CHALMERS

2021-05-24

Tentamen i EDA333 (DIT 122) Datorsystemteknik

Tid: 3 juni, 8:30 - 13:00 (Canvas)

Ansvarig för examination: Per Stenström, Tel: 0730-346 340

Tillåtna hjälpmedel: Lärobok och föreläsningsanteckningar

Tentamensvisning: Mer information om detta kommer att publiceras på Canvas.

Betygsinterval:

- **Underkänd**: Resultat < 24
- **Betyg 3**: 24 <= Resultat < 36
- **Betyg 4:** 36 <= Resultat < 48
- **Betyg 5:** 48 <= Resultat

Följande gäller:

- Kursbetyg är en sammanvägning av betyg på tentamen och projektuppgift. För godkänt måste båda moment vara godkända. Därutöver kommer slutbetyget att bestämmas genom att bonuspoäng motsvarande 6p och 12p adderas till tentapoängen vid betyg 4 och 5, respektive, på projektuppgiften.
- Då icke-svensk talande assistenter deltar i tentarättning ber jag om svar på engelska.
- Då flera assistenter är med vid tentarättning ha endast en uppgift per blad.
- Läs i övrigt noga anvisningarna för on-line tentamen som finns på Canvassidan för denna examen.

Lycka till!

Per Stenström

CHALMERS UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
412 96 Göteborg
Visiting address: Rännvägen 5
Phone: 031-772 1761 Fax: 031-772 3663
Org. Nr: 556479-5598
E-mail: pers@chalmers.se

[Generell uppmaning:
Om du finner att nödvändiga fakta inte finns angivet för att lösa en uppgift välj då att antingen 1) fråga läraren vid hans/hennes besök eller 2) gör dina egna antaganden. Dessa kommer att godkännas om de är nödvändiga för att lösa uppgiften. Var noga att alltid noga motivera dina svar.]

Uppgift 1

Följande kod summerar två vektorer (A och B) och lägger resultatet i en tredje vektor (C):

```
int A[200], B[200] och C[200]
for (i=1; i<200; i++)
    C[i]=A[i]+B[i];
```

a) Koda denna högnivåkod i MIPs assembly. De enda värden som finns i registren före exekvering är basadresserna för A, B och C, i $a0, $a1 och $a2, respektive. Följ register- och anropskonventionen för MIPS processorn. Kommentera koden flitigt. **(8 poäng)**
b) Hur många accesser (läsningar och skrivningar) sker till dataminnet och till instruktionsminnet vid exekvering av programmet. (**4 poäng**)

Uppgift 2

Betrakta en 5-stegs pipeline (F-Fetch, D-Decode, E-Execute, M-Memory Access och W-Write Back) som vi gått igenom i kursen. Till skillnad mot denna finns det dock bara ett gemensamt minne för instruktioner och data. För det andra, saknas det forwarding så det enda sättet att lösa upp datakonflikter är genom att stalla en instruktion till en föregående har skrivit tillbaka resultatet i registerbanken. Man får anta att en instruktion kan läsa det tillbakaskrivna värdet i samma cykel. För det tredje kan hoppadress och hoppvillkor beräknas redan i D-steget och hoppinstruktionerna ger en fördröjning så att instruktionerna som hämtas tills hoppadress och villkor beräknats kommer att utföras.

Betrakta följande MIPS-kod:

```
loop:  lw  $t1, 0($a1)
       lw  $t2, 0($a2)
       add  $t3, $t1, $t2
       sw  $t3, 4($a1)
       addi  $a1, 4
       addi  $a2, 4
       addi  $t0, 1
       bne  $t0, $t5, loop
```

a) Upprätta ett pipelinediagram (X-axel tid i klockcykler och Y-axel instruktionerna i koden) som visar hur varje instruktion i koden ovan inom **en** iteration förflyttar sig genom pipelinestegen. Om den behöver stoppas markeras en ´stall´ cykel med X. (**6 poäng**)

b) En kompilator kan förflytta instruktionerna så att antalet ´stall´cykler kan minskas. Visa hur antalet ´stall´cykler kan reduceras till en enda genom förflyttning. (**4 poäng**)

c) Hur många ´stall´cykler kan elimineras om man hade lagt till forwarding mellan EX-steget och D-steget och mellan ME-steget och D-steget? (**2 poäng**)

Uppgift 3

Det nya datorföretaget Orange har lanserat en ny dator som ska vara bra på AI beräkningar. För en specifik AI tillämpning som många kunder använder gäller följande. Programmet består av två faser: en initierings- och en iterationsfas. I initieringsfasen exekveras 2 miljoner instruktioner och CPI är 1,5 och det finns 200 000 instruktioner som accessar minnet. I den iterationsfasen exekveras 100 000 instruktioner i varje iteration och det totala antalet iterationer är 10. CPI är 2 och det finns 20000 instruktioner i varje iteration som accessar minnet.

