

Lösningar till tentamen i kursen EDA330 för D och EDA370 för E

# Datorsystemteknik

25/8 2001

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

1.

- a. sign = 1 (negativt)  
 exponent field = 01011000<sub>2</sub> = 88<sub>10</sub> = exponent +127 =>  
 exponent = -39  
 significand field = 0010000000000000000000<sub>2</sub> =>  
 significand = 1.001<sub>2</sub>  
 Flyttalet är  $-1.001 * 2^{-39} = -1001 * 2^{-26} = -9 * 2^{-42}$
- b. Tvåkomplementsform, och talet är negativt. Ta fram det positiva talet genom att ta tvåkomplementet:  
 Invertera: 01010011111011111111111111111111  
 Addera 1: 01010011111100000000000000000000  
 Detta tal kan skrivas  $2^{30} + 2^{28} + 2^{25} + 2^{24} + 2^{23} + 2^{22} + 2^{21} + 2^{20} =$   
 $(2^{10} + 2^8 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0) * 2^{20} =$   
 $(1024 + 256 + 32 + 16 + 8 + 4 + 2 + 1) * 2^{20} = 1343 * 2^{20}$   
 Det sökta talet är alltså  $-1343 * 2^{20}$
- c. 101011 00000 10000 0000000000000000 = op=101011=43=sw (format I),  
 rs=00000=0 (\$zero), rt=10000=16 (\$s0), imm=0000000000000000=0.  
 Instruktionen är **sw \$s0, 0(\$zero)**.
- d. **Se avsnitt 4.8 i kursboken.**
- e.

1 0 1 1	
X 0 1 1 0 0	
-----	
0 0 0 0	00 = gör inget
0 0 0 0	00 = gör inget
- 1 0 1 1	10 = subtrahera
0 0 0 0	11 = gör inget
+ 1 0 1 1	01 = addera
-----	
1 0 0 0 0 1 0 0	

2.

- a. Först måste tiden att få tillgång till bussen för en blocköverföring till instruktionscache beräknas. 60% av tiden är bussen ledig och det tar noll cykler. Övrig tid måste en blocköverföring av fyra ord avslutas först. En sådan överföring tar (1 cykel för att skicka adress + 75 ns (hämta två ord)/15 ns (busscykeltid) cykler + max(2 cykler (överför två ord på bussen), 25 ns/15 ns (hämta nästa två ord)) + 2 cykler (överför två ord på bussen)) = 1+5+2+2 busscykler = 10 busscykler. I genomsnitt återstår halva överföringen, så statistiskt

tar det  $0,6*0 + 0,4*10/2 = 2$  cykler att få tillgång till bussen för en instruktionshämtning.

Hämtningstid för instruktionsblock om 2 ord = (2 cykler (få bussen) + 1 cykel (skicka adress) + 5 cykler (hämta första två ord) + 2 cykler (överför två ord på bussen)) \* 15 ns (busscykeltid) = 150 ns = 150 ns/(5 ns/processorcykel) = 30 processorcykler = miss penalty vid 2 ord/block.

Hämtningstid för instruktionsblock om 4 ord = (2 cykler (få bussen) + 1 cykel (skicka adress) + 5 cykler (hämta första två ord) + 2 cykler (överför två ord, hämta nästa två) + 2 cykler (överför två ord)) \* 15 ns (busscykeltid) = 180 ns = 180 ns/(5 ns/processorcykel) = 36 processorcykler = miss penalty vid 4 ord/block.

På motsvarande sätt blir miss penalty vid 8 ord/block 48 processorcykler. I fallet 16 ord/block kan hela blocket inte överföras med en minnesåtkomst, utan flera busstransaktioner krävs. Det bästa är då att göra två överföringar om 8 ord. Eftersom instruktionsöverföringar alltid har högst prioritet på bussen kan dessa ske direkt efter varandra, så miss penalty vid 16 ord/block blir  $2*48$  processorcykler - (2 busscykler \* 3 processorcykler/busscykel) = 90 processorcykler.

Antalet stoppcyklar per instruktionshämtning kan räknas ut som miss rate \* miss penalty. Vid 2 ord/block blir antalet stoppcyklar  $4%*30 = 1,2$ . Vid 4 ord/block blir antalet stoppcyklar  $2%*36 = 0,72$ . Vid 8 ord/block blir antalet stoppcyklar  $1%*48 = 0,48$ . Vid 16 ord/block blir antalet stoppcyklar  $0,8%*90 = 0,72$ . Minst antal stoppcyklar fås alltså vid **8 ord/block**.

- b. Under en tid T hämtas  $T*f/CPI$  ( $f$  = processorfrekvens = 200 MHz) instruktioner.  $CPI = CPI0 + \text{stoppcyklar/instruktionshämtning}$ , där  $CPI0$  är  $CPI$  utan inverkan av instruktionshämtningsmissar. Enligt uppgiften är  $CPI0 = 1,6$ . 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 =  $200 \text{ MHz}/(1,6 + \text{stoppcyklar/instruktionshämtning}) * \text{missannolikhet} * \text{busstid/miss}$

Belastning vid 2 ord/block =  $200 \text{ MHz}/(1,6 + 1,2) * 4% * 8/(67 \text{ MHz}) = 4%*8*3/(1,6 + 1,2) = 34%$ .

