

Egzamino pavyzdys su sprendimais

1. Perversti dešimtainį skaičių -19,019 į šešioliktainę.
2. Užrašyti dešimtainį skaičių -78 skaičiaus su ženklų formatu dviejuose baituose šešioliktaine sistema.
3. Užrašyti dešimtainį skaičių -3 slankaus kabelio formatu keturiuose baituose šešioliktaine sistema.
4. Apskaičiuoti valdymo perdavimo adresą:
71E0 E8 D1A2 call number
Per egzaminą buvo toks:
9854 EC90 jmp poslinkis
5. Registrų reikšmės yra: DS = 21FE, SS = 5634, CS = 31CC, ES = 41E3, BP = 9A32, BX = 7536, SI = 45FA, DI = 22F1. Apskaičiuoti operando efektyvų adresą pagal adresavimo baitą AE. Po adresavimo baito yra 8974.
6. Registrų reikšmės yra: DS = 21FE, SS = 5634, CS = 31CC, ES = 41E3, BP = 9A32, BX = 7536, SI = 45FA, DI = 22F1. Apskaičiuoti operando absoliutų adresą pagal adresavimo baitą AE. Po adresavimo baito yra 8974.
7. Parašyti mikrokomentą, kuri skaičių 16383 nusiunčia į registrą MBR.
Per egzaminą reikėjo užrašyti 34 į MBR dviem komandomis MPL kalba.
8. Registras SF = 0000. Baitų atimties operacija iš dešimtainės reikšmės 88 yra atimta dešimtainė reikšmė (-44). Pagal rezultatą užrašyti naują registro SF reikšmę.
9. Registras AL = 07, registras AH = 05, registras BX = AF00, registras CX = 0001. Kokia bus registro AX reikšmė šešioliktainėje sistemoje įvykdžius komandą AAD?
10. Registrų SI ir DI reikšmės yra 001A, registras CX = 0002, registras SF = 0C00. Kokia bus registrų SI ir DI reikšmių suma, įvykdžius komandą: rep stow?

Ats.:

1. 13,0(4DD2F1A9FBE76C8B439581062)₁₆
2. FFB2
3. C0 40 00 00
4. a) 14B4; b) 9842.
5. 0EBB
6. 571FB
7. a) RIGHT_SHIFT(COM(SIGN) + 0)
b) X = 15; D = 1 + 1;
MBR = LEFT_SHIFT(X + D);
8. 0885
9. 0039
10. 0030

Sprendimai:

1. Pirmiausia paverčiame sveikąją dalį $19_{10} = 13_{16}$.

Tuomet verčiame trupmeninę dalį $0,019_{10} = 0,0(4DD2F1A9FBE76C8B439581062)_{16}$

Versdami trupmeninę dalį dauginame ją iš 16 ir kiekvieną kartą užsirašome sveikąją dalį.

Skliauteliuose dešimtainės sistemos skaičiai verčiami į šešioliktinės sistemos skaitmenis:

$$0,019 * 16 = 0,304 (0 = 0)$$

$$0,304 * 16 = 4,864 (4 = 4)$$

$$0,864 * 16 = 13,824 (13 = D)$$

$$0,824 * 16 = 13,184 (13 = D)$$

$$0,184 * 16 = 2,944 (2 = 2)$$

$$0,944 * 16 = 15,104 (15 = F)$$

$$0,104 * 16 = 1,664 (1 = 1)$$

$$0,664 * 16 = 10,624 (10 = A)$$

$$0,624 * 16 = 9,984 (9 = 9)$$

$$0,984 * 16 = 15,744 (15 = F)$$

$$0,744 * 16 = 11,904 (11 = B)$$

$$0,904 * 16 = 14,464 (14 = E)$$

$$0,464 * 16 = 7,424 (7 = 7)$$

$$0,424 * 16 = 6,784 (6 = 6)$$

$$0,784 * 16 = 12,544 (12 = C)$$

$$0,544 * 16 = 8,704 (8 = 8)$$

$$0,704 * 16 = 11,264 (11 = B)$$

$$0,264 * 16 = 4,224 (4 = 4)$$

$$0,224 * 16 = 3,584 (3 = 3)$$

$$0,584 * 16 = 9,344 (9 = 9)$$

$$0,344 * 16 = 5,504 (5 = 5)$$

$$0,504 * 16 = 8,064 (8 = 8)$$

$$0,064 * 16 = 1,024 (1 = 1)$$

$$0,024 * 16 = 0,384 (0 = 0)$$

$$0,384 * 16 = 6,144 (6 = 6)$$

$$0,144 * 16 = 2,304 (2 = 2)$$

$$0,304 * 16 = 4,864 \text{ (pradedą periodiškai kartotis)}$$

Taigi, $-19,019_{10} = -13,0(4DD2F1A9FBE76C8B439581062)_{16}$

2. $78_{10} = 0100\ 1110_2$

Dviejuose baituose tai atrodytų:

0000 0000 0100 1110

1111 1111 1011 0001 (invertuojame)

1111 1111 1011 0010 (+1)

Tai ir yra -78. Šešioliktinėje sistemoje tai būtų FFB2.

