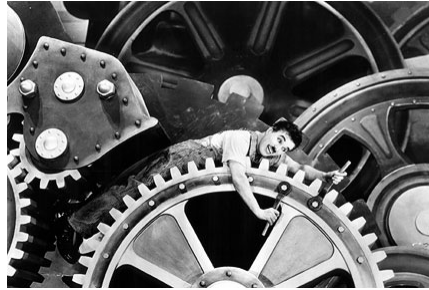


Tentamen onsdag 19 december 2007

Datorsystem 1 del 1 (1IT050) och Datorarkitektur 1 (1DT016)



Examinator:

- Karl Marklund
- 018 – 471 10 49
- 0704 – 73 32 17
- karl.marklund@it.uu.se

Tillåtna hjälpmedel:

- Penna
- Radergummi
- Linjal

Skrivningstid: 5 timmar.

Namn (efternamn först):

Namn-teckning:

Personnummer (10 siffror)

Jag läser:

- IT (1IT050)
- Elektro (1DT016)
- _____

Termin och år då jag först registrerades på kursen:

- Jag har kontrollerat att detta häfte innehåller 19 sidor med 9 stycken olika frågor om sammanlagt 45 poäng.

Klockslag för inlämning: _____

Följande gäller:

- Du får endast skriva dina svar på avsedd plats i detta häfte (linjerat område).
- Observera att lösningar som inte uppfyller kravet ovan ej kommer att rättas.

Bonus:

- De som följt kursen denna termin och blivit godkända på samtliga inlämningsuppgifter i MIPS-programmering får tillgodoräkna sig fråga-1.
- Bonuspoäng för seminarier (max 6 poäng) får du lägga till resultatet på denna tenta.

Gräns för godkänt: 22 poäng.

“It is by will alone I set my mind in motion.”

Piter De Vries
From the movie Dune by David Lynch

Fråga 1

(5p)

Du som är godkänd på samtliga inlämningsuppgifter i assemblerprogramering i MIPS (3 st för IT och 4 för Elektro) kan tillgodräkna dig denna uppgift, dvs du behöver **ej** lämna in någon lösning på denna uppgift utan får 5 poäng i bonus för väl utfört arbete.

Samtliga uppgifter skall vara godkända i Course Manager senast 2007-12-18 kl 23:59 för att du skall kunna tillgodoräkna dig denna uppgift.

Har du inte blivit godkänd på samtliga uppgifter skall du lämna in en lösning på denna uppgift.

Jag avser att:

- tillgodoräkna mig denna uppgift.
- lämna in en lösning på denna uppgift.

På nästa sida hittar du ett kort MIPS-program.

(1p) Du skall lista ut vad **subrutinen mystery** gör och skriva en kortfattad dokumentation vid kommentarfälten DESCRIPTION, INPUT och OUTPUT:

(1p) Skriv kortfattade men **meningsfulla** kommentarer vid fälten C1, C2, ..., C9.

Att direkt beskriva vad assemblerkoden gör, till exempel

```
add      $t1, $t1, $t3    # Register t1 = t1 + t3
```

är **inte** att betrakta som meningsfullt.

Vid testkörning i SPIM visar det sig att programmet inte fungerar som det är tänkt. En närmare analys visar att två buggar smugit sig in. Redogör för vilka dessa fel är och vilka ändringar i programmet du behöver göra för att det skall fungera som det var tänkt.

0.5p **Bugg 1:**

0.5p **Ändring:**

0.5p **Bugg 2:**

0.5p **Ändring:**

Vilket värde ligger på stackens topp när det fungerande programmet kört klart?

