Asemblersko programiranje u Intel 64 arhitekturi

Ivana Tanasijević

1 Sintaksa

Opšta sintaksa asemblera je takva da se čita linija po linija. Linije mogu biti prazne u kom slučaju se ignorišu. Linija takođe može biti direktiva (počinje simbolom .) ili instrukcija. Komentarima se smatra sav sadrzaj do kraja linije nakon simbola `#`. Linija takođe može sadržati samo komentar.

Definicija labele se sastoji od identifikatora iza koga stoji simbol :. Identifikator mora početi slovom ili `_` a može sadržati slova, brojeve i `_`. Labele zamenjuju adrese podataka i instrukcija. One se prilikom prevođenja programa prevode u memorijske adrese.

Direktive imaju posebno značenje:

- `.intel_syntax noprefix` - označava da se koristi Intel-ova sintaksa, dok se imena registara koriste bez prefiksa `%`
- `.data` - započinje sekciju inicijalizovanih podataka
- `.bss` - započinje sekciju neinicijalizovanih podataka
- `.text` - započinje sekciju koda
- `.asciz` - kreira se ASCII niska na čijem se kraju automatski navodi terminirajuća nula
- `.int` - kreira se jedan ili niz celih brojeva, članovi su razvojeni zapetom
- `.byte` - kreira se jedan ili niz bajtova
- `.word` - kreira se jedan ili niz slogova od dva bajta
- `.long` - kreira se jedan ili niz slogova dužine četiri bajta
- `.quad` - kreira se jedan ili niz slogova dužine osam bajtova
- `.global ime` - označava se labela ime kao globalna, čime se omogućava linkeru da poveže definisane simbole.

Jedna instrukcija se sastoji od koda instrukcije i operanda (jednog ili više) ukoliko instrukcija zahteva operande. Svaka instrukcija ima svoju simboličku oznaku. Opšti oblik instrukcije sa dva argumenta je

```
instr dest, src
```

Operandi mogu biti

- Registarski - navodi se registar
- Memorijski - navodi se adresa na kojoj se nalazi vrednost sa kojom radimo
- Neposredni - navodi se sama vrednost sa kojom radimo

Memorijsko adresiranje ima oblik $[B + S * I + D]$, gde je B bazna adresa, S je faktor, I je indeks, dok je D pomeraj. Izostavljanjem nekog od ovih elemenata dobijaju se različite kombinacije izračunavanja adrese operanda. Na primer, moguće je navesti samo baznu adresu $[B]$, ili baznu adresu i pomeraj $[B + D]$. Oblik $[B + S * I]$ je zgodan prilikom rada sa nizovima: u bazni registar se smesti adresa početka niza, a u indeksni registar se smesti tekući indeks (koji se ažurira u svakoj iteraciji). Za faktor se uzima veličina elementa niza.

U slučaju da instrukcija koja sadrži memorijski operand nema ni jedan registarski operand, tada se mora eksplicitno specificirati širina memorijskog operanda. Ovo se radi navođenjem jednog od prefiksa:

- byte ptr - za jednobajtni podatak
- word ptr - za dvobajtni podatak
- dword ptr - za četvorobajtni podatak
- qword ptr - za osmobajtni podatak

Ovo nije neophodno ako registarski operand postoji u instrukciji, zato što onda njegova širina implicitno određuje širinu operanada koji se koriste.

2 Registri opšte namene

Intel 64 arhitektura ima 16 64-bitnih registara opšte namene: RAX, RBX, RCX, RDX, RSI, RDI, RSP, RBP, kao i registre R8-R15. Nizim delovima registara se može pristupiti preko oznaka EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP, uz dodatak registara R8D-R15D.

Ovi registri su generalno opšte namene, mada u praksi neki imaju specijalne uloge. Tako, na primer, RSP se uvek koristi kao pokazivač vrha steka, RBP je pokazivač tekućeg okvira steka. Ostali registri u nekim situacijama mogu imati specijalne uloge (npr. pri množenju i deljenju se uvek implicitno koriste registri RAX i RDX, dok se pri radu sa stringovima za pokazivače uvek koriste registri RSI i RDI). Za brojačke petlje se uglavnom koristi registar RCX, pošto određene instrukcije kontrole toka implicitno koriste i menjaju ovaj registar.

