

Computer organisatie

5 februari, 2007 (18:30-21:15)

- Dit examen bestaat uit 5 vragen met subvragen. Elke subvraag is 10 punten waard.
- Begin elke vraag op een nieuwe bladzijde. Zet overal je naam en studentnummer boven.
- Gebruik van boeken of ander studiemateriaal is niet toegestaan, een rekenmachine wel.
- Houd je antwoorden bondig. Schrijf niet met potlood of rode inkt.

Vraag 1

Een processor heeft een kloksnelheid van 1 GHz en de volgende timing informatie:

Type	CPI
ALU operations	1
FP operations	10
Branches	3
Loads	3
Stores	2

Op deze processor wordt een benchmark uitgevoerd met de volgende instructie-mix:

Type	Percentage
ALU operations	50 %
FP operations	10 %
Branches	20 %
Loads	10%
Stores	10 %

(a) (i) Wat is the overall CPI? (ii) Wat is de totale CPU time die nodig is om het programma uit te voeren?

Een nieuwere uitvoering van de architectuur heeft een pipeline, bestaande uit een 7-stage pipeline. De pipeline verlengt de (ideale) clock cycle per stage met 0.2ns. Neem aan dat we te maken hebben met een ideale pipeline die gemiddeld 1 instructie per cycle uitvoert.

(b) Hoeveel sneller is de gepipeline-de uitvoering?

Vraag 2

(a) Leg uit hoe een Hamming code werkt en waarvoor deze dient.

(b) Geef voor Hammingcodes een formule die het aantal pariteitsbits geeft als functie van het aantal databits.

(c) Zie het geheugen in Figuur 1. Het bestaat uit 4 banken van 128 Mbit en rijen en kolommen worden apart op de bus gezet (m.b.v. de RAS en CAS signalen). Als 13 adreslijnen worden gebruikt voor de kolommen (CAS), hoeveel worden dan gebruikt voor de rijen (RAS)? *Leg uit!*

Vraag 3

Geef van de volgende uitspraken aan of ze goed of fout zijn (JA/NEE):

1. Nieuwe trends in geheugen (bijv. static RAM) zorgen ervoor dat de snelheids-bottleneck weer bij de CPU komt te liggen.
2. Als onze microarchitectuur, zoals behandeld in H4 over de IJVM implementatie, te veranderen is (d.w.z. het is niet vastgebakken in de hardware), dan kunnen we in een later stadium de instructie set architectuur nog wijzigen.
3. Voor een FOR-loop geeft een 2-bit branch predictor betere performance dan een 1-bit predictor, maar als die loop in een andere loop staat (bijv. een geneste FOR loop), dan loop je tegen dezelfde problemen aan als bij een 1-bit predictor.
4. Translation Lookaside Buffers (TLBs) bewaren de meest recente branch targets.

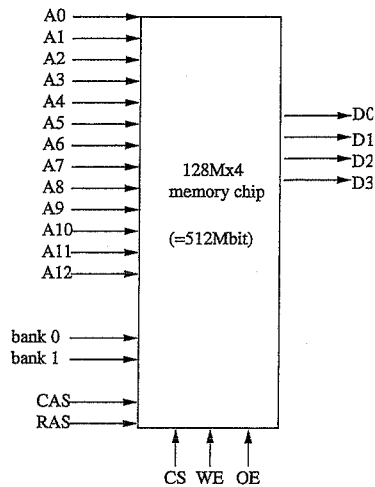


Figure 1: Memory module

5. In moderne machines, met veel geheugen is *demand paging* minder belangrijk dan vroeger, toen geheugen nog schaars was.
6. Het voornaamste doel van *Pass 1* van het assembly process is het genereren van de *symbol table*.
7. Als je op hoog niveau programmeert, bijvoorbeeld in Java, hoef je geen rekening meer te houden met hardware-specifieke zaken (zoals caches, aantallen registers, e.d.)
8. Very Long Instruction Word (VLIW) processors geven veel parallelisme, maar maken het schrijven van een goede compiler moeilijk.
9. Als alleen weak consistency wordt afgedwongen door de hardware, dan is dat strikt genomen niet altijd voldoende, omdat sommige consistentie niet op een hoger niveau kan worden afgedwongen.
10. Bij Cache-Coherent Non-Uniform Memory Access (CC-NUMA) machines maakt het voor de snelheid niet op welke nodes verschillende processen worden uitgevoerd, omdat de caches de trage toegang tot het geheugen camoufleren. Ga hierbij uit van voldoende grote caches en processen die veel communiceren via shared memory.

Vraag 4

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

(a) Gegeven het decimale adres 897. (i) In welke cache line komt de inhoud van dit adres als $X = 16$? (ii) Wat is dan de tag?

(b) Geef een waarde voor X zodanig dat het adres op cacheline 3 uitkomt.

Ga weer uit van een initieel lege cache. Stel, een typisch leespatroon van een programma is om achtereenvolgens de data te lezen op de volgende adressen:

1283, 66947, 1284, 66948, 1285, 66949, ...

(c) We hebben de keuze uit twee ontwerpen. Bij het ene ontwerp is $X = 16$ en bij het andere $X = 32$. Gegeven bovenstaand leespatroon, welke cache zou u nemen, of maakt het niet uit? Motiveer uw antwoord.

Vraag 5

Gegeven een machine die werkt met segments en pages. Een adres bestaat uit twee delen: een segment selector en een ge-paged adres binnen dat segment. De page size is 4 Kbyte. Stel dat de segment selector 16 bits lang is en de waarde 5 bevat. Het adres binnen dit segment is eveneens 16 bits lang en bevat de decimale waarde 32771 ($= 0x8003 = \text{binair: } 1000000000000011$).

(a) Hoe groot is de page table?

(b) Leg uit hoe het bijbehorende fysieke adres wordt gevonden. Neem daarbij de concrete waarden mee om uit te leggen op welke posities in welke tabellen wordt gekeken.