

# CAPITOLUL 2

## Introducere în controlul bazat pe microprocesor

---

### OBIECTIVELE

După studierea acestui capitol, ar trebui să puteți să:

- Înțelegeți ce este un microprocesor, ce face și cum funcționează.
- Înțelegeți conceptele de memorie RAM și ROM ale computerului și modul în care memoria este accesată prin intermediul magistralei de adrese și al magistralei de date.
- Înțelegeți cum funcționează interfețele de date paralele și seriale.
- Efectuați calcule relevante referitoare la convertoarele analogice-digitale și convertoarele digital-analogice.
- Înțelegeți principiile software-ului controlerului digital.
- Recunoașteți și descrieți caracteristicile diferitelor tipuri de controlere digitale disponibile, adică microcontrolere, computere cu o singură placă, controlere logice programabile și computere personale.

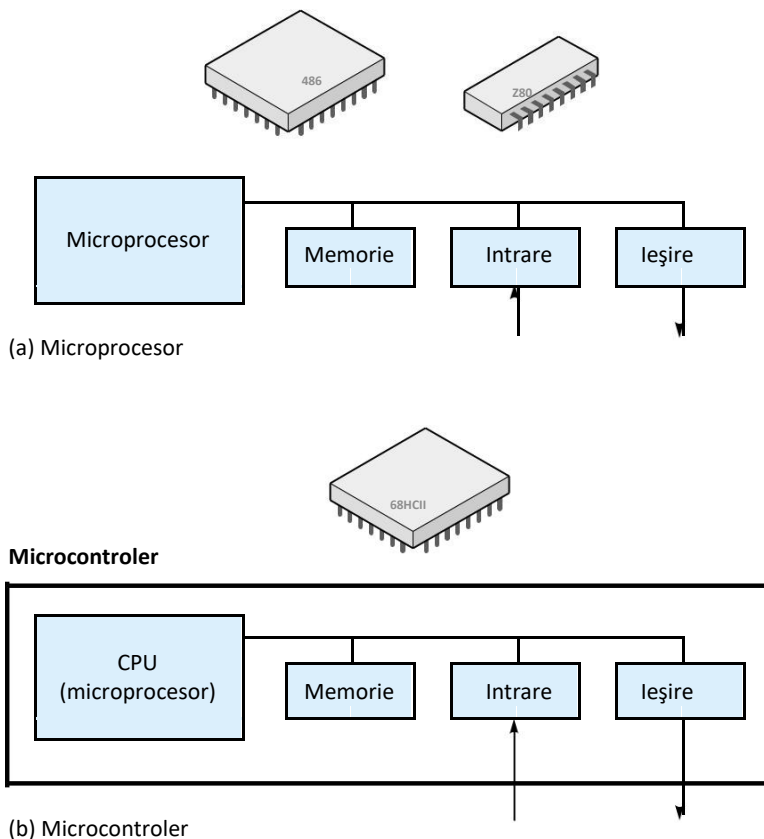
---

### INTRODUCERE

Circuitul digital integrat (IC) numit **microprocesor** [Figura 2.1(a)], a inaugurat o nouă eră pentru electronica sistemelor de control. Această revoluție a avut loc deoarece microprocesorul aduce flexibilitatea prin controlul programului și puterea computațională a unui computer pentru a rezolva orice problemă. Aplicațiile de control automat sunt foarte potrivite pentru a profita de această tehnologie, iar sistemele de control bazate pe microprocesoare înlocuiesc rapid multe sisteme de control mai vechi bazate pe circuite analogice sau relee electromecanice. Unul dintre primele controlere bazate pe microprocesor realizat special pentru aplicațiile de control a fost controlerul logic programabil (PLC), care este discutat mai târziu în acest capitol și în capitolul 12. Un microprocesor în sine nu este un computer; componente suplimentare, cum ar fi circuitele de memorie și intrare/ieșire, sunt necesare pentru a-l face operațional. Cu toate acestea, **microcontrolerul** [Figura 2.1 litera (b)], care este

**Figura 2.1**

Microprocesor și  
microcontroler.



o rudă apropiată a microprocesorului, *conține* toate funcțiile computerului pe un singur IC. Microcontrolerelor le lipsește o parte din puterea și viteza microprocesoarelor mai noi, dar compactitatea lor este ideală pentru multe aplicații de control; majoritatea așa-numitelor dispozitive controlate de microprocesoare, cum ar fi automatele, folosesc într-adevăr microcontrolere. Unele motive specifice pentru utilizarea în design digital a microprocesoarelor în sistemele de control sunt următoarele:

- Semnalele de nivel scăzut de la senzori, care odată convertite în digital, pot fi transmise pe distanțe lungi practic fără erori.
- Un microprocesor poate gestiona cu ușurință calcule complexe și strategii de control.
- Memoria pe termen lung este disponibilă pentru a urmări parametrii din sistemele cu mișcare lentă.
- Schimbarea strategiei de control este ușoară prin încărcarea într-un nou program; nu sunt necesare modificări hardware.
- Controlerile bazate pe microprocesoare sunt mai ușor conectate la rețeaua de calculatoare din cadrul unei organizații. Acest lucru permite designerilor să introducă modificări de program și să citească starea sistemului de la terminalele lor de birou.

În acest capitol, vom prezenta conceptele de bază ale unui sistem bazat pe microprocesoare și microcontrolere, cu un accent deosebit pe aplicațiile sistemului de control.

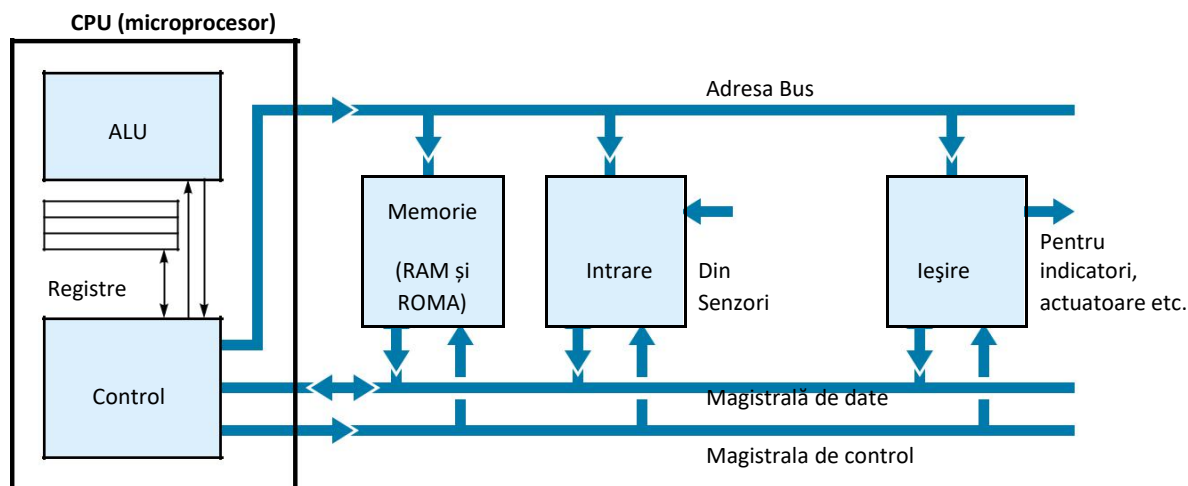
În primele secțiuni ale acestui capitol sunt introduse conceptele de bază ale hardware-ului și funcționării microprocesorului (aceste concepte se aplică și microcontrolerelor). Am inclus acest material deoarece studiul sistemelor moderne de control ar trebui să aibă cel puțin o cunoaștere generală a modului în care microprocesorul își îndeplinește activitatea.

## 2.1 INTRODUCERE ÎN MICROPROCESORE HARDWARE-UL SISTEMULUI

Un computer este format din patru unități funcționale de bază: unitatea centrală de procesare (CPU), memoria, intrarea și ieșirea (I/O). **Unitatea centrală** de procesare realizează calculul computațional efectiv și este compusă din două sub-părți: unitatea aritmetică și logică și secțiunile de control (Figura 2.2). Unitatea logică aritmetică (ALU) efectuează calculele numerice și logice reale, cum ar fi adunarea, scăderea, AND, OR și așa mai departe. Secțiunea de control a procesorului gestionează fluxul de date, cum ar fi citirea și executarea instrucțiunilor programului. Dacă datele necesită calcule, secțiunea de control o predă ALU pentru procesare. Într-un computer bazat pe microprocesor, microprocesorul este procesorul.

**Figura 2.2**

O diagramă bloc a unui computer bazat pe microprocesor.



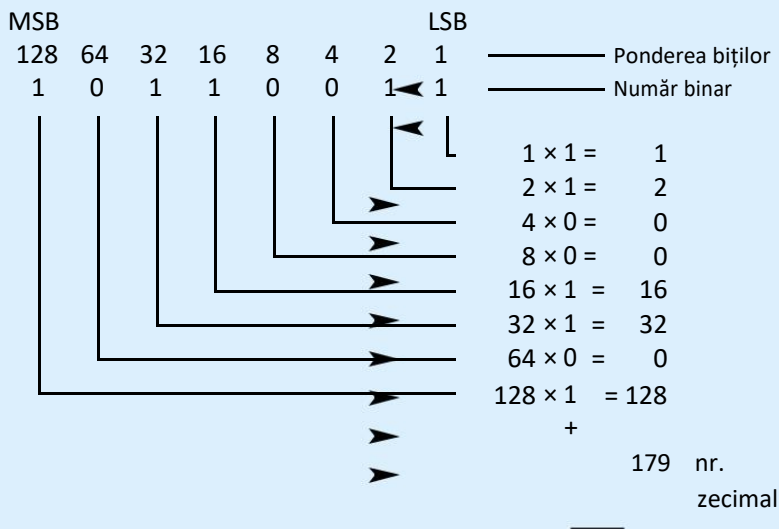
Datele digitale sunt sub formă de **biți**, unde fiecare bit are o valoare de 1 sau 0. Circuitele digitale folosesc de obicei 5 Vdc pentru a reprezenta 1 logic și 0 Vdc pentru a reprezenta 0 logic. Opt biți împreună se numește un **octet**. Un microprocesor gestionează datele digitale în **cuvinte**, unde un cuvânt poate avea o lățime de 8, 16 sau 32 de biți. De exemplu, un microprocesor pe 8 biți are un cuvânt de mărimea unui octet, cu o valoare zecimală maximă de 255. (Computerele reprezintă numerele din *sistem de numere binare*; de exemplu, 11111111 binar = 255 zecimal.) Bitul cel mai din dreapta dintr-un număr binar are cea mai mică valoare (de obicei 1) și se numește bitul cel mai puțin **semnificativ** (LSB). Bitul din stânga reprezintă cea mai mare valoare și se numește bitul **cel mai semnificativ** (MSB). Conversia între binar și zecimal poate fi efectuată direct cu majoritatea calculatoarelor științifice sau manual folosind tehnica prezentată în exemplul 2.1. Pentru a exprima valori mai mari de 255, două sau mai multe cuvinte sunt puse împreună. În acest text, vom presupune că microprocesoarele pe 8 biți sunt utilizate, cu excepția cazului în care se prevede altfel.

### EXEMPLUL 2.1

Găsiți valoarea zecimală a numărului binar pe 8 biți 10110011.

#### SOLUȚIE

Fiecare bit din numărul binar are o valoare sau o greutate diferită. LSB are o pondere de 1. Bitul din stânga LSB are o pondere de 2, al treilea bit are o pondere de 4 și așa mai departe, ponderea dublându-se pentru fiecare bit până la 128 pentru MSB. Pentru a găsi valoarea unui număr pe 8 biți, puteți să configurați o diagramă (prezentată mai jos) și apoi să însumați valorile care corespund 1s din numărul binar. Secțiunea de **memorie** a computerului este un loc în care sunt stocate date digitale în formă binară (1s și 0s). Memoria constă din celule organizate în grupuri pe 8 biți. Fiecărui octet



i se da o **adresa numerică** unică, care reprezintă locația acesteia la fel cum o adresă stradală reprezintă locația unei case. Datele sunt scrise în memorie și citite din memorie, bazate exclusiv pe adresa lor. Într-un anumit circuit de memorie, adresele ar putea începe de la 1000 și să ruleze consecutiv până în 2000. Figura 2.3 sunt prezentate diagrame dintr-o secțiune de memorie. Rețineți că primul octet de date are o valoare zecimală de 2 (00000010 = 2 zecimale) și o adresă de 1000.

