ARS I Avditorne vaje

Pri nekem programu je potrebno izvršiti N=1620 ukazov. Pogostost in trajanje posameznih vrst ukazov računalnika sta naslednja:

Vrsta ukaza	Štev. urinih period	Pogostost
Prenosi podatkov	8	31%
ALE ukazi	5	48%
Kontrolni ukazi	6	21%

Zanima nas:

a) Kolikšen je CPI za ta program?
b) Koliko MIPS ima ta računalnik, če je frekvenca ure f_{CPE} = 300 MHz?
c) Kakšen je CPE čas tega programa?

Rešitev:

a) $CPI = \sum_{i=1}^{3} CPI_i * p_i = 8*0{,}31 + 5*0{,}48 + 6*0{,}21 = 6{,}14$

Računalnik potrebuje povprečno 6,14 urine periode za en ukaz.

b) $MIPS = \dfrac{f_{CPE}}{CPI*10^6} = \dfrac{300*10^6}{6{,}14*10^6} = 48{,}85$

Računalnik izvede povprečno 48.850.000 ukazov na sekundo.

c) $CPE_{čas} = \dfrac{Število\,ukazov}{MIPS*10^6} = \dfrac{1620}{48{,}85*10^6} = 33{,}16*10^{-6} = 33{,}16\mu s$

Program s 1620 ukazi se izvede v 33,16 μs.

Druga oblika enačbe za izračun CPE časa pa je:

$$CPE_{čas} = Število\_ukazov * CPI * t_{CPE} = \frac{število\_ukazov * CPI}{f_{CPE}}$$

Na računalniku s frekvenco urinega signala 250 MHz je v povprečju potrebnih 5 urinih period za en ukaz. Zaradi prekinitev se zmogljivost CPE merjena v MIPS zmanjša za 0,12%. Ugotoviti želimo povprečni čas med dvema prekinitvama, če se ob vsaki prekinitvi porabi 48 urinih period za klic prekinitvenega servisnega programa in 27 urinih period za vračanje iz njega.

Rešitev:

f_{CPE} = 250 MHz
CPI = 5
ΔMIPS = 0,12%
t_{pv} = 48 (porabljen čas za klic (vstop) prekinitvenega podprograma)
t_{pi} = 27 (porabljen čas za vračanje (izstop) iz prekinitvenega podprograma)

$$MIPS = \frac{f_{CPE}}{CPI * 10^6} = \frac{250 * 10^6}{5 * 10^6} = 50$$

Zmanjšanje zmogljivosti zaradi prekinitev ΔMIPS je 0,12%. Ta čas se porabi za vstopni in izstopni del prekinitvenih podprogramov. Velja torej:

$$\Delta MIPS * 10^6 * CPI = N_p * (t_{pv} + t_{pi})$$

kjer je N_p število prekinitev na sekundo.

$$N_p = \frac{\Delta MIPS * 10^6 * CPI}{t_{pv} + t_{pi}} = \frac{0{,}0012 * 50 * 10^6 * 5}{75} = 4 * 10^3 \ \textit{prekinitev/s}$$

Povprečni čas med dvema prekinitvama t_P pa je:

$$t_P = \frac{1}{N_p} = \frac{1}{4 * 10^3} = 0{,}25 * 10^{-3} s = 0{,}25 ms$$

Delovanje nekega računalnika želimo pohitriti z elektronskim nadomestkom za trdi disk, ki je v povprečju 50-krat hitrejši od trdega diska. Kolikšen procent časa se mora uporabljati disk, da bo povečanje hitrosti 2,5 kratno?

Rešitev:

Uporabimo Amdahlov zakon:

$$S(N) = \frac{N}{1 + (N-1) * f}$$

f = delež operacij , ki se ne pohitrijo
N = faktor pohitritve za (1 - f) operacij
S(N) = povečanje hitrosti celotnega računalnika

V našem primeru je S(N) = 2,5; N = 50; iščemo pa (1 - f)

$$1 - f = 1 - \frac{N - S(N)}{S(N) * (N-1)} = 1 - \frac{50 - 2{,}5}{2{,}5 * 49} = 1 - 0{,}3878 = 0{,}6122$$

Disk se mora uporabljati 61% časa, da bo delovanje računalnika 2,5 krat hitrejše.

