

Tentamen Computer Architectuur

10 januari 2005

10.00 – 13.00

- Het tentamen bestaat uit 5 opgaven waarvoor 2 punten per opgave behaald kunnen worden.
- Gebruik voor elke opgave een nieuwe pagina. Schrijf op elk vel dat U inlevert uw naam, collegekaartnummer en opgavenummer. Vermeld op het eerste vel ook uw adres en studierichting.
- Bij het tentamen mogen geen boeken, artikelen, aantekeningen e.d. gebruikt worden.
- Schrijf duidelijk en *niet* met rode pen of potlood.
- Vul s.v.p. na het tentamen de enquête in.
- *Succes!*

1. We beschouwen een load/store machine. Veronderstel de volgende karakterisatie van de operaties op deze machine

type	CPI	frequentie
ALU operaties	1	40%
Loads	3	25%
Stores	2	12%
Branches	2	23%

- (a) Wat is de CPI van deze machine?
 - (b) Leg uit wat Amdahl's Law inhoudt.
 - (c) Stel dat 30% van de instructies uit floating point instructies bestaat. Door genoeg hardware in te zetten kunnen deze flink versneld worden. Wat is de maximale speedup die bereikt kan worden door floating point instructies te versnellen?
 - (d) Leg uit wat *sub-word parallelisme* betekent. Waarvoor of voor welke applicaties is dit een interessante techniek?
2. In deze opgave worden caches besproken.

- (a) Leg uit wat *temporal* en *spatial locality* is en hoe dit door een cache uitgebuit wordt.

Veronderstel dat we een byte adresseerbare machine hebben, met 32 bits adressen, 4 bytes in een word en 4 words in een cache line. Veronderstel dat de cache 32 KByte groot is en direct mapped.

- (b) Gegeven het hexadecimale adres 1234 5678. In welke cache line komt het byte op dit adres terecht? Leg precies uit.
- (c) Bespreek **gedetailleerd** wat er gebeurt als vervolgens de hexadecimale adressen 1234 5679 en 1235 5678 aangesproken worden. Leg precies uit hoe de hardware tot conclusie 'hit' of 'miss' komt.
- (d) Leg uit hoe een *victim cache* werkt.

- (e) Neem aan dat voor een processor de cache miss penalty 100 clock cycles is. Neem verder aan dat de gemiddelde miss rate 2% is en dat er gemiddeld 1.5 memory references per instructie zijn. Hoeveel is deze processor sneller dan dezelfde processor *zonder cache*?
3. In het boek hebben we een 5 stage pipeline besproken voor de MIPS ISA. De stages zijn: (1) Instruction fetch (IF); (2) Instruction decode/register fetch (ID); (3) Execution/effective address calculation (EX); (4) Memory access/branch completion (MEM); en (5) Write back (WB).
- (a) Leg precies uit wat er in deze verschillende cycles gebeurt voor de verschillende soorten instructies die er voor de MIPS bestaan.
 - (b) Bespreek de drie soorten hazards die kunnen bestaan.
 - (c) Leg uit wat bypassing of forwarding is en waarom dat gebruikt wordt.
 - (d) Kunnen alle data hazards opgelost worden met behulp van forwarding? Zo nee, welke niet?

4. Branches

- (a) Wat verstaan we onder *delayed branching*? Wat zijn *delay slots*?
- (b) Beschrijf de drie manieren om delay slots te vullen. Wanneer mogen ze toegepast worden? Wanneer verbeteren ze de performance?
- (c) Beschouw het volgende programma fragment

```

loop:  ADD R4, R6, R8
        ST  R4, 0(R1)
        SUB R1, R1, #8
        BNEZ R1, loop
        delay slot
        ADD R6, R10, R12
        ADD R4, R6, R12

```

Wat is het resultaat van de drie manieren besproken onder **b** om het delay slot in bovenstaand fragment te vullen?

- (d) Leg uit waarom delayed branching en een pipelined architectuur zo goed samen gaan. Wat moet er in het datapad veranderen om delayed branching te implementeren?

5. Branch prediction

- (a) In een moderne superscalar pipeline kunnen 4 instructies in parallel geëxecuteerd worden. De branch penalty hier is 10 cycles. Als 1 op de 6 instructies een branch is en er gemiddeld 3 instructies in parallel uitgevoerd kunnen worden, wat is dan het percentage van de tijd dat de processor ge-stalled is als we altijd stallen op een branch?
- (b) Een branch predictor heeft een prediction accuracy van 90%. Wat is de speedup die verkregen wordt als hij in deze superscalar ingebouwd wordt?
- (c) Om speculatief instructies te kunnen uitvoeren wordt er een Re-Order Buffer (ROB) gebruikt. Leg uit hoe een ROB werkt en hoe hij gebruikt kan worden om met speculatie om te gaan.
- (d) Leg uit hoe GSHARE werkt.