

Kolega **mu3pm** je davne 2010. odgovorio na pitanja iz pdf-a kolege walla (2.MI v.2.2). Ovo su njegovi odgovori, koje sam tu navela jer ih nisam vidjela na materijalima, a moglo bi biti korisno. Ne garantiram i ne znam točnost i ispričavam se ako ih već negdje ima navedeno u nekom dokumentu.

Gaala

1.

- a) $t_1 = 1.8667\text{ms}$, $t_2 = 1\text{ms}$ --> Cpu2 ima bolje performanse
b) $\text{CPI}(\text{novi D}) = 2.4$, $\text{CPI}(\text{novi C}) = 2,3$ --> promjena $\text{CPI}(\text{C})$ bi bila opravdanija

2.

ovaj sam nekaj muljavio jer mi nije skroz jasan...

al sam si zamislio da $nv=20$ znaci da vektorski proc obradi 20 instrukcija dok ovaj skalarni obradi 1...pa bi vektorski bio 20x brzi

$s=20$

$x=?$

a) $p=2$ (100% ubrzanje = 2 puta brze), isad se Amdahlov zakon ($p=1/((1-x)+(x/s))$) koristi i ispadne --> $x=0.5263$

b) $p=10$ (pola od maksimalnog zanci 10x brze) --> $x=0.947$

c) $p=2$, $s=100$ (jer je $nv=100$) --> $x=0.505$

3.

op kod = de --> 11011110

ra = r23 --> 010111

rb = r23 --> 010111

c2 = -9 --> 111111110111

rjesenje: 11011110|010111|010111|111111110111

4.

ako instrukcija treba bit dugacka 16 bita a trebamo sve operacije zadržat onda ce ovak valjda bit:

op kod=5

ra=2

rb=2

rc=2

c2=5

registre sam uzeo po 2 bita, jer ako se uzme vise, ostaje samo 2 bita za konstantu, a to je malo neprakticno...

ovak mozemo koristit 4 registra (adreseiramo ih s 2 bita, pa su moguće 4 adrese)...

konstanta ce nam imat 5 bita...

ovo ce smanjit kolicnu registara, smanjit direktan pristup memorijskim lokacijama...

trebat ce vise naredbi za neke stvari napraviti...

bit ce potrebno pazljivije baratanje registrima...

5.

prema sadržaju mamorije vidimo da treba napraviti $r_1, C_2(r_3)$ --> u registar r_1 spremit vrijednost $C_2 + R[r_3]$ (zbroj konstante i vrijednosti koja se nalazi u r_3)..

C2 = 100A0 --> treba prosiriti predznak, pa dobijemo FFFF00A0
r3 = 05000005

C2+r3 = 04FF00A5 --> ovo spremamo u r1

6.

opet isto, samo kaj je sad u r3 nula, pa radimo la r3, C2
spremamo u r3 vrijednost C2

C2 = 10001 --> prosirenjem to je FFFF0001
i sad spremimo C2 u r3...

7.

a) la r7, 32 --> spremamo dekadskih 32 u r7, tj hexa 00000020 stavimo u r7
b) la r7, 32 (r5), uz to da je sadržaj r5 = 510 (valjda dekadski)
sad u r7 spremamo C2+ sadržaj r5

C2=00000020
r5=000001FE

C2+r5=0000021E --> spremamo u r7

8.

a) radimo ld --> u r1 spremamo sadržaj koji se u memoriji nalazi na lokaciji (C2+r5)
za nas zadatak imamo ld r1, C2(r4)

C2=00000021
r4 =00000004

C2+r4 = 25 --> u r1 spremamo ono što je u memoriji na 00000025 (nemamo zadano zadatkom što je tamo, pa napisemo odgovor ovako)

b) la r1, C2(r4)
C2+r4=00000025
spremamo u r1 hexadekadski podatak 00000025...

9.

IF --> pribavljanje instrukcije iz cache-a, PC+4-->PC
ID --> dekodiranje instrukcije, pribavljanje registara, prosljeđivanje konstanti
EX --> ALU operacije
MEM --> zapisivanje/citanje iz memorije
WB --> zapisivanje u registar

N>>M --> pretpostavljam da je to ASR za M bitova (CISC kod)

10.

model je onaj kaj je u predavanju broj 7...

vrijeme obrade jedne instrukcije je u neprotocnoj strukturi jednako 50ns

vrijeme obrade u protocnoj strukturi jednako je (broj segmenata svih instrukcija * trajanje jednog segmenta / N) --> $(10^7+4)*10/10^7 = 10.000004\text{ns}$

- 10^7+4 sam uzeo zato sto ako imamo jednu instrukciju, ona ima 5 segmenata, 2 instrukcije imaju 6 segmenata, itd...tj. broj segmenata = broj instrukcija + 4

odnos neprotoc/protoc = $50/10.0000004=4.999998$...dakle protocni model je skoro 5x brzi

11.

