

OPERATSIYALARNI BAJARISH

Sultonmuratova Mashhura Quvonchbek qizi

TATU Urganch filiali talabasi

E-mail: sultonmuratovamashhura@gmail.com

Annotatsiya. Ushbu maqolada assembler tilida shartli va shartsiz operatsiyalarni bajarish haqida so'z yuritiladi. Hozirgi kunda mobil qurilmalar zamonaviy hayotimizning ajralmas qismi sanalib, massivlarni qayta ishlashni o'rganish va uzilishlarni tashkil qilish mikroprotssessor va assembler tili darsi jarayonida zamonaviy pedagogik texnologiyalardan unumli foydalanish, o'quvchilarning muhandislik sohasiga bo'lgan qiziqishini yanada oshirish va ularning axborot texnologiyalari bo'yicha egallaydigan bilim, ko'nikma va malakalarini yuqori bo'lishini ta'minlaydi.

Аннотация. В этой статье рассказывается о выполнении условных и безусловных операций на языке ассемблера. В наше время мобильные устройства считаются неотъемлемой частью нашей современной жизни, а обучение обработке массивов и организации прерываний – это эффективное использование современных педагогических технологий в ходе урока микропроцессора и языка ассемблера для дальнейшего повышения интереса учащихся к данной области. инженерного дела, а также повысить их интерес к информационным технологиям, обеспечивает высокий уровень приобретенных знаний, навыков и квалификации.

Annotation. This article explores about performing conditional and unconditional operations in assembly language. Nowadays, mobile devices are considered an integral part of our modern life, and learning to process arrays and organizing interrupts is an effective use of modern pedagogical technologies in the course of a microprocessor and assembly language lesson, to further increase students' interest in the field of engineering, and to increase their interest in

information technology. ensures that the acquired knowledge, skills and qualifications are high.

Kalit soʻzlar: *assembler tili, shartli va shartsiz operatsiyalar, massivlar, registr.*

Ключевые слова: *язык ассемблера, условные и безусловные операции, массивы, регистр.*

Key words: *assembly language, conditional and unconditional operations, arrays, register.*

Kirish. Mikroprotessor va assembler tili bugungi kunda eng keng tarqalgan axborot texnologiyalari hisoblanadi. Assembler tili - kompyuterlar uchun dasturlarni yozishda foydalaniladigan yuqori darajadagi dastur tili hisoblanadi. Assembler tili, kompyuterning birinchi tartibli dastur tili boʻlib, muharrir (text editor) yordamida yozilgan kodlarni kompyuter koʻrish uchun masofaviy maʼlumotlarga aylantiradi. Assembler tilida yozilgan dasturlar, "assembler" deb nomlanuvchi dasturlash dasturida ishga tushiriladi. Assembler dasturida yozilgan kodlar, kompyuter arxitekturasi va mikroprotessorga mos keladigan bitlarga oʻgirishlar sifatida yoritiladi.

Assembler tilida yozilgan dasturlar, yaqin dastur tili sifatida hisoblanadi, chunki ularda kompyuterning arxitekturasida toʻgʻri ishlovchi va arxitekturaga mos keladigan bitlarga oʻgirishlar ishlatiladi. Assembler tilida yozilgan dasturlar yuqori darajadagi nazorat va turli ilovalarni yaratishda, kompyuter jadvalini oʻrganish va korrektsiyalashda keng qoʻllaniladi. Assembler tili, dasturlash tillari oilasiga mansub boʻlgan va kompyuterlarning arxitekturasiga toʻgʻri keluvchi uchta darajali dastur tili hisoblanadi: arxitektura, assembler tilida yozilgan dastur va oʻqish uchun qurilmalar. Assembler tili bilan yozilgan dasturlar, kompyuterning bitlarga oʻgirishlarini toʻgʻri bajarib boʻlmaydigan, birinchi darajali dasturlar hisoblanadi. Assembler tili, kompyuterning arxitekturasiga toʻgʻri keladigan instruksiyalar va direktivlardan iborat. Assembler tilida yozilgan kodlar, instruksiyalar va direktivalar kompyuter arxitekturasi va mikroprotessorga bogʻliq boʻlib, bitlarga oʻgirishlar sifatida yoritiladi.