2.1 Instrukcije za rad sa registrima opšte namene

2.1.1 Instrukcije transfera

- MOV op1, op2 - premeštanje (op1 = op2)
- MOVZX op1, op2 - u prvi operand se smešta vrednost drugog operanda proširena nulama
- MOVSX op1, op2 - u prvi operand se smešta vrednost drugog operanda proširena u skladu sa znakom
- LEA op1, op2 - učitavanje adrese drugog operanda u prvi operand, pri čemu je drugi operand memorijski operand.

Prilikom premeštanja vrednosti, ukoliko nije navedeno drugačije, premešta se vrednost koja je veličine registra koji učestvuje u operaciji. Što znači da ako se kao operand navede memorijska lokacija, onda se radi sa veličinom koju određuje registar.

2.1.2 Aritmetičke instrukcije

- ADD op1, op2 - sabiranje (op1 = op1 + op2)
- SUB op1, op2 - oduzimanje (op1 = op1 - op2)
- CDQE - označeno proširivanje eax na rax
- CDQ - označeno proširivanje eax na edx:eax
- CQO - označeno proširivanje rax na rdx:rax
- MUL op - množenje neoznačenih celih brojeva (rdx:rax = rax * op)
- IMUL op - množenje označenih celih brojeva (rdx:rax = rax * op)
- DIV op - deljenje neoznačenih celih brojeva (rax = rdx:rax / op, rdx = rdx:rax % op)
- IDIV op - deljenje označenih celih brojeva (rax = rdx:rax / op, rdx = rdx:rax % op)
- NEG op - promena znaka (op = -op)
- INC op - uvećanje (op = op + 1)
- DEC op - umanjenje (op = op - 1)

Oznaka rdx:rax znači 128-bitni ceo broj čiji su viši bitovi u rdx a niži u rax.

2.1.3 Logičke instrukcije

- AND op1, op2 - bitovska logička konjukcija (op1 = op1 & op2)
- OR op1, op2 - bitovska logička disjunkcija (op1 = op1 | op2)
- XOR op1, op2 - bitovska logička ekskluzivna disjunkcija (op1 = op1 ^ op2)
- NOT op - bitovska negacija (op = ~op)
- SHL op1, op2 - shift-ovanje ulevo (op1 = op1 << op2). op2 je konstanta.
- SHR op1, op2 - shift-ovanje udesno (logičko) (op1 = op1 >> op2). op2 je konstanta.
- SAR op1, op2 - shift-ovanje udesno (aritmetičko) (op1 = op1 >> op2). op2 je konstanta.

2.1.4 Instrukcije poređenja

- CMP - upoređivanje (oduzimanje bez upisivanja rezultata)
- TEST - testiranje bitova (bitovska konjukcija bez upisivanja rezultata)

2.1.5 Instrukcije za rad sa stekom

- PUSH op - postavljanje na stek
- POP op - skidanje sa steka

2.1.6 Instrukcije kontrole toka

- JMP op - bezuslovni skok na adresu op (memorijski operand)
- CALL op - bezuslovni skok uz pamćenje povratne adrese na steku.
- RET - skida sa steka adresu i skače na tu adresu.
- JZ op - skače ako je rezultat prethodne instrukcije nula.
- JE op - skače ako je rezultat prethodnog poređenja jednakost (ekvivalentno sa JZ)
- JNZ op - skače ako je rezultat prethodne operacije različit od nule
- JNE op - skače ako je rezultat prethodnog poređenja različitost (ekvivalentno sa JNZ)
- JA op - skače ako je rezultat prethodnog poređenja veće (neoznačeni brojevi)
- JB op - skače ako je rezultat prethodnog poređenja manje (neoznačeni brojevi)
- JAE op - skače ako je rezultat prethodnog poređenja vece ili jednako (neoznačeni brojevi)
- JBE op - skače ako je rezultat prethodnog poređenja manje ili jednako (neoznačeni brojevi)
- JG op - skače ako je rezultat prethodnog poređenja veće (označeni brojevi)
- JL op - skače ako je rezultat prethodnog poređenja manje (označeni brojevi)
- JGE op - skače ako je rezultat prethodnog poređenja veće ili jednako (označeni brojevi)
- JLE op - skače ako je rezultat prethodnog poređenja manje ili jednako (označeni brojevi)

Slično, postoje i negacije gornjih instrukcija uslovnog skoka: JNA, JNB, JNAE, JNBE, JNG, JNL, JNGE, JNLE.

