2.Rulare pas cu pas

Cuprins

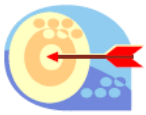
Cuprins Laborator 2

Cunoştinţe preliminare

- Înţelegerea principiului limbajului de asamblare şi a noţiunii de registru, cunoaşterea sistemului de numeraţie hexazecimal şi a conversiilor în bazele de numeraţie 2 şi 10.
- Cunoaşterea instrucţiunilor de MOV, OUT şi JMP.
- Cunoaşterea unui program Symbolic Debugger.

Introducere

În Laboratorul 2 studenţii rulează individual, pas cu pas, programe simple care aprind toate LED-urile, sting toate LED-urile, apoi crează propria secvenţă de aprindere, cât mai atractivă. La sfârşitul lucrării programele se rulează cu viteza procesorului şi se constată că aprinderea nu este cea aşteptată. Această observaţie este discutată cu studenţii.



Obiective

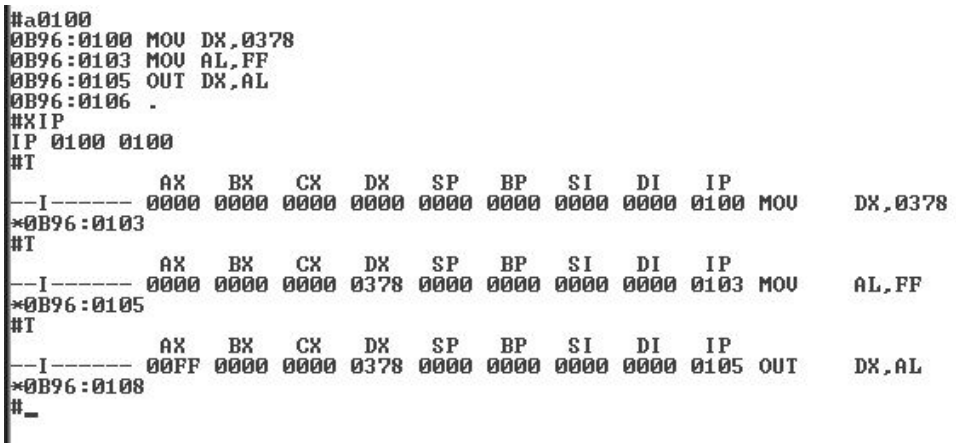
După parcurgerea acestui modul studenţii vor şti să lucreze cu programul SID ca să scrie şi să ruleze programe simple. Vor vedea modificarea conţinutului registrelor după fiecare instrucţiune şi astfel vor înţelege mai în detaliu funcţionarea procesorului.

Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

2.1.Scrierea unui program şi rularea pas cu pas

Primul program este cel care aprinde toate LED-urile prin rulare pas cu pas. Programul este dat în figura 2.1.

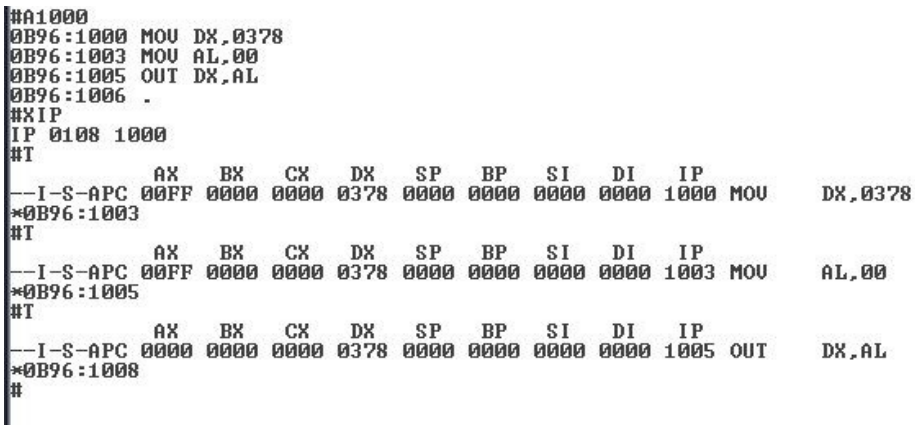


```
#a0100
0B96:0100 MOV DX,0378
0B96:0103 MOV AL,FF
0B96:0105 OUT DX,AL
0B96:0106 .
#XIP
IP 0100 0100
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I------ 0000 0000 0000 0000 0000 0000 0000 0000 0100 MOV     DX,0378
*0B96:0103
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I------ 0000 0000 0000 0378 0000 0000 0000 0000 0103 MOV     AL,FF
*0B96:0105
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I------ 00FF 0000 0000 0378 0000 0000 0000 0000 0105 OUT     DX,AL
*0B96:0108
#_
```