Belastning vid 4 ord/block =  $2%*10*3/(1,6 + 0,72) = 26%$ .

Belastning vid 8 ord/block =  $1%*14*3/(1,6 + 0,48) = 20%$ .

Belastning vid 16 ord/block =  $0,8%*28*3/(1,6 + 0,72) = 29%$ .

Med **8 ord/block** fås minst andel av tiden då bussen är reserverad för läsning av instruktioner.

- c. Vi räknar om antalet stoppcykler för 4 ord/block från deluppgift a enligt: Väntetid att få bussen blir  $0,4 * 1/2 * (1 \text{ cykel för att skicka adress} + 75 \text{ ns (hämta två ord)}) / 10 \text{ ns (busscykeltid)}$  cykler +  $\max(2 \text{ cykler (överför två ord på bussen), } 25 \text{ ns} / 10 \text{ ns (hämta nästa två ord)}) + 2 \text{ cykler (överför två ord på bussen)} = 0,2 * (1+8+3+2) = 2,8$  busscykler. Observera att antalet busscykler för olika operationer måste vara ett helt antal. Hämtningstid för instruktionsblock om 4 ord bli alltså  $2,8 + 14 = 16,8$  busscykler =  $168 \text{ ns} = 168 \text{ ns} / (5 \text{ ns/processorcykel}) = 34$  processorcykler = miss penalty vid 4 ord/block. Antal stoppcykler blir alltså  $2\% * 34 = 0,68$ . Därigenom ökar prestanda med en faktor  $(1,6+0,72)/(1,6+0,68) = 1,02$ . Bussbelastning vid 4 ord/block =  $200 \text{ MHz} / (1,6 + 0,68) * 2\% * 14 / (100 \text{ MHz}) = 2\% * 14 * 2 / (1,6 + 0,68) = 25\%$ .

**Processorprestanda ökar med ungefär 2%, och bussbelastningen minskar med ungefär 1%.**

3.

- a. Datorerna med 1 GHz processorer blir 2 ggr snabbare. Undersök om datorerna med 600 MHz processorerna är snabbare. Utgå från formeln för CPU-tid:

$$T = I * CPI * T_C$$

Kalla totaltiden för konvertering med det gamla systemet för  $T_{\text{gammal}}$ :

$$T_{\text{gammal}} = T_{\text{multimedia,gammal}} + T_{\text{övriga,gammal}}$$

$$\text{där } T_{\text{multimedia,gammal}} = 0,4 * I_{\text{gammal}} * 3,0 / (500 * 10^6) \text{ och}$$

$$T_{\text{övriga,gammal}} = 0,6 * I_{\text{gammal}} * 1,0 / (500 * 10^6)$$

Totaltiden för det nya systemet,  $T_{\text{ny}}$ , blir:

$$T_{\text{ny}} = T_{\text{multimedia,ny}} + T_{\text{övriga,ny}}$$

$$\text{där } T_{\text{multimedia,ny}} = 0,1 * I_{\text{ny}} * 0,5 / (600 * 10^6) \text{ och}$$

$$T_{\text{övriga,ny}} = 0,9 * I_{\text{ny}} * 1,0 / (600 * 10^6)$$

Vi vet att  $0,9 * I_{\text{ny}} = 0,6 * I_{\text{gammal}}$  vilket medför att

$$I_{\text{ny}} = I_{\text{gammal}} * 0,6 / 0,9 = 2/3 * I_{\text{gammal}}$$

$$T_{\text{ny}} = (0,1 * 2/3 * I_{\text{gammal}} * 0,5 + 0,9 * 2/3 * I_{\text{gammal}} * 1,0) / (600 * 10^6)$$

$$\text{Uppsnabbning} = T_{\text{gammal}} / T_{\text{ny}} = ((0,4 * I_{\text{gammal}} * 3,0 + 0,6 * I_{\text{gammal}} * 1,0) / (500 * 10^6)) / ((0,1 * 2/3 * I_{\text{gammal}} * 0,5 + 0,9 * 2/3 * I_{\text{gammal}} * 1,0) / (600 * 10^6)) = ((0,4 * 3 + 0,6 * 1) / 500) / ((0,1 * 2/3 * 0,5 + 0,9 * 2/3 * 1) / 600) = \mathbf{3,41 \text{ ggr snabbare med datorerna med 600 MHz processorer.}}$$

- b. **Se kursboken.**

4.

- a. Antag sw hämtas i cykel 0. Stalling innebär tre extra cykler fördröjning efter lw, slti, beq (nop eller "bubbla" läggs in av hårdvaran efter ID-steget). Det går alltså 4 cykler per instruktion efter sw-instruktionen fram tills att add-instruktionen hämtas i cykel  $1+3*4=13$ . Det tar sedan ytterligare 5 cykler (en för varje pipelinesteg) tills add lämnat WB. Totala antalet cykler blir alltså  $13+5 = \mathbf{18}$ .
- b. Med data forwarding från MEM och WB till EX kan datakonflikten mellan slti och beq undvikas helt medan konflikten mellan lw och slti fortfarande kräver en cykel stalling (två stall-cykler försvinner dock). Styrkonflikten i samband med beq kan inte lösas på detta sätt. Totalt försvinner alltså  $2+3=5$  stallcykler, så totala antalet cykler minskar med **5** till 13.
- c. 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.**

5.

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