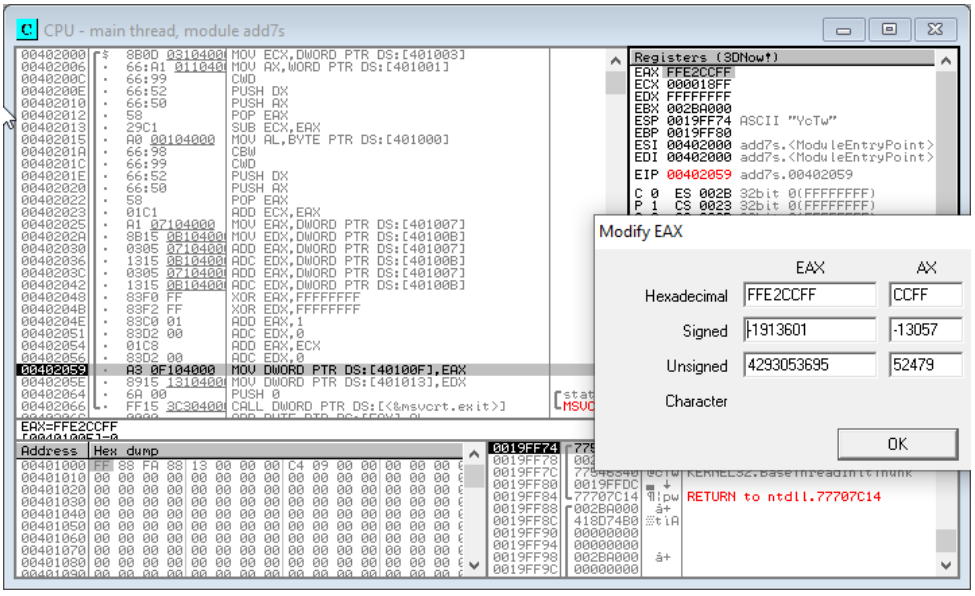
Neacșu-Miclea Liviu-Ștefan
Grupa 215

1. Adunări, Scăderi (1) – Ex. 7 – Signed

c-(d+d+d)+(a-b)

```
a DB       -1
b DW    -1400
c DD     5000
d DQ   640000
x RESQ      1
```

```
mov  ECX ,       [ c ] ;   ECX = c              =     5000
mov   AX ,       [ b ] ;    AX = b              =    -1400
cwd                    ; DX:AX = b (signed)     =    -1400
push DX
push AX
pop EAX                ;   EAX = b (signed)     =    -1400
sub  ECX ,        EAX  ;   ECX = c - b          =     6400
mov   AL ,       [ a ] ;    AL = a              =       -1
cbw                    ;    AX = a (signed)     =       -1
cwd                    ; DX:AX = a (signed)     =       -1
push DX
push AX
pop EAX                ;   EAX = a (signed)     =       -1
add  ECX ,        EAX  ; ECX = c - b + a        =     6399
mov  EAX ,       [ d ] ;
mov  EDX ,       [d+4] ; EDX:EAX = d           =   640000
add  EAX , DWORD [ d ]
adc  EDX , DWORD [d+4] ; EDX:EAX = d+d         = 1280000
add  EAX , DWORD [ d ]
adc  EDX , DWORD [d+4] ; EDX:EAX = 2*d+d       = 1920000
xor  EAX , 0xFFFFFFFF  ; EDX:EAX = -EDX:EAX /
xor  EDX , 0xFFFFFFFF  ; -x = (~x) + 1
add  EAX , 1
adc  EDX , 0           ; EDX:EAX = -(3*d)      =-1920000
add  EAX ,  ECX
adc  EDX ,   0         ; EDX:EDX = ECX - 3*d   =-1913601
mov [ x ], EAX
mov [x+4], EDX         ; store  result
```



