**Laborator 4**

**Operații pe biţi**

**AND**

***Sintaxă:***

and <regd>, <regs>; <regd> ← <regd> AND <regs>

and <reg>, <mem>; <reg> ← <reg> AND <mem>

and <mem>, <reg>; <mem> ← <mem> AND <reg>

and <reg>, <con>; <reg> ← <reg> AND <con>

and <mem>, <con>; <mem> ← <mem> AND <con>

***Semantică:***

* Execută operația logică AND asupra operanzilor, punând rezultatul în primul operand.

***Exemplu:***

and EAX, 0Fh; EAX ← EAX AND 0Fh, are loc zerorizarea tuturor biţilor din EAX cu excepția ultimilor patru

**OR**

***Sintaxă:***

or <regd>, <regs>; <regd> ← <regd> OR <regs>

or <reg>, <mem>; <reg> ← <reg> OR <mem>

or <mem>, <reg>; <mem> ← <mem> OR <reg>

or <reg>, <con>; <reg> ← <reg> OR <con>

or <mem>, <con>; <mem> ← <mem> OR <con>

***Semantică:***

* Execută operația logică OR asupra operanzilor, punând rezultatul în primul operand.

***Exemplu:***

or EAX, 0Fh; EAX ← EAX OR 0Fh, ultimii 4 biți din EAX vor fi setați la 1, restul rămânând neschimbați

**XOR**

***Sintaxă:***

xor <regd>, <regs>; <regd> ← <regd> XOR <regs>

xor <reg>, <mem>; <reg> ← <reg> XOR <mem>

xor <mem>, <reg>; <mem> ← <mem> XOR <reg>

xor <reg>, <con>; <reg> ← <reg> XOR <con>

xor <mem>, <con>; <mem> ← <mem> XOR <con>

***Semantică:***

* Execută operația logică XOR asupra operanzilor, punând rezultatul în primul operand.

***Exemplu:***

xor EDX, EDX; zerorizează conținutul lui EDX

**TEST**

***Sintaxă:***

test <regd>, <regs>; <regd> AND <regs>

test <reg>, <mem>; <reg> AND <mem>

test <mem>, <reg>; <mem> AND <reg>

test <reg>, <con>; <reg> AND <con>

test <mem>, <con>; <mem> AND <con>

***Semantică:***

* Execută operația logică AND asupra operanzilor, fără să pună rezultatul în primul operand.

***Exemplu:***

TEST AL, 01h; se poate testa în acest mod dacă numărul e par sau impar

**NOT**

***Sintaxă:***

not <reg>

not <mem>

***Semantică:***

* Execută operația logică NOT asupra operandului (inversează fiecare bit).

***Exemplu:***

not BYTE PTR [var] ; se inversează fiecare bit din [var]

**SHL**

***Sintaxă:***

shl <reg>, <con8>

shl <mem>, <con8>

shl <reg>, CL

shl <mem>, CL

***Semantică:***

* Biţii stocați în destinație se deplasează număr poziții (modulo 32) spre stânga. Bitul (sau biţii) cel mai din dreapta se completează cu 0. Ultimul bit care iese în stânga se păstrează în CF.

***Exemplu:***

mov al, 00110011b

mov cl, 2

shl al, cl ; → al = 11001100b, CF = 0

**SHR**

***Sintaxă:***

shr <reg>, <con8>

shr <mem>, <con8>

shr <reg>, CL

shr <mem>, CL

***Semantică:***

* Biţii stocați în destinație se deplasează număr poziții (modulo 32) spre dreapta. Bitul (sau biţii) cel mai din stânga se completează cu 0. Ultimul bit care iese în dreapta se păstrează în CF.

