

Asembler x86_64 – instrukcije opšte namene

— seminarski rad —

Arhitektura i operativni sistemi

Ljubica Peleksić i Milica Kojičić

November 5, 2013

Asemblerski jezik

Asemblerski jezik

- Svaki računar ima svoju arhitekturu skupa instrukcija (Instruction Set Architecture, ISA) koja se naziva mašinski jezik

Asemblerski jezik

- Svaki računar ima svoju arhitekturu skupa instrukcija (Instruction Set Architecture, ISA) koja se naziva mašinski jezik
- Asemblerski jezik je simbolička predstava te arhitekture

Asemblerski jezik

- Svaki računar ima svoju arhitekturu skupa instrukcija (Instruction Set Architecture, ISA) koja se naziva mašinski jezik
- Asemblerski jezik je simbolička predstava te arhitekture
- Binarni brojevi iz mašinskog jezika su sada zamenjeni mnemoničkim oznakama

Asemblerski jezik

- Svaki računar ima svoju arhitekturu skupa instrukcija (Instruction Set Architecture, ISA) koja se naziva mašinski jezik
- Asemblerski jezik je simbolička predstava te arhitekture
- Binarni brojevi iz mašinskog jezika su sada zamenjeni mnemoničkim oznakama
- Osim kratkih reči (ADD, SUB, MUL) koje se koriste za mašinske instrukcije, asembleri dozvoljavaju korišćenje simboličkih imena za konstante i oznaka koje se odnose na instrukcije i memorijske adrese

Zašto assembler?

Zašto assembler?

”Assembler je program programa, alat svih alata i on je ozbiljno oružje u rukama pravog programera”

Zašto assembler?

”Assembler je program programa, alat svih alata i on je ozbiljno oružje u rukama pravog programera”

- Najbliža forma komunikacije između čoveka i mašine

Zašto assembler?

”Assembler je program programa, alat svih alata i on je ozbiljno oružje u rukama pravog programera”

- Najbliža forma komunikacije između čoveka i mašine
- Prepravljanje koda za koji nemamo izvorni kod

Zašto assembler?

”Assembler je program programa, alat svih alata i on je ozbiljno oružje u rukama pravog programera”

- Najbliža forma komunikacije između čoveka i mašine
- Prepravljanje koda za koji nemamo izvorni kod
- Pisanje koda kritičnih performansi

Zašto assembler?

”Assembler je program programa, alat svih alata i on je ozbiljno oružje u rukama pravog programera”

- Najbliža forma komunikacije između čoveka i mašine
- Prepravljanje koda za koji nemamo izvorni kod
- Pisanje koda kritičnih performansi
- Pristupanje novim karakteristikama, pre nego što ih uvedu u kompajler

Arhitektura x86-64

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:
 - ima veće količine virtuelne i fizičke memorije

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:
 - ima veće količine virtuelne i fizičke memorije
 - ima 64-bitne registre

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:
 - ima veće količine virtuelne i fizičke memorije
 - ima 64-bitne registre
 - potpuno je unazad kompatibilna sa 16-bitnom i 32-bitnom arhitekturom

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:
 - ima veće količine virtuelne i fizičke memorije
 - ima 64-bitne registre
 - potpuno je unazad kompatibilna sa 16-bitnom i 32-bitnom arhitekturom
 - zadržava set instrukcija skoro u potpunosti

Arhitektura x86-64

- 64-bitni računari zamenjuju 32-bitne
- x64-ime za Intelovu i AMD-ovu nadogradnju 32-bitne arhitekture skupa instrukcija
- AMD-ova prva verzija x64 se zvala x86-64, a zatim su je preimenovali u AMD64
- Intel je svoju implementaciju nazvao IA-32e, a zatim EMT64
- Ova arhitektura:
 - ima veće količine virtuelne i fizičke memorije
 - ima 64-bitne registre
 - potpuno je unazad kompatibilna sa 16-bitnom i 32-bitnom arhitekturom
 - zadržava set instrukcija skoro u potpunosti
 - ima slične načine adresiranja

Organizacija registara

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita
- U 64-bitnoj arhitekturi se i dalje koriste 32-bitni, 16-bitni i 8-bitni registri

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita
- U 64-bitnoj arhitekturi se i dalje koriste 32-bitni, 16-bitni i 8-bitni registri
- U x86 assembleru postoji 8 primarnih registara: RAX(akumulator), RCX(brojac), RDX(registar podataka), RBX(bazni registar), RSP(pokazivac na stek), RBP(pokazivac na narednu instrukciju) RSI i RDI.

