Computersystemen en -architectuur

Computersystemen en -architectuur	
Richting	Informatica
Jaar	1BINF

Professor Vangheluwe

Theorie

Het theorie gedeelte van het examen is simpele vraag & antwoord op het niveau van de middelbare school. Er wordt niet veel inzicht in de leerstof verwacht en alles staat haast letterlijk in de cursus. Het is aangeraden om het boek te lezen, en voornamelijk te leren van de slides aangevuld met persoonlijke nota's. Elk onderdeel van de theorie wordt getoetst, dus leer zeker ook alle verschillende representaties van getallen en hoe de onderlinge conversies juist gebeuren.

Praktijk

Het praktijkexamen bestaat erin om een skelet datapath verder uit te bouwen. Er zijn drie à vier opgaven die gegeven worden, dus als je in de groep zit die het laatst het examen krijgt kan je je opgaven al op voorhand te weten komen. Het datapath dat gemaakt moet worden is een simpelere versie van die je tijdens het jaar moet maken, met een beperkte instructieset.

Puntenverdeling

Meer dan de helft van de punten staat op dagelijks werk dat in groepjes van twee moet ingediend worden. De rest van de punten wordt bepaald tijdens de examenperiode. Er is ook een mondelinge verdediging van de projecten uit het dagelijks werk, zorg dus dat je goed snapt hoe alle opdrachten die je groepje heeft ingediend werkt.

Examenvragen

Academiejaar 2021 - 2022 (1ste zittijd)

Theorie

Zie Academiejaar 2018 - 2019 (2de zittijd).

Praktijk

Hetzelfde als vorige jaren. Een aantal vragen die hij tijdens de verdediging stelde:

1. Vertaal een getal van decimaal naar binair en hexadecimaal.
2. Wat is het verschil tussen modulo-rekenen en overflow-rekenen?
3. Verschillende vragen over de werking van uw datapd (branch, PC, PC offset...)

Academiejaar 2020 - 2021 (1ste zittijd)

Theorie

Zie Academiejaar 2018 - 2019 (2de zittijd).

Praktijk

Hetzelfde als vorige jaren.

Academiejaar 2019 - 2020 (2ste zittijd)

Theorie

Zie Academiejaar 2018 - 2019 (2de zittijd).

Praktijk

Hetzelfde als vorige jaren.

Academiejaar 2019 - 2020 (1ste zittijd)

Theorie

Zie Academiejaar 2018 - 2019 (2de zittijd).

Academiejaar 2018 - 2019 (2de zittijd)

Theorie

1. Geef Amdahl's Law [1]
    1. Veronderstel een machine ...
        1. 1 instructie [0,5]
        2. Zoja, welke? Zo niet, met beide, indien toepasbaar geef de speedup[0,5]
2. Wat is het verschil tussen prefix'en (mega) in de fysica en in de informatica (druk uit in machten). [1]
3. Wat zijn de prefixen voor [1]
    1. 10181018
    2. 10−910−9
    3. 10−1210−12
    4. 10−1810−18
4. Wat is de basis 2 voorstelling van de octale notatie voor 231.12510231.12510 [0,5]
    1. Wat is de basis 2 voorstelling in hexadecimale notatie? [0,5]

