

L11. NOȚIUNEA DE ÎNTRERUPERE. APLICAȚII DE ROTAȚIE A LED-ULUI APRINS LA APĂSAREA UNUI BUTON CARE GENEREAZĂ ÎNTRERUPERE.

1. Obiective

Prin parcurgerea acestei ședințe de laborator studenții vor fi capabili:

- Să definească o rutină de tratare a unei întreruperi;
- Să valideze o cerere de întrerupere;
- Să valideze global cererile de întrerupere;
- Să transfere corect informații între rutinele de tratare a întreruperilor și programul principal

2. Întreruperile microprocesorului 8085

Microprocesorul 8085A are integrat un controler de gestionare a cererilor multiple pentru cele 5 linii distincte de întrerupere accesibile la pini.

Linia de întrerupere	Adresa de salt	Nivel de prioritate	Mod de activare	Tipul întreruperii	Utilizare în cadrul microsistemului
TRAP	0024h	1 (max)	front și nivel	nemascabilă	Buton TRP
RST 7.5	003Ch	2	front	mascabilă	Buton INT 7.5
RST 6.5	0034h	3	nivel logic "1"	mascabilă	SW pas cu pas
RST 5.5	002Ch	4	nivel logic "1"	mascabilă	Controler tastatură
INTR	Data de CALL addr	5 (min)	nivel logic "1"	mascabilă	Controler de întreruperi

În tabelul de mai sus se prezintă numele acestor pini, adresele de tratare a solicitărilor, prioritatea lor, modul de prezentare a cererilor, tipul întreruperilor și modul de utilizare a acestora în cadrul microsistemului EMAC Universal Trainer din laborator. Programul monitor rezident pe microsistem, fiind amplasat în memoria EPROM (care începe de la adresa 0000h), este primul care interceptează întreruperile sosite pe primele 4 linii din tabel. Cele prezentate pe linia INTR, la care este conectat un controler de întreruperi 8259A, pot fi amplasate oriunde în memorie, deoarece pentru a ajunge la adresa de tratare controlerul de întreruperi folosește instrucțiuni de tip CALL *addr*.

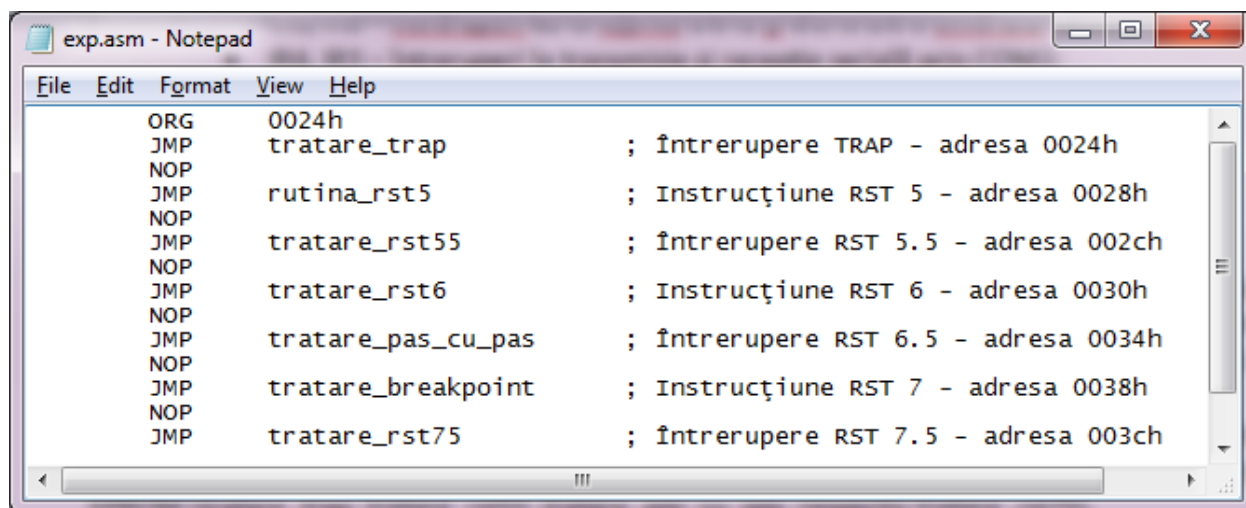
În figura de mai jos se prezintă structura hardware a sistemului de întreruperi al microsistemului din laborator. Liniile TRAP și RST 7.5 pot fi activate de către utilizator la apăsarea butoanelor TRP, respectiv INT 7.5. Întreruperea nemascabilă TRAP este utilizată de programul monitor pentru a opri execuția programului utilizatorului la apăsarea butonului TRP. Întreruperea RST 7.5 nu este utilizată de monitor, fiind la dispoziția programului utilizatorului.

