Asemblersko programiranje

Asemblersko programiranje

- Zasniva se na korišćenju **asemblerskih naredbi**
- Programski jezici visokog nivoa i niskog nivoa
- Mane i prednosti asemblerskog programiranja
- Primene asemblerskog programiranja
- **Svaki asemblerski jezik** vezan je za neki **konkretan procesor**
- **Hipotetički procesor KONCEPT** – ovladavanje principima rada računara

Procesor KONCEPT

- **16-bitne lokacije**, ukupno 2^{16} adresa za lokacije
- **Registri opšte namene** %0, %1, ..., %15
- **Status registar** – flegovi N, M, P, V
- **Neposredno**, **direktno**, **registarsko**, **posredno** (indirektno) i **indeksno adresiranje**
- **Dvoadresne naredbe** (0, 1, ili 2 operanda)
- **Prvi operand** je **ulazni**, **drugi** je **ulazno-izlazni**

Celobrojna aritmetika

- Sabiranje: **SABERI**, **SABERI_P** – posmatra se i logička promenljiva P, za višestruku preciznost
- Oduzimanje: **ODUZMI**, **ODUZMI_P**
- Inkrement, dekrement: **DODAJ_1**, **ODBIJ_1**
- Poređenje: **UPOREDI** – oduzimanje i postavljanje vrednosti logičkih promenljivih

SABERI

PO_i	DO_i	P_i	Z_i	P_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$Z_i = PO_i \hat{} DO_i \hat{} P_i$

$P_{i+1} = ((PO_i | DO_i) \& P_i) | (PO_i \& DO_i)$

$P_0 = 0$

SABERI_P: $P_0 = P$

DODAJ_1

$N = \sim(Z_{15}|Z_{14}|\ Z_{13}|Z_{12}|\ Z_{11}|Z_{10}|\ Z_9|Z_8|\ Z_7|Z_6|\ Z_5|Z_4|\ Z_3|Z_2|\ Z_1|Z_0)$

$M = Z_{15}$

$P = P_{16}$

$V = ((\sim PO_{15})\&(\sim DO_{15})\&Z_{15})|(PO_{15}\&DO_{15}\&(\sim Z_{15}))$

ODUZMI

DO_i	PO_i	P_i	R_i	P_{i+1}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$R_i = PO_i \hat{} DO_i \hat{} P_i$

$P_{i+1} = ((DO_i | P_i) \& (\sim PO_i)) | (DO_i \& P_i)$

$P_0 = 0$

ODUZMI_P: $P_0 = P$

ODBIJ_1, **UPOREDI**

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$

$M = R_{15}$

$P = P_{16}$

$V = ((\sim PO_{15}) \& DO_{15} \& R_{15}) | (PO_{15} \& (\sim DO_{15}) \& (\sim R_{15}))$

Rukovanje bitovima

- Logičke operacije na nivou bitova:
  - logičko i – naredba **I**
  - logičko ili – naredba **ILI**
  - logičko ne – naredba **NE**

- Pomeranje na nivou bitova:
  - za jedno mesto u levo, množenje sa 2 – **LEVO**
  - za jedno mesto u desno, deljenje sa 2 – **DESNO**

I, ILI, NE

I

$R_i = PO_i \& DO_i$

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$
$M = R_{15}$
$P = 0$
$V = 0$

ILI

$R_i = PO_i | DO_i$

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$
$M = R_{15}$
$P = 0$
$V = 0$

NE

$R_i = \sim PO_i$

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$
$M = R_{15}$
$P = 0$
$V = 0$

LEVO, DESNO

LEVO

$R_{i+1} = PO_i$
$R_0 = 0$

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$
$M = R_{15}$
$P = PO_{15}$
$V = PO_{15} \wedge PO_{14}$

DESNO

$R_i = PO_{i+1}$
$R_{15} = PO_{15}$

$N = \sim(R_{15}|R_{14}|R_{13}|R_{12}|R_{11}|R_{10}|R_9|R_8|R_7|R_6|R_5|R_4|R_3|R_2|R_1|R_0)$
$M = R_{15}$
$P = PO_0$
$V = 0$