5. Waarvoor staat de afkorting BCD? [0,5]
6. Wat is de BCD voorstelling van 872110872110 in binaire notatie? [1]
7. Wat zijn de voor- en nadelen van BCD? [0,5]
8. Wat zijn de 8-bit voorstellingen in binaire notatie van −2510−2510 [1]
    1. in one's complement
    2. in 2's complement
9. Voor N-bit signed 2's complement voorstelling van ZZ getallen, wat is:
    1. Het kleinst voorstelbare getal [0,5]
    2. Het grootste [0,5]
    3. De kleinste afstand tussen 2 voorstelbare getallen [0,5]
    4. De grootste afstand
10. Voor een floating point voorstelling met basis b, aantal significante digits in de fractie s, de grootst voorstelbare exponent waarde M en de kleinste exponentwaarde m, geef de formule voor:
    1. Het kleinst voorstelbare getal [0,5]
    2. Het grootste [0,5]
    3. De kleinste afstand [0,5]
    4. De grootste afstand [0,5]
11. Converteer in 3 stappen −0,254·103−0,254·103 naar een 16 bit genormaliseerde floating point voorstelling. Deze voorstelling bestaat uit een genormaliseerde basis 8 floating point formaat, met een teken bit gevolgd door 3 bit 2's complement exponent, gevolgd door 4 basis 8 digits. Het eindresultaat is een binaire string. Beschrijf elke stap. [2]
12. Converteer in 2 stappen −12.62510−12.62510 naar IEEE-754 [2]
13. Leg "denormalized" getalvoorstelling uit. Wat zijn de
    1. Voordelen? [0,5]
    2. Nadelen? [0,5]
14. Schrijf in hexadecimale voorsteling de ASCII codering van de C-stijl character string "String !\n". (ASCII tabel is gegeven) [1]
15. Wat is de kortste character string? [0,5]
    1. Hoeveel byte neemt die in C-stijl in beslag? [0,5]
16. Leg kort volgende termen uit en teken de relatie tussen de termen. Link ze allemaal aan elkaar.
    1. Character [0,25]
    2. Character name [0,25]
    3. Semantics [0,25]
    4. Code point [0,25]
    5. Encoding [0,5]
    6. Glyph [0,25]
    7. Font [0,25]
17. Leg het verschil uit tussen big-endian en little endian (illustreer)

18. Vragen ivm Matrices
    1. Wat is row-major. [1]
    2. Wat is column-major. [2]
    3. Geef de formule voor het berekenen van een element uit de matrix m(r,k) vanaf een basisoffset b. [2]
19. Geef in hexadecimale notatie de waarden ((1)) en ((2)) die in onderstaande codering van de instructies bne en j moeten ingevuld worden in onderstaande code. [2]

```
loop: sll $t1 $s3 2    | 1c50083c | 0 0 19 9 4 0
      add $t1 $t1 $s6  | 1c500840 | 0 9 22 9 0 32
      lw $t0 0($t1)    | 1c500844 | 35 9 8 0
      bne $t0 %s3 Exit | 1c500848 | 5 8 21 ((1))
      addi $s3 $s3 1   | 1c50084c | 8 19 19 1
      j loop           | 1c500850 | 2 ((2))
Exit: ...              | 1c500854 |
```

1. Hoe kan een functie met 4 argumenten f(A1,A2,A3,A4)f(A1,A2,A3,A4) die 2 resultaten R1,R2R1,R2 teruggeeft in Mips assembler gecodeerd worden? Dit zowel aan de "caller" als de "callee" zijde. Gebruik geen stack frame. Geef 2 alternatieven voor zowel de argumenten als de resultaten. [4]
2. Zet de volgende C-code om naar Assembly code, gebruik makende van de stack en de framepointer [4]

```
double f2c(double fahr) {
     double fraction, diff;
     fraction = (5.0/9.0);
     diff = (fahr-32.0);
     return fraction*diff;
}
float result;
result = f2c(15.0);
```

1. Wat is de taak van de linker (en waar komt de naam "link editor" vandaan) [1]
2. Leg aan de hand van de gegeven figuur uit hoe "dynamic linking" werkt. [1]
3. Leg aan de hand van de 2 onderstaande fig uit hoe in de Mips architectuur "exceptions" gebruikt worden voor interrupt-based memory mapped I/O. [2]
4. Geef de waarheidstabel voor een adder [2]
5. Datapad: Welk type instructie duurt (in een niet-gepipelinede architectuur) het langst? Waarom? [1]
6. Leg pipelining uit. [1]
7. Zorgt pipelining voor een verhoging in performantie? Waarom ja en nee? [0,5]
8. Leid de formule voor performantiewinst af voor een pipeline met K stages. Doe dit voor N instructies en neem dan de limiet voor N→+infN→+inf

Praktijk

Media:CSA_P_1819_2z.pdf

Academiejaar 2018 - 2019 (1ste zittijd)

Dit examen was bijna identiek aan het examen van 2015-2016. Een vraag die nog niet daar vermeld stond was "Wat is de naam van volgende getallen in SI-eenheden en omgekeerd (dus welk veelvoud er bij welke SI eenheid hoort)?".