2.2 Konvencije za pozivanje funkcija

Instrukcija kojom se poziva funkcija je

```
call naziv
```

Celobrojni parametri funkcija, uključujući i adrese, prenose se, redom, preko registara rdi, rsi, rdx, rcx, r8, r9. Ukoliko ima više od šest parametara, ostali se smeštaju na stek u obrnutom redosledu - s desna na levo. Povratna vrednost funkcije se nalazi u rax registru.

Prilikom poziva funkcije, moguće je da će ona izmeniti neke registre. Registri koji pripadaju pozivajućoj funkciji su rbx, rbp, r12-15. Sadržaj ovih registara se mora sačuvati ukoliko se menja u funkciji. Ostali registri se mogu menjati bez čuvanja. Registri koji pripadaju pozvanoj funkciji su rax, rdi, rsi, rdx, rcx, r8-r11. Na sadržaje ovih registara ne može računati pozivajuća funkcija. Ukoliko je korišćen stek prilikom poziva funkcije, neophodno je da vrh steka bude poravnat sa adresom deljivom sa 16.

Na početku svake funkcije se nalazi prolog u kome je potrebno izvršiti sledeće instrukcije

```
push rbp
mov rbp, rsp
sub rsp, N
```

Alternativa je

```
ENTER N, 0
```

gde N označava broj bajtova koji odvajamo za lokalne promenljive.
Na kraju svake funkcije se nalazi epilog u kome se izvršavaju instrukcije

```
mov rsp, rbp
pop rbp
```

Alternativa je

```
LEAVE
```

Povratak iz funkcije se izvršava instrukcijom

```
RET
```

Ova instrukcija skida povratnu adresu sa steka i prelazi na izvršavanje instrukcije sa te adrese.

2.3 Prevođenje

Izvorni kod sa asemblerskim funkcijama 1.s se prevodi na sledeći način:

- gcc 1.s

Moguće je prevoditi kod iz više izvornih datoteka navodeći ih u jednoj komandi:

- gcc 1.c 1.s

3 Rad sa realnim brojevima

Instrukcije koje rade paralelno nad više podataka se nazivaju SIMD (single instruction multiple data). Primeri takvih instrukcija su SSE i SSE2. Ove instrukcije koriste xmm registre za rad sa realnim brojevima. Na raspolaganju je 16 128-bitnih registara xmm0 do xmm15 za smeštanje realnih brojeva. SSE instrukcije koriste ove registre za rad sa realnim brojevima jednostruke preciznosti (32-bitni podatak), dok SSE2 instrukcije rade sa brojevima dvostruke preciznosti (64-bitni podatak). U svaki od registara mogu stati po četiri, donosno dva, ovakva broja, što se koristi za izvršavanje instrukcije paralelno nad ovim podacima. Xmm registri se mogu koristiti samo za izračunavanja, ali ne i za adresiranje. Za to se koriste registri opšte namene. Podaci mogu da se smeste u registre ili iz registara u memoriju sa 32-bitnim, 64-bitnim ili 128-bitnim inkrementom. Adresa 128-bitno pakovanog operanda iz memorije mora da bude poravnata sa 16, osim kada se koristi instrukcija MOVUPS koja podržava neporavnate operande ili kod skalarnih instrukcija koje koriste 4-bajtne (kod SSE instrukcija), odnosno 8-bajtne (kod SSE2 instrukcija), memorijske operande.

3.1 Konvencije za pozivanje funkcija

Realni parametri se prenose preko xmm registara, dok se celi brojevi, kao i pokazivači (bilo kog tipa, int *, double *, float *, ...) prenose preko registara opšte namene kao što je prethodno opisano.

