la receptor, apoi prezintă câteva metode de codare, cu exemple: NRZ, NRZI, Manchester și FM. Se estimează eficiența codării și se compară eficiențele metodelor prezentate după care se definește autosincronizabilitatea. Este prezentat principiul codării de grup (cu inserare de biți). La sfârșit sunt prezentate structurile unui cuvânt serial asincron și a unui cadru de date sincron. În aceste structuri este pusă în evidență informația de sincronizare.



### Bibliografie

1. M. Romanca, P. Ogrutan, *Sisteme cu calculator incorporat. Aplicații cu microcontrollere*, Editura Universității Transilvania Brașov, 2011, pag. 1-4 online la:  
<http://vega.unitbv.ro/~ogrutan/Microcontrollere2011/1-introducere-interfete%20paralele%20si%20seriale.pdf>
2. Communication Systems, Line Codes, online la:  
[http://en.wikibooks.org/wiki/Communication\\_Systems/Line\\_Codes](http://en.wikibooks.org/wiki/Communication_Systems/Line_Codes)
3. Sebastian Andrews, Communication Networks, online la:  
<http://services.eng.uts.edu.au/~kumbes/ra/Transmission/Digital%20Wave%20Formatting/index.htm>
4. Bit stuffing, online la:  
[http://en.wikipedia.org/wiki/Bit\\_stuffing](http://en.wikipedia.org/wiki/Bit_stuffing)

### Test de autoevaluare



**1. Transmisia serială cu tact standard asigură viteza cea mai mare de transfer de date**

**R**

adevărat

fals

**I. vezi pagina 2 și 3**

**2. Codați NRZ șirul de 16 biți: 1110 0101 0001 1010 Eficiența codării este:**

**R**

(a) 2,28 biți/tranziție

(b) 0,66 biți/ tranziție

(c) 1,78 biți/ tranziție

(d) 2,45 biți/ tranziție

**I. vezi pagina 4**

**3. Codați NRZI șirul de 16 biți: 1110 0101 0001 1010 Eficiența codării este:**

**R**

(a) 2,28 biți/tranziție

(b) 2 biți/ tranziție

(c) 0,33 biți/ tranziție

(d) 1,78 biți/ tranziție

**I. vezi pagina 4**

**4.Codați Manchester șirul de 16 biți: 1110 0101 0001 1010**

**Eficiența codării este:**

**R**

- (a) 0,62 biți/tranziție
- (b) 1,2 biți/ tranziție
- (c) 0,33 biți/ tranziție
- (d) 0,73 biți/ tranziție

**I. vezi pagina 4 și 5**

**5.Codați FM șirul de 16 biți: 1110 0101 0001 1010 Eficiența codării este:**

**R**

- (a) 1.2 biți/tranziție
- (b) 0,66 biți/ tranziție
- (c) 0,33 biți/ tranziție
- (d) 0,22 biți/ tranziție

**I. vezi pagina 5**

**6.Codarea de grup (cu inserare de bți) se aplică la un cod autosincronizabil și urmărește obținerea unei eficiențe mai mari de codare**

**R**

- adevărat
- fals

**I. vezi pagina 7**

**2. Se trimite serial asincron fără bit de paritate cuvântul de 8 biți: 1110 0101. Câte tranziții are semnalul pe linie?**

- (a) 4 tranziții
- (b) 5 tranziții
- (c) 6 tranziții
- (d) 7 tranziții

**R**

**I. vezi pagina 9**

**R**

**Răspunsuri corecte:**

1. fals, vezi caracteristicile transmisiei cu tact standard de la pagina 2 respectiv 3
2. c, pentru că apar 9 tranziții, iar  $16/9=1,78$ , pagina 4
3. b, sunt 8 tranziții, una la fiecare bit de 1, deci  $16/8=2$ , pagina 4
4. d, sunt 22 tranziții, deci  $16/22=0,73$ , pagina 4 și 5
5. c, sunt 16 tacte la fiecare început de celulă, deci 32 tranziții și 8 biți de 1, deci 16 tranziții.  $16/48=0,33$ , pagina 5
6. fals, codarea de grup (cu inserare de bți) se aplică la un cod NEautosincronizabil și urmărește obținerea unui cod autosincronizabil
7. c, datele au 4 tranziții, START are 2 tranziții și STOP nicio tranziție pentru că ultimul bit este 1.