Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Ime:	***Luka***	***Broj indeksa:***	***18587***
Prezime:	***Veličković***		
LV po redu:	*II*	***Termin:***	*2*
Datum i vreme početka izrade	*18.11.2022 21:28*		

Zadatak:

Izračunati, bez transformacije, i uz optimizaciju deljenja stepenom 2, vrednost izraza
(B1+1)*([B1*]-1) / (B1-B2^2/4),
pri čemu su: B1 32b, B2 32b, oba neoznačeni podaci, a operator [X*] znači:
zadržani bitovi na parnim pozicijama podatka X, a ostali bitovi postavljeni na 1.

a) Da li ovaj problem može da se reši u zadatim okvirima? Ukoliko ne može, pod kojim uslovima bi mogao da se reši? Rešiti zadati problem u okvirima u kojima može da se reši.
Formirati primere početnih vrednosti koji demonstriraju sve osobine zadatog problema, posebno u pogledu međuprenosa, izlaznog prenosa, ostatka pri deljenju, vrednosti nekog od međurezultata jednakoj nuli, i za eventualne slučajeve kada nije moguće doći do tačnog rezultata.

   Za svaki primer početnih vrednosti:

   - navesti po čemu je karakterističan - koju osobinu demonstrira
   - izračunati rešenje i pokazati kako se dolazi do tog rešenja po koracima nacrtane šeme postupka.

   Za svaku operaciju iz izraza datog u problemu, nacrtati šemu izvođenja te operacije kada bi se izvodio instrukcijama x86-32 arhitekture procesora. Na šemi treba da se vide težine pojedinih delova operanada, redosled redosled izračunavanja, operandi i odredišta svakog međurezultata i konačnog rezultata. Šeme prikazati u redosledu u kome bi operacije trebalo izvoditi u asemblerskom programu.

b) Napisati kod na asemblerskom jeziku za sprovođenje izračunavanja po šemi postupka iz a). Na šemi postupka označiti registre koji su korišćeni u kodu. Uneti napisani kod u emulator na adresi https://carlosrafaelgn.com.br/Asm86/; primere početnih vrednosti uneti kao komplete instrukcija upisivanja vrednosti u promenljive, pri čemu su svi kompleti, osim jednog, podešeni kao komentar.
Izvršiti napisani kod u emulatoru za sve primere početnih vrednosti i ustanoviti da li program radi kako je očekivano.

   U izveštaju napisati kratak tekst o tome šta program treba da radi, da li se izvršava ili postoji greška (i gde je greška - priložiti snimak ekrana!) i da li se rezultati poklapaju sa očekivanim rezultatima iz a) za sve primere.

   Ukoliko se za neki primer rezultati ne poklapaju sa očekivanima, ustanoviti na kom mestu u kodu dolazi do odstupanja.

Rešenje:

a) Primeri vrednosti, šeme operacija

Dati zadatak nije moguće u potpunosti rešiti u zadatim okvirima. U zadatom problemu pojavljuje se izraz (B1 - B2^2/4), a kako su brojevi B_1 i B_2 neoznačeni, slučajeve u kojima je $B_1 \leq B_2^2/4$ nije moguće rešiti u zadatim okvirima. Za brojeve koji ne zadovoljavaju prethodni uslov moguće je rešiti zadati problem. Može se i bliže odrediti maksimalna vrednost koju broj B_2 može imati.
Kako se radi o 32-bitnim brojevima, maksimalna vrednost koju broj B_1 može imati je 0xFFFFFFFF. Neophodno je da izraz $B_2^2/4$ bude manji od zadatog broja. Množenjem maksimalne vrednosti broja B_1 sa 4 zaključujemo da B_2^2 može maksimalno imati vrednost 0x3FFFFFFFC. Nalaženjem korena datog broja zaključuje se da B_2 maksimalno može imati vrednost 0x1FFFF, pri čemu da bi problem bio rešiv neophodno je da B_1 ima vrednost veću od 0xFFFF0000.
Primeri početnih vrednost:

