Tema 4: **PROGRAMAREA DISPOZITIVELOR I/E ÎN APLICAțII TIMP - REAL**

§4.1. Structura hardware generala a calculatorului de proces

Gestiunea comunicarii aplicației timp-real cu dispozitivele I/E presupune utilizarea unor tehnici de programare specifice, destinate sa sincronizeze operațiile de I/E cu prelucrarile interne. Daca de exemplu datele sunt citite înainte ca un dispozitiv sa le poata furniza, atunci rezultatul este eronat. De asemenea, daca aplicația nu preia datele suficient de rapid, dispozitivul periferic poate sa le înlocuiasca cu altele noi fara ca acest lucru sa poata fi depistat. Prin urmare, este necesar ca în fluxul prelucrarilor sa existe secvențe de program care sa realizeze sincronizarea operațiilor de I/E cu programele de aplicație (detectarea daca un periferic este gata sa furnizeze/primeasca date si sa realizeze transferul acestora).

Pentru sincronizarea comunicarii între aplicație si echipamentele periferice exista 2 tehnici de baza:

• aplicația interogheaza periodic dispozitivele (polling) citind registrele de stare pentru a determina când se poate comunica cu acesta;
• dispozitivele periferice întrerup procesorul pentru execuția unor proceduri specifice de comunicare.

4.1.1 Metoda Polling

Procesorul interogheaza dispozitivele periodic, citind unul sau mai multe registre de stare a caror valoare permite procesorului sa decida când dispozitivul este pregatit pentru comunicare. Daca dispozitivul solicita serviciile procesorului, este apelata o rutina de tratare specifica, în caz contrar procesorul fie continua interogarea fie executa alte prelucrari.

Bucla de polling poate sa fie implementata în doua moduri:

1. Asteptare în bucla de test pâna când dispozitivul este gata si apoi transfer de date.
2. Daca dispozitivul nu este gata în momentul interogarii se continua cu alte prelucrari (inclusiv interogarea altor dispozitive), iar când este gata se face transferul de date.

Desi polling este cea mai simpla metoda de comunicare, prezinta unele dezavantaje:

• Aplicația trebuie sa fie capabila sa execute întreaga bucla suficient de rapid pentru a putea ține cont de toate cerințele perifericelor.
• În funcție de încarcarea sistemului, o bucla polling poate fi suficient de rapida în anumite condiții de funcționare a dispozitivelor, iar în alte condiții poate fi extrem de lenta (de exemplu daca se executa multe sarcini într-un anumit pas al buclei sau daca sunt interogate mai multe dispozitive si toate cer servicii simultan).
• Când complexitatea programului creste ca urmare a introducerii de noi prelucrari, o bucla polling, care la origine lucra bine, poate deveni prea lunga.
• Daca se cere executarea unor acțiuni cu o baza de timp prestabilita, o bucla polling nu permite întotdeauna ca aceasta sa fie controlata pentru a asigura precizia adecvata. Daca baza de timp se calculeaza ținând cont de durata de execuție a instrucțiunilor masina, schimbarea tactului procesorului implica modificarea programului.

4.1.2 Metoda întreruperilor externe

Prin aceasta metoda dispozitivul atenționeaza aplicația generând întreruperi externe. Tratarea prompta a întreruperilor de la toate dispozitivele este posibila atât timp cât cererile catre procesor sunt rezonabile, acesta este capabil sa lanseze rapid procedurile (rutinele) de tratare iar timpul de execuție al acestora este suficient de mic. În particular, metoda întreruperilor externe este de preferat în aplicații care cer precizie pentru timpul de achiziție de date si control, în timp ce procesorul executa si alte sarcini. De asemenea, este utila daca mai multe dispozitive solicita asincron servicii, la intervale de timp nepredictibile. Pentru programarea întreruperilor sunt necesare cunoasterea sistemului de întreruperi al calculatorului si modul de programare al acestuia. De asemenea, este necesara cunoasterea limbajului de asamblare al procesorului sau cel puțin cuvintele cheie si procedurile speciale pentru întreruperi furnizate de anumite medii de dezvoltare pentru limbaje evoluate. (Unele limbaje precum C/C++, PASCAL, ADA furnizeaza mijloace de tratare a întreruperilor cu proceduri care realizeaza citirea/scrierea prin adresarea absoluta a locațiilor de memorie, operații de intrare/iesire la nivel fizic prin programarea porturilor sau locațiilor de memorie, secvențe de intrare si de iesire din procedurile de tratare a întreruperilor). Succesiunea execuției instrucțiunilor de catre procesor depinde de apariția evenimentelor care declanseaza proceduri de tratare a întreruperilor. Greselile în tratarea sistemului de întreruperi al calculatorului provoaca erori grave în execuția programului si afecteaza negativ funcționarea sistemului, ceea ce face ca programele care utilizeaza întreruperile sa fie greu de depanat.

4.1.3 Metoda Intrari/iEsiri buffer-ate

Aceasta metoda îmbina avantajele polling-ului cu avantajele întreruperilor. Este utilizata în aplicații precum bucle de control sau în aplicații de supraveghere, unde datele sunt prelucrate si afisate în paralel cu achiziția lor.

Implementarea uzuala a *intrarilor bufferate* consta în:

• o rutina de tratare a întreruperilor care citeste (achiziționeaza de la un dispozitiv extern) datele si le memoreaza într-un buffer circular;

• un task scris ca o bucla polling în care aplicația asteapta datele iar atunci când sunt în memorie le prelucreaza.

Contorul de inserare în buffer este gestionat de rutina de tratare a întreruperilor si indica mereu urmatoarea locație în care trebuie depuse datele achiziționate. Contorul este incrementat circular.

Contorul de extragere este gestionat de task-ul care efectueaza în polling prelucrarea datelor din buffer. Permanent, task-ul principal compara cele doua contoare: daca acestea sunt diferite, exista date noi între locațiile indicate de contorul de extragere (care indica cea mai veche data depusa în buffer si netratata) si contorul de inserare – 1. daca nu exista date noi procedura polling fie asteapta date, fie efectueaza alte prelucrari. Trebuie sa existe un echilibru între rata de achiziție si viteza de prelucrare astfel încât buffer-ul sa nu conțina mai mult decât câteva intrari noi, iar contorul de inserare sa nu depaseasca pe cel de extragere (circular).

Iesirile buffer-ate pot fi tratate similar. Task-ul principal depune datele într-un buffer circular (container), pe masura ce ele sunt prelucrate, iar rutinele de tratare a întreruperilor le extrag si le trimit catre dispozitivul de iesire. În acest caz, *contorul de inserare* este menținut de task-ul principal, iar cel de *extragere* de rutina de tratare a întreruperii. Daca datele trebuie transferate catre dispozitivul extern la intervale de timp egale (de exemplu pentru generarea unor forme de unda), task-ul trebuie sa fie capabil sa depuna date suficient de rapid în buffer, pentru ca acesta sa nu fie vid atunci când se executa rutina care transfera o noua data catre dispozitiv.

4.1.4 Alegerea strategiei optime

Trebuie subliniat ca utilizarea întreruperilor nu este întotdeauna mai buna decât polling-ul, acesta oferind avantaje semnificative în cazul anumitor aplicații.

Cea mai buna strategie pentru achiziție foarte rapida de date (fara alte prelucrari în paralel) este utilizarea polling-ului cu bucle scurte scrise în limbaj de asamblare, care testeaza continuu un registru de stare al perifericului. Perifericul este servit imediat ce condiția este îndeplinita îi testarea începe din

nou. Spre deosebire de bucla generala polling descrisa anterior, aceasta bucla nu executa si alte operații. Daca precizia timpului de transfer este considerata de mare importanța si daca bucla polling scurta nu permite controlul acestuia, atunci acesta trebuie controlat de o baza de timp externa. Aceasta se poate realiza prin impulsuri generate de un numarator programabil sau un generator de frecvența (realizat de exemplu cu I8254, AMD 9513 etc. - vezi 4.3).

Pe de alta parte, întreruperile sunt o buna modalitate de a controla achiziția de date atunci când rata de achiziție este suficient de scazuta si exista timp disponibil suficient între întreruperi. De asemenea întreruperile sunt mai avantajoase decât polling-ul daca programul trebuie sa comunice asincron cu mai multe dispozitive sau daca programul trebuie sa execute mai multe task-uri în paralel cu sarcinile de achiziție de date si comenzi. Multe din aplicațiile în întreruperi sunt cel mai bine tratate daca se utilizeaza I/E buffer-ate. Utilizarea întreruperilor permite procesorului sa execute task-uri precum înregistrarea datelor pe disc, citirea datelor de pe disc, execuția funcțiilor de control, afisarea pe display si / sau la imprimanta, dialogul cu operatorul etc. concurent cu achiziția de date si comenzi. Alegerea strategiei optime este determinata prin urmare de cerințele aplicației si de resursele hard/soft disponibile. Înainte de a decide daca se utilizeaza întreruperi sau polling, trebuie sa se estimeze supraîncarcarea întreruperilor (salvarea/restaurarea registrelor alterabile) care depinde printre altele si de procesor si de frecvența tactului folosit de calculator. Pentru un PC limitele practice în utilizarea întreruperilor pentru transferuri de date cu dispozitive CAN / CNA poate varia între câteva esantioane/sec. pâna la mii de esantioane/sec., depinzând de strategia aleasa si de celelalte task-uri pe care sistemul trebuie sa le execute în paralel.

4.2 Programarea utilizând întreruperile

Tehnica programarii în întreruperi presupune în primul rând cunoasterea sistemului de întreruperi al calculatorului utilizat. În cele ce urmeaza se reamintesc unele noțiuni referitoare la sistemul de întreruperi al PC si se prezinta detalii privind modul de programare al acestuia în modul real al procesoarelor x86.

4.2.1 Prezentarea sistemului de întreruperi la PC

Pentru tratarea întreruperilor, programul trebuie sa conțina rutine speciale de tratare a acestora (*interrupt handlers*). Adresele acestor rutine se plaseaza în locații speciale ale memoriei, astfel încât ele sa poata fi executate la apariția întreruperilor.

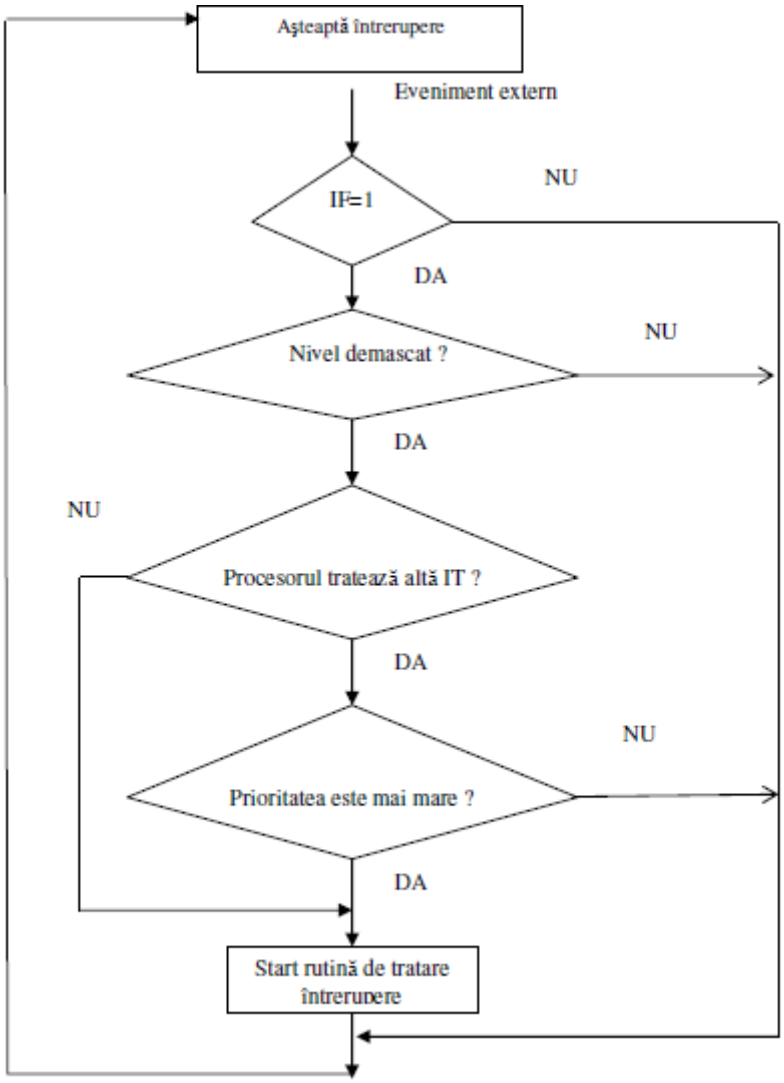


Fig. 4.2.1-1Automatul de lansare IT externe la I8259A

La PC, gestiunea întreruperilor externe mascabile (vezi si 4.2.3) este realizata de catre doua dispozitive I8259A, cascadate. Caracteristicile de baza ale I8259A sunt:

• gestioneaza 8 niveluri de întreruperi, fiecare având asociat un nivel de prioritate (0 cel mai prioritar, 7 cel mai puțin prioritar);

• întreruperile cu prioritate mai mare pot întrerupe execuția rutinelor de tratare a unor întreruperi mai puțin prioritare;

• fiecare nivel de întrerupere poate sa fie mascat sau demascat;

• I8259 pot fi cascadate pentru a creste numarul de întreruperi.

Întreruperile externe mascabile pot fi dezactivate prin program executând instrucțiunea *disable interrupt* (cli), care reseteaza Interrupt Flag (IF) din PSW. Daca procesorul primeste întreruperi când acestea sunt dezactivate, nu va raspunde pâna când prin program nu se executa instrucțiunea *enable interrupt* (sti), care seteaza IF. Daca o întrerupere exista memorata în momentul execuției sti, procesorul efectueaza operațiile pentru lansarea rutinei de tratare a întreruperii.

I8259A identifica automat sursa de întrerupere externa si daca nivelul pe care este conectata aceasta nu este inhibat (nu este mascata sau nu este în curs de execuție o rutina de tratare întrerupere mai prioritara) activeaza semnalul Interrupt Request de la procesor. Procesorul, daca IF este setat (întreruperile sunt autorizate) memoreaza starea întreruperii, termina normal instrucțiunea în curs de execuție la apariția întreruperii (procesorul raspunde la întreruperi între instrucțiuni) si activeaza semnalul de acceptare a întreruperii (interrupt acknowledge). La detectarea acestui semnal, controllerul comunica procesorului *numarul vectorului de întrerupere* asociat întreruperii (pe 8 biți), pe baza caruia procesorul identifica *adresa primei instrucțiuni executabile din rutina de tratare* si o încarca în CS:IP. Algoritmul clasic de lansare a unei rutine de tratare, implementat în controllerele care gestioneaza întreruperile externe la PC, este prezentat în figura 4.2.1-1.