Pe linia INTR se conectează un controler de întreruperi 8259A, care gestionează 8 linii întrerupere:

- IR7 – întrerupere de la convertorul A/D;
- IR6, IR5 – întreruperi de la ieșirile OUT2 și OUT0 ale a timerului 8253;
- IR4, IR3 – întreruperi la transmisie și recepție serială prin COM2;
- IR2, IR1 – întreruperi la transmisie și recepție serială prin COM1;
- IRO – întrerupere comandată de la butonul INTO.

2.1. Utilizarea liniilor de întrerupere cu adrese de tratare fixate

În cadrul liniilor de întrerupere cu adrese de tratare fixate intră și liniile TRAP și RST 7.5. Codul sursă al programului monitor din zona adreselor de tratare a acestor întreruperi este prezentat în cele ce urmează:



```

exp.asm - Notepad
File Edit Format View Help
ORG      0024h
JMP      tratare_trap      ; întrerupere TRAP - adresa 0024h
NOP
JMP      rutina_rst5       ; Instrucțiune RST 5 - adresa 0028h
NOP
JMP      tratare_rst55     ; întrerupere RST 5.5 - adresa 002ch
NOP
JMP      tratare_rst6      ; Instrucțiune RST 6 - adresa 0030h
NOP
JMP      tratare_pas_cu_pas ; întrerupere RST 6.5 - adresa 0034h
NOP
JMP      tratare_breakpoint ; Instrucțiune RST 7 - adresa 0038h
NOP
JMP      tratare_rst75     ; întrerupere RST 7.5 - adresa 003ch
  
```

Se observă că la adresele fixe de tratare a întreruperilor TRAP și RST se află amplasate instrucțiuni de salt mai departe, la rutinele propriu-zise de tratare: *tratare_trap*, *tratare_rst55*, *tratare_pas_cu_pas*, respectiv *tratare_rst75*.

Întreruperea nemascabilă TRAP este tratată în programul monitor. În varianta actuală de monitor, programul utilizatorului nu poate schimba rutina de tratare a întreruperii nemascabile cu una proprie, deoarece această întrerupere este folosită în exclusivitate de monitor pentru a opri execuția programului utilizatorului. Linia de întrerupere TRAP este nemascabilă, adică cererile sunt întotdeauna servite de către microprocesor, la sfârșitul instrucțiunii curente, oricare ar fi aceasta. Celelalte 4 linii sunt mascabile, adică este posibil ca microprocesorul să ignore cererile dacă liniile de întrerupere mascabile sunt inhibitate la nivel global (**cu instrucțiunea DI – Disable Interrupts**) sau doar individual (numai liniile RST x.5), cu instrucțiunea **SIM - Set Interrupt Mask**.

După resetare, sistemul de întreruperi este inactiv. Întreruperile pot fi validate individual (cu aceeași instrucțiune **SIM – Set Interrupt Mask**) și global (cu **instrucțiunea EI – Enable Interrupts**) dacă dorim să permitem tratarea lor de către microprocesor.

Instrucțiunea **SIM (Set Interrupt Mask)** utilizează octetul din acumulator și permite realizarea următoarelor funcții:

7	6	5	4	3	2	1	0
SOD	SOE	x	R7.5	MSE	M7.5	M6.5	M5.5

- resetarea bistabilului care reține cererea de întrerupere apărută pe RST 7.5, înainte de a fi tratată (dacă R7.5=1);
- mascarea/demascarea individuală a liniilor de întrerupere RST (M7.5, M6.5 și M5.5), dacă MSE=1 (Mask Set Enable). Ex.: Dacă MSE=1, M5.5=1 și M7.5=0, atunci linia RST 5.5 este mascată, iar linia RST 7.5 este validată.