1) Ulazne vrednosti koje dovode do toga da se broj B_2 nakon kvadriranja nalazi u registru EDX:EAX, pri čemu taj broj nije deljiv brojem 4, pa se ostatak pri računanju zanemaruje.
   B_1 = 0x81CA2454
   B_2 = 0x12F43
   Rešenje: - Računamo B_2^2 i međurezultat se nalazi u registrima EDX:EAX
   B_2^2 = 0x12F43 * 0x12F43 = 0x1673FAB89
   Registar EDX nakon kvadriranja:
   0b00000000000000000000000000000001
   Registar EAX nakon kvadriranja:
   0b01100111001111111010101110001001
   Deljenje brojem 4 podrazumeva pomeranje registra 2 mesta udesno. Kako se broj B_2^2 nalazi u dva registra, neophodno je pamtiti šta se dešava sa bitovima registra koji pamti cifre najveće težine. Zbog toga je dvostruko pomeranje udesno neophodno izvršiti izvršavanjem jednostrukog pomeranja dva puta, između kojih se izlazna cifra iz registra EDX umeće na mesto cifre najveće težine registra EAX.
   Registar EDX nakon deljenja sa 2, tj. nakon pomeranja sadržaja udesno za 1 mesto:
   0b00000000000000000000000000000000,
   cifra najmanje težine datog registra (1) sačuvana u carry flagu, iz koga se prebacuje u registar ECX, čije je stanje sada 0b00000000000000000000000000000001, a nakon pomeranja sadržaja za 31 mesto ulevo 0b10000000000000000000000000000000.
   Registar EAX nakon deljena sa 2, tj. nakon pomeranja sadržaja udesno za 1 mesto:
   0b00110011100111111101010111000100,
   pri čemu je cifra najmanje težine, koja je istisnuta iz broja (1) ignorisana.
   Da bi se cifra koja je izbačena iz registra EDX, a sada se nalazi na mestu cifre najveće težine u registru ECX, umetnula na mesto cifre najveće težine registra EAX, potrebno je izvršiti OR između odgovarajućih bitova registara EAX i ECX.

Stanje registra EAX je sada 0b10110011100111111101010111000100 što odgovara heksadekadnom broju 0xB39FD5C4.
Opisani postupak se ponovi još jednom, nakon čega je stanje registra EAX 0b01011001110011111110101011100010, što odgovara broju 0x59CFEAE2.
Kako je 0x59CFEAE2 * 4 = 0x1673FAB88, zaključujemo da je ignorisani ostatak 1.
Dalje se od broja B_1 oduzima izračunati broj i dobija se 0x27FA3972, što predstavlja vrednost izraza (B_1 - B_2^2/4).
Sledeći korak je izračunavanje vrednosti izraza ([B_1*] - 1). Kako bi se izračunalo [B_1*], neophodno je izvršiti operaciju OR između odgovarajućih bitova maskom koja će bitove na neparnim pozicijama postaviti na 1, a bitove na parnim zadržati. Na primeru 4 bita $b_3b_2b_1b_0$ primećujemo da, kako bi se ovaj efekat postigao, je potrebno primeniti masku 1 0 1 0, što odgovara heksadekadnom broju 0xA. Shodno tome, za postizanje željenog efekta na 32-bitnom broju potrebno je primeniti masku 0xAAAAAAAA. U konkretnom primeru to izgleda ovako: B1 – 0b10000001110010100010010001010100
OR 0b10101010101010101010101010101010

0b10101011111010101010111011111110,
što je jednako 0xABEAAEFE, što nakon dekrementiranja postaje 0xABEAAEFD.
Sledeći korak je inkrementiranje broja B_1, što daje broj 0x81CA2455.
Na kraju se početni izraz sveo na 0x81CA2455 * 0xABEAAEFD / 0x27FA3972 čija je vrednost 0x22E23D574, a ostatak 0x65CC59.

