Trabalho Prático Nº 3

Cálculo de expressões aritméticas e lógicas

Instruções do 8086:ADD, ADC, INC, SUB, SBB, DEC, MUL, IMUL, DIV, IDIV, NEG, CBW, CWD, SHL, SAL, SHR, SAR, ROR, ROL, RCR, RCL, AND, OR, XOR, NOT, JMP, Jcc, TEST,

1) Considere as seguintes instruções de atribuição em Pascal.

```
x:=y+z;              x:=y+z+w;                  x:=y-z;
x:=y-z-w;            x:=x-(y+z);                x:=x+y*z
x:=-y;               x:=(x+y)*(y+w);            x:= y div x;
x:= w mod x;         x:= ((y+x)*(x-w)) div 10;  x:= (x*y) mod 15;
y:=y+1;              x:=y-1;                    x:=(y-x)-1;
```

Codifique em assembly 8086 cada uma das instruções anteriores, considerando:

a) x,y,z,w variáveis do tipo byte;

b) x,y,z,w variáveis do tipo integer.

2) Considere as seguintes declarações em Pascal.

```
Var x,y:Byte; sx,sy:Shortint; xx:Word; sxx,syy:Integer;
```

Codifique em assembly 8086 cada uma das instruções seguintes:

```
xx:=xx+y;                    sxx:=syy+sy;
xx:=x*y;                     sxx:=sx*sy;
x:=xx div x;                 sx:=sxx mod sx;
```

3) Considere as seguintes declarações em Pascal.

```
Var xx,yy:Longint;
```

Codifique em assembly 8086 cada uma das instruções seguintes:

```
xx:=xx+yy;        xx:=yy-xx;
```

4) Utilizando os operadores lógicos AND, OR, XOR e NOT, codifique em assembly 8086 as seguintes operações:

complementar os bits b_7, b_6, b_5, b_4 da varíavel A;

colocar a 1 os bits b_5, b_2, b_0 da variável A;

colocar a 0 os bits b_6, b_3, b_1 da variável A;

complementar todos os bits da variável A;

considerando a variável A do tipo byte.

5) Considere as seguintes declarações em Pascal.

```
Var A,B,C,D,E:Boolean;
```

Codifique em assembly 8086 cada uma das instruções seguintes:

```
A:= B and C;                 A:= B and C and D ;
A:= B or C;                  A:= B or C or D ;
A:=(B or C) and (E or D);    A:=(B and C) or (E and D);
A:=(Not A) or (C and D);     A:=(A xor C) and ( A and (not B));
```

6) Considere as seguintes instruções:

```
A:= x>=Y;
A:= x=Y;
A:= x<Y;
A:= (x-y)<>10;
A:= ((x-y)=z) and (w>=y);
A:= (x>=z) or not (w>=y);
A:= ((x<=z) and (w>y)) or (y=z);
A:= ((x=z) or (w>=y)) and (y<>z);
```

Codifique em assembly 8086 cada uma das instruções anteriores, considerando:
a) x,y,z,w variáveis do tipo byte e A do tipo Boolean;
b) x,y,z,w variáveis do tipo integer e A do tipo Boolean.

7) Considere as seguintes declarações:

```
Var x:byte; sx:shortint; xx:word; sxx:integer; a,b,c:boolean;
```

Codifique em assembly 8086 as seguintes instruções:

```
A:=x<>xx;
A:=xx>=x;
A:=sxx<=sx;
A:=(x=xx) and not b;
A:= a and (xx>=x);
B:=(sx>=sxx) or ( a and c);
```

8) Considerando:

```
Var a,b:word; x,y,z:integer; lxx:longint;
```

Não utilizando as instruções de multiplicação e divisão (MUL, IMUL, DIV, IDIV) codifique as intruções:

```
y:= 10*x;          y:= 7*x;           z:=12*x;
y:= x div 2;       b:=a div 8;        x:=x div 16;
b:= a mod 2;       b:= a mod 16;
lxx:=lxx * 2;      lxx:=lxx*4;        lxx:=lxx div 2;
```