Assembler tillari ilk bor IBM kompaniyasi tomonidan 1950-yillarda yaratilgan kompyuterlar uchun ishlatilgan. Bu tillar, kompyuter arxitekturasining tushunilishi va instruksiyalarni to'g'ri bajarish uchun yordam berish maqsadida yaratilgan. Assembler tillari, instruksiyalarni kompyuter bittanga o'girishlar sifatida yoritish va kompyuterning boshqaruvini osonlashtirishga qaratilgan. Assembler tili, kompyuter arxitekturasiga bog'liq bo'lgan instruksiyalarni yoritish uchun ko'rsatmalar va direktivalardan foydalanadi. Ko'rsatmalar, kompyuterda amalga oshirilishi kerak bo'lgan arifmetik, mantiqiy, xotira va boshqa operatsiyalarni belgilaydi. Direktivalar esa dasturlash jarayonini boshqarish maqsadida foydalaniladi, masalan, dastur segmentlarini belgilash, ma'lumotlar to'plamlarini tanlash va hokazo. Assembler tillari, avvalo, ma'lumotlar to'plamining binokorlar tomonidan yaratilgan dastur boshqaruvchilari uchun ishlatilgan. Ular dasturlarni bittanga o'girishlar sifatida yoritish imkonini berardi. Assembler tilida bir nechta operatsiyalar mavjud, ulardan asosiylari quyidagilar:

1. Mov (Move): Uzilishlarni o'zgartirish uchun ishlatiladi. Misol uchun, mov ax, bx atamaları orqali bx uzilishini ax uzilishiga ko'chiradi.
2. Add (Qo'shish): Uzilishlarni qo'shish uchun ishlatiladi. Misol uchun, add ax, bx atamaları orqali ax uzilishiga bx uzilishini qo'shib qo'yadi.
3. Sub (Ayirish): Uzilishlarni ayirish uchun ishlatiladi. Misol uchun, sub ax, bx atamaları orqali bx uzilishini ax uzilishidan ayiradi.
4. Mul (Ko'paytirish): Uzilishlarni ko'paytirish uchun ishlatiladi. Misol uchun, mul ax, bx atamaları orqali ax uzilishini bx uzilishiga ko'paytiradi.
5. Div (Bo'lish): Uzilishlarni bo'lish uchun ishlatiladi. Misol uchun, div ax, bx atamaları orqali ax uzilishini bx uzilishiga bo'lib qo'yadi.
6. Jmp (Jump): Shartlarni tekshirish va dastur o'tish uchun ishlatiladi. Misol uchun, jmp label atamasi orqali belgilangan label ga dastur o'tkaziladi.
7. Cmp (Compare): Uzilishlarni solishtirish uchun ishlatiladi. Misol uchun, cmp ax, bx atamaları orqali ax va bx uzilishlarini solishtiradi.
8. Je (Jump if Equal): Uzilishlar bir-biriga teng bo'lsa, belgilangan joyga dastur o'tkaziladi.

9. Jne (Jump if Not Equal): Uzilishlar bir-biriga teng bo'lmasa, belgilangan joyga dastur o'tkaziladi.

10. Je (Jump if Equal): Uzilishlar bir-biriga teng bo'lsa, belgilangan joyga dastur o'tkaziladi.

11. Je (Jump if Equal): Uzilishlar bir-biriga teng bo'lsa, belgilangan joyga dastur o'tkaziladi.

Assembler dasturlash tillarida operatsiyalar kengayib boradi va har bir kompyuter arxitekturasi xususiy operatsiyalarga ega bo'ladi. Assembler tilida yozilgan dasturlarda operatsiyalar kompyuter arxitekturasiga va tilga bog'liq bo'ladi.