Academiejaar 2017 - 2018 - 2e zittijd

Academiejaar 2015 - 2016 - 1e zittijd

Theorie

Hierbij een aantal voorbeelden van examenvragen, let wel op: het volledige examen was nog langer en niet alle vragen staan hieronder!

- Geef Amendahl’s Law.
- Wat is BCD
- Wat zijn de voor- en nadelen van BCD
- Geef 8023 (b10) in BCD via de hexadecimale waarde. (Parafrase hiervan)
- Geef de waarheidstabel van een adder
- Geef een bijbehorende booleaanse vergelijking
- Teken een adder met basis logische blokjes
- Teken een OR poort met alleen NOR poorten
- Een programma bevat 40% Floating Point vermenigvuldigingen, 30% FP delingen en 30% overige instructies. We willen een computer maken met die 4 maal sneller is dan de huidige. We kunnen de deling 4 maal zo snel krijgen, en de vermenigvuldigingen hoogstens 5 keer zo snel.
    - Kunnen we het gewenste resultaat bereiken door één van de twee instructies te verbeteren?
    - Zoja welke, zo neen lukt het wel met beide. Zoja geef de ‘speedup'
- Geef -25 weer in een 8-bit voorstelling van:
    - one’s complement
    - two’s complement
- Wat is het verschil tussen big-endian en little-endian. Geef het verschil weer aan de hand van een weergave van de voorstelling in het geheugen.
- Beschrijf kort hoe pipelining werkt.
- Wat is de langste instructie van de MIPS architectuur. Geef in het bijzonder aan welke delen van de architectuur de instructie gebruikt.
- Leid de formule van de performancewinst af van het aantal instructies, n, en het aantal stages ‘K’ die de pipelined architectuur heeft. Neem de limiet van N --> +∞∞
- Uitgaande van 8-bits two’s complement:
    - wat is het grootst mogelijke voorstelbare getal.
    - wat is het kleinst mogelijk voorstelbare getal.
    - wat is de kleinst mogelijke afstand tussen getallen.
    - wat is de grootst mogelijke afstand tussen getallen.
- Zet de volgende C-code om naar Assembly code, gebruik makende van de stack en de framepointer:

“Programma van Fahrenheit naar ºC mét gebruik van Double Precision FP”

- Zet -0.254 (b10) in 3 stappen om naar single precision FP.
- Gegeven een FP voorstelling met fractie s, de kleinst mogelijke exponent m en de grootst mogelijke exponent M en basis b.
  - wat is het grootst mogelijke voorstelbare getal.
  - wat is het kleinst mogelijk voorstelbare getal.
  - wat is de kleinst mogelijke afstand tussen getallen.
  - wat is de grootst mogelijke afstand tussen getallen.
- Beschrijf de MIPS exception afhandeling adhv. memory mapped I/O.
- Vragen ivm Matrices
  - Wat is row-major.
  - Wat is column-major.
  - Geef de formule voor het berekenen van een element uit de matrix m(r,k) vanaf een basisoffset b.

Praktijk

Media:CSA 15-16.pdf

Academiejaar 2013 - 2014 - 1e zittijd

Praktijk

Media:CSA 13-14.pdf

Academiejaar 2010 - 2011 - 1ste zittijd

Praktijk

Gegeven is een volledig uitgewerkte Program Counter/Register File en ALU. In het geheugen is plaats voor 1024 instructies. De RF bevat 8 registers, per conventie worden deze zo benut. (Noot: Registers worden aangeduid door notatie Ri en Rj.)

- R0 – R3 : Tijdelijke registers (`$t0-$t3`)
- R4 : Return Adress (`$ra`)
- R5 : Stack Pointer (`$sp`)
- R6 : `$v0` (analogie met mips)
- R7 : Argument Register (`$a0`)

Je datapad moet de volgende instructies kunnen uitvoeren.