Raziskovalna skupina želi rešiti nek problem za katerega obstaja zelo dober algoritem pri katerem se lahko 99% operacij izvede vzporedno. Za rešitev tega problema so se odločili uporabiti MIMD računalnik, na katerem bi se omenjeni algoritem vzporedno izvajal. Da bi omejili stroške, so se odločili, da bo dovolj, če dosežejo 90% maksimalne pohitritve. Koliko procesorjev mora imeti MIMD računalnik?

Rešitev:

Maksimalna pohitritev je $\lim_{N\to\infty} S(N) = \dfrac{N}{1+(N-1)*f} = \dfrac{1}{f}$

Delež operacij, ki se jih ne da pohitriti f = 0,01. Iz tega sledi, da je maskimalna možna pohitritev z dodajanjem procesorjev v MIMD računalnik enaka 100. Zaradi omejitve stroškov se omejimo na 90% maksimalne pohitritve ali na 90-kratno pohitritev. Iz

$90 = \dfrac{N}{1+(N-1)f}$ dobimo torej N = 891.

Imamo nek preprost 16 bitni računalnik, ki ima 16 splošno namenskih registrov. Registre lahko uporabljajo vsi ukazi, tako za ciljne kot izvorne operande. Za ta računalnik so predvideni naslednji trije formati ukazov:

Format 1:

4	4	4	4
op.koda	R_d	R_{s1}	R_{s2}

Format 2:

8	4	4
op.koda	R_d	R_s

Format 3:

16
op.koda

Operacijska koda določa tip ukaza, polja z operandi pa variacije ukaza. Vzemimo, da ima 15 tipov ukazov format 1 in 14 tipov ukazov format 2. Koliko je vseh možnih variacij ukazov v formatu 1 skupaj in koliko je vseh možnih variacij ukazov v formatu 2 skupaj? Koliko tipov ukazov imamo lahko v formatu 3?

Rešitev:

Na mestih rezerviranih za parametre imamo lahko karkoli in tako imamo lahko v formatu 1 za vsak ukaz 2^(4+4+4) = 4096 različnih variacij. Ker imamo 15 različnih tipov ukazov je vseh možnih variacij torej 4096*15 = 61440.

V formatu 2 imamo za vsak ukaz 2^(4+4) = 256 možnih različnih variacij. Ker imamo 14 različnih tipov ukazov, je vseh možnih variacij v formatu 2 torej 256*14 = 3584.

Zadnji del naloge rešimo tako, da ugotovimo, koliko operacijskih kod je še na voljo za format 3. Glavno vodilo je, da morajo biti operacijske kode enolično določene. Za format 1 smo že uporabili 15 od 16 možnih operacijskih kod. Neizrabljena koda nam določa prve 4 bite operacijske kode formatov 2 in 3. Preostali 4 biti operacijske kode za format 2 nam dovoljujejo 16 različnih ukazov v formatu 2, izrabljenih pa je 14. Za format 3 nam tako ostaneta 2 neizrabljeni operacijski kodi formata 2. Preostalih 8 bitov operacijske kode formata 3 nam dovoljuje 256 različnih ukazov in ker lahko za prvih 8 bitov uporabimo 2 kombinaciji nam to skupaj da 512 različnih operacijskih kod in s tem tipov ukazov v formatu 3.

Desetiško predznačeno število –25 zapišite v osembitni predstavitvi s fiksno vejico v vseh štirih 8 bitnih načinih za predstavitev števil s predznakom. Enako naredite še s številom 33. Števili zapišite v dvojiškem in šestnajstiškem sistemu. Razmislite, kako poteka seštevanje (množenje) teh dveh števil (dvojiško).