***Exemplu:***

mov al, 01011110b

mov cl, 2

shr al, cl ; → al = 00010111b, CF = 1

**SAL**

***Sintaxă:***

sal <reg>, <con8>

sal <mem>, <con8>

sal <reg>, CL

sal <mem>, CL

***Semantică:***

* Biţii stocați în destinație se deplasează număr poziții (modulo 32) spre stânga. Bitul (sau biţii) cel mai din dreapta se completează cu 0. Ultimul bit care iese în stânga se păstrează în CF.
* Efectul instrucțiunii SAL este identic cu cel al instrucțiunii SHL

***Exemplu:***

mov al, 00110011b

mov cl, 2

sal al, cl ; → al = 11001100b, CF = 0

**SAR**

***Sintaxă:***

sar <reg>, <con8>

sar <mem>, <con8>

sar <reg>, CL

sar <mem>, CL

***Semantică:***

* Biţii stocați în destinație se deplasează număr poziții (modulo 32) spre dreapta. Bitul (sau biţii) cel mai din stânga se completează cu bitul de semn (bitul cel mai din stânga înainte de shiftare). Ultimul bit care iese în dreapta se păstrează în CF.

***Exemplu:***

mov al, 11011110b

mov cl, 2

sar al, cl ; → al = 11110111b, CF = 1

**ROL**

***Sintaxă:***

rol <reg>, <con8>

rol <mem>, <con8>

rol <reg>, CL

rol <mem>, CL

***Semantică:***

* Biţii stocați în destinație se rotesc număr poziții (modulo 32) spre stânga. Odată un bit ieșit în stânga el se adaugă automat în partea dreaptă a destinației. Ultimul bit rotit se păstrează în CF.

***Exemplu:***

mov al, 00110011b

mov cl, 2

rol al, cl ; → al = 11001100b, CF = 0

**ROR**

***Sintaxă:***

ror <reg>, <con8>

ror <mem>, <con8>

ror <reg>, CL

ror <mem>, CL

***Semantică:***

* Biţii stocați în destinație se rotesc număr poziții (modulo 32) spre dreapta. Odată un bit ieșit în dreapta el se adaugă automat în partea stângă a destinației. Ultimul bit rotit se păstrează în CF.

***Exemplu:***

mov al, 00111110b

mov cl, 2

ror al, cl ; → al = 10001111b, CF = 1

**RCL**

***Sintaxă:***

rcl <reg>, <con8>

rcl <mem>, <con8>

rcl <reg>, CL

rcl <mem>, CL

***Semantică:***

* Biţii stocați în destinație se rotesc număr poziții spre stânga. Odată un bit ieșit în stânga el se păstrează în CF. Valoarea anterioară din CF se adaugă automat în partea dreaptă a destinației.

***Exemplu:***

stc ; CF = 1 (set carry)

mov al, 00110011b

mov cl, 2

rcl al, cl ; → al = 11001110b, CF = 0

**RCR**

***Sintaxă:***

rcr <reg>, <con8>

rcr <mem>, <con8>

rcr <reg>, CL

rcr <mem>, CL

***Semantică:***

* Biţii stocați în destinație se rotesc număr poziții (modulo 32) spre dreapta. Odată un bit ieșit în dreapta el se păstrează în CF. Valoarea anterioară din CF se adaugă automat în partea stângă a destinației.

***Exemplu:***

stc ; CF = 1 (set carry)

mov al, 00110011b

mov cl, 2

rcr al, cl ; → al = 11001100b, CF = 1

**Operații pe biţi**

; Se dau cuvintele A si B. Se cere cuvântul C format astfel:

;- biţii 0-2 ai lui C coincid cu biţii 10-12 ai lui B

;- biţii 3-6 ai lui C au valoarea 1

;- biţii 7-10 ai lui C coincid cu biţii 1-4 ai lui A

;- biţii 11-12 ai valoarea 0

;- biţii 13-15 ai lui C coincid cu inversul biților 9-11 ai lui B

; Vom obține cuvântul C prin operații succesive de "izolare". Numim operația

; de izolare a biţilor 10-12 ai lui B, păstrarea intacta a valorii acestor

; biţi, si inițializarea cu 0 a celorlalți biţi. Operațiunea de izolare se

; realizează cu ajutorul operatorului AND între cuvântul B şi masca

; 0001110000000000. Odată biții izolați, printr-o operație de rotire se

; deplasează grupul de biți doriți in poziția dorita. Cuvântul final se

; obține prin aplicarea operatorului OR între rezultatele intermediare

; obținute in urma izolării şi rotirii.