- **OR** Ri, Rj # Ri = Ri | Rj
- **ADD** Ri, Rj # Ri = Ri+Rj
- **SUB** Ri, Rj # Ri = Ri - Rj
- **BLT** Ri, Rj, offset # Als Ri kleiner is dan Rj, wordt er gebrancht naar PC+1+offset

- **SW** Ri, Rj, offset # De waarde van Ri wordt gestored op adress : Rj+offset
- **LD** Ri, Rj, offset # Laad in Ri, Rj+offset
- **LDI** Ri, immediate # Laad in Ri, een getal in
- **JR** Ri, offset # Jump naar Ri+offset
- **JAL** <8bit-adress> # Jump naar <adres> EN save het huidige adres in R4

De instructies hebben volgende binaire voorstelling:

- OR : 000iiijjj000
- ADD : 000iiijjj001
- SUB : 000iiijjj010
- BLT : 001iiijjjooo
- SW : 010iiijjjooo
- LD : 011iiijjjooo
- LDI : 100iiimmmmmm
- JR : 101iiiooooo
- JAL : 110xaaaaaaaa

- ooo : 3-bit 2's complement offset.
- Iiijjj : Met iii en jjj respectievelijk de unsigned voorstelling van het registernummer (0-7)
- mmmmmm : Unsigned Immediate
- aaaaaaaa : Unsigned Adress

Dit wetende, implementeer een 12-Bit datapad dat de gegeven instructieset ondersteunt.

Vragen

- Hoe zou je de pseudo instructie NOOP (NO Operation) implementeren dmv bestaande instructies te gebruiken? (noot: meerdere oplossingen mogelijk)
- Hoe zou je de pseudoinstructie CP (Copy Ri, Rj # kopieer de waarde van Rj, naar Ri) implementeren door gebruik te maken van de bestaande instructies?
- Hoeveel extra instructies kunnen aan deze instructieset worden toegevoegd door de huidige binaire vorm te behouden?
- Program Counter
    - Is de program counter, Signed of Unsigned?
    - Hoeveel bits moet de PC groot zijn?
    - Hoe groot moet het Instructie geheugen zijn (in bytes)?

Divers

Schrijf een programma dat de volgende code in C voorstelt door gebruik te maken van de gegeven instructies. (noot: Je kan geen gebruik maken van labels, adressen dienen door hun numerieke waarden opgeroepen worden.)

```
{value = 2
push(value)
value = value+i
pop(value)
}
```

Schrijf een eenvoudig programma dat de volledige functionaliteit van je Datapad toont.

Professor D'Haene

Bespreking

Theorie

Op zich is dit vak vrij eenvoudig: de cursus is niet erg uitgebreid, de leerstof is op zich niet zo moeilijk, noch is het examen dit. Het is echter een feit dat er niet met punten wordt gegooid en dat dit vak vaak wordt onderschat zodat niet iedereen slaagt.

Groso modo probeert dit vak een inleiding te geven in de werking van computersystemen. De theorie van dit vak is onderverdeeld in de gewone theorie waarvan de cursus bestaat uit slides en dan de theorie over besturingsystemen.

De gewone theorie wordt gegeven door professor D' Haene zijn dus dan zullen vele zaken van deze bespreking niet meer gelden) en omschrijft de inwendige bouw van computers, geschiedenis van de computers, inleiding electronica, inleiding Floating-Point getallen, telsystemen. Op zich is dit alles interessant, ware het niet dat de prof dezelfde slides als in de cursus gebruikt voor zijn lessen en deze nogal letterlijk afleest voor de studenten (in het vak Computerarchitectuur verbetert dit wel sterk aangezien deze stof interessanter is voor hem). Elke week zag je dan ook het aantal studenten die de lessen bijwoonden dalen. Kortom, het is dus niet ECHT nodig om naar de les te komen hier (al is het wel handig).

Het theoretisch examen telt vrij veel vragen zodat, als je een bepaalde vraag niet goed weet, je dit nog kan ophalen. Denk eraan voldoende uitleg te geven overal en zie dat je de flip-flop goed kent aangezien dit altijd gevraagd wordt en op veel punten gaat. Het mondeling examen is bij een goed schriftelijk examen een rustig gesprek waarbij je met een beetje geluk naar een kennisdemonstratie van D'Haene kan luisteren en hij eigenlijk meer spreekt dan jij.

Over de theorie van de besturingssystemen weet ik zelf niet echt hoe dit in zijn werk ging dit jaar.