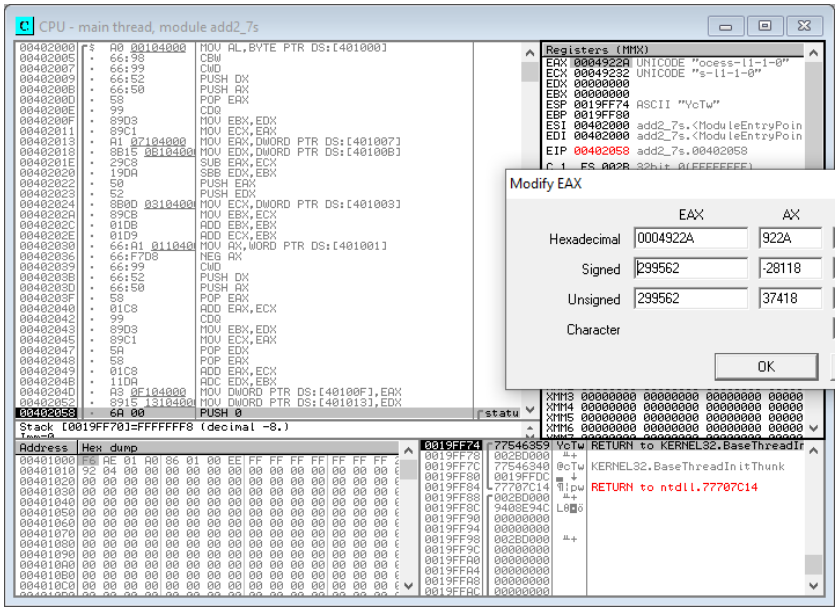
Neacșu-Miclea Liviu-Ștefan
Grupa 215

2. Adunări, Scăderi (2) – Ex. 7 – Signed

(c+c+c)-b+(d-a)

```
a DB     -10
b DW     430
c DD 100000
d DQ     -18
x RESQ 1
```

```
mov   AL , [ a ] ;     AL = a = -10
cbw              ;     AX = a = -10
cwd              ;  DX:AX = a = -10
push  DX
push  AX
pop   EAX        ;     EAX = a = -10
cdq              ; EDX:EAX = a = -10
mov   EBX ,  EDX
mov   ECX ,  EAX  ; EBX:ECX = a = -10
mov   EAX , [ d ]
mov   EDX , [d+4] ; EDX:EAX = d = -18
sub   EAX , ECX
sbb   EDX , EBX   ; EDX:EAX = d - a = -8
push EAX
push EDX          ; backup (d-a)
mov   ECX , [ c ] ;     ECX = c      = 100000
mov   EBX ,  ECX
add   EBX ,  EBX  ;     EBX = c+c    = 200000
add   ECX ,  EBX  ;     ECX = c+c+c  = 300000
mov    AX , [ b ] ;      AX =  b     =  430
neg    AX         ;      AX = -b     = -430
cwd               ;   DX:AX = -b     = -430
push  DX
push  AX
pop   EAX         ;     EAX = -b       = -430
add   EAX ,  ECX  ;     EAX = 3*c - b =  299570
cdq               ; EDX:EAX = 3*c - b =  299570
mov   EBX ,  EDX
mov   ECX ,  EAX  ; EBX:ECX = EDX:EAX =  299570
pop   EDX
pop   EAX         ; restore (d-a)
add   EAX ,  ECX
adc   EDX ,  EBX  ; EAX:EDX = (d-a)+(3*c-b) = 299562
mov [ x ],  EAX
mov [x+4],  EDX
```

