VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ
3 KURSAS

# Realios ir virtualios mašinų aprašas

Atliko:

Augustinas Jarockis,

Arminas Bražėnas

Vilnius

2025-02

## Sąvokos

- Bitas – informacijos saugojimo vienetas, kuris gali būti 0 arba 1.
- Baitas – 8 bitai.

## Bito pozicija - Bitų numeravimas

- Registro bitai numeruojami iš dešinės į kairę, pradedant nuo 0. Pavyzdžiui, 32 bitų registro pirmasis bitas (žiūrint iš dešinės) bus vadinamas 31-uoju bitų, o pirmasis iš kairės – nuliniu bitu.

## Baitų numeravimas

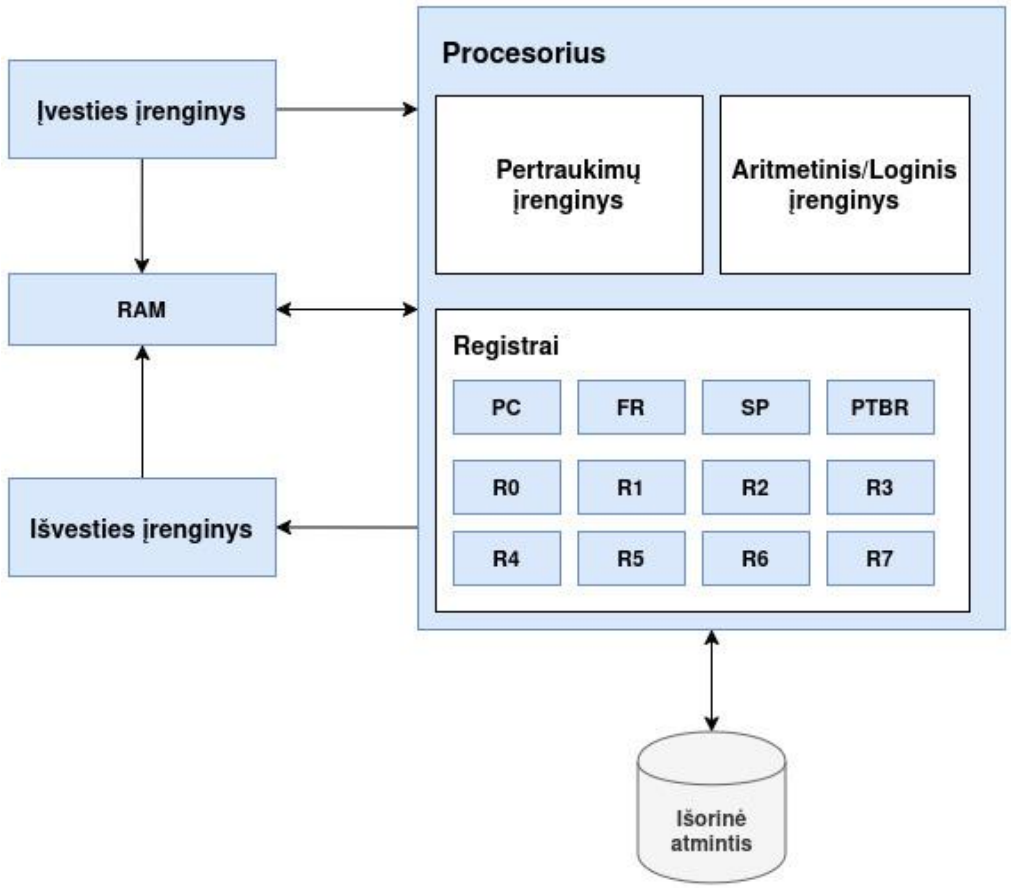
- Baitai numeruojami iš kairės į dešinę, pradedant nuo vieno. Pavyzdžiui, registre, pirmam baitui priklausys 31-24 bitai, antram baitui – 23-16 bitai, trečiam baitui – 15-8 bitai, o ketvirtam – 7-0 bitai.
- Procesorius naudoja big-endian architektūrą, t. y. labiausiai reikšmingas baitas yra saugomas pirmas. Pavyzdžiui, jeigu turime 32 bitų skaičių 0xAABBCCDD, tuomet jis atmintyje bus saugomas taip: AA BB CC DD

## Neigiamų skaičių vaizdavimas

Neigiamiems skaičiams vaizduoti naudojamas dvejeto papildymo metodas.