Praktijk

De praktijk is dan weer wel interessant om bij te wonen, de sfeer is er luchtig en de assistent (Kurt Vanmechelen) is tof om mee te werken. In het begin kan de materie van de praktijk wat vaag overkomen (machinetaal, omzetten talsystemen), maar door samen

te werken wordt dit al snel een routinezaak.

Het praktijkexamen van dit gedeelte bevat geen echte verrassingen en zijn varianten op behandelde zaken tijdens de oefeningensessies. Zie wel dat je het gedeelte over machinetaal goed begrijpt.

De praktijk over de Operating Systems bestaat erin in Solaris wat BASH-scripts te schrijven onder leiding van David Dewolfs om kennis te maken met UNIX. Dit gedeelte gaat maar op 3 punten van het totaalpakket, op zich niet zo veel, maar omdat deze introductie voor velen een eerste kennismaking is met het UNIX-besturingssysteem kan je beter komen.

Het examengedeelte van dit laatste bestaat er dan ook uit enkele ietwat complexere BASH-scripts te schrijven.

Puntenverdeling

Computersystemen: 13/20 (waarvan 7/20 op theorie en 6/20 op praktijk), besturingssystemen op 7/20 (waarvan 3/20 op praktijk en 4/20 op theorie). Dit kan echter afgelopen jaar zijn gewijzigd.

Examenvragen

Academiejaar 2007 - 2008 - 1ste zittijd

Theorie

1. Verklaar de volgende afkortingen:
    - ISP
    - ISA
    - PMT
    - IP
    - RGB
    - FCFS
    - IEEE
    - EBCDIC
2. Leg uit: CPU scheduling algoritmes.
3. Teken en leg uit : Stroomdeler.
4. Leg uit de manieren van VideoCompressie.
5. Leg uit: Non-Preemtive en Preemtive scheduling.
6. Leg uit, Fetch , decode, execute Cycle.
7. Invertor maken met transistoren.
8. Nyquist en Shanon Criteria.
9. DSL (uitgebreid + mondeling toelichten).
10. Disk - Scheduling algoritmes (uitgebreid + mondeling toelichten).
11. SR-latch met NOR Poorten (uitgebreid + mondeling toelichten).

Praktijk

1. Do the following conversions:
    1. (1210212) in basis 3 naar basis 9.
    2. (110110) in basis 2 (6-bit 1's complement) = (?) in basis 2 excess 32.
2. Calculate the sum
    1. (1A)basis 12 + (13.3)basis 4 = (?)basis 2
3. We consider a floating point representation with a sign bit, followed by an excess-15 exponent of 5 bits, followed by a normalized 8-bit fraction with a hidden bit. The comma is located behind the first non-zero digit. Numbers that cannot be represented exactly are rounded to the nearest number that can. 0 is represented by the reserved bit pattern 000000000000000. The exponents 00000 en 111111 are reserved for special cases and cannot be used for normal number representations.
    1. What is the number of representable numbers?
    2. (-0.123)basis 4 to this floating-point representation.
4. Construct a truth table for a function that calculates the sum of 3 - 1 bit input signals A1 ; A2 ; A3 . The output is the sum of the bits that consists S1,S2 (vb A1 = 1 en A2 = 0 en A3 = 1 =>> S1 = 1 S2 = 0). Implement also with 2 8-to-1 multiplexers *(Grappige noot: oorspronkelijk zou de oefening met een 4-to-1 multiplexer zijn geweest maar een kleine typo was in het examen geslopen)*.
5. Write Pep/7 assembly code that reads a number between 1 and 26 as input, and prints the respective character to screen. For Ex if the input is 5 , the output should be E and if the input is 26 , the output should be Z.
6. What is the content of the accumulator (A) after execution of the following code?

> LOADA d#0,i LOADX Z,d
>
> X : STOREX X,d Y: ADDA X,d SUB d#1,i BRGT Y STOP Z : .WORD d#5 .END

Academiejaar 2000 - 2001 - 1ste zittijd

Theorie

1. Noem de 5 elementen van een computer volgens het Von Neumann model.
2. Hoeveel megahertz zijn er in 1 gigahertz?
3. Is het mogelijk om elk decimaal nummer exact voor te stellen als een binair nummer?

