EDA 332/DIT122 Datorsystemteknik

7 Juni 2019

1 a) Instruktionsminnet fyra gånger (eftersom det finns fyra instruktioner), och dataminnet två gånger (en gång vid lw och en gång vid sw).

b) MIPS kod:

```
        li      $t0, 1          # Start index till i
        li      $t5, 200        # Loop gräns
loop:
        lw      $t1, 0($a1)     # Load A[i-1]
        lw      $t2, 4($a2)     # Load B[i]
        add     $t3, $t1, $t2   # A[i-1] + B[i]
        sw      $t3, 4($a1)     # A[i] = A[i-1] + B[i]
        addi    $a1, 4          # Gå till i+1
        addi    $a2, 4          # Gå till i+1
        addi    $t0, 1          # Inkrementera index variabel
        bne     $t0, $t5, loop  # Jämför med loop gräns
```

2 a)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
addu	$2, $2, $3	F	D	E	M	W																				
l.d	$f0, 0($2)		F	.	.	D	E	M	W																	
mul.d	$f2, $f2, $f0					F	.	.	D	E	.	.	.	M	W											
l.d	$f8, 1000($2)								F	D	.	.	.	E	M	W										
mul.d	$f4, $f4, $f8									F	.	.	.	.	.	D	E	.	.	.	M	W				
add.d	$f6, $f2, $f4															F	.	.	.	.	.	D	E	.	M	W
s.d	$f6, 0($4)																								F	D
subui	$4, $4, 4																									F
beq	$2, $4, L46																									

b) En RAW (Read After Write) hazard via register $2 pga de båda första instruktionerna (addu och l.d). Kan elimineras om man inför forwardinglogik mellan utgången på ALU till dess ingångar.

3. (a)
Fyra ord a´ åtta bytes hanteras i varje DMA överföring. Varje DMA överföring kräver fem busscykler a´ 5 ns (en cykel för adress och fyra cykler för data). Den totala användbara bussbandbredden är $(4 \cdot 8)/(5 \cdot 5 \cdot 10^{-9}) = 1{,}28 \cdot 10^9$ B/s. Av denna bandbredd är 40% (=$5{,}12 \cdot 10^8$ B/s) tillgängligt för I/O.

Diskarna spinner med 7200 RPM = 7200/60 = 120 varv per sekund.

Vi vet inget om tid för initiering av DMA-överföringarna så vi antar den =0.

RW-huvudena passerar varje varv 500 sektorer innehållande 512 bytes vardera. Alltså kan varje RW-huvud generera $120 \cdot 500 \cdot 512 = 3{,}072 \cdot 10^7$ bytes/s. Vilket för 8 ytor per drive ger 245,76 * 10^6 bytes/s. Den tillgängliga bussbandbredden kan alltså då maximalt hantera $5{,}12 \cdot 10^8 / 245{,}76 \cdot 10^6 = 2{,}08$ disk drives. Dvs avrundat nedåt till 2 disk drives.

Varje spår på skivorna innehåller 500 sektorer med 512 bytes vardera, så $500 \cdot 512 = 256000$ bytes. Det finns 30000 spår alltså har vi $30000 \cdot 256000 = 7{,}68 \cdot 10^9$ bytes per skivyta. Per disk drive blir detta $4*2*7{,}68 \cdot 10^9 = 61{,}44 * 10^9$ bytes.

Totalt för 2 diskdrives = $2 \cdot 61{,}44 * 10^9 = 1{,}23 \cdot 10^{11}$ bytes.

(b)
Läs eller skriv av ett 8 KiB block tar $8192/(3{,}072 \cdot 10^7) = 2{,}66 \cdot 10^{-4}$ s.

Till detta kommer söktid (10 ms) och ett halvt varv rotationsfördröjning. (0,5/120 = $4{,}17 \cdot 10^{-3}$ s). Vi vet inget om ”controller overhead” så vi antar den är =0.

Andelen blir alltså (10 + 4,17)/(10 + 4,17 + 0,266) = 0,982 av den totala accesstiden.

(c)
Nu behövs bara söktid och rotationsfördröjning göras i 1% av alla accesserna. Vi räknar på 100 accesser då får vi att den totala accesstiden blir $10 + 4{,}17 + 100 \cdot 0{,}266 = 40{,}77$ ms. Av detta är 10 + 4,17 = 14,17 ms sök- och rotationstid, eller 14,17/40,77=34,8%.

4.
(a) Sidstorlek = 4KiB = 2^{12} -> #bitar = $\log_2 2^{12} = 12$ = 3 hex siffror -> PageOffset = 0x548
(b) Resterande bitar av den virtuella adressen = 0xBCDEF9876
(c) 36 - 12 bitar = 24 bitar
(d) 0x46844548
(e) Cache blockstorlek = 16Bytes = 2^4 -> #bitar = $\log_2 2^4 = 4$ -> CacheOffset = 0x8
(f) #blocks = Cachestorlek / blockstorlek = 256KiB/16 = $2^{18-4} = 2^{14}$
#sets = #blocks/associativitet = $2^{14}/2^4 = 2^{10}$ -> #bitar = $\log_2 2^{10} = 10$ bitar-> setIndex = 10 låga bitarna av blockadressen = 00 0101 0100 = 0x054
(g)
Fysisk address = 0x46844548

hex	0x4	6	8	4	4	5	4	8
binärt	0100	0110	1000	0100	0100	0101	0100	1000

bitfält 01000110100001000100010101001000
Cache Tag = 0x11A11, Set Index=0x054, CacheOffset=0x8