

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

Datorsystemteknik

17/1 2002

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

1.

- a. 200, 0, 204, 100, 208, 212, 216, 100, 220, 224, 200, 4, 204, 104, 208, 212, 216, 104, 220, 224, 200, 8, 204, 108, 208, 212, 216, 108, 220, 224, 200, 12, 204, 112, 208, 212, 216, 112, 220, 224, 200, 16, 204, 116, 208, 212, 216, 116, 220, 224.

- b. Block-offset = byte-adress mod 8 (2 ord/block = 8 byte/block)
 Block = byte-adress/8
 Index = block mod 4 ((16 ord/(2 ord/block))/(2 block/set) = 4 set)
 Tag = block/4.

c.

Set/block					0		1		2		3	
Byte	Block	Index	Tag	Hit	0	1	0	1	0	1	0	1
200	25	1	6	n			6					
0	0	0	0	n	0		6					
204	25	1	6	y	0		6					
100	12	0	3	n	0	3	6					
208	26	2	6	n	0	3	6		6			
212	26	2	6	y	0	3	6		6			
216	27	3	6	n	0	3	6		6		6	
100	12	0	3	y	0	3	6		6		6	
220	27	3	6	y	0	3	6		6		6	
224	28	0	7	n	7	3	6		6		6	

Hit rate = 4/10 = 40%.

2.

- a. **Se avsnitt 6.7 i kursboken och nedan.**
- b. Observation: snurran löper tre varv. $1 + 3 \cdot 5 + 1 = 17$ instruktioner ska exekveras. Inga stalls pga datakonflikter. Om rätt instruktion alltid hämtas efter hoppinstruktionen så fås inte heller några styrkonflikter. Första instruktionen utförs då i WB-steget i cykel 5, och sista instruktionen i cykel $5 + 16 = 21$. Utan styrkonflikter tar programmet alltså 21 cykler, och det återstår bara att för varje metod beräkna hur många extra cykler (t.ex pga stalls) som krävs.
 Always stall:
 I detta fall görs en stall till hoppinstruktionen utförts i MEM-steget. Det blir alltså ett tillägg av tre cykler för varje gång hoppinstruktionen utförs, dvs $3 \cdot 3 = 9$ extra cykler. **Vid always stall tar programmet 30 cykler.**

Assume not taken:

I detta fall gissar man att hoppet inte ska tas och hämtar därför instruktioner direkt efter hoppinstruktionen. Om det är en felaktig gissning, så upptäcks detta efter tre cykler som alltså blir bortkastade. I detta fall tas hoppet två gånger, och en gång inte. Det blir alltså två felaktiga gissningar som kostar sex extra cykler.

Vid assume not taken tar programmet 27 cykler.

Assume taken:

Detta är raka motsatsen till föregående fall, och eftersom hoppet utförs två gånger och passerar en gång, så blir effekten endast en felaktig gissning som kostar tre extra cykler. **Vid assume taken tar programmet 24 cykler.**

3.

a.
$$\text{CPU-tid}_{\text{FP}} = \text{CPI}_{\text{FP}} * I_{\text{FP}} * T_{\text{cFP}} = (5*0,2 + 4*0,1 + 16*0,05 + 2*0,65) * I_{\text{FP}} * 1/(500*10^6)$$

$$\text{CPU-tid}_{\text{IP}} = \text{CPI}_{\text{IP}} * I_{\text{IP}} * T_{\text{cIP}} = 1*(30*0,2 + 20*0,1 + 50*0,05 + 0,65) * I_{\text{IP}} * 1/f_{\text{IP}} \Rightarrow$$

$$f_{\text{IP}} = (30*0,2 + 20*0,1 + 50*0,05 + 0,65) / ((5*0,2 + 4*0,1 + 16*0,05 + 2*0,65) * 1/(500*10^6)) = 1,59 \text{ GHz}$$

Klockfrekvensen fördubblas på 18 månader $\Rightarrow f_{\text{ny}} = 2^{m/18} * f_{\text{gammal}} \Rightarrow$
 $2^{m/18} = f_{\text{ny}} / f_{\text{gammal}} \Rightarrow m = 18 * (\ln(f_{\text{ny}} / f_{\text{gammal}}) / \ln 2)$, f_{IP} initialt 1 GHz,
 skall nå 1,59 GHz $\Rightarrow m = 18 * (\ln(1,59/1) / \ln 2) = \text{ca } 12 \text{ månader.}$

b. Ökning för FP = $((5*0,2 + 4*0,1 + 16*0,05 + 2*0,65) * I_{\text{FP}} * 1/(500*10^6)) / ((5*0,1 + 4*0,06 + 16*0,02 + 2*0,82) * 1,3 * I_{\text{FP}} * 1/(500*10^6)) = 1,0$ (ingen skillnad)

Ökning för IP = $((1*(30*0,2 + 20*0,1 + 50*0,05 + 0,65) * I_{\text{IP}} * 1/f_{\text{IP}}) / ((1*(30*0,1 + 20*0,06 + 50*0,02 + 0,82) * 1,3 * I_{\text{IP}} * 1/f_{\text{IP}})) = 1,42 \text{ ggr}$

4.

a. Bussen utnyttjas effektivast vid överföring av 4 dataord åt gången. Fyra ord överförs då på fem busscykler, och det går $100 * 10^6$ busscykler/s (100 MHz). Maximal databandbredd för bussen är alltså $4 \text{ ord}/5 \text{ cykler} * 4 \text{ B/ord} * 100 * 10^6 \text{ busscykler/s} = 320 \text{ MB/s}$.
 Överföring av 64 KB tar $1 \text{ ms} + 10 \text{ ms} + 64 \text{ KB}/(10 \text{ MB/s}) = 17,4 \text{ ms}$. På 17,4 ms överförs $2*64 \text{ KB}$, vilket leder till en effektiv databandbredd på $128 \text{ KB}/17,4 \text{ ms} = 7,36 \text{ MB/s}$.
 Vid I/O utnyttjas bussens maximala databandbredd 320 MB/s. Eftersom bussen är ledig för I/O 56% av tiden kan I/O därför totalt tillåtas en databandbredd på $0,56 * 320 \text{ MB/s} = 179,2 \text{ MB/s}$. Det innebär att bakplansbussen kan belastas med maximalt $179,2/7,36 = 24 \text{ I/O-bussar}$.

b. **Se avsnitt 8.4 i kursboken.**

c. **Se kursboken.** Asynkron buss kräver ej distribution av klocksignal, tillåter kommunikation mellan komponenter med helt olika hastighet, och kan göras lång. Synkron buss kan göras snabbare, och får oftast en enklare protokollimplementering.

5.

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