3.2 Korišćenje xmm registara

Pre rada sa SSE instrukcijama proverava se da li procesor podržava ove instrukcije. To se može uraditi na sledeći način:

```
mov     rax, 1
cpuid
test    rdx, 0x2000000
jz not_supported
```

gde se nakon labele `not_supported` navode instrukcije za završetak programa.
Zatim se čuva sadržaj SSE registara i registara koprocesora:

```
mov     rdx, rsp
and     rsp, 0xfffffffffffffff0
sub     rsp, 512
fxsave  [rsp]
```

Na kraju rada sa ovim registrima se vraća njihov prvobitni sadržaj:

```
fxrstor [rsp]
mov     rsp, rdx
```

- FXSAVE - smešta registre numeričkog koprocesora i sadržaje xmm registara u memoriju
- FXRSTOR - upisuje iz memorije u registre numeričkog koprocesora i u xmm registre sačuvane vrednost

3.3 SSE Instrukcije

3.3.1 Instrukcije transfera

Ove instrukcije vrše transfer između dva xmm registra ili xmm registra i memorije.

- MOVAPS - smešta poravnata pakovana 4 realna broja jednostruke preciznosti u lokaciju zadatu prvim operandom
- MOVUPS - smešta neporavnata pakovana 4 realna broja jednostruke preciznosti u odredišni operand
- MOVSS - smešta jedan realan broj jednostruke preciznosti u odredišni operand
- MOVLPS - smešta dva realna broja sa niže adrese u memoriju ili iz memorije u niži deo registra, preostali deo registra ostaje nepromenjen
- MOVHPS - smešta dva realna broja sa više adrese u memoriju ili iz memorije u viši deo registra, preostali deo registra ostaje nepromenjen
- MOVLHPS - smešta niži deo registra u viši deo, niži deo ostaje nepromenjen
- MOVHLPS - smešta viši deo registra u niži deo, viši deo ostaje nepromenjen
- MOVMSKPS - smešta po jedan najviši bit sva četiri podatka u registre opšte namene

3.3.2 Aritmetičke instrukcije

- ADDPS - sabira paralelno po četiri podatka
- SUBPS - oduzima paraleleno po četiri podatka
- ADDSS - sabira samo podatak na najnižoj adresi
- SUBSS - oduzima samo podatak na najnižoj adresi
- MULPS - množi paralelno četiri podatka
- MULSS - množi samo podatke na najnižoj adresi
- DIVPS - deli paralelno četiri podatka
- DIVSS - deli samo podatke na najnižoj adresi
- RCPPS - računa paralelno recipročne vrednosti podataka
- RCPSS - računa recipročnu vrednost podatka na najnižoj adresi
- SQRTPS - računa paralelno kvadratni koren
- SQRTSS - računa kvadratni koren podatka na najnižoj adresi
- RSQRTPS - računa recipročnu vrednost kvadratnog korena paralelno
- RSQRTPS - računa recipročnu vrednost kvadratnog korena podatka na najnižoj adresi

- MAXPS - računa maksimume paralelno
- MAXSS - računa maksimum podatka na najnižoj adresi
- MINPS - računa minimume paralelno
- MINSS - računa minumume podatka na najnižoj adresi

3.3.3 Logičke instrukcije

- ANDPS - logička konjunkcija
- ANDNPS - logička negacija konjunkcije
- ORPS - logička disjunkcija
- XORPS - logička ekskluzivna disjunkcija

3.3.4 Instrukcije poređenja

- CMPPS - poredi prema trećem argumentu instrukcije, popunjava svim jedinicama ako su podaci jednaki, u suprotnom popunjava svim nulama
- CMPSS - poredi samo podatak na najnižoj adresi prema trećem argumentu instrukcije, popunjava svim jedinicama ako su podaci jednaki, u suprotnom popunjava svim nulama
- COMISS - poredi samo podatak na najnižoj adresi i postavlja EFLAGS registar, tako da se mogu koristiti instrukcije skoka za neoznačene operande, signalizira izuzetak ako je operand NaN

