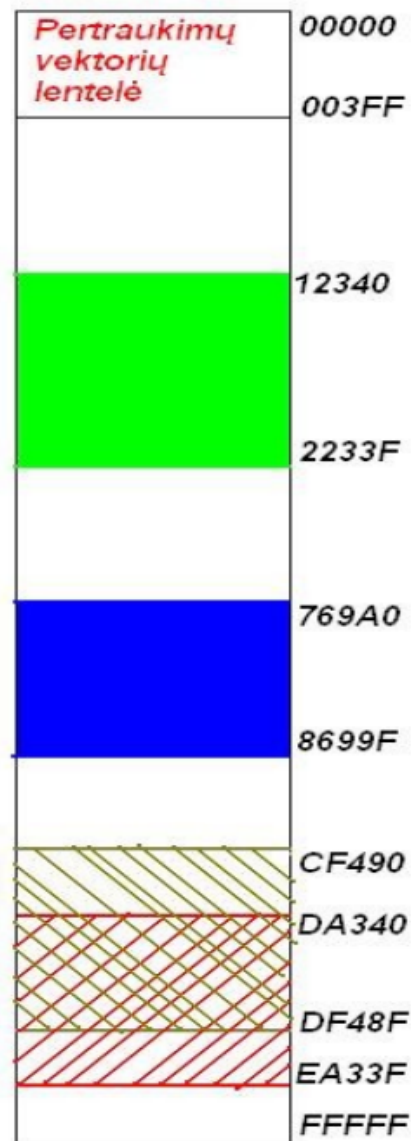


Atmintis ir jos sandara

- Mūsų nagrinėjamoje architektūroje atminties dydis yra 2^{20} baitų (1 MB)
- Paragrafas – 16 baitų (10h) dydžio blokas.
- Atmintis yra skirstoma į segmentus:
 - CS – Code Segment
 - DS – Data Segment
 - ES – Extra Segment (extra data segment / papildomas duomenų segmentas)
 - SS – Stack Segment
- Segmento dydis: 10000h (arba 64K)
- Ir atmintis, ir segmentai yra **cikliniai**:
 - FFFFF+1=00000h
 - FF8A+AB=0035h



- Segmentiniai registrai yra 16 bitų reikšmėse. Pagal pateiktą paveikslėlį (*gali būti kitaip!*) segmentinių registrų reikšmės yra:
 - CS = 1234h
 - DS = 769Ah
 - ES = CF49h
 - SS = DA34h
- Segmentinis registras nurodo segmento pradžią atmintyje, kuri yra adresu: segmento registro reikšmė * paragrafo dydis (10h)
 - Pvz.: DS pradžia atmintyje yra $769A * 10 = 769A0h$

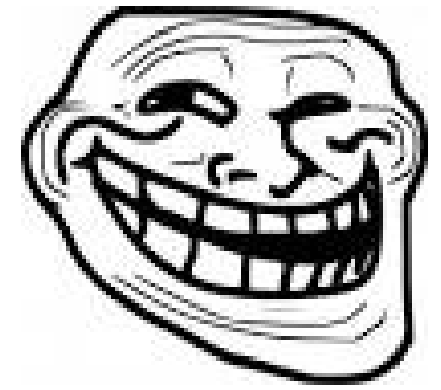
Efektyvus ir absoliutus adresas

- Efektyvus adresas (EA): 4 šešioliktainiai skaitmenys (pvz. ACDCh). Jis parodo poslinkį nuo segmento pradžios.
- Absoliutus adresas (AA): 5 šešioliktainiai skaitmenys (pvz. BABA5h). Jis parodo elemento adresą atmintyje.

$$\mathbf{AA = segmentinio_registro_reikšmė * 10h + EA}$$

(dar užrašomas seg:EA, pvz.: DS:EA)

Adresavime svarbūs registrai



- SI – Source Index. Dažniausiai naudojamas su DS (DS:SI)
- DI – Destination Index. Dažniausiai naudojamas su ES (ES:DI)
- IP – Instruction Pointer. **Komandos vykdymo metu rodo į sekančios vykdymos komandos pradžią.** Rodo poslinkį nuo kodo segmento pradžios, dažniausiai naudojamas su CS (CS:IP)
- BP – Base Pointer. Dažniausiai naudojamas kartu su SS (SS:BP)
- SP – Stack Pointer. Rodo į steko viršūnę, dažniausiai naudojamas kartu su SS (SS:SP)

Uždavinukas: Registų reikšmės: SS=ABCD, SP=0005, BP=1456, SI=BOOB, DI=7536. Duotame kode 9A yra operacijos kodas, EB yra automatizacijos plėtinys, FE yra mikrokomandos GOGO reikšmė. Vykdomoji komanda kode yra sudaryta iš 5 baitų. Kokia bus IP reikšmė komandos vykdymo metu?