Daugiau informacijos: https://en.wikipedia.org/wiki/Two%27s_complement

## Realios mašinos schema




*1 pav. Realios mašinos schema*

## Registrai

Visi mašinos registrai yra po 4 baitus. Jie skirstomi į:

- Bendrosios paskirties:
  - **R0, R1, R2, R3, R4, R5, R6, R7**
- Būklės valdymo:
  - **PC** (program counter) - saugo sekančios išrenkamos instrukcijos atmintyje adresą.
  - **FR** (flags register) - saugo procesoriaus būseną:
    - **C** – carry flag (0 pozicija)
      - Tampa 1, jei operacijos rezultatas pakeistų 33 bitą, jeigu jis egzistuotų registre. Tampa 0 kitu atveju.
    - **Z** – zero flag (1 pozicija)
      - Tampa 1, jei operacijos rezultatas yra lygus 0. Tampa 0 kitu atveju.
    - **MODE** – mode flag (2 pozicija)
      - Žymi mašinos režimą; 0 – reali mašina, 1 - virtuali
    - **S** – sign flag (3 pozicija)
      - Tampa 1, jei operacijos rezultatas yra neigiamas
  - **SP** (stack pointer) – saugo stacko viršaus adresą

- **PTBR** (page table base register) – saugo puslapių lentelės pradžios adresą

# Operacijų sistema:

Visos instrukcijos yra 4 baitų dydžio. Visos operacijos pasiekiamos ir realioje, ir virtualioje mašinoje, nebent yra nurodyta kitaip.

## Registrų kodai:

1. R0 – 00000
2. R1 – 00001
3. R2 – 00010
4. R3 – 00011
5. R4 – 00100
6. R5 – 00101
7. R6 – 00110
8. R7 – 00111
9. SP – 01000
10. PC – 01001
11. PTBR – 01010 (Pasiekiamas tik su MOV, LOAD ir STORE komandomis)
12. FR – 01011 (Pasiekiamas tik su MOV, LOAD ir STORE komandomis. LOAD ir STORE komandose gali būti tik pirmasis registras)

## Flagų nustatymo principas:

*Loginių ir aritmetinių operacijų flagai*

Aritmetinės ir loginės operacijos gali pakeisti požymių (flagų) registro reikšmes. Taisyklės, pagal kurias keičiamos flagų reikšmės:

- C (CARRY) flagas tampa 1 tada, kai aritmetinės operacijos rezultatas pakeistų 32-o ar didesnio registro bito vertę, jeigu jis egzistuotų. Pavyzdžiui, sudedant du šešioliktainius skaičius 0x70000000 ir 0x90000000, CARRY flagas pavirs 1, nes operacijos rezultatas (0x100000000) yra didesnis nei telpa 32 bituose.
- Z (ZERO) flagas tampa 1 tada, kai operacijos rezultatas yra lygus nuliui. Pavyzdžiui, operacija XOR 0x2 0x5 yra lygi nuliui, todėl operacijos rezultatas irgi bus lygus nuliui.
- S (SIGN) flagas žymi, ar operacijos rezultato 31-asis bitas yra lygus vienam. Kadangi neigiamiems skaičiams žymėti mūsų sistemoje naudojamas dvejeto papildinio metodas, tai efektyviai žymi, ar operacijos rezultatas yra neigiamas. Pavyzdžiui, 1 – 5 operaciją atitinkančios komandos SUB 0x1 0x5 rezultatas yra lygus -4, kas būtų mūsų sistemoje atvaizduota kaip 0xffff fffc, t. y. 31-asis bitas būtų lygus 1, todėl SIGN bitas taip pat taptų lygus 1.

*MODE flagas*

Šis flagas keičiamas ENTER, HALT operacijomis.