După acceptarea unei întreruperi pe oricare din liniile RST, toate întreruperile mascabile sunt dezactivate automat (IE=0) și ele trebuie revalidate cu o instrucțiune EI. EI și DI nu afectează starea măștilor individuale pentru liniile RST.

La rândul ei, **instrucțiunea RIM (Read Interrupt Mask)** permite citirea în acumulator a stării sistemului de întreruperi:

- citirea stării bistabililor de validare (IE) și de mascare a întreruperilor RST (M7.5, M6.5, M5.5);
- citirea solicitărilor de întrerupere pe liniile RST în curs de servire (I7.5, I6.5, I5.5).

7	6	5	4	3	2	1	0
SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5

Starea sistemului de întreruperi poate fi urmărită și în NoICE, pentru că octetul care poate fi citit cu instrucțiunea RIM este afișat pe bara de registre ca fiind IM – Interrupt Mask.

Linia de întrerupere RST 7.5 poate fi activată de la butonul INT7.5, iar la adresa *tratare_rst75* se află următoarea secvență de instrucțiuni:

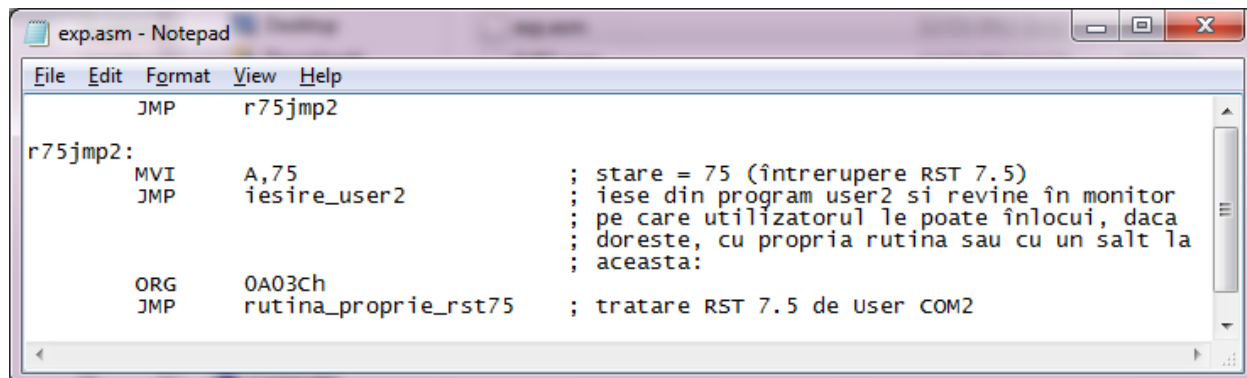
```

exp.asm - Notepad
File Edit Format View Help
|      JMP      r75jmp2
r75jmp2:
MVI     A,75          ; stare = 75 (întrerupere RST 7.5)
JMP     iesire_user2   ; iese din program user2 si revine în m
                        ; pe care utilizatorul le poate înlocui
                        ; dorește, cu propria rutina sau cu un
                        ; aceasta:
ORG     0A03Ch
JMP     rutina_proprie_rst75 ; tratare RST 7.5 de User COM2
  
```

Dar ce se întâmplă dacă apare întreruperea RST 7.5, dar utilizatorul nu dorește sa trateze această întrerupere și în consecință nu a prevăzut o rutină de tratare a ei la adresa *user_codeX+3Ch* ?

Dacă întreruperile nu sunt validate (IE=0), nu se întâmplă nimic, întreruperea rămâne în așteptare, căci microprocesorul o ignoră. Dar programul de aplicație poate să aibă nevoie să lucreze cu altă linie de întrerupere, deci sistemul de întreruperi va trebui să fie la un moment dat validat (cu instrucțiunea EI).