1234 9A EBF6789453124

CALL TEXT (1234 yra poslinkis kodo segmente)

Ats: $1234 + 5 = \underline{1239h}$

Plėtimo pagal ženklą taisyklė

- Plėsti pagal ženklą reikia tuomet, kai prie žodžio (2 baitų) reikia pridėti 1 baitą.
- Plečiama **pagal ženklo bitą** (patį kairiausią bitą). Koks ženklo bitas, tokį bitą reikia prirašyti iš kairės 8 kartus.
 - 0011 0101 -> 0000 0000 0011 0101 (35h -> 0035h)
 - 1000 0001 -> 1111 1111 1000 0001 (81h -> FF81h)

1. IP registro reikšmė yra 2222h, poslinkis 7Fh. Kokia bus nauja IP reikšmė?

Ats.: 22A1h

2. IP registro reikšmė yra 4587h, poslinkis 87h. Kokia bus nauja IP reikšmė?

Ats.: 450Eh

Adresavimo būdai

- Tiesioginė adresacija:
 - Pagal adresavimo baitą
 - Pagal operacijos kodą:
 - Paprasta
 - Santykinė
 - Absoliuti
- Netiesioginė adresacija

Adresavimo būdai

- Pagal adresavimo baitą:
MOV AX, [bx+55] -> 8B 47 55 (*47 yra adresavimo baitas*)
- Paprasta tiesioginė adresacija:
MOV AX, word ptr wild_stuff -> A1 05 00 (*0005 yra operando EA*)
- Santykinė tiesioginė adresacija:
JE wild_place -> 74 F8 (*F8 yra poslinkis nuo esamos pozicijos*)
- Absoliuti tiesioginė adresacija:
JMP 5511h:8722h -> EA 22 87 11 55 (*komandoje tiesiogiai nurodytas EA (8722) ir segmento registro reikšmė (5511)*)
- Netiesioginė adresacija:
JMP [bx+si] -> FF 20 (*pagal adresavimo baitą suformuojama AA; **tuo adresu esanti reikšmė** laikoma operando adresu*)

Adresavimo baitas

Adresavimo (adresacijos) baitas eina po operacijos kodo (OPK). Adresavimo baito sandara:

mod	reg	r/m
2 bitai	3 bitai	3 bitai

Pvz.: Adresacijos baitas yra 8A.

8Ah=**10**|**001**|**010**
mod reg r/m

Mod nusako, koks yra poslinkis:

- 00 – 0 baitų poslinkis (išimtis, kai r/m = 110)
- 01 – 1 baito poslinkis (reiks plėst pagal ženklą!)
- 10 – 2 baitų poslinkis
- 11 – r/m laukas imamas kaip registras (t.y., nėra operando atminty)

Smagioji lentelė

Lentelę reikia mokėti mintinai!

w (word/width) – bitas, kuris nurodo, ar operuojama baitais, ar žodžiais. Šis bitas būna operacijos kodo viduje.

- w=0 (word yra FALSE) → operuojama baitais
- w=1 (word yra TRUE) → operuojama žodžiais

d (destination) – bitas OPK viduje, kuris nurodo kryptį (kas yra operandas-šaltinis, o kas yra operandas-rezultatas)

- d=0 → šaltinis = reg, rezultatas = r/m
- d=1 → šaltinis = r/m, rezultatas = reg

Pvz.: 1000 10dw mod reg r/m [poslinkis] – MOV
registas ←→ registras/atmintis

EA nustatom pagal r/m!
Naudinga:

uosis.mif.vu.lt/~julius/Tools/asm/KomKodaiViso.pdf

r/m – register or memory, registras arba atmintis.

Reg arba r/m	mod=00	mod=01, 10	mod=11	
			w=0	w=1
000	BX+SI+poslinkis		AL	AX
001	BX+DI+poslinkis		CL	CX
010	BP+SI+poslinkis		DL	DX
011	BP+DI+poslinkis		BL	BX
100	SI+poslinkis		AH	SP
101	DI+poslinkis		CH	BP
110	tiesioginis adresas*	BP+poslinkis	DH	SI
111	BX+poslinkis		BH	DI

*Tiesioginis adresas – Dviejų baitų poslinkis, be jokių pridėtų registrų.