0.25p Decimalt: _____ 0.75p Hexadecimalt: _____

```

.data
array: .byte 13, 9, 7, 5
.text
.globl main
main:
    la    $t0, array

    addi  $sp, $sp, -4 # C1: _____
    sw   $t0, 0($sp)

    addi  $t2, $zero, 3 # C2: _____
    addi  $sp, $sp, -4
    sw   $t2, 0($sp)

    jal  mystery

    lw   $ra, 0($sp) # C3: _____
    addi $sp, $sp, 4

    addi  $sp, $sp, -4 # C4: _____
    sw   $v0, 0($sp)

    jr   $ra

#####
#
# DESCRIPTION: _____
#
# _____
#
# _____
#
# INPUT: _____
#
# _____
#
# OUTPUT: _____
#
# _____
#
#####

mystery:

    lw   $t0, 0($sp) # C5: _____
    addi $sp, $sp, 4

    lw   $t1, 0($sp) # C6: _____
    addi $sp, $sp, 4

# C7: _____

while:    beq    $t2, $t0, done

    lb   $t3, 0($t1) # C8: _____
    add  $v0, $v0, $t3

    addi $t1, $t1, 1 # C9: _____

```

Fråga 2

(4p)

I boken och på föreläsningar har vi diskuterat fyra olika principer vid design av hårdvara.

- (a) Redogör kortfattat för varje princip.
- (b) För varje princip, ge ett exempel från MIPS-arkitekturen där principen har använts som vägledning vid designen.



0.5p Princip 1

0.5p Exempel



0.5p Princip 2

0.5p Exempel



0.5p Princip 3

0.5p Exempel



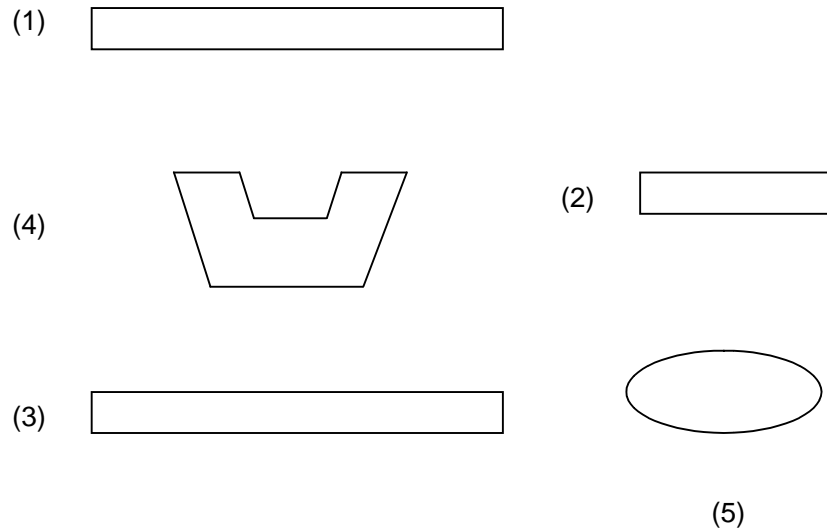
0.5p Princip 4

0.5p Exempel

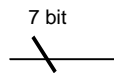
Fråga 3

(6p)

I figuren nedan finns schematiska bygg-stenar som du skall använda för att designa en multiplikator i hårdvara för tal om fyra bitar.



(2p) Sammanbind byggstenarna med "ledningar" och ange med ett sned-streck på detta sätt:



hur många bitar som "ledningen" avser.

Ange vad varje byggsten representerar och vilken funktion den har vid multiplikationen:

1 0.25p **Representerar:**

0.25p **Funktion:**

2 0.25p **Representerar:**

0.25p **Funktion:**

3 0.25p **Representerar:**

0.25p **Funktion:**

4 0.25p **Representerar:**

0.25p **Funktion:**

5 0.5p **Representerar:**

0.5p **Funktion:**

Antag att vi skall multiplicera följande två fyra-bitars tal 1011_2 och 0011_2 . Multiplikationen sker stegvis. För blocken 1,2 och 3, ange innehållet i varje steg.

	Enhet	Steg 1	Steg 2	Steg 3	Steg 4
0.25p	1				
0.25p	2				
0.25p	3				
0.25p	4				

Fråga 4

(2p)

Namnge och förklara vilka de fyra huvudsakliga faktorer är som avgör hur lång tid det tar att läsa data från en hårddisk.