Figura 2.1. Program de aprindere a LED-urilor

Scrierea programului începe cu stabilirea adresei de început cu **Aadresa** în forma **A0100.** A se poate scrie cu litere mici sau mari. Se scriu apoi instrucțiunile pentru procesor, apoi se iese cu punct. Cu XIP se stabileşte adresa IP, adică adresa primei instrucțiuni de executat, adresa la care a început scrierea programului. Cu T se execută prima instrucțiune, apoi a doua, etc. Se observă după fiecare instrucțiune efectul asupra registrelor procesorului. După instrucțiunea de OUT toate LED-urile se vor aprinde.

Programul care stinge toate LED-urile este dat în figura 2.2.



```
#A1000
0B96:1000 MOV DX,0378
0B96:1003 MOV AL,00
0B96:1005 OUT DX,AL
0B96:1006 .
#XIP
IP 0108 1000
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I-S-APC 00FF 0000 0000 0378 0000 0000 0000 0000 1000 MOV     DX,0378
*0B96:1003
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I-S-APC 00FF 0000 0000 0378 0000 0000 0000 0000 1003 MOV     AL,00
*0B96:1005
#T
             AX   BX   CX   DX   SP   BP   SI   DI   IP
--I-S-APC 0000 0000 0000 0378 0000 0000 0000 0000 1005 OUT     DX,AL
*0B96:1008
#
```

Figura 2.2. Program de stingere a LED-urilor

Al doilea exemplu de program este cel care stinge toate LED-urile, a cărui scriere şi rulare este asemănătoare cu a programului precedent. Scrierea programului s-a făcut la altă adresă.

2.2.Scrierea unui program de aprindere de către fiecare student

Pentru scrierea următorului program studenții pot să-şi manifeste imaginația. Se cere scrierea unui program care rulat pas cu pas să realizeze o secvență cât mai interesantă, creată individual de fiecare student. Primul student care termină şi studentul care realizează secvența cea mai frumoasă, aleasă prin vot al studenților sunt bonificați.

Cel mai simplu de realizat este scrierea unui program care aprinde cele 8 LED-uri cu 8 secvențe de OUT, conform cu schema logică din figura 2.3.

