

# Tentamen Computer Architectuur

14 januari 2008, 14.00 – 17.00u.

Dit tentamen bestaat uit 5 opgaven, waarmee in totaal 100 punten te behalen zijn. Iedere opgave valt uiteen in verschillende onderdelen. Het maximaal aantal te behalen punten per onderdeel staat vermeld tussen [..].

Het is niet toegestaan andere informatie te raadplegen dan eigen kennis en hetgeen eventueel tijdens het tentamen door de surveillanten verteld of toegelicht wordt.

Tenslotte: vergeet niet je naam te vermelden, beargumenteer ALLE antwoorden (losse getallen, ja/nee, en dergelijke worden NIET gehonoreerd!!!), en schrijf duidelijk. Succes!

## Opgave 1 Performance

Gegeven de performance van een memory benchmark en een processor benchmark op twee verschillende processoren

Chip	#cores	Clock frequency (MHz)	Memory performance	Dhrystone performance
Athlon 64 X2 3800+	2	3.200	2.941	17.129
Pentium 4	1	2.800	2.731	7.621

- Als je weet dat een applicatie 30% van de tijd bestaat uit memory-gerelateerde operaties, en 70% van de tijd uit processor-gerelateerde operaties, bereken dan de gewogen performance op de Pentium 4 en de Athlon 64 X2 3800+. [2]
- Hoeveel speedup verwacht je als je gaat van de Pentium 4 naar een Athlon 64 X2 3800+ voor een memory-intensieve applicatie? [2]

Veronderstel nu dat je de beschikking krijgt over een nieuwe dual processor, en dat je software moet optimaliseren voor deze processor. Je zult twee applicaties moeten draaien op deze dual processor, maar de resource requirements van iedere applicatie zijn verschillend: de eerste applicatie gebruikt 75% van de resources en de tweede slechts 25% van de resources.

- Gegeven dat 60% van de eerste applicatie paralleliseerbaar is, welke speedup zou je dan behalen als je alleen deze applicatie draait? [3]
- Gegeven dat 95% van de tweede applicatie paralleliseerbaar is, welke speedup zou je dan behalen als je alleen deze applicatie draait? [3]
- Gegeven dat 60% van de eerste applicatie paralleliseerbaar is, hoeveel *overall system* speedup zou je waarnemen als je die wel paralleliseert en de tweede applicatie niet? [3]
- Hoeveel overall system speedup zou je bereiken als je beide applicaties paralleliseert gegeven de informatie in **c.** en **d.**? [3]
- Wat is je uiteindelijke conclusie over het gebruik van deze dual processor voor deze applicaties? [3]

## Opgave 2 Pipelining en Branch Prediction

Veronderstel een 5-stage single-pipeline micro-architectuur (fetch, decode, execute, memory, write back). Gegeven de volgende code

```

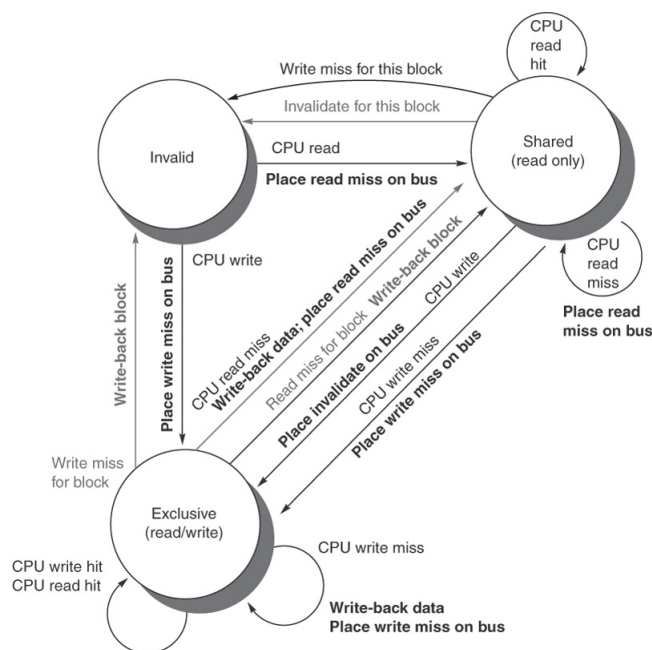
Loop: LW   R3, 0(R0)
      LW   R1, 0(R3)
      ADDI R1, R1, #1
      SUB  R4, R3, R2
      SW   R1, 0(R3)
      BNZ  R4, Loop
  
```

Alle ops zijn 1 cycle behalve LW and SW, die 1 + 2 cycles kosten, en branches, die 1 + 1 cycles kosten. Er is geen forwarding.