Prebacivanje

- Naredba za prebacivanje sadržaja između registara procesora i memorijskih lokacija: **PREBACI**
  - u lokaciju određenu prvim operandom se smešta vrednost određena drugim operandom
  - ne utiče na vrednosti logičkih promenljivih
  - kombinacije sufiksa: **RR**, **NR**, **DR**, **PR**, **IR**, **RD**, **RP**, **RI**

Upravljačke naredbe

- **Mašinske naredbe** se **smeštaju u redosledu izvršavanja u memorijske lokacije sa uzastopnim rastućim adresama**
- Nije podesno u slučajevima kada obrada podataka zavisi od vrednosti obrađivanih podataka
- Redosled izvršavanja naredbi mora se određivati dinamički u toku izvršavanja programa
- **Uslovne** i **bezuslovne upravljačke naredbe**
- **Uslovne**: jedini operand je adresa ciljne naredbe ako je ispunjen uslov

Uslovne upravljačke naredbe

Oznaka uslovne upravljačke naredbe	uslov uslovne upravljačke naredbe
SKOČI_ZA_== ili SKOČI_ZA_N	N
SKOČI_ZA_!= ili SKOČI_ZA_NE_N	~N
SKOČI_ZA_< ili SKOČI_ZA_P	P
SKOČI_ZA_>= ili SKOČI_ZA_NE_P	~P
SKOČI_ZA_>	(~P)&(~N)
SKOČI_ZA_<=	P|N
SKOČI_ZA_±_<	M^V
SKOČI_ZA_±_>=	~(M^V)
SKOČI_ZA_±_>	(~(M^V))&(~N)
SKOČI_ZA_±_<=	(M^V) |N
SKOČI_ZA_M	M
SKOČI_ZA_NE_M	~M
SKOČI_ZA_V	V
SKOČI_ZA_NE_V	~V

Bezuslovne upravljačke naredbe

- **SKOČI** – jedan operand (ciljna adresa), jedino izmena redosleda izvršavanja
- **POZOVI** – jedan operand (ciljna adresa)
  - Izmena redosleda izvršavanja i smeštanje povratne adrese (engl. *return address* – adresa naredbe neposredno iza naredbe POZOVI) u registar %15
- **NATRAG** – nema operanada, jedino izmena redosleda izvršavanja
  - skok na adresu iz %15

Asemblerski jezik Koncept

EBNF notacija

Proširena Bekus-Naurova forma
(engl. *Extended Backus-Naur Form* – EBNF):

- ime_pravila –> način_formiranja
- razmak – razdvajanje delova pravila
- | – alternative
- ( ) – grupisanje delova pravila
- [ ] – pojavljivanje nijednom ili jednom
- { } – pojavljivanje nijednom, jednom ili više puta
- “ ” – poništavanje značenja specijalnih znakova

EBNF definicije

- malo_slovo –> a|b|c|č|ć|d|đ|e|f|g|h|i|j|k|l|m|n|o|p|r|s|š|t|u|v|z|ž
- cifra –> 0|1|2|3|4|5|6|7|8|9
- decimalni_broj –> cifra{cifra}
- heksa_cifra –> cifra|A|B|C|D|E|F
- heksadecimalni_broj –> 0x(heksa_cifra){heksa_cifra}
- broj –> decimalni_broj|heksadecimalni_broj
- labela –> malo_slovo{malo_slovo|cifra|_}

EBNF definicije

- neposredni_operand –> ($broj|$labela)
- direktni_operand –> labela
- registar –> %(0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15)
- registarski_operand –> registar
- posredni_operand –> “(“registar“)“
- indeksni_operand –> (broj|labela)“(“registar“)“
- razmak –> “ “ {“ “}
- nova_linija –> “početak linije“ {“ “}

EBNF definicije

- Obrazovanje **osnovnih asemblerskih naredbi**:

```
osnovna_naredba -> nova_linija [labela:] razmak
          ((SABERI|SABERI_P) razmak registarski_operand,registarski_operand
          |(ODUZMI|ODUZMI_P) razmak registarski_operand,registarski_operand
          |UPOREDI           razmak registarski_operand,registarski_operand
          |(I|ILI)           razmak registarski_operand,registarski_operand
          |(DODAJ_1|ODBIJ_1) razmak registarski_operand
          |(NE|LEVO|DESNO)   razmak registarski_operand)
```

