

# Examen computerarchitectuur

Dinsdag 14 juni 2011, 8u30

Prof. Koen De Bosschere

Richting:

Naam:

## Belangrijk

1. Vergeet niet uw naam te vermelden.
2. Schrijf de antwoorden in de daarvoor voorziene ruimte. Bereid uw antwoord voor in het klad, en schrijf het naderhand over. De antwoorden zijn meestal kort.
3. Het examen duurt 3 uur.
4. Gelieve geen rode inkt te gebruiken.
5. Het examen is open boek.
6. U mag geen computer, gsm of rekenmachine gebruiken bij de oplossing van de vragen.
7. Gelieve uw mobieltje uit te schakelen.
8. Onregelmatigheden worden aan de examencommissie gemeld.

Veel succes!

Ik verklaar op erewoord dat ik noch hulp geboden heb aan, noch hulp ontvangen heb van derden tijdens het oplossen van dit examen.

Handtekening:

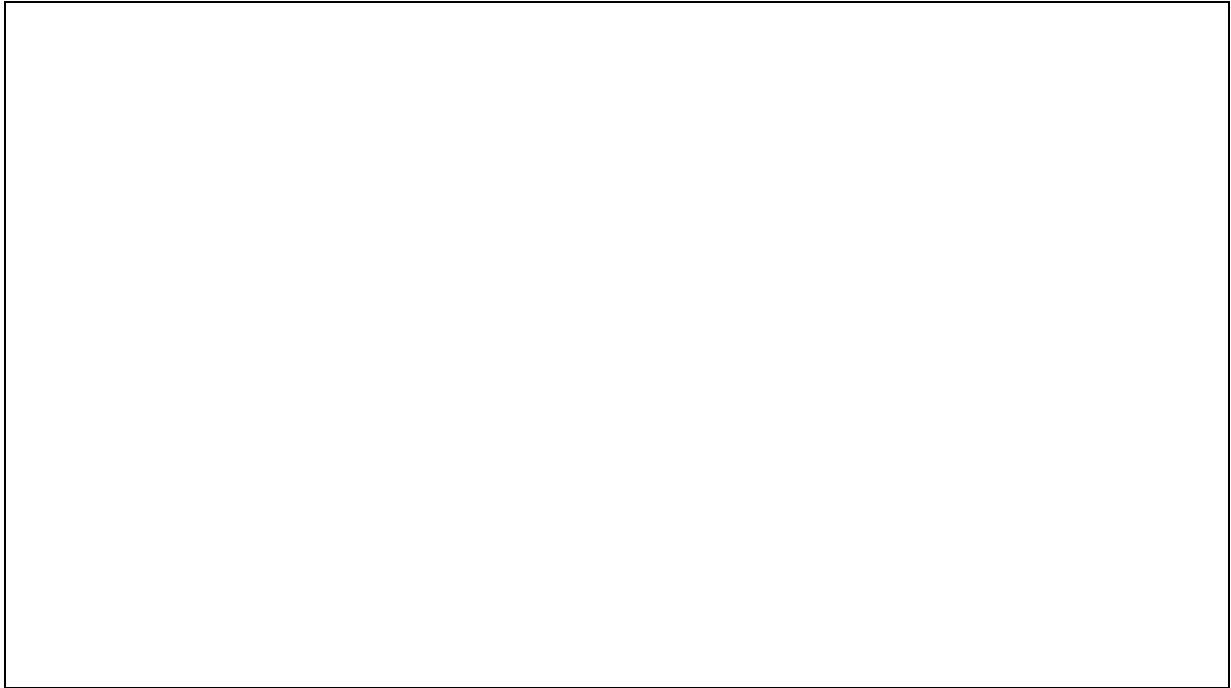
Schrijf hier eventuele opmerkingen die van belang kunnen zijn bij de quotering (ziekte, topsport, gemaakte afspraken, enz.).