## Aritmetinės operacijos:

- ADD – 00000001 00000000 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
  - Sudeda pirmo ir antro registrų reikšmes ir patalpina rezultatą į pirmą registrą.
  - Rezultatas patalpinamas pirmajame registre.
  - Veikia Z, C, S flagus

- SUB – 00000001 00000001 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Atima antro registro reikšmę iš pirmojo registro reikšmės
    - Rezultatas patalpinamas pirmajame registre.
    - Veikia Z, C ir S flagus
- MUL – 00000001 00000010 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Sudaugina pirmo ir antro registro reikšmes.
    - Rezultatas patalpinamas pirmajame R1 ir R2 registruose. R1 registre patalpinami didesnės vertės rezultato bitai, o R2 registre – mažesnės reikšmės rezultato bitai.
    - Veikia Z, C ir S flagus
- DIV – 00000001 00000011 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Padalina pirmojo registro reikšmę iš antrojo registro reikšmės ir patalpina rezultatą pirmajame registre.
    - Jei antrojo registro reikšmė yra nulis, iškviečiama INT 0 operacija.
    - Veikia Z, C ir S flagus
- CMP – 00000001 00000100 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Veikia panašiai, kaip SUB operacija, tik rezultatas nėra niekur patalpinamas, tiesiog pakeičiami flagai.

## Loginės operacijos:

- NEG – 00000001 00010000 00000000 000[5 bitai registrui]
    - Invertuoja visus nurodyto registro bitus.
    - Veikia S flagą
- AND – 00000001 00010001 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Atlieka loginę konjunkcijos operaciją su registrų bitais ir patalpina rezultatą pirmajame nurodytame registre.
    - Veikia S ir Z flagus
- OR – 00000001 00010010 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Atlieka loginę disjunkcijos operaciją su registrų bitais ir patalpina rezultatą pirmajame nurodytame registre.
    - Veikia S ir Z flagus
- XOR – 00000001 00010011 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Atlieka loginę griežtos disjunkcijos operaciją su registrų bitais ir patalpina rezultatą pirmajame nurodytame registre.
    - Veikia S ir Z flagus

## Valdymo perdavimo operacijos:

- JMP – 00000001 00100000 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę.
- JC – 00000001 00100001 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei C (CARRY) flagas yra lygus 1.
- JE – 00000001 00100010 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei Z (ZERO) flagas yra lygus 1.
- JNE – 00000001 00100011 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei Z (ZERO) flagas yra lygus 0.
- JNC– 00000001 00100100 00000000 000[5 bitai registrui]

    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei C (CARRY) flagas yra lygus 0.
- JL – 00000001 00100101 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei Z (ZERO) flagas yra lygus 0 ir S (SIGN) flagas yra lygus 1.
- JM - 00000001 00100110 00000000 000[5 bitai registrui]
    - Pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę, jei Z (ZERO) flagas yra lygus 0 ir S (SIGN) flagas yra lygus 0.
- LOOP – 00000001 00100111 00000000 000[5 bitai registrui]
    - Jei R7 registre saugoma reikšmė nėra lygi 0, pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę ir R7 registre saugomą reikšmę sumažina 1.
- RET – 00000001 00101000 00000000 00000000
    - Pakeičia PC saugomą adresą į stacko viršuje esančius 4 viršutinius baitus. Didžiausias PC baitas pakeičiamas į pačiame viršuje stacko esantį baitą, antras pagal dydį į antrą, ir t. t. Tada padidina SP reikšmę keturiais vienetais.
- CALL – 00000001 00101001 00000000 000[5 bitai registrui]
    - Įdeda į stacko viršų dabartinę PC registre saugomą reikšmę ir pakeičia PC saugomą adresą į nurodytame registre saugomą reikšmę.
    - Pakeičia SP reikšmę, kad jis rodytų į naują stacko viršų.