; Observație: rangul biților se numără de la dreapta la stânga

bits 32 ;asamblare si compilare pentru arhitectura de 32 biţi

; definim punctul de intrare in programul principal

global start

extern exit ; indicăm asamblorului că exit există, chiar daca noi nu o vom defini

import exit msvcrt.dll; exit este o funcție care încheie procesul, este definita in msvcrt.dll

; msvcrt.dll conține exit, printf şi toate celelalte funcții C-runtime importante

segment data use32 class=data ; segmentul de date în care se vor defini variabilele

a dw 0111011101010111b

b dw 1001101110111110b

c dw 0

segment code use32 class=code ; segmentul de cod

start:

mov bx, 0 ; în registrul BX vom calcula rezultatul

mov ax, [b] ; izolăm biții 10-12 ai lui b

and ax, 0001110000000000b

mov cl, 10

ror ax, cl ; rotim 10 poziții spre dreapta

or bx, ax ; punem biţii in rezultat

or bx, 0000000001111000b ; facem biţi 3-6 din rezultat să aibă valoarea 1

mov ax, [a] ; izolăm biții 1-4 ai lui a

and ax, 0000000000011110b

mov cl, 6

rol ax, cl ; rotim 6 poziții spre stânga

or bx, ax ; punem biții în rezultat

and bx, 1110011111111111b ; facem biţi 11-12 din rezultat să aibă valoarea 0

mov ax, [b]