În acest caz, trebuie inhibate liniile de întrerupere pentru care nu se prevăd rutine de tratare, cu ajutorul instrucțiunii SIM. Programul monitor își ia o măsură de precauție suplimentară, astfel încât dacă programul utilizatorului „uită” să inhibe o linie de întrerupere pentru care nu a prevăzut rutină de tratare sau o validează din greșeală, atunci la adresa *user_codeX+3Ch* microprocesorul va găsi o rutină de tratare pusă de monitor și care face același lucru ca și întreruperea TRAP: este oprit programul utilizatorului și controlul revine programului monitor. Spre exemplu, în zona de memorie RAM rezervată utilizatorului conectat pe COM2, la adresa 903Ch se găsește următoarea secvență de instrucțiuni, pusă acolo la inițializarea microsistemului:



```

exp.asm - Notepad
File Edit Format View Help
JMP r75jmp2
r75jmp2:
MVI A,75 ; stare = 75 (întrerupere RST 7.5)
JMP iesire_user2 ; iese din program user2 si revine în monitor
; pe care utilizatorul le poate înlocui, daca
; doreste, cu propria rutina sau cu un salt la
; aceasta:
ORG 0A03Ch
JMP rutina_proprie_rst75 ; tratare RST 7.5 de User COM2

```

Codul din acumulator (75) ne permite apoi să identificăm ușor cauza care a produs oprirea programului (în acest caz întreruperea RST 7.5 incorect gestionată).

Exemplul 1:

Programul care urmează afișează pe primele 4 celule ale afișajului numărul de apăsări pe butonul INT7.5. Pentru aceasta se folosește o variabilă contor, care este inițializată cu 0 în programul principal și este incrementată cu 1 în rutina de tratare a întreruperii definită în programul utilizatorului. În varianta actuală, programul principal afișează această variabilă într-o buclă infinită.

```

exemplul1.asm - Notepad
File Edit Format View Help
include "services.asm"
ORG      8000h          ; 9000h pentru COM2
start:
MVI      A,00h          ; Fortez A=00h
MOV      D,A            ; D=0
STA      contor          ; contor = 0
RIM      ; citesc în A starea sistemului de întreruperi
ANI      00011011b      ; forteaza M7.5=0
ORI      00011000b      ; forteaza R7.5=MSE=1
SIM      ; sterge o eventuala cerere pe RST 7.5 în
              ; asteptare si valideaza linia RST 7.5
EI        ; valideaza întreruperile mascabile
loop:
LDA      contor          ; A=contor
MOV      E,A            ; E=contor si DE=contor (pentru ca D=0)
MVI      C,LEDDEC        ; C=codul functiei de afisare
CALL     BIOS            ; Afisare contor
JMP      loop            ; reia

; rutina de tratare a întreruperii RST 7.5.
; se ajunge aici având pe stiva adresa de revenire în programul principal si
; PSW, deci PSW nu mai trebuie salvat în rutina utilizatorului, dar trebuie
; extras la revenire
ORG      803Ch          ; 903Ch pentru COM2
tratare_rst75:
LDA      contor          ; A = contor
INR      A              ; contor=contor+1
STA      contor          ; actualizare contor
POP      PSW             ; refac PSW salvat de monitor
EI        ; se revalideaza întreruperile mascabile
RET       ; se revine în programul întrerupt

contor: DS      1        ; rezerva 1 octet pentru contor

```

2.2. Utilizarea întreruperilor gestionate de controlerul 8259A

Pentru a fi utilizate întreruperile gestionate de controlerul de întreruperi 8259A, acesta trebuie inițializat cu o secvență de maxim 4 cuvinte de comandă de inițializare denumite **ICW1**, **ICW2**, **ICW3** și **ICW4** (**ICW – Initialization Command Word**).

ICW1 și ICW2 sunt suficiente pentru ca un sistem format dintr-un singur controler să funcționeze în modul cu priorități fixe. Ele stabilesc adresa de început a tabelului de întreruperi ce este organizat în memorie de fiecare utilizator.