## Atminties valdymo operacijos:

- LOAD (4 baitai) – 00000010 00000000 000000[5 bitai pirmam registrui] [5 bitai antram registrui]
    - Užkrauna į pirmąjį registrą atmintyje saugomą informaciją, kurios adresas yra antrajame registre saugoma informacija.
- STORE (4 baitai) - 00000010 00000001 000000[5 bitai pirmam registrui] [5 bitai antram registrui]
    - Užkrauna į atmintį pirmajame registre saugomą informaciją. Atminties adresas, į kurį rašoma, yra nurodytas antrajame registre saugoma reikšme.
- LOADB (1 baitas) – 00000010 00000010 000000[5 bitai pirmam registrui] [5 bitai antram registrui]
    - Užkrauna į pirmąjį registrą atmintyje saugomą informaciją, kurios adresas yra antrajame registre saugoma informacija.
    - Užkraunamas tik 1 baitas. Baitas patalpinamas 7-0 registro bituose.
- STOREB (1 baitas) - 00000010 00000011 000000[5 bitai pirmam registrui] [5 bitai antram registrui]
    - Užkrauna į atmintį pirmajame registre saugomą informaciją. Atminties adresas, į kurį rašoma, yra nurodytas antrajame registre saugoma reikšme.
    - Užkraunamas tik 1 baitas.Užkraunami 7-0 registro bitai.
- MOV - 00000001 00110000 000000 [5 bitai pirmam registrui] [5 bitai antram registrui]
    - Perkelia į pirmąjį registrą antrojo registro reikšmę.
- MOVD - 001 [5 bitai registrui] [24 bitai skaičiui, kuris bus įdėtas į registrą]
    - Įkelia paskutiniuose 24 operacijos bituose nurodytą skaičių į registrą.
    - Pirmi 8 skaičiaus bitai bus lygus nuliui.
- PUSH - 00000001 00110001 00000000 000 [5 bitai registrui]
    - Įkelia nurodytame registre esančią reikšmę į stacko viršų.
    - Pakeičia SP reikšmę, kad jis rodytų į naują stacko viršų.
    - Mašinos naudotojas turi pats pasirūpinti, kad stackas neužrašytų svarbių duomenų.
- POP - 00000001 00110010 00000000 000 [5 bitai registrui]

    - Paima stacko viršutiniuose keturiuose baituose esančią reikšmę ir įkelia ją į nurodytą registrą ir padidina SP reikšmę 4 vienetais.
- PUSHALL - 00000001 00110011 00000000 00000000
    - Įkelia visų (išskyrus PC registrą) registrų reikšmes į stacką.
    - Pakeičia SP reikšmę, kad jis rodytų į naują stacko viršų.
    - Mašinos naudotojas turi pats pasirūpinti, kad stackas neužrašytų svarbių duomenų.
- POPALL – 00000001 00110100 00000000 00000000
    - Ištraukia iš stacko visų (išskyrus PC registrą) registrų reikšmes ir sudeda jas į registrus.
    - Reikšmės iš stacko išimamos ir priskiriamos registrams tokia atvirkščia tvarka nei PUSHALL operacijoje. Ši operacija yra atvirkštinė PUSHALL. Iškvietus PUSHALL ir POPALL vieną paskui kitą, stacko ir registrų reikšmės lieka nepakitusios.
    - Pakeičia SP reikšmę, kad jis rodytų į naują stacko viršų.
- POPINT – 00000001 00110101 00000000 00000000
    - Atstato registrų reikšmes priešingai nei jos buvo išsaugotos prieš pradedant pertraukimo paprogramės vykdymą.
    - Ši operacija pasiekiama tik realioje mašinoje.

### Mašinos būsenos operacijos:

- INT - 00000001 01000000 00000000 [8 bitų pertraukimo kodas]
    - Programinis pertraukimas; pertraukimų mechanizmas aprašytas apačioje
- ENTER – 00000001 01000001 00000000 00000000
    - Pereina iš realios mašinos režimo į virtualios mašinos režimą
    - Ši operacija pasiekiama tik realioje mašinoje
    - Realios mašinos SP reikšmė išsaugoma atmintyje adresu 0x00000400
    - Realios mašinos PC reikšmė išsaugoma atmintyje adresu 0x00000408
    - Virtualios mašinos SP reikšmė užkraunama iš atminties adresu 0x00000404
    - Virtualios mašinos PC reikšmė užkraunama iš atminties adresu 0x0000040C
    - FR registro MODE bitas nustatomas į 1
- HALT – 00000001 01000010 00000000 00000000
    - Nurodo virtualios mašinos darbo pabaigą
    - Ši operacija pasiekiama tik virtualioje mašinoje
    - Realios mašinos PC reikšmė užkraunama iš atminties adresu 0x00000408
    - Realios mašinos SP reikšmė užkraunama iš atminties adresu 0x00000400
    - FR registro MODE bitas nustatomas į 0