Computerele au de obicei două tipuri de memorie adresabilă. Prima este **memoria cu acces aleator** (RAM), care permite computerului să citească și să scrie date la oricare dintre adresele sale (se mai numește și memorie de citire/scriere sau RWM). Toate datele din acest tip de memorie se pierd atunci când alimentarea este oprită și se numește **memorie volatilă** (o excepție este designul în care memoria RAM este menținută "în viață" cu o baterie mică). Al doilea tip de memorie este memoria doar în **citire** (ROM), care este similară cu memoria RAM, cu excepția faptului că datele noi nu pot fi scrise în; toate datele din ROM sunt încărcate din fabrică și nu pot fi modificate de computer. Această memorie nu își pierde datele atunci când alimentarea este oprită și se numește **memorie non-volatilă**. Majoritatea sistemelor de microprocesoare au atât RAM, cât și ROM. MEMORIA RAM este utilizată pentru stocarea temporară a programelor și ca memorie temporară pentru CPU. ROM-ul este utilizat pentru a stoca programe și date care trebuie să fie întotdeauna disponibile. De fapt, multe computere folosesc un EPROM (memorie programabilă doar în citire) sau un EEPROM (ROM programabil care poate fi șters electric) în loc de un ROM pentru memorare pe termen lung. EPROM-urile pot fi șterse cu o lumină UV puternică și reprogramate. EEPROM-urile pot fi șterse și reprogramate electric. Unitățile de disc stochează, de asemenea, date digitale, dar într-o formă care trebuie procesată înainte ca acestea să fie accesibile microprocesorului.

Secțiunea de **intrare/ieșire** (I/O) a computerului îi permite să interacționeze cu lumea de afară. Secțiunea de intrare este portul prin care noi programe și date sunt introduse în computer, iar secțiunea de ieșire permite computerului să comunice rezultatele sale. O interfață I/O se numește **port**. Un port de intrare este un circuit care conectează dispozitivele de intrare la computer; exemple de dispozitive de intrare sunt tastaturile, senzorii și comutatoarele. Un port de ieșire este un circuit care conectează computerul la dispozitivele de ieșire. Exemple de dispozitive de ieșire sunt lămpile indicatoare, dispozitivele de acționare și monitoare. Intrarea/ieșirea este discutată mai detaliat în secțiunea următoare.

Referindu-ne din nou la Figura 2.2, vedem că blocurile sunt conectate prin trei linii etichetate magistrală de adrese, magistrală de date și magistrală de control. **Magistrala de adrese** este un grup de fire care transportă o adresă (în formă binară) de la CPU la memorie și circuite I/O. De asemenea,

**Figura 2.3**  
O secțiune de  
memorie.

Adresă	Date
1000	00000010
1001	01100100
1002	00000000
1003	01011100
1004	10011010
↓	↓

nevoia de memorie pentru a primi adrese a fost deja discutată, dar porturile I/O au nevoie de adrese. Se pare că toate porturile I/O sunt atribuite adrese și sunt tratate în esență ca locații de memorie de către CPU. CPU utilizează ieșiri de date în lumea exterioară prin trimiterea lor la o adresă de port. Când circuitele portului de ieșire detectează adresa atribuită pe magistrala de adrese, se deschide și permite trecerea datelor de la magistrala de date la orice este conectat la port. Există două moduri în care I / O abordarea se face. Unele microprocesoare utilizează ceea ce se numește **intrare/ieșire mapată de memorie**, unde o adresă I/O este tratată la fel ca o altă adresă de memorie. Alte microprocesoare tratează adresele I/O complet separate de adresele de memorie.

Magistrala **de date** este un grup de opt "fire" care transportă datele numerice reale din loc în loc în computer. Figura 2.2 arată modul în care magistrala de date interconectează toate blocurile. Fluxul de date în ambele direcții pe magistrala de date. De exemplu, datele de intrare intră prin portul de intrare și trec prin magistrala de date la CPU. În cazul în care procesorul trebuie să stocheze aceste date, le va trimite înapoi prin magistrala de date în memorie. Datele care urmează să fie generate sunt trimise (de CPU) prin magistrala de date la portul de ieșire. Dacă magistrala de date se conectează la toate blocurile, cum știu datele la ce bloc să meargă? Răspunsul este sistemul de adrese. De exemplu, atunci când procesorul trimite date în memorie, o face în doi pași: În primul rând, pune adresa de memorie de destinație pe magistrala de adrese; în al doilea rând, pune datele pe magistrala de date. Când memoria desemnată detectează propria adresă, aceasta "se trezește" și preia datele din magistrala de date. Celelalte blocuri conectate la magistrale vor ignora întreaga secvență pentru că nu au fost abordate. O analogie bună este sistemul de telefonie, în cazul în care numărul de telefon este analog cu adresa de memorie. Chiar dacă mii de telefoane pot fi conectate la sistem, atunci când formați un număr, doar telefonul desemnat sună. Frumusețea sistemului de magistrale este că este extensibil. Memoria sau unitățile I/O adresabile pot fi adăugate la sistem prin simpla conectare a acestora la magistrale.

**Magistrala de control** (a se vedea figura 2.2) constă în semnale de sincronizare și de control al evenimentelor de la CPU. Aceste semnale sunt utilizate pentru a controla fluxul de date pe magistrala de date. De exemplu, unul dintre semnalele de control este linia **de citire/scriere** (R/W). Acest semnal informează memoria în cazul în care procesorul dorește să citească datele existente din memorie sau scrie date noi în memorie. Mașinile care nu sunt mapate la memorie au o linie de control *memorie-I/O*. Acest semnal informează sistemul dacă schimbul curent de date implică memorie sau un port I/O. În general, magistral de control nu este la fel de standardizată ca și magistrala de adrese și date.

## 2.2 INTRODUCERE ÎN MICROPROCESOARE – OPERAȚII

Microprocesorul funcționează prin executarea unui program de instrucțiuni. Crearea pro este similar în concept cu programarea în BASIC, C, sau orice alt limbaj computer la nivel înalt. Fiecare tip de microprocesor are propriul set de **instrucțiuni**, care este setul de comenzi pe care a fost proiectat să le recunoască și să le respecte. Instrucțiunile Microprocesorului

sunt sunt foarte elementare și specifice. Multe instrucțiuni de microprocesor pur și simplu muta datele dintr-un loc în altul în sunt elementare și specifice. Multe instrucțiuni ale microprocesoarelor mută datele dintr-o locație în alta; altele efectuează operații matematic sau logice. Încă un alt grup de instrucțiuni de control pentru fluxul programului, cum ar fi salturile înainte sau înapoi în program. Fiecărei instrucțiune din setul de instrucțiuni îi este atribuit propriului cod unic **de operare** (care are de obicei 8 biți lungime și este denumit **op-code**). CPU utilizează acest număr de 8 biți pentru a identifica instrucțiunea.

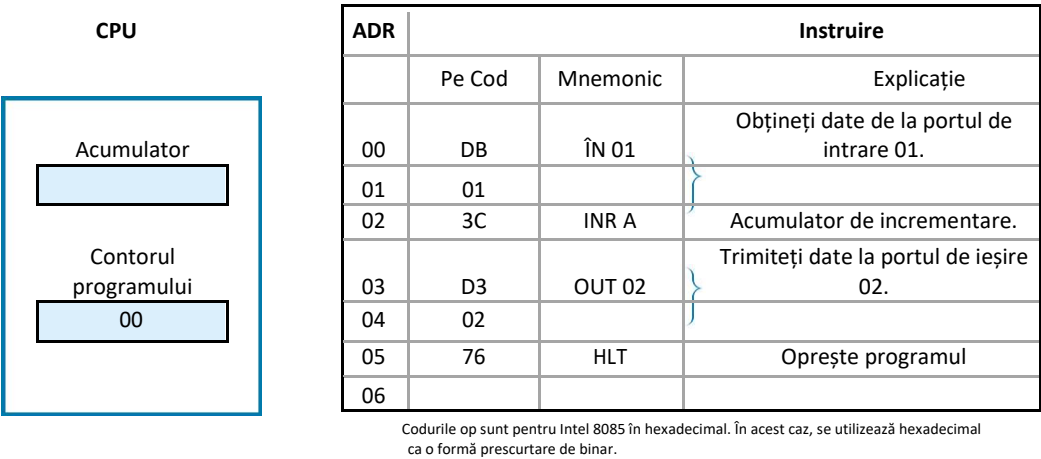
Toate microprocesoarele au cel puțin un **acumulator** [Figura 2.4(a)], care este un registru de reținere a datelor în CPU. Acumulatorul acționează ca o "zonă de așteptare" pentru date. Este comun pentru ca datele care vin la CPU să meargă mai întâi la acumulator, unde pot fi operate. În mod similar, majoritatea datelor care părăsesc procesorul ies din acumulator. Operațiile matematice stochează de obicei rezultatul în acumulator. Multe dintre instrucțiuni implică acumulatorul într-un fel sau altul.

Un program **de limbaj de mașină** este o listă de instrucțiuni (sub formă de op-cod) pe care microprocesorul trebuie să le urmeze. Înainte ca programul să poată fi executat, acesta trebuie mai întâi încărcat secvențial în memorie. Codul op pentru prima instrucțiune este încărcat la prima locație de adresă, codul op pentru a doua instrucțiune este încărcat în continuare în linie și așa mai departe.

Figura 2.4(b) prezintă o secțiune de memorie cu un program scurt încărcat. Lista pro-gram include adresa, op-code, mnemonic, și o scurtă explicație. (Un **mnemonic** este o abreviere în limba engleză a unei instrucțiuni. O listă de programe folosind doar mnemonics se numește **limbaj de asamblare**. ) Programul din Figura 2.4 (b) direcționează CPU pentru a obține 1 octet de date de la portul de intrare 01, adăugați 1 la acesta și trimiteți rezultatul la portul de ieșire 02. Înainte ca execuția să poată începe, adresa primei instrucțiuni trebuie încărcată în contorul programului. Contorul **de program** este un registru special de stocare a adreselor pe care procesorul îl utilizează pentru a urmări unde se află în program, la fel ca un marcaj. Contorul program deține întotdeauna adresa următoarei instrucțiuni care urmează să fie executată. O dată ce

Figura 2.4

Procesorul utilizează un acumulator și un contor de program pentru a executa un program simplu.



(a) Microprocesor  
Registre

(b) Exemplu de program de microprocesor

microprocesorul este activat, execuția programului este complet automată. Procesul de execuție este o serie de cicluri de **preluare-executare**, prin care microprocesorul preia mai întâi instrucțiunea din memorie și apoi o execută. Următoarele sunt etapele specifice prin care ar trece microprocesorul pentru a executa programul din figura 2.4 (b):

1. Microprocesorul preia prima instrucțiune din memorie. Știe unde să găsească instrucțiunea, deoarece adresa sa este în contorul programului.
2. Odată ajuns în CPU, codul op este decodat pentru a vedea ce instrucțiune este, apoi hardware-ul adecvat este activat pentru a executa această instrucțiune. În programul de exemplu din Figura 2.4 (b), prima instrucțiune (IN 01) are o lungime de 2 octeți. Primul octet al instrucțiunii este op-cod-ul, îi spune CPU pentru a introduce date de la un port. Al doilea octet al instrucțiunii îi spune procesorului *din ce* port să citească. Execuția acestei instrucțiuni face ca datele din portul de intrare 01 să călătorească de-a lungul magistralei de date la acumulator. De asemenea, contorul program avansează la 02 (adresa următoarei instrucțiuni). Executarea primei instrucțiuni este acum completă.
3. Începe următorul ciclu de preluare-executare, de data aceasta preluând instrucțiunea de la adresa 02. Noua instrucțiune (INR A) este "incrementați acumulatorul", astfel încât acumulatorul este trimis la ALU pentru a fi incrementat (adăugați 1) și rezultatul pus înapoi în acumulator. Contorul programului avansează la 03, care este adresa următoarei instrucțiuni.
4. Începe următorul ciclu de preluare-executare, de data aceasta preluând instrucțiunea de la adresa 03. Instrucțiunea (OUT 02) este executată, determinând trimiterea datelor acumulatorului la portul de ieșire 02.
5. Instrucțiunea finală este preluată. Este o "oprire", care face ca microprocesorul să înceteze să funcționeze și să intre într-un mod de așteptare.

## 2.3 INTERFAȚAREA CU UN MICROPROCESOR

**O parte importantă a oricărui sistem de control este legătura dintre controler și lumea reală.** Pentru un controler digital, datele intră și ies printr-o interfață paralelă sau printr-o interfață serială. Ambele formate de date sunt discutate în continuare.