Assembler tilida operatsiyalar bilan ishlash uchun sintaksis quyidagicha bo'ladi:

1. Operatsiya nomi: Har bir assembler operatsiyasining nomi mavjud bo'ladi, masalan, ``mov``, ``add``, ``sub``, ``jmp``, va hokazo. Operatsiya nomi dasturda qo'llanilgan operatsiyani belgilaydi.

2. Operatsiya argumetlari: Operatsiyalarning bir yoki bir nechta argumentlari (uzilishlar) bo'lishi mumkin. Argumentlar odatda registrlar (masalan, ``ax``, ``bx``, ``cx``), xotira manzillari (masalan, ``[bx]``, ``[1234]``), sonlar yoki o'zgaruvchilar bo'lishi mumkin.

3. Kommentariyalar: Assembler tilida ``;`` belgisi orqali kommentariyalar yozish mumkin. Kommentariyalar dasturda izohlar qo'yish uchun foydalaniladi va dasturni tushuntirishga yordam beradi.

Sintaksis misol bilan quyidagicha ko'rinadi:

```
# Bu assembler dasturida ax va bx uzilishlarini yig'indisini hisoblaydi
```

```
mov ax, 10 ; ax uzilishiga 10 qiymatini beramiz
```

```
mov bx, 20 ; bx uzilishiga 20 qiymatini beramiz
```

```
add ax, bx ; ax uzilishiga bx uzilishini qo'shamiz
```

Ushbu misolda, ``mov`` operatsiyasi orqali ``ax`` va ``bx`` uzilishlariga qiymatlar beriladi, keyin ``add`` operatsiyasi orqali ``ax`` uzilishiga ``bx`` uzilishini qo'shadi.

Assembler tilida to'g'ri sintaksisga amal qilish uchun, operatsiya nomi, argumentlar va belgilangan sintaksis qoidalariga amal qilish kerak. Har bir

kompyuter arxitekturasi o'zining xususiy assembler sintaksisga ega bo'lishi mumkin, shuning uchun assemblerning dokumentatsiyasidan foydalanish tavsiya etiladi.

Shartli va shartsiz operatsiyalar - dasturlash sohasida foydalaniladigan ikki asosiy kategoriya bo'lib, ulardan har birining o'ziga xos xususiyatlari mavjud:

1. Shartsiz operatsiyalar: Shartsiz operatsiyalar, operandlarga to'g'ridan-to'g'ri amal bajaradi va shartlar bilan bog'liq bo'lmaydi. Bu turidagi operatsiyalar odatda arifmetik, mantiqiy va boshqa hisoblash amallarini o'z ichiga oladi. Masalan, yig'ish (+), ayirish (-), ko'paytirish (*), bo'lish (/) shartsiz arifmetik operatsiyalardir. Bu operatsiyalar operandlarga tegishli amallarni bajarish uchun foydalaniladi. Misol uchun, "x = 5 + 3" dasturida "+" shartsiz arifmetik operatsiya sifatida foydalanilgan.

2. Shartli operatsiyalar: Shartli operatsiyalar, amalni bajarishda shartlarga bog'liq bo'lishi mumkin bo'lgan amallardir. Shartli operatsiyalar odatda mantiqiy muqobil operatorlar (==, !=, <, >, <=, >=) va shartlarni tekshirish (if, else, while, for) bilan bog'liq bo'lishi bilan bilinadi. Bu operatsiyalar dasturda belgilangan shartlarni tekshirish, qayta-qayta amal bajarish, tanlovlar va boshqa muammo yechish uchun foydalaniladi. Misol uchun, "if (x > 10) { y = 5; }" dasturida ">" shartli mantiqiy operatsiya sifatida foydalanilgan.