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita
- U 64-bitnoj arhitekturi se i dalje koriste 32-bitni, 16-bitni i 8-bitni registri
- U x86 assembleru postoji 8 primarnih registara: RAX(akumulator), RCX(brojac), RDX(registar podataka), RBX(bazni registar), RSP(pokazivac na stek), RBP(pokazivac na narednu instrukciju) RSI i RDI.
- Floating-point registri

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita
- U 64-bitnoj arhitekturi se i dalje koriste 32-bitni, 16-bitni i 8-bitni registri
- U x86 assembleru postoji 8 primarnih registara: RAX(akumulator), RCX(brojac), RDX(registar podataka), RBX(bazni registar), RSP(pokazivac na stek), RBP(pokazivac na narednu instrukciju) RSI i RDI.
- Floating-point registri
- Registar RFLAGS je proširenje EFLAGS i kao i on, skладиšti flagove, a sadržaj se menja u zavisnosti od rezultata aritmetičkih operacija.

Organizacija registara

- Registri se u odnosu na one iz 32-bitnog asemblera proširuju sa 32 na 64 bita
- U 64-bitnoj arhitekturi se i dalje koriste 32-bitni, 16-bitni i 8-bitni registri
- U x86 assembleru postoji 8 primarnih registara: RAX(akumulator), RCX(brojac), RDX(registar podataka), RBX(bazni registar), RSP(pokazivac na stek), RBP(pokazivac na narednu instrukciju) RSI i RDI.
- Floating-point registri
- Registar RFLAGS je proširenje EFLAGS i kao i on, skладиšti flagove, a sadržaj se menja u zavisnosti od rezultata aritmetičkih operacija.

Načini adresiranja

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka
 - Indirektno registarsko-adresa operanda nalazi u nekom od registara RBX, RSI ili RDI.

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka
 - Indirektno registarsko-adresa operanda nalazi u nekom od registara RBX, RSI ili RDI.
 - Registarsko adresiranje sa pomerajem-konstanta se postavi ispred registra, adresa se pronalazi dodavanjem sadržaja registra konstanti.

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka
 - Indirektno registarsko-adresa operanda nalazi u nekom od registara RBX, RSI ili RDI.
 - Registarsko adresiranje sa pomerajem-konstanta se postavi ispred registra, adresa se pronalazi dodavanjem sadržaja registra konstanti.
 - Registarsko indeksno podrazumeva da je u RDI početna adresa niza, a brojač u RCX.

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka
 - Indirektno registarsko-adresa operanda nalazi u nekom od registara RBX, RSI ili RDI.
 - Registarsko adresiranje sa pomerajem-konstanta se postavi ispred registra, adresa se pronalazi dodavanjem sadržaja registra konstanti.
 - Registarsko indeksno podrazumeva da je u RDI početna adresa niza, a brojač u RCX.
 - Registarsko indeksno sa pomerajem-kombinacija prethodna dva

Načini adresiranja

- Načini adresiranja su zapravo načini na koje instrukcija može da pristupi registrima ili memoriji
- Načini adresiranja:
 - Neposredno adresiranje-operand u instrukciji je konstantan bajt ili reč
 - Direktno adresiranje-operand u instrukciji je adresa podatka
 - Indirektno registarsko-adresa operanda nalazi u nekom od registara RBX, RSI ili RDI.
 - Registarsko adresiranje sa pomerajem-konstanta se postavi ispred registra, adresa se pronalazi dodavanjem sadržaja registra konstanti.
 - Registarsko indeksno podrazumeva da je u RDI početna adresa niza, a brojač u RCX.
 - Registarsko indeksno sa pomerajem-kombinacija prethodna dva

Aritmetičke instrukcije

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:
ADC RAX, 4

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:
ADC RAX, 4
- Instrukcija SBB vrši oduzimanje s pozajmicom i ima dva operanda:

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:
ADC RAX, 4
- Instrukcija SBB vrši oduzimanje s pozajmicom i ima dva operanda:
SBB RBX, x

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:
ADC RAX, 4
- Instrukcija SBB vrši oduzimanje s pozajmicom i ima dva operanda:
SBB RBX, x
- Instrukcija NEG vrši negaciju i ima jedan operand kome menja znak.

