



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

## Proiectarea cu Microprocesoare

### **10. Funcționarea în întreruperi**



Înteruperile microprocesorului 8086 se mai pot clasifica în trei grupe: înteruperi predefinite generate de funcții speciale, înteruperi hardware definite de utilizator și înteruperi software definite de utilizator. Vom descrie pe scurt aceste tipuri de înteruperi, vezi și figura 1.47.

## Înteruperi predefinite

La invocarea din hardware sau din software a unei înteruperi predefinite microprocesorul va transfera controlul rutinei a cărei adresă este specificată de vectorul asociat tipului de înterupere. Utilizatorul are sarcina să scrie rutinele de serviciu și să inițializeze tabela vectorilor cu adresele corespunzătoare.

*Tipul 0* – împărțire la 0. Această înterupere este invocată la orice încercare de împărțire pentru care câtul depășește valoarea maximă, cum este, de exemplu, cazul împărțirii la zero. Înteruperea nu este mascabilă și poate fi considerată ca o secvență aparținând operației de împărțire.

*Tipul 1* – pas-cu-pas. Înteruperea apare după o instrucțiune de la poziționarea indicatorului de condiție *Derută*, T, în registrul de stare al microprocesorului. Se permite astfel introducerea software a funcționării pas-cu-pas în cadrul unei secvențe de program. Secvența de înterupere începe cu salvarea registrului de stare și a numărătorului de program și ștergerea indicatorului T permițându-se astfel execuția normală, nu pas-cu-pas, a rutinei de serviciu specifice. La revenire în programul principal, modul de execuție pas-cu-pas, este asigurat prin restaurarea IP, CS și a indicatorilor de condiții, deci și a lui T. Nici această înterupere nu poate fi mascată.

*Tipul 2* – înterupere nemascabilă, NMI. Înteruperea este de tip hardware și are prioritatea cea mai înaltă. Intrarea corespunzătoare a microprocesorului, conexiunea NMI, este comutată pe front și apoi sincronizată în interiorul procesorului cu ceasul CLK. Pentru ca o înterupere NMI să fie recunoscută, semnalul intern sincronizat trebuie să fie activ cel puțin două perioade ale ceasului. De asemenea, dacă intrarea NMI rămâne pe "1" mai multă vreme, revenirea pe "0", timpul cât stă pe "0", înainte ca o nouă înterupere să fie declanșată este de minimum două perioade CLK. Semnalul de la intrarea lui 8086 poate fi dezactivat înainte de intrarea în rutina de serviciu. O atenție deosebită trebuie acordată eliminării *spike*-urilor, impulsurilor parazite, care pot genera înteruperi. Înteruperea NMI este rezervată, de obicei, evenimentelor catastrofale cum sunt căderile de tensiune sau semnalizările de la un sistem de supraveghere de tip *ceas de gardă*, *watchdog*.

*Tipul 3* – înterupere pe un octet. Este o înterupere software nemascabilă invocată de o instrucțiune reprezentată pe un octet, INT 3, destinată în primul rând ca înterupere de tip *breakpoint* pe parcursul punerii la punct a programelor.

*Tipul 4* – înterupere la depășire. Este o înterupere software nemascabilă care apare la execuția instrucțiunii INTO dacă indicatorul de condiție *Depășire*, O, este poziționat. Instrucțiunea permite derutarea programului la o rutină de serviciu în cazul apariției unei erori de depășire.

Înteruperile de tip 0 sau 2 pot apărea fără ca programatorul să acționeze în vreun fel specific, cu excepția unei împărțiri la 0 care ar putea fi o greșeală de programare. Înteruperile de tip 1, 3 și 4 necesită pentru a fi generate acte conștiente din partea programatorului. Toate tipurile de înteruperi prezentate mai sus, cu excepția NMI, sunt invocate software și sunt asociate direct cu câte o instrucțiune specifică.

## Întreruperi software definite de utilizator

Întreruperile software sunt generate de instrucțiunea `INT nn` unde `nn` reprezintă numărul tipului de întrerupere definit de utilizator. Instrucțiunile `INT`, deci întreruperile software, nu sunt mascabile cu ajutorul indicatorului de condiții *Validare/Invalidare Întrerupere*, `I`. Revenirea din rutina de serviciu apelată printr-o întrerupere software se face cu instrucțiunea `IRET`.

Transferul controlului la o întrerupere software se face la sfârșitul instrucțiunii `INT nn` fără ca microprocesorul să inițieze pe magistrală un ciclu de achitare `INTA`. De asemenea, întreruperile software vor invalida întreruperile mascabile prin punerea pe "0" a indicatorilor `I` și `T`.

## Întreruperi hardware definite de utilizator

Așa cum am mai spus, întreruperile hardware definite de utilizator sunt inițiate de circuite speciale prin activarea, trecerea pe "1", a semnalului de la intrarea `INTR` a procesorului. Aceste întreruperi sunt mascabile cu ajutorul bitului `I` din registrul de stare.