2) Ulazne vrednosti kod kojih je jedno od međurešenja jednako 0, pri čemu je i konačno rešenje jednako 0, a pojavljuje se izlazni prenos.
B_1 = 0xFFFFFFFF
B_2 = 0x126C2
Primenom opisanog postupka za izračunavanje izraza (B1 - B2^2/4), opisanog u prvom primeru, dobija se vrednost 0x54D88AC1. Primena operatora [X*] na operand B_1 nema efekta. Dekrementiranje datog operanda daje vrednost 0xFFFFFFFE.
Inkrementiranje datog operanda, pri izračunavanju vrednosti prvog izraza daje vrednost 0b100000000, pri čemu će se u registru EAX naći samo 0b00000000, a 1 predstavlja izlazni prenos. Na osnovu šeme izračunavanja konačna vrednost će biti 0, jer se izlazni prenos ne uračunava. Tačno rešenje za date vrednosti je 0x2FFFFFFA.

3) Primer graničnog slučaja, pri čemu je B_2 najamnji broj za koji je neophodno pamtiti u 32 bita, a broj B_1 najmanji mogući broj koji se može uzeti za datu vrednost broja B_2
B_1 = 0x40000001
B_2 = 0x10000
Vrednost izraza nakon primenjivanja neophodnih operacija je 0x3AAAAAAC55555554

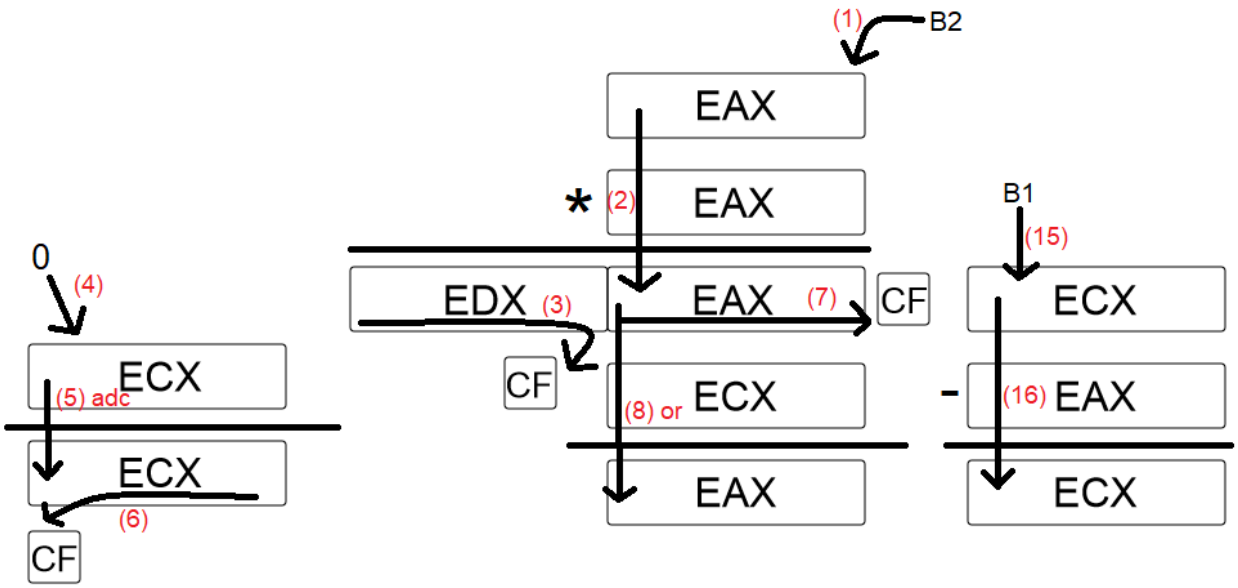
Nakon priloženih primera moguće je izvršiti dopunu zaključka navedenog na početku koji govori o uslovima pod kojima je moguće rešiti zadati problem. Uslov navodi da je maksimalna vrednost koju broj B_2 može imati 0x1FFFF, pri čemu B_1 mora biti veće od $B_2^2/4$, tj. veće od 0xFFFF0000. Na osnovu primera 2 zaključujemo da B_1 ne može imati vrednost 0xFFFFFFFF.
Na osnovu primera 1 i 3 zaključujemo da postoje primeri u kojima proizvod prva dva izraza prilikom deljenja trecim izrazom dovodi do toga da konačno rešenje ima 9 ili više bajtova, a kako deljenje 32-bitnim brojem smešta rezultat u registar EAX, koji je 8-bajtni, dolazi do prekoračenja. Zbog toga je neophodno da broj B_2 bude manji od 0x10000.