Aritmetičke instrukcije

- Instrukcija ADD vrši sabiranje i ima dva operanda:
ADD RAX, 4
- Instrukcija SUB vrši oduzimanje i ima dva operanda:
SUB RBX, x
- Instrukcija ADC vrši sabiranje s prenosom i ima dva operanda:
ADC RAX, 4
- Instrukcija SBB vrši oduzimanje s pozajmicom i ima dva operanda:
SBB RBX, x
- Instrukcija NEG vrši negaciju i ima jedan operand kome menja znak.
NEG X

Aritmetičke instrukcije

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)
DEC RDX (smanjuje vrednost registra RDX za 1)

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)
DEC RDX (smanjuje vrednost registra RDX za 1)
- Razlikujemo dve instrukcije za množenje: MUL za neoznačene i IMUL za označene brojeve. Instrukcija MUL ima samo jedan operand a IMUL može imati jedan operand i u tom slučaju je analogna instrukciji MUL, ali može imati i dva i tri operanda.

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)
DEC RDX (smanjuje vrednost registra RDX za 1)
- Razlikujemo dve instrukcije za množenje: MUL za neoznačene i IMUL za označene brojeve. Instrukcija MUL ima samo jedan operand a IMUL može imati jedan operand i u tom slučaju je analogna instrukciji MUL, ali može imati i dva i tri operanda.
MUL RBX

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)
DEC RDX (smanjuje vrednost registra RDX za 1)
- Razlikujemo dve instrukcije za množenje: MUL za neoznačene i IMUL za označene brojeve. Instrukcija MUL ima samo jedan operand a IMUL može imati jedan operand i u tom slučaju je analogna instrukciji MUL, ali može imati i dva i tri operanda.
MUL RBX
- Razlikujemo dve instrukcije za deljenje: DIV se odnosi na deljenje neoznačenih, a IDIV na deljenje označenih brojeva; Instrukcija DIV ima jedan operand.IDIV je analogna instrukciji DIV.

Aritmetičke instrukcije

- Instrukcije INC i DEC vrše (redom) uvecavanje i smanjivanje argumenta za jedan i imaju po jedan operand.
INC RDX (uvećava vrednost registra RDX za 1)
DEC RDX (smanjuje vrednost registra RDX za 1)
- Razlikujemo dve instrukcije za množenje: MUL za neoznačene i IMUL za označene brojeve. Instrukcija MUL ima samo jedan operand a IMUL može imati jedan operand i u tom slučaju je analogna instrukciji MUL, ali može imati i dva i tri operanda.
MUL RBX
- Rezlikujemo dve instrukcije za deljenje: DIV se odnosi na deljenje neoznačenih, a IDIV na deljenje označenih brojeva; Instrukcija DIV ima jedan operand.IDIV je analogna instrukciji DIV.
DIV RCX

Logičke instrukcije

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:
OR RDX, RCX

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:
OR RDX, RCX
- Instrukcija XOR vrši bitovsku ekskluzivnu disjunkciju i ima dva operanda:

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:
OR RDX, RCX
- Instrukcija XOR vrši bitovsku ekskluzivnu disjunkciju i ima dva operanda:
XOR RDX, RCX

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:
OR RDX, RCX
- Instrukcija XOR vrši bitovsku ekskluzivnu disjunkciju i ima dva operanda:
XOR RDX, RCX
- Instrukcija NOT vrši bitovsku negaciju i ima jedan operand:

Logičke instrukcije

- Instrukcija AND vrši bitovsku konjukciju i ima dva operanda:
AND RDX, RCX
- Instrukcija OR vrši bitovsku disjunkciju i ima dva operanda:
OR RDX, RCX
- Instrukcija XOR vrši bitovsku ekskluzivnu disjunkciju i ima dva operanda:
XOR RDX, RCX
- Instrukcija NOT vrši bitovsku negaciju i ima jedan operand:
NOT RDX

Instrukcije za šiftovanje i rotaciju

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan argument

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan argument
- Instrukcije za šiftovanje sa prenosom
 - Instrukcija SCR vrši šiftovanje udesno sa prenosom i ima jedan argument

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan argument
- Instrukcije za šiftovanje sa prenosom
 - Instrukcija SCR vrši šiftovanje udesno sa prenosom i ima jedan argument
 - Instrukcija SCL vrši šiftovanje ulevo sa prenosom i ima jedan argument