Starea intrării `INTR` este testată în timpul ultimului ciclu de ceas al fiecărei instrucțiuni. Excepție de la această regulă fac instrucțiunile `MOV` și `POP` cu un registru de segment, instrucțiunea `WAIT`, instrucțiunile pe șiruri precedate de prefixul de repetare `REP`.

În cazul instrucțiunilor `MOV` și `POP` cu un registru de segment microprocesorul va testa starea conexiunii `INTR` după executarea instrucțiunii următoare instrucțiunilor `MOV` și `POP` menționate. Astfel, se permite încărcarea unui *pointer* de stivă de 32 de biți în registrele `SS` și `SP` fără pericolul apariției unei întreruperi între cele două încărcări. O secvență cum este și cea de mai jos nu va fi deci interuptibilă:

```
mov ss, segment_stiva_nou
mov sp, pointer_stiva_nou
```

În timpul instrucțiunii `WAIT` care așteaptă trecerea pe "0" a intrării *TEST* a microprocesorului se testează și starea semnalului la conexiunea `INTR` pentru a se permite executarea rutinelor de întrerupere în timpul așteptării. Particularitatea în această situație constă în faptul că atunci când se detectează o întrerupere, 8086 mai extrage încă o dată instrucțiunea `WAIT` înainte de a transfera controlul rutinei de serviciu. Aceasta pentru a garanta revenirea din rutină tot la instrucțiunea de așteptare `WAIT`.

O altă situație specială apare și în cazul instrucțiunilor cu prefix. Deoarece prefixele sunt considerate ca parte a instrucțiunii pe care o preced, procesorul va eșantiona semnalul `INTR` la sfârșitul execuției instrucțiunii incluzând în aceasta și prefixele. Excepția survine la instrucțiunile pe șiruri precedate de prefixul `REP`. Pentru aceste instrucțiuni testarea întreruperii se face la sfârșitul fiecărei repetări. Atunci când instrucțiunile repetitive pe șiruri mai sunt precedate și de alte prefixe, de exemplu `LOCK`, și apare o întrerupere microprocesorul va restaura la revenirea din rutina de serviciu numai prefixul imediat precedent instrucțiunii. De aceea, pentru a nu perturba printr-o întrerupere execuția completă a instrucțiunii repetitive trebuie utilizată o secvență de program de tipul:

```
transfer_repetitiv: lock rep movs dest,cs:sursa  
                   and cx,cx  
                   jnz transfer_repetitiv
```

Codul obiect pe octeți generat de asamblorul 8086 pentru instrucțiunea MOVSB va fi în ordinea crescătoare a adreselor: prefixul LOCK, prefixul REP, prefixul *Depășire Segment (:)*, codul propriu-zis al instrucțiunii MOVSB. La revenirea dintr-o rutină de serviciu a întreruperii va fi refăcut numai prefixul *Depășire Segment* garantându-se astfel execuția corectă a încă unui transfer. Următoarea instrucțiune verifică terminarea transferului șirului prin testarea valorii contorului asociat instrucțiunii MOVSB, registrul CX. Dacă această valoare e diferită de zero se sare la începutul secvenței pentru a se termina execuția instrucțiunii repetitive cu prefixe.

### **Prioritatea întreruperilor și a cererii DMA**

Ordinea de prioritate a întreruperilor respectă următoarele reguli:

- INTR, întreruperea hardware, este singura întrerupere mascabilă și dacă ea este detectată simultan cu alte întreruperi, ștergerea indicatorului I din registrul de stare o va masca. Aceasta face ca INTR să fie de fapt întreruperea cu prioritatea cea mai scăzută atâta timp cât rutinele de serviciu ale celorlalte întreruperi nu revalidează INTR prin poziționarea lui I;