3. $-3_{10} = -11 * 2^0 = (-1)^1 * 1,1 * 2^1$ (1 laipsnis prie dvejetainio yra eilė). Charakteristika = eilė + 7F = 1 + 7F = 80

Skaičius slankaus kablelio formatu 4 baituose atrodo taip: ženklui 1 bitas (jei teigiamas - 0, jei neigiamas - 1),

charakteristikai 8 bitai, o likę 23 mantisei. -3 slankaus kablelio pavidale yra:

1100 0000 0100 0000 0000 0000 0000 0000. Šešioliktiniu pavidalu tai yra C0 40 00 00.

Jei reiktų apskaičiuoti pvz. 3,1 tai trupmeninė dalis gaunasi periodinė ir tada mantisę pildom tol, kol yra vietos.

4. E8 yra call vidinis tiesioginis, t.y. poslinkis 2 baitai. Pati komanda užima 3 baitus (E8, D1 ir A2), tad IP padidiname 3, ($71E0 + 3 = 71E3$). Tada adresas bus perduodamas $71E3 + A2D1$ (sukeista vietomis, nes pirmiau mašiniame kode eina jaunesnysis baitas, po to vyresnysis) = 14B4.
EB yra jmp vidinis artimas, t.y. poslinkis 1 baitas. Tada adresas bus perduodamas: 9856 (prie esamo adreso pridedame 2, nes IP yra ne einamoji komanda, o kitos skaitomos komandos adresas t.y. 9856) + FFEC (priekyje prirašome FF, nes EC yra neigiamas, nes vyriausiam bite yra vienetas, o poslinkis traktuojamas kaip -128 - 127) = 9842.
5. AE = 10101110, pirmi du bitai 10 yra mod reikšmė, jie reiškia, jog poslinkis 2 baitų. Paskutiniai trys bitai 110 yra r/m ir reiškia, jog efektyvus adresas bus formuojamas EA := BP + poslinkis. EA = $9A32 + 7489 = 10EBB$. Adreso ilgis yra 4 bitai, tad pirmąjį šešiolyktainį skaitmenį pašalinam (EA = 0EBB). Taip pat svarbu prisiminti, jog rašydami poslinkį jo baitus sukeičiame vietomis, nes mašiniame kode pirmiau eina jaunesnysis baitas, po to vyresnysis.
6. Absoliutus adresas operandui yra formuojamas pagal šabloną DS : <operando EA> (yra kelios išimtys, kurias reiktų žinoti). Pvz. šiuo atveju EA formavimui buvo panaudotas registras BP, o tai iššaukia steko segmento (SS) registro panaudojimą vietoj DS. Operando EA turime iš 5 uždavinio, tai absoliutus adresas 5634 : 8974. Atsakyme pateikiant absoliutų adresą reikia segmento registrą padauginti iš 10_{16} ir pridėti EA. Taigi, AA := $56340 + 0EBB = 571FB$.
7. Idėja tokia – siunčiame į kairįjį sumatorių SIGN(1000000000000000) ir jį invertuojame ($0111111111111111_2 = 32767_{10}$) ir gautą rezultatą pastumiame į dešinę, t.y. div 2 ($001111111111111111_2 = 16383_{10}$). Tai būtų MBR = RIGHT_SHIFT(COM(SIGN) + 0).
8. Dėl šito uždavinio tiksliai nežinau, bet mano manymu formotu CF (nes yra skolinimasis), PF (rezultato vienetų skaičius lyginis), OF (perpildymas, nes rezultatas netelpa intervale -128 - 127), SF (nes vyriausiam bite 1). Sudedant reiktų skaičius pasiversti į dvejetainę skaičiavimo sistemą ir atimti, tada šios žymės paaiškėtų.
Taigi SF = 0000 1000 1000 0101 = 0885.
9. Komanda AAD atlieka tokius pertvarkymus:
AL := AH * 10 + AL;
AH := 0;
AL reikšmė bus $57_{10} = 39_{16}$
AH := 0;
AX reikšmė bus 0039.
10. Čia reikia žinoti eilutinių komandų vykdymo schemą ir SF registro sudėtį. Iš SF nustatome, kad DF = 1, o stosw (raide w rodo, kad bus operuojama žodžiais). Vadinasi SI ir DI reikšmės bus mažinamos 2. (iš DF = 1 ir tai kad operuojama žodžiais). Komanda STOS modifikuoja tik DI registro reikšmę. Pirmą kartą vykdoma komanda rep stosw (atliekami tokie veiksmai: tikrina ar CX = 0 (jei lygu, baigia pakartojimo komandą), jei CX <> 0, tai CX := CX - 1 (šiuo atveju CX := CX - 1 = 0001), tada DI := DI - 2 = 0018). Antrą kartą vykdoma komanda rep stosw (CX <> 0, tai CX := CX - 1 (šiuo atveju CX := CX - 1 = 0000), tada DI := DI - 2 = 0016). Trečią kartą vykdoma komanda rep stosw (CX = 0, baigiama vykdyti pakartojimo komanda). SI reikšmė nepakito (išliko 001A), o DI tapo 0016. Tada DI

+ SI = 0016 + 001A = 0030. (Reikia atkreipti dėmesį, kad pakartojimo komandos nutraukiamos ir dar kai $Z = ZF$, t.y. specifiniai atvejai).