

Lösningar till tentamen i kursen EDA330

Datorsystemteknik

16/1 1998

Följande är skisser till lösningar av uppgifterna. Full poäng på en uppgift kräver i de flesta fall en något fylligare motivering. I en del fall är alternativa lösningar möjliga.

1.

a. User CPU time = $I \times \text{CPI} \times T_c = I \times \text{CPI} / f$

$$\text{CPI} = f \times \text{User CPU time} / I$$

$$N=20: \quad I=200 \times 10^6 + 20 \times 2 \times 10^6 = 240 \times 10^6$$

$$\text{CPI} = 100 \times 10^6 \times 4,2 / 240 \times 10^6 = 4,2 / 2,4 = \mathbf{1,75}$$

$$N=200: \quad I=200 \times 10^6 + 200 \times 2 \times 10^6 = 600 \times 10^6$$

$$\text{CPI} = 100 \times 10^6 \times 11,4 / 600 \times 10^6 = 11,4 / 6,0 = \mathbf{1,9}$$

$$N=400: \quad I=200 \times 10^6 + 400 \times 2 \times 10^6 = 1000 \times 10^6$$

$$\text{CPI} = 100 \times 10^6 \times 19,4 / 1000 \times 10^6 = 19,4 / 10,0 = \mathbf{1,94}$$

CPI ökar med ökande N, beroende på att andelen minnesåtkomster som tenderar att öka CPI pga cachemissar ökar.

b. $\text{CPI} = \text{CPI}_0 + I_D / I \times m_D \times \text{miss penalty} + m_I \times \text{miss penalty} = 1,5 + 10 \times I_D / I \times m_D + 10 \times m_I$

CPI_0 är CPI utan inverkan av minnesåtkomster. Denna är enligt uppgiften oberoende av N och lika med 1,5.

I_D är antal instruktioner som gör dataminnesåtkomst.

m_D är missannolikheten för datacache

m_I är missannolikheten för instruktionscache

$$N=20: \quad I_D = 0,1 \times 200 \times 10^6 + N \times 0,4 \times 2 \times 10^6 = 36 \times 10^6$$

$$N=200: \quad I_D = 0,1 \times 200 \times 10^6 + N \times 0,4 \times 2 \times 10^6 = 180 \times 10^6$$

$$N=400: \quad I_D = 0,1 \times 200 \times 10^6 + N \times 0,4 \times 2 \times 10^6 = 340 \times 10^6$$

Två obekanta. Välj två mätningar av CPI och sätt upp ekvationssystem:

$$N=20: \quad 1,5 + 1,5 \times m_D + 10 \times m_I = 1,75$$

$$N=200: \quad 1,5 + 3,0 \times m_D + 10 \times m_I = 1,9$$

Detta ger **missannolikhet för datacache = 10%**, och **missannolikhet för instruktionscache = 1%**.

2.

a. **Se avsnitt 3.8 i kursboken.** Register addressing, base addressing, immediate addressing, PC-relative addressing.

b.

```
slt $1, $15, $0
bne $1, $0, DEST

beq $15, $0, DEST

slt $1, $0, $15
bne $1, $0, DEST

slt $1, $15, $0
beq $1, $0, DEST
```

c. **Se avsnitt 3.7, sid. 127-130, i kursboken.** Hoppadressen beräknas genom att addera en konstant operand till det aktuella PC-värdet (=adress till hoppinstruktionen + 4). Operanden ryms i 16 bitar, men multipliceras med 4 innan den adderas till programräknarvärdet, och hoppavstånden sträcker sig alltså mellan -2^{18} och $+2^{18}-4$. Längsta avståndet från hoppinstruktionen blir alltså 2^{18} byte. Längre villkorliga hopp kan konstrueras enligt följande exempel:

```
L1:      beq $1, $0, L2
```

kan om $\text{abs}(L2-L1) > 2^{18}$ ersättas med:

```
L1:      bne $1, $0, L3
          j    L2
L3:
```

3.

a. **Se avsnitt 5.6 i kursboken.** I/O device request, invoke OS from user program, arithmetic overflow, undefined instruction, hardware malfunction.