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan argument
- Instrukcije za šiftovanje sa prenosom
 - Instrukcija SCR vrši šiftovanje udesno sa prenosom i ima jedan argument
 - Instrukcija SCL vrši šiftovanje ulevo sa prenosom i ima jedan argument
- Instrukcije za rotaciju
 - Instrukcija ROR vrši rotaciju udesno i ima jedan operand

Instrukcije za šiftovanje i rotaciju

- Logičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan operand
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan operand
- Aritmetičko šiftovanje
 - Instrukcija SHR vrši šiftovanje udesno i ima jedan argument
 - Instrukcija SHL vrši šiftovanje ulevo i ima jedan argument
- Instrukcije za šiftovanje sa prenosom
 - Instrukcija SCR vrši šiftovanje udesno sa prenosom i ima jedan argument
 - Instrukcija SCL vrši šiftovanje ulevo sa prenosom i ima jedan argument
- Instrukcije za rotaciju
 - Instrukcija ROR vrši rotaciju udesno i ima jedan operand
 - Instrukcija ROL vrši rotaciju ulevo i ima jedan operand

Instrukcije za transfer podataka

Instrukcije za transfer podataka

- Instrukcija MOV kopira vrednost drugog operanda u prvi pri čemu se nijedan fleg ne menja

Instrukcije za transfer podataka

- Instrukcija MOV kopira vrednost drugog operanda u prvi pri čemu se nijedan fleg ne menja
- Instrukcija XCHG zamenjuje vrednosti prvog i drugog operanda, pri čemu se takodje nijedan fleg ne menja

Instrukcije za transfer podataka

- Instrukcija MOV kopira vrednost drugog operanda u prvi pri čemu se nijedan fleg ne menja
- Instrukcija XCHG zamenjuje vrednosti prvog i drugog operanda, pri čemu se takodje nijedan fleg ne menja
- Instrukcije pri kopiranju drugog operanda u prvi nepopunjena mesta u prvom argumentu popunjavaju nulama ili znakom drugog argumenta:

Instrukcije za transfer podataka

- Instrukcija MOV kopira vrednost drugog operanda u prvi pri čemu se nijedan fleg ne menja
- Instrukcija XCHG zamenjuje vrednosti prvog i drugog operanda, pri čemu se takodje nijedan fleg ne menja
- Instrukcije pri kopiranju drugog operanda u prvi nepopunjena mesta u prvom argumentu popunjavaju nulama ili znakom drugog argumenta:
- Instrukcija MOVSB nema nijedan operand i kopira jedan bajt sa lokacije koja se nalazi u RSI registru na lokaciju koja se nalazi u RDI. Instrukcija takodje ne menja flegove
- MOVESW je instrukcija koja nema nijedan operand i kopira dva bajta (jednu reč) iz RSI u RDI

Instrukcije za kontrolu toka

Instrukcije za kontrolu toka

- Instrukcije za poredjenje

Instrukcije za kontrolu toka

- Instrukcije za poredjenje
 - TEST je instrukcija koja izvršava bitovski konjukciju nad dva operanda i postavlja flegove, ali ne čuva rezultat

Instrukcije za kontrolu toka

- Instrukcije za poredjenje
 - TEST je instrukcija koja izvršava bitovski konjukciju nad dva operanda i postavlja flegove, ali ne čuva rezultat
 - CMP je instrukcija koja vrši oduzimanje izmedju dva operanda, postavlja flegove, ali ne čuva rezultat

Instrukcije za kontrolu toka

- Instrukcije za poredjenje
 - TEST je instrukcija koja izvršava bitovski konjukciju nad dva operanda i postavlja flegove, ali ne čuva rezultat
 - CMP je instrukcija koja vrši oduzimanje izmedju dva operanda, postavlja flegove, ali ne čuva rezultat
- Instrukcije za poziv procedura

Instrukcije za kontrolu toka

- Instrukcije za poredjenje
 - TEST je instrukcija koja izvršava bitovski konjukciju nad dva operanda i postavlja flegove, ali ne čuva rezultat
 - CMP je instrukcija koja vrši oduzimanje izmedju dva operanda, postavlja flegove, ali ne čuva rezultat
- Instrukcije za poziv procedura
 - Instrukcija CALL bezuslovno skače na adresu koja je zadata kao operand, ali prethodno stavlja na vrh steka adresu povratka iz RIP registra

Instrukcije za kontrolu toka

- Instrukcije za poredjenje
 - TEST je instrukcija koja izvršava bitovski konjukciju nad dva operanda i postavlja flegove, ali ne čuva rezultat
 - CMP je instrukcija koja vrši oduzimanje izmedju dva operanda, postavlja flegove, ali ne čuva rezultat
- Instrukcije za poziv procedura
 - Instrukcija CALL bezuslovno skače na adresu koja je zadata kao operand, ali prethodno stavlja na vrh steka adresu povratka iz RIP registra
 - Instrukcija RET završava potrogram, ona skida sa steka povratnu adresu, stavlja je u RIP registar i vrši bezuslovan skok na tu adresu.