Rešitev:

a) predznak in velikost:

$V(b) = (-1)^{b_{n-1}} \sum_{i=0}^{n-2} b_i 2^i$ obseg -127…+127

$25_{(10)} = ?_{(2)} = 11001_{(2)}$
25 : 2 = 12 + 1
12 : 2 = 6 + 0
6 : 2 = 3 + 0
3 : 2 = 1 + 1
1 : 2 = 0 + 1

$33_{(10)} = ?_{(2)} = 100001_{(2)}$
33 : 2 = 16 + 1
16 : 2 = 8 + 0
8 : 2 = 4 + 0
4 : 2 = 2 + 0
2 : 2 = 1 + 0
1 : 2 = 0 + 1

torej: $-25_{(10)} = 10011001_{(2)}$ ($99_{(16)}$), $33_{(10)} = 00100001_{(2)}$ ($21_{(16)}$)
Pri seštevanju je potrebno posebej upoštevati predznak, ravno tako pri množenju.

b) predstavitev z odmikom

$V(b) = \sum_{i=0}^{n-1} b_i 2^i - 2^{n-1}$ obseg -128…+127

$-25+2^{n-1} = -25 + 128 = 103$

$103_{(10)} = ?_{(2)} = 1100111_{(2)}$

$3+2^{n-1} = 33 + 128 = 161$

$161_{(10)} = ?_{(2)} = 10100001_{(2)}$

torej: $-25_{(10)} = 01100111_{(2)}$ ($67_{(16)}$), $33_{(10)} = 10100001_{(2)}$ ($A1_{(16)}$)

Pri seštevanju ima rezultat odmik upoštevan 2-krat, torej je potrebno en odmik odšteti. Pri množenju je potrebno pred množenjem odmik odšteti, zmnožiti števili in nato rezultatu prišteti odmik.

c) eniški komplement

$$V(b)=\sum_{i=0}^{n-1} b_i 2^i - b_{n-1}(2^n - 1)$$ obseg -127…+127

$25_{(10)} = 11001_{(2)} = 00011001_{(2)}$

torej: $-25_{(10)} = 11100110_{(2)}$ ($E6_{(16)}$), $33_{(10)} = 00100001_{(2)}$ ($21_{(16)}$)

Pri seštevanju moramo ob prenosu iz mesta *n-1* k rezultatu prišteti 1, sicer seštevamo predznak enako kot ostale bite.

d) dvojiški komplement

$$V(b)=\sum_{i=0}^{n-1} b_i 2^i - b_{n-1}(2^n)$$ obseg -128…+127

$$\begin{aligned}
25_{(10)} = {} & 00011001_{(2)} \\
& 11100110_{(2)} \\
& +00000001_{(2)} \\
& =11100111_{(2)}
\end{aligned}$$

torej: $-25_{(10)} = 11100111_{(2)}$ ($E7_{(16)}$), $33_{(10)} = 00100001_{(2)}$

Števila normalno seštevamo. Za množenje obstaja Boothov algoritem (seštevanja + pomiki).

Imamo naslednja tri števila 7, -81 in 200.

a) Minimalno koliko bajtov potrebujemo za njihovo predstavitev, če jih želimo zapisati kot predznačena števila v dvojiškem komplementu?
b) Števila zapišite po zahtevah naloge a.
c) Vsa števila ustrezno razširite na 32 bitov.