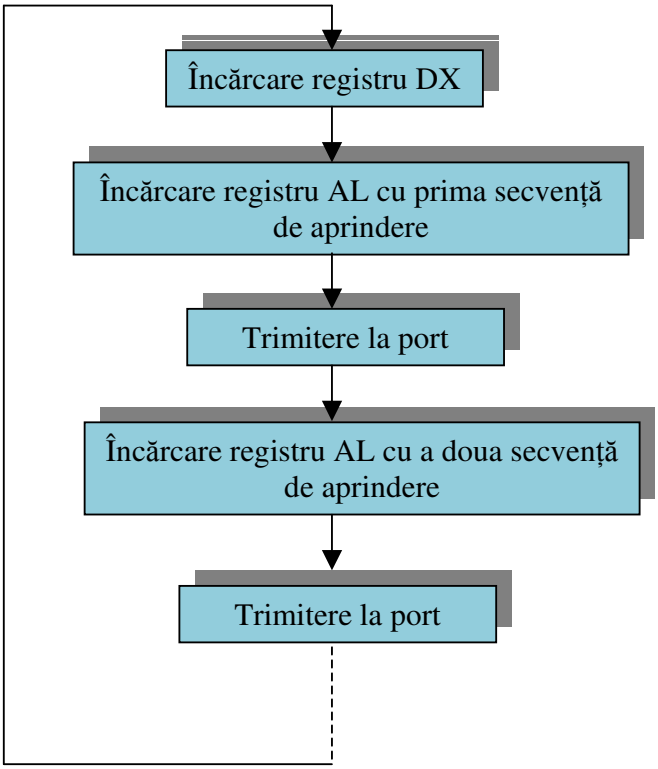


Figura 2.3. Schema logică a programului de trimitere secvențe de aprindere

O altă variantă mai elegantă care ar putea fi luată în considerare este modificarea în buclă a registrului AL care duce la scrierea unui cod mai compact. Sunt foarte simple variantele de a afişa pe LED-uri o numărare binară, adăugând unu la fiecare pas sau aprinderea LED-urilor pe rând, prin deplasarea unei valori de 1, acest al doilea exemplu având schema logică în figura 2.4.

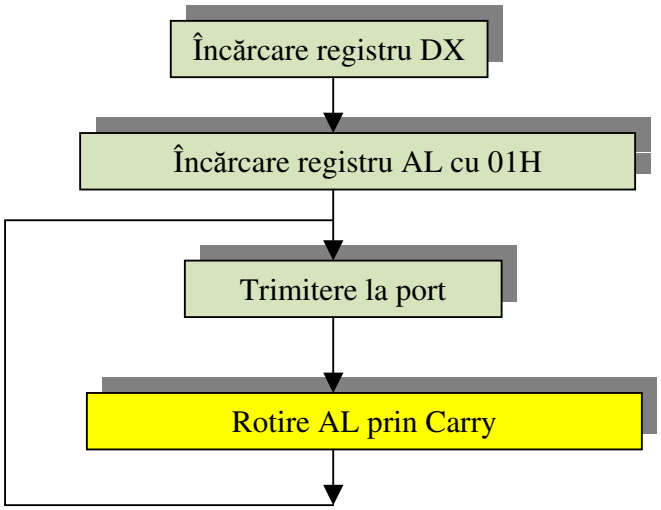
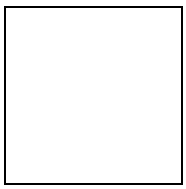


Figura 2.3. Schema logică a programului de deplasare a aprinderii unui LED

Se poate remarca că schema logică este mai simplă şi codul rezultat este mai scurt.



Dacă sintaxa instrucțiunii scrise în SID este eronată, programul va pune un semn de întrebare, ca în figura 2.4.

```
#a1000
0B9A:1000 moval,bl
?
0B9A:1000
```

Figura 2.4. Eroare de sintaxă

În acest caz se poate rescrie instrucțiunea corect, fără a ieşi şi a reintra în modul de scriere program.

2.3.Rularea programului cu viteza procesorului

La sfârşitul lucrării programul se rulează în buclă cu **Gadresa** şi studenții văd că toate LED-urile sunt aprinse, ceva mai slab decât la rularea pas cu pas. Studenții primesc tema de a găsi o explicație. Primul student care găseşte explicația este bonificat. Dacă niciun student nu găseşte explicația corectă atunci se propune ca discuția să fie amânată pâna la următoarea lucrare.

Rezumat

Această şedinţă de laborator este prima şedinţă în care studenţii lucrează efectiv. În prima etapă scriu şi rulează un program de aprindere LED-uri, apoi stingere, apoi concep individual un program care crează o secvenţă de aprindere. Rularea programelor se face pas cu pas şi se observă modificările în registrele procesorului. La sfârşit programul se rulează la viteza procesorului.

Bibliografie

1. Petre Ogrutan, Carmen Gerigan, *Memorii, interfete si periferice*, Indrumar de laborator, Reprografia Universitatii 1998, online la: http://vega.unitbv.ro/~ogrutan/lab/index.html
2. P. A. Carter, *PC AssemblyLanguage, 2003* http://pdos.csail.mit.edu/6.828/2012/readings/pcasm-book.pdf