(a) 0.1p **Namn:**

0.4p **Förklaring:**

(b) 0.1p **Namn:**

0.4p **Förklaring:**

(c) 0.1p **Namn:**

0.4p **Förklaring:**

(d) 0.1p **Namn:**

0.4p **Förklaring:**

Fråga 5

(5p)

Vilka två egenskaper hos program/programkod utnyttjas för att med hjälp av cache snabba upp skrivning och läsning till minnet.

Namnge dessa två egenskaper samt förklara vad de innebär.

Ge för varje egenskap ett konkret exempel på när denna egenskap ”visar” sig.

0.25p Egenskap 1:

0.25p Förklaring:

0.5p Exempel:

Vid implementation av ett cache-system kan vi påverka hur mycket av denna egenskap som kan utnyttjas genom att

0.5p ändra:

1p Motivering:

0.25p Egenskap 2:

0.25p Förklaring:

0.5p Exempel:

Vid implementation av ett cache-system kan vi påverka hur mycket av denna egenskap som kan utnyttjas genom att

0.5p ändra:

1p Motivering:

Fråga 6

(9p)

Adresser i minnet kan mappas till cachet på olika sätt. Namnge och förklara den grundläggande principen för tre olika sådana metoder.

0.25p **Namn:**

0.75p **Förklaring:**

0.25p **Namn:**

0.75p **Förklaring:**

0.25p **Namn:**

0.75p **Förklaring:**

(2p) Vilka för- och nackdelar finns med de olika metoderna ovan?

Vid mappning till cache delas adressen upp i **index** och **tag**. Vad betecknar dessa fält och hur används de?

1p **Index:**

1p **Tag:**

(1p) Vilket samband gör att det är enkelt att i hårdvara beräkna index-fältet?

(1p) Vilken förutsättning måste vara uppfylld för att detta ”trick” skall fungera?

Fråga 7

(5p)

När du skriver ett program i ett högnivåspråk genomgår detta program en rad översättningar innan det till slut resulterar i en serie instruktioner som hårdvaran kan utföra.

Beskriv i fyra steg vad som avgör hur lång tid det tar att exekvera ett program skrivit i ett högnivåspråk.

Resultatet av varje steg kan mätas i något som vi bör kunna mäta/kvantifiera. För varje steg, ange en lämplig benämning eller förkortning samt enhet.

0.25p Steg 1:

0.25p Benämning eller Förkortning samt enhet:

0.25p Steg 2:

0.25p Benämning eller Förkortning samt enhet:

0.25p Steg 3:

0.25p Benämning eller Förkortning samt enhet:

0.25p Steg 4:

0.25p Benämning eller Förkortning samt enhet:

(1.5p) Du skall nu sammanfatta hur lång tid det tar att köra ett program p i en matematisk formel. Använd de benämningar/förkortningar som du infört tidigare.

(0.5p) Visa att formeln stämmer med avseende på enheter, dvs resulterar i en enhet som används för storheten tid.

$$T_{cpu}(p) =$$

(1p) **Motivera att formeln ovan är rimlig:**

Fråga 8

(4p)

Redogör för skillnaden mellan **exception** och **interrupt**. Ange även ett exempel när ett exception och interrupt kan inträffa.

0.5p Exception:

0.5p Exempel:

0.5p Interrupt:

0.5p Exempel:

(0.5p) Vad händer i SPIM vid ett exception eller interrupt?

(0.5p) Hur kan vi i PIM skilja på olika typer av exceptions och interrupts.

(1p) Vad måste man vara observant på när man skriver kod som hanterar exceptions/interrupts?

Fråga 9

(5p)

(1p) Vilket problem med **single-cycle-implementation** löser man genom **multi-cycle-implementenation**?

(1p) Vilken är den grundläggande observationen som ligger till grund för metoden med **pipelining**?