Rešitev:

a) Za števili 7 in –81 je dovolj en bajt, za število 200 pa moramo uporabiti dva bajta, če ga želimo zapisati kot predznačeno število, čeprav je za nepredznačen zapis dovolj le en bajt. Števila 200 ne moremo predstaviti z 8 biti, ker prekorači obseg 8-bitnih predznačenih števil –128..+127. Število 11001000 (nepredznačeno 200) predstavlja 8-bitni predznačen zapis števila –56.

b) 7 = 00000111

-81: Najprej 81 pretvorimo v dvojiški zapis in dobimo 01010001, nato izračunamo še dvojiški komplement in dobimo 10101111. Torej

-81 = 10101111
200 = 0000000011001000

c)	00000111	--->	00000000000000000000000000000111
	10101111	--->	11111111111111111111111110101111
	0000000011001000	--->	00000000000000000000000011001000

Števila 75, -7.5, 0.75 in -0.375 pretvorite v IEEE 754 32-bitni zapis.

Rešitev:

75 = 1001011.0 = 1.001011 * 2^{6} ---> prištejemo odmik 127 k eksponentu --->
---> 1.001011 * 2^{133} ---> 0 – predznak, 10000101 – eksponent in
00101100000000000000000 – mantisa; enica pred decimalno piko je skrita
(implicitna) ---> 01000010100101100000000000000000

-7.5 = -111.1 = -1.111 * 2^{2} --- > -1.111 * 2^{129} --->
---> 11000000111100000000000000000000

0.75= 0.11 = 1.1 * 2^{-1} ---> 1.1 * 2^{126} ---> 00111111010000000000000000000000

-0.375 = -0.011 = -1.1 * 2^{-2} ---> -1.1 * 2^{125} ---> 10111110110000000000000000000000

0.375 * 2 = **0**.75
0.75 * 2 = **1**.5
0.5 * 2 = **1**.0

Izračunajte izraz A+(-B) kjer je A = 11111110 (8-bitno predznačeno celo število) in B = 01000000010000000000000000000000 (število v IEEE 754 32-bitni obliki). Postopek: število B najprej pretvorite v 8-bitno predznačeno celo število, izračunajte dvojiški komplement in števili seštejte.

Rešitev:

Število B: eksponent je 128, mantisa je 1.1 ---> B = +1.1 * 2^1 = 1.5 * 2 = 3

B = 00000011 dv. kompl B = 11111101

A + (-B) = -2 + (-3) = 11111110 + 11111101 = 11111011 = -5

Imamo števili v 32-bitni IEEE 754 obliki in sicer A = 00111110110100000010000000000000 in B = 00111110010000000001000000000000. Izračunajte

a) vsoto
b) zmnožek

Rešitev:

A = +1.1010000001 * $2^{(125=-2+127)}$ = 4.0649414062500000E-01
B = +1.10000000001 * $2^{(124=-3+127)}$ = 1.8756103515625000E-01

a) A + B

Mantiso pri številu B (število z manjšim eksponentom) pomaknemo za eno mesto v desno, ker morata biti eksponenta enaka.

```
A+B = + 1.1010000001      * 2^(125=-2+127)
      + 0.110000000001    * 2^(125=-2+127)
----------------------------------------------------
      +10.011000000101    * 2^(125=-2+127)
```

Poravnamo mantiso v obliko 1.? ter popravimo eksponent in dobimo

```
      +1.0011000000101    * 2^(126=-1+127)
```

A + B = 0011 1111 0001 1000 0001 0100 00000000 = 5.9405517578125000E-01

b) A * B

Zmnožimo mantisi

```
  11010000001 * 110000000001
----------------------------------------------------
  11010000001
   11010000001
    00000000000
     00000000000
      00000000000
       00000000000
        00000000000
         00000000000
          00000000000
           00000000000
            00000000000
             11010000001
-----------------------------------------------------
100111000001001010000001
```

Seštejemo eksponente in odštejemo en odmik 125 + 124 – 127 = 122. Po množenju mantis in seštevanju eksponentov dobimo

$$+10.0111000001001010000001 \quad * \ 2^{(122=-5+127)}$$

Poravnamo mantiso v obliko 1.? ter popravimo eksponent in dobimo

$$+1.00111000001001010000001 \quad * \ 2^{(123=-4+127)}$$

A * B = 0011 1101 1001 1100 0010 0101 0000 0010 = 7.6242461800575256E-02