--	--	--	--	--	--	--

### Vraag 1 (3 punten)

De reflectieve gray-code uit de cursus heeft als nadeel dat ze bestaat uit een aantal symbolen dat een macht van twee is. Er bestaat met andere woorden geen reflectieve gray-code die de getallen 0-9 codeert.

Kan je een alternatieve (4-bit) gray-code bedenken die wel 10 decimalen kan coderen, en na 9 terug op 0 valt? Geef deze code hieronder weer.



Kan je aantonen/weerleggen dat dit mogelijk is voor alle mogelijk waarden kleiner dan  $2^n$ , met n het aantal bits?



## Vraag 2 (3 punten)

Gegeven de volgende C-functie:

```
unsigned int g(unsigned int x)
{
    unsigned int c = 0;
    while (x > 0) {
        x = x & (x-1);
        c++;
    }
    return c;
}
```

Met als assemblercode:

```
g:
    xorl    %edx, %edx
    testl  %edi, %edi
    je     .L4
.L5:
    leal  -1(%edi), %eax
    addl  $1, %edx
    andl  %eax, %edi
    jne  .L5
.L4:
    movl  %edx, %eax
    ret
```

Bereken de waarde voor de functieoproep `g(0x000001c)`.

Welke functie wordt hier berekend?

## Vraag 3 (4 punten)

Gegeven de C-functie

```
#include <stdio.h>

void bubbleSort(long numbers[], long array_size)
{
    long i, j, temp;

    for (i = array_size-1; i > 0; i--)
    {
        for (j = 1; j <= i; j++)
        {
            if (numbers[j-1] > numbers[j])
            {
                temp = numbers[j-1];
                numbers[j-1] = numbers[j];
                numbers[j] = temp;
            }
        }
    }
}

long a[10] = {5,6,2,4,0,9,8,7,1,3};

int main()
{
    long k;

    bubbleSort(a, (long)10);
    for (k=0; k<10; k++) printf("%d ", a[k]);
    puts("");
}
```