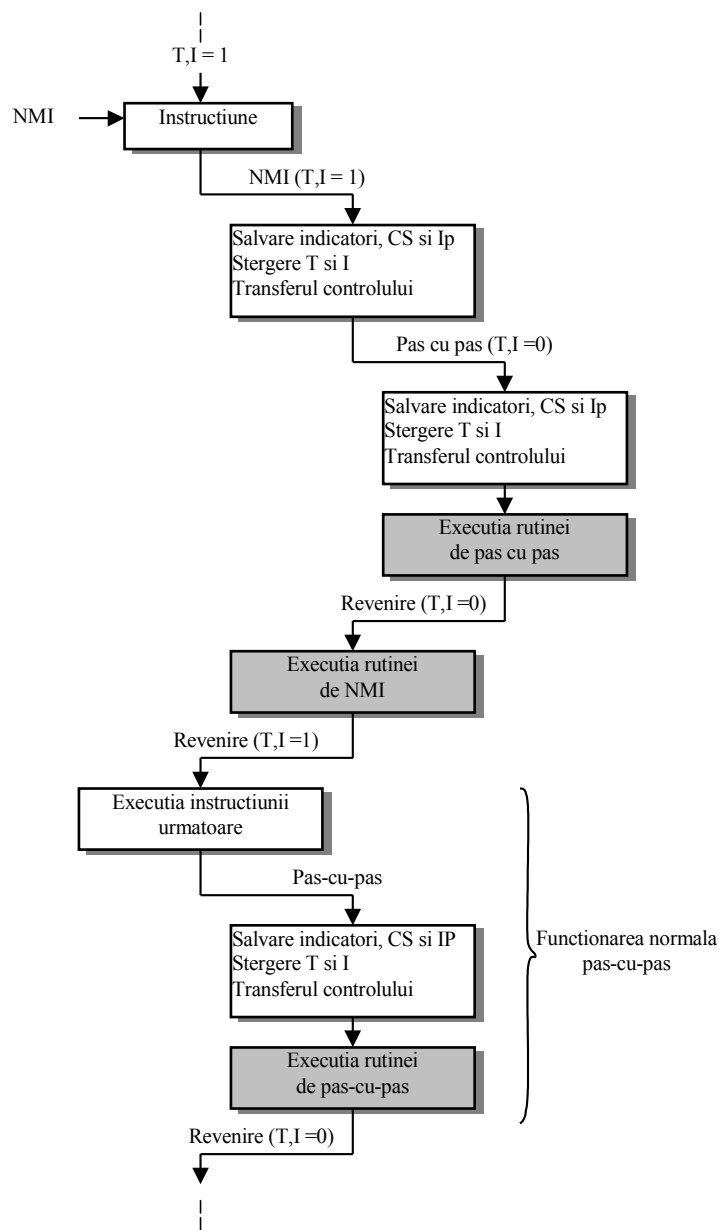


Figura 1.48. O întrerupere NMI pe timpul funcționării pas-cu-pas și funcționarea normală pas-cu-pas

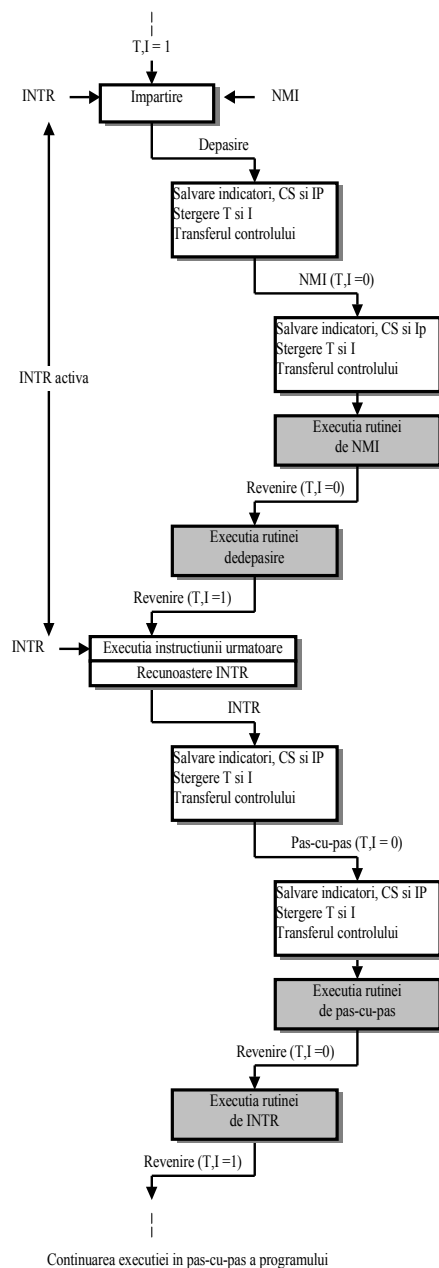


Figura 1.49. Apariția simultană a întreruperilor NMI, INTR, pas-cu-pas și întârzierii de depășire

În cadrul întreruperilor nemascabile, NMI, pas-cu-pas și întreruperi software, întreruperea pas-cu-pas are prioritatea cea mai înaltă urmată de NMI și întreruperile software. Această ordine de prioritate este valabilă numai în relațiile dintre două tipuri de întreruperi nemascabile, ca în exemplul din figura

1.48. Excepția apare în cazul în care toate cele trei întreruperi nemascabile sunt în așteptare. În această situație ordinea de execuție a acestor rutine este următoarea: NMI, întreruperea software, întreruperea pas-cu-pas. Așadar funcționarea pas-cu-pas se reia la execuția instrucțiunii următoare instrucțiunii care a provocat întreruperea software, figura 1.49.

În legătură cu cererea DMA precizăm că apariția simultană a unei cereri DMA și a întreruperii INTR, de exemplu în timpul execuției unei instrucțiuni precedate de prefixul LOCK, se rezolvă prin servirea întâi a cererii DMA și apoi a întreruperii, HOLD fiind deci prioritar față de INTR.