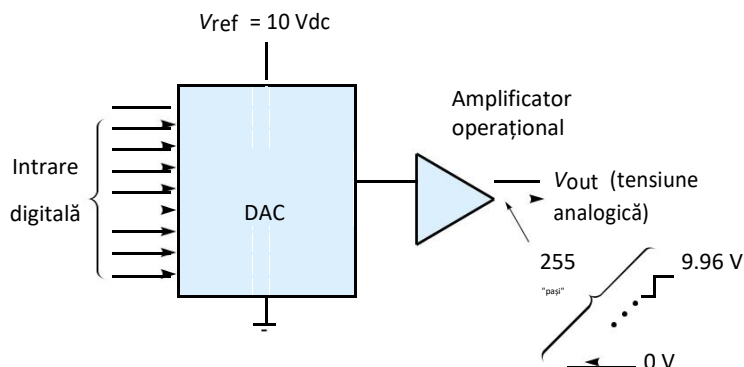
### Interfața paralelă

**Interfața paralelă** transferă date pe 8 biti (sau mai mult) în același timp, folosind opt fire separate. Este, în esență, o extensie a magistralei de date în lumea exterioară. Interfața paralelă este ideală pentru introducerea sau ieșirea datelor de pe dispozitive care sunt fie pornite, fie oprite. De exemplu, un comutator cu o singură limită utilizează un singur bit de intrare, iar un semnal on-off la un motor necesită un singur bit de ieșire. Aceste semnale pe 1 bit se numesc **variabile logice**, și opt astfel de semnale pot fi furnizate de la un singur port (8-bit). Acest concept va fi extins mai târziu în această secțiune.



**Figura 2.5**

O diagramă bloc a convertorului digital-analogic (DAC).



În alte aplicații, controlerul poate utiliza o interfață paralelă pentru a se conecta la un dispozitiv analogic, de exemplu, conducerea unui motor de curent continuu cu turație variabilă. Într-un astfel de caz, ieșirea binară a controlerului trebuie mai întâi convertită într-o tensiune analogică înainte de a putea conduce motorul. Această operație este efectuată de un circuit special numit convertor digital-analogic.

### Conversia digitală în analogică

**Convertorul digital-analogic (DAC)** este un circuit care transformă un cuvânt digital într-o tensiune analogică. Nu este în domeniul de aplicare al acestui text pentru a descrie funcționarea internă a DAC, dar o înțelegere generală a parametrilor de funcționare este adecvată.

Figura 2.5 prezintă diagrama bloc a unui DAC tipic pe 8 biți. Intrarea este un cuvânt digital pe 8 biți. Ieșirea este un curent care este proporțional cu valoarea de intrare binară și trebuie convertit la o tensiune cu un amplificator operațional. Totuși trebuie furnizată o tensiune de referință stabilă ( $V_{ref}$ ). Această tensiune definește tensiunea analogică maximă - adică pentru o intrare digitală a 11111111,  $V_{out}$  este în esență  $V_{ref}$ . Dacă intrarea este 00000000,  $V_{out}$  va fi 0 Vdc. Pentru toate valorile dintre ele, tensiunea de ieșire este un procent liniar de  $V_{ref}$ . Mai exact, tensiunea de ieșire pentru orice intrare digitală (pentru DAC-ul pe 8 biți) este:

$$V_{out} = (input \times V_{ref})/256 \quad (2.1)$$

Unde:

$V_{out}$  = ieșirea în volți a DAC-ului

Input = valoare digital de intrare

$V_{ref}$  = valoarea de referință a DAC-ului

### EXEMPLUL 2.2

Un DAC pe 8 biți are  $V_{ref}$  de 10 V. Intrarea binară este 10011011. Găsiți tensiunea de ieșire analogică.

**SOLUȚIE**

Intrarea binară a 10011011 are o valoare zecimală de 155. Aplicând ecuația 2.1, putem calcula tensiunea de ieșire analogică:

$$V_{out} = (input \times V_{ref})/256 = (155 \times 10V)/256 = 6,05V$$

Prin urmare, 6.05 V este tensiunea pe care ne-am aștepta să o găsim pe pinul de ieșire analogic. [Este interesant de observat că, dacă intrarea ar fi pusă pe 1 bit (care este o valoare zecimală de 255), ieșirea ar fi  $(255/256) \times 10 \text{ V} = 9.96 \text{ V}$ , nu 10 V cum v-ați putea aștepta. Aceasta este o caracteristică a DAC.]

Un aspect important al conversiei digital-analogice este **rezoluția**. De rezoluția unui DAC depinde eroarea care este introdusă la conversia între digital și analogic. Această eroare apare deoarece cuvintele digitale pot reprezenta numai valori discrete, așa cum indică diagrama scărilor din figura 2.5. De exemplu, valoarea maximă a unui număr de 8 biți este de 255 zecimale, ceea ce înseamnă că există 255 de "pași" posibili ai tensiunii de ieșire. Diferența dintre pași este valoarea bitului cel mai puțin semnificativ (LSB). Deoarece cea mai mică creștere este un pas, rezoluția (pentru datele pe 8 biți) este 1 parte din 255 sau 0,39%. Această rezoluție este adecvată pentru multe aplicații, dar dacă este nevoie de mai mult, două (sau mai multe) porturi pe 8 biți pot fi utilizate împreună. Două porturi furnizează 16 biți de date. Valoarea zecimală maximă de 16 biți este de 65.535. Posibilitatea de a împărți un număr analogic în 65.535 de părți înseamnă că fiecare parte va fi mult mai mică, astfel încât să putem reprezenta mai precis acel număr.

**EXEMPLUL 2.3**

Un computer utilizează un DAC pentru a crea o tensiune care reprezintă poziția unei antene. Antena se poate roti la  $180^\circ$  și trebuie poziționată la o distanță de  $1^\circ$ . Poate fi utilizat un port pe 8 biți?

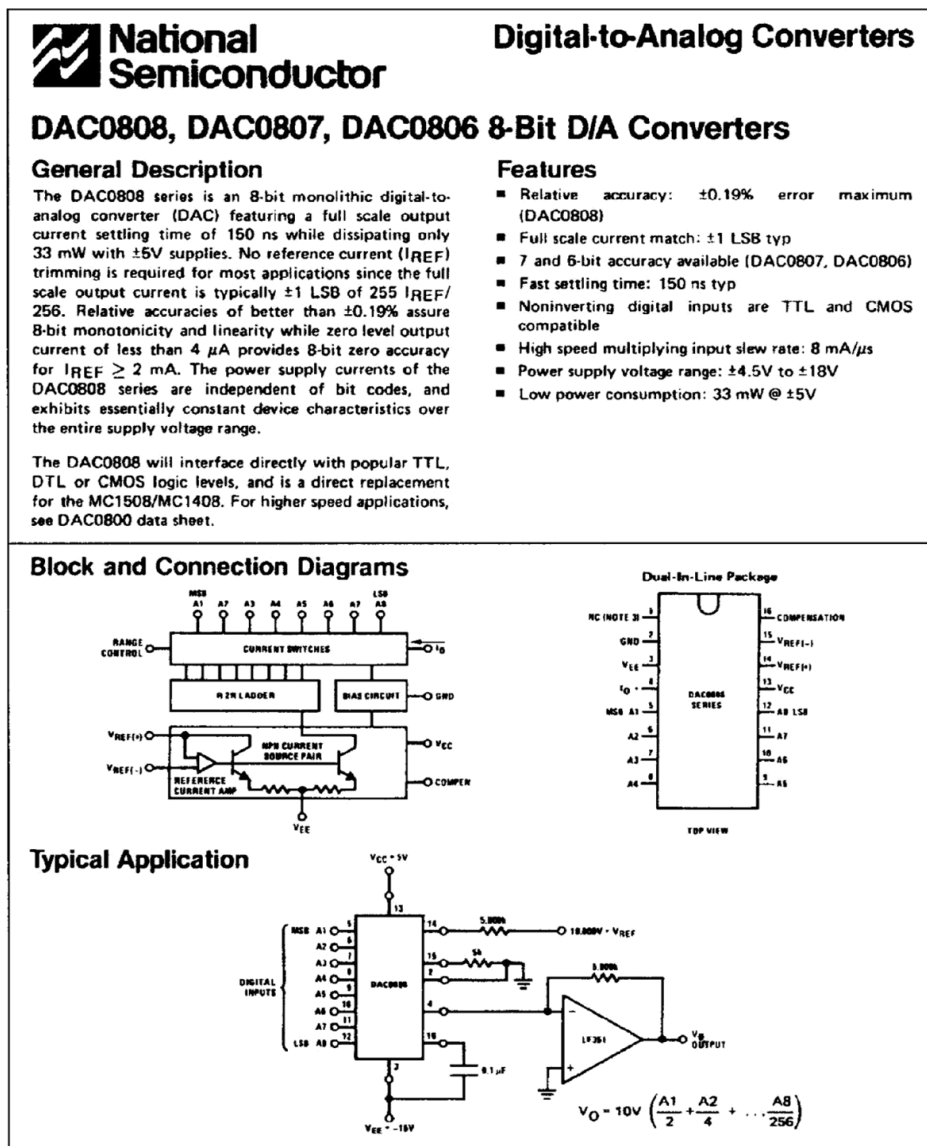
**SOLUȚIE**

Rezoluția necesară este de 1 parte din 180. Deoarece 8 biți oferă o rezoluție de 1 parte în 255, un port pe 8 biți este cu siguranță adecvat. De fapt, avem de ales: Am putea avea  $LSB = 1^\circ$ , caz în care valorile de intrare ar varia de la 0 la 180, sau am putea echivala  $180^\circ$  cu 255, ceea ce face ca  $LSB = 0,706^\circ$ . Acesta din urmă utilizează la maximum cei 8 biți pentru a oferi o rezoluție mai bună, dar dacă sistemul nu are nevoie de ea, relația clară și simplă a  $LSB = 1^\circ$  este de dorit.

Figura 2.6 prezintă o fișă tehnică pentru un DAC pe 8 biți (DAC0808). Acest dispozitiv vine ca un DIP cu 16 pini (pachet dual în linie) și utilizează un amplificator operațional extern (cum ar fi LF 351), două rezistoare și un condensator pentru a finaliza circuitul. Este nevoie de plus și minus tensiunile de alimentare cu energie electrică. Timpul pentru a finaliza o conversie este un rapid 150 ns (nanosecunde). (Circuitul prezentat în Figura 2.6 are un  $V_{ref}$  de 10 Vdc.)

**Figura 2.6**

Fișa tehnică  
pentru DAC0808,  
un convertor  
digital-analogic  
pe 8 biți.



### Conversie analogică în digitală

Un **convertor analog-digital** (ADC) este un circuit care convertește o tensiune analogică într-un cuvânt digital. Un ADC tipic constă dintr-un singur IC cu câteva componente de suport. Conversia analogică în digitală este un proces mai complicat (decât pentru DAC), iar hardware-ul necesită un timp de conversie, care este de obicei în intervalul de microsecunde. Timpul de conversie necesar depinde de tipul de ADC, de frecvența ceasului aplicată și de numărul de biți converțiți. Figura 2.7 prezintă o diagramă bloc pentru un ADC pe 8 biți. Intrarea  $V_{in}$  poate fi orice tensiune între 0 V și  $V_{ref}$ . Când  $V_{in}$  este 0 Vdc, ieșirea este 00000000; când  $V_{in}$  este  $V_{ref}$ , ieșirea este 11111111 (255 zecimal). Pentru tensiunile de intrare între 0 și  $V_{ref}$ , ieșirea crește liniar cu  $V_{in}$ ; prin urmare, putem dezvolta un raport simplu pentru ADC:

$$output/V_{in} = 255/V_{ref} \quad (\text{pentru 8 biți})$$

Rezolvarea pentru ieșire oferă următoarele:

$$Output = (V_{in} \times 255)/V_{ref} \quad (2.2)$$

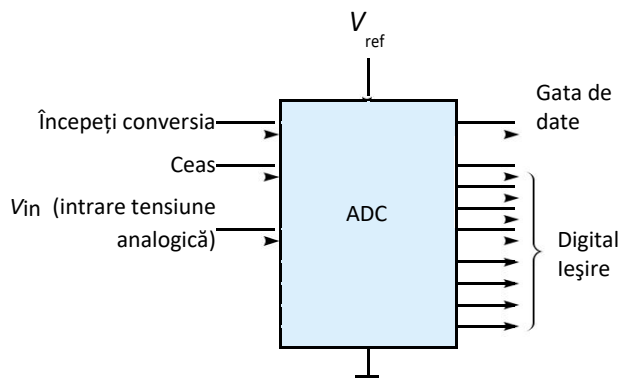
unde

$Output$  = valoarea zecimală de ieșire a unui ADC pe 8 biți

$V_{in}$  = tensiune de intrare analogică la ADC

$V_{ref}$  = tensiune de referință ADC

Pentru a începe procesul de conversie, un impuls de pornire de conversie este trimis la ADC. ADC apoi eșantionează intrarea analogică și o convertește în binar. Când procesul este finalizat, ADC activează ieșirea gata de date. Acest semnal poate fi folosit pentru a informa computerul că poate citi datele binare.