Genom mätningar har man funnit följande

- Missrate i datacachen är 10% i initieringsfasen och 20% i den iterativa fasen och tiden att hantera en miss är 10 cykler
- Missrate i instruktionscachen är 4% i såväl initieringsfasen som den iterativa fasen och tiden att hantera en miss är 10 cykler
- Klockfrekvensen är 1 GHz

a) Vad är exekveringstiden för programmet? (**3 poäng**)

b) Hur mycket bidrar datacachen till CPI i initerings- och i den iterativa fasen? (**6 poäng**)

c) Hur mycket bidrar instruktionscachen till CPI i initerings- och i den iterativa fasen? (**3 poäng**)

d) Om vi hade haft ett idealt minnessystem utan cache missar, vilket CPI uppnås? (**6 poäng**)

Uppgift 4

Företaget Orange har funnit att deras cacheminnessystem inte levererar den beräkningsprestanda som de hade hoppats på. De hoppas att du kan hjälpa dem.

Deras cacheminnessystem består av en tvånivås cachehierarki med en 64 KiB förstanivå- och en 1 MiB andranivåcache. Accesstiden för förstanivåcachen är 1 nanosekund, accesstiden för andranivåcachen är 10 nanosekunder och accesstiden för primärminnet är 100 nanosekunder. Blockstorleken för båda cacharna är 64 B. I övrigt gäller följande:

- Förstanivåcachen är en tvåvägs set-associativ cache
- Andranivåcachen är en fyrvägs set-associativ cache
- En minnesadress innehåller 64 bitar
- Processorn är kopplad till cachehierarkin och exekverar ett program där 30% av instruktionerna accessar data med en träffsannolikhet i första- och andranivåcachen på 99% och 50%, respektive. Motsvarande träffsannolikheter för instruktioner är 98% och 70%, respektive. Idealt CPI är 1 om alla dataaccesser hanteras av förstanivåcachen.
- Processorns klockfrekvens är 1 GHz.

a) Hur många bitar finns i tag fältet i första- och andranivåcachen? (**6 poäng**)

b) Hur lång tid tar det att hantera en miss i första- och andranivåcachen? (**4 poäng**)

c) Hur stor del av exekveringstiden går åt att hantera missar i cachehierarkin? (**8 poäng**)

****** Lycka till! ******

Lösningar till examen i EDA333/DIT122 2021-06-03

Uppgift 1

a)

```
loop:  lw  $t1, 0($a1)        # läs in A[i] i t1
       lw  $t2, 0($a2)        # läs in B[i] i t2
       add  $t3, $t1, $t2     # lägg A[i] + B[i] i t3
       sw  $t3, 4($a3)        # skriv tillbaks t3 i C[i]
       addi  $a1, 4           # öka adressregister för A[i]
       addi  $a2, 4           # öka adressregister för B[i]
       addi  $a3, 4           # öka adressregister för C[i]
       addi  $t0, 1           # i=i+1;
       bne  $t0, $t5, loop    # t5 antas innehålla 200. Hoppa om i < 200
```

b)

Instruktionsminnet: Varje iteration innehåller 9 instruktioner och det utförs 200 iterationer. Alltså accessas instruktionsminnet 9 x 200 = 1800 gånger

Dataminnet: I varje iteration utförs två läsningar och en skrivning. Alltså utförs 3 x 200 = 600 accesser till dataminnet.

Uppgift 2

Låt oss numrera instruktionerna:

```
I1:loop:  lw  $t1, 0($a1)
I2:        lw  $t2, 0($a2)
I3:        add  $t3, $t1, $t2
I4:        sw  $t3, 4($a1)
I5:        addi  $a1, 4
I6:        addi  $a2, 4
I7:        addi  $t0, 1
I8:        bne  $t0, $t5, loop
```

a)

```
I1:loop:  lw  $t1, 0($a1)
I2:       lw  $t2, 0($a2)
I3:       add  $t3, $t1, $t2
I4:       sw  $t3, 4($a1)
I5:       addi  $a1, 4
I6:       addi  $a2, 4
I7:       addi  $t0, 1
I8:       bne  $t0, $t5, loop
```

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C 13	C 14	C 15	C 16
I1	F	D	E	M	W											
I2		F	D	E	M	W										
I3			F	D	X	X	E	M	W							
I4				F	D	X	X	X	X	E	M	W				
I5					F	D	X	X	X	X	E	M	W			
I6						F	D	X	X	X	X	E	M	W		
I7							F	D	X	X	X	X	E	M	W	
I8								F	D	X	X	X	X	E	M	W