Cuvântul de comandă de inițializare ICW1 se înscrie în registrul intern de comandă selectat cu A0=0.

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	A_7	A_6	A_5	1	LTM	ADI	S	IC4

A7÷A5 – biții de rang 7, 6 și 5 ai adresei tabelului de întreruperi, adresă utilizată de instrucțiunea CALL *addr*, furnizată de controler în ciclul de acceptare a întreruperii și utilizată de microprocesor pentru recunoașterea sursei întreruperii și apelarea rutinei de tratare specifice.

LTM – Level Triggered Mode – stabilește forma de prezentare a cererilor pe liniile IR0÷IR7:

- LTM = 0 – cereri active pe front crescător (Edge Triggered Mode)
- LTM = 1 – cereri active pe nivel ridicat.

În tabelul de întreruperi se alocă un spațiu de 4 sau 8 octeți pentru fiecare din cele 8 linii de întrerupere, în funcție de bitul **ADI (call Address Interval)**:

- ADI=1 – interval de 4 octeți
- ADI=0 – interval de 8 octeți

Rezultă o zonă de $4 \times 8 = 32 = 2^5$ octeți sau de $8 \times 8 = 64 = 2^6$ octeți denumită tabela cu vectori de întrerupere (TABVI), amplasată la o adresă multiplu de 32 sau de 64. Vectorii de întrerupere (adresele de tratare) se regăsesc în TABVI la adrese multiplu de 4, respectiv de 8.

S – Single – indică dacă există un singur controler în sistem (S=0 – mai multe controlere, în cascadă)

IC4 – indică dacă se folosește ICW4 (IC4=1) sau nu (IC4=0).

Cuvântul de comandă de inițializare ICW2 se înscrie în registrul intern de comandă selectat cu A0=1.

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈

A15÷A5 – cei mai semnificativi biți ai adresei tabelii vectorilor de întrerupere, adresă utilizată de instrucțiunea CALL *addr*, furnizată de controler în ciclul de acceptare a întreruperii și utilizată de μP pentru recunoașterea sursei întreruperii și apelarea rutinei de tratare specifice.

La apariția unei întreruperi pe una din liniile IR0-IR7 controlerul furnizează codul instrucțiunii CALL *addr*, câmpul de adresă indicând adresa rutinei de tratare amplasată în TABVI specifică unei anumite linii de întrerupere – cea pentru care este înregistrată cererea validă cea mai prioritară la momentul respectiv.

Pentru un interval de 4 octeți:

- biții A15÷A5 sunt furnizați de controler așa cum sunt primiți în ICW1 și ICW2.
- biții A4÷A2 ai adresei reprezintă codul liniei de întrerupere.
- biții A1 și A0 sunt forțați pe 0 - vector de întrerupere multiplu de 4.