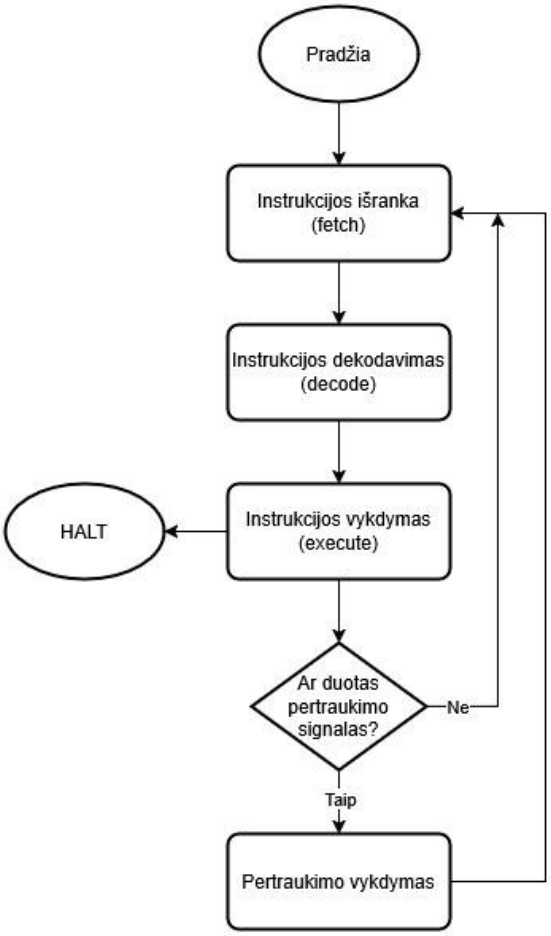
## Procesoriaus darbo algoritmas:

Ciklo veikimo principas:

1. **Išranka**
    a. PC saugomas sekančios išrenkamos instrukcijos adresas
    b. Perskaitoma instrukcija iš adreso, saugomo PC registre
    c. PC reikšmė padidinama 4 vienetais (nes instrukcijos dydis 4 baitai).
2. **Dekodavimas**
    a. Valdymo įrenginys dekoduoja instrukciją:
        i. Koks operacijos kodas

ii. Kokius registrus naudoti
iii. Kokias reikšmes ar atminties adresus naudoti

3. **Vykdymas**
   a. Jeigu pavyko dekoduoti instrukciją, procesorius ją įvykdo
   b. Jeigu nepavyko dekoduoti instrukcijos, iškviečiamas INT 2
4. **Pertraukimo signalo patikrinimas**
   a. Procesorius patikrina aparatūrinių pertraukimų įrenginį, kad sužinotų ar įvyko pertraukimas
   b. Jeigu įvyko pertraukimas, vykdomas pertraukimo mechanizmas (žiūrėti apačioje)
5. **Ciklas kartojamas iš naujo nuo pirmo žingsnio**




*2 pav. Procesoriaus darbo algoritmo būsenų diagrama*

## Pertraukimų mechanizmas

Pertraukimas - signalas apie įvykusį įvykį. Kiekvienas pertraukimas turi savo identifikacinį numerį. Šis numeris naudojamas pertraukimų vektorių lentelėje atrasti adresą paprogramės, kuri turėtų apdoroti pertraukimą.

