Student: **Collegekaartnummer:**

Tentamen Computersystemen

