

Computer organisation

December 22, 2004 (9:30-12:30)



- This exam consists of 5 questions with subquestions. Each *subquestion* has the same weight.
- Start each question on a *new page*. Make sure to write your name and student number on each page.
- The use of books, notes or papers is not allowed, but a calculator is permitted (and, indeed, seems advisable).
- Make your answers brief and to the point, do *not* use pencil or red ink.

Question 1

We execute the following program (given in pseudo-assembly):

Code	Description
1. LOAD R1,(R3)	; load in register R1 the value stored at the memory address contained in R3
2. L1: SUB R1,R1,#1	; subtract 1 from the value in R1 - result goes in R1
3. ADD R2,R1,R2	; add contents of R1 and R2 - result goes in R2
4. BNEZ R1,L1	; if R1 does not equal zero, branch to L1
5. STORE R2,(R3)	; store contents of R2 in memory at address contained in R3

Assume that R3 contains the address of memory location 1024 (decimal) and the value stored in memory at 1024 is 1000 (decimal). R1 and R2 are initialised to 0.

(a) (i) What is the value that is loaded in R1 in instruction 1? (ii) How often will each instruction (1-5) be executed? (iii) Describe *in one sentence* what this code does.

The code is executed on a processor with a clock-rate of 1GHz which has the following timing information:

Type	CPI
ALU operations	1
Branches	3
Loads	3
Stores	2

(b) Assuming the processor runs just this code: (i) what is the overall CPI (clock ticks per instruction), and (ii) what is the total CPU time that is needed to execute the program?

The MIPS rate of a processor is defined as “millions of instructions executed per second”.

(c) What is the MIPS rate in this case?

(d) “The performance of two processors can be compared meaningfully just by looking at their clock rates, provided they have the same instruction set architecture.”

Do you agree with this statement? Why/why not?.

Question 2

(a) The memory word ‘101100011’ should be protected by a Hamming code. Construct the full code word (i.e., after the parity bits are added). Assume even parity.

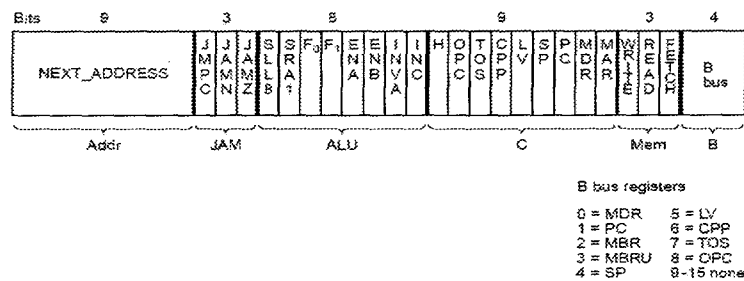
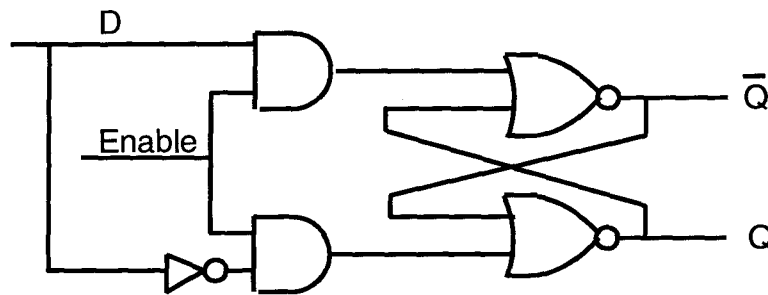
Consider the digital logic design in Figure 1.

(b) Write down the truth table for this circuit.

The ENABLE line is used to determine *when* the outputs can be modified (i.e., as long as ENABLE is not asserted, the outputs don’t change).

(c) Rather than allowing the outputs to change whenever ENABLE is asserted, we want this to happen whenever ENABLE changes value from low to high (“edge triggered”). Show how this can be done (by modifying the ENABLE input).

(d) Same as above, but now we want to enable the outputs both on the rising edge (from low to high) and on the falling edge (from high to low).



Question 3

Consider the microinstruction in Figure 2.

- (a) Explain how JNPC is used to execute the correct microinstructions for the following two IJVM code fragments:

Fragment 1	Fragment 2
ILOAD index1	WIDE
	ILOAD index2

(where index1 is one byte long and index2 is two bytes long)

- (b) What are the tasks of the instruction fetch unit (IFU)?
- (c) Why do we require instruction *issue* to be faster than instruction *execution* in a super-scalar processor?

Question 4

Consider a byte-addressable machine with 32 bits addresses, 8 bytes in a word and 8 words in a cache line. The size of the cache is 64 KByte and the cache is direct mapped. Initially, it is empty.

- (a) Given the hexadecimal address 1234 5678. In what cache line will the byte at this address be stored? Explain. Also give the tag corresponding to this entry.
- (b) Explain in detail what happens if subsequently the following (hexadecimal) addresses are referenced: (i) 1234 8765, (ii) 1234 5688, (iii) 4936 5678, (iv) 1234 5680
- (c) Will any of the cache misses in your previous answer become hits in case a 2-way set associative cache with LRU replacement policy is used (same size cache, 8 byte words, 8 words in a cache line)? If so, which one(s) and why? (Assume that again the first read is performed on 0x1234 5678.)

Question 5

- (a) When caching is not possible, stack-based architectures are inherently slower than register-based machines. Explain why.