Instrukcije za kontrolu toka

Skokovi

- Bezuslovni skok - JMP
- Skok ako je jednako - JE
- Skok ako nije jednako - JNE
- Skok ako je veće - JG,JA i JGE,JAЕ
- Skok ako je manje - JL,JB i JLE,JBE
- Skok ako postoji prekoračenje - JO
- Skok u slučaju nule - JNZ i JZ

Instrukcije za kontrolu toka

- Instrukcije brojačke petlje

Instrukcije za kontrolu toka

- Instrukcije brojačke petlje
 - Instrukcija LOOP ima samo jedan argument koji predstavlja adresu memorijske lokacije. Ona umanjuje vrednost registra RCX za jedan i skače na zadatu adresu sve dok vrednost registra RAX ne bude nula. LOOP ne postavlja flegove

Instrukcije za kontrolu toka

- Instrukcije brojačke petlje
 - Instrukcija LOOP ima samo jedan argument koji predstavlja adresu memorijske lokacije. Ona umanjuje vrednost registra RCX za jedan i skače na zadatu adresu sve dok vrednost registra RAX ne bude nula. LOOP ne postavlja flegove
 - Instrukcija LOOPX umanjuje vrednost registra RCX i skače na adresu zadatu operandom ako je njen uslov zadovoljen, tj. ako je postavljen odgovarajući fleg:
 - LOOPE skače ako je jednako
 - LOOPZ skače ako je nula
 - LOOPNE skače ako nije jednako
 - LOOPNZ skače ako nije nula

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir
- Druge instrukcije kontrole

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir
- Druge instrukcije kontrole
 - HLT instrukcija zaustavlja procesor

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir
- Druge instrukcije kontrole
 - HLT instrukcija zaustavlja procesor
 - NOP znači da nema operacije. Ova instrukcija ne radi nista, tj. samo troši instrukcioni ciklus

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir
- Druge instrukcije kontrole
 - HLT instrukcija zaustavlja procesor
 - NOP znači da nema operacije. Ova instrukcija ne radi nista, tj. samo troši instrukcioni ciklus
 - LOCK instrukcija ističe procesorov signal LOCK koji je istaknut tokom operacije koja ide uz instrukciju

Instrukcije za kontrolu toka

- Instrukcije ENTER i LEAVE
 - Instrukcija ENTER pravi stek okvir sa odredjenom količinom prostora alociranom na steku. Obično se taj prostor rezerviše za lokalne promenljive koje se koriste u procedurama
 - Instrukcija LEAVE uništava trenutni stek okvir i restaurira prethodni okvir
- Druge instrukcije kontrole
 - HLT instrukcija zaustavlja procesor
 - NOP znači da nema operacije. Ova instrukcija ne radi nista, tj. samo troši instrukcioni ciklus
 - LOCK instrukcija ističe procesorov signal LOCK koji je istaknut tokom operacije koja ide uz instrukciju
 - WAIT instrukcija ceka da procesor završi svoju poslednju operaciju

Instrukcije za rad sa stekom

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:
PUSH EAX

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:
PUSH EAX
- Instrukcija POP stavlja podatke koji se nalaze na lokaciji na koju pokazuje pokazivač na stek u argument, a zatim povećava pokazivač na stek za 1:

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:
PUSH EAX
- Instrukcija POP stavlja podatke koji se nalaze na lokaciji na koju pokazuje pokazivač na stek u argument, a zatim povećava pokazivač na stek za 1:
POP EAX
- Instrukcija PUSHF smanjuje pokazivač na stek za 1 a zatim na lokaciju na koju pokazuje pokazivač na stek, stavlja sadržaj flag registra

