

Tentamen Computerarchitectuur en Netwerken 9 juli 2010

De nagekeken tentamens zijn in te zien op kamer Bernoulliborg 366.

Opmerkingen:

- Schrijf **netjes** en duidelijk, met zwarte of blauwe pen.
- Zet op het eerste blad alle gegevens als naam, etc., en het totaal aantal ingeleverde bladen, en nummer de ingeleverde bladen.
- Lees de opgaven eerst goed door.
- Omdat de vragen over computerarchitectuur en netwerken door verschillende docenten worden nagekeken wordt aangeraden om de antwoorden op de vragen over het onderdeel computerarchitectuur en het onderdeel netwerken op afzonderlijk antwoordpapier uit te werken. Wie dat nalaat riskeert een belangrijke vertraging bij de correctie van het ingeleverde werk.
- Beantwoord de vragen kort en bondig.
- Succes!

deel: COMPUTERARCHITECTUUR (60%)

1. Geef een eindige toestandsautomaat voor een 'even parity checker'. De automaat geeft een 1 wanneer een even aantal 1-bits aan de automaat zijn aangeboden en een 0 wanneer het aantal oneven is.
2. Een zeer oude computer gebruikt als data-eenheid 4-bits waarden. Het geheugen van die computer maakt gebruik van de Hamming fout correctie methode. Geef de representatie van de waarde 10 in een geheugencel. Laat ook zien hoe de methode kan detecteren dat 10 wordt bedoeld wanneer in de bits die worden gebruikt voor de opslag van de 16 mogelijke waarden de waarde 11 wordt aangetroffen (wanneer bijvoorbeeld ten gevolge van een fout in de geheugencel het minst significante data bit is gezet).
3. Teken het logische circuit voor een 4-kanaals multiplexer. Laat zien hoe bij ingangswaarde 3 op beide aansturinglijnen kanaal 3 (begin te tellen bij 0) aan de uitgang verschijnt.
4. Bij de aanroep van een functie spelen vaak argument en lokale variabelen een rol. Een vertaler zorgt ervoor dat de argumenten eerst op de stack worden geplaatst en dan ten opzichte van een 'frame pointer' (EBP) kunnen worden benaderd. Bij een stack die 'van hoog naar laag' groeit bevinden de argumenten zich dus op hogere adressen dan de frame pointer. Lokale variabelen worden onder de frame pointer op de stack geplaatst. We beschouwen een computer met de volgende eigenschappen:

EBP: het base (frame) pointer register
call functie: de assembler instructie om 'functie' aan te roepen.
deze instructie plaatst het return adres (RA) op de
stack en zet dan de instructie pointer op het adres van
de eerste instructie van 'functie'
de stack groeit van hoge naar lage adressen.

In de programmeertaal C worden functieargumenten in omgekeerde volgorde op de stack geplaatst (eerst het laatste argument, als laatste het eerste argument).

Teken het stackframe zoals dat wordt gebruikt bij de functie

```
void show(int a, int b)
{
    int v1;
    int v2;
}
```

Deze functie heeft twee parameters (a en b) en twee lokale variabelen (v1 en v2). In de tekening moet duidelijk zijn waar zich a, b, v1, v2, de oude frame pointer en het return adres (RA) bevinden en waar zich de hoge en lage stack-adressen bevinden. Geef ook aan op welke lokatie EBP wijst.

5. Beschrijf kort de werking van het virtueel geheugen. Beschrijf daarbij de werking van het 'paging' mechanisme. Werk het voorbeeld uit aan de hand van een denkbeeldige computer met 16 fysieke pagina's die elk 8 kByte groot zijn, terwijl gebruik wordt gemaakt van 32 bits adressen. Beantwoord de volgende vragen:

- Hoe groot is het fysieke geheugen (in bytes)
- Hoe groot is de page tabel (in bytes)
- Welk deel (welke bits) van het adres hebben betrekking op fysieke offsets binnen pagina's en welke bits worden gebruikt als pagina-index?

Hoe groot lijkt het geheugen van de computer voor de programmeur?

6. Teken een Omega-switching netwerk voor 4 computers en 4 geheugens. Laat ook zien hoe, wanneer computer 2 een verzoek doet aan geheugen 0 het nummer van de computer wordt gebruikt om de route van geheugen 0 terug te vinden naar computer 2.

deel: NETWERKEN (40%)

7. Lagenmodel

De functies van een netwerk zijn gerealiseerd via een lagenmodel (layers). Het Internet-model geeft aan dat er 5 lagen zijn. Leg uit wat de belangrijkste functies van elk van deze is en vermeldt bij elke laag wat de zender respectievelijk de ontvanger is.

8. Link-laag

De link-laag (link-layer) zorgt ervoor dat frames worden verstuurd van verzender naar ontvanger. De link-laag kent implementaties voor een point-to-point-medium respectievelijk een broadcast-medium. De aard van het onderliggende medium zorgt er voor dat de implementaties sterk verschillen. Geef aan wat deze verschillen zijn en waar deze door veroorzaakt worden.

9. Flow Control

Het lagenmodel geeft aan dat flow control zowel op de link-laag (link-layer) als transport-laag (transport-layer) aanwezig kan zijn. Praktijkvoorbeelden hiervan zijn HDLC resp. TCP.

- Leg uit waarom flow control zowel in HDLC als TCP voorkomt, en welk nadeel dit met zich mee brengt.
- Go Back N en Selective Repeat zijn mogelijk invullingen van flow control. Geef aan wanneer (en waarom) de ene invulling en wanneer de andere invulling de voorkeur verdient.

10. UDP vs TCP

UDP en TCP zijn invullingen van de transport-laag. De header van TCP is veel groter dan die van UDP omdat TCP meer functies biedt.

- Welk belangrijke functie biedt TCP extra t.o.v. UDP. Noem twee velden die hierom voorkomen bij de TCP-header en niet bij UDP-header.
- Bij welk type applicaties is UDP toch te prefereren boven TCP, en waarom juist bij deze applicaties?