Pertraukimai skirstomi į:

- Aparatūrinius - sukeliami išorinių prietaisų:
  - INT 0 – dalyba iš nulio
  - INT 1 – klaviatūros įvesties perskaitymas

    - INT 2 – neteisingas opkodas
    - INT 3 – virtuali mašina neturi prieigos teisių į atminties puslapį
    - INT 5 – periodinis pertraukimas
- Programinius - sukeliami programinės įrangos:
    - INT 4 – duomenų išvedimas į terminalą
    - INT 6 – duomenų rašymas į išorinę atmintį
    - INT 7 - duomenų skaitymas iš išorinės atminties

Aparatūriniai pertraukimai yra saugomi pertraukimų įrenginyje LIFO (Last-in First-out) principu, t. y., jeigu įvyko pertraukimai A ir B, pradžiai bus iškviečiama pertraukimo A paprogramė, o po šios paprogramės įvykdymo, bus iškviečiama pertraukimo B paprogramė.

Jeigu pertraukimas įvyko realios mašinos režime:

1. Registrų reikšmės išsaugomos steke panaudojant PUSHALL komandą
2. Iškviečiama paprogramė, kurios adresas saugomas 4 * (pertraukimo kodo skaitinė reikšmė)

Jeigu pertraukimas įvyko virtualios mašinos režime:

1. Virtualios mašinos SP registro reikšmė išsaugoma atmintyje adresu 0x00000404
2. Virtualios mašinos PC registro reikšmė išsaugoma atmintyje adresu 0x0000040C
3. Užkraunama realios mašinos SP registro reikšmė iš atminties adreso 0x00000400
4. Pereinama į realios mašinos režimą
5. Virtualios mašinos registrų reikšmės išsaugomos realios mašinos steke panaudojant PUSHALL komandą
6. Iškviečiama paprogramė, kurios adresas saugomas 4 * (pertraukimo kodo skaitinė reikšmė)

## Laiko skaičiavimas

Mašina turi laikmatį, kuris seka laiką nuo procesoriaus darbo pradžios. Laikas matuojamas milisekundėmis. Šį laiką galima pasiekti per RAM, adresu 0x00000412. Laikas saugomas 4-iuose baituose. Jei nuo laikmačio darbo pradžios praėjo tiek laiko, kad jo nebegalima pavaizduoti 4 baituose, laikas pradedamas skaičiuoti vėl nuo nulio.

Taip pat yra programuojamas intervalų laikmatis (PIT), kuris periodiškai (kas 1 sekundę) iškviečia pertraukimą INT 5.

## Įrenginiai, darbas su jais

### Įvesties įrenginiai

- Klaviatūra:
    - Paspaudus klavišą, atminties adrese 0x00000416 įrašoma paspausto klavišo reikšmė ASCII koduotėje. Tuomet iškviečiamas aparatūrinis pertraukimas INT 1, kuris apdoroja mygtuko paspaudimą operacinės sistemos nustatytų būdu.

### Išvesties įrenginiai

- Tekstinis terminalas:
    - Iškvietus INT 4, mašina paima R3 registre saugomą atminties adresą ir išveda į terminalą viską atmintyje nuo R3 saugomo adreso iki pirmo sutikto nulinio baito. Išvedimui naudojama ASCII koduotė. Išvedimas vyksta imant po vietą baitą ir įrašant jį į atmintį adresu 0x00000417, kur jis yra

automatiškai nuskaitomas ir parodomas terminale. Nuskaičius baitą iš nurodytos atminties vietos, ta atminties vieta yra nunulinama.