Adresa primei instrucțiuni executabile a rutinei de tratare a întreruperii este memorata în prealabil (prin programul utilizator sau de catre sistemul de operare) în *tabela vectorilor de întrerupere*, în locațiile rezervate pentru intrarile de întrerupere ale controllerelor I8259A.
Tabela vectorilor de întrerupere este zona de memorie dintre adresele fizice [0,3ffh] si conține 256 de vectori a 4 octeți fiecare. În fiecare vector se memoreaza valorile care trebuie încarcate în registrele IP si CS pentru a adresa prima instrucțiune a rutinei de tratare. Primii 2 octeți din vector sunt pentru IP (în ordinea low - high) iar urmatorii 2 pentru CS (ordinea low - high).

Pentru lansarea în execuție a rutinei de tratare a întreruperii, procesorul efectueaza urmatoarele operațiuni:
• salveaza în stiva (vârful stivei este indicat de SS:SP) conținutul registrelor care indica adresa instrucțiunii urmatoare (CS si IP) si cuvântul de stare al procesorului (PSW);
• activeaza semnalul de acceptare a întreruperii (interrupt acknowledge);
• încarca în registrele de adresa (CS si IP) adresa rutinei de tratare a întreruperii din tabela vectorilor de întrerupere.

La lansarea rutinei de tratare a întreruperii, procesorul reseteaza IF, ceea ce înseamna ca în timpul execuției acesteia întreruperile mascabile sunt dezactivate. Totusi, IF poate fi setat prin program în rutina de tratare.

Ultima instrucțiune din rutina de tratare a întreruperii este *interrupt return* (iret), care este similara cu instrucțiunea ret. Adresa instrucțiunii urmatoare si starea originale sunt restaurate (în CS:IP si respectiv PSW) si procesorul continua cu execuția instrucțiunii urmatoare celei la care a fost întrerupt.

Ori de câte ori controllerul genereaza o întrerupere, întreruperile de prioritate mai mica sunt automat inhibate pâna când *prin program este achitat*a *întreruperea* (vezi 4.2.3) a carei rutina este în curs de execuție. Cererile de întreruperi pe niveluri de prioritate mai mica sunt totusi memorate de controller si se declanseaza mecanismul de cerere de întrerupere la procesor atunci când sunt îndeplinite condițiile de prioritate. Întreruperile cu prioritate mai mare pot întrerupe rutinele de tratare a întreruperilor cu prioritate mai mica daca acestea din urma au executat instrucțiunea sti (au activat sistemul de întreruperi externe mascabile). Daca acestea nu au executat instrucțiunea sti ele nu pot fi întrerupte deoarece, dupa cum s-a aratat, *la lansare în execuție a rutinei de tratare se dezactiveaz*a *automat sistemul de întreruperi*.

La PC (sub DOS), *BIOS inițializeaz*a *la startare controllerele I8259 s*i *completeaz*a *o parte dintre vectorii din tabela de întreruperi cu adresele rutinelor de tratare;* este indicat ca programele de sistem si utilizator sa țina cont de aceasta inițializare. Sistemul de întreruperi externe este programat astfel încât pentru întreruperile de la 0 la 7 (de la controllerul 1) sa fie lansate în execuție rutinele ale caror adrese de start se gasesc în vectorii de la 08h la 0fh, iar pentru întreruperile de la 8 la 0fh (de la controllerul 2) sa fie lansate în execuție rutinele ale caror adrese de start se gasesc în vectorii de la 70h la 77h. *Adresa fizic*a *de memorie pentru adresa memorat*a *în vectorul de întrerupere n este n*4. (Ex: pentru vectorul 8 adresa de memorie este 20h)*.

O rutina de tratare a unei întreruperi externe poate fi lansata (de obicei pentru testare sau simulare) si prin program, cu instrucțiunea int n..
Exemple (vezi si 4.2.2):
• int 9 – lanseaza în execuție rutina de tratare a întreruperii pentru tastatura;
• int 8 – lanseaza în execuție rutina de tratare a întreruperii pentru ceasul de timp real;
• int 0BH - lanseaza în execuție rutina de tratare a întreruperii pentru COM2.

4.2.2 Plasarea I8259 în spațiul de I/E la PC

La PC, porturile de I/E de la 0 la OFFH sunt rezervate pentru a fi utilizate pe placa de baza a calculatorului. Porturile de la 100h pâna la 3FFH sunt utilizate de catre dispozitivele conectate pe magistrala I/E a calculatorului. Porturile cu adrese mai mari de 400h nu sunt disponibile pe placa de baza în sistem.

Asignarea uzuala a porturilor la dispozitive este urmatoarea:
• de la 0 la 0Fh - porturile pentru controller-ul DMA1, 4 canale I8237;
• de la 20h la 21h - primul controller I8259A pentru întreruperi, la care sunt asignate urmatoarele întreruperi externe mascabile:

- pe canalul 0 - întreruperea de la ceasul de timp real;
- pe canalul 1 - întreruperea de la tastatura;
- pe canalul 2 - la AT cascadare catre al doilea controller I8259;
- pe canalul 3 - întrerupere de la COM2;
- pe canalul 4 - întrerupere de la COM1;
- pe canalul 5 - întrerupere de la a doua imprimanta LPT2 sau placa de rețea;
- pe canalul 6 - întrerupere de la controller-ul floppy discului;
- pe canalul 7 - întrerupere de la prima imprimanta LPT1.

• de la 40h la 43h - porturile pentru dispozitivul numarator/periodizator I8254; un registru de control si 3 numaratoare de 16 biți sunt disponibile pe acest dispozitiv:
- Numaratorul 0 - ceasul de timp real;
- Numaratorul 1 - este utilizat pentru reîmprospatarea memoriei;
- Numaratorul 2 - este utilizat pentru difuzor.

• de la 60h la 64h la AT - controller-ul pentru tastatura;
• de la 70h la 71h - pentru accesul la CMOS si ceasul de timp real din CMOS;
• de la 80h la 8Fh - regiunea de pagina DMA;
• de la A0h la A1h - adresele pentru al 2-lea controller de întreruperi I8259 la AT; la acest controller, sunt
asignate urmatoarele întreruperi externe mascabile:
- pe canalul 0 - întreruperea de la ceasul de timp real din CMOS (utilizata de sistemul de operare OS/2);
- pe canalul 1 - rezervat, la AT preia funcția canalului 2 de la primul controller care este utilizat pentru cascadare;
- canalele 2,3,4 - disponibile pentru întreruperi de la alte dispozitive decât cele sistem;
- canalul 5 - rezervat pentru coprocesorul matematic;
- canalul 6 - rezervat pentru controller-ul de hard disc;
- canalul 7 - disponibil;

• C0h-CFh - al doilea controller DMA;
• F0h-F1h - coprocesor matematic;
• 170h-177h - al 2-lea hard disc;
• 1F0h-1F7h - primul hard disc;
• 200h-207h - games;
• 278h-27Fh - interfața paralela pentru LPT2;
• 2E8h-2EFh - porturi pentru COM4;
• 2F8h-2FFh - porturi pentru COM2;
• 370h-377h - al doilea controller al floppy discului;
• 378h-37Fh - porturi pentru LPT1;
• 3E8h-3EFh - porturi pentru COM3;
• 3F0h-3F7h - primul controller floppy;
• 3F8h-3FFh -porturi pentru COM1;

4.2.3 Detalii de programare utilizând întreruperile la PC cu SO DOS

Se reaminteste faptul ca la microprocesoarele din familia 80x86 întreruperile pot fi *interne* sau *externe*.

Întreruperile interne sunt generate fie de instrucțiunile software (int n), fie de procesor în anumite situații (depasire aritmetica, TF poziționat etc).

Întreruperile externe sunt generate de hardware si pot fi:
• nemascabile (pe intrarea NMI la procesor), sau
• mascabile (prin intermediul controller-elor I8259);

Întreruperile nemascabile întrerup secvența curenta de instrucțiuni indiferent de situație.

Întreruperile mascabile întrerup secvența curenta de instrucțiuni numai daca IF=1, daca bitul asociat în registrul masca al I8259 (vezi mai jos) este pus la 0 si daca nu este în curs de tratare o alta întrerupere externa mai prioritara.

În continuare sunt prezentate aspecte practice de utilizare a sistemului de întreruperi externe la PC, *dupa ce BIOS a executat procedura de inițializare.*

În practica, se întâlnesc doua modalitați de utilizare a întreruperilor:

1. apelarea procedurilor pe care BIOS-ul sau alte programe/drivere/TSR-uri le pun la dispoziție;
2. capturarea vectorului de întrerupere si depunerea în tabela vectorilor de întrerupere a adresei unei rutine de tratare utilizator.

Lansarea în execuție a rutinelor de tratare, în ambele cazuri, se face în funcție de tipul întreruperii:

• prin program cu instrucțiunile int pentru întreruperi software;
• declansat de catre semnale externe/interne cablate pe nivelurile întreruperilor externe;

Rutinele de tratare asociate întreruperilor externe pot fi apelate si cu instrucțiuni int, de obicei pentru testare si depanare.

Apelul prin program cu instrucțiunea int presupune urmatorii pasi:

1. salvarea registrelor procesorului (daca este cazul);
2. încarcarea în registrele procesorului sau în anumite locații de memorie a unor valori solicitate de rutina de tratare a întreruperii;
3. executarea instrucțiunii int (int n);
4. tratarea iesirilor (daca exista) din rutina de tratare a întreruperii;
5. refacerea registrelor salvate la primul pas.

În funcție de specificul rutinei de tratare a întreruperilor anumiți pasi pot fi omisi (de exemplu pentru rutinele care nu afecteaza registrele sau indicatorii de condiții nu sunt neaparat necesari pasii privind salvarea si restaurarea registrelor si indicatorilor). În cazul în care programul apeleaza propria rutina de tratare a întreruperii este necesar sa fie tratate toate aspectele referitoare la salvarea si restaurarea registrelor si indicatorilor, pentru a nu provoca efecte colaterale dezastruoase pentru funcționarea sistemului.

Pentru capturarea unui vector de întrerupere, în program trebuie sa apara urmatoarele secvențe:

• salvarea în memorie (de obicei în segmentul curent de date sau cod) a adreselor rutinelor de tratare curente, care urmeaza a fi înlocuite cu rutine utilizator ;
• depunerea în tabela vectorilor de întreruperi a adresei rutinei de tratare utilizator;

Observații:

1. Exista situații în care rutina de tratare a întreruperii utilizator, înainte sau dupa efectuarea unor prelucrari specifice aplicației, trebuie sa apeleze si rutina inițiala de tratare a întreruperii (daca de exemplu în BIOS este prevazuta o astfel de rutina) – vezi si exemplul de la 4.2.4.
2. La sfârsitul programului scris de utilizator trebuie ca adresele originale ale rutinelor de tratare utilizate în program sa fie refacute în tabela vectorilor de întrerupere, în caz contrar aparând efecte imprevizibile în funcționarea sistemului.

Atunci când utilizeaza întreruperile mascabile externe, programatorul trebuie sa trateze în mod adecvat urmatoarele:

• *activarea/dezactivarea* sistemului de întreruperi mascabile;
• *mascarea/demascarea* nivelurilor de întrerupere la controllerul I8259;
• registrele procesorului trebuie *salvate* la începutul execuției rutinei de tratare si *restaurate* la sfârsitul acesteia;
• *achitarea* întreruperii externe în curs de execuție;
• *comunicarea* (sincronizarea) cu programul/programele utilizator pe care rutina îl/le deserveste.

Sistemul de întreruperi mascabile *se activeaza* dupa execuția instrucțiunii sti sau dupa execuția unei instrucțiuni popf în care indicatorul IF din cuvântul extras din stiva este 1 si *se dezactiveaza* cu instrucțiunea cli sau daca se executa instrucțiunea popf în care indicatorului IF în cuvântul extras din stiva este 0.

Cât timp sistemul de întreruperi este activat, pot avea acces la procesor numai întreruperile care *nu sunt mascate* în octetul de masti din controllerul I8259. La primul controller de întreruperi registrul de mascare este la adresa 21h, iar la cel de-al 2-lea la adresa 0A1h. În aceste registre, pentru fiecare întrerupere exista câte un bit de mascare.

Registrul masca de la adresa 21h:

bit: 7 6 5 4 3 2 1 0

| …… | | |
IRQ7 IRQ2 IRQ1 IRQ0
Registrul masca de la adresa 0A1h :
bit: 7 6 5 4 3 2 1 0
| ….. | | |
IRQF IRQA IRQ9 IRQ8

Daca bitul corespunzator IRQn este egal cu 0, nivelul de întrerupere este demascat; daca este egal cu 1, nivelul de întrerupere este mascat si întreruperea asociata nu este tratata de catre controller.

Exemplu: Mascarea/demascarea unei întreruperi pe nivelul 1 (pentru IRQ1)
IRQ1-demascare:
in al,21h
mov oldmask1,al
and al,0FDh
out 21h,al
IRQ1-mascare:
in al,21h
mov oldmask1,al
or al,2
out 21h,al
Refacerea mastii inițiale se face cu secvența de instrucțiuni:
mov al,oldmask1
out 21h,al
Exemplu: Mascarea/demascarea unei întreruperi pe nivelul 9 (pentru IRQ9)
IRQ9-demascare:
in al,a1h
mov oldmask2,al
and al,0FDh
out a1h,al
out a1h,al
IRQ9-mascare:
in al,a1h
mov oldmask2,al
or al,2
out a1h,al

Refacerea mastii inițiale se face cu secvența de instrucțiuni:
mov al,oldmask2
out a1h,al

Atunci când prin hardware se lanseaza în execuție o rutina de tratare a întreruperii, se *salveaz*a automat în stiva (în aceasta ordine): IP, CS si PSW ; structura PSW pentru modul de lucru real al procesorului este urmatoarea:
15 ……. 0
| |
x x x x O D I T S Z x A x P x C
unde: x nu conteaza iar O, D, I,T,S,Z, A, P, C sunt indicatorii; I este Interrupt Flag. Prin program, trebuie *salvați* (cu instrucțiuni push) registrii procesor care sunt utilizați în rutina - pentru siguranța, este bine sa fie salvați toți.

La terminarea rutinei de tratare, se *restaureaz*a toți registrii salvați la început (cu instrucțiuni pop, *în ordinea invers*a *salv*a*rii*) apoi se executa iret. Când se executa instrucțiunea iret, se refac automat PSW,CS,IP.

• În timpul execuției rutinei de tratare, dupa efectuarea prelucrarilor specifice pentru nivelul de prioritate al întreruperii, trebuie facuta *achitarea* întreruperii externe, cu urmatoarele secvențe de instrucțiuni:
; pentru primul I8259:

```
mov al,20h
out 20h,al
; pentru la 2-lea I8259:
mov al,20h
out 0a0h,al
```

Daca întreruperea nu este achitata, nu vor mai putea fi lansate ulterior rutinele de tratare cu prioritate mai mica sau egala cu a întreruperii neachitate.

Deoarece programul în curs de execuție nu cunoaste când apar întreruperile, este necesar ca rutina de tratare si acesta *sa comunice/sa se sincronizeze*; de regula comunicarea/sincronizarea se realizeaza prin intermediul unor zone comune de memorie (memorie partajata). Accesul la aceste zone trebuie facut cu *interblocare*, pentru a evita erorile.