met de assemblercode voor een x86-64 machine:

```
bubbleSort:
    subq $1, %rsi
    testq %rsi, %rsi
    jle .L13
.L11:
    movq $1, %rax
.L4:
    movq -8(%rdi,%rax,8), %rcx
    movq (%rdi,%rax,8), %rdx
    cmpq %rdx, %rcx
    jle .L5
    movq %rdx, -8(%rdi,%rax,8)
    movq %rcx, (%rdi,%rax,8)
.L5:
    addq $1, %rax
    cmpq %rsi, %rax
    jle .L4
    subq $1, %rsi
    jne .L11
    ret
.L13:
    ret
```

```

main:
    movq $10, %rsi
    movq $a, %rdi
    xorq %rbx, %rbx
    call bubbleSort
.L15:
    movq a(,%rbx,8), %rsi
    movq $.LC0, %rdi
    addq $1, %rbx
    call printf
    cmpq $10, %rbx
    jne .L15
    movq $.LC1, %rdi
    call puts
    xorq rax, rax
    ret

a:
    .quad 5
    .quad 6
    .quad 2
    .quad 4
    .quad 0
    .quad 9
    .quad 8
    .quad 7
    .quad 1
    .quad 3

.LC0:
    .string "%d "
.LC1:
    .string ""

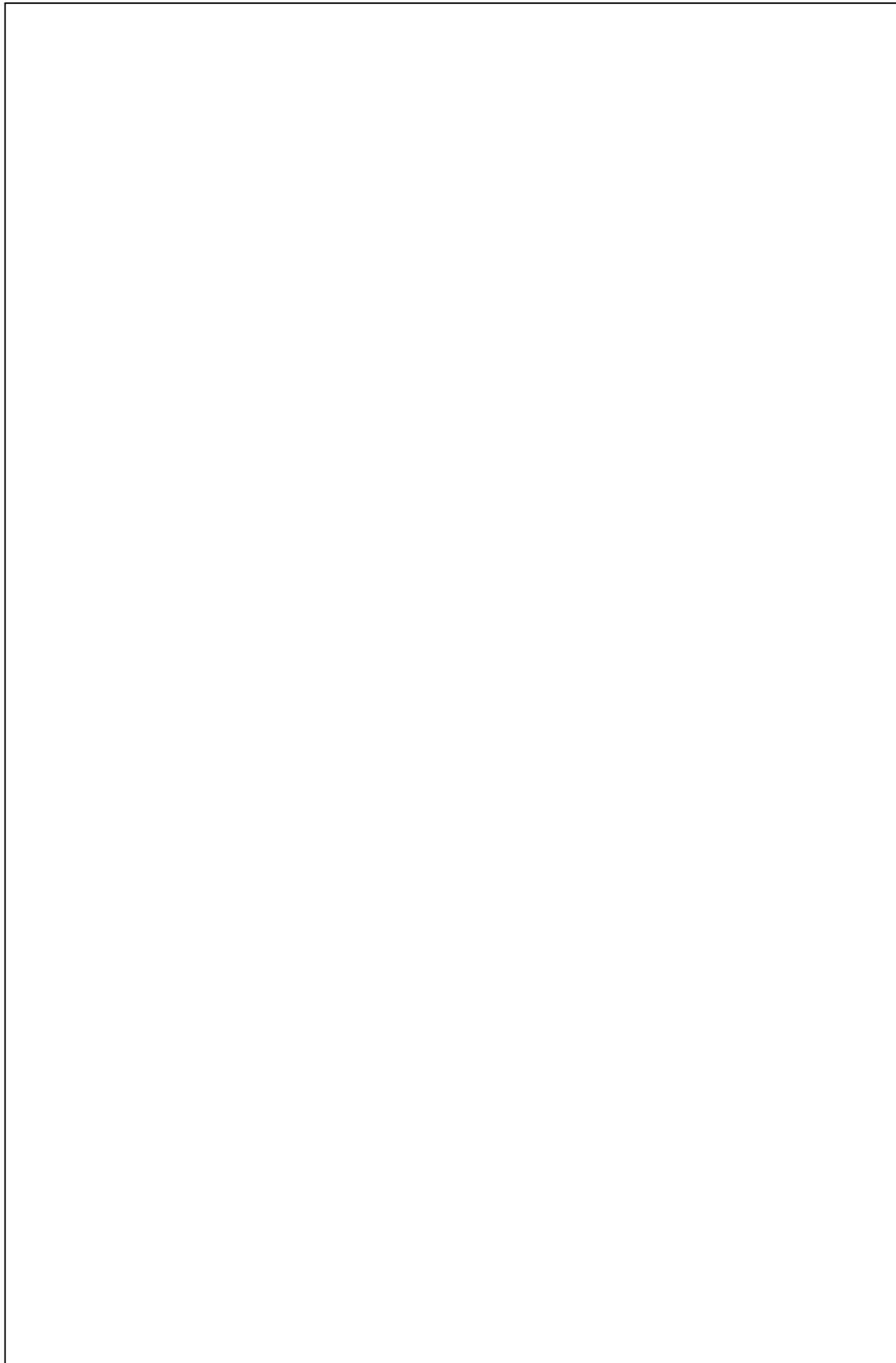
```

In welke registers/geheugenplaatsen liggen de volgende veranderlijken het grootste deel van de tijd opgeslagen?