4. Schrijf het binair getal (101101011101)2(101101011101)2 in het 8-deling getallenstelsel (octaal)=(.....)8(octaal)=(.....)8

Praktijk (compilatie van meerdere jaren)

1. Kan je de binaire getallen gemakkelijk omzetten naar hexadecimale getallen door het binair op te delen in groep jes van 3 binairen?
2. Is het grootste natuurlijke getal in een n-bit voorstelling 2n?12n?1?
3. Is 01111111 de one's complement voorstelling van -128 in een 8-bit voorstelling?
4. Zijn de getallen (1010100011001)2(1010100011001)2 en (12431)8(12431)8 gelijk aan elkaar?
5. Is 218 het tussenresultaat als je -37 omzet naar one's complement in een 8-bit voorstelling?
6. Is het kleinste getal bij one's complement 2n−1−12n−1−1 in een n-bit voorstelling?
7. Als je weet dat een Floating Point voorstelling B=4, e=7, m=37 en het decimaal punt voor bit nummer 5 heeft, is de nauwkeurigheid dan 2?172?17 ?
8. Stel dat je beschikt over een achttallig geheugen, waarvan de woordlengte 44 bits lang is. De exponent is 7 bits lang en staat in binary offset. Geef al de specificaties, tekening, absolute fout, normalizatie, nauwkeurigeid en het bereik als je weet dat het decimaal punt achteraan staat.
9. Hoe wordt -0.05 voorgesteld in het bovenvermelde geheugen?
10. Je hebt een Floating Point voorstelling met B = 2, e = 3, m = 3 en het decimaal punt vooraan. Liggen dan alle getallen tussen 2323 en 2?52?5 ?
11. Stelt 01110110 in binary offset met een 8-bit voorstelling het getal -10 voor?

Praktijk UNIX

1. Maak een script dat een lijst strings als parameters inleest en een aantal bewerkingen op deze lijst toepast naargelang de opties die aan het script worden meegegeven. De volgende lijst geeft een overzichtvan de opties en de bewerkingen die het script moet kunnen verwerken:

    - **-l** druk de lijst op het scherm
    - **-i** druk de lijst achterstevoren op het scherm
    - **-a** druk de eerste en laatste string van de lijst op het scherm (indien maar 1 string wordt meegegeven aan het script, druk je deze twee keer af)
    - **-m** de string in het midden van de lijst wordt afgedrukt (indien er een even aantal strings wordt meegegeven, druk je maar 1 string af)

    Er is niet op voorhand bekend hoeveel strings er worden meegegeven. Het moet mogelijk zijn meerdere opties mee te geven. Een voorbeeld van het gebruik: myscript -l -mstr1str2str3. Dit moet het volgende als uitvoer geven:
    str1str2str3
    str2
    (Hint : maak gebruik van arrays om de parameters in op te slaan)

2. Maak een script dat een rij getallen als parameter inleest en hiervan een aantal grootheden berekent naargelang de optie die het script meekrijgt. De volgende lijst geeft een overzicht van de opties en de grootheden die berekend moeten worden.

    - **-s** de som van de getallen
    - **-p** het product van de getallen
    - **-g** het grootste getal
    - **-k** het kleinste getal
    - **-m** het gemiddelde van de getallen

    Er is niet op voorhand bekend hoeveel getallen er worden meegegeven. Het moet mogelijk zijn meerdere opties mee te geven. Een voorbeeld van het gebruik:

    > #> myscript -k -p 12 4 20 31 1 kleinste: 1 product: 29760

3. Maak een script dat een bestand inleest dat twee rijen getallen bevat. Het script moet deze twee rijen getallen van plaats verwisselen zonder een extra bestand te gebruiken. De naam van het bestand wordt als parameter aan het script meegegeven. Het bestand dat moet worden ingelezen ziet er als volgt uit:

   > 12 23 23 34 34 45 45 56 56 67 67 78 78 89 89 90

   Maak zelf een dergelijk bestand aan om je script te testen. Het script verwisselt de kolommen en drukt ze op het scherm op de volgende manier:

   > 23 12 34 23 45 34 56 45 67 56 78 67 89 78 90 89

   (Hint : maak gebruik van arrays en de for-lus om de rijen van plaats te verwisselen.)