- Obrazovanje **asemblerskih naredbi prebacivanja**:

```
naredba_prebacivanja -> nova_linija [labela:] razmak
            (PREBACI_RR razmak registarski_operand,registarski_operand
            |PREBACI_NR razmak registarski_operand,neposredni_operand
            |PREBACI_DR razmak registarski_operand,direktni_operand
            |PREBACI_PR razmak registarski_operand,posredni_operand
            |PREBACI_IR razmak registarski_operand,indeksni_operand
            |PREBACI_RD razmak direktni_operand,registarski_operand
            |PREBACI_RP razmak posredni_operand,registarski_operand
            |PREBACI_RI razmak indeksni_operand,registarski_operand)
```

EBNF definicije

- Obrazovanje **asemblerskih upravljačkih naredbi**:

```
upravljačka_naredba -> nova_linija [labela:] razmak
                       ((SKOČI
                       |SKOČI_ZA_==
                       |SKOČI_ZA_!=
                       |SKOČI_ZA_<
                       |SKOČI_ZA_>=
                       |SKOČI_ZA_>
                       |SKOČI_ZA_<=
                       |SKOČI_ZA_±_<
                       |SKOČI_ZA_±_>=
                       |SKOČI_ZA_±_>
                       |SKOČI_ZA_±_<=
                       |SKOČI_ZA_N
                       |SKOČI_ZA_NE_N
                       |SKOČI_ZA_P
                       |SKOČI_ZA_NE_P
                       |SKOČI_ZA_M
                       |SKOČI_ZA_NE_M
                       |SKOČI_ZA_V
                       |SKOČI_ZA_NE_V
                       |POZOVI) razmak labela)
                       |NATRAG)
```

EBNF za asemblerske direktive

- Direktive upućene asembleru za **zauzimanje i inicijalizaciju memorijskog prostora – ZAUZMI** i **NAPUNI**
- Direktive **nisu izvršne**, već su **uputstva asembleru**, prethode naredbama

direktiva –> nova_linija [labela:] razmak

(ZAUZMI|NAPUNI) razmak broj

EBNF za asemblerski program

telo –> {direktiva
|osnovna_naredba
|naredba_prebacivanja
|upravljačka_naredba}

program –> POČETAK
razmak labela telo nova_linija
KRAJ

Primeri asemblerskih programa

Primer: Program za računanje NZD

```
                POČETAK          ulaz
ulaz:           PREBACI_NR       $12,%0
                PREBACI_NR       $10,%1
ponovo:         UPOREDI          %1,%0
                SKOČI_ZA_==      kraj
                SKOČI_ZA_<       manje
veće:           ODUZMI           %1,%0
                SKOČI            ponovo
manje:          ODUZMI           %0,%1
                SKOČI            ponovo
kraj:           SKOČI            kraj
                KRAJ
```

Dvostruka preciznost – neoznačeni brojevi

```
                POČETAK         ulaz
a_donji:        NAPUNI          0xFFFF
a_gornji:       NAPUNI          0xFFFF
b_donji:        NAPUNI          0xFFFF
b_gornji:       NAPUNI          0xFFFF
greška:         NAPUNI          0
ulaz:           PREBACI_DR      a_donji,%0
                PREBACI_DR      a_gornji,%1
                PREBACI_DR      b_donji,%2
                PREBACI_DR      b_gornji,%3
                SABERI          %2,%0
                SABERI_P        %3,%1
                SKOČI_ZA_P      van_opsega
                SKOČI           kraj
van_opsega:     PREBACI_NR      $1,%4
                PREBACI_RD      %4,greška
kraj:           SKOČI           kraj
                KRAJ
```

Dvostruka preciznost – označeni brojevi

```
                POČETAK         ulaz
a_donji:        NAPUNI          0xFFFF
a_gornji:       NAPUNI          0xFFFF
b_donji:        NAPUNI          0xFFFF
b_gornji:       NAPUNI          0xFFFF
greška:         NAPUNI          0
ulaz:           PREBACI_DR      a_donji,%0
                PREBACI_DR      a_gornji,%1
                PREBACI_DR      b_donji,%2
                PREBACI_DR      b_gornji,%3
                SABERI          %2,%0
                SABERI_P        %3,%1
                SKOČI_ZA_V      van_opsega
                SKOČI           kraj
van_opsega:     PREBACI_NR      $1,%4
                PREBACI_RD      %4,greška
kraj:           SKOČI           kraj
                KRAJ
```