**Figura 2.7**

O diagramă bloc convertor analog-digital (ADC).

Figura 2.8

Fișa tehnică  
pentru ADC0804,  
un convertor  
analog-digital pe  
8 biți.



## Analog-to-Digital Converters

### ADC0801, ADC0802, ADC0803, ADC0804 8-Bit $\mu$ P Compatible A/D Converters

#### General Description

The ADC0801, ADC0802, ADC0803, ADC0804 are CMOS 8-bit, successive approximation A/D converters which use a modified potentiometric ladder—similar to the 256R products. They are designed to meet the NSC MICROBUS™ standard to allow operation with the 8080A control bus, and TRI-STATE® output latches directly drive the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

A new differential analog voltage input allows increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

#### Features

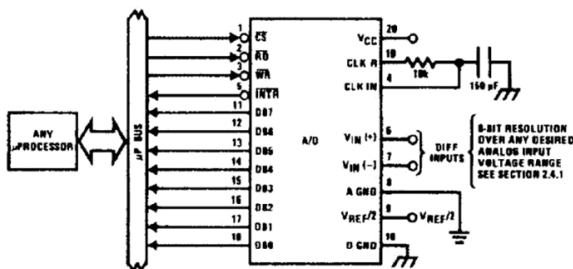
- MICROBUS (8080A) compatible—no interfacing logic needed
- Easy interface to all microprocessors, or operates "stand alone"

- Differential analog voltage inputs
- Logic inputs and outputs meet T<sup>2</sup>L voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package

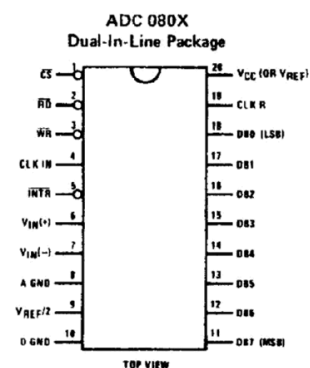
#### Key Specifications

- |  |  |
|--|--|
| ■ Resolution   | 8 bits                                       |
| ■ Total error  | $\pm 1/4$ LSB, $\pm 1/2$ LSB and $\pm 1$ LSB |
| ■ Conversion time  | 100 $\mu$ s                                  |
| ■ Access time  | 135 ns                                       |
| ■ Single supply  | 5 V <sub>DC</sub>                            |
| ■ Operates ratiometrically or with 5 V <sub>DC</sub> , 2.5 V <sub>DC</sub> , or analog span adjusted voltage reference |  |

#### Typical Applications



#### Connection Diagrams



**EXEMPLUL 2.4**

Un ADC pe 8 biți are  $V_{ref}$  de 7 Vdc; intrarea analogică este valoarea de 2,5 Vdc. Care este valoarea binară a ADC?

**SOLUȚIE**

Ieșirea este un cuvânt pe 8 biți care are o valoare zecimală maximă de 255 (decimal) atunci când  $V_{in} = V_{ref}$ . Prin urmare, o tensiune de intrare analogică ( $V_{in}$ ) de 7 Vdc ar fi convertită la 255 zecimale. Folosind acest set de date I/O, putem dezvolta un raport și apoi îl putem folosi pentru a găsi ieșirea pentru intrarea specifică de 2,5 Vdc:

$$output/V_{in} = 255/(7 \text{ Vdc})$$

Rezolvarea pentru ieșire oferă următoarele:

$$Output = 2,5 \text{ Vdc} \times 255/7 \text{ Vdc} = 91$$

Rezultatul este de 91 zecimal = 01011011 binar. Aceasta este ieșirea care ar apărea pe cele opt linii de ieșire ale ADC.

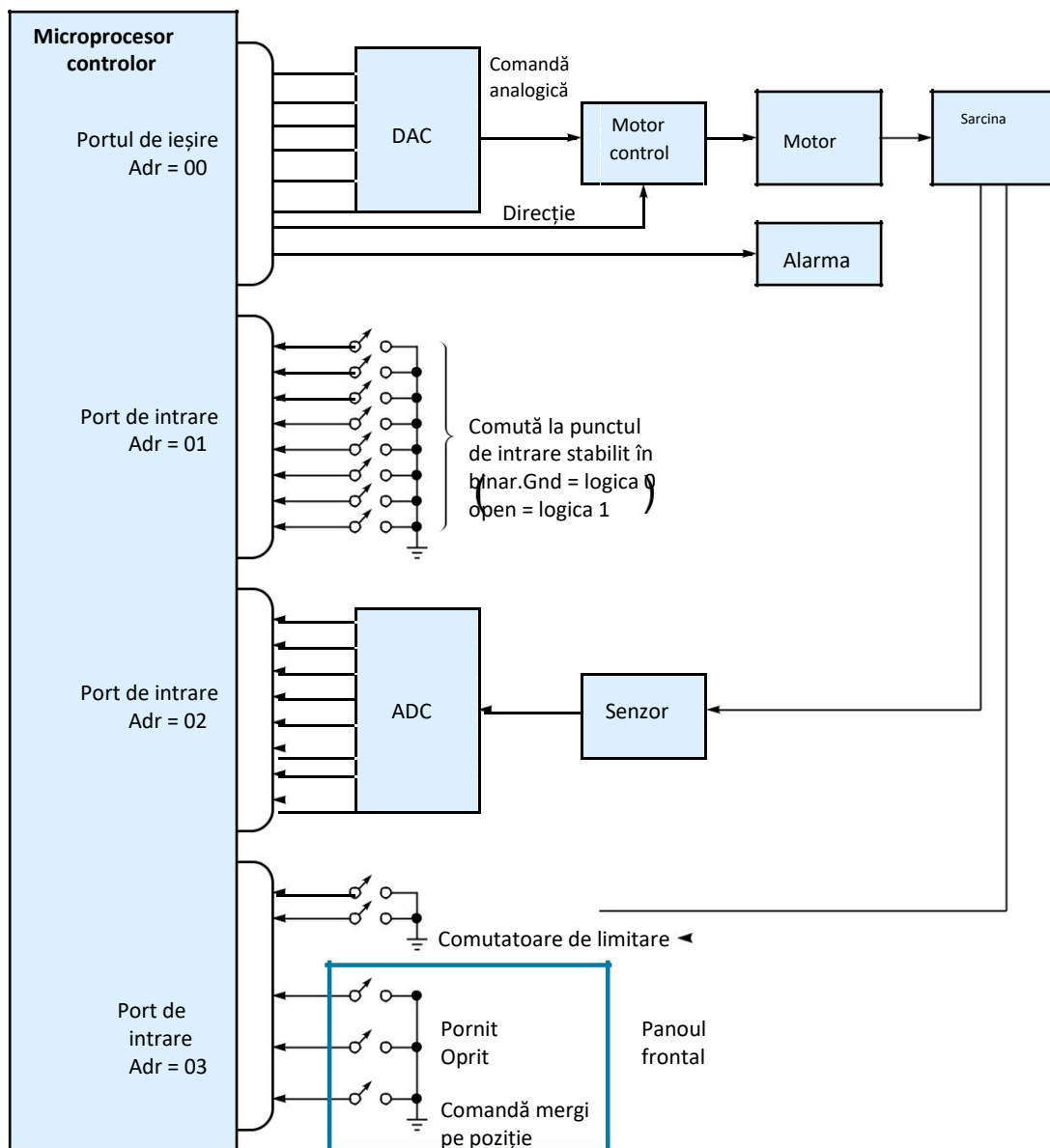
Figura 2.8 prezintă o fișă tehnică pentru un ADC pe 8 biți (ADC0804). Încapsulat ca un DIP cu 20 de pini, acest dispozitiv poate funcționa pe o singură sursă de alimentare de 5 Vdc și necesită un rezistor extern și un condensator pentru a finaliza circuitul ADC. Pulsul de start-conversion este aplicat pinului 3 (WR), iar semnalul gata de date vine de la pinul 5 (INTR). Observați, de asemenea, că PIN-ul etichetat  $V_{ref/2}$  (pinul 9) trebuie să fie setat la jumătate din *ref reale* V. De exemplu, dacă cerințele necesită o gamă de tensiune analogică de 0-5 Vdc, apoi PIN-ul 9 ar fi setat la 2.5 Vdc. Timpul pentru finalizarea unei conversii este de aproximativ 100 μs (micro-secunde), ceea ce îl face de aproape 700 de ori mai lent decât DAC0808 discutat mai devreme.

**Un sistem de control care utilizează porturi paralele**

Figura 2.9 prezintă un sistem de control al poziției care utilizează un controler bazat pe microprocesor cu porturi paralele. Acest sistem special are un port de ieșire și trei porturi de intrare (fiecare port are propria adresă). Portul de ieșire este partiționat: Șase biți sunt converțiți într-un DAC pentru a furniza semnalul analogic de acționare a motorului, al șaptelea bit specifică direcția motorului (1 = sensul acelor de ceasornic, 0 = în sens invers acelor de ceasornic), iar al optulea bit pornește o alarmă audio dacă este detectată o situație de urgență. Primul port de intrare introduce datele set-point, al doilea introduce datele ADC de la senzor, iar al treilea introduce diverse variabile logice pe 1 bit. În acest caz, sistemul are trei comutatoare pe panoul frontal, precum și două micro-întrerupătoare (micro-switch) folosite ca limitatoare de mișcare.

**Figura 2.9**

Un sistem de control folosind interfața paralelă.



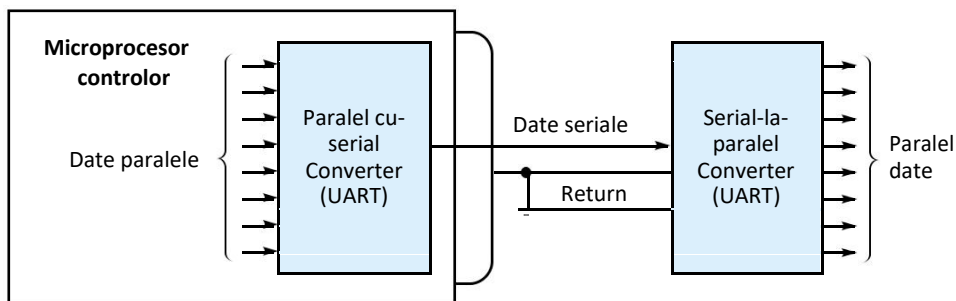
Comutatoarele limită sunt utilizate ca o "rezervă" pentru a detecta dacă sarcina a ieșit din intervalul desemnat. Funcționarea sistemului are loc după cum urmează: Controlerul introduce datele din portul 03 pentru a determina dacă butonul de pornire (sau oprire) a fost apăsat. Dacă butonul de pornire a fost apăsat, atunci punctul de setare este citit din portul 01 și datele senzorului digitalizat sunt citite din portul 02. Pe baza strategiei sale de control, controlerul transmite la portul 00 un cuvânt binar reprezentând tensiunea de control a motorului. Aceste date digitale sunt convertite la o tensiune analogică cu DAC-ul. Întreaga secvență se repetă de mai multe ori până când butonul de oprire este apăsat.

## Interfața serială

Într-o **interfață serială**, datele sunt trimise 1 bit după celălalt pe un singur fir (linie de comunicație). Există o serie de motive pentru a face acest lucru. În primul rând, cablarea este mai simplă, deoarece sunt necesare doar două fire (la un nivel minim), cele fiind "date" și "return." În al doilea rând, ecranarea unui mic grup de fire, care este adesea necesară într-un mediu industrial zgomotos electric, este mai ușoară. În al treilea rând, datele seriale pot utiliza liniile de date existente cu un singur canal, cum ar fi sistemul telefonic (care poate necesita utilizarea unui modem). Din aceste motive, transferul de date serial este de obicei recomandat pentru distanțe mai mari de 10-30 ft.

Deoarece datele există întotdeauna într-o formă paralelă în interiorul computerului, acestea trebuie să fie convertite la date seriale înainte de a ieși din portul serial. Acest lucru se realizează în special cu un convertor paralel-la-serial convertor IC numit un **transmițător universal receptor asincron (UART)**. La celălalt capăt al liniei, un receptor trebuie să convertească datele seriale înapoi în date paralele, ceea ce se face cu un alt UART. Figura 2.10 prezintă circuitul de bază al datelor seriale.

Datele seriale sunt clasificate ca fiind sincrone sau asincrone. *Datele sincrone* necesită ca octeții de date să fie trimiși ca grup într-un "pachet". Este folosit în sistemele de comunicații complexe care transmit volume mari de date și nu va fi discutat în continuare aici. *Transferul de date asincron* este tipul mai comun (dar mai lent) de transfer serial și permite trimiterea octeților individuali atunci când este necesar.