not ax ; inversăm valoarea lui b

and ax, 0000111000000000b ; izolăm biţi 9-11 ai lui b

mov cl, 4

rol ax, cl ; deplasăm biții 4 poziții spre stânga

or bx, ax ; punem biţii în rezultat

mov [c], bx ; punem valoarea din registru în variabila rezultat

push dword 0 ;se pune pe stivă codul de retur al funcției exit

call [exit] ;apelul funcției sistem exit pentru terminarea execuției programului

**Laborator 4 - Probleme propuse**

**Probleme propuse**

1. Se dau cuvintele A si B. Să se obţină dublucuvântul C:
   * biţii 0-4 ai lui C coincid cu biţii 11-15 ai lui A
   * biţii 5-11 ai lui C au valoarea 1
   * biţii 12-15 ai lui C coincid cu biţii 8-11 ai lui B
   * biţii 16-31 ai lui C coincid cu biţii lui A
2. Se dau cuvintele A si B. Se cere dublucuvântul C:
   * biţii 0-3 ai lui C coincid cu biţii 5-8 ai lui B
   * biţii 4-8 ai lui C coincid cu biţii 0-4 ai lui A
   * biţii 9-15 ai lui C coincid cu biţii 6-12 ai lui A
   * biţii 16-31 ai lui C coincid cu biţii lui B
3. Se dau cuvintele A si B. Să se obțină dublucuvântul C:
   * biţii 0-2 ai lui C coincid cu biţii 12-14 ai lui A
   * biţii 3-8 ai lui C coincid cu biţii 0-5 ai lui B
   * biţii 9-15 ai lui C coincid cu biţii 3-9 ai lui A
   * biţii 16-31 ai lui C coincid cu biţii lui A
4. Se da octetul A. Să se obţină numărul întreg n reprezentat de biţii 2-4 ai lui A. Să se obţină apoi în B octetul rezultat prin rotirea spre dreapta a lui A cu n poziții. Să se obţină dublucuvântul C:
   * biţii 8-15 ai lui C sunt 0
   * biţii 16-23 ai lui C coincid cu biţii lui B
   * biţii 24-31 ai lui C coincid cu biţii lui A
   * biţii 0-7 ai lui C sunt 1
5. Se dau octeţii A si B. Să se obţină dublucuvântul C:
   * biţii 16-31 ai lui C sunt 1
   * biţii 0-3 ai lui C coincid cu biţii 3-6 ai lui B
   * biţii 4-7 ai lui C au valoarea 0
   * biţii 8-10 ai lui C au valoarea 110
   * biţii 11-15 ai lui C coincid cu biţii 0-4 ai lui A
6. Se da cuvântul A. Să se obţină numărul întreg n reprezentat de biţii 0-2 ai lui A. Să se obţină apoi in B cuvântul rezultat prin rotirea spre dreapta (fără carry) a lui A cu n poziții. Să se obţină dublucuvântul C:
   * biţii 8-15 ai lui C sunt 0
   * biţii 16-23 ai lui C coincid cu biţii lui 2-9 ai lui B
   * biţii 24-31 ai lui C coincid cu biţii lui 7-14 ai lui A
   * biţii 0-7 ai lui C sunt 1
7. Se dau doua cuvinte A si B. Să se obţină dublucuvântul C:
   * biţii 0-4 ai lui C au valoarea 1
   * biţii 5-11 ai lui C coincid cu biţii 0-6 ai lui A
   * biţii 16-31 ai lui C au valoarea 0000000001100101b
   * biţii 12-15 ai lui C coincid cu biţii 8-11 ai lui B
8. Se dau doua cuvinte A si B. Să se obţină un octet C care are:
   * pe biţii 0-5, biţii 5-10 ai cuvântului A
   * pe biţii 6-7 biţii 1-2 ai cuvântului B.

Să se obţină dublucuvântul D care are :

* + pe biţii 8-15, biţii lui C
  + pe biţii 0-7, biţii 8-15 din B
  + pe biţii 24-31, biţii 0-7 din A
  + iar pe biţii 16-23, biţii 8-15 din A.

