Lösningar till tentamen i kursen EDA330

Datorsystemteknik

31/5 1997

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. $T = I * CPI * T_C$. $I = T/(CPI* T_C) = T*f/CPI = 60*10^{-3}*50*10^6/1,5 =$ **$2*10^6$**

b. Extra tid = $4*10^{-3}$ s = antal läs/skriv-instruktioner * miss rate * miss penalty cycles * cycle time = I_{ls} * (1-0,95) * 10 / ($50*10^6$). I_{ls} = **$0,4*10^6$**.

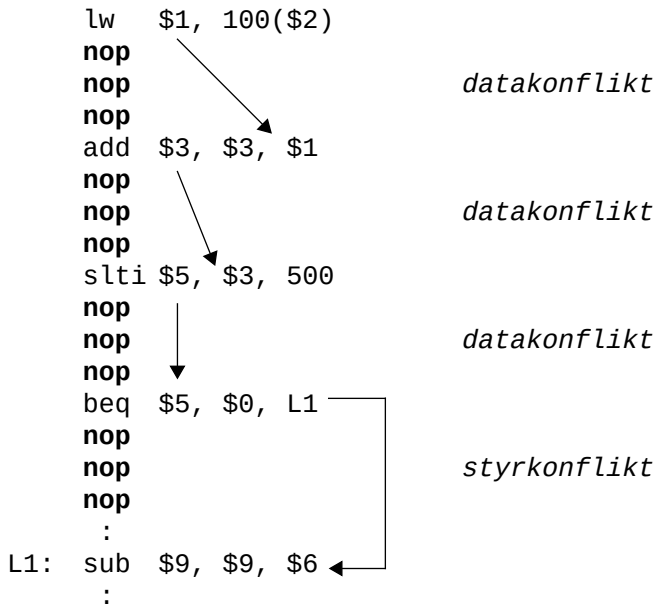
c. $24_{10} = 1100_2 = 1.100_2 * 2^4$. Sign = 0 (positivt). Exponent (8 bitar) = $4_{10}+127_{10} = 131_{10} = 10000011_2$. Significand (32-1-8=23 bitar) = 100... Hela talet i IEEE 754 single precision = sign, exponent, significand = **$01000001100000000000000000000000_2$**.

d. `001001 01000 00101 0000000010000001 = op=001001=9=addiu (format I), rs=01000=8, rt=00101=5, imm=0000000010000001=129.` Instruktionen är **addiu $5, $8, 129 som adderar det teckenlösa talet 129 till innehållet i register 8 och lägger resultatet i register 5**.
`100011 00000 00010 0000000010000001 = op=100011=35=lw (format I), rs=00000=0, rt=00010=2, addr=0000000010000001=129.` Instruktionen är **lw $2, 129($0) som hämtar data från adressen 129+$0=129 och lagrar i register 2**.

e. 65540 = 65536+4, laddas med **`lui $8, 1; addi $8, $8, 4`**.
65536 laddas med **`lui $8, 1`**.
4 laddas med **`addi $8, $0, 4`**.
-16 laddas med **`addi $8, $0, -16`**.

2.

a. Lägg till tre nop-instruktioner efter lw, add, slti (pga datakonflikter) och efter beq (pga styrkonflikt). Alltså:



b. Antag lw hämtas i cykel 0. Stalling innebär tre extra cykler fördröjning efter lw, add, slti, beq (nop eller "bubbla" läggs in av hårdvaran efter ID-steget på motsvarande sätt som i deluppgift a). Det går alltså 4 cykler per instruktion fram tills sub-instruktionen hämtas i cykel 0+4*4=16. Det tar sedan ytterligare 5 cykler (en för varje pipelinesteg) tills sub lämnat WB. Totala antalet cykler blir alltså 16+5=**21**.

c. Med data forwarding från MEM och WB till EX kan alla datakonflikter undvikas helt utom konflikten mellan lw och add som fortfarande kräver en cykel stalling (två stall-cykler försvinner dock). Styrkonflikten i samband med beq kan inte lösas på detta sätt. Totalt försvinner alltså 2+3+3=8 stallcykler, så totala antalet cykler minskar med **8** till 13.

d. Branch prediction (t.ex. assume branch not taken, assume branch taken, eller history-based prediction) med tömning av pipelinen vid felaktig gissning, eller tidigarelagd beräkning av hoppadress och hoppvillkor. **Se kursboken**.