3.3.5 Instrukcije šiftovanja

- SHUFPS - smešta bilo koja dva od četiri podatka u niži deo odredišnog operanda i bilo koja dva podatka u viši deo odredišnog operanda. Ukoliko su oba operanda isti registar smešta podatke u bilo kom redosledu
- UNPCKHPS - kombinuje po dva podatka sa viših adresa
- UNPCKLPS - kombinuje po dva podatka sa nižih adresa

3.3.6 Instrukcije za konverziju

- CVTPI2PS - konvertuje cele brojeve u jednostruku preciznost paralelno
- CVTSI2SS - konvertuje ceo broj u jednostruku preciznost
- CVTPS2PI - konvertuje jednostruku preciznost u cele brojeve paralelno
- CVTSS2SI - konvertuje jednostruku preciznost u ceo broj

3.4 SSE2 instrukcije

3.4.1 Instrukcije transfera

Vrše transfer između dva xmm registra ili xmm registra i memorije.

- MOVAPD - smešta poravnata pakovana dva realna broja dvostruke preciznosti u lokaciju zadatu prvim operandom
- MOVUPD - smešta neporavnata pakovana dva realna broja dvostruke preciznosti
- MOVLPD - smešta realni broj sa niže adrese u memoriju ili iz memorije u niži deo registra, preostali deo registra ostaje nepromenjen
- MOVHPD - smešta realni broj sa više adrese u memoriju ili iz memorije u viši deo registra, preostali deo registra ostaje nepromenjen
- MOVMSKPS - smešta po jedan najviši bit oba podatka u registre opšte namene
- MOVSD - smešta jedan realni broj dvostruke preciznosti u xmm registar ili u memoriju

3.4.2 Aritmetičke instrukcije

- ADDPD - sabira paralelno po dva podatka
- SUBPD - oduzima paraleleno po dva podatka
- ADDSD - sabira samo podatke na nižoj adresi
- SUBSD - oduzima samo podatke na nižoj adresi
- MULPD - množi paralelno dva podatka
- MULSD - množi samo podatke na nižoj adresi
- DIVPD - deli paralelno dva podatka
- DIVSD - deli samo podatke na nižoj adresi
- SQRTPD - računa paralelno kvadratni koren
- SQRTSD - računa kvadratni koren podatka na nižoj adresi
- MAXPD - računa maksimume paralelno
- MAXSD - računa maksimum podataka na nižoj adresi
- MINPD - računa minimume paralelno
- MINSD - računa minumume podataka na nižoj adresi

3.4.3 Logičke instrukcije

- ANDPD - logička konjunkcija
- ANDNPD - logička negacija konjunkcije
- ORPSD- logička disjunkcija
- XORPD - logička ekskluzivna disjunkcija

3.4.4 Instrukcije poređenja

- CMPPD - poredi prema trećem argumentu instrukcije, popunjava svim jedinicama ako su podaci jednaki, u suprotnom popunjava svim nulama
- CMPSD - poredi samo podatak na najnižoj adresi prema trećem argumentu instrukcije, popunjava svim jedinicama ako su podaci jednaki, u suprotnom popunjava svim nulama
- COMISD - poredi samo podatak na najnižoj adresi i postavlja EFLAGS registar, tako da se mogu koristiti instrukcije skoka za neoznačene operande, signalizira izuzetak ako je operand NaN

3.4.5 Instrukcije šiftovanja

- SHUFPD - smešta bilo koji od dva podatka u niži deo odredisnog operanda i bilo koji od dva podatka u viši deo odredišnog operanda. Ukoliko su oba operanda isti registar smešta podatke u bilo kom redosledu
- UNPCKHPD - kombinuje podatke sa viših adresa
- UNPCKLPD - kombinuje podatke sa nižih adresa

3.4.6 Instrukcije za konverziju

- CVTPI2PD - konvertuje cele brojeve u dvostruku preciznost paralelno
- CVTSI2SD - konvertuje ceo broj u dvostruku preciznost
- CVTPD2PI - konvertuje dvostruku preciznost u cele brojeve paralelno
- CVTSD2SI - konvertuje dvostruku preciznost u ceo broj
- CVTPS2PD - konvertuje jednostruku preciznost u dvostruku preciznost
- CVTPD2PS - konvertuje dvostruku preciznost u jednostruku preciznost