1. Se de cuvântul A si octetul B. Să se obţină dublucuvântul C astfel:
   * biţii 0-3 ai lui C coincid cu biţii 6-9 ai lui A
   * biţii 4-5 ai lui C au valoarea 1
   * biţii 6-7 ai lui C coincid cu biţii 1-2 ai lui B
   * biţii 8-23 ai lui C coincid cu biţii lui A
   * biţii 24-31 ai lui C coincid cu biţii lui B
2. Sa se înlocuiască biţii 0-3 ai octetului B cu biţii 8-11 ai cuvântului A.
3. Se dau un octet A si un cuvânt B. Să se obţină un octet C care are pe biţii 0-3 biţii 2-5 ai lui A, iar pe biţii 4-7 biţii 6-9 ai lui B.
4. Se dau doua cuvinte A si B. Să se obţină dublucuvântul C:
   * biţii 0-6 ai lui C au valoarea 0
   * biţii 7-9 ai lui C coincid cu biţii 0-2 ai lui A
   * biţii 10-15 ai lui C coincid cu biţii 8-13 ai lui B
   * biţii 16-31 ai lui C au valoarea 1
5. Dându-se 4 octeţi, Să se obţină in AX suma numerelor întregi reprezentate de biţii 4-6 ai celor 4 octeţi.
6. Se da dublucuvântul A. Să se obţină numărul întreg n reprezentat de biţii 14-17 ai lui A. Să se obţină apoi in B dublucuvântul rezultat prin rotirea spre stânga a lui A cu n poziții. Să se obţină apoi octet C astfel:
   * biţii 0-5 ai lui C coincid cu biţii 1-6 ai lui B
   * biţii 6-7 ai lui C coincid cu biţii 17-18 ai lui B
7. Se dau cuvintele A si B. Se cere dublucuvântul C:
   * biţii 0-2 ai lui C au valoarea 0
   * biţii 3-5 ai lui C au valoarea 1
   * biţii 6-9 ai lui C coincid cu biţii 11-14 ai lui A
   * biţii 10-15 ai lui C coincid cu biţii 1-6 ai lui B
   * biţii 16-31 ai lui C au valoarea 1
8. Se dau octetul A si cuvântul B. Să se obţină dublucuvântul C:
   * biţii 0-7 ai lui C au valoarea 1
   * biţii 8-11 ai lui C coincid cu biţii 4-7 ai lui A
   * biţii 12-19 coincid cu biţii 2-9 ai lui B
   * biţii 20-23 coincid cu biţii 0-3 ai lui A
9. Se dau cuvântul A si octetul B. Să se obţină dublucuvatul C:
   * biţii 0-3 ai lui C au valoarea 1
   * biţii 4-7 ai lui C coincid cu biţii 0-3 ai lui A
   * biţii 8-13 ai lui C au valoarea 0
   * biţii 14-23 ai lui C coincid cu biţii 4-13 ai lui A
   * biţii 24-29 ai lui C coincid cu biţii 2-7 ai lui B
   * biţii 30-31 au valoarea 1
10. Se da un cuvânt A. Să se obţină dublucuvântul B astfel:
    * biţii 0-3 ai lui B sunt 0;
    * biţii 4-7 ai lui B sunt biţii 8-11 ai lui A
    * biţii 8-9 si 10-11 ai lui B sunt biţii 0-1 inversați ca valoare ai lui A (deci de 2 ori) ;
    * biţii 12-15 ai lui B sunt biţi de 1
    * biţii 16-31 ai lui B sunt identici cu biţii 0-15 ai lui B.
11. Se da un cuvânt A. Să se obţină dublucuvântul B astfel:
    * biţii 28-31 ai lui B sunt 1;
    * biţii 24- 25 si 26-27 ai lui B sunt biţii 8-9 ai lui A
    * biţii 20-23 ai lui B sunt biţii 0-3 inversați ca valoare ai lui A ;
    * biţii 16-19 ai lui B sunt biţi de 0
    * biţii 0-15 ai lui B sunt identici cu biţii 16-31 ai lui B.
12. Se dau cuvintele A si B. Se cere dublucuvântul C:
    * biţii 0-5 ai lui C coincid cu biţii 3-8 ai lui A
    * biţii 6-8 ai lui C coincid cu biţii 2-4 ai lui B
    * biţii 9-15 ai lui C reprezintă biţii 6-12 ai lui A
    * biţii 16-31 ai lui C sunt 0
13. Se dau cuvintele A si B. Se cere dublucuvântul C:
    * biţii 0-3 ai lui C coincid cu biţii 5-8 ai lui B
    * biţii 4-10 ai lui C sunt invers fata de biţii 0-6 ai lui B
    * biţii 11-18 ai lui C sunt 1
    * biţii 19-31 ai lui C coincid cu biţii 3-15 ai lui B
14. Se da dublucuvântul A si cuvântul B. Sa se formeze cuvântul C:
    * biţii 0-4 ai lui C sunt invers fata de biţii 20-24 ai lui A
    * biţii 5-8 ai lui C sunt 1
    * biţii 9-12 ai lui C sunt identici cu biţii 12-15 ai lui B
    * biţii 13-15 ai lui C sunt biţii 7-9 ai lui A
15. Se da octetul A si cuvântul B. Sa se formeze dublucuvântul C:
    * biţii 24-31 ai lui C sunt biţii lui A
    * biţii 16-23 ai lui C sunt inversul biţilor din octetul cel mai puțin semnificativ al lui B
    * biţii 10-15 ai lui C sunt 1
    * biţii 2-9 ai lui C sunt biţii din octetul cel mai semnificativ al lui B
    * biţii 0-1 se completează cu valoarea bitului de semn al lui A
16. Se da dublucuvântul M. Să se obţină dublucuvântul MNew astfel:
    * biţii 0-3 a lui MNew sunt identici cu biţii 5-8 a lui M
    * biţii 4-7 a lui MNew au valoarea 1
    * biţii 27-31 a lui MNew au valoarea 0
    * biţii 8-26 din MNew sunt identici cu biţii 8-26 a lui M.
17. Se dau 2 dublucuvinte M si N. Să se obţină dublucuvântul P astfel:
    * biţii 0-6 din P coincid cu biţii 10-16 a lui M
    * biţii 7-20 din P coincid cu biţii obținuți 7-20 in urma aplicării M AND N.
    * biţii 21-31 din P coincid cu biţii 1-11 a lui N.
18. Se dau 2 dublucuvinte R si T. Să se obţină dublucuvântul Q astfel:
    * biţii 0-6 din Q coincid cu biţii 10-16 a lui T
    * biţii 7-24 din Q coincid cu biţii obținuți 7-24 in urma aplicării R XOR T.
    * biţii 25-31 din Q coincid cu biţii 5-11 a lui R.
19. Se da quadwordul A. Să se obţină numărul întreg N reprezentat de biţii 35-37 ai lui A. Să se obţină apoi in B dublucuvântul rezultat prin rotirea spre dreapta a dublucuvântului inferior al lui A cu N poziții. Să se obţină octetul C astfel:
    * biţii 0-3 ai lui C sunt biţii 8-11 ai lui B
    * biţii 4-7 ai lui C sunt biţii 16-19 ai lui B
20. Se da quadwordul A. Să se obţină numărul întreg N reprezentat de biţii 17-19 ai lui A. Să se obţină apoi in B dublucuvântul rezultat prin rotirea spre stânga a dublucuvântului superior al lui A cu N poziții. Să se obţină octetul C astfel:
    * biţii 0-2 ai lui C sunt biţii 9-11 ai lui B
    * biţii 3-7 ai lui C sunt biţii 20-24 ai lui B
21. Se dau dublucuvintele A si B. Sa se formeze qwordul C in felul următor:
    * biţii 0-7 ai lui C sunt biţii 21-28 ai lui A
    * biţii 8-15 ai lui C sunt biţii 23-30 ai lui B complementaţi
    * biţii 16-21 ai lui C sunt 101010
    * biţii 22-31 ai lui C au valoarea 0
    * biţii 32-42 ai lui C sunt biţii 21-31 ai lui B
    * biţii 43-55 ai lui C sunt biţii 1-13 ai lui A
    * biţii 56-63 ai lui C sunt biţii 24-31 ai rezultatului A XOR 0ABh
22. Se da cuvântul A. Sa se formeze doublewordul B in felul următor:
    * biţii 0-3 ai lui B sunt biţii 1-4 ai rezultatului A XOR 0Ah
    * biţii 4-11 ai lui B sunt biţii 7-14 ai lui A
    * biţii 12-19 ai lui B au valoarea 0
    * biţii 20-25 ai lui B au valoarea 1
    * biţii 26-31 ai lui C sunt biţii 3-8 ai lui A complementați
23. Se dau cuvintele A, B si C. Sa se formeze cuvântul D ca suma a numerelor reprezentate de:
    * biţii de pe poziţiile 1-5 ai lui A
    * biţii de pe poziţiile 6-10 ai lui B
    * biţii de pe poziţiile 11-15 ai lui C
24. Se dau cuvintele A, B si C. Să se obţină octetul D ca sumă a numerelor reprezentate de:
    * biţii de pe poziţiile 0-4 ai lui A
    * biţii de pe poziţiile 5-9 ai lui B

Octetul E este numărul reprezentat de biţii 10-14 ai lui C. Să se obţină octetul F ca rezultatul scăderii D-E.