imamo 40ns i 45ns...

vrijeme obrade jedne instrukcije je u neprotocnoj strukturi jednako $(3*40+4*45) = 300\text{ns}$

svaki segment protodne ce biti 45ns --> 7. predavanje 21 str. (uzima se trajanje najveceg kod protodne)

vrijeme obrade u protocnoj strukturi jednako je (broj segmenata svih instrukcija * trajanje jednog segmenta / N) --> $(10000+6)*45/10000 = 45.027\text{ns}$

- $10000+6$ sam uzeo zato sto ako imamo jednu instrukciju, ona ima 7 segmenata, 2 instrukcije imaju 8 segmenata, itd...tj. broj segmenata = broj instrukcija + 6

odnos neprotoc/protoc = $300/45.027=6.6627...$

12.

INC M (CISC)

RISC:

ld r1, M

addi r2, r1, 1 (RAW hazard - citamo r1, nakon sto smo zapisivali u njega)

st r2, M (RAW hazard - citamo r2, nakon sto smo u njega zapisivali)

segmenata po instrukciji ima 4...

dakle

1. _____ (ld)

2. .. _____ (nop zbog RAW)

3. ... _____ (addi)

4. _____ (nop zbog RAW)

5. _____ (st)

broj perioda = 8

13.

n=12000, nema hazarda

za protocnu ce nam trajanje segmenta bit 18ns...

a) $(n+4)*ts/n \rightarrow (12004*18/12000) = 18.006ns$
ovo +4 je uzeto jer ima 5 segmenata po instrukc kao u prethodnim zadacima

b) $t_i = 76ns$ za neprotocnu
neprotocna: 76ns
protocna: 18.006ns
omjer = $76/18.006 = 4.2208$ puta brze

14.

skicirat na temelju predavanja 7...
po meni je to naredba JMP x...

IF --> dohvati instrukciju, PC+4
ID --> dekodiraj instrukciju, proslijedi x
EX --> PC+4+x
MEM --> nista
WB --> upisi novi PC

bonus ne znam...

15.

latencija protocnog = broj segmenata * trajanje najduljeg segmenta (isto za svaku naredbu)
latencija neprotocnog --> samo zbroj trajanja segmenata koje te naredba sadrzi (razlicito ovisno o naredbi)

dakle:

neprotocno:

ld (ima svih 5 segmenata) --> latencija = 625ps
st (nema WB segment) --> latencija = 525ps
sub (nema MEM) --> latencija = 475ps
addi (nema MEM) --> latencija = 475ps

protocno:

najdulje trajanje jest 150ps (IF i MEM)
svaka naredba ima latenciju $5*150 = 750ps$

16.

$r1 \leftarrow M[40+r6]$
 $r6 \leftarrow r2+r2$ (WAR - pisanje u r6, nakon sto je koristen r6 za citanje)

$M[50+r1] \leftarrow r6$ (RAW - citamo r6 nakon sto smo u njega nesto upisivali)

$r5 \leftarrow M[r5-16]$

$M[r5-16] \leftarrow r5$ (RAW - citamo r5 nakon upisivanja, WAR - upisujemo na [r5-16] nakon sto smo ga citali)

$r5 \leftarrow r5 + r5$ (WAR - upisujemo u r5, nakon sto smo ga prethodno citali)

Izmedju 1. i 3. instrukcije postoji RAW r1

Izmedju 4. i 6. RAW, WAR i WAW r6.

17.

opet ne znam dok ne dobijem odgovor na mejl kak točno ovo treba gledat...

al po meni se vjerojatno gleda da svaka perioda traje isto...

a) perioda ce trajat ko max segment --> 80ns...

perioda ce biti $N+3$ (jer je 4 segmenta) --> 10003...

trajanje jedne instrukcije = $80 * 10003 / 10000 = 80.024ns...$

b) $N \rightarrow$ beskonacno...

e sad mislim da mozemo uzet da je trajanje instrukcije = 80ns, a ne 80.024..

-kod jako velikog broja instrukcija prosj trajanje jedne se priblizava 80, a kod beskonacno je upravo = 80ns...

neprotocno: $40+70+50+80 = 240$

protocno: 80

ubrzanje: $240/80 = 3x...$

18.

a) prosjecno vrijeme = $(10*(N1+7) + T_{rekonfig} + 10*(N2+7)) / (N1+N2) = 10.00667ns$

b) $S = 64 / 10.00667 = 6.3957x$

19.

ovo bubam, ne nam jel to moguće tak izvest..

1. naredba:

tam di Rs ulazi u alu spojimo zbrajalo...pa izlaz zbrajala spojimo na reg cache...

2.naredba:

Rd spojimo na izlaz r2 iz reg cache-a, pa tak on moze doc u ALU...

20.

a)

do{

```
a=b;  
a+=25;  
b=a;  
b+=4;  
}while(b != c);
```

b)
nema pojma...

c)
neam pojma...