3.

a. 1 block = 4 ord = 16 byte = 128 bitar. Av adressen krävs alltså 4 bitar i block-offset. Cacheminnet kan lagra 16 KB/16B = 1024 block. Tvåvägs associativitet innebär 2 block/set, och alltså totalt 512 set i cacheminnet. Av adressen krävs därför 9 (2^9 = 512) indexbitar för att hitta i vilket set ett block kan ligga. Det finns maximalt 32 MB primärminne så de fysiska adresserna som används av cachet måste vara 25 bitar breda (2^{25} B=32 MB). För varje block i cacheminnet måste därför 25-9-4 = 12 bitar tag lagras. För varje block krävs dessutom en valid bit och en dirty bit (pga write-back-strategin). Totalt krävs alltså 128+12+1+1 = 142 bitar/block. Med 1024 block blir totala cache-storleken 1024*142 = **145 408 bitar**.

b. Sidoffset kräver 14 bitar (2^{14} B = 16 KB). Virtuella sidnummer kräver därmed 32-14 = 18 bitar, vilket motsvarar 2^{18} = 256 K sidor. Fysiska sidnummer kräver 25-14 = 11 bitar. I sidtabellen ska för varje virtuell sida kunna lagras fysiskt sidnummer (11 bitar), valid bit, dirty bit, protection bit, och två bitar för utbytesalgoritmen, dvs totalt 16 bitar = 2 byte. Varje sidtabell kräver därmed 256 K sidor * 2 B/sida = 512 KB. Med 16 processer (som var och en har sin egen sidtabell) krävs totalt 16 * 512 KB = **8 MB** för sidtabeller.

c. TLB kan lagra information om 32 virtuella sidor. Den information som behöver lagras är fysiskt sidnummer, protection bit, och dirty bit (talar om ifall sidan uppdaterats), totalt 13 bitar. För varje sida måste också lagras en valid bit och en tag. Eftersom TLB är fullt associativ och uppslagning sker med virtuella sidnumret som adress, så måste hela virtuella sidnumret (18 bitar) lagras som tag. Alltså måste 11 + 1 + 1 + 1 + 18 = 32 bitar lagras för varje plats i TLB, och totala antalet bitar i TLB blir 32*32 = **1024 bitar**.

d. För att indexera cacheminnet krävs de 4+9 = 13 minst signifikanta bitarna av adressen. Den del av de virtuella adresserna som inte behöver översättas, sidoffset, är 14 bitar. Uppslagningen i cacheminnet kan därför påbörjas parallellt med översättningen av sidnummer. Översättningen med hjälp av TLB tar 20 ns och är alltså klar i god tid för tag-jämförelsen i cache (vilken sker i slutet av de 30 ns som en cache-uppslagning enligt uppgift tar). Vid träff i TLB och cache bör därför en minnesåtkomst klaras på 30 ns. Då denna tid enligt uppgiften definierar en processorklockcykel, kan processorn som bäst köra med **33 MHz** klockfrekvens.

4.

a. Bussen utnyttjas effektivast vid överföring av 4 dataord åt gången. Fyra ord överförs då på fem bussyckler, och det går 20 * 10^6 busscykler/s (20 MHz). Maximal databandbredd för bussen är alltså 4 ord/5 cykler * 4 B/ord * 20 * 10^6 busscykler/s = **64 MB/s**.

b. Processorn utför 48 MHz/1,5 = 32 * 10^6 instruktioner/s. Dela upp bandbreddskravet i bidrag från läsningar i instruktionscache, läsningar i datacache, och skrivningar i datacache. Läsning i instruktionscache: 32 * 10^6 läsningar/s * 0,01 missar/läsning = 0,32 * 10^6 missar/s, 1 block=4 ord=16 B per miss, 16 B/miss * 0,32 * 10^6 missar/s = 5,12 MB/s. Läsning i datacache: 0,2 läsningar/instruktion * 32 * 10^6 instruktioner/s * 0,05 missar/läsning * 16 B/miss = 5,12 MB/s. Skrivning i datacache: 0,1 skrivningar/instruktion * 32 * 10^6 instruktioner/s = 3,2 * 10^6 skrivningar/s, write-through innebär 1 ord=4 B på bussen för varje skrivning, 3,2 * 10^6 skrivningar/s * 4 B/skrivning = 12,8 MB/s. Totalt databandbredd för cacheåtkomster 5,12 + 5,12 + 12,8 MB/s = **23 MB/s**.

c. Överföring av 16 KB tar 2 ms + 20 ms + 16 KB/(2 MB/s) = 30 ms. På 30 ms överförs 2*16 KB, vilket leder till en effektiv databandbredd på 32 KB/30 ms = **1,1 MB/s**.

d. Här måste beaktas hur stor andel av tiden som bussen är upptagen av trafik till och från cacheminnen. Dela som tidigare upp i tre bidrag. Läsningar i instruktionscache tar 5 busscykler/miss * 1/(20 * 10^6) s/busscykel = 0,25 μs/miss, missar inträffar i genomsnitt varje 1/0,32 * 10^6 s, och tar därför upp 0,25 * 10^{-6}* 0,32 * 10^6 = 8% av tiden på bussen. Samma gäller för läsningar i datacache som också tar upp 8% av tiden. Skrivningar i datacache tar upp 32% av tiden. Totalt är alltså bussen upptagen med cacheminnestrafik 48% av tiden, och är därmed ledig för I/O 52% av tiden. Vid I/O utnyttjas bussens maximala databandbredd 64 MB/s. I/O kan därför totalt tillåtas en databandbredd på 0,52 * 64 MB/s = 33,3 MB/s. Det innebär att bakplansbussen kan belastas med maximalt 33,3/1,1 = **30 I/O-bussar**.

5.

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