**Figura 2.10**  
Componente  
într-un circuit  
de interfață  
serială.



**Figura 2.11**

Format de date  
seriale pentru  
cuvântul binar  
10110010 (cu  
chiar paritate).

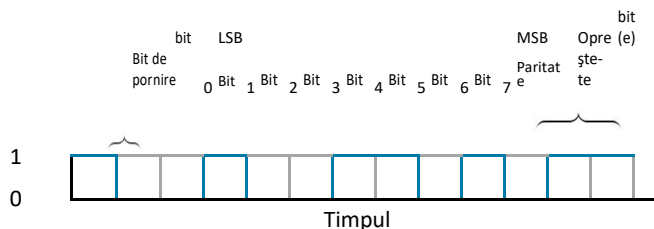


Figura 2.11 prezintă formatul standard pentru datele seriale asincrone.

În primul rând, la început este transmis un bit de STAR , apoi datele (LSB mai întâi), apoi un bit de verificare paritate-eroare, și în cele din urmă un bit de STOP.

Unele variații sunt permise la acest format, dar atât emițătorul, cât și receptorul trebuie să de biți trimiși pe secundă (frecvența acestora este denumită rata **baud**, deși termenul este bps, 2400 bps, 9600 bps, 14.400 bps 28.800 bps, 33.600 bps și 57.600 bps.

Cu toate acestea, pentru multe aplicații, în special controlul proceselor , timpii mai lungi de transfer de date nu sunt o problemă.

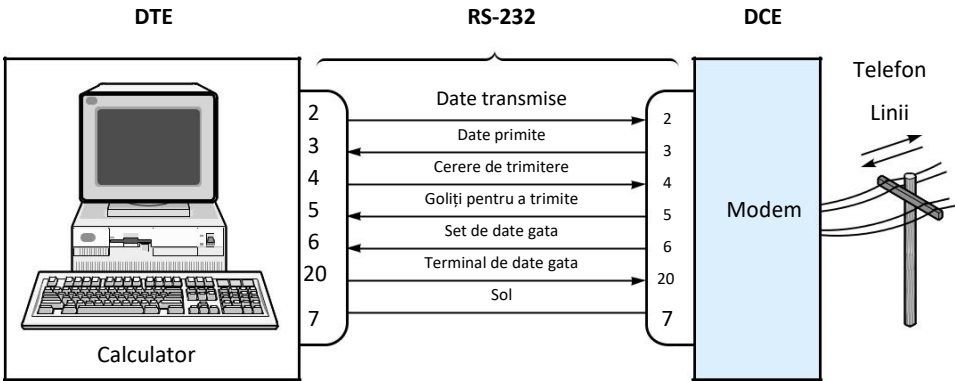
### RS 232

Pentru a face interfața serială practică, un set de specificații numit RS-232 a stabilit standardul. Oficial, **standardul RS-232** specifică datele seriale interfață între **echipamentele terminale de date** (DTE) și **echipamentele de comunicații de date** (DCE). O aplicație comună a RS-232 este interfața dintre un PC și modem, caz în care computerul este DTE și modemul este DCE [a se vedea figura nume, numere pin și tensiuni. În practică, standardul RS-232 poate fi aplicat la orice interfață serială, atâta timp cât o unitate acționează ca un DTE, iar cealaltă ca DCE. În cazul în care două

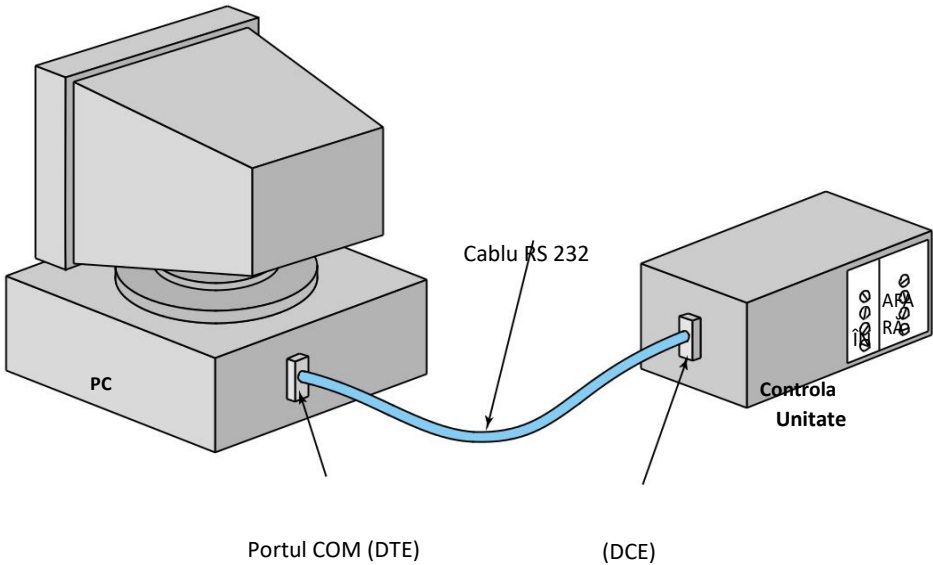
Unitățile DTE trebuie să interacționeze între ele - de exemplu, un PC cu un PC - un mod special se utilizează cablu numit **modem nul** sau **cablu crossover**. RS-232 este frecvent utilizat în câmpul de control atunci când două unități trebuie să facă schimb de date, de exemplu, pentru a conecta un PC la o unitate de control locală în scopul descărcării unui nou program de control, ilustrată în figura 2.12 (b).

Transferul serial de date RS-232 este ceva mai complicat decât transferul paralel de date, dar oferă avantaje, cum ar fi comunicațiile cu două fire și o Interfață universală acceptată. Hardware-ul pentru a gestiona datele seriale este standardizat, ușor disponibil, și de încredere.

**Figura 2.12**  
Interfața serială  
RS-232.



(a) Interfața dintre echipamentele terminale de date (DTE) și echipamentele de comunicații de date (DCE). Datele seriale sunt transferate pe pinii 2 și 3; celelalte semnale controlează fluxul de date.



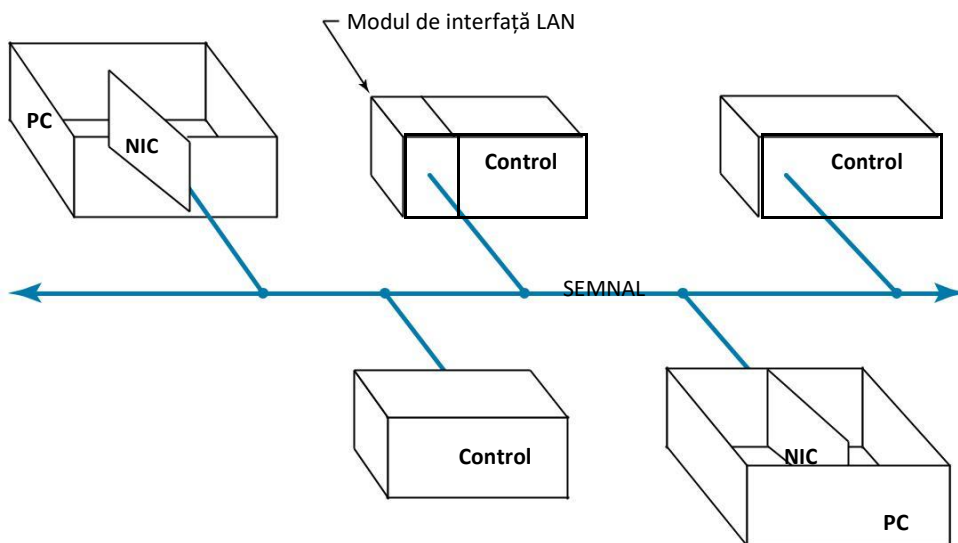
(b) Utilizarea unui cablu serial RS-232 pentru a conecta un PC la un controler

**Networking**

Probabil cea mai frecventă utilizare a datelor seriale este în rețeaua de date. Din ce în ce mai mult, rețelele de date sunt utilizate pentru a interconecta toate unitățile și dispozitivele din sistemul de control. Cablarea de rețea diferă în funcție de tipul de **rețea locală zonă (LAN)**, dar cele mai multe utilizează sistemul de magistrală prezentat în figura 2.13. De obicei, fiecare unitate de pe net are un număr unic de adresă și, de asemenea, circuite de detectare a adresei. Când o unitate vrea să vorbească cu o altă unitate, mai întâi difuzează adresa unității cu care vrea să vorbească (în serie, desigur, pe firul de semnal) și apoi trimite datele (seri-

**Figura 2.13**

Toplogia de interconectare într-un LAN.



aliat), care constau dintr-un anumit număr de octeți. Toate unitățile de pe net vor primi adresa, dar numai receptorul destinat se va activa și apoi va citi în date. Interfața dintre cablul de rețea și PC se face printr-o placă de expansiune a interfeței disponibilă din punct de vedere comercial, numită placă de interfață de rețea (NIC). Alte dispozitive de pe net, cum ar fi unitățile de control, ar necesita un circuit special de interfață, care poate fi încorporat sau disponibil ca modul extern. Rețelele de sisteme de control sunt discutate în continuare în capitolul 12.

## 2.4 INTRODUCERE ÎN PROGRAMAREA CONTROLERULUI

Este dincolo de domeniul de aplicare al acestui curs pentru a prezenta o discuție detaliată a modului de a programa un microprocesor în limbajul mașinii. Cu toate acestea, este util să investigați într-un mod general ce trebuie să facă software-ul. Un controler digital este un computer care funcționează în **timp real**.

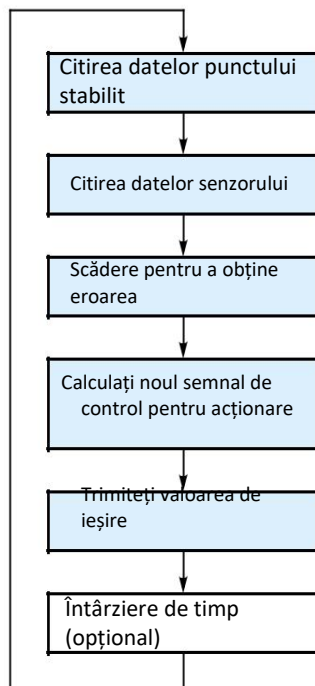
Aceasta înseamnă că *programul rulează tot timpul - luând în mod repetat cele mai noi date ale senzorului și apoi calculând o nouă ieșire pentru actuator*.

Structura de bază a unui program de controler este o *bucură*. Într-o structură de buclă, aceeași secvență de instrucțiuni este executată din nou și din nou, iar fiecare trecere prin buclă se numește **iterație** sau **scanare**. Figura 2.14 arată un program de controler generalizat și urmează o explicație a programului:

1. Programul citește în datele punctului de setare (amintiți-vă că punctul setat este poziția dorită a variabilei controlate). Aceste date pot fi citite dintr-un port de intrare sau din memorie.

**Figura 2.14**

Un program de controlor generalizat.



2. Programul direcționează computerul să citească (de la un senzor) valoarea reală a variabilei controlate.
3. Datele reale sunt scăzute din punctul setat pentru a obține eroarea.
4. Pe baza datelor de eroare, computerul calculează un nou semnal de control al actuatorului.
5. Noua ieșire este trimisă la actuator.
6. Programele sare înapoi la pasul 1 și începe din nou.

Timpul necesar pentru ca computerul să execute o trecere prin buclă determină intervalul de timp dintre citirile de intrare (cunoscut sub numele de rata de **eșantionare**). Dacă acest interval este prea lung, computerul nu poate obține o imagine exactă a ceea ce variabila controlată face cu adevărat (a se vedea capitolul 11 pentru o discuție despre aliasing). Executarea buclei poate fi accelerată prin utilizarea unui computer mai rapid sau eficientizarea programului. În alte situații, computerul trebuie să pauză și să aștepte. De exemplu, se poate introduce o pauză pentru a oferi unui operator timp pentru a efectua o anumită ajustare sau pentru a permite unui motor să se "rotească în invers". Acest lucru se face prin introducerea buclelor de întârziere în program. O **bucă de întârziere în timp** este pur și simplu o buclă de așteptare, în cazul în care computerul este instruit să numere până la un număr mare. Folosind această tehnică, putem face ca programul să se oprească pentru orice perioadă de timp - de la câteva microsecunde la ore. Dacă se introduce o buclă de întârziere în bucla principală de program (așa cum se arată în figura 2.14), efectul este de a încetini timpul ciclului pentru bucla principală. Acest lucru se face uneori pentru a forța potrivirea ratei de eșantionare la o valoare predeterminată. La un moment dat, oamenii au crezut că cele mai bune și mai eficiente programe pentru microprocesoare au fost cele scrise direct în limbajul de asamblare - adică programatorul ar selecta direct instrucțiunile de asamblare ale mașinii. Astăzi, programe sofisticate (numite

*compilatoare*) pot converti un program scris într-un limbaj de nivel înalt, în primul rând C, în limbaj mașină foarte eficient. Utilizarea unui limbaj de nivel înalt pentru a scrie programe pentru un microprocesor oferă avantaje mari, cum ar fi listarea mai compactă a programelor, ușurința de a scrie ecuații și o documentație mai ușor de înțeles. De asemenea, programele scrise într-un limbaj de nivel înalt pot fi compilate pentru a rula pe orice model de microprocesor.

