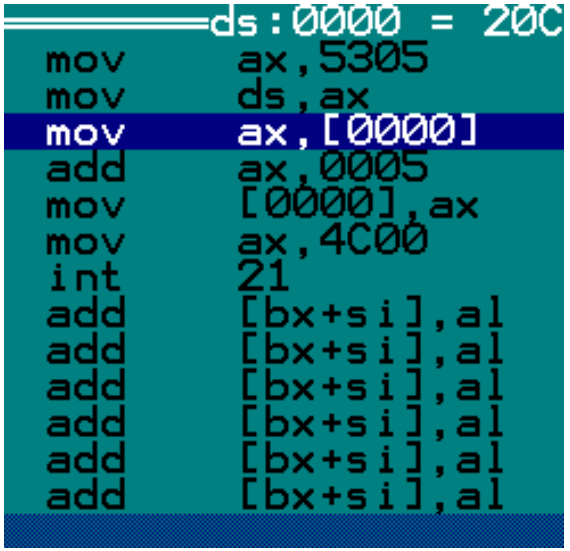
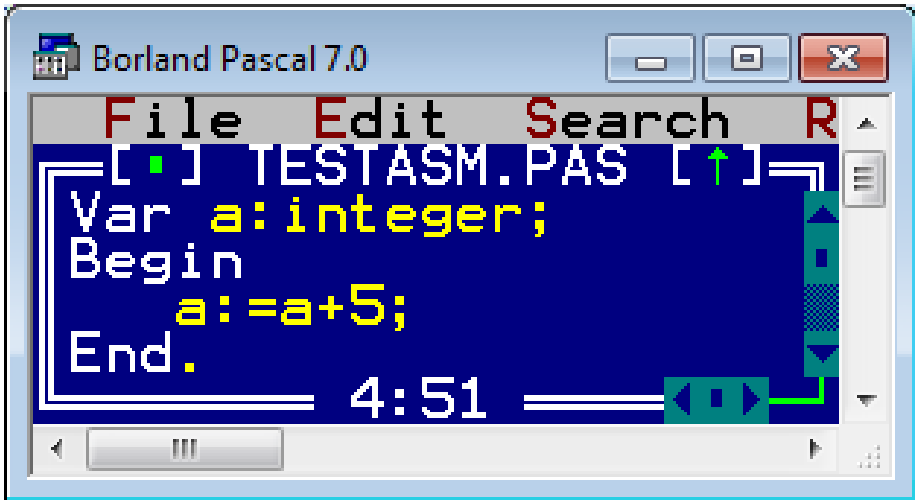
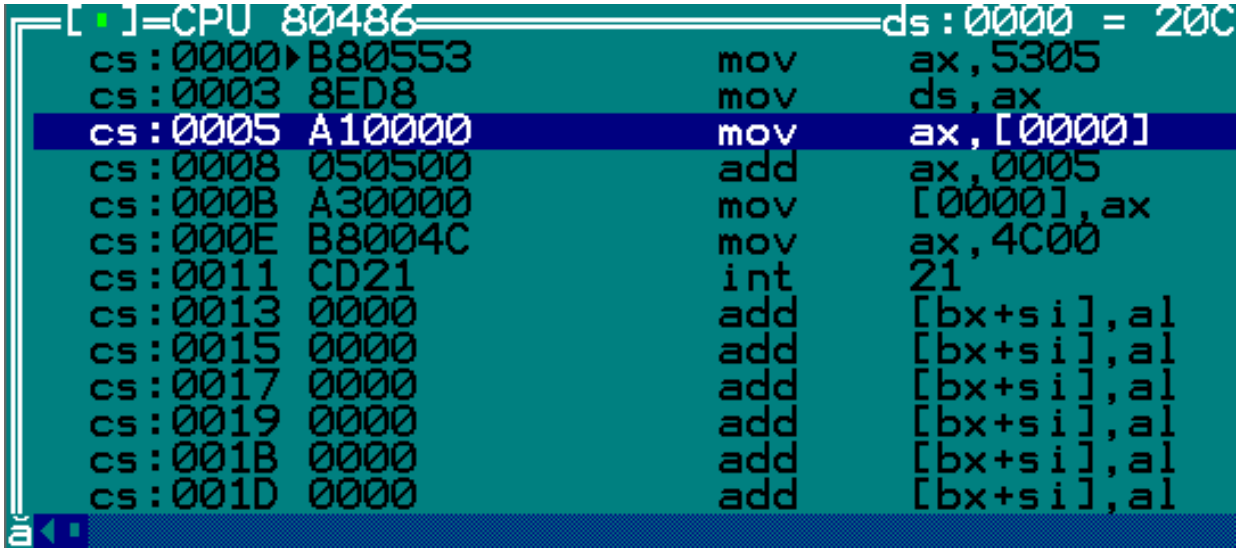
Aceasta parte (prezentare) nu este pentru examen

