

**2^{de} deel Tentamen Architectuur en
Computerorganisatie
2014-2015**

21 oktober 2014, 9.00 - 11.00 uur
Tentamenzaal JWS zaal 1
James Watt straat 78

Schrijf op ieder vel uw **naam** en
collegekaartnummer.

Er zijn in totaal **3 opgaven**. Iedere deelvraag
telt even zwaar in de waardering. Lees eerst
alle vragen door voor te beginnen met het
beantwoorden!

**Ieder antwoord van een duidelijk uitleg
voorzien en bijlagen inleveren!**

Succes!

Opgave 1 – H4 deel 2

- a) Beschrijf in het kort de 'structural hazard' en de 'control' hazard.
- b) Behandel de 'data hazard':
 - I. Waardoor wordt deze veroorzaakt en ondersteun het antwoord met een stukje zelf bedachte code (gebruik tevens deze code in bijlage 1a en 1b)
 - II. Behandel een oplossing met stalls en forwarding (illustreer dit in de bijlage 1a en 1b)
- c) Teken in bijlage 2 alleen de forwarding signalen die nodig zijn voor een data hazard die volgt uit de directe opvolging van twee afhankelijke instructies.

Bijvoorbeeld:

add \$1, \$1, \$2

add \$1, \$1, \$3

Teken welke informatie uit de pipeline registers moet worden gehaald naar de forwarding-unit en welke signalen uit de forwarding-unit moeten komen naar de multiplexers. Label de signalen aan de ingang van de forwarding unit.

- d) Leg het verschil uit tussen 'static branch prediction' en 'dynamic branch prediction'.
- e) Zonder een specifiek mechanisme zou één enkele register-file kunnen leiden tot een 'structural hazard'. Leg het mechanisme uit waardoor deze hazard wordt voorkomen.

Opgave 2 – H5

- a) Gegeven de onderstaande “pseudo-“code:

```
for (i = 0; i < 100; i++)  
    getallenlijst[i+1] = getallenlijst[i] * 2;
```

Leg uit welke vorm(en) van lokaliteit er wordt/worden gebruikt in deze code en hoe een data en/of instructie cache hier gebruik van maakt.

- b) Beschrijf de voor- en nadelen van een 'fully associative cache' in vergelijking met een 'direct mapped cache'. Leg ook uit hoe een 'set-associative cache' de voordelen van fully associative en direct mapped caches probeert te combineren.
- c) Er wordt een cache ontworpen voor een computer met 2^{32} bytes aan geheugen. De cache heeft 2048 cache blokken van 16-bytes elk. Geef aan, b.v. door middel van een tekening, voor zowel een 4-way set-associative cache als een direct-mapped cache hoeveel bits de verschillende adres-velden innemen die gebruikt worden om data in een cache te zoeken.
- d) Wat zijn twee essentiële verschillen tussen de register-file en de cache.
- e) Leg twee verschillen uit tussen Write-through en Write-back?

Opgave 3 – H6

- a) Welke twee grote voordelen ontstaan er bij het introduceren van zogenaamde 'vector instructies'. Zie onderstaande code.

Conventional MIPS code

```
l.d      $f0,a($sp)      ;load scalar a
addiu    r4,$s0,#512      ;upper bound of what to load
loop:    l.d      $f2,0($s0) ;load x(i)
mul.d    $f2,$f2,$f0      ;a × x(i)
l.d      $f4,0($s1)      ;load y(i)
add.d    $f4,$f4,$f2      ;a × x(i) + y(i)
s.d      $f4,0($s1)      ;store into y(i)
addiu    $s0,$s0,#8       ;increment index to x
addiu    $s1,$s1,#8       ;increment index to y
subu     $t0,r4,$s0       ;compute bound
bne      $t0,$zero,loop   ;check if done
```

Vector MIPS code

```
l.d      $f0,a($sp)      ;load scalar a
lv       $v1,0($s0)       ;load vector x
mulvs.d  $v2,$v1,$f0      ;vector-scalar multiply
lv       $v3,0($s1)       ;load vector y
addv.d   $v4,$v2,$v3      ;add y to product
sv       $v4,0($s1)       ;store the result
```

- b) Voor het bereiken van een versnelling van 80 keer de executie snelheid van 1 processor worden 100 processoren ingezet.
Welk percentage van de originele code mag nog sequentieel zijn opdat deze versnelling bereikt wordt?
- c) Behandel het verschil tussen "strong scaling" en "weak scaling".

Einde van dit tentamen