## 2.5 CONTROLERE BAZATE PE MICROPROCESOARE

### Microcalculatoare cu un singur cip (microcontrolere)

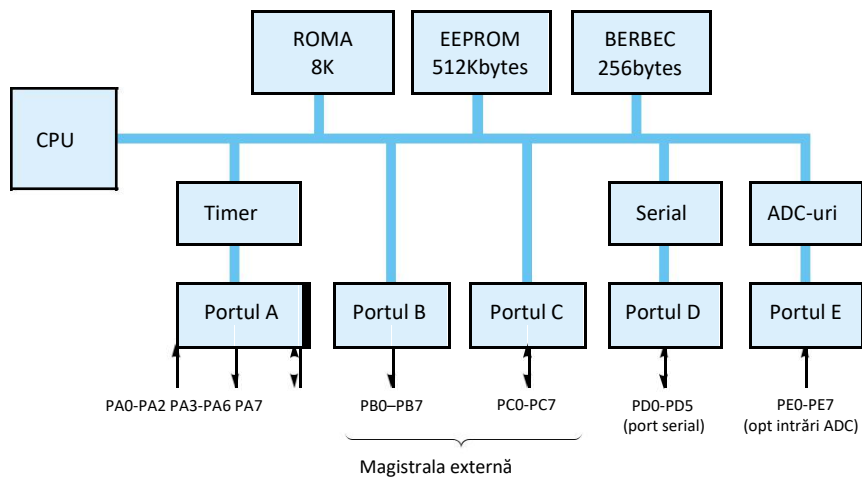
Un microprocesor în sine nu este un computer. Pentru a fi funcțional, microprocesorul trebuie conectat la alte circuite integrate care asigură memoria și capacitatea I/O. Un microcontroler este un computer pe un singur IC, conceput special pentru aplicații de control. Se compune dintr-un microprocesor, memorie (atât RAM cât și ROM), porturi I/O și, eventual, alte caracteristici, cum ar fi cronometre și ADC/DACs. Având controlerul complet pe un singur cip se permite designului hardware să fie simplu și foarte ieftin. Microcontrolerele apar din ce în ce mai mult în produse la fel de variate precum aplicațiile industriale, electrocasnicele și jucării. În astfel de utilizări, ele sunt numite **controlere încorporate**, deoarece controlerul este situat fizic în echipamentul controlat.

Principala diferență dintre microprocesoare și microcontrolere este că microprocesoarele sunt proiectate pentru a fi utilizate în microcalculatoare unde viteza mai mare și dimensiunea mai mare a cuvintelor sunt cerințe de conducere a proceselor, în timp ce microcontrolerele evoluează spre un număr redus de cipuri prin integrarea mai multor funcții hardware pe cip. Majoritatea aplicațiilor de control nu au nevoie de dimensiunea cuvântului pe 32 de biți și de viteza de 500 MHz (megahertz) a microprocesoarelor mai noi. Opt sau 16 biți și 1 MHz vor funcționa bine în multe aplicații, iar microcontrolerul cu un singur cip costă mult mai puțin.

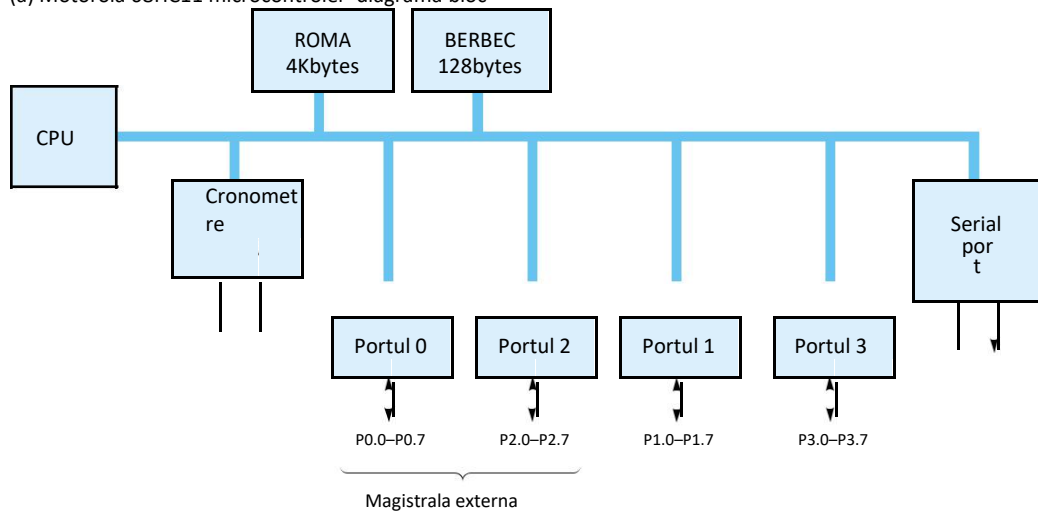
O altă diferență între microprocesoare și microcontrolere se referă la setul de instrucțiuni. Microprocesorul tinde să fie bogat în instrucțiuni care se ocupă de mutarea datelor în și din memorie. Microcontrolerul are mai puține instrucțiuni de mutare a datelor memoriei și mai multe instrucțiuni de manipulare a biților. Motivul pentru lipsa instrucțiunilor pentru memorie este că microcontrolerul are de obicei doar o cantitate mică de RAM, pe care o folosește doar ca un "tampon". Instrucțiunile suplimentare de manipulare a biților au fost incluse, deoarece sunt atât de utile în aplicațiile sistemului de control. De exemplu, într-un sistem de control, fiecare bit separat al unui cuvânt de ieșire paralel poate controla un alt dispozitiv, cum ar fi un motor sau un indicator luminos. Instrucțiunile de manipulare a biților permit software-ului să pornească sau să se dezactiveze cu ușurință un dispozitiv, fără a-i afecta pe ceilalți.

Motorola 68HC11 este un microcontroler popular pe 8 biți care are 256 octeți de RAM, 8K de ROM și octeți 512K de EPROM (a se vedea figura 2.15a). De asemenea, are cinci porturi pe 8 biți cu transfer de date serial încorporat și capacitate ADC. Un alt microcontroler comun pe 8 biți

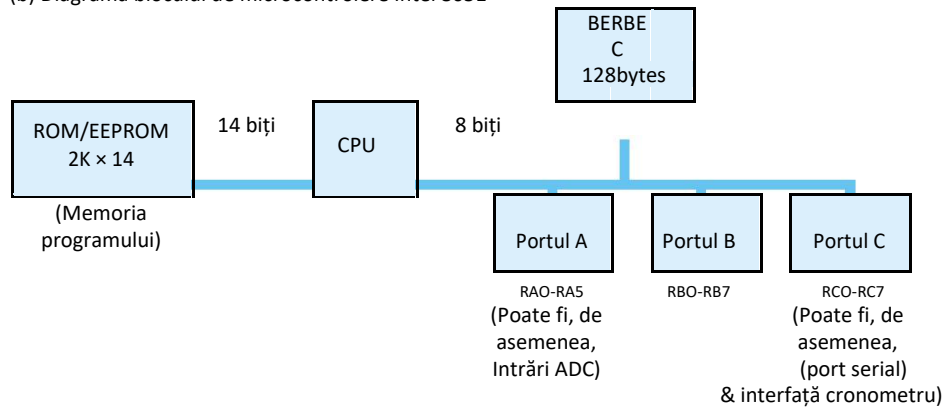
**Figura 2.15**  
Diagrame bloc  
de  
microcontrolere.



(a) Motorola 68HC11 microcontroler diagramă bloc



(b) Diagrama blocului de microcontrolere Intel 8051



(c) PIC 16C72 micro controler diagramă bloc

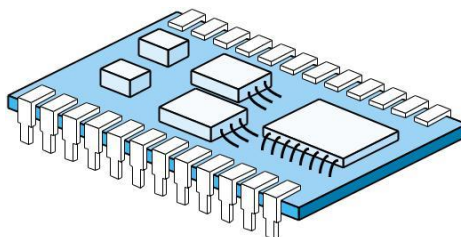
este Intel 8051, care are 128 octeți de RAM și octeți 4K de ROM, patru porturi de date paralele și un port serial (a se vedea figura 2.15b). Pentru aplicațiile de control, aceste aranjamente hardware sunt de obicei suficiente: ROM-ul este utilizat pentru a stoca programul de control, iar memoria RAM este utilizată ca registre de date și un "tampon de date". Liniile de semnal I/O pot fi de obicei conectate direct la microcontroler fără port suplimentar. Software-ul este de obicei scris în C++ sau într-un alt limbaj (inclusiv limbajul de asamblare) și apoi convertit în limbaj de mașină cu un compilator sau un program de asamblare. Programul de limbaj al mașinii va fi apoi încărcat în ROM-ul sau EPROM-ul microcontrolerului.

Un alt microcontroler popular este PIC de la Microchip Technology. De exemplu, familia PIC16Cxx de microcontrolere pe 8 biți este un produs low-cost, versatile, care a găsit o utilizare largă [a se vedea figura 2.15 (c)]. Există o gamă largă de opțiuni, inclusiv ROM, EPROM, EEPROM, ADC-uri, Cronometre și porturi seriale. PIC utilizează o arhitectură ușor diferită de 68HC11 și 8051 prin faptul că ROM-ul (sau EPROM) care conține programul se conectează la CPU cu propria magistrală pe 14 biți, în timp ce magistrala de date este de 8 biți. Folosirea a 14 biți pentru memoria programului înseamnă că toate instrucțiunile sunt doar un cuvânt. Dispozitivul are trei porturi I/O, dar mulți dintre biții I/O pot fi utilizați în moduri diferite (cum ar fi pentru un comutator de pornire/oprire sau o intrare ADC), în funcție de modul în care sunt programați.

În cele din urmă, un alt produs numit BASIC Stamp de la Parallax Inc este, de obicei, considerat un microcontroler, deși este de fapt o placă de circuit foarte mică, cu câteva CIRCUITE și porturi. Întreaga placă de circuit se conectează la o priză IC, ca și cum ar fi un IC (a se vedea figura 2.16 ).

În rezumat, o mare varietate de microcontrolere sunt disponibile. La partea cea mai de jos sunt modelele pe 4 biți, care sunt mai mult decât adecvate pentru aparate și jucării. Acestea tind să fie aplicații de volum mare, cu costuri reduse. Microcontrolerele pe opt biți (cum ar fi 68HC11 și 8051 menționate mai devreme) sunt foarte populare, deoarece 8 biți se dovedesc a fi o dimensiune convocată atât pentru datele numerice, cât și pentru cele de caracter. La partea cea mai de sus, micro-controlerele pe 16 și 32 de biți sunt disponibile pentru sistemele de control sofisticate, de mare viteză pentru aplicații precum servomecanisme complicate, avionică sau procesare a imaginilor.

**Figura 2.16**  
BASIC Stamp  
modul (BS2-IC).



## Calculatoare cu o singură placă

**Computerele cu o singură placă** sunt computere bazate pe microprocesor în afara celor construite pe o singură placă cu circuit imprimat (Figura 2.17). Ele vin în multe configurații, dar în general folosesc un microprocesor standard, cum ar fi Zilog Z80, familia Intel x86, Motorola 68000 sau un microcontroller. Acestea includ, de asemenea, circuite integrate de memorie (atât RAM cât și ROM), capacitate I/O și, probabil, circuite speciale de interfață, cum ar fi ADC sau DAC-uri. Calculatoarele cu o singură placă sunt fabricate de mari producători de microprocesoare, cum ar fi Intel și Motorola, precum și de multe alte companii mai mici. Unele computere sunt concepute pentru a se conecta la un PC ca un card de expansiune. Avantajul evident al utilizării unei plăci de microprocesor gata făcute este că elimină timpul de testare a designului și a plăcii. Acest lucru este deosebit de important în producția de volum mic sau în sistemele de un singur tip.