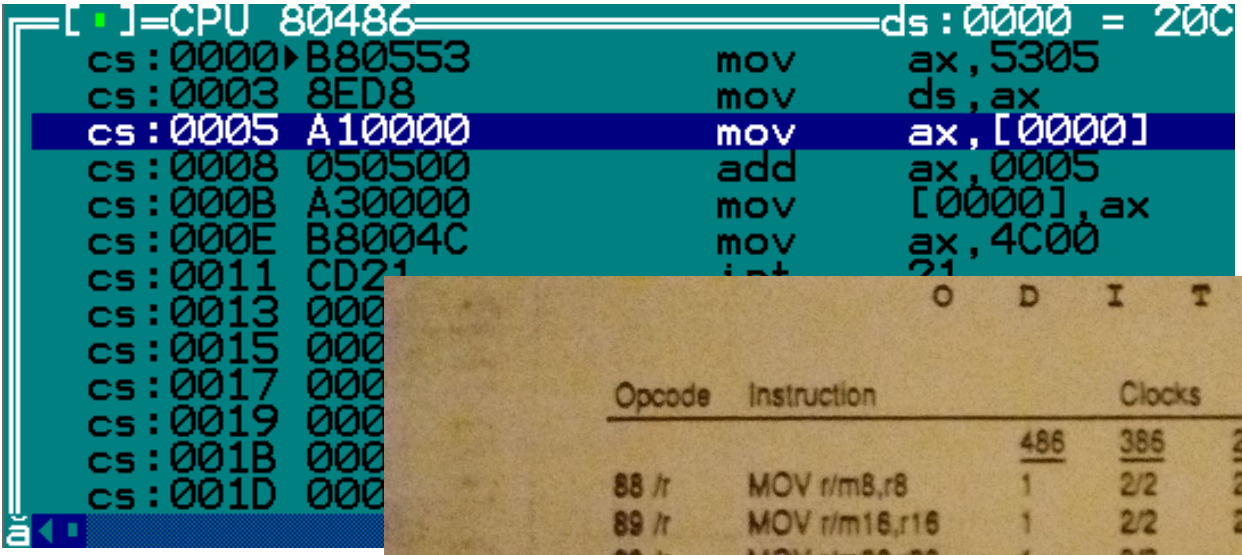
Un exemplu minimal de program in limbaj de asamblare

```
ASSUME cs: code, ds:data
  ;spunem asamblorului care sunt segmentele folosite de noi

data SEGMENT
  ;data - segmentul de date in care vom defini variabilele
data ENDS

code SEGMENT
            ;code - numele segmentului de cod

start:
  mov ax,data       ;adresa segmentului de date se copiaza in ax
  mov ds,ax         ;continutul lui ax se copiaza in ds
   ; .....
   ;aici avem instructiunile programului nostru
   ;.....
  mov ax,4C00h
  int 21h           ;finalul executiei programului

code ENDS
END start
```



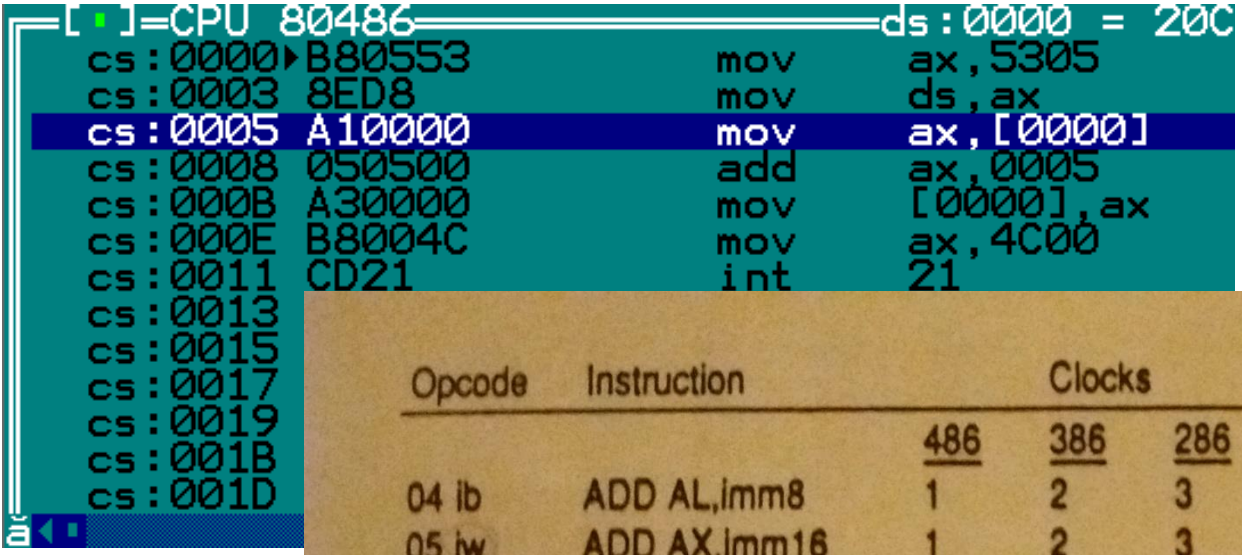

Borland Pascal 7.0
File Edit Search R
[ ] TESTASM.PAS [ ]
Var a:integer;
Begin
a:=a+5;
End.
4:51