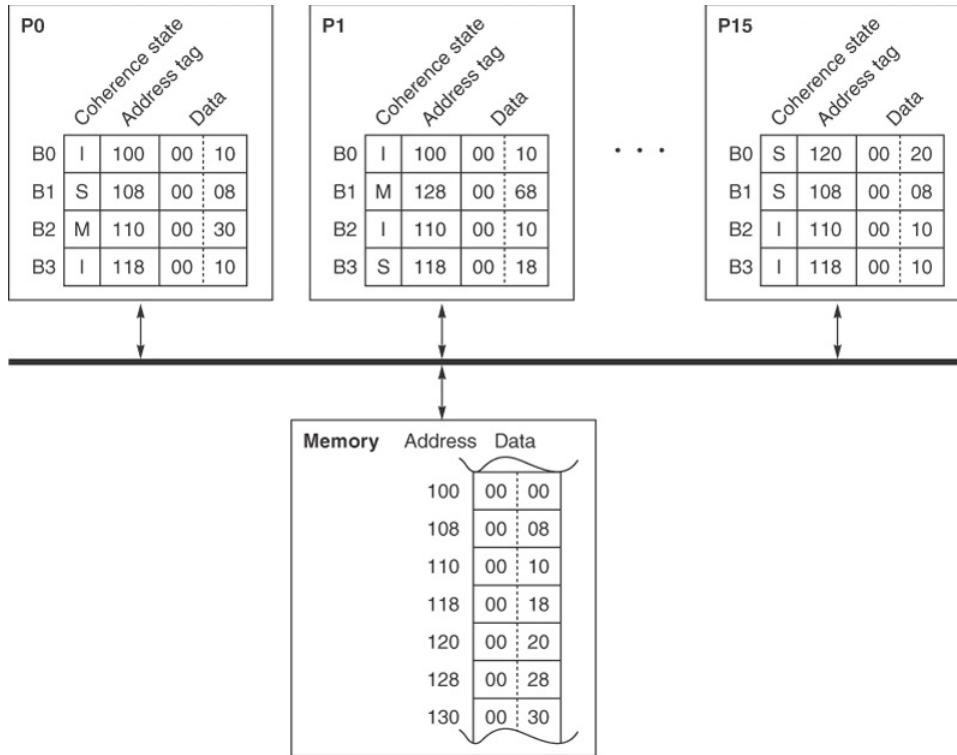
- Laat voor één iteratie van de loop zien hoe de pipeline per clock cycle gevuld wordt. [8]
- Hoeveel clock cycles per loop iteratie gaan verloren aan branch overhead? [4]
- Veronderstel een static branch predictor, die een backwards branch in de decode stage kan herkennen. Hoeveel clock cycles gaan nu verloren aan branch overhead? [4]
- Veronderstel nu een dynamic branch predictor. Hoeveel cycles gaan dan verloren bij een correcte prediction? [4]

## Opgave 3 Memory coherence

Gegeven het snooping coherence protocol



Als iedere processor een eigen private cache heeft en gebruik maakt van bovenstaand snooping coherence protocol kan de volgende eenvoudige bus-based multiprocessor, die een symmetrische shared-memory architectuur implementeert, gerealiseerd worden



Iedere cache is direct-mapped en heeft vier blocks die ieder twee words kunnen bevatten. Ter vereenvoudiging is in de figuur het cache address tag het volledige address en wordt van elk word alleen twee minst significante hex characters weergegeven. De coherence states worden weergegeven als M, S, en I voor Modified, Shared, en Invalid, resp.