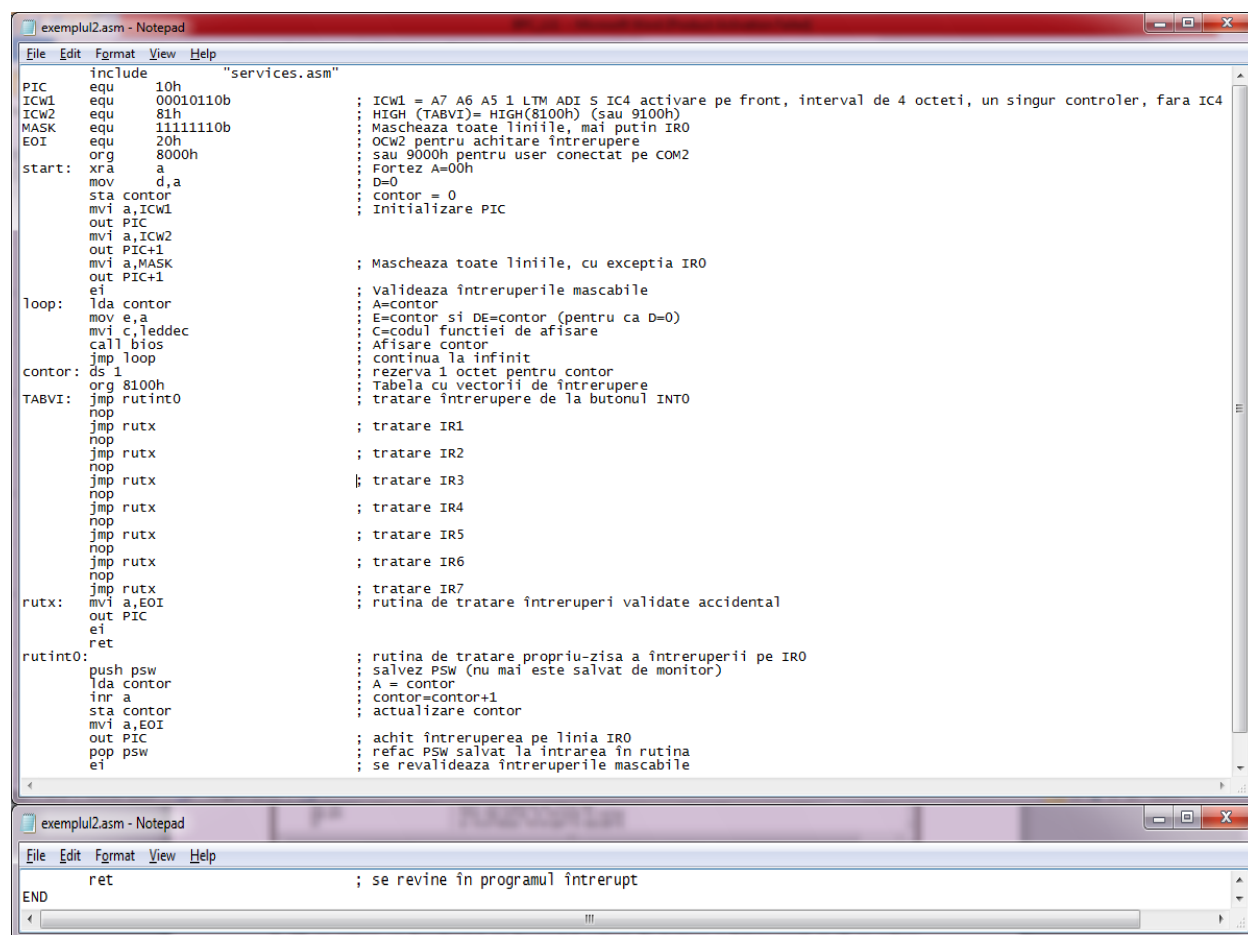
Stabilirea modului de funcționare a dispozitivului 8259A se realizează după inițializare, prin intermediul cuvintelor de comandă de operare (**Operation Command Words**) corespunzătoare.

OCW1 - asigură mascarea uneia sau mai multor linii de întrerupere și are forma următoare:

Mi=1 (i=7÷0) asigură mascarea liniilor de întrerupere IR_i

Exemplul de utilizare a întreruperii generate de la butonul INTO, similar cu cel anterior este prezentat în cele ce urmează:

Exemplul 2:



```

exemplul2.asm - Notepad
File Edit Format View Help
include "services.asm"
PIC equ 10h
ICW1 equ 00010110b ; ICW1 = A7 A6 A5 1 LTM ADI 5 IC4 activare pe front, interval de 4 octeti, un singur controler, fara IC4
ICW2 equ 81h ; HIGH (TABVI)= HIGH(8100h) (sau 9100h)
MASK equ 11111110b ; Mascheaza toate liniile, mai putin IR0
EOI equ 20h ; OCW2 pentru achitare întrerupere
org 8000h ; sau 9000h pentru user conectat pe COM2
start: xra a ; Fortez A=00h
mov d,a ; D=0
sta contor ; contor = 0
mvi a,ICW1 ; Initializare PIC
out PIC
mvi a,ICW2
out PIC+1
mvi a,MASK ; Mascheaza toate liniile, cu exceptia IR0
out PIC+1
ei ; Valideaza întreruperile mascabile
loop: lda contor ; A=contor
mov e,a ; E=contor si DE=contor (pentru ca D=0)
mvi c,1eddec ; C=codul functiei de afisare
call bios ; Afisare contor
jmp loop ; continua la infinit
contor: org 8100h ; rezerva 1 octet pentru contor
TABVI: jmp rutint0 ; Tabela cu vectorii de întrerupere
nop ; tratare întrerupere de la butonul INTO
nop ; tratare IR1
jmp rutx ; tratare IR2
nop ; tratare IR3
jmp rutx ; tratare IR4
nop ; tratare IR5
jmp rutx ; tratare IR6
nop ; tratare IR7
jmp rutx ; rutina de tratare întreruperi validate accidental
rutx: mvi a,EOI
out PIC
ei
ret
rutint0: ; rutina de tratare propriu-zisa a întreruperii pe IR0
push psw ; salvez PSW (nu mai este salvat de monitor)
lda contor ; A = contor
inr a ; contor=contor+1
sta contor ; actualizare contor
mvi a,EOI ; achit întreruperea pe linia IR0
out PIC ; refac PSW salvat la intrarea în rutina
pop psw ; se revalideaza întreruperile mascabile
ei

exemplul2asm - Notepad
File Edit Format View Help
ret ; se revine în programul întrerupt
END
  
```

Din exemplele prezentate se constată că transferul de informații între programul principal și rutinele de tratare a întreruperilor nu se poate face prin intermediul registrelor sau a stivei, așa cum se întâmpla cu subrutinele obișnuite, ci prin variabile amplasate în memorie la adrese considerate ca fiind cunoscute atât de programul principal, cât și de rutinele de tratare (vezi variabila contor).

Astfel de variabile sunt folosite atât pentru a transmite parametri către rutinele de tratare, cât și pentru a primi rezultate de la rutinele de tratare. Se observă că rutina de tratare primește valoarea curentă a contorului și furnizează valoarea actualizată a acestuia.

3. Aplicații propuse

3.1. Să se scrie codul în limbaj de asamblare pentru programul care urmează afișează pe primele 4 celule de pe display numărul de apăsări ale butonului INT7.5.

- a) Să se execute programul în regim pas cu pas, urmărindu-se registrele, variabila contor și starea sistemului de întreruperi (**IM**).
- b) Să se seteze un breakpoint pe începutul rutinei de tratare a întreruperii și să se ruleze programul liber (comanda **Run**).
- c) Să se observe oprirea programului la adresa de breakpoint la apăsarea butonului INT 7.5.
- d) Să se ruleze rutina de tratare în regim pas cu pas, urmărindu-se conținutul stivei și revenirea în programul întrerupt.

Indicații: Exemplul 1.

3.2. Exemplificați utilizarea întreruperii generate de la butonul INTO.

- a) Să se execute programul în regim pas cu pas, urmărindu-se registrele, variabila contor și starea sistemului de întreruperi (**IM**).
- b) Să se seteze un breakpoint pe începutul rutinei de tratare a întreruperii și să se ruleze programul liber (comanda **Run**).
- c) Să se observe oprirea programului la adresa de breakpoint la apăsarea butonului INT 0.
- d) Să se ruleze rutina de tratare în regim pas cu pas, urmărindu-se conținutul stivei și revenirea în programul întrerupt.
- e) Să se modifice programul astfel încât să utilizeze întreruperea RST7.5 pentru incrementarea contorului și întreruperea pe linia IRO pentru decrementarea lui.

Indicații: Exemplul 2.

3.3. *Să se scrie codul în limbaj de asamblare pentru programul care rotește un led la dreapta cu câte o poziție, de fiecare dată când este apăsat butonul INT7.5.

4. Referințe bibliografice

[1] *Sisteme cu microprocesoare*, Îndrumar de laborator.

[2] C.Huțanu, M.Postolache, *Sisteme cu microprocesoare*, Editura Academică, Iași, 2001.

[3] Gh.Toacșe, *Introducere în microprocesoare*, Editura Științifică și Enciclopedică, București, 1985.