4) Primer prosečnih vrednosti operanada B_1 i B_2
   B_1 = 0xE723C3
   B_2 = 0x3C4
   $B_2^2/4$ = 0x38B84
   $B_1 - B_2^2/4$ = 0xE3983F
   [B1*] – 1 = 0xAAEFABEA
   Konačno rešenje 0xAD9944D6, ostatak 0xCD9C7E.

Dati problem rastavlja se na 4 potproblema:

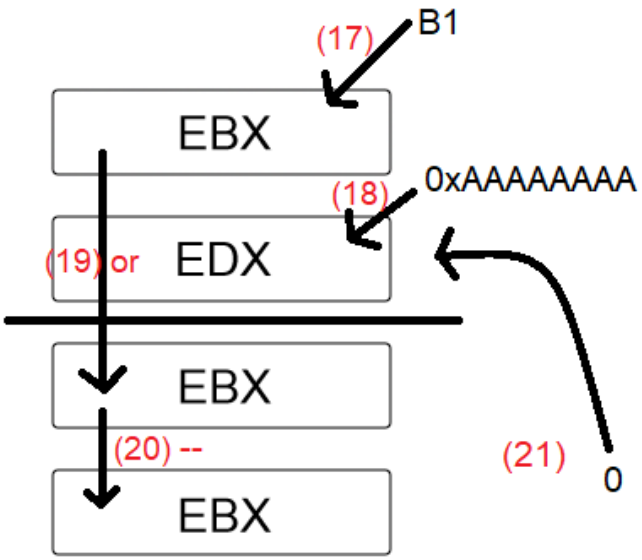
- izračunavanje izraza ($B_1 - B_2^2/4$) i smeštanje međurezultata u registar ECX
- izračunavanje izraza ($[B_1*] - 1$) i smeštanje međurezultata u registar EBX
- izračunavanje izraza ($B_1 + 1$) i smeštanje međurezultata u registar EAX
- izračunavanje izraza oblika A * B / C pri čemu se konačni rezultat smešta u registru EAX

1) Izračunavanje izraza ($B_1 - B_2^2/4$) i smeštanje međurezultata u registar ECX

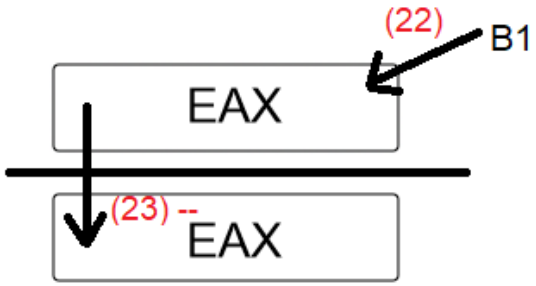


Nakon koraka 8, koraci [3-8] se ponavljaju još jednom, u istom redosledu u kome su izvršeni.