### Išorinė atmintis

- Išorinė atmintis yra 64MB dydžio
- Išorinė atmintis yra suskaidyta į blokus po 4KB; blokai numeruojami pradedant nuo 0
- Rašymas į išorinę atmintį:
    - Iškvietus INT 6, mašina paima R3 registre saugomą atminties adresą ir įrašo į išorinę atmintį tiek blokų, kiek nurodyta registre R4. Išorinės atminties bloko numeris, į kurį rašyti duomenis, nurodomas registre R5. Kiekvieno bloko rašymas vyksta taip:
        - Imamas 4KB blokas ir įrašomas į operatyviąją atmintį, pradedant adresu 0x00012004
        - Pirmuose 31 bituose, pradedant adresu 0x00012000, įrašomas išorinės atminties bloko numeris, į kurį rašyt
        - Atminties adreso 0x00012003 baito 0 pozicijos bitas nustatomas į 1. Mašina aptinka, kad šis bitas yra nustatytas ir įrašo bloką į išorinę atmintį. Įrašius bloką į išorinę atmintį, bitas nustatomas atgal į 0.
- Skaitymas iš išorinės atminties:
    - Iškvietus INT 7, iš išorinės atminties mašina nuskaito į R3 registre saugomą atminties adresą tiek blokų, kiek nurodyta registre R4. Išorinės atminties bloko numeris, iš kurio skaityti, nurodomas registre R5. Kiekvieno bloko nuskaitymas vyksta taip:
        - Pirmuose 31 bituose, pradedant adresu 0x00013004, įrašomas išorinės atminties bloko numeris, iš kurio skaityti
        - Operatyviosios atminties adreso 0x00013007 baito 0 pozicijos bitas nustatomas į 1. Mašina aptinka, kad šis bitas yra nustatytas ir nuskaito bloką iš išorinės atminties į operatyviąją atmintį, pradedant adresu 0x00013008.
        - Nuskaičius bloką į operatyviąją atmintį, 0x00013007 baito 0 pozicijos bitas nustatomas į 0.

## Fizinė atmintis

- Mašina turi 4GB fizinės atminties
- Atmintis yra suskirstyta į puslapius po 4KB
- Puslapiai yra numeruojami pradedant nuo 0 (pvz.: 2 puslapis yra fizinėje atmintyje nuo 8KB iki 12KB)

## Virtuali atmintis

- Kiekvienai virtualiai mašinai priklauso atminties puslapiai, kurių negali pasiekti kitos virtualios mašinos
- Virtuali mašina atmintį mato kaip vientisą, prasidedančią nuo 0x00000000
- Virtualios mašinos puslapių lentelės pradžios adresas saugomas PTBR registre
- Pirmuose 4-iuose puslapių lentelės baituose yra nurodytas lentelės dydis, t.y. kiek atminties puslapių priklauso virtualiai mašinai. Toliau saugomi lentelės įrašai.
- Puslapių lentelės įrašas yra sudarytas iš 32 bitų:
    - (0 pozicija) P bitas - nurodo ar virtuali mašina turi prieigos teisę į fizinės atminties puslapį:
        - 0 – mašina neturi prieigos teisių į šį atminties puslapį
        - 1 – mašina turi prieigos teises į šį atminties puslapį
    - (1-31 pozicijos) Visi likę bitai – nurodo fizinės atminties puslapio numerį

- Prieš vykdant skaitymo ar rašymo į atmintį operacijas, patikrinama ar virtuali mašina turi prieigos teises į nurodytą atminties vietą. Pavyzdžiui:
    - Tarkime, kad virtuali mašina nori pasiekti adresą 0x00001010 ir PTBR registro reikšmė yra 0x00005000. Virtuali mašina naudoja 3 atminties puslapius, kurie nukreipia į atitinkamus fizinės atminties puslapius.
    - Puslapių lentelės dydis išsaugotas adrese 0x0005000 = 3. Toliau saugomi puslapių lentelės įrašai:
        - 0x00005004 -> 0x00006001
        - 0x00005008 -> 0x00007001
        - 0x0000500C -> 0x00008001
    - Kadangi puslapio dydis yra 4KB, adresas 0x00001010 yra antrajame atminties puslapyje su poslinkiu 0x00000010. Šio puslapio numeris yra 1.
    - Patikrinama ar puslapis yra puslapių lentelės ribose: 1 <= 3 – 1
    - Patikrinama ar virtuali mašina turi prieigos teisę į antrąjį virtualios atminties puslapį pagal P bito reikšmę: virtuali mašina turi prieigos teisę į atminties puslapį, kadangi 0x00006001 nulinis bitas yra 1
    - Virtualus puslapis ir poslinkis pakeičiamas fiziniu adresu: 0x00006000 * 0x1000 + 0x10 = 0x06000010