b. **Se avsnitt 6.8 i kursboken.** De modifieringar som behöver göras är att lägga till ett register för att lagra adress (+4) till den instruktion som orsakat avbrottet, möjliga att tömma pipelinen fram till den orsakande instruktionen, ett cause-register för att lagra orsaken till avbrottet, samt en möjlighet att ladda PC med adressen till avbrottshanteraren.

c. **Se avsnitt 6.8 i kursboken.**

d. **Se avsnitt 7.3, sid 498-500, (samt A.7 för fördjupning) i kursboken.** Förhindra ytterligare avbrott genom att maska bort dem. Spara undan EPC och Cause. Avgör orsak till avbrottet. Spara eventuellt undan tillstånd för användarprogrammet. Tillåt ytterligare avbrott igen.

4.

- a. **Skillnaden beror på att den höga rumslokaliteten i instruktionshämtning utnyttjas bättre med större block.**
- b. Dels **tiden för arbitrering**, och dels **belastningen på bussen** (påverkar sannolikheten att bussen redan är reserverad samt tiden den är reserverad).
- c. Hämtningstid för block om 2 ord = (2 cykler (reservera bussen) + 1 cykel (skicka adress) + 150 ns (hämta två ord)/30 ns (busscykeltid) cykler + 2 cykler (överför två ord på bussen)) * 30 ns (busscykeltid) = 300 ns = 300 ns / (10 ns/processorcykel) = 30 processorcykler = miss penalty

Stoppcyklar per instruktionshämtning för 2 ord/block = missannolikhet * misspenalty = 2% * 30 cykler = 0,6 cykler

Hämtningstid för block om 8 ord = (2 cykler (reservera bussen) + 1 cykel (skicka adress) + 150 ns (hämta två ord)/30 ns (busscykeltid) cykler + 4*2 cykler (överför två ord på bussen, hämta nästa två ord i minnet)) * 30 ns (busscykeltid) = 480 ns = 480 ns / (10 ns/processorcykel) = 48 processorcykler = miss penalty

Stoppcyklar per instruktionshämtning för 8 ord/block = missannolikhet * misspenalty = 1% * 48 cykler = 0,48 cykler

8 ord/block ger minst antal cykler då processorn måste stoppas

- d. Under en tid T hämtas $T \cdot f / \text{CPI}$ (f = processorfrekvens = 100 MHz) instruktioner. $\text{CPI} = \text{CPI}_0 + \text{stoppcyklar/instruktionshämtning}$, där CPI_0 är CPI utan inverkan av instruktionshämtningsmissar. Stoppcyklar/instruktionshämtning fås från förra deluppgiften.

En instruktionsmiss belastar bussen under alla busscykler beräknade ovan utom två då bussen reserveras. Belastning på bussen (andel av tiden som bussen reserveras) under en tid T blir:

antal hämtade instruktioner under tiden T * missannolikhet * busstid/miss / T = $100 \text{ MHz} / (\text{CPI}_0 + \text{stoppcyklar/instruktionshämtning}) * \text{missannolikhet} * \text{busstid/miss}$

Belastning vid 2 ord/block = $100 \text{ MHz} / (\text{CPI}_0 + 0,6) * 2\% * 8 / (33 \text{ MHz}) = 0,48 / (\text{CPI}_0 + 0,6)$

Belastning vid 8 ord/block = $100 \text{ MHz} / (\text{CPI}_0 + 0,48) * 1\% * 14 / (33 \text{ MHz}) = 0,42 / (\text{CPI}_0 + 0,48)$

CPI_0 kan antas opåverkad av blockstorleken för instruktionshämtning, så vi ser efter vilka värden CPI_0 måste ha för belastningen ska bli störst med 2 ord/block:

$0,48 / (\text{CPI}_0 + 0,6) > 0,42 / (\text{CPI}_0 + 0,48) \Rightarrow \text{CPI}_0 > 0,36$

Eftersom processorn ej är superskalär, så vet vi att CPI_0 måste vara större än 1, och alltså ger **8 ord/block** minst andel av tiden som bakplansbussen är reserverad för läsningar till instruktionscachet.

5.

Deluppgift	1	X	2
a	1		
b			2
c		X	
d		X	
e	1		
f	1		
g			2
h	1		
i	1		
j			2
k		X	
l		X	