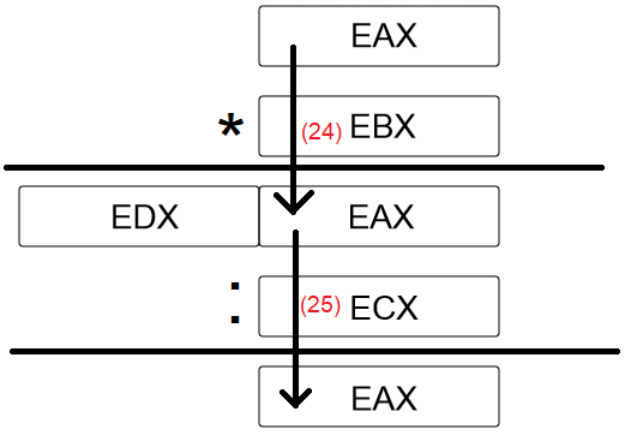
2) Izračunavanje izraza ([B_1*] - 1) i smeštanje međurezultata u registar EBX



3) Izračunavanje izraza (B_1 + 1) i smeštanje međurezultata u registar EAX



4) Izračunavanje izraza oblika A * B / C pri čemu se konačni rezultat smešta u registru EAX



b) **Kod rešenja, izveštaj o testiranju**

```
; 1.
MOV EAX, 0x12F43
; 0x126C2, 0x10000, 0x3C4
MUL EAX
; 2.
SHR EDX, 1
ADC ECX, 0
SHL ECX, 31
SHR EAX, 1
OR EAX, ECX

SHR EDX, 1
MOV ECX, 0
ADC ECX, 0
SHL ECX, 31
SHR EAX, 1
OR EAX, ECX

MOV ECX, 0x81CA2454
;0xFFFFFFFF, 40000001, E723C3
SUB ECX, EAX
MOV EAX, 0
; 3.
MOV EBX, 0x81CA2454
;0xFFFFFFFF, 40000001, E723C3
MOV EDX, 0AAAAAAAAH
OR EBX, EDX
DEC EBX
MOV EDX, 0
; 4.
MOV EAX, 0x81CA2454
;0xFFFFFFFF, 40000001, E723C3
INC EAX

MUL EBX

DIV ECX
```

Prvi deo koda izračunava vrednost broja B_2^2. Drugi deo koda vrši deljenje brojem 4, optimizovano u vidu korišćenja operacije pomeranja binarnog sadržaja udesno. Kako je moguća situacija u kojoj broj B_2^2 zauzima više od 8 bajtova, tj situacija u kojoj je deo kvadriranog broja u registru EDX, neophodno je obezbediti da se istisnute cifre iz registra EDX umetnu na mesto bita najveće težine registra EAX. Ovo je izvršeno tako što se nakon istiskivanja cifra pamti u registru ECX, na mesto cifre najmanje težine. Pomeranjem binarnog sadržaja za 31 mesto ulevo omogućava da se primenom OR operacije između registara EAX i ECX ta cifra umetne na odgovarajuću poziciju. Pošto je deljenje sa 4 ekvivalentno pomeranju binarnog sadržaja za 2 mesta udesno, opisani postupak potrebno je ponoviti 2 puta. Broj koji se dobija smešten je u registar ECX.

Treći deo koda vrši maskiranje i dekrementiranje broja B_1. Razlog za primenu date maske opisan je u diskusiji prvog primera ulaznih vrednosti. Vrednost izraza smeštena je u registar EBX.

Ostatak koda izvršava inkrementiranje broja B_1, pri čemu se vrednost smeštena u akumulator. Vrši se množenje akumulatora registrom EBX i deljenje (sada registra EDX:EAX) registrom ECX.

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Priloženo rešenje ne radi za primere kod kojih je konačno rešenje broj koji ne može da se smesti u jedan registar. Na početku dokumenta i kroz dokument definisani su pojedini uslovi koji ograničavaju vrednosti koje brojevi mogu uzeti.