Det tar 16 cykler tills sista instruktionen nått W steget.

b)

Genom att flytta I5 och I7 mellan I2 och I3 kan två ´stall´cykler elimineras. Genom att också flytta I4 efter hoppinstruktionen sparas en ´stall´ cykler Men notera att vi måste korrigera offset i I4 eftersom I5 har flyttats före I4.

```
I1:loop:  lw  $t1, 0($a1)
I2:       lw  $t2, 0($a2)
I7:       addi  $t0, 1
I5:       addi  $a1, 4
I3:       add  $t3, $t1, $t2
I6:       addi  $a2, 4
I8:       bne  $t0, $t5, loop
I4:       sw  $t3, -4($a1)
```

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
I1	F	D	E	M	W							
I2		F	D	E	M	W						
I7			F	D	E	M	W					
I5				F	D	E	M	W				
I3					F	D	E	M	W			
I8						F	D	E	M	W		
I6							F	D	E	M	W	
I4								F	D	E	M	W

c)

Utan kodförflyttning hade det endast varit en ´stall´cykel mellan I2 och I3 samt en stall cykel efter I8, dvs två ´stall´cykler.

Uppgift 3

a) Vi använder $T_{CPU} = IC \times CPI \times T_c$

Initieringsfasen: IC=$2x10^6$, CPI=1,5 och T_c=1 x 10^{-9}s
$T_{CPU,init} = 2x10^6x1,5x1x10^{-9} = 3x10^{-3}$s = 3 millisekunder.

Iterationsfasen: IC= 10 x 100000 = $1x10^6$, CPI=2, T_c=1 x 10^{-9}s
$T_{CPU,iter} = 1x10^6x2x1x10^{-9} = 2x10^{-3}$s = 2 millisekunder.

$T_{CPU} = T_{CPU,init} + T_{CPU,iter}$=5 millisekunder

b)
Initieringsfasen: MR=0,1 och miss penalty är 10 cykler. Andelen data accesser är 200 000/2000000=0,10. Alltså blir tillskottet till CPI 0,10x0,1*10 = 0,1
Iterationsfasen: MR=0,2 och miss penalty är 10 cykler. Andelen data accesser är 200 000/1000000=0,2. Alltså blir tillskottet till CPI 0,2x0,2x10 = 0,4

c)
Initieringsfasen: MR=0,04 och miss penalty är 10 cykler. Alltså blir tillskottet till CPI 0,04*10 = 0,4
Iterationsfasen: MR=0,04 och miss penalty är 10 cykler. Alltså blir tillskottet till CPI 0,04*10 = 0,4

d)

Initieringsfasen: Ursprungligt CPI är 1,5. Detta minskas med inverkan av data- och instruktionscacharna framtagit i b) och c) CPI = 1,5 -0,1-0,4 = 1

Iterationsfasen: Ursprungligt CPI är 2. Detta minskas med inverkan av data- och instruktionscacharna framtagit i b) och c) CPI = 2 – 0,4 – 0,4 = 1,2

Uppgift 4

a)

Förstanivåcachen: Det finns 64 KiB/64 B = 2^{10} block i cachen. Men den är tvåvägs set-associativ så det finns 2^9 set. En minnesadress är 64 bitar varav 6 bitar används som offset inom ett block 9 bitar till att välja set och resten är taggbitar dvs 64-9-6 = 49 bitar

Andranivåcachen: Det finns 1 MiB/64 B = 2^{14} block i cachen. Men den är tvåvägs set-associativ så det finns 2^{13} set. En minnesadress är 64 bitar varav 6 bitar används som offset inom ett block 13 bitar till att välja set och resten är taggbitar dvs 64-13-6 = 45 bitar

b)
Fall 1: miss i förstanivåcachen men träff i andranivåcachen: 1 ns + 10 ns = 11 ns

Fall 2: miss i förstanivåcachen och i andranivå cachen: 11ns + 10 ns + 100 ns = 121 ns

c)
CPI med hänsyn tagen till **instruktionsmissar** i första- och andranivåcachen är

Sannolikhet för fall 1 i b) är 0,02x0,7 = 0,014 och för fall 2 är 0,02x0,4 = 0,008

CPI = 1 + 0,014x10 + 0,008x100 = 1 + 0,94

CPI med hänsyn tagen till **datamissar** i första- och andranivåcachen är

Sannolikhet för fall 1 i b) är 0,05x0,5 = 0,025 och för fall 2 är 0,05x0,5 = 0,025

CPI = 1 + 0,3x0,025x10 + 0,3x0,025x100 = 1 + 0,825

CPI_{tot}=1 + 0,94 + 0,825 = 2,765

Alltså tillbringar programmet andelen (2,765 – 1)/2,765 =0,64 dvs 64% av tiden att hantera missar.