Shartli va shartsiz operatsiyalar bir xil dasturda ham foydalanilishi mumkin va ulardan foydalanish dastur yozishda muhimdir. Shartsiz operatsiyalar amal bajarish uchun shartlarga bog'liq emas, shartli operatsiyalar esa dastur bajarishda shartlarni tekshirish va mantiqiy muomalalar uchun foydalaniladi. Shartlar va shartli operatsiyalar dasturlarni qo'llashda juda muhim bo'lib, dasturlashda mantiqiy to'g'rimlik va qo'llanuvchining kiritgan ma'lumotlarni tekshirish uchun keng qo'llaniladi.

Besh va ikki sonni taqqoslash uchun quyidagi misolni ko'ramiz:

1. Shartsiz operatsiya misoli:

$$x = 7 + 3$$

Natija: x ning qiymati 10 bo'ladi. Shu odatiy arifmetik operatsiya orqali 7 va 3 ni qo'shib olish amalga oshiriladi.

2. Shartli operatsiya misoli:

```
x = 5
if (x > 10) {
    y = 3;
} else {
    y = 7;
}
```

Natija: x ning qiymati 5 bo'lgani uchun if sharti bajarilmaydi va y ning qiymati 7 bo'ladi. Shu bilan birga, x ning qiymati 10 dan katta bo'lsa if sharti qanoatlantiriladi va y ning qiymati 3 bo'ladi. Bu misollar shartsiz va shartli operatsiyalarning amalga oshirish usullarini namoyish etish uchun berilgan. Shartsiz operatsiyalar, operandlarni to'g'ridan-to'g'ri amalga oshiradi, shartli operatsiyalar esa amalning qaysi shartlar asosida bajarilishi kerakligini tekshiradi va boshqaradi.

Assembler tilida shartli va shartsiz operatsiyalarni bajarish uchun ko'rsatmalar va direktivalar mavjud. Quyidagi misollar assembler tilida shartli va shartsiz operatsiyalarni namoyish etishga yordam beradi:

1. Shartsiz operatsiya misoli (arifmetik operatsiya):

```
MOV AX, 5 ; AX registrga 5 qiymatini yuklash
ADD AX, 3 ; AX registrga 3 ni qo'shish
```

Natija: AX registri 8 qiymatini oladi. Bu misolda ADD ko'rsatmasi arifmetik qo'shish operatsiyasini bajaradi. Bu misolda shartsiz arifmetik operatsiyalar (MOV va ADD) yordamida AX registridagi qiymatlar ustida amal bajariladi. MOV AX, 5: Bu qator, AX registrga 5 qiymatini yuklash amalini bajaradi. Ya'ni, AX registri 5 ga teng bo'ladi. ADD AX, 3: Bu qator, AX registrga 3 ni qo'shish amalini bajaradi. Ya'ni, AX registridagi qiymatga 3 ni qo'shib natijada AX registri 8 ga teng bo'ladi. Natijada, AX registri 8 qiymatini oladi. Shu bilan birga, shartsiz operatsiyalar yordamida arifmetik amallar bajariladi va operandlarga to'g'ridan-to'g'ri amalni o'tkazish imkonini beradi.

2. Shartli operatsiya misoli (jump ko'rsatmalari):

```
MOV AX, 10 ; AX registrga 10 qiymatini yuklash
CMP AX, 5 ; AX registri 5 ga teng bo'lganligini tekshirish
```

JG greater ; Agar AX 5 dan katta bo'lsa "greater" belgilangan joyga o'tish

JMP end ; Aks holda "end" belgilangan joyga o'tish

greater:

MOV BX, 1 ; BX registrga 1 qiymatini yuklash

end:

Natija: Agar AX registri 5 dan katta bo'lsa, "greater" belgilangan joyga o'tish va BX registri 1 qiymatini olish. Aks holda "end" belgilangan joyga o'tish. Bu misolda shartli operatsiyalar (CMP, JG, JMP) ishlatilgan: MOV AX, 10: Bu qator, AX registrga 10 qiymatini yuklash amalini bajaradi. Ya'ni, AX registri 10 ga teng bo'ladi. CMP AX, 5: Bu qator, AX registri bilan 5 ni solishtiradi (CMP), ya'ni 5 ga teng bo'lganligini tekshiradi. JG greater: Agar AX 5 dan katta bo'lsa (JG - "katta" ma'nosidagi jump), "greater" belgilangan joyga o'tish (jump) amalga oshiriladi. "greater" belgilangan joydan dastur izlanayotgan holda davom etadi. JMP end: Aks holda (yani, AX 5 dan katta emas), "end" belgilangan joyga o'tish amalga oshiriladi. "end" belgilangan joydan dastur izlanayotgan holda davom etadi. greater: "greater" belgilangan joyga o'tilganda, yani AX 5 dan katta bo'lganda, dastur bu qatordan boshlab davom etadi. MOV BX, 1: "greater" belgilangan joyga o'tilganda, BX registrga 1 qiymatini yuklash amalini bajaradi. Ya'ni, BX registri 1 ga teng bo'ladi. end: Dastur "end" belgilangan joyga o'tganda, ya'ni AX 5 dan katta bo'lmasa, dastur bu qatordan boshlab davom etadi. Natijada, agar AX registri 5 dan katta bo'lsa, dastur "greater" belgilangan joyga o'tadi va BX registri 1 qiymatini oladi. Aks holda, ya'ni AX registri 5 dan katta emas, dastur "end" belgilangan joyga o'tadi.

Assembler tilida shartsiz operatsiyalar odatiy arifmetik amallarni bajarish uchun foydalaniladi va operandlarga to'g'ridan-to'g'ri amal bajarish imkonini beradi. Shartli operatsiyalar esa muhim qarorlar olish, shartlarni tekshirish va dastur bajarishning belgilangan yo'ldan boshqasiga o'tishni ta'minlash uchun foydalaniladi. Shartli operatsiyalar, CMP (compare) ko'rsatmasi orqali muqobil operatorlarni (misol uchun, <, >, =) foydalanish bilan amalga oshiriladi va jump (o'tish) ko'rsatmalari shartlarning natijalariga qarab dastur bajarishni o'zgartiradi.

Xulosa. Mazkur maqolada assembler tilida shartli va shartsiz operatsiyalarni

bajarish, shuningdek, ulardan samarali foydalanish haqida so'z yuritiladi. Assembler tilida shartli va shartsiz operatsiyalarni bajarish orqali assembler tilida, dastur qismi yoziladi va bu qism shartli va shartsiz operatsiyalarni o'z ichiga oladi, masalan, if-else, while, for, switch; assembler dasturining boshida, to'plam va o'zgaruvchilar uchun joy ayriladi, bu o'zgaruvchilar va to'plamlar dastur davomida foydalaniladigan ma'lumotlarni saqlash uchun ishlatiladi; shartli operatsiyalarni bajarish uchun, assembler tilida shartlar hisoblanadi va natijada shartni bajarishga qaror qabul qilinadi; shartsiz operatsiyalar esa shartlar hisoblanmasdan to'g'ridan-to'g'ri bajariladi va o'zgaruvchilar ustida arifmetik amallar, bog'lanishlar va boshqa buyruqlar orqali ma'lumotlar ustida amallar bajarishni o'z ichiga oladi.

FOYDALANILGAN ADABIYOTLAR

1. Mikroprotessor (X.Yu.Abasxonova, M.B.Mirzayeva, S.S.Parsiyev)
2. Gerasimenko V.A. Защита информации в автоматизированных системах обработки данных kn. 1.-M.: Energoatomizdat. -1994.-400s.
3. Massivlarni tashkil etish (Gulomov S.S.)
4. Stamp Mark. Information security: principles and practice. USA, 2011.

Internet resurslari

6. <http://www.google.com>
7. <http://www.it-ebooks.info>
8. www.poe.com