veranderlijke	register/geheugenplaats
i	
j	
k	

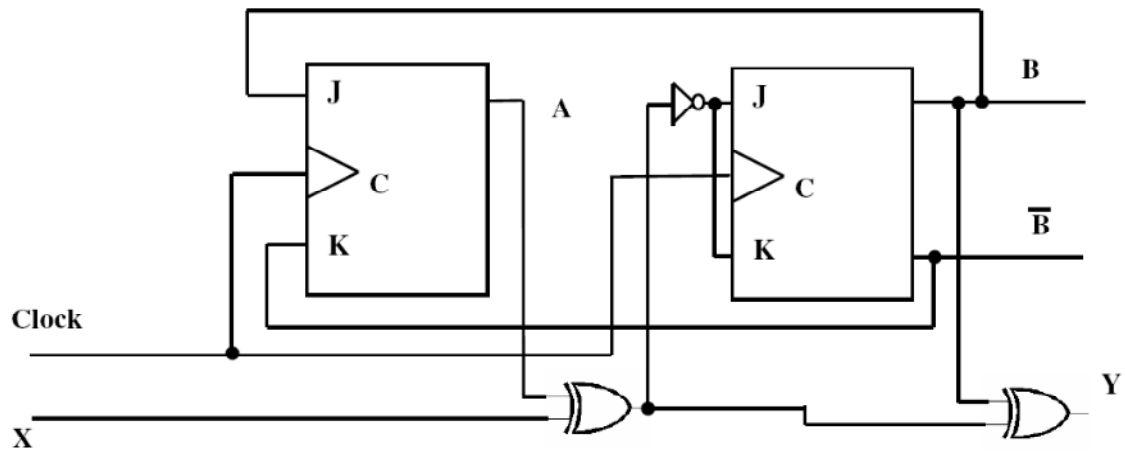
Welke oproepconventies worden hier gebruikt?

Teken de controleverloopgraaf van deze code (je mag printf en puts als 1 basisblok voorstellen).



### Vraag 4 (4 punten)

Gegeven de schakeling



Stel de toestandsvergelijkingen op voor het bovenstaande circuit (tip: de vergelijking voor een JK-flipflop is  $Q_{t+1} = J \bar{Q}_t + K Q_t$ ).

$$A_{t+1} = f(A_t, B_t, X)$$

$$B_{t+1} = g(A_t, B_t, X)$$

$$Y = h(A_t, B_t, X)$$

Stel aan de hand van de toestandsvergelijkingen het toestandstransitiediagramma op:

$A_t$	$B_t$	$X$	$A_{t+1}$	$B_{t+1}$	$Y$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Teken de toestandsovergangen in het toestandstransitiediagramma ( $xy=A_tB_t$ ):

00

10

01

11



## Vraag 5 (4 punten)

Gegeven het volgende C programma.

```
#include <stdio.h>

int i;
int a[10] = {0,1,2,3,4,5,6,7,8,9};
int b[10] = {0,1,2,3,4,5,6,7,8,9};
int c[10];

int main()
{
    for (i=0; i<10; i++){
        c[i] = a[i] * b[i];
    }
}
```

Met de bijhorende assemblercode:

```
main:
    leal    4(%esp), %ecx
    andl   $-16, %esp
    pushl  -4(%ecx)
    xorl   %edx, %edx
    pushl  %ebp
    movl   %esp, %ebp
    pushl  %ecx
    movl   $0, i
.L2:
    movl   b(,%edx,4), %eax
    imull  a(,%edx,4), %eax
    movl   %eax, c(,%edx,4)
    addl   $1, %edx
    cmpl   $10, %edx
    jne    .L2
    movl   $10, i
    popl   %ecx
    popl   %ebp
    leal   -4(%ecx), %esp
    ret
```

En met de veronderstelling dat de statisch gealloceerde data zich allemaal aaneensluitend in het geheugen bevinden, en dat de veranderlijke *i* gealigneerd is op een veelvoud van 16 bytes. Een int is 32 bit.

Je mag uitgaan van een volledig associatieve cache van 8 KiB die bij de start van de uitvoering volledig leeg is. De cacheblokken zijn 16 byte lang. Als vervangingstrategie wordt LRU gebruikt.



## Vraag 6 (2 punten)

Wat is de bandbreedte die vereist is voor het versturen van hoge definitie video?  
Welk aspect van de HiPEAC roadmap wordt hiermee geïllustreerd?

Wat zijn de uitdagingen die ARM te wachten staan om door te breken in de desktop en servermarkt?

Voor hoeveel jaren voorspelde Gordon Moore een constante toename van het aantal transistors per chip?

Wat is de gemiddelde belasting van een server en op welke manier kan deze verhoogd worden (en tot welke waarde)?