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:
PUSH EAX
- Instrukcija POP stavlja podatke koji se nalaze na lokaciji na koju pokazuje pokazivač na stek u argument, a zatim povećava pokazivač na stek za 1:
POP EAX
- Instrukcija PUSHF smanjuje pokazivač na stek za 1 a zatim na lokaciju na koju pokazuje pokazivač na stek, stavlja sadržaj flag registra
- Instrukcija POPF u flag registar stavlja sadržaj memorijske adrese na koju pokazuje pokazivač na stek, a zatim pokazivač na stek uvećava za 1

Instrukcije za rad sa stekom

- Instrukcija PUSH smanjuje za 1 pokazivač na stek i podatak koji se nalazi u argumentu stavlja na lokaciju na koju pokazuje pokazivač na stek:
PUSH EAX
- Instrukcija POP stavlja podatke koji se nalaze na lokaciji na koju pokazuje pokazivač na stek u argument, a zatim povećava pokazivač na stek za 1:
POP EAX
- Instrukcija PUSHF smanjuje pokazivač na stek za 1 a zatim na lokaciju na koju pokazuje pokazivač na stek, stavlja sadržaj flag registra
- Instrukcija POPF u flag registar stavlja sadržaj memorijske adrese na koju pokazuje pokazivač na stek, a zatim pokazivač na stek uvećava za 1

Instrukcije za rad sa flagovima

- Fleg za prekide

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos
 - Instrukcija CLC poništava fleg za prenos

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos
 - Instrukcija CLC poništava fleg za prenos
 - Instrukcija CMC komplementira fleg za prenos

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos
 - Instrukcija CLC poništava fleg za prenos
 - Instrukcija CMC komplementira fleg za prenos
- Ostali flag-ovi
 - Instrukcija SAHF čuva sadržaj AH registra u nižim bajtovima fleg registra

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos
 - Instrukcija CLC poništava fleg za prenos
 - Instrukcija CMC komplementira fleg za prenos
- Ostali flag-ovi
 - Instrukcija SAHF čuva sadržaj AH registra u nižim bajtovima fleg registra
 - Instrukcija LAHF stavlja u AH registar ono što se nalazi u nižim bajtovima fleg registra

Instrukcije za rad sa flagovima

- Fleg za prekide
 - Instrukcija STI postavlja fleg za prekide; u tom slučaju procesor prihvata zahtev za prekid
 - Instrukcija CLI poništava fleg za prekide
- Fleg za prenos
 - Instrukcija STC postavlja fleg za prenos
 - Instrukcija CLC poništava fleg za prenos
 - Instrukcija CMC komplementira fleg za prenos
- Ostali flag-ovi
 - Instrukcija SAHF čuva sadržaj AH registra u nižim bajtovima fleg registra
 - Instrukcija LAHF stavlja u AH registar ono što se nalazi u nižim bajtovima fleg registra

Ulazno-izlazne instrukcije

Ulazno-izlazne instrukcije

- Instrukcija IN skoro uvek radi sa operandima AX i DX (tj EAX,EDX ili RAX,RDX).DX često drži adresu porta za čitanje,a AX prima podatke od porta

Ulazno-izlazne instrukcije

- Instrukcija IN skoro uvek radi sa operandima AX i DX (tj EAX,EDX ili RAX,RDX).DX često drži adresu porta za čitanje,a AX prima podatke od porta
- Instrukcija OUT je veoma slična instrukciji IN.OUT stavlja podatke iz registra koje je u prvom operandu,na port dat u drugom operandu

Sistemske instrukcije

Ulazno-izlazne instrukcije

- Instrukcija IN skoro uvek radi sa operandima AX i DX (tj EAX,EDX ili RAX,RDX).DX često drži adresu porta za čitanje,a AX prima podatke od porta
- Instrukcija OUT je veoma slična instrukciji IN.OUT stavlja podatke iz registra koje je u prvom operandu,na port dat u drugom operandu

Sistemske instrukcije

- Instrukcija SYSENTER uvodi procesor u zaštićeni režim

Ulazno-izlazne instrukcije

- Instrukcija IN skoro uvek radi sa operandima AX i DX (tj EAX,EDX ili RAX,RDX).DX često drži adresu porta za čitanje,a AX prima podatke od porta
- Instrukcija OUT je veoma slična instrukciji IN.OUT stavlja podatke iz registra koje je u prvom operandu,na port dat u drugom operandu

Sistemske instrukcije

- Instrukcija SYSENTER uvodi procesor u zaštićeni režim
- Instrukcija SYSEXIT izvlači procesor iz zaštićenog režima i uvodi ga u režim za korisnika