Exemplu: un program pe 16 biți, modul real partajeaza cu o rutina de tratare a întreruperii un întreg long (pe 4 octeți), a carui scriere/citire se face prin doua instrucțiuni procesor. Atunci când programul utilizator citeste întregul fara interblocare, exista pericolul ca între cele doua instrucțiuni de citire sa se declanseze rutina de tratare care modifica valoarea; în aceasta situație, programul utilizator va citi partea low din vechea valoare si partea high din noua valoare. Evitarea acestei situații se face utilizând mecanisme de interblocare puse la dispoziție de catre sistemul de operare sau implementate în program. Cea mai simpla metoda, este ca accesul în scriere/citire din programul utilizator la zonele comune sa se faca cu secvența:

```
cli ; dezactiveaza întreruperi
…
Acces la zona comuna
…
sti ; activeaza întreruperi
```

În general, în rutina de tratare nu trebuie sa fie executate apeluri sistem sau subprograme comune cu programele utilizator, daca acestea nu sunt reentrante. De asemenea, accesul la porturi, zone comune de memorie sau alte resurse comune cu sistemul de operare sau programele utilizator trebuie sa se faca cu interblocare.

4.2.4 Exemple de utilizare a întreruperilor

Având ca sistem de operare MS DOS, sa se elaboreze un program în limbaj de asamblare care la fiecare 10 întreruperi de la ceasul de timp real sa afiseze pe monitor caracterul ‘A’. O întrerupere de la ceasul de timp real al PC apare de aproximativ 18.2 ori pe secunda (la aprox. 55 ms) – vezi si 4.3.4. Oprirea programului sa se poata face fie la acționarea tastelor CTRL+BREAK, fie dupa ce s-a afisat de 1000 ori caracterul ‘A’.

Pentru a scrie acest program este necesar sa fie capturați vectorii de întrerupere de pe nivelul 8, unde este depusa adresa rutinei de tratare pentru IRQ0 si de pe nivelul 23h asociat instrucțiunii int 23, unde este depusa adresa rutinei de tratare pentru CTRL+BREAK. Pentru afisarea pe monitor a caracterului ‘A’ se va folosi funcția DOS 09h iar pentru oprirea normala a programului se va folosi funcția DOS 4ch a întreruperii software int 21h.

Detalii privind funcțiile DOS/BIOS utilizatepot fi furnizate de programul « Help » disponibil la laborator.

```
dosseg
.model small
.stack 200h
.data
intț8țip dw ?
intț8țcs dw ?
intț23țip dw ?
intț23țcs dw ?
msg db 'A$'
timer dw 0 ;contor
```

```
zece dw 10
oldțmask db ?
.code
savețds dw ?
programțceas proc near
mov ax,@data
mov ds,ax
mov cs:savețds,ax ; salvare ds
xor ax,ax
mov es,ax
;salvare adresele BIOS pentru rutinele de tratare pentru nivelurile 8h si
;23h
mov ax,es:(8*4)
mov intț8țip,ax
mov ax,es: (8*4+2)
mov intț8țcs,ax
mov ax,es: ( 23h*4 )
mov intț23țip, ax
mov ax,es: ( 23h*4+2 )
mov intț23țcs , ax
;Se dezactiveaza sistemul de întreruperi pentru a nu apare
;accidental întreruperi in timp ce se fac modificari in tabela
;vectorilor de întreruperi
cli
;Depune in tabela adresele rutinelor de tratare utilizator
mov word ptr es: ( 8*4 ) , offset ițțceas
mov word ptr es: ( 8*4+2 ) , cs
mov word ptr es: ( 23h*4 ) , offset ctrlțbreak
mov word ptr es: ( 23h*4+2 ) , cs
;demascare nivelul 0 la primul I8259
in al,21h
mov oldțmask , al
and al, 0FEh
out 21h, al ; demascare nivel 0.
; activeaza sistemul de întreruperi mascabile
sti
;aici incepe bucla de prelucrare in programul utilizator
mov si,timer
repeta:
mov dx, 0
mov ax, timer
cmp ax, si
jz repeta ;daca nu s-a modificat contorul, repeta
mov si,ax
div zece
cmp dx, 0
jnz repeta; numai la 10 tick-uri se face afisarea
mov dx,offset msg
mov ah,9 ; functia DOS de afisare sir de caractere terminat cu '$'
int 21h ; afiseaza 'A'
mov ax,timer
cmp ax, 1000
jc repeta
call far ptr restorețvect ;restaureaza vectorii initiali
mov ah,4ch ;termina programul
```

```
int 21h
programțceas endp
itțceas proc far ; rutina de tratare I.T ceas
push ax
push ds
push bx
push cx
push dx
push si
push di
push bp
push es
mov ax,cs:savețds
mov ds,ax ;reface ds al programului asociat rutinei
inc timer
pop es
pop bp
pop di
pop si
pop dx
pop cx
pop bx
pop ds
pop ax
jmp dword ptr intț8țip
itțceas endp
ctrlțbrk proc far ; rutina de tratare CTRL+BREAK
push ax
push ds
push bx
push cx
push dx
push si
push di
push bp
push es
mov ax,cs:savețds
mov ds,ax
xor ax,ax
mov es,ax
call far ptr restorețvect
pop es
pop bp
pop di
pop si
pop dx
pop cx
pop bx
pop ds
pop ax
jmp dword ptr intț23țip
ctrlțbrk endp
restorețvect proc far ; restaurare vectori initiali si masca la
iesirea din program…
cli
```

```
mov ax,intţ8ţip
mov es:(8*4),ax
mov ax,intţ8ţcs
mov es:(8*4+2),ax
mov ax,intţ23ţip
mov es:(23h*4),ax
mov ax,intţ23ţcs
mov es:(23h*4+2),ax
mov al,oldţmask
out 21h,al
sti
ret
restoreţvect endp
end programţceas
```

Observaţii:

1. Rutinele de tratare pentru întreruperea de la ceas (itţceas) si pentru tratarea acţionarii tastelor ctrlţbreak (ctrlţbrk) se termina cu instrucţiunea jmp la adresa vectorului originar de tratare a întreruperii, salvat în programul utilizator. Acest lucru este necesar deoarece exista în sistemul de operare rutine care trateaza aceste întreruperi, în cadrul tratarii efectuându-se operaţii absolut necesare pentru sistem – de exemplu oprirea motorului de la floppy disc este gestionata în rutina de ceas originala. Nu toate întreruperile au proceduri de tratare în cadrul BIOS sau sistemului de operare. În astfel de situaţii modul de terminare depinde de tipul întreruperii: daca întreruperile sunt externe si mascabile, atunci înainte de terminare se face achitarea întreruperii în curs de tratare si apoi se executa instrucţiunea iret; pentru celelalte tipuri de întreruperi terminarea se face pur si simplu cu iret; În cazul de faţa daca sistemul de operare nu ar avea rutina pentru întreruperea de la ceas instrucţiunile:

```
pop ax
jmp dword ptr intţ8ţip
```

ar trebui înlocuite cu:

```
mov al,20h;
out 20h,al
pop ax
iret
```

Pentru rutina de tratare acţionarea CTRL+BREAK terminarea s-ar putea face înlocuind

```
jmp dword ptr intţ23ţip
```

cu iret.

2. Întreruperea unui program DOS din execuţie se poate face utilizând CTRL+C sau CTRL+BREAK. Prin utilizarea comenzii BREAK în CONFIG.SYS sau ca o comanda obisnuita DOS, sistemul de operare poate fi determinat sa verifice acţionarea tastelor respective în timpul oricarui apel sistem (BREAK=ON) sau sa faca verificarea numai în timpul apelurilor sistemului pentru operaţiile de I/E cu echipamente periferice standard (BREAK=OFF). În program se foloseste astfel apelul sistem 09H pentru afisarea unui sir de caractere iar în timpul execuţiei acestui apel se face testul de acţionare CTRL+BREAK.

Programul de mai sus se poate scrie si în limbajul C/C++, PASCAL etc. fie prin utilizarea funcţiilor speciale pentru întreruperi, existente în bibliotecile mediilor de dezvoltare, fie prin utilizarea programarii mixte limbaj evoluat / limbaj de asamblare. O versiune a programului scrisa în Borland C/C++ este urmatoarea:

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#define INTRţ8 0X08 /* Întreruperea de la ceas, la aprox. 55ms*/
#define INTRţ23 0X23 /* Întreruperea CTRLţBREAK */
#ifdef ţţcplusplus
#define ţţCPPARGS ...
#else
```