1. B_1 = 0x81CA2456, B_2 = 0x12F43
   Za ovaj primer početnih vrednosti očekivan rezultat je 0x22E23D574.
   Kako je konačni rezultat 9 bajta, a registar u kome se smešta celobrojna vrednost 8, dolazi do greške „Division Quotient Overflow“. Do poslednjeg koraka, u kome se javila greška, izvršavanje programa je teklo očekivano.
2. B_1 = 0xFFFFFFFF, B_2 = 0x126C2
   Za ovaj primer početnih vrednosti očekivan rezultat je 0x2FFFFFFA.
   Ovo rešenje se ne poklapa, iz razloga navedenih prilikom predstavljanja početnih vrednosti. Odstupanje od ispravnog rešenja događa se prilikom množenja registara u liniji 35.
3. B_1 = 0x40000001, B_2 = 0x10000
   Očekivano rešenje: 0x3AAAAAAC55555554.
   Primer ulaznih vrednosti kod koga se odstupanje javlja u poslednjem koraku prilikom deljenja, uz isto objašnjenje kao za prvi primer ulaznih vrednosti.
4. B_1 = 0xE723C3, B_2 = 0x3C4
   Kod datog primera ulaznih vrednosti dobija se tačno rešenje. Očekivano rešenje: 0xAD9944D6.

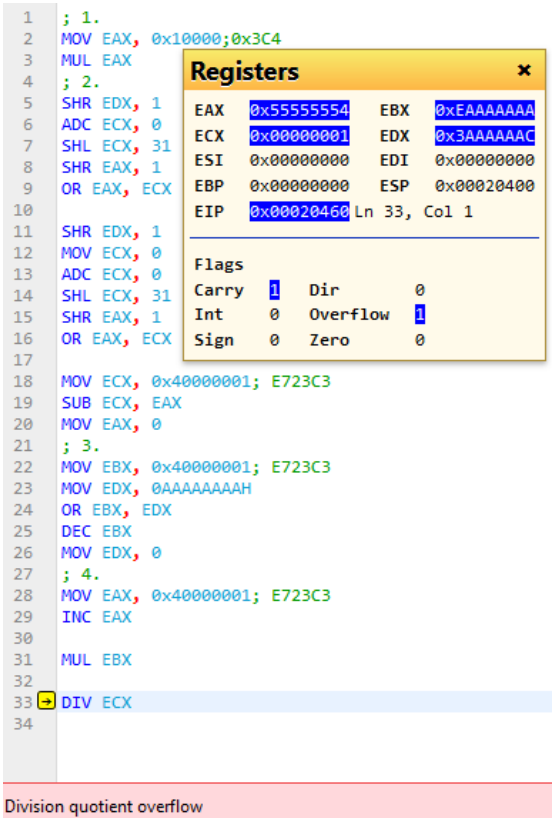
```
2  MOV EAX, 0x12F43
3  ; 0x126C2, 0x10000, 0x3C4
4  MUL EAX
5  ; 2.
6  SHR EDX, 1
7  ADC ECX, 0
8  SHL ECX, 31
9  SHR EAX, 1
10 OR EAX, ECX
11
12 SHR EDX, 1
13 MOV ECX, 0
14 ADC ECX, 0
15 SHL ECX, 31
16 SHR EAX, 1
17 OR EAX, ECX
18
19 MOV ECX, 0x81CA2454
20 ;0xFFFFFFFF, 40000001, E723C3
21 SUB ECX, EAX
22 MOV EAX, 0
23 ; 3.
24 MOV EBX, 0x81CA2454
25 ;0xFFFFFFFF, 40000001, E723C3
26 MOV EDX, 0AAAAAAAAH
27 OR EBX, EDX
28 DEC EBX
29 MOV EDX, 0
30 ; 4.
31 MOV EAX, 0x81CA2454
32 ;0xFFFFFFFF, 40000001, E723C3
33 INC EAX
34
35 MUL EBX
36
37 DIV ECX
```