We specificeren een CPU operatie in de vorm  $P\# : \langle op \rangle \langle address \rangle [ \langle \text{---} \rangle \langle value \rangle ]$ , waar  $P\#$  staat voor de CPU (bijv. P0),  $\langle op \rangle$  is de CPU operation (bijv. read of write),  $\langle address \rangle$  is het memory address, en  $\langle value \rangle$  geeft de waarde weer die gebruikt moet worden bij een write operatie.

Beschouw elke actie hieronder als onafhankelijk, en steeds toegepast op de initiële toestand zoals hierboven weergegeven. De vraag is steeds wat de resulterende toestand (coherence state, tags, en data) van de caches en het memory is na de gegeven actie. Laat alleen de gewijzigde blocks zien en wel in de volgende voorbeeldvorm: P0.B0: (I, 120, 00 01) betekent dat block B0 van CPU P0 eindtoestand I heeft, tag 120, en data words 00 en 01. Geef ook aan wat de waarde is die opgeleverd wordt bij een read operation.

- P15: read 118 [3]
- P15: write 100 <--- 48 [3]
- P15: write 118 <--- 80 [3]
- P15: write 108 <--- 80 [3]

- e. P15: read 110 [3]
- f. P15: read 128 [3]
- g. P15: write 110 <-- 40 [3]

#### Opgave 4 Virtual Machines

Virtual machines (VMs) kunnen tot verbeteringen van computer systemen leiden, zoals een betere ‘total cost of ownership’ of beschikbaarheid. Kunnen VMs ook aangewend worden om de volgende verbeteringen te realiseren? Geef steeds aan waarom wel of waarom niet.

- a. Beperken van schade veroorzaakt door computer virussen, wormen of spyware? [5]
- b. Hogere performance voor I/O-intensieve applicaties? [5]
- c. Het afschermen van fouten tussen verschillende applicaties, waardoor een betere beschikbaarheid van de services bereikt wordt? [5]
- d. Uitvoeren van software onderhoud op systemen, terwijl applicaties zonder significante onderbreking kunnen blijven draaien? [5]

#### Opgave 5 RAID systemen

Na een failure kan de reconstructie van een RAID systeem op twee manieren uitgevoerd worden: offline of online. Bij een offline reconstructie worden alle resources van het RAID systeem aangewend om de reconstructie uit te voeren, en wordt geen enkel ander verzoek gehonoreerd. Dit in tegenstelling tot een online reconstructie waarbij dergelijke verzoeken wel afgehandeld worden. Vaak heeft het reconstructie proces in dat geval slechts een deel van de totale bandbreedte van het RAID system tot zijn beschikking.

Veronderstel een RAID systeem met zes 37 GB SCSI disks, ieder met een MTTF van 1,2M uur. Elke disk kan data sequentieel lezen met een pieksnelheid van 142 MB/s en sequentieel schrijven met een piek van 85 MB/s. Neem verder aan dat de disks aangesloten zijn op een Ultra320 SCSI bus met een transferbandbreedte van 320 MB/s. Verder mag verondersteld worden dat elke disk failure onafhankelijk is en mogen andere failures van het systeem buiten beschouwing gelaten worden. In het reconstructie proces is de overhead voor een XOR berekening of een memory copy verwaarloosbaar. Tijdens een online reconstructie veronderstellen we dat het reconstructie proces beperkt is tot het gebruik van een maximale bandbreedte van 10 MB/s.

- a. Stel dat het systeem een RAID 4 systeem is. Teken een eenvoudig diagram dat de layout van de blocks over de 6 disks weergeeft voor dit RAID system. [4]
- b. Als één van de disks faalt zal het RAID 4 systeem een reconstructie uitvoeren. Wat is de verwachte tijd totdat zo’n failure zal optreden? [4]
- c. Welke read en write operaties zijn noodzakelijk om de reconstructie uit te voeren? [4]
- d. Hoelang duurt de reconstructie bij een offline reconstructie? [4]
- e. En hoelang bij een online reconstructie? [4]

# # # # #