In a multiprocessor 3 processors WA, WB, WC write variables X and Y. WA writes 1 in both variables, while WB writes 2 and WC writes 3. Three other CPUs (RA, RB en RC) read these variables a few times in a loop. This is what they see:

RA	RB	RC
X=1	Y=2	X=2
X=2	X=2	Y=1
Y=2	X=3	X=1
Y=3	Y=3	Y=1

(b) Do the memory semantics of this system conform to: (i) sequential consistency? (ii) processor consistency? (iii) weak consistency? Explain.

NEDERLANDSE VERSIE

Vraag 1

Het volgende programma wordt uitgevoerd (gegeven in pseudo-assembly):

Code	Description
1. LOAD R1,(R3)	; laad in register R1 de waarde in memory op het adres dat zich in R3 bevindt
2. L1: SUB R1,R1,#1	; trek 1 af van R1 - resultaat gaat in R1
3. ADD R2,R1,R2	; tel waarden van R1 en R2 bij elkaar op - resultaat in R2
4. BNEZ R1,L1	; als R1 niet gelijk 0, spring naar L1
5. STORE R2,(R3)	; zet waarde van R2 in geheugen op het adres dat zich in R3 bevindt

Neem aan dat R3 de waarde 1024 bevat (te interpreteren als een *decimaal* adres) en dat de waarde die zich in het geheugen bevindt op adres 1024 gelijk is aan 1000 (decimaal). R1 en R2 zijn initieel 0.

(a) (i) Wat is de waarde die in instructie 1 in R1 wordt gezet? (ii) hoe vaak wordt elke instructie (1-5) uitgevoerd? (iii) Beschrijf *in een zin* wat dit programma doet.

De code wordt uitgevoerd op een processor met een kloksnelheid van 1 GHz en met de volgende timing informatie:

Type	CPI
ALU operations	1
Branches	3
Loads	3
Stores	2

(b) Neem aan dat de processor alleen deze code uitvoert. (i) Wat is de overall CPI (clock ticks per instruction)? (ii) wat is de totale CPU tijd die nodig is om dit programma uit te voeren?

The MIPS rate van een processor is gedefinieerd als “millions of instructions executed per second”.

(c) Wat is the MIPS rate in dit geval?

(d) “De performance van twee processoren kan zinnig vergeleken worden door alleen naar de kloksnelheid te kijken, vooropgesteld dat ze dezelfde instruction set architecture hebben.” Ben je het eens of oneens met deze stelling? Leg uit.

Vraag 2

(a) Het geheugenwoord ‘101100011’ moet beschermd worden m.b.v. een Hamming code. Construeer het volledige codewoord (d.w.z., nadat de pariteitsbits zijn toegevoegd). Ga uit van even pariteit.

Zie het digitale logica design in Figure 1.

(b) Geef de waarheidstabel voor dit circuit.

De ENABLE lijn wordt gebruikt om te bepalen wanneer de outputs mogen veranderen (zolang ENABLE laag is, veranderen de outputs niet).

(c) In plaats van dat de outputs mogen veranderen zolang ENABLE hoog is, willen we dat dit alleen kan wanneer ENABLE een *transitie* maakt van laag naar hoog ("edge triggered"). Laat zien hoe dit kan door de ENABLE input te veranderen.

(d) Idem, maar nu willen we dat de output kan veranderen bij zowel een transitie van laag naar hoog ("rising edge") als van hoog naar laag ("falling edge").

Vraag 3

Zie de microinstructie in Figuur 2.

(a) Leg uit hoe JMPC wordt gebruikt om de correcte microinstructies uit te voeren voor de volgende IJVM code fragmenten:

Fragment 1	Fragment 2
ILOAD index1	WIDE
	ILOAD index2

(waarbij index1 1 byte lang is en index2 twee bytes).

(b) Wat zijn de taken van de instruction fetch unit (IFU)?

(c) Waarom moet instruction *issue* sneller zijn dan instruction *execution* in een superscalar processor?

Vraag 4

Gegeven een byte-adresseerbare machine met 32 bits adressen, 8 bytes in een woord, en 8 woorden in een cacheline. De grootte van de cache is 64 KByte en de cache is direct-mapped. Initieel is de cache leeg.

Gegeven het hexadecimale adres 1234 5678. In welke cacheline komt de byte die zich op dit adres bevindt? Leg uit. Geef ook de tag die erbij hoort.

(a) Leg in detail uit wat er gebeurt als hierna de volgende (hexadecimale) adressen worden aangesproken: (i) 1234 8765, (ii) 1234 5688, (iii) 4936 5678, (iv) 1234 5680

(b) Zullen een of meer van de cache misses uit je vorige antwoord hits worden als er een 2-way set associative cache wordt gebruikt (zelfde cache grootte, 8 byte woorden, 8 woorden in een cache line)? Zoja, welke en waarom? (Neem opnieuw aan dat de eerste read operatie plaatsvindt op adres 0x1234 5678.)

Vraag 5

(a) By afwezigheid van caches zijn stack-based processoren inherent trager dan dan register-based machines. Waarom?

In een multiprocessor schrijven drie processoren WA, WB, en WC de variabelen X en Y. WA schrijft 1 in beide variabelen, WB schrijft 2 en WC schrijft 3. Drie andere CPUs (RA, RB en RC) lezen deze variabelen een aantal keer in een loop. Dit is wat ze zien:

RA	RB	RC
X=1	Y=2	X=2
X=2	X=2	Y=1
Y=2	X=3	X=1
Y=3	Y=3	Y=1

(b) Is de semantiek van dit geheugen: (i) sequentieel consistent? (ii) processor consistent? (iii) zwak consistent? Licht je antwoord toe.