## Controlere logice programabile

Un **controler logic programabil** (PLC) este o unitate bazată pe microprocesor autonomă, concepută special pentru a fi un controler. PLC-ul include o secțiune I/O care se poate

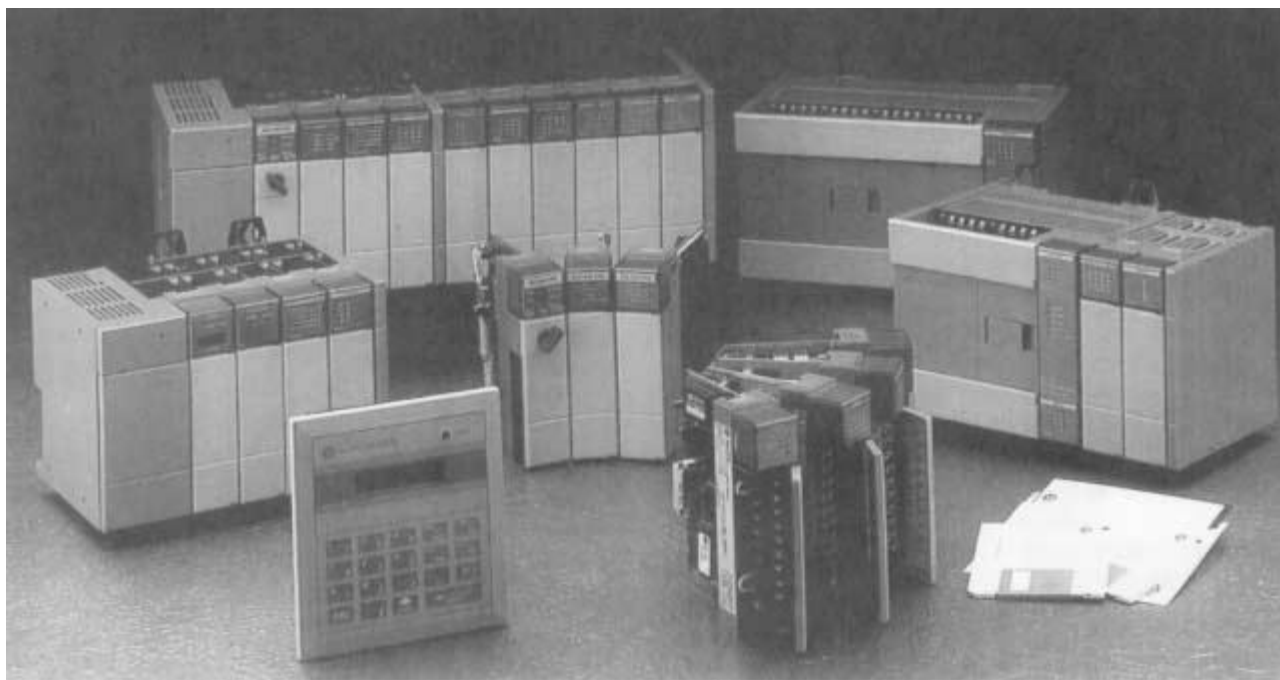
**Figura 2.17**

Un computer cu o singură placă.  
(Prin amabilitatea Vesta Technology, Inc)





interfața direct la astfel de componente ale sistemului, cum ar fi comutatoare, relee, motoare mici și lumini. Dezvoltat la sfârșitul anilor 1960 pentru a înlocui controlerele logice releu, PLC-urile au evoluat pentru a putea gestiona aplicații sofisticate de control al mișcării. PLC-urile vin în diferite dimensiuni și capacități; Figura 2.18 prezintă o selecție de PLC-uri. Diferența mare dintre PLC-uri și celelalte dispozitive discutate în această secțiune este că PLC-ul are microprocesorul, porturile și sursa de alimentare încorporate într-un pachet care a fost proiectat robust pentru un mediu industrial. Instalarea este foarte ușoară, deoarece în multe cazuri senzorii și actuatorii pot fi conectați direct la PLC. Odată instalat, programul de micro-procesor este **descărcat** în PLC de la o anumită sursă, cum ar fi un computer personal. Producătorii PLC-uri furnizează de obicei un software pentru a facilita funcționarea programului. Acest aplicații software permit utilizatorului să scrie un program cu instrucțiuni line-by-line sau poate converti direct o diagramă de cablare logică releu (diagrama scării) direct



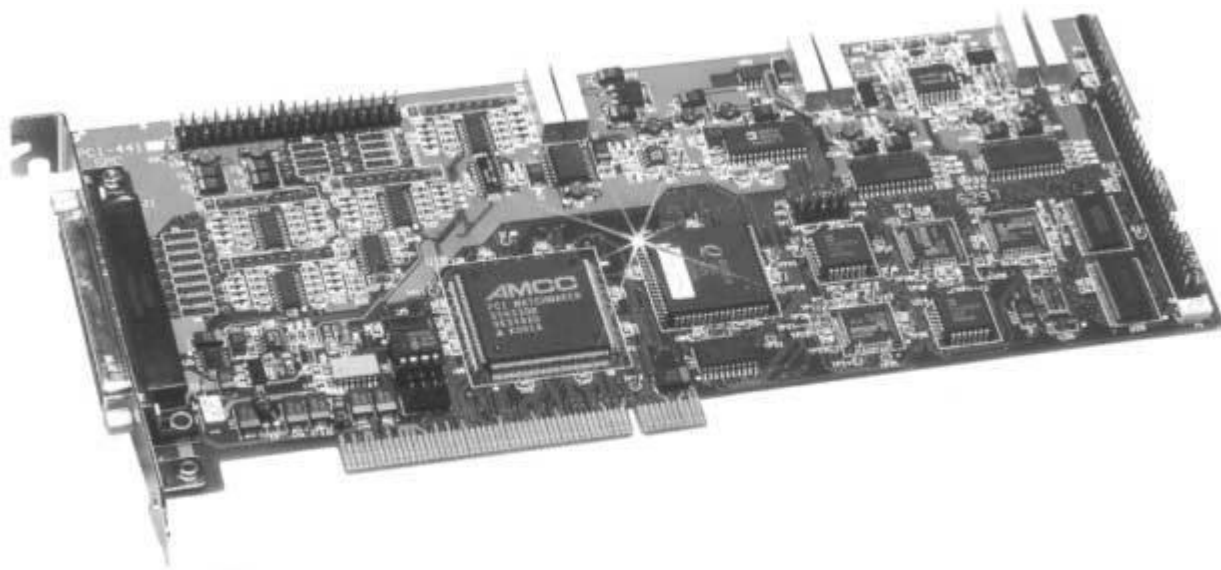
**Figura 2.18**  
Controlere logice  
programabile.

într-un program PLC. Mai multe PLC-uri dintr-o instalație pot fi interconectate, astfel încât unitățile individuale să poată fi monitorizate și programate dintr-o singură stație. Aceasta este o formă de control distribuit al calculatorului (DCC) discutată în capitolul 1. PLC-urile sunt discutate în detaliu în capitolul 12.

### Calculatoare personale utilizate în sistemele de control

Disponibilitatea **computerelor personale** (PC-uri) relativ ieftine, off-the-shelf, le-a făcut o alternativă atractivă pentru aplicațiile de control mici, unice. Pachetele software de sistem de control sunt disponibile în comerț pentru PC-ul care rulează sub DOS și Windows. Aceste programe sunt adaptabile și permit utilizatorului să adapteze software-ul pentru a se potrivi aplicației de control, transformând în esență un PC într-un PLC (deși nu la fel de robust). Cele mai multe dintre aceste pachete folosesc grafica interactivă pentru a lega animația cu valorile procesului. Unele programe au posibilități pentru a simula matematic procesul controlat pentru a ajuta la optimizarea coeficienților controlerului.

Un PC standard vine cu sloturi de **extensie**, care sunt conectori de circuit-card poziționate pe placa de *baza* (placa principală) a computerului. **Cardurile de extensie** se conectează la aceste sloturi și formează o punte între computer și lumea exterioară. Sunt disponibile multe tipuri de carduri de interfață, cum ar fi porturile de date seriale și paralele I/O,



**Figura 2.19**

Card I / O bord, include  
ADC, DAC, și porturi  
digitale I / O

ADC-uri, DAC-uri și relee de ieșire controlate de computer, pentru a numi câteva. Figura 2.19 prezintă un exemplu de card de expansiune a interfeței.

Din punct de vedere istoric, funcțiile de achiziție și control al datelor au fost procesate separat. Controlerul a rulat procesul, iar alte instrumente au măsurat și înregistrat rezultatul. Conceptul de a avea un singur PC pentru a efectua ambele sarcini pare logic; la urma urmei, PC-ul își poate folosi capacitatea de calcul mai întâi ca controler și apoi poate memora datele de performanță ale sistemului. Aceste date pot fi stocate pe disc și/sau afișate pe monitor.

O potențială problemă poate apărea deoarece controlerul trebuie să funcționeze în timp real. În cazul în care un computer trebuie să controleze un proces și să îl monitorizeze în același timp, procesul de prelucrare a datelor nu trebuie să dureze atât de mult încât să interfereze cu sarcinile de control; un răspuns de control nu poate aștepta. O modalitate de a depăși această problemă este de a împărți sarcinile de control și de achiziție a datelor între mai multe procesoare. Folosind PC-ul ca computer principal, un micro-procesor separat pe un card de expansiune poate efectua colectarea datelor neîntreruptă. Un tip de card de controler I/O are sloturi pentru trei plăci mai mici. Aceste plăci mai mici au diferite combinații de porturi I/O analogice și digitale și cronometre. Unele plăci sunt disponibile cu relee solid-state, care pot fi utilizate pentru a controla direct motoarele AC și DC.

Un PC cu carduri de expansiune I/O costă adesea mai puțin decât un sistem control computerizat de sine stătător. Cardurile nu au nevoie de o incintă separată și utilizează sursa de alimentare a PC-ului, tastatura pentru intrare și monitorul pentru afișare. De asemenea, utilizarea unui PC standard înseamnă că programele pot fi dezvoltate pe un alt computer compatibil, eliminând timpul de nefuncționare a procesului.

Numeroși producători vând PC-uri robuste care pot supraviețui în medii dure de tip industrial. Aceste computere utilizează de obicei o tastatură de tip membrană (placa de teste apare ca o foaie continuă de plastic flexibil) și au carcase sigilate și filtre care acoperă gurile de aerisire. Unele modele ale acestor computere sunt puse în rack și conțin propria sursă de alimentare de rezervă a bateriei.

---

## REZUMAT

Un microprocesor este un circuit integrat digital care efectuează operațiunile de bază ale unui computer. Microprocesoarele sunt utilizate pe scară largă ca bază a unui controler digital. Sistemele de control digital sunt avantajoase, deoarece datele digitale pot fi transferate și stocate fără erori, iar strategia de control poate fi schimbată prin simpla reprogramare.

Un computer este format din patru unități funcționale de bază: (1) CPU (microprocesor), care execută instrucțiunile programate și efectuează calculele; (2) memoria, care stochează programul și datele; (3) intrare; și (4) ieșire. Interfața de intrare /ieșire conectează un computer la lumea exterioară. Un computer bazat pe microprocesor interconectează aceste unități cu trei grupuri de semnale numite magistrale. Magistrala de adrese poartă adresa datelor care urmează să fie prelucrate. Magistrala de date transportă datele, iar magistrala de control transportă semnale de sincronizare și control. Computerele gestionează datele ca grupuri de biți binari. Multe controlere bazate pe microprocesor gestionează datele în grupuri pe 8 biți numite octet.

Un microprocesor are un set de instrucțiuni pe care le poate executa (numit setul de instrucțiuni). Fiecare instrucțiune este identificată printr-un cod digital numit codul de operare (op-code).

Un program constă dintr-o listă a acestor op-coduri stocate în memorie. Microprocesorul preia automat instrucțiunile din memorie și le execută, unul câte unul.

Un controler digital poate avea două tipuri de interfețe de date: paralele și seriale. Interfața paralelă este cel mai simplu sistem, unde toți cei 8 biți sunt trimiși în același timp pe opt fire separate. În interfața serială, datele sunt trimise 1 bit după celălalt pe un singur fir. Transferul de date serial este mai bun pentru distanțe mai lungi.

Multe sisteme de control utilizează componente care necesită o interfață de semnal analogică; prin urmare, semnalele către sau de la controlerul digital trebuie convertite cu un ADC (convertor analogic-digital) sau un DAC (convertor digital-analogic). Ambele circuite sunt disponibile sub formă de IC.

Programul de controler digital are un format standard. În primul rând, citește punctul setat și valorile senzorului. Apoi scade aceste valori pentru a determina eroarea de sistem. Pe baza valorii de eroare, acesta calculează în continuare semnalul de răspuns al actuatorului corespunzător și îl trimite. Apoi se bucle înapoi la începutul programului și execută același set de instrucțiuni în buclă.