[ ]=CPU 80486 ds:0000 = 20C
cs:0000 B80553 mov ax,5305
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 050500 add ax,0005
cs:000B A30000 mov [0000],ax
cs:000E B8004C mov ax,4C00
cs:0011 CD21 int 21
cs:0013 0000 add [bx+si],al
cs:0015 0000 add [bx+si],al
cs:0017 0000 add [bx+si],al
cs:0019 0000 add [bx+si],al
cs:001B 0000 add [bx+si],al
cs:001D 0000 add [bx+si],al



O D I T S Z A P C

Opcode	Instruction	Clocks				Description
		486	386	286	86	
88 /r	MOV r/m8,r8	1	2/2	2/3	2/9+EA	Move byte register into r/m byte
89 /r	MOV r/m16,r16	1	2/2	2/3	2/9+EA	Move word register into r/m word
89 /r	MOV r/m32,r32	1	2/2			Move dword register to r/m dword
8A /r	MOV r8,r/m8	1	2/4	2/5	2/8+EA	Move r/m byte into byte register
8B /r	MOV r16,r/m16	1	2/4	2/5	2/8+EA	Move r/m word into word register
8B /r	MOV r32,r/m32	1	2/4			Move r/m dword into dword register
8C /r	MOV r/m16,Sreg	3/3	2/2	2/3	2/9+EA	Move segment register to r/m regist
[illegible] /r	MOV Sreg,r/m16	3/9	2/5,pm=1/19	2/5,pm=17/19	2/8+EA	Move r/m word to segment register
A0	MOV AL,moffs8	1	4	5	10	Move byte at (seg:offset) to AX
A1	MOV AX,moffs16	1	4	5	10	Move word at (seg:offset) to AX
A1	MOV EAX,moffs32	1	4			Move dword at (seg:offset) to EAX
A2	MOV moffs8,AL	1	4	3	10	Move AL to (seg:offset)
A3	MOV moffs16,AX	1	2	3	10	Move AX to (seg:offset)
A3	MOV moffs32,EAX	1	2			Move EAX to (seg:offset)
B0+ rb	MOV reg8,imm8	1	2	2	4	Move immediate byte to register
B8+ rw	MOV reg16,imm16	1	2	2	4	Move immediate word to register
B8+rd	MOV reg32,imm32	1	2			Move immediate dword to register
C6	MOV r/m8,imm8	1	2/2	2/3	4/10+EA	Move immediate byte to r/m byte
C7	MOV r/m16,imm16	1	2/2	2/3	4/10+EA	Move immediate word to r/m word
C7	MOV r/m32,imm32	1	2/2			Move immediate dword to r/m dwo



Opcode	Instruction	Clocks				Description
		486	386	286	86	
04 ib	ADD AL,imm8	1	2	3	4	Add immediate byte to AL
05 iw	ADD AX,imm16	1	2	3	4	Add immediate word to A
05 id	ADD EAX,imm32	1	2			Add immediate dword to E
80 /0 ib	ADD r/m8,imm8	1/3	2/7	3/7	4/17+EA	Add immediate byte to r/m
81 /0 iw	ADD r/m16,imm16	1/3	2/7	3/7	4/17+EA	Add immediate word to r/
81 /0 id	ADD r/m32,imm32	1/3	2/7			Add immediate dword to
83 /0 ib	ADD r/m16,imm8	1/3	2/7	3/7	4/17+EA	Add sign-extended immed to r/m word
83 /0 ib	ADD r/m32,imm8	1/3	2/7			Add sign-extended immed to r/m dword
00 /r	ADD r/m8,r8	1/3	2/7	2/7	3/16+EA	Add byte register to r/m
01 /r	ADD r/m16,r16	1/3	2/7	2/7	3/16+EA	Add word register to r/m
01 /r	ADD r/m32,r32	1/3	2/7			Add dword register to r/
02 /r	ADD r8,r/m8	1/2	2/6	2/7	3/9+EA	Add r/m byte to byte reg
03 /r	ADD r16,r/m16	1/2	2/6	2/7	3/9+EA	Add r/m word to word re
03 /r	ADD r32,r/m32	1/2	2/6			Add r/m dword to dword

ADD performs an integer addition of the two operands (DEST and