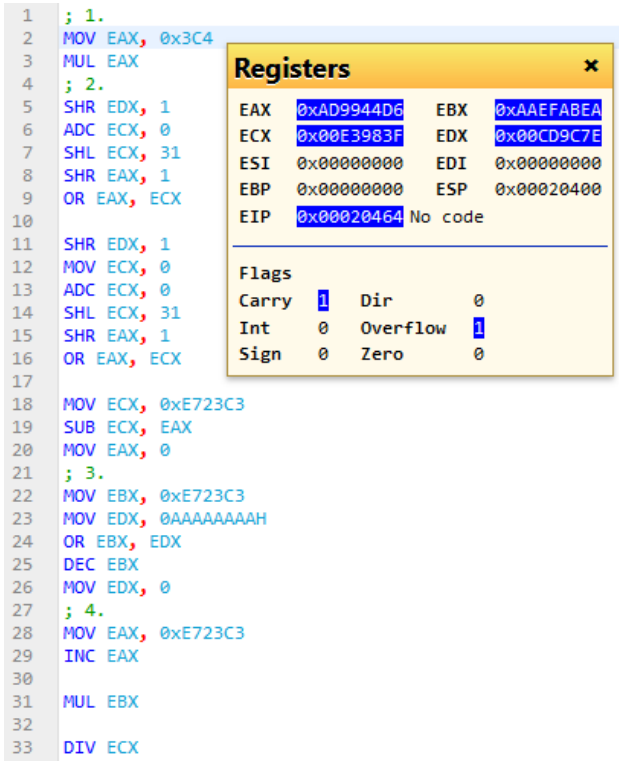
```
Registers
EAX  0xA629AE01   EBX  0xABEAAEFD
ECX  0x27FA3972   EDX  0x572901C1
ESI  0x00000000   EDI  0x00000000
EBP  0x00000000   ESP  0x00020400
EIP  0x00020460 Ln 37, Col 1

Flags
Carry 1   Dir      0
Int   0   Overflow 1
Sign  1   Zero     0
```

```
Division quotient overflow
```

```
2  MOV EAX, 0x126c2
3  ; 0x10000, 0x3C4
4  MUL EAX
5  ; 2.
6  SHR EDX, 1
7  ADC ECX, 0
8  SHL ECX, 31
9  SHR EAX, 1
10 OR EAX, ECX
11
12 SHR EDX, 1
13 MOV ECX, 0
14 ADC ECX, 0
15 SHL ECX, 31
16 SHR EAX, 1
17 OR EAX, ECX
18
19 MOV ECX, 0xffffffff
20 ;40000001, E723C3
21 SUB ECX, EAX
22 MOV EAX, 0
23 ; 3.
24 MOV EBX, 0xffffffff
25 ;40000001, E723C3
26 MOV EDX, 0AAAAAAAAH
27 OR EBX, EDX
28 DEC EBX
29 MOV EDX, 0
30 ; 4.
31 MOV EAX, 0xffffffff
32 ;40000001, E723C3
33 INC EAX
34
35 MUL EBX
36
37 DIV ECX
```

```
Registers
EAX  0x00000000   EBX  0xFFFFFFFE
ECX  0xAB27753E   EDX  0x00000000
ESI  0x00000000   EDI  0x00000000
EBP  0x00000000   ESP  0x00020400
EIP  0x00020464 No code

Flags
Carry 0   Dir      0
Int   0   Overflow 0
Sign  0   Zero     1
```





Samoevaluacija

Na skali 0-5 (0 - „nikako“, „nimalo“; 5 - „potpuno“), u kom stepenu smatrate da ste:

1) ***bili savladali gradivo PRE početka rada na vežbi***	***4***
2) ***razumeli zadatak***	***3***
3) ***ispunili zahteve zadatka a)***	***4***
4) ***ispunili zahteve zadatka b)***	***4***
5) ***istestirali i opisali funkcionisanje svog rešenja***	***5***
6) ***razumeli ponašanje svog rešenja i pojedinih instrukcija i mehanizama***	***4***
7) ***imali dovoljno vremena za vežbu***	***5***
8) ***unapredili svoje znanje u toku vežbe***	***4***

<Luka Veličković, 22.11.2022, 17:10 >