Controlerele bazate pe microprocesoare vin într-o serie de formulare standard. Un micro-controler include un microprocesor, memorie, și de intrare / ieșire toate pe un singur IC. Un computer cu o singură placă este un computer off-the-shelf bazat pe microprocesor, asamblat pe o singură placă de circuite imprimate. Un controler logic programabil (PLC) este o unitate autonomă special concepută pentru a fi un controler. Un computer personal (PC) este un computer de uz general, autonom; cu toate acestea, odată cu adăugarea de carduri de expansiune a interfeței, un PC devine un controler foarte adaptabil și rentabil.

---

## GLOSAR

**acumulator** Un registru temporar, digital de stocare a datelor în microprocesor utilizat în multe operații matematice, logice și de mutare a datelor.

**ADC** A se vedea convertor **analogic-digital**.

**adresa** Un număr care reprezintă locația a 1 octet de date în memorie sau un port de intrare/ieșire specific.

**magistrală de adrese** Un grup de semnale care provin de la microprocesor la memorie și porturi I/O, specificând adresa.

**ALU** A se vedea unitatea **logică aritmetică**.

**unitate logică aritmetică (ALU)** Partea procesorului care efectuează operații aritmetice și logice.

**convertor analog-digital (ADC)** Un dispozitiv (de obicei un IC) care poate converti o tensiune analogică în echivalentul său binar digital.

**limba de asamblare** Un program de calculator este scris în mnemonics, care sunt abrevieri asemănătoare limbii engleze pentru instrucțiunile codului mașinii.

**baud** Rata la care stările de semnal se schimbă; frecvent utilizate pentru a însemna "biți pe secundă."

**bit** Cea mai mică unitate de date digitale, care are o valoare de 1 sau 0.

**byte** Un cuvânt digital pe 8 biți.

**unitate centrală de procesare (CPU)** Partea centrală a unui computer, CPU efectuează toate calculele și se ocupă de funcțiile de control ale computerului.

**magistrala de control** Un grup de semnale de sincronizare și control care provin de la microprocesor la memorie și porturi I/O.

**cablu crossover** A se vedea modem **nul**.

**CPU** A se vedea unitatea centrală de **procesare**.

**DAC** Vezi **convertor digital-analogic**.

**magistrală de date** Un grup de semnale care merg la și de la microprocesor, memorie și porturi I/O. Magistrala de date transportă datele reale care sunt prelucrate.

**echipamente de comunicații de date (DCE)** Una dintre cele două unități specificate de standardul RS-232 (pentru transferul de date seriale); DCE este de obicei un modem.

**echipamente terminale de date (DTE)** Una dintre cele două unități specificate de standardul RS-232 (pentru transferul de date seriale). DTE este, de obicei, computerul.

**DCE** A se vedea echipamente de comunicare a **datelor**.

**convertor digital-analogic (DAC)** Un circuit care traduce datele digitale într-o tensiune ana-log.

**descărcare** Pentru a transfera un program de calculator sau date într-un computer (de pe alt computer).

**DTE** Vezi echipamente terminale de date.

**controler încorporat** Un controler mic bazat pe microprocesor care este instalat permanent în mașina pe care o controlează.

**card de expansiune /slot** Un card de expansiune este un card de circuit imprimat care se conectează la un slot de expansiune de pe placa de bază a unui computer personal (PC). Placa de expansiune, de obicei, interfețe PC-ul la lumea exterioară.

**ciclu de preluare-executare** Un ciclu de calculator în cazul în care procesorul preia o instrucțiune și apoi o execută.

**date de intrare/ieșire (I/O)** din lumea reală care se deplasează în și din computer.

**I/O** A se vedea intrarea/ieșirea.

**set de instrucțiuni** Setul de comenzi de program pe care un anumit microprocesor este proiectat să le recunoască și să le execute.

**iterație** O trecere prin programul de calculator fiind executat de controlerul digital; fiecare iterație "citește" datele punctului de setare și senzorul și calculează ieșirea la actuator.

**LAN** A se vedea rețeaua locală.

**bit cel mai puțin semnificativ (LSB)** Partea cea mai din dreapta a unui număr binar. Poate însemna, de asemenea, cea mai mică creștere a schimbării.

**variabilă logică** Un singur bit de date în acele cazuri în care un singur bit este utilizat pentru a comanda un comutator on-off, controlul pornit al motorului și așa mai departe.

**rețea locală (LAN)** Un sistem care permite mai multor unități să comunice între ele, toate partajând același fir de interconectare.

**LSB** A se vedea cel mai puțin semnificative pic.

**limbaj mașină** Setul de coduri de operare pe care un procesor le poate executa.

**memorie** Partea computerului care stochează date digitale. Datele de memorie sunt stocate ca octeți, unde fiecărui octet i se dă o adresă.

**intrare/ieșire mapată de memorie** Un sistem în care porturile I/O sunt tratate exact ca locațiile de memorie.

**microcontroler** Un circuit integrat care include un microprocesor, memorie, și de intrare / ieșire; în esență, un "computer pe un cip."

**microprocesor** Un circuit digital integrat care efectuează operațiunile de bază ale unui computer, dar necesită unele circuite integrate de suport pentru a fi funcțional.

**cel mai semnificativ bit (MSB)** Bitul cel mai din stânga într-un număr binar.

**mnemonic** O abreviere asemănătoare limbii engleze a unui cod de operare.

**modem** Un circuit care convertește datele seriale din formă digitală în tonuri care pot fi trimise prin sistemul telefonic.

**MSB** A se vedea bit cele mai semnificative.

**memorie nevolatilă** Memorie computer, cum ar fi ROM-ul care nu va pierde datele sale atunci când alimentarea este oprită.

**modem nul** Un cablu care permite două unități DTE să comunice între ele (a se vedea RS-232).

**cod de operare (cod op)** Un cuvânt de cod digital folosit de microprocesor pentru a identifica o anumită instrucțiune.

**interfață paralelă** Un tip de interfață de date în care 8 biți intră sau părăsesc o unitate în același timp pe opt fire.

**PC** Vezi **computerul personal**.

**computer personal (PC)** Un **computer** de uz general bazat pe microprocesor, autonom (de obicei se referă la un IBM sau la un computer compatibil).

**PLC** A se vedea controler logic **programabil**.

**port** Partea unui computer unde sunt conectate linii de date I/O; fiecare port are o adresă.

**contor de program** Un registru special de deținere a adreselor într-un computer care deține adresa următoarei instrucțiuni care urmează să fie executată.

**controler logic programabil (PLC)** Un controler robust, bazat pe microprocesor autonom, conceput special pentru a fi utilizat într-un mediu industrial.

**RAM** Vezi **memoria cu acces aleator**.

**memorie cu acces aleator (RAM)** Uneori numită memorie de citire/scriere, un circuit de memorie folosind adrese în care datele pot fi scrise sau citite; MEMORIA RAM își pierde conținutul atunci când alimentarea este oprită.

**memorie doar în citire (ROM)** Similar cu RAM în care este adresabil de memorie, dar vine preprogramat și nu poate fi scris; de asemenea, nu își pierde datele atunci când puterea este oprit.

**linie de citire/scriere (R/W)** Un semnal de control care trece de la microprocesor la memorie.

**în timp real** Se referă la un computer care prelucrează date în același *timp* în care datele sunt generate de sistem.

**rezolvare** În conversia digital-analogică, eroarea care apare deoarece datele digitale pot avea doar anumite valori discrete.

**ROM** *Vedeți* **memoria doar în citire**.

**RS-232 standard** Un standard de transmisie a datelor seriale care specifică nivelurile de tensiune și protocolul de semnal între un DTE (computer) și un DCE (modem sau alt dispozitiv).

**R / W** A se vedea linia de citire / **scriere**.

**rata de eșantionare** De câte ori pe secundă un controler digital citește datele senzorului.

**scanare** A se vedea **iterație**.

**interfață serială** Un tip de interfață în care datele sunt transferate 1 bit după celălalt pe un singur fir.

**calculator cu o singură placă** Un computer pe bază de microprocesor asamblat pe un singur card cu circuit imprimat.

---

**bucă de întârziere a timpului** O tehnică de programare în care computerul primește o comandă "do-nothing", cum ar fi numărarea până la un număr mare în scopul întârzierii timpului.

**UART** *Vezi* **transmițător receptor asincron universal**.

**transmițător receptor asincron universal (UART)** Un circuit integrat cu scop special care convertește datele din format paralel în format serial și invers.

**memorie volatilă** Memorie computer, cum ar fi RAM, care își va pierde datele atunci când alimentarea este oprită.

**cuvânt** O unitate de date digitale pe care o utilizează un anumit computer; dimensiunile comune ale cuvintelor sunt de 4, 8, 16 și 32 de biți.

---

## EXERCIȚII

### Secțiunea 2.1

1. Descrieți pe scurt funcțiile ALU, unitatea de control, memoria procesorului și intrarea /ieșirea.
2. Ce măsuri ia microprocesorul pentru a citi datele la adresa 1020? (Specificați acțiunile magistralei de adrese și ale magistralei de date din răspunsul dvs.)
3. Definiți pe scurt *magistrala de adrese*, *magistrala de date* și *magistrala de control*.
4. Utilizați metoda prezentată în exemplul 2.1 pentru a găsi valoarea zecimală a numărului binar 01011101.
5. Utilizați metoda prezentată în exemplul 2.1 pentru a găsi valoarea zecimală a numărului binar 11011010.

### Secțiunea 2.2

6. Ce este un set de *instrucțiuni* microprocesor și cum este diferit de un limbaj de nivel înalt, cum ar fi BASIC?
7. Un anumit microprocesor are un set simplu de instrucțiuni prezentat mai jos.

#### SET DE INSTRUCȚIUNI

<u>După cod</u>	<u>Explicație</u>
76	Opriți microprocesorul.
C6*	Adăugarea octetului următor la acumulator.
D6*	Subtract următorul octet de la acumulator.
3C	Incrementați acumulator.
3D	Decrementează acumulator.
3E*	Mutarea următorului octet în acumulator.

\*Aceleste instrucțiuni folosesc doi octeți.

Ce număr ar fi în acumulator după ce programul prezentat mai jos a fost rulat?



## PROGRAM

<u>Adresa</u>	<u>Op-code</u>
001	3E
002	05
003	D6
004	02
005	3C
006	76

## Secțiunea 2.3

8. Valorile de temperatură de la  $-20^{\circ}\text{F}$  la  $120^{\circ}\text{F}$  sunt date de intrare pentru un microprocesor. Sunt 8 biți suficienți pentru a reprezenta valorile analogice? Dacă da, care este rezoluția?
9. Explicați funcția următoarelor: portul de *date paralel* și portul de *date seriale*.
10. Datele seriale sunt trimise la 1200 bps folosind formatul figurii 2.11, cu un singur bit de oprire. Cât timp ar dura pentru a trimite 1000 octeți de date?
11. Un DAC pe 8 biți are o tensiune de referință de 9 V. Intrarea binară este 11001100. Găsiți tensiunea de ieșire analogică.
12. Datele binare de pe computer într-o anumită aplicație sunt de așteptat să meargă de la 000000000 la numai 00111111. Aceste date sunt introduse unui DAC. Ieșirea analogică ar trebui să meargă 0-5 V. Găsiți tensiunea de referință DAC necesară pentru a face acest lucru.
13. Un ADC pe 8 biți are o tensiune de referință de 12 V și o intrare analogică de 3,7 V. Găsiți ieșirea binară.
14. Ieșirea binară a unui ADC ar trebui să aibă intervalul 000000000-11111111 corespundând la o intrare de 0-6 V. Găsiți tensiunea de referință necesară.

## Secțiunea 2.4

15. Ce este *calculul în timp real* și de ce este necesar pentru sistemele de control?
16. Descrieți pașii de bază dintr-o scanare a programului de control (bucă).
17. La un moment dat în program se dorește ca computerul să aștepte 5 s pentru un răspuns al operatorului. Cum s-ar realiza această întârziere în software?
18. Un program conține 150 de instrucțiuni, iar timpul mediu de execuție per instrucție este de 2  $\mu\text{s}$ . Găsiți rata de eșantionare a acestui program.

## Secțiunea 2.5

19. Ce este un *microcontroller* și care sunt unele diferențe între un microcontroller și un microprocesor?
20. Ce este un *controller logic programabil*?
21. Doriți să utilizați un computer personal pentru a controla un braț robot simplu. Brațul are două articulații, un cot și o încheietură. Fiecare îmbinare are un motor de curent continuu și un senzor de poziție care generează o tensiune dc. Aveți deja un PC ; faceți o listă cu ceea ce ar trebui să achiziționați pentru a face acest sistem să funcționeze.
22. Comparați următoarele: un *microprocesor*, un *microcontroller*, un *controller logic programabil* și un *computer personal*.