

Tentamen Computerarchitectuur en Netwerken 24 juni 2011

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, studentnummer, studierichting 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, lange antwoorden kunnen een reden zijn voor een onvoldoende.
- Succes!

deel: COMPUTERARCHITECTUUR (60%)

1. Wat zijn de niveaus die door Tanenbaum in de computerarchitectuur worden onderscheiden. Geef bij elk niveau kort aan wat de functie van het niveau is.
2. Het geheugen van een oude computer bestaat uit 4-bits waarden. Het geheugen van die computer maakt gebruik van de Hamming foutcorrectiemethode. Geef de representatie van de waarde 10 in een geheugencel.
Ten gevolge van een toevallige fout in een geheugencel waarin de waarde 10 zou moeten zijn opgeslagen is de waarde 11 in die geheugencel opgeslagen. Laat zien hoe Hamming's foutcorrectiemethode deze fout kan detecteren en repareren.
3. Construeer, gebruik makend van inverters, and-gates and or-gates een schakeling voor de expressie

$$!A \&\& B \ || \ A \&\& !B$$

Hierbij is '!' de not-operator, '&&' de and-operator, '||' de or-operator. '!' heeft de hoogste prioriteit, '||' de laagste. A en B zijn ingangskanalen die elk, onafhankelijk van de ander, 1 of 0 kunnen zijn.
Geef ook de waarheidstabel van de expressie.

4. De IJVM virtuele machine gebruikt een stack om tussentijdse resultaten op te slaan, en heeft eenvoudige instructies, zoals:

```
push a2    - plaats de waarde van variabele a2 bovenop de stack
pop  a2    - ken de bovenste waarde v.d. stack toe aan variabele a2
          en verwijder dan de bovenste waarde van de stack
add        - haal (verwijder) de bovenste twee waarden van de stack
          en plaats hun som terug op de stack.
mul        - haal (verwijder) de bovenste twee waarden van de stack
          en plaats hun product terug op de stack.
```

Beschouw de expressie

$$a1 = (a2 + a1) * a3$$

Zet deze expressie om in instructies voor de IJVM. Uitgaande van een initieel lege stack, geef na elke instructie aan wat de inhoud van de elementen van de stack is (elementen hebben bijvoorbeeld waarden als: $a2$, $a2 + a1$, etc.).

5. Hogere programmeertalen kennen vaak het if-else statement. De algemene vorm van het if-else statement is:

```
if (conditie)
    statement-if-true
else
    statement-if-false
```

Beschouw de volgende conditie: $a \&\& b$

Een ISA kent een 'test' instructie (bv. 'test a') die de Zero-flag zet wanneer a gelijk is aan 0, en de Zero-flag wist indien dat niet zo is. Bovendien kent de ISA conditionele jumps:

```
jz label    - jump naar een label wanneer de Zero-flag is gezet
nz label    - jump naar een label wanneer de Zero-flag niet is gezet
mp label    - jump onconditioneel naar een label.
```

Zet bovengenoemd if-else statement om in 'jump-code' waarbij b niet wordt geevalueerd wanneer dat logisch gezien niet meer nodig is.

De code voor de twee statements kan worden beperkt tot het vermelden van hun naam op de plaats in de code waar ze moeten worden uitgevoerd. Labels worden aangegeven als namen (letters) gevolgd door een ':'. Bijvoorbeeld:

```
jnz einde    - spring naar het label einde indien de Zero-flag niet
              is gezet
...          - eventuele tussenliggende instructies en labels
einde:       - label waarnaartoe wordt gesprongen.
```

6. Bij het gebruik van parallelle computers is geheugenconsistentie vaak lastig te realiseren. Er zijn verschillende modellen voor geheugenconsistentie ontwikkeld. Geef kort aan (en geef bij het eerste model een kort voorbeeld) wat wordt verstaan onder:

- processor geheugenconsistentie,
- zwakke geheugenconsistentie,
- release geheugenconsistentie.

deel: NETWERKEN (40%)

7. Fysieke laag

- Er zijn 2 fundamentele redenen waarom de capaciteit van een communicatiekanaal niet oneindig groot is. Noem deze redenen en benoem en beschrijf de bijbehorende theorema's (stellingen).
- Gegeven een communicatiekanaal met een bandbreedte van 4 kHz. Iemand beweert dat het niet mogelijk is informatie met een snelheid van 10kbit/s over dit kanaal te verzenden. Is dit juist? Motiveer je antwoord!

8. datalink laag

De datalink laag verzorgt de communicatie tussen 2 nodes verbonden door een link (een "draad"). Het kan hierbij gaan om twee opeenvolgende nodes in een netwerkpad of nodes verbonden door een LAN.

- Er zijn 2 basistypen datalinkprotocollen. Benoem en beschrijf deze typen.
- Wanneer er sprake is van meerdere nodes die een medium delen dan is een MAC-protocol noodzakelijk. Waar staan de letters MAC voor? Wat regelt een MAC-protocol?
- Welke drie typen MAC-protocollen zijn er? Benoem en beschrijf ze, en geef van elk type een voorbeeld.

9. betrouwbaarheid datatransport

- Benoem 4 faciliteiten of onderdelen die nodig zijn om betrouwbaar datatransport te realiseren. Leg uit waarom ze nodig zijn.
Tijdens het college zijn 2 "sliding window" protocollen behandeld voor het realiseren van betrouwbaar datatransport: GBN en SR.
- Stel dat de kanaalcapaciteit beperkt is, welk protocol zou je dan kiezen? Motiveer je antwoord.
- Stel de ontvanger is een embedded applicatie waarbij erg gelet is op de kosten, welk protocol zou je dan kiezen? Motiveer je antwoord.

10. transport laag

UDP en TCP zijn invullingen van de transportlaag in het internet model. Bij de transportlaag gaat het om een logische verbinding tussen processen.

- a. UDP en TCP gebruiken poortnummers om de bron en bestemming aan te duiden. Geef tenminste twee redenen waarom het gebruik van een proces ID (PID) niet voldoende is.
- b. Hoeveel UDP/TCP poorten zijn er?
- c. Stel dat host A aan host B een TCP-segment, verpakt in een IP pakket, toezendt. Hoe weet de IP laag op host B dat het de payload van het IP pakket aan TCP in plaats van UDP (of iets anders) moet doorgeven?
- d. De header van TCP is veel groter dan die van UDP omdat TCP meer functies biedt. Welke belangrijke functie biedt TCP extra t.o.v. UDP? Noem twee velden die hierdoor voor komen bij de TCP-header en niet bij UDP-header.
- e. Gaming is een populair tijdverdrijf onder studenten. Wanneer je client een verbinding maakt met de gameserver, zou je dan kiezen voor UDP of TCP? Motiveer je antwoord.