Pavyzdinė užduotis (1)

CS=1111, ES=5342, SS=6987, DS = 2345, BX=1111. Vykdoma komanda 8B. Duotas mašininis kodas: 8B 8F 12 34. Koks operando atmintyje EA ir AA?

- **8B – operacijos kodas.** Išskleidžiam: 1000 10¹₁

- d=1 → šaltinis r/m, rezultatas reg
- w=1 → operuojama žodžiais

- **8F – adresavimo baidas.**

Išskleidžiam: 10 001 111

- mod=10 → poslinkis 2 baitų
- reg = 001, operuojam žodžiais (nes w=1) → komandoje naudotas registras yra CX
- r/m = 111 → BX + poslinkis

Kadangi mod=10, poslinkis yra 2 baitų, imam 2 baitus po adresavimo baito kaip poslinkį:

- 12 34 → poslinkis bus 34 12h

Svarbu: mašininiame kode žodis yra užrašomas tokia tvarka: *j.b v.b.*, tačiau assembler'yje (o ir atliekant veiksmus) tvarka yra *v.b. j.b.*, todėl reikia nepamiršt baitų apkeisti vietomis!

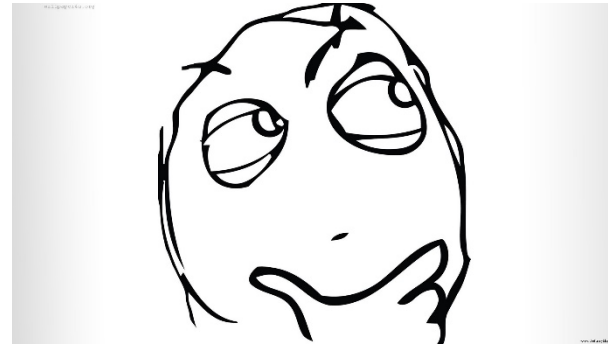
Pavyzdinė užduotis (2)

Koks bus EA?

- Iš adresavimo baido išsiaiškinom, kad EA formuosim taip: $BX + \text{poslinkis} \rightarrow 1111 + 3412 = 4523h$
- EA = 4523h

Kaip gaut AA?

- Žinom formulę: $AA = \text{seg} * 10h + EA$.
- $EA = 4523$, o koks čia segmentas?



Reik dar teorijos!

Segmento nustatymas adresavime

Prefiksas – tai prieš OPK mašininiame kode einantis baitas, kuris gali pakeisti naudojamą segmentą. Pvz.: 26 8B 8F 12 34

Trys taisyklės, kurias reik žinoti:

1. Ar buvo panaudotas prefiksas? Jei taip, imam prefiksą atitinkantį segmentą.
2. Ar formuojant EA buvo panaudotas BP? Jei taip, imam SS.
3. Jei formuojant EA nepanaudotas BP → imam DS.

Segmentas	Maš. Kodas	Fokusas
ES	26	Europos Sąjunga
CS	2E	Žaidė CS'ą
SS	36	Sovietų Sąjunga
DS	3E	Dėstė Saikingai (Dėjo Sk**są)

Pavyzdinė užduotis (3)

Tikrinam pagal taisykles:

- Ar buvo prefiksas? Ne, nes maš. kodas buvo 8B 8F 12 34.
- Ar naudojom BP? Ne, nes EA sudarėm pagal BX + poslinkis.
- Reiškia, segmentas bus DS.

$$AA = DS * 10h + EA = 2345 * 10 + 4523 = 27973h$$

Ats.: AA = 27973h, EA = 4523h

Būtina! parašykit atsakymą!

Uždaviniai

1. Visų registrų reikšmės yra FFFF. Kokia komanda vykdoma ir koks operando atmintyje EA ir AA, jei komandos mašininis kodas yra: 8A F1 55 88.
2. AX=0000, BX=1111, CX=2222, DX=3333, SI=4444, DI=5555, BP=6666, SP=7777, IP=8888, CS=1234, DS=2345, SS=FFEE, ES=4567. Koks operando atmintyje EA ir AA, jei komandos mašininis kodas yra 2E 8C 1E 28 37.
3. 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 82.
Po adresavimo baito seka baitai: FE01
4. DS=FE21, SS=3456, CS=CC31, ES=E341, BP=329A, BX=3675, SI=FA45, DI=F122.
Apskaičiuoti operando absoliutų adresą pagal adresavimo baitą 6E. Po adresavimo baito seka baitai: ABBA.

Daugiau uždavinių galite rasti [Beno konspekte](#) arba dėstytojo J. Andrikonio [skaidrėse](#).