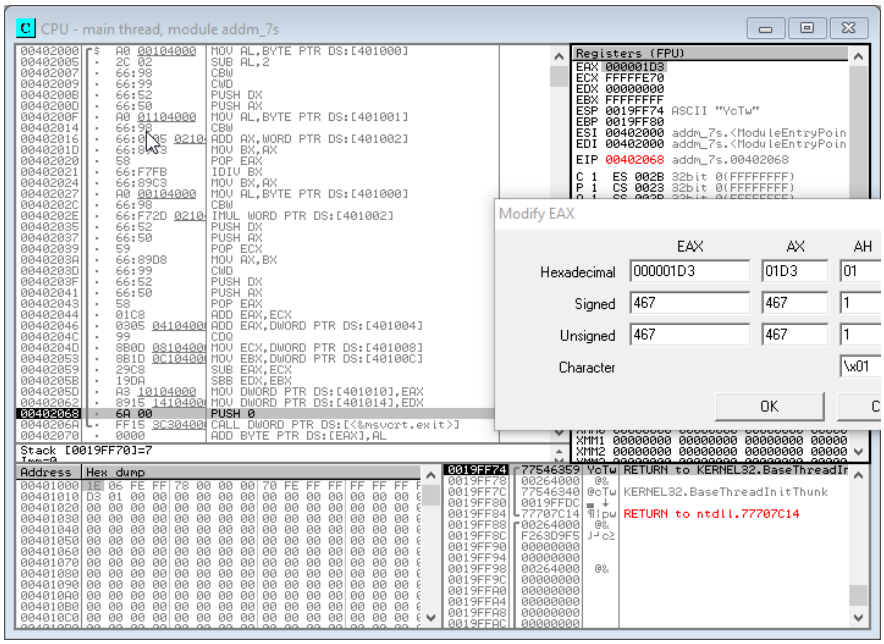


3. Înmulțiri, Împărțiri – Ex. 7 – Signed

(a-2)/(b+c)+a*c+e-x

```
a DB 30
b DB  6
c DW -2
e DD 120
x DQ -400
r RESQ 1
```

```
mov    AL , [ a ] ;    AL = a = 30
sub    AL ,   2   ;    AL = a-2 = 28
cbw               ;    AX = a-2 = 28
cwd               ; DX:AX = a-2 = 28
push   DX
push   AX         ; backup; pop when needed
mov    AL , [ b ] ;   AL = b   = 6
cbw               ;   AX = b   = 6
add    AX , [ c ] ;   AX = b+c = 4
mov    BX ,   AX  ;   BX = b+c = 4
pop   EAX         ; retrieve EAX = (a-2) = 28
idiv   BX         ; EAX/BX = AX(7) r DX(ignore)
mov    BX ,   AX  ; BX = (a-2)/(b+c) = 7
mov    AL , [ a ] ; AL = a = 30
cbw               ; AX = a = 30
imul WORD   [ c ] ; DX:AX = a*c = -60
push   DX
push   AX
pop   ECX         ; ECX = c * a = -60
mov    AX ,   BX  ; AX = (a-2)/(b+c) = 7
cwd               ; DX:AX = 7
push   DX
push   AX
pop   EAX         ; EAX = 7
add   EAX ,  ECX  ; EAX = 7 + a*c = -53
add   EAX , [ e ] ; EAX = -53 + e =  67
cdq               ; EDX:EAX = 67
mov   ECX , [ x ] ;
mov   EBX , [x+4] ; EBX:ECX = x = -400
sub   EAX ,  ECX
sbb   EDX ,  EBX  ; EDX:EAX = 67 - x = 467
mov  [ r ],  EAX
mov  [r+4],  EDX  ; r = EDX:EAX = 467
```



Neacșu-Miclea Liviu-Ștefan
Grupa 215

4. Adunări, Scăderi (2) – Ex. 7 – Unsigned

(c+c+c)-b+(d-a)

```
a DB 3
b DW 300
c DD 100000
d DQ 1000000
r RESQ 1
```

```
mov   AL , [ a ] ;    AL = a = 30
mov  EAX , [ c ] ; EAX = c = 100000
mov  EBX ,  EAX  ; EBX = c = 100000
add  EBX ,  EBX  ; EBX = 2*c
add  EAX ,  EBX  ; EAX = c+2*c = 300000
xor  EBX ,  EBX  ; EBX = 0
mov   BX , [ b ] ; (E)BX = b = 300
sub  EAX ,  EBX  ; EAX = 3*c - b = 299700
xor  EDX ,  EDX  ; EDX = 0
mov  ECX , [ d ]
mov  EBX , [d+4] ; EBX:ECX = d = 1000000
add  EAX ,  ECX
adc  EDX ,  EBX  ; EAX:EDX = d + (3*c-b) = 1299700
xor  EBX ,  EBX  ; EBX = 0
xor  ECX ,  ECX  ; ECX = 0
mov   CL , [ a ] ; CL(ECX) = 3
sub  EAX ,  ECX
sbb  EDX ,  EBX  ; EAX:EDX = d + (3*c-b) - a = 1299697
mov  [ r ], EAX
mov  [r+4], EDX ; r = EDX:EAX = 1299697
```