Rad sa mašinskom normalizovanom formom

$-1.5_{10} ==$
$-1.1000000_2 \times 2^0 ==$
1100000001000000_2

```
realan:          NAPUNI          0xC040
```

Postavljanje predznaka na 0:

```
PREBACI_DR       realan,%0
PREBACI_NR       $0x7FFF,%1
I                %1,%0
```

Postavljanje predznaka na 1:

```
PREBACI_NR       $0x8000,%1
ILI              %1,%0
```

Rad sa mašinskom normalizovanom formom

Izdvajanje eksponenta:

```
                PREBACI_DR      realan,%0
                PREBACI_NR      $0x7F80,%1
                I               %1,%0
                PREBACI_NR      $7,%2
ponovo:         DESNO           %0
                ODBIJ_1         %2
                SKOČI_ZA_NE_N   ponovo
```

Rukovanje logičkim vrednostima

```
int a, b;

char c, d;

...

c = (a!=b);

d = c;
```

```
a:          ZAUZMI          1
b:          ZAUZMI          1
c:          ZAUZMI          1
d:          ZAUZMI          1
            ...
            PREBACI_DR      a,%0
            PREBACI_DR      b,%1
            PREBACI_NR      $1,%2
            UPOREDI         %1,%0
            SKOČI_ZA_!=     dalje
            PREBACI_NR      $0,%2
dalje:      PREBACI_RD      %2,c
            PREBACI_RD      %2,d
```

Računanje vrednosti celobrojnih izraza

unsigned int a, b, c; //neoznačeni celi brojevi

c = (a-b)*2+(a+b)/2;

```
a:          ZAUZMI          1
b:          ZAUZMI          1
c:          ZAUZMI          1
            ...
            PREBACI_DR      a,%0
            PREBACI_DR      b,%1
            PREBACI_DR      a,%2
            ODUZMI          %1,%2
            LEVO            %2      <-- (a-b)*2
            SABERI          %1,%0
            DESNO           %0      <-- (a+b)/2
            SABERI          %0,%2
            PREBACI_RD      %2,c
```

Rukovanje nizovima

- t[365] – niz od 365 vrednosti temperatura
- **prvi**, **poslednji** – indeks prvog i poslednjeg elementa podintervala
- **donja**, **gornja** – donja i gornja granica temperature
- **broj** = 0 – brojač dana sa temperaturom iz opsega

```
for (i = prvi;  i<=poslednji;  i++)
        if ((t[i]>=donja) && (t[i]<=gornja))
                broj++;
```

```
t:              ZAUZMI          365
prvi:           ZAUZMI          1
poslednji:      ZAUZMI          1
donja:          ZAUZMI          1
gornja:         ZAUZMI          1
broj:           ZAUZMI          1
                ...
                PREBACI_NR      $0,%0
                PREBACI_DR      prvi,%1
                PREBACI_DR      poslednji,%2
                PREBACI_DR      donja,%3
                PREBACI_DR      gornja,%4
provera:        UPOREDI         %2,%1
                SKOČI_ZA_>      izlaz
ponovo:         UPOREDI         %3,t(%1)
                SKOČI_ZA_±_<    naredni
                UPOREDI         %4,t(%1)
                SKOČI_ZA_±_>    naredni
                DODAJ_1         %0
naredni:        DODAJ_1         %1
                SKOČI           provera
izlaz:          PREBACI_RD      %0,broj
```


brojač


indeks


uslov za kraj *for*


t[i]>=donja


t[i]<=gornja


broj=broj+1

Rukovanje slogovima

```
struct vreme
{
        unsigned sat;
        unsigned minut;
        unsigned sekund;
} a, b;

...

b = a;
```

Rukovanje slogovima

```
a:      ZAUZMI          3
b:      ZAUZMI          3
        ...
indeks  PREBACI_NR      $0,%0
        PREBACI_IR      a(%0),%1
        PREBACI_RI      %1,b(%0)
        DODAJ_1         %0
        PREBACI_IR      a(%0),%1
        PREBACI_RI      %1,b(%0)
        DODAJ_1         %0
        PREBACI_IR      a(%0),%1
        PREBACI_RI      %1,b(%0)
```