```
#define ţţCPPARGS
#endif
unsigned char oldmaskţ1=0;
void interrupt ( *oldint8)(ţţCPPARGS);
void interrupt ( *oldint23)(ţţCPPARGS);
int count=0,oldcount=0,ctrlţbk=0;
void interrupt ceas(ţţCPPARGS){
/* incrementeaza contorul */
count++;
/* achitarea întreruperii */
outportb(0x20,0x20);
}
void interrupt ctrlţbrk(ţţCPPARGS){
ctrlţbk=1;
}
void main(void){
/* salveaza vechii vectori de întrerupere */
oldint8 = getvect(INTRţ8);
oldint23 = getvect(INTRţ23);
/* dezactiveaza sistemul de întreruperi mascabile
*/
asm cli;
/* seteaza noii vectori de întrerupere */
setvect(INTRţ8, ceas);
setvect(INTRţ23,ctrlţbrk);
/*salveaza masca pentru primul 8259*/
oldmaskţ1=inportb(0x21);
/*demascheaza întreruperea 0*/
outportb(0x21,oldmaskţ1 & 0x0fe);
/* activeaza sistemul de întreruperi mascabile
*/
asm sti;
/* bucla de prelucrare a programului principal */
clrscr();
while ((count<1000)&&(!ctrlţbk)){
if(count!=oldcount)
if((count % 10)==0)
{
printf("A”);
oldcount=count;
}
}
/* dezactiveaza sistemul de întreruperi mascabile
*/
asm cli;
/* seteaza vechii vectori de întrerupere */
setvect(INTRţ8, oldint8);
setvect(INTRţ23,oldint23);
/*seteaza vechea masca pentru primul 8259*/
outportb(0x21,oldmaskţ1);
/* activeaza sistemul de întreruperi mascabile
*/
asm sti;
}
```

Se remarca:

• utilizarea cuvântului cheie interrupt, care indica compilatorului sa genereze automat secvențe de salvare/restaurare registre si execuție iret la iesire;
• utilizare funcții getvect/setvect pentru salvare/capturare vector de întrerupere;
• utilizare funcții inportb/outportb pentru citire/scriere din/în porturi pe un octet;
• pentru activarea/dezactivarea sistemului de întreruperi s-au utilizat instrucțiuni în limbajul de asamblare
(asamblare in line); se pot însa folosi în acest scop si funcțiile enable() pentru activare si disable() pentru dezactivare întreruperi.

Probleme propuse

1. Sa se modifice programul programțceas astfel încât oprirea sa se realizeze dupa 1000000 de întreruperi de la ceas (timer pe 4 octeți).
2. Sa se modifice programul programțceas si rutina ittceas astfel încât la fiecare 10 întreruperi de la ceasul de timp real programul sa afiseze „tack" iar rutina de tratare a întreruperii de la ceas sa afiseze la fiecare apel „tick".

Indicație: În programul utilizator se va folosi pentru afisarea sirului „tack" apelul sistem 9 al int21h, iar în rutina de tratare a întreruperii de la ceas se va folosi subfuncția 0eh de la int 10h (BIOS) pentru afisarea caracterelor din sirul „tick". Aceasta, deoarece apelul la sistem prin int 21h *nu este reentrant*, utilizarea unor astfel de apeluri în rutinele de tratare a întreruperilor putând duce la blocarea calculatorului!

3. Sa se rezolve problema utilizând pentru întreruperea de la ceas vectorul 1ch (int 1ch) – detalii despre int 1ch se gasesc în programul HELP disponibil la laborator.
4. Program care implementeaza intrari bufferate cu funcțiile:
• Rutina de tratare a întreruperii de la ceas genereaza într-un buffer de 1000 octeți (coada circulara) o secvența de caractere ASCII 1,2,…,9 formata dintr-un numar de caractere egal cu modulo 10 al numarului de întreruperi care au aparut de la lansarea programului.
• Programul utilizator afiseaza pe monitor caracterele noi, însoțite de nr. de întreruperi memorate în contor.
• Programul utilizator si rutina de tratare a întreruperii trebuie sa detecteze eventualele umpleri de buffer respectiv golire de buffer, în vederea sincronizarii.
5. Program pentru iesiri bufferate (buffer circular de 1000 octeți). Programul utilizator pune în buffer date citite dintr-un fisier text linie cu linie, la intervale multiplu de aproximativ 0,1 sec. specificate interactiv de utilizator. Rutina de întrerupere de la ceas afiseaza liniile nou introduse. Programul utilizator trebuie sa detecteze umplerea de buffer si sa atenționeze printr-un mesaj afisat pe ecran, iar rutina de întrerupere trebuie sa detecteze golirea buffer-ului si sa atenționeze sonor (beep).
6. Dezasamblați programul scris în limbajul C/C++; ce remarcați?

4.3 Dispozitive pentru generarea bazei de timp si numararea de evenimente

4.3.1 Dispozitivul mumarator/periodizator I8254

I8254 conține (figura 4.3.1-1):
• Magistrala de date bidirecționala, tree-state pe 8 biți.;
• Logica de citire/scriere (semnale: RD, WR, A0, A1, CS);
• Registru de control (numai de iesire);
• 3 numaratoare pe 16 biți programabile sa funcționeze independent. Numararea se face descrescator. Oricare numarator poate fi programat sa lucreze în unul dintre cele 6 moduri de lucru disponibile.
Transferurile de date între dispozitiv si magistrala calculatorului se fac pe 8 biți si depind de modul în care se face comanda scrierilor sau citirilor prin intermediul registrelor de control.
• Intrarile/iesirile asociate numaratoarelor sunt:
• intrari de tact (clock) (CLKi , i e [0,2]); pe fiecare intrare de tact pot fi impulsuri cu frecvența maxima de
8MHz;
• intrari de poarta GATEi , i e [0,2], utilizate pentru triggerarea intrarilor de clock;
• iesiri OUTi , i e [0,2].
În figura 4.3.1-2 se prezinta schema bloc interna a unui numarator. Intern, dispozitivul conține pentru fiecare numarator 3 registri:

• registrul CR (Count Register) pe 16 biți – la programare datele sunt înscrise în acest registru, iar la startarea numararii ele sunt încarcate în registrul CE în care efectueaza numararea. CRM este pentru octetul MSB, iar CRL pentru octetul LSB;
• registrul CE (Counting Element) – registrul de numarare;
• OL (Output Latch) – "Latch" în care se memoreaza la o anumita comanda valoarea curenta instantanee conținuta în registrul CE. Datele memorate pot fi citite ulterior. OLM este pentru octetul cel mai semnificativ, iar OLL pentru octetul cel mai puțin semnificativ.
Fiecare numarator se încarca inițial, iar dupa startarea numararii se decrementeaza pe frontul descrescator al
impulsului de tact.
Triggerarea se face astfel:
• GATEi =1 numararea autorizata
• GATEi =0 numararea inhibata
Nivelul logic al iesirii OUT i depinde de modul de lucru ales.

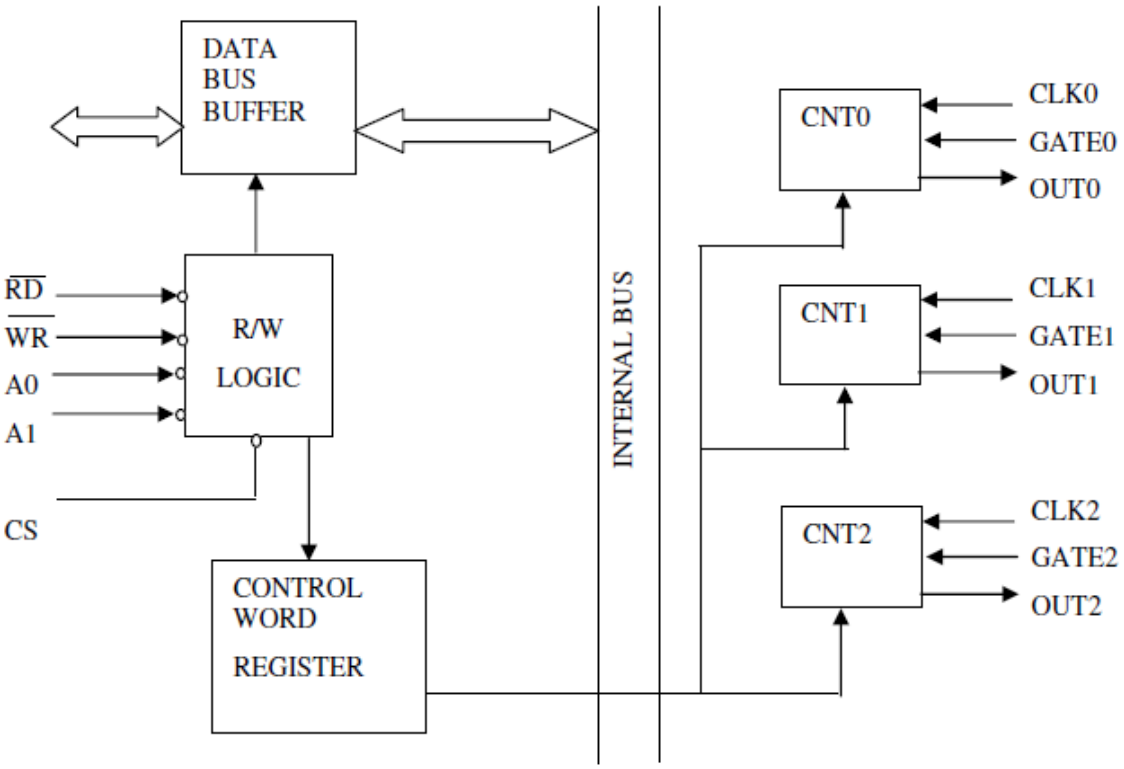


Fig. 4.3.1 – 1 Schema bloc a dispozitivului I8254 (conform catalog Intel)

Nivelul logic al iesirii OUT i depinde de modul de lucru ales.

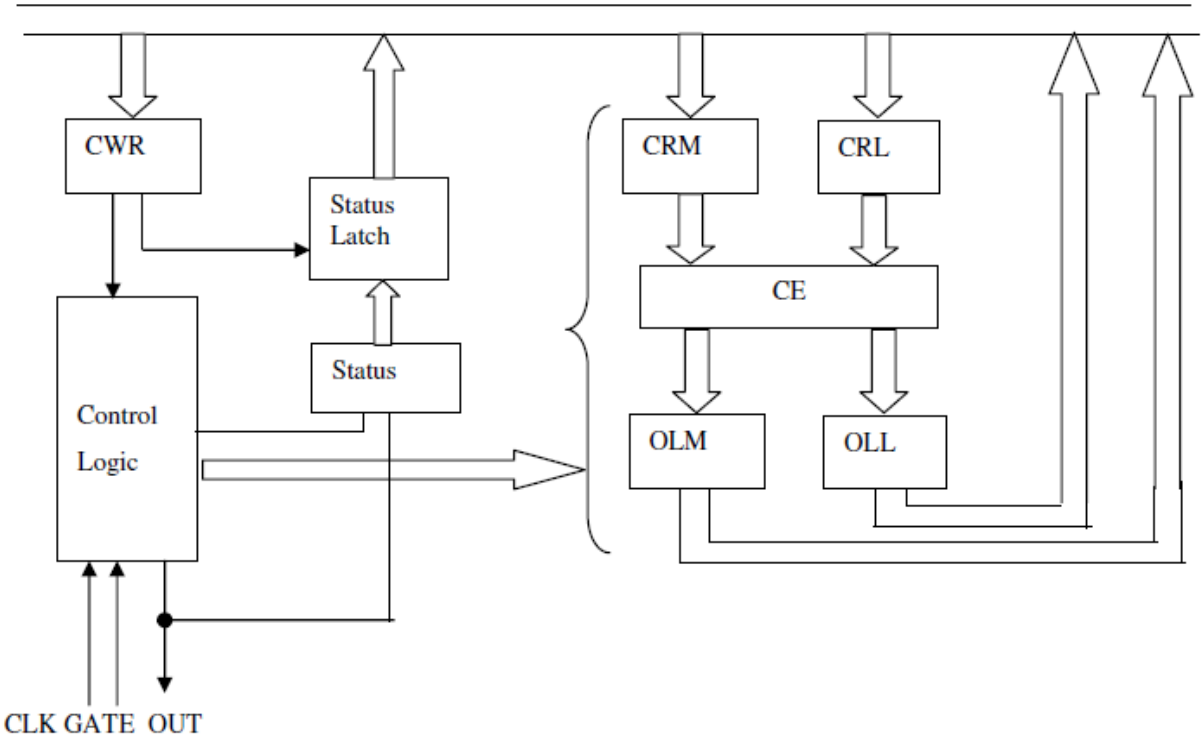


Fig. 4.2.1-2 – Schema bloc interna a unui numarator (conform catalog Intel)

Formatul cuvântului de control este:

SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Unde:

BCD = 0 numarare binara pe 16 biți (0-65535 evenimente)

= 1 numarare BCD (0 – 9999 evenimente)

M2 M1 M0 - selecție mod de lucru:

0 0 0 Mod 0 - oprire la sfârsitul numararii

0 0 1 Mod 1 - monostabil programabil

x 1 0 Mod 2 - generator de impulsuri divizate cu N

x 1 1 Mod 3 - generator de impulsuri dreptunghiulare

1 0 0 Mod 4 - strob comandat software

1 0 1 Mod 5 - strob comandat hardware

RW1 RW0 - comanda citire / scriere

0 0 buffer-are numaratoare pentru citire

0 1 R/W numai octetul LSB

1 0 R/W numai octetul MSB

1 1 R/W întâi octetul LSB si apoi imediat octetul MSB

SC1 SC0 – selecție numarator

0 0 selecție numarator 0

0 1 selecție numarator 1

1 0 selecție numarator 2

1 1 comanda de citire înapoi (readback command)

Programarea si apoi operarea se face independent pentru fiecare numarator.

Valorile oricarui numarator pot fi citite „din mers” (caz în care pot apare erori la tranzițiile de stare ale numaratorului) sau pot fi buffer-ate si apoi citite.

Operații de scriere: - se menționeaza urmatoarele convenții:

1. Pentru oricare numarator cuvântul de control trebuie scris înaintea scrierii valorii inițiale a numaratorului.

2. Valoarea inițiala a numaratorului se înscrie în conformitate cu combinația RW1 RW0.

În cazul în care RW1 RW0 = 11, în program nu trebuie sa se mai faca alta operație cu I8254 între cele doua
operații de scriere succesive.

Daca exista acest pericol (datorita apariției întreruperilor de exemplu), atunci operațiile de scriere se vor face
încadrate de cli si sti.
Operațiile de citire:
1. Numaratoarele pot fi citite fara a afecta numararea; citirea directa poate fi eronata daca CE este în tranziție de stare. De aceea, este de preferat ca înainte de citire sa se scrie un cuvânt de control corespunzator numaratorului care urmeaza a fi citit în care RW1 RW0=00. Dupa aceasta comanda, valoarea buffer-ata poate fi citita în orice moment; daca se reprogrameaza numaratorul, valoarea memorata este pierduta.
2. Citirea înapoi – permite sa se citeasca valoarea simultana a tuturor numaratoarelor, modul de lucru programat, starea curenta a iesirii OUTi etc.
Alte detalii se gasesc în filele de catalog ale I8254.

4.3.2 Utilizarea I8254 pentru numararea de evenimente

Oricare din cele 3 numaratoare pe 16 biți opereaza prin numararea impulsurilor de la intrare. Numaratoarele
numara prin decrementare pornind de la o valoare inițiala încarcata prin software.
Utilizând *modul de lucru 0* în care valoarea inițiala din numarator este 0, oricare numarator poate numara
65536 evenimente.
Intrarea CLK a numaratorului se conecteaza la un generator de tact extern.
Intrarea GATE a numaratorului este utilizata pentru a autoriza sau stopa numararea. Daca nu se doreste utilizarea acestei facilitați, *atunci intrarea GATE se pune la +5V*, caz în care se face permanent numararea.
Pentru modul de lucru 0, iesirea OUTi nu este utilizata.
Dupa ce un numarator a fost inițializat prin software, orice impuls de pe intrarea de tact decrementeaza numaratorul cu 1 pe front descrescator. Primul impuls totusi nu decrementeaza numaratorul, el fiind utilizat
pentru a încarca valoarea din CR in CE.
Exemplu: daca în numarator se încarca valoarea inițiala 0, numarul evenimentelor numarate este determinat
scazând valoarea curenta pe 16 biți, citita din numarator, din 65536 si apoi adunând 1 la rezultat pentru a
considera impulsul de încarcare a valorii inițiale.
Urmatorul subprogram exemplifica modul de utilizare a numaratorului 0 pentru numararea de evenimente. În
acest exemplu, intrarea GATE se presupune a fi la 5V, iar CA se considera a fi adresa de baza a dispozitivului.

```
COUNT DW ?
.
.
MOV DX,CA+3
MOV AL,30H;pentru NUM 0 citeste/scrie LSB urmat de MSB, mod lucru 0
OUT DX,AL
NOP ;introduce o “intarziere”
MOV DX,CA
XOR AL,AL
OUT DX,AL ; scrie 0 in LSB (valoare initiala)
NOP
OUT DX,AL ; scrie 0 in MSB
.
.
; in orice moment se poate citi numarul curent de evenimente astfel:
MOV DX,CA+3
XOR AL,AL
OUT DX,AL ; buffer-area numaratorului pentru a putea fi citit
```

```
NOP
MOV DX,CA
IN AL,DX
MOV BYTE PTR COUNT,AL ; memoreaza LSB
IN AL,DX
MOV BYTE PTR COUNT +1, AL ;memoreaza MSB
```

4.3.3 Utilizarea lui I8254 pentru generarea de întreruperi

Modurile de lucru 2 si 3 ale lui I8254A pot fi utilizate pentru generarea bazei de timp în cadrul sistemelor de achiziție de date si control. În astfel de aplicații, întreruperea de la I8254 va aparea periodic, în funcție de intrarea CLK si de programarea numaratorului, iar în cadrul rutinei de tratare a întreruperii se va face achiziția efectiva a datelor utilizând tehnicile cunoscute (de obicei, tehnica intrari buffer-ate).

Astfel, se cunoaste cu o buna precizie intervalul de timp dintre doua esantioane succesive din buffer-ul de achiziție.

În *modul de lucru 2*, oricare numarator opereaza ca generator de impulsuri divizate cu N. Când numaratorul trece prin 0, iesirea OUT trece în starea 0, pentru o perioada a impulsului de pe intrarea CLK si apoi revine în starea 1. Numaratorul este reîncarcat automat cu valoarea programata inițial si continua decrementarea (Fig. 4.3.3-1).

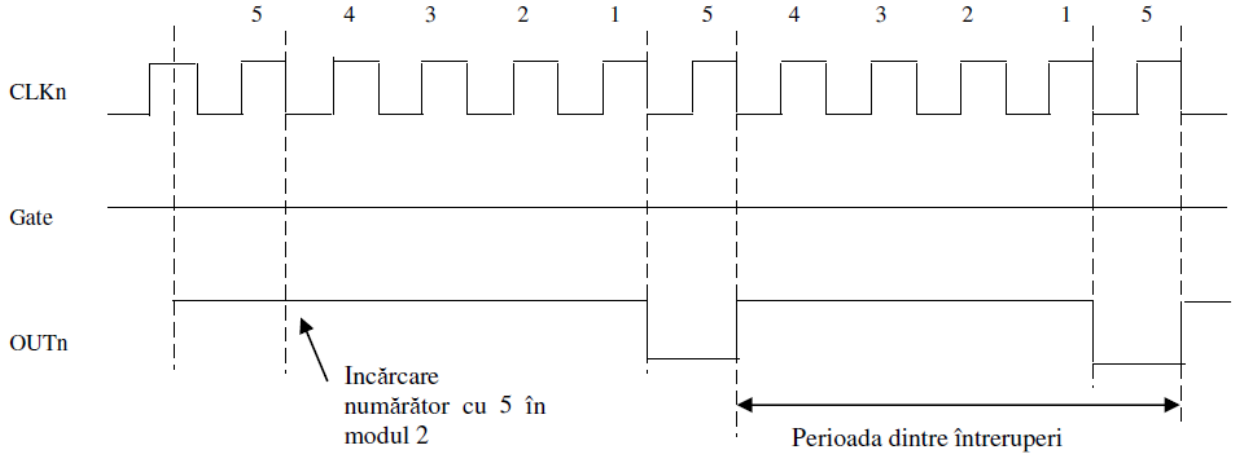


Fig. 4.3.3-1 – Generarea de intreruperi in modul 2 de lucru

Operația se reia periodic, generând întreruperi la intervale de timp egale cu perioada întreruperii. Se pot utiliza 2-3 numaratoare cascadate, în funcție de frecvența minima ceruta pentru întrerupere si de intrarea CLK a primului numarator (intrarea CLKi a numaratorului urmator este legata la iesirea OUTi-1 a numaratorului anterior).

Pentru exemplificare, se presupune intrarea CLK0 la 5MHZ (perioada pentru un impuls este de 200 ns). În acest caz, daca inițial numaratorul se încarca cu 0, deci se numara 65536 evenimente, atunci perioada impulsurilor de iesire va fi 65536 x 200ns = 13,1072 ms, sau 76 Hz. Cascadând numaratoarele, perioada impulsurilor pentru ultimul numarator din cascada se determina înmulțind valorile din numaratoarele cascadate si înmulțind apoi numarul obținut cu perioada impulsului de la intrarea primului numarator. Pentru 2 numaratoare cascadate, perioada maxima este 65536 x 65536 x 200 ns = 859,99 secunde iar pentru 3 numaratoare cascadate rezulta perioada maxima 65536 x 65536 x 65536 x 200 ns = 1,7851 ani.

Evident, nu este nevoie ca în toate numaratoarele sa se depuna valoarea maxima.

Se poate utiliza GATE pentru a controla din exterior generarea întreruperilor. Daca nu se doreste acest lucru,
GATE se pune la 5V.
În continuare se prezinta exemple de programarea I8254 pentru generarea de întreruperi pentru achiziții de date
utilizând modulul de achiziție data existent în laborator (vezi si 4.4).
*Exemplul 1: Generare de întreruperi pentru perioade mai mici decât 13,1072 ms (76 Hz £ frecvența £ 2,5
MHz).*
Se poate utiliza oricare dintre cele 3 numaratoare în modul de lucru 2 sau 3. Se poate utiliza poarta GATE
pentru a inhiba generarea de întreruperi din exterior. Ca suport hardware pentru acest exemplu se pot utiliza
modulele de achiziție ADA 2100 sau ADA 3100 de la laborator, modul în care se fac conexiunile fiind prezentat
în cartea tehnica a fiecarui modul.
Ca intrare CLKi (i=0,2) se utilizeaza semnalul XTAL de 5 MHz sau se poate utiliza o intrare de tact externa.
Daca GATE nu se utilizeaza pentru triggerare, atunci se pune la +5 V. Iesirea OUT se poate conecta prin jumperi
la un nivel de întrerupere (IRQ 2 ÷ IRQ 7) al calculatorului.
Programarea numaratorului si startarea generarii de întreruperi se face astfel:
1. Se calculeaza numarul ce trebuie încarcat în numarator
2. Se încarca cuvântul de control pentru numaratorul selectat (modul 2 de lucru)
3. Se încarca numarul în numarator
4. Numararea evenimentelor se starteaza automat la primul impuls pe intrarea clock de dupa încarcarea numarului în numarator. Oprirea se poate face trecând intrarea GATE la 0 sau reprogramând
numaratorul sau oprind calculatorul.
Urmatoarea secvența de program programeaza numaratorul 0 sa genereze întreruperi la 51 μs, considerând
intrarea de clock la 5 MHz, adresa de baza CA iar GATE este pusa la +5 V.
Numarul care se încarca în contor este 51/0.2 = 255.

```
MOV DX,CA+3
MOV AL,14H ; numarator 0, MOD 2, R/W numai LSB
OUT DX,AL
MOV AL,0FFH
MOV DX,CA
OUT DX,AL ; încarca nr. 255 în numaratorul 0
```

Frecvența întreruperii este 1/51μs = 19,5 KHz.
Exemplul 2: Generarea de întreruperi cu perioade mai mari ca 13,1072 ms (frecvența< 76 Hz).
Se cascadeaza 2 sau 3 numaratoare, toate programate în modul 2. Iesirea OUTi-1 se leaga la intrarea CLKi, i=1,2.
Se folosesc cele 3 numaratoare cascadate pentru ADA2100, unde:
CLK0 ® 5 MHz
OUT0 ® CLK1
OUT1 ® CLK2
OUT2 ® IRQn (n Î {2, 3, …, 7})
Programarea numaratoarelor si startarea generarii de întreruperi se face astfel:
1. Se calculeaza valoarea ce se depune în fiecare numarator.
2. Se programeaza numaratoarele în modul 2.
3. Se încarca succesiv numaratoarele începând cu ultimul din cascada si terminând cu primul din cascada.
În aceste condiții startarea se va face dupa încarcarea primului numarator, ce are intrarea CLK activa. Intrarile
GATE ale fiecarui numarator sunt puse la +5 V.

În exemplul de mai jos, se doreste generarea de întreruperi la 1s, utilizând cele 3 numaratoare ale I8254, intrarea
CLK0 a primului numarator este la 5Mhz
Mod de calcul :
1s = 109 ns;
o perioada a intrarii CLK este = 200 ns=0.200μs
Numarul de impulsuri CLK necesare este 109 ns/ 200ns = 5 x 106 = 5.000.000.
Se distribuie de exemplu acest numar pe cele 3 numaratoare astfel:
1000 în NUM2
1000 în NUM1
5 în NUM0
Programul este urmatorul:

```
MOV DX, CA+3
MOV AL, 34H ; NUM0, MOD2, R/W LSB apoi MSB
OUT DX, AL
NOP
MOV AL, 74H ; NUM1, MOD2, R/W LSB apoi MSB
OUT DX, AL
NOP
MOV AL, 0B4H ; NUM2, MOD2, R/W apoi MSB
OUT DX, AL
DEC DX ; adresa NUM2
MOV AL, 0E8h ;1000
OUT DX, AL
MOV AL, 03H
OUT DX, AL
DEC DX ; adresa NUM1
MOV AL, 0E8H ;1000
OUT DX, AL
MOV AL, 03H
OUT DX, AL
DEC DX ;adresa NUM0
MOV AL, 5H ;5
OUT DX, AL
XOR AL, AL
OUT DX, AL
.
.
.
.
```

4.3.4 Exemplu de utilizare a dispozitivului I8254 pentru generarea bazei de timp

Se cere implementarea controller-ului simplu PID prezentat la 2.2.2, considerând rata de esantionare pentru
calcule de 0.01 secunde (10 ms).
Pentru generarea bazei de timp se va utiliza un I8254 cu spațiul de adresare între 214h – 217h, intrarea GATE0
este la +5 V, intrarea de ceas de la NUM0 este conectata la un generator de impulsuri cu frecvența 5MHz iar
iesirea OUT0 este conectata la intrarea IRQ5 de la primul controller I8259A al PC. Modul de lucru pentru
numaratorul 0 va fi 2 iar valoarea inițiala care se va încarca în numarator este 10 000 000ns /200ns = 50 000 =
c350h.
În programul prezentat în continuare, comunicația între rutina de tratare si programul utilizator se face prin

variabila partajata contor:
• rutina de tratare a întreruperii incrementeaza variabila;
• programul utilizator, asteapta ca variabila sa fie diferita de 0:
ț daca este 1 efectueaza calculele (a trecut perioada T) si îi atribuie valoarea 0;
ț daca este mai mare decât 1 afiseaza eroare si termina programul deoarece programul principal nu se executa suficient de rapid pentru a se sincroniza cu dispozitivul extern.

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#define INTRț0d 0x0d/*Întreruperea IRQ5 de la I8254*/
#define CA 0x214
# define KPVAL 1.0
# define KIVAL 0.8
# define KDVAL 0.3
#ifdef ţţcplusplus
#define ţţCPPARGS ...
#else
#define ţţCPPARGS
#endif
unsigned char oldmaskț1=0;
int count=0,oldcount=0;
float s, kp, ki, kd, en, enold, mn;
/*functia can achizitioneaza de la convertorul analog numeric, calculeaza
si returneaza valoarea erorii e; este dependenta de hardware utilizat – de
exemplu poate fi scrisa pentru ADA2100*/
extern float can(void);
/*functia cna care primeste la intrare corectia (valoarea de actionare) si
o transmite la convertorul numeric – analogic; este dependenta de hardware
utilizat – de exemplu poate fi scrisa pentru ADA2100 */
extern void cna(float mn);
void interrupt ( *oldint0d)(ţţCPPARGS);
void InitNum0 (void){
outportb (CA+3, 0x34);
outportb (CA,0x50);
outportb (CA,0xc3);
}
/* rutina de tratare a întreruperii de la contorul 0 al I8254*/
void interrupt irq5(ţţCPPARGS){
/* incrementeaza contorul */
count++;
/* achitarea întreruperii */
outportb(0x20,0x20);
}
void main(void){
s = 0.0; kp = KPVAL; ki = KIVAL; kd = KDVAL;
/* salveaza vechiul vector de întrerupere */
oldint0d = getvect(INTRț0d);
/* dezactiveaza sistemul de întreruperi mascabile*/
disable();
/* seteaza noul vector de întrerupere */
setvect(INTRț0d, irq5);
/*salveaza masca pentru primul 8259*/
oldmaskț1=inportb(0x21);
/*demascheaza întreruperea 0*/
outportb(0x21,oldmaskț1 & 0x0df);
```

```
/* activeaza sistemul de întreruperi mascabile */
enable();
clrscr();
enold = can();
InitNum0();
/* bucla de prelucrare a programului principal */
while (!kbhit()){/*se opreste la apasarea oricareai taste*/
if(count){
disable();
oldcount = count;
count = 0;
enable();
if (oldcount == 1){
en = can();
s = s+en;
mn = kp*en + ki*s + kd * (en - enold);
cna(mn);
enold = en;
}
else{
printf ("Eroare, lipsa sincronizare");
break;
}
}
}
/* dezactiveaza sistemul de întreruperi mascabile
*/
disable();
/* seteaza vechii vectori de întrerupere */
setvect(INTRț0d, oldint0d);
/*seteaza vechea masca pentru primul 8259*/
outportb(0x21,oldmaskț1);
/* activeaza sistemul de întreruperi mascabile*/
enable();
}
```

Probleme propuse:

1. I8254 are spațiul de adresare între 214h si 217h. Intrarea GATE0 este la +5 V, iar intrarea de CLK0 este legata la un circuit formator de impulsuri de la un comutator acționat manual. Sa se elaboreze un program care sa afiseze pe display un mesaj la fiecare zece acționari ale comutatorului si care sa se opreasca dupa 200 acționari ale acestuia. Programul se va scrie astfel încât numararea impulsurilor sa se faca de catre o rutina de tratare a întreruperilor de la ceasul sistem, iar afisarea mesajului sa se faca în programul principal.
2. I8254 are spațiul de adresare între 214h – 217h. Intrarile GATE pentru numaratoare sunt la +5 V iar intrarea de ceas de la NUM0 este legata la un generator de impulsuri cu frecvența 5MHz. Presupunem ca orice iesire OUTi poate fi conectata la oricare dintre canalele de întreruperi ale circuitului I8259 si de asemenea la intrarea CLK a urmatorului numarator. Într-o aplicație de achiziție de date este necesar ca o intrare analogica sa fie esantionata la 10ms, a doua intrare analogica la aproximativ 200 ms iar a treia la 1000s. Sa se elaboreze un program care programeaza I8254 astfel încât sa genereze întreruperi pe nivelurile 10, 11 si 15 (al doilea circuit I8259 de la un calculator compatibil IBM PC/AT), fiecarui numarator corespunzându-i una din întreruperile de mai sus. Simularea achiziției de date la intervalele specificate se va face afisând caracterul * la primul interval de timp, $ la al doilea interval de timp si @ la al treilea interval de timp. Programul poate fi oprit prin CTRL-BREAK. Sa se deseneze schema de cascadare a numaratoarelor si de conectare la canalele de întrerupere.
3. Elaborați o schema bloc de masura si un program pentru determinarea si afisarea vitezei de rotație instantanee a unui motor care are max. 3000 rot/min, utilizând I8254. Pe axul motorului exista un disc

cu 10 fante echidistante si un dispozitiv optic care genereaza un impuls catre calculator la trecerea unei fante prin dreptul sau.

4.3.5 Reprogramarea ceasului sistem

Placa de baza a calculatorului PC conține un dispozitiv I8254A, care ocupa spațiul de adrese de I/E de la 40h la
43h. Intrarea CLK0 este la 1,193187 MHz iar iesirea OUT0 este conectata la intrarea IRQ0 a primului controller
I8259A. BIOS programeaza NUM0 în modul de lucru 3 si cu valoarea 0, generându-se astfel întreruperi cu o
perioada de timp de aprox. 54.92517 ms (aprox. 18.206588 Hz). Aceasta întrerupere este utilizata de SO DOS
pentru ceasul sistem si alte operații (precum oprirea motorului discului flexibil atunci când acesta nu mai este
utilizat). În tabela vectorilor de întrerupere, adresa rutinei de tratare a IRQ0 se gaseste la adresa 20H (corespunde
la int 8).

Daca aplicația necesita întreruperi cu frecvența diferita de aprox. 18,2 Hz si daca în calculator nu exista un
dispozitiv suplimentar I8254, se poate utiliza numaratorul 0 al circuitului I8254 de pe placa de baza. Urmatoarea
secvența va reprograma ceasul pentru a da întreruperi la o noua perioada de timp:

```
MOV AL, 36H ; MOD3, R/W MSB apoi LSB binar pentru NUM0
OUT 43H,AL
MOV AX, timerțcount ; in AX noua valoare a NUM
OUT 40H, AL
MOV AL,AH
OUT 40H, AL
```

La sfârsitul programului utilizator este necesar sa se restaureze ceasul sistem la rata de întrerupere inițiala astfel:

```
MOV AL, 36H
OUT 43H, AL
XOR AL, AL
OUT 40H, AL
OUT 40H, AL
```

Valoarea care se scrie în NUM0 se calculeaza astfel:

Timerțcount = Ti * F0

Unde F0 este frecvența în Hz a semnalului de tact pentru numarator (1.193.187 Hz) iar Ti este perioada
întreruperilor (intervalul de timp dintre doua întreruperi succesive).

Exemplu: pentru generarea de întreruperi la 10 ms rezulta timerțcount = 10/1000 * 1.193.187 » 11932.

Dupa cum se stie in numarator se pot depune valori între 0 si 65535, 0 corespunzând la 65536 de impulsuri.

Daca se doreste o valoare mai mare, atunci se împarte valoarea respectiva printr-un numar astfel încât valoarea
care se introduce în numarator sa fie între 0 si 65535. Notam acest numar NDIV. Rutina de tratare a
întreruperilor va gestiona un numarator soft într-o locație de memorie, ce este încarcata inițial cu NDIV si care
se decrementeaza prin program cu 1 la fiecare întrerupere. Când ajunge la 0, rutina de tratare a întreruperii îl
reîncarca cu NDIV si apoi apeleaza rutina de achiziție.

Exemplu: daca se doreste achiziție la fiecare secunda, atunci timerțcount = 1 * 1.193.187 » 20 x 59659 rezulta
ca NDIV = 20 si timerțcount = 59659.

Daca se doreste menținerea datei si orei sistem, care în mod curent sunt gestionate de rutina de tratare BIOS

pentru int 8, trebuie sa se apeleze din rutina de tratare utilizator, atunci când este cazul (la perioade de timp
de aprox. 55ms), rutina BIOS de pe nivelul 8 din tabela vectorilor de întreruperi.
Urmatoarea rutina poate fi considerata ca o schița pentru o rutina de tratare a întreruperii prin reprogramarea
ceasului sistem.

```
PCțINT8 Label DWORD
PCțOFF8 DW ? ; IP
PCțSEG8 DW ? ; CS
DSEG DW ? ; rezervat pentru memorarea reg. DS
MYțINT8:
PUSH DS
PUSH AX
; …… eventual alte salvari
MOV AX, CS:DSEG ;incarca ds cu valoarea corecta
MOV DS,AX
;Urmatoarea secventa este ceruta numai daca se utilizeaza numaratoare mai
;mari decât 65536:
DEC NDIVțCT;decrementare contor soft
JNZ NOTțZERO
MOV AX,NDIV
MOV NDIVțCT,AX
CALL achizitie ;apelare rutina ce face efectiv achizitia
:
NOT-ZERO:
;….. eventual alte restaurari
POP AX
POP DS
```

Se poate insera în acest loc instrucțiunea
JMP CS: PC țINT8
pentru apelul rutinei originale de tratare a întreruperilor de ceas la 18,2 Hz; daca nu executa saltul prin JMP la
rutina de ceas, în acest loc trebuie sa existe instrucțiunea iret.
Problema propusa
Scrieți funcția *delayțms(unsigned int t)* care realizeaza o întârziere de aproximativ *t* milisecunde, indiferent de
procesorul PC-ului, fara a reprograma ceasul sistem. Indicație: se va utiliza numaratorul 0, care este preprogramat de catre BIOS, citind periodic valoarea din contor, facând diferențele între citiri succesive si
calculând numarul de milisecunde.

4.4 Exemplu de sistem de achiziție de date

Nota :
Paragraful 4.4 conține detalii tehnice necesare pentru instalarea, calibrarea si programarea achiziției de
date cu modulul ADA2100.
Pentru rezolvarea problemelor la examen, se va studia 4.4.11.
Ca alternativa pentru rezolvarea problemelor la examen, informațiile din curs pot fi completate cu
documentațiile tehnice ale modulelor PCM 3718, PCM 3780 si PCM 3712 precum si ale dispozitivelor
I8254 si I8255 studiate la laborator:

• "PCM-3718 Series, PC/104 12-bit DAS Module with Programmable Gain - User Manual", prezentat in
documentul PCM-3718-Ed3.pdf , .

• "PCM-3780, 2-ch Counter/Timer with 24-ch TTL DIO Module - User Manual" prezentat în documentul
PCM-3780țManualțEd1.1.pdf
• Subsistemul iesirilor analogice pentru sistemul PC104 de la laborator, prezentat în documentul PCM-3712-ed1.pdf
• 8255țdatasheet.pdf - capitolele "Functional description", "Operational description" si "Operating modes".
• I8254țdataţsheet.pdf - capitolele "Functional description", "Operational description".

4.4.1 ADA 2100 - date generale si prezentarea resurselor

Interfața:
• compatibil IBM PC /XT/AT
• adresa de baza selectabila prin jumper, mapata I/E
• întreruperi selectabile prin jumperi

Intrari analogice:
• 4 intrari diferențiale sau 8 simple, selectabile prin switch
• impedanța de intrare: >10 Mohmi
• câștig, selectiv soft: 1,2,4,8,16
• eroare câștig.: 0,5% tipic, max. 1%
• opțiuni intrare:
ț domeniu 10V: bipolar+/-5V; liniaritate garantata -5V...+5V
ț domeniu 10V: unipolar 0-10V; liniaritate garantata 0...9,5V
ț domeniu 20V . . bipolar +/-10V; liniaritate garantata -9,5V...+9,5V
• Domeniul: selectabil cu jumper
• Polaritatea: selectabila cu switch
• timp de stabilizare: max 3 µs
• tensiune de intrare mod comun : +/-10V
• protecție la supratensiuni +/-35Vdc

Convertor A/N
• tip: aprox. succesive
• rezoluție: 12 biți (domeniul 10V -2.44mV/bit, domeniul 20 V - 4.88 mV/bit)
• viteza de conversie . tipic 20 usec
• liniaritate : +/-1 bit
• timp total de raspuns S/H: max 6 usec
• rata maxima de esantionare: 38 KHz
•

Convertor N/A
• iesiri analogice: 2
• rezoluție: 12 biți
• precizie relativa: +/-1 bit max
• precizie capat de scala: +/-3 bit max
• neliniaritate: +/-1 bit max
• izolare canal-la-canal: tipic 84 dB

*Iesirea analogic*a
• gama de iesire, chip selectabila:
ț opțiunea 1: 0 la +5V selectabila cu jumper +/-5V
ț opțiunea 2: 0 la 10 V selectabila cu jumper la +/-10V
• timp de stabilizare la 0,01%FSR: 1.8ms tipic; 3.3 ms max
• eroare de zero: +/-1/2 bit
• crosstalk.....90 dB tipic

Alimentari: +/- 12V,+5V de la PC, masa de la PC

Contor/Periodizator: -3 numaratoare de 8 MHz, pe 16 biți fiecare

Linii I/E numerice:16 compatibile TTL/CMOS, porturile A si C de la I8255

Cerințe de alimentare
• +5V: 240 mA
• +12V: 30 mA

• -12V: 35 mA
Conectori I/E conector frontal de 40 pini
Condiții de mediu
• temp. de lucru : 0 ... +70 grd C
• temp. înmagazinare : -40 ... +85 grd C
• umiditate : 0 ... 90 %
Dimensiuni:
• înalțime : 99 mm
• lațime: 165 mm

4.4.2 Generalitați

ADA 2100 reprezinta un modul complex compatibil IBM PC-Bus, "Short-Size" ce poate fi introdus direct în
orice slot neutilizat dintr-un calculator compatibil PC/XT/AT.
Este compus din urmatoarele blocuri:
• convertor analog-numeric (de 12 biți) de înalta rezoluție;
• convertor numeric-analogic (12 biți) dublu;
• circuit interfața paralela (PPI - I8255);
• circuit numarator/periodizator (PIT - I8254);
toate asigurând flexibilitatea interfeței pentru multe aplicații.
Se caracterizeaza printr-o construcție în 6 planuri (straturi) incluzând planuri separate pentru masa si alimentare
ceea ce conduce la caracteristici de zgomot redus.
ADA 2100 asigura o conversie pe 12 biți analog-numerica multicanal atât pentru semnale diferențiale cât si SE
(single - ended). Aceasta facilitate confera calculatorului compatibil IBM PC/XT/AT „gazda" posibilitatea
achiziției de date si controlul în timp real putând genera si analiza semnale analogice numerice.
Viteza si rezoluția sunt caracteristicile principale luate în calcul la un convertor A/N. Pentru majoritatea
aplicațiilor o rezoluție de 12 biți este suficienta. Aceasta asigura incremenți de tensiune de 1.22 mV pentru o
gama de 5 V, 2.44 mV pentru o gama de 10 V si 4.88 mV pentru o gama de 20 V.
O conversie pe 8 biți este mai puțin precisa dar mai rapida. Prin programare (S) si/sau setare de switch-uri (H)
de pe placa se pot:
• selecta adresa I/E de baza (H);
• alege 4 canale de intrare analogice diferențiale sau 8 simple (SE) (H);
• selecta canalul activ (S);
• selecta câstigul canalului (S);
• selecta gama si polaritatea tensiunii analogice de intrare (H);
• selecta polaritatea tensiunilor analogice de iesire (H);
• controla 16 linii I/E TTL/CMOS (S);
• controla 3 circuite numarator/periodizator pe 16 biți de 8 MHz (S);
• monitoriza conversia A/N folosind semnalul EOC (End-Of-Convert) (S,H);
• genera semnale de întrerupere (H).

4.4.3 Setarea adresei de baza

Interfața fiind introdusa în canalul I/E adresa sa va fi configurata în spațiul 200H...3FFH.
Notam cu BA adresa de baza a modulului.
ADA 2100 utilizeaza 24 locații de adresa în spațiul I/E al calculatorului începând cu BA.

4.4.4 Harta I/E pentru ADA 2100

Funcție A4 A3 A2 A1 A0 R/W BA+Hex
1. PPI 8255
Port A 0 0 0 0 0 R/W 0
Port B (sel.canal+câstig) 0 0 0 0 1 W 1
Port C 0 0 0 1 0 R/W 2

Cuvânt de control 0 0 0 1 1 W 3
2. Circuit conversie A/N
Start conversie 12 biți 0 0 1 x 0 W 4 sau 6
Start conversie 8 biți 0 0 1 x 1 W 5 sau 7
Citire MSB 0 0 1 x 0 R 4 sau 6
Citire LSB 0 0 1 x 1 R 5 sau 7
3. Circuit conversie N/A
Program AOUT1 LSB 0 1 0 0 0 W 8
Program AOUT1 MSB 0 1 0 0 1 W 9
Program AOUT2 LSB 0 1 0 1 0 W A
Program AOUT2 MSB 0 1 0 1 1 W B
Conversie/actualizare
AOUT1/AOUT 0 1 1 x x W C, D, E sau F
Clear AOUT1/AOUT2 1 0 0 x x W 10,11,12 sau 13
4.PIT 8254
NUM 0 1 0 1 0 0 R/W 14
NUM 1 1 0 1 0 1 R/W 15
NUM 2 1 0 1 1 0 R/W 16
Cuvânt de control 1 0 1 1 1 W 17
---
cu x s-a notat un bit a carui valoare nu conteaza pentru respectiva combinație.
Este important de subliniat ca unele din locațiile de adresa din spațiul I/E al calculatorului sunt deja ocupate de
porturile interne si alte periferice. Daca modulul ADA 2100 încearca sa utilizeze locațiile de adresa I/E deja
folosite de un alt dispozitiv din sistem va rezulta un conflict de adresare. De aceea modulul nu va mai fi
operațional sau va opera defectuos.
Pentru a evita aceasta situație se foloseste un circuit de selecție a adresei de baza cu „jumper". Prin schimbarea
poziției acestuia la conectorul notat P2, se poate selecta adresa de baza I/E (BA) la oricare din cele 8 locații de
mai jos:
200 240 280 2C0 300 340 380 3C0
Modulele livrate de firma sunt selectate de la adresa de baza 200H.

4.4.5 Modul de configurare al modulului

În tabelul de mai jos se prezinta funcțiile configurabile ale modulului si modul cum sunt setate de firma:
Funcție | Setare de firma

1.Adresa de baza I/E | 200 hex
2.Tip canal de intrare analogic | 4 canale diferențiale
3.Selecția canalului analogic
de intrare | Controlabila prin software
4.Selecția câstigului intrarii
analogice | Controlabila prin software
5.Domeniul si polaritatea tensi | Este specificata de utilizator
unii analogice de intrare | la comandarea interfeței
6.Monitorizarea semnalului EOC | Conectat la bit A7 PPI
7. Polaritatea tensiunilor ana | Domeniu unipolar pozitiv
logice de iesire
8.16 linii I/E de la PPI si circu
it numarator/periodizator pro
gramabil (PIT) | Controlabile prin software
9.Moduri de lucru | Controlabile prin software

10.Configurație I/E PIT | Intrare ceas: 5 MHz
| Intrare GATE: +5 V
11.Întreruperi | Dezactivate

4.4.6 Descriere la nivel de schema bloc

Modulul ADA 2100 are 4 blocuri mari funcționale:
• lanțul de conversie analog-numerica (A/N)
• lanțul de conversie numeric-analogica (N/A)
• circuitul interfața paralela programabila (PPI)
• circuitul numarator/periodizator programabil (PIT)

4.4.6.1 Lanțul de conversie analog-numerica

Principala funcție a modulului ADA 2100 este de a asigura facilitați de conversie analog-numerica de mare
viteza pentru achiziția de date.
Circuitele de conversie analog-numerice primesc intrari de la 4 canale analogice diferențiale sau 8 canale
simple, selecteaza un canal activ si realizeaza conversia analog-numerica a valorii tensiunii citite pe acel canal.
Viteza de conversie este tipic de 38 KHz.
Multiplexoarele
Este folosit un multiplexor analogic de 8 biți pentru a conecta fie unul din cele 8 canale SE fie unul din cele 4
diferențiale la circuitul de câstig. Cele 3 comutatoare S1 cele mai din stânga modulului programeaza
multiplexorul sa primeasca intrari diferențiale sau SE asa cum a fost descris anterior. Selecția unuia din canale
se face prin program scriind numarul de canal (0,1,2 3…) în cei 4 biți LSB ai portului B al PPI (BA+1).
Când cele 3 switch-uri sunt sus, multiplexorul este programat SE, iar când sunt jos, este configurat diferențial.
Circuitele de control al câstigului (amplificarea)
Circuitele de control al câstigului programabil pot oferi un factor de câstig de 1, 2, 4, 8 sau 16. Selecția
câstigului se face prin scrierea în registrul B al PPI (adresa BA + 1), tetrada MSB (tetrada are valorile 0 câstig
1, 1 câstig 2, 2 câstig 4, 4 câstig 8, 8 câstig 16). Factorul de câstig este controlat prin setarea a 4 comutatoare
analogice. Pentru un câstig de 2, 4, 8 sau 16 aceasta operație de scriere va închide unul din cele 4 comutatoare;
în cazul câstigului unitar toate comutatoarele sunt deschise. Nu se recomanda programarea altor factori de
câstig decât cei 5 aratați mai sus.
Circuitele de esantionare-memorare (S/H)
Un circuit de esantionare si memorare (S/H - sample and hold) este folosit între iesirea circuitelor de control al
câstigului si intrarea A/N pentru a asigura ca semnalele analogice dinamice sa fie digitizate precis de catre
convertorul A/N. Se foloseste un condensator de 0.001 mF ce minimizeaza timpul de achiziție (6 microsec.
tipic). Timpul si viteza de esantionare si memorare sunt determinate de semnalul EOC generat de convertorul
A/N. Când EOC este 1, logic amplificatorul esantioneaza intrarea analogica; când EOC este 0 amplificatorul
memoreaza intrarea.
Convertorul A/N (HI 574)
Convertorul A/N este un circuit integrat ce realizeaza o conversie de mare viteza pe 12 biți caruia îi corespunde

o conversie în aproximativ 20 microsec. Se pot efectua de asemenea conversii pe 8 biți în aplicațiile în care
viteza este mai critica decât rezoluția. Conversiile pe 8 biți dureaza aproximativ 13 microsec., permițând
conversii rapide a intrarilor analogice dinamice. Convertorul suporta semnale de intrare analogice de 10 sau 20
V; însa nu poate suporta domeniu de intrare unipolar de 20 V deoarece tensiunea de alimentare în aplicațiile cu
ADA 2100 este de numai +12 V.
O conversie pe 8 sau 12 biți este inițiata printr-o operație de scriere la adresa I/E corespunzatoare. O data ce o
conversie a început, starea acesteia poate fi urmarita prin citirea semnalului de stare al convertorului A/N (STS)
care este o iesire a circuitului convertor A/N si inversat înainte de a fi folosit de alte circuite de pe interfața ca
semnal EOC (**E**nd-**O**f-**C**onvert). Semnalul EOC este 0 logic în timpul unei conversii.

4.4.6.2 Realizarea unei citiri A/N

Dupa ce s-a selectat o intrare analogica si s-a setat câstigul, se poate efectua o citire A/N. De notat ca odata
câstigul si canalul setate, ele ramân la aceste valori pâna ce utilizatorul le modifica. Deci nu trebuie setate la
fiecare citire A/N.
De fiecare data când se termina o conversie A/N, se genereaza un semnal EOC.
Se poate monitoriza starea conversiei A/N cu acest semnal prin configurarea bit 7 al port A sau port C ca linii de
intrare si conectarea semnalului EOC pe aceasta linie. Pentru o conversie pe 12 biți, data A/N citita este aliniata
la stânga într-un cuvânt pe 16 biți cu cei mai puțin semnificativi 4 biți egali cu 0 asa cum se observa mai jos:
MSB
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
DB12 DB11 DB10 DB9 DB8 DB7 DB6 DB5 DB4 DB3 DB2 DB1 0 0 0 0

4.4.6.3 Circuitele de conversie numeric-analogica

Doua iesiri analogice obținute de catre un convertor N/A pe 12 biți sunt disponibile în conectorul de I/E .
Tensiunea de referința (de 10 V) se obține cu ajutorul circuitului integrat REF01.
Conversia numeric-analogica (N/A) de 12 biți este realizata cu ajutorul convertorului dublu de tip AD 7537.
Tensiunile de iesire sunt disponibile prin intermediul unor amplificatoare operaționale de tip AD 712, care
realizeaza scalarea semnalelor.

4.4.6.4 Interfața paralela **programabil**a **(PPI - I8255)**

PPI ofera 16 linii numerice I/E TTL/CMOS care pot fi configurate în funcție de necesitați. Liniile disponibile
pentru intrari/iesiri numerice sunt porturile A si C. PPI 8255 are în total 24 linii I/E numerice, dintre care 8 sunt
folosite pentru a controla selecția canalului A/N si circuitul de câstig (port B). Celelalte 16 linii sunt disponibile
pentru exterior la conectorul P15. Portul C este subdivizat în 2 porturi de 4 biți port C low (CL) (PC0-PC3) si
port C high (CH) (PC4-PC7). Portul B este întotdeauna în mod 0 iesire, fiind rezervat. Foile de catalog pentru
PPI 8255 prezinta detaliile necesare pentru programare.

4.4.6.5 PIT numa**rator/periodizator programabil (I8254)**

Acest circuit integrat conține 3 numaratoare/periodizatoare pe 16 biți independente TC0, TC1, TC2 care
realizeaza funcția de numarare înapoi. Aceste numaratoare pot lucra cu incremenți de timp pâna la 125 nsec.
Cea mai uzuala folosire este de a programa intervale de timp precise. Cele 3 circuite numarator/periodizator sunt
independente, ele putând fi cascadate pentru numarari inverse mai mari decât capacitatea de 16 biți. De exemplu
semnalul OUT al TC0 se poate conecta la CLK pentru TC1 si OUT de la TC1 la CLK de la TC2.
Oricare din iesirile OUT0 - OUT2 poate fi utilizata la întrerupere pentru PC.
Foile de catalog pentru 8254 prezinta detaliile necesare pentru programare.

4.4.7 Prezentarea comutatoarelor si a programatoarelor hardware

S1 - selecție tip semnal de intrare analogic

S1 reprezinta un set de 4 comutatoare (switch-uri) ce au rolul de a configura multiplexoarele pentru intrari SE
sau diferențiale si selecteaza o gama de tensiuni de intrare unipolare sau bipolare. Primele 3 comutatoare ale lui
S1 opereaza ca un grup. Când ele sunt în poziția superioara multiplexoarele sunt configurate pentru intrari
simple. Când toate sunt în poziție inferioara multiplexoarele sunt configurate pentru intrari diferențiale. Pentru
ca multiplexoarele sa funcționeze corect este obligatoriu ca toate comutatoarele 1...3 sa fie în aceeasi poziție.
Cel de-al patrulea controleaza polaritatea tensiunii de intrare. Când switch-ul e în poziție superioara gama
tensiunii de intrare este unipolara; poziției sale inferioare îi corespunde o gama bipolara.
Acest comutator, în conjuncție cu selecția gamei tensiunii, setata la conectorul P9, determina tensiunile de
intrare analogice suportate de convertorul A/N. De subliniat ca în cazul oricarei schimbari a polarității, trebuie
calibrat circuitul de conversie A/N.

*P2 - conector pentru fixarea adresei de baz*a

Modul de selectare a adresei de baza a fost prezentat anterior.

*P3 - conector I/E pentru circuitul num*ar*ator/periodizator (PIT)*

PIT conține 3 circuite numarator/periodizator independente de 16 biți fiind un modul de tip I8254. Fiecare din
aceste circuite are 3 semnale I/E asociate cu: ceas (CLK), poarta (GATE) si iesire (OUT). Fiecare dintre
numaratoare este setat de firma astfel:
- intrare ceas (CLKi) - XTAL (generator intern de 5 MHz)
- poarta (GTi) - la +5V
- iesire (OUTi) - COi

unde i = 0, 1, 2

Conectorul P3 este poziționat în 3 grupuri funcționale TC0, TC1 si TC2 ce corespund celor 3 timere. Pentru
TC0 corespund semnalele CK0, GT0 si OUT0 în dreapta P3 cele 3 corespunzând numaratorului 0 si în stânga
XTAL, ECO, +5 V, EG0, CO0 si CO0 negat;
unde: EC0 = **E**xternal **C**lock - ceas extern
EG0 = **E**xternal **G**ate
CO0, CO0 negat = **C**lock **O**utput - iesire numarator
Deci ceasul poate fi conectat la XTAL sau EC0 (sursa externa), validarea numaratorului (prin intrarea GATE) la
+5V sau EG0 (validare externa) oferindu-se si iesirea negata (CO0 negat).
Analog pentru TC1 si TC2 în cazul celorlalte timere.

Se pot cascada cele 3 timere prin legarea iesirii unuia la intrarea celuilalt. ECi sunt legate împreuna la pinul 33
al conectorului P8 de I/E, EGi sunt legate la pinul 35 al conectorului P8, iar una dintre cele 3 iesiri OUTi se
poate lega prin jumper la pinul 36 din conectorul P8.

P5, P7 - conectori pentru întreruperi

Conectorii P5, P7 sunt utilizați pentru a conecta diversele semnale generate de ADA2100 la canalele de
întreruperi ale calculatorului. Canalele de întrerupere disponibile pe modul sunt IRQ2...IRQ7. De notat ca la un
moment dat în calculator o singura întrerupere poate fi conectata la un canal de întrerupere.
Trebuie evitate conflictele la selectarea canalelor de întrerupere atât cu semnalele ADA 2100 cât si cu alte
dispozitive din calculator.
Sursele de întrerupere la ADA2100 sunt iesiri de tip TTL totem-pole si nu open-collector, deci nu se vor conecta
împreuna 2 astfel de surse de întrerupere.

P5 - întreruperile de la iesirile PIT

Conectorul P5 prezentat în figura de mai jos este folosit pentru a conecta una din iesirile OUT0, OUT1 sau
OUT2 la unul din canalele de întrerupere IRQ2...IRQ7 ale calculatorului. Selecția se face tot cu 2 jumperi (unul
selecteaza întreruperea, celalalt iesirea). În figura de mai jos se arata modul de plasare al jumperilor astfel încât
OUT2 sa fie conectat la IRQ3.

P5 |---- |
| | IRQ7
| | IRQ6
| | IRQ5
| | IRQ4
| - - | IRQ3
| | IRQ2
| | OUT0
| | OUT1
| - - | OUT2
|---- |

*P7 - întrerupere generat*a *de A/N End-Of-Convert (EOC)*

Conectorul P7 prezentat mai jos este folosit pentru a conecta semnalul Sfârsit de Conversie (EOC) al
convertorului A/N la una din liniile de întrerupere IRQ2...IRQ7. Semnalul EOC este conectat la canalul IRQ
prin instalarea unui singur jumper orizontal peste pinii canalului IRQ selectat. În exemplul de mai jos se arata
modul de conectare al semnalului EOC la IRQ4.

P7 |----|
| | IRQ7
| | IRQ6
| | IRQ5
| -- | IRQ4
| | IRQ3
| | IRQ2

conector pentru monitorizarea EOC

Asa cum s-a amintit mai sus, semnalul EOC poate fi utilizat pentru a genera o întrerupere. Daca acest semnal nu
este utilizat ca o întrerupere, el poate fi utilizat pentru urmarirea prin polling a starii procesului de conversie

A/N. Conectorul P6 ofera 2 linii prin care EOC poate fi monitorizat de la PPI, PA7 sau PC7. Una din aceste 2
linii I/E digitale este selectata pentru monitorizarea EOC prin instalarea unui jumper orizontal peste perechea de
pini corespunzatoare.
Linia numerica I/E selectata, PA7 sau PC7, trebuie sa fie configurata ca intrare în mod 0 la programarea PPI.
În figura de mai jos se prezinta P6 cu un jumper instalat în poziția setata de firma pentru monitorizarea EOC
prin PA7:
P6 |----|
| | PC7
| -- | PA7
|----|
P9 - Conector pentru selecția domeniului tensiunii convertorului A/N
Conectorul P9 ilustrat în figura de mai jos este folosit pentru a selecta tensiuni de intrare analogice a convertorului analog-numeric.
Un jumper este instalat vertical peste pinii marcați „10V" pentru a suporta un domeniu de 10 V (0 la 10 V, fie -5
la +5 V) sau peste pinii marcați 20 V pentru a suporta un domeniu de 20 V (-10 la +10 V). Fixarea acestui
jumper în conjuncție cu fixarea comutatorului DIP S1-4 care selecteaza un domeniu unipolar sau bipolar,
determina domeniul tensiunii de intrare al convertorului A/N. P9 este configurat de firma în concordanța cu
specificațiile clientului pentru domeniul tensiunii de intrare.
Combinațiile valide ale P9 si S1-4 sunt ilustrate mai jos:
Domeniu Setare P9 Setare S1-4
-5 V la +5 V 10 V (dreapta) jos (bipolar)
0 V la +10 V 10 V (dreapta) sus (unipolar)
-10 V la +10 V 20 V (stânga) jos (bipolar)
|-----|
P9 20 V| | | 10 V
|-----|
P10 - conector pentru fixarea polarității tensiunilor de iesire
Este utilizat pentru a selecta polaritatea tensiunilor de iesire pe cele doua iesiri analogice AOUT1 si AOUT2.
Fiecare canal poate fi iesire unipolara (+) sau bipolara (-/+). Selecția se face cu jumpere instalate vertical ca în
figura de mai jos.
AOUT1 AOUT2
----- -----
---------- P10
| |
| | | |
| |

+ + + +
/ /
- -
Asignarea pinilor la conectorul P8
Nr. pin Nume semnal Nr. pin Nume semnal SE/DIFF
1 AN1/AN1+ 2 GND
3 AN5/AN1- 4 GND
5 AN2/AN2+ 6 GND

7 AN6/AN1- 8 GND
9 AN3/AN3+ 10 GND
11 AN7/AN1- 12 GND
13 AN4/AN4+ 14 GND
15 AN8/AN1- 16 GND
17 GND 18 GND
19 AOUT1 20 GND
21 AOUT2 22 GND
23 GND 24 GND
25 PC7 26 PC6
27 PC5 28 PC4
29 PC3 30 PC2
31 PC1 22 PC0
33 EXTCLK 34 GND
35 EXTGATE 36 CLKOUT
37 +12V 38 +5V
39 -12V 40 GND

4.4.8 Proceduri de calibrare

Se prezinta procedurile de calibrare pentru domeniul tensiunii de intrare a convertorului A/N si câstigul acestuia.

În funcție de comanda specifica a utilizatorului performanțele ADA2100: offsetul si capatul de scala pentru
convertor sunt calibrate de firma. Circuitul de câstig este de asemenea calibrat înainte de trimitere. Urmatoarea
procedura permite o verificare rapida a preciziei acestor circuite.

Dupa calibrare se alimenteaza calculatorul si se permite ADA2100 un interval de 15 min. pentru stabilizare.

Echipament necesar pentru calibrare

- surse de tensiune de precizie: 0 - +(-)10 V
- voltmetru digital: 5 - 1/2 digiți
- surubelnița mica (pentru ajustare potențiometre)

Calibrarea A/N

În timpul acestei proceduri trebuie facute conexiuni la câteva din intrarile analogice pe conectorul I/E extern P8.

Sunt necesare 2 ajustari pentru a calibra complet convertorul A/N pentru operații unipolare sau bipolare.

Acestea afecteaza offset-ul si capatul de scala. Ambii pasi ai calibrarii se efectueaza folosind potențiometrele
TR5 si TR6 sau TR6 si TR7. TR5 sau TR7 se folosesc pentru a aduce la zero eroarea de offset a convertorului
A/N si TR6 se foloseste pentru ajustarea capatului de scala. În urmatoarea procedura se foloseste canalul
analogic 1 de intrare si se alege câstigul 1 pentru el. Aceasta se realizeaza prin scrierea de zerouri la locația de
adresa I/E BA+1. Trebuie sa va asigurați ca secțiunea 4 a comutatorului S1 este setata pentru polaritatea dorita
si jumperul de pe conectorul P9 este în poziția 10V.

*Calibrarea unipolar*a

Sunt necesare doua ajustari pentru a calibra convertorul A/N pentru domeniul unipolar de 0...+10V, una pentru
offset si cealalta pentru capatul de scala. Pentru a ajusta offset-ul, o tensiune analogica de intrare foarte mica,
aratata sub rubrica „Offset” în tabelul de mai jos, este aplicata la canalul 1 al multiplexorului (P8-1). La P8-2

se conecteaza referința pentru masa. Afisând continuu conversiile analog/digitale se ajusteaza TR7 pâna când
datele oscileaza între cele doua valori specificate în tabel la rubrica „Offset”.
Dupa aceea se foloseste TR6 pentru ajustarea valorii capatului de scala. Desi tensiunile de intrare pentru capatul
de scala specificate în tabel nu reprezinta de fapt capatul de scala pentru un domeniu ideal de 0...+10V, ele
reprezinta tensiunile maxime pentru care conversia analog digitala este garantata a fi liniara. Orice valoare peste
aceasta limita depaseste domeniul liniar si afecteaza calibrarea corecta. Dupa conectarea tensiunii de capat de
scala la canalul 1 de intrare, se ajusteaza TR6 pâna când datele vor oscila între cele doua valori din rubrica
„Cap de scala”.
*Calibrare unipolar*a *(domeniul 0...+10V)*
| Offset (TR7) | Cap de scala (TR6)

Tensiune de | |
intrare | +1,22070 mV | +9,49829

Date A/N | 0000 0000 0000 | 1111 0011 0010
| 0000 0000 0001 | 1111 0011 0011
*Calibrare bipolar*a
Daca se selecteaza un domeniu de intrare pentru tensiuni de la -5V la +5V sau -10V la +10V procedura de
calibrare este aceeasi. Deci jumperul pe conectorul P9 va fi instalat în poziția 10V. Daca se lucreaza cu gama de
tensiuni -10V...+10V, se va repoziționa jumperul pe poziția 20V numai dupa efectuarea procedurilor de
calibrare de mai jos.
Sunt necesare doua ajustari pentru a calibra convertorul, una pentru offset iar cealalta pentru capat de scala.
Pentru a ajusta offsetul se conecteaza o tensiune ca cea afisata în rubrica „Offset” din tabelul de mai jos la
canalul 1 al multiplexorului. În timp ce se afiseaza rezultatul conversiei pe 12 biți, se ajusteaza TR5 pâna când
data va lua una din valorile din rubrica „Offset”. Dupa aceea se conecteaza tensiunea de capat de scala din tabel
la canalul 1 si se ajusteaza TR6 pâna când data ia valoare între cele doua limite din tabel în rubrica „Capat de
scala”.
Calibrare bipolara (domeniul -5V...+5V sau -10V...+10V)
| Offset (TR5) | Cap de scala (TR6)

Tensiune de | |
intrare | -4,99878 V | +4,99634 V

Date A/N | 0000 0000 0000 | 1111 1111 1110
| 0000 0000 0001 | 1111 1111 1111
Tabelul de mai jos ofera o referința pentru tensiunea de intrare ideala pentru convertorul A/N pentru ponderea
fiecarui bit în fiecare domeniu de tensiuni.
Tabelul arata capatul ideal de scala (toate pozițiile 1) în prima linie si apoi câte un zero în liniile urmatoare
corespunzator unor biți. Valorile tensiunilor sunt date în mV.

Ponderea biților pentru convertorul A/N

Pondere bit A/N	Tensiune de intrare ideala (mV)		
	-5V...+5V	-10V...+10V	0...+10V
4095-Cap scala	+4997,6	+9995,1	+9997,6
2048	0000,0	0000,0	+5000,0
256	-4375,0	-8750,0	+625,0
32	-4921,9	-9843,8	+78,125
4	-4990,2	-9980,5	+9,7656
0	-5000,0	-10000,0	0,0000

Calibrarea circuitului de câstig

Pentru ajustarea circuitului de câstig se folosesc potențiometrele TR1-TR4, câte unul pentru câstigurile de 2, 4, 8 si respectiv 16. Pentru a calibra acest circuit se aplica o tensiune de 39,063 mV la intrarea canalului 1. Dupa aceea, prin scrierea cuvântului corect la locația BA+1, se seteaza câstigul la 2 si se ajusteaza potențiometrul TR1 pentru a obține iesirea convertorului A/N pe 12 biți pentru gama tensiunilor pe modul asa cum se prezinta în tabelul de mai jos. Se repeta dupa aceea procedura pentru celelalte 3 câstiguri ajustând potențiometrul corespunzator pâna se ating valorile corecte din tabelul de mai jos:

Câstig	Potențiometrul	Domeniul tensiunii de intrare		
		-5V...+5V	-10V...+10V	0...+10V
2	TR1	100000100000	100000010000	000000100000
4	TR2	100001000000	100000100000	000001000000
8	TR3	100010000000	100001000000	000010000000
16	TR4	100100000000	100010000000	000100000000

4.4.9 Conectarea modulului la proces

Legarea la proces se face prin intermediul unui conector de 40 de pini montat pe modul (P8). În conector se introduc panglicile cu conectoare pereche ale celor de pe modul, acestea fiind livrate o data cu modulul si etichetate la capatul dinspre proces. Panglica pentru semnale analogice este realizata într-un mod adecvat semnalelor analogice, având fiecare intrare torsadata cu câte un fir de masa. Legarea efectiva la proces se face prin intermediul conectoarelor originale ale panglicilor.

4.4.10 Programe de test

Exista programe de test pe disc flexibil realizate de firma "RTD" Inc – vezi laborator.

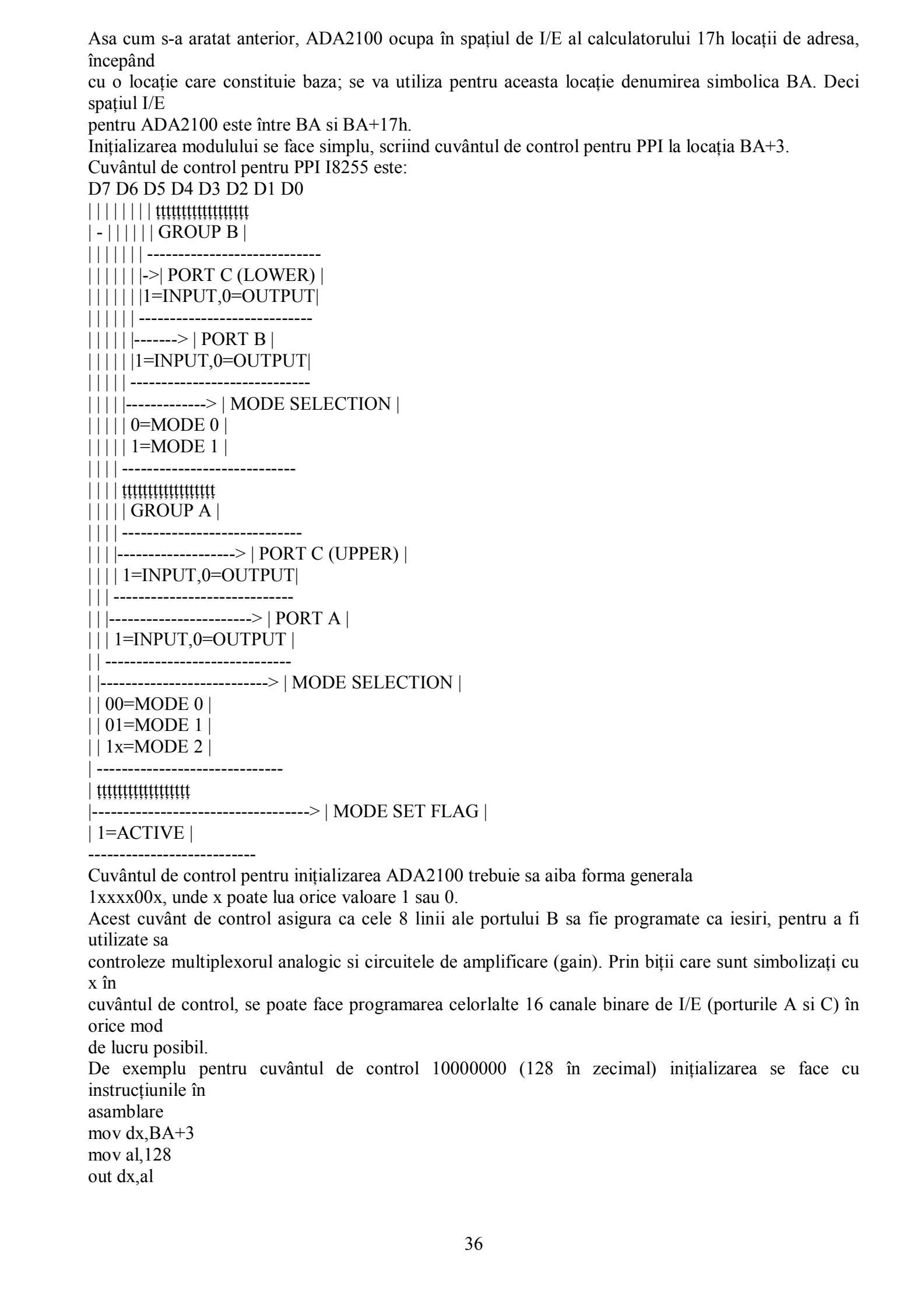
4.4.11 Programarea modulului ADA 2100

În acest paragraf se prezinta modul de programare a modulului ADA2100 pentru achiziția pe canalele analogice de intrare; programarea interfețelor PPI I8255 si PIT I8254 este prezentata în foile de catalog.

Inițializarea ADA2100

Înainte de a utiliza în program ADA2100, modulul trebuie inițializat. Acest lucru trebuie facut ori de câte ori calculatorul este pornit sau resetat. Prin inițializare se programeaza PPI astfel încât sa comunice cu circuitele de conversie analog/numerica. Daca nu se face inițializarea, modulul nu va raspunde corect la comenzi, iar efectele sunt imprevizibile.

Asa cum s-a aratat anterior, ADA2100 ocupa în spațiul de I/E al calculatorului 17h locații de adresa, începând
cu o locație care constituie baza; se va utiliza pentru aceasta locație denumirea simbolica BA. Deci spațiul I/E
pentru ADA2100 este între BA si BA+17h.
Inițializarea modulului se face simplu, scriind cuvântul de control pentru PPI la locația BA+3.
Cuvântul de control pentru PPI I8255 este:



Cuvântul de control pentru inițializarea ADA2100 trebuie sa aiba forma generala
1xxxx00x, unde x poate lua orice valoare 1 sau 0.
Acest cuvânt de control asigura ca cele 8 linii ale portului B sa fie programate ca iesiri, pentru a fi utilizate sa
controleze multiplexorul analogic si circuitele de amplificare (gain). Prin biții care sunt simbolizați cu x în
cuvântul de control, se poate face programarea celorlalte 16 canale binare de I/E (porturile A si C) în orice mod
de lucru posibil.
De exemplu pentru cuvântul de control 10000000 (128 în zecimal) inițializarea se face cu instrucțiunile în
asamblare

```
mov dx,BA+3
mov al,128
out dx,al
```

În acest caz, si porturile C si A sunt programate ca iesiri, putând fi disponibile pentru comenzi în proces la
conectorul P8.
Daca inițializarea se face cu 10001001 (137 zecimal), portul C este programat ca intrari numerice.
De notat ca portul A bitul 7 (PA7), poate fi configurat sa permita monitorizarea semnalului EOC de la convertorul analog-numeric. În acest caz, trebuie ca portul A sa fie programat ca intrari numerice, cuvântul de
control având forma generala 1xx1x00x.
Portul B este dedicat pentru selecția canalului analogic de intrare si a amplificarii. Cei mai puțin semnificativi 4
biți ai portului B (PB0-:-PB3) controleaza selecția canalului, iar cei mai semnificativi 4 biți controleaza selecția
amplificarii.
Asignarea biților în port este:
MSBs LSBs
7 6 5 4 3 2 1 0 PPI port B (BA + 1)
\ţţţ^ţţţţ/ \ţţţ^ţţţţ/
gain select channel select
0000 = 1x 0000=channel 1
0001 = 2x 0001=channel 2
0010 = 4x 0010=channel 3
0100 = 8x 0011=channel 4
1000 = 16x 0100=channel 5
0101=channel 6
0110=channel 7
0111=channel 8
Imediat dupa inițializare, portul B se încarca cu valoarea 0, corespunzator canal 1, amplificare 1x. Pentru a
schimba aceasta programare, de exemplu pentru canal 8 amplificare 2x, se scrie secvența de program
mov dx,BA+1
mov al,00010111b
out dx,al
Selectarea unui canal analogic de intrare
Canalul analogic de intrare se selecteaza cu cei mai puțin semnificativi 4 biți din portul B al PPI, care se gaseste
la adresa BA+1.
Algoritmul general de selectare a canalului este:
1. Citeste starea curenta a portului B
2. Reține cei mai semnificativi 4 biți, pentru a pastra amplificarea
3. Depune numarul de canal în cei mai puțin semnificativi biți
4. Scrie noua valoare în portul B.
O secvența de program în asamblare pentru selecția canalului 6 (de exemplu) este:
mov dx,BA+1
in al,dx
and al,0f0h
or al,6-1
out dx,al
Setarea amplificarii pe intrarile analogice
Configurația celor mai semnificativi 4 biți ai portului B pentru setarea amplificarii a fost prezentata mai sus.
Algoritmul general pentru setarea amplificarii este:
1. Citeste starea curenta a portului B
2. Pastreaza cei mai puțin semnificativi 4 biți, pentru a nu schimba canalul
3. Seteaza noua amplificare cu una din valorile (în zecimal):
pentru 1x ---- 0

pentru 2x ---- 16
pentru 4x ---- 32
pentru 8x ---- 64
pentru 16x ---- 128
4. Scrie noua valoare în portul B. Secvența de program în asamblare pentru setarea amplificarii 8x (de exemplu) este:
mov dx,BA+1
in al,dx
and al,0fh
or al,64
out dx,al
Programarea si citirea convertorului analog-numeric
Dupa ce s-a facut selecția canalului si setarea amplificarii, se poate face citirea valorii convertita, ori de câte ori
este nevoie. Schimbarea canalului si/sau amplificarii se face numai atunci când este nevoie de alte valori pentru
ele, nu la fiecare citire din convertor.
Ori de câte ori o conversie analog-numerica este completa, se genereaza semnalul end-of-convert (EOC) pentru
a specifica sfârsitul conversiei. Acest semnal poate fi utilizat în mai multe feluri:
• conectat la bitul 7 din portul A al PPI (PA7)
• conectat la bitul 7 din portul C al PPI (PC7)
• conectat la una din întreruperile IRQ2-:-IRQ7.
Conectarea la PA7 sau PC7 se face în conectorul P6, iar la una dintre întreruperi în conectorul P7.
Algoritmul general pentru controlul conversiei analog-numerice pe 12 biți si citirea valorii convertite este:
1. Starteaza o conversie pe 12 biți, executând o instrucțiune OUT la BA+4 (sau BA+6); nu conteaza valoarea care se scrie în port
2. Asteapta 20 microsecunde sau testeaza prin polling starea semnalului EOC pe PA7 (sau PC7)
3. Citeste octetul cel mai puțin semnificativ al valorii convertite, de la BA+5 (sau BA+7)
4. Citeste octetul cel mai semnificativ al valorii convertite, de la BA+4 (sau BA+6)
5. Combina cei doi octeți într-o valoare pe 12 biți, utilizând relația:
rezultat=(MSB*16)+(LSB div 16)
Este nevoie de ajustarea valorii citite (pasul 5), deoarece modulul ofera o valoare pe 12 biți prin intermediul a
doua porturi pe 8 biți; acestea constituie un cuvânt pe 16 biți, în care cei 12 biți ai valorii convertite sunt cadrari
la stânga, ca în figura de mai jos:
MSB LSB
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
DB12 DB11 DB10 DB9 DB8 DB7 DB6 DB5 DB4 DB3 DB2 DB1 0 0 0 0
Formatul cuvântului de conversie A/N
Dupa ce se citeste si ajusteaza valoarea, aceasta trebuie scalata în funcție de domeniul si polaritatea tensiunii
de intrare:
Domeniu de intrare Factor de scalare Greutatea pe bit
---------- ---------- ----------
+/- 5V Scade 2048 2.4414 mV
+/- 10V Scade 2048 4.8828 mV
0 la 10V NU 2.4414 mV
De exemplu, daca valoarea citita este 1024 iar domeniul de intrare este +/-5V, tensiunea analogica de intrare
este calculata astfel:
(1024-2048)*2.4414mV/bit=-2.49999 V.
Pentru +/-10V la intrare, aceasta tensiune se calculeaza astfel:

(1024-2048)*4.8828mV/bit=-4.9999V.
Pentru 0 -:- 10V la intrare, nu se face scalare, si tensiunea se calculeaza astfel:
1024*2.4414mV/bit=2.49999 V.
Pentru conversia din volți în unitați ingineresti corespunzatoare marimii achiziționate (atm, bari, Amperi, m/s
etc), se ține cont daca traductorul de intrare are sau nu caracteristica de intrare/iesire liniara. Daca da, se poate
face translatarea de domeniu din volți în valori ingineresti, în caz contrar urmând a se consulta cartea tehnica a
traductorului.
De notat ca ADA2100 efectueaza si conversii pe 8 biți. Startul conversiei se da cu o instrucțiune OUT la adresa
BA+5 (sau BA+7). Desi are o rezoluție mai proasta, conversia pe 8 biți este uneori preferata, datorita vitezei mai
mari a ciclului de conversie (13μs în loc de 20μs cât este la conversia pe 20biți).
Observații:
1. În aplicații, este bine ca chiar daca programarea conversiei si citirea datelor se fac în limbaj de asamblare, conversia în volți si în unitați ingineresti sa se faca în limbaj evoluat (C, PASCAL, BASIC etc.). Limbajele evoluate conțin de regula si funcții de I/E care permit programarea porturilor.
2. Pentru aprofundarea modului de programare a modulului ADA2100 în diverse limbaje, se vor studia programele sursa livrate de firma producatoare. Acestea conțin exemple care prezinta diverse moduri de utilizare a modulului ADA2100 pentru achiziția de date.
*Problem*a *propus*a
Sa se adapteze programul de prezentat la 4.3.4 si sa se scrie funcțiile can()si cna()pentru cazul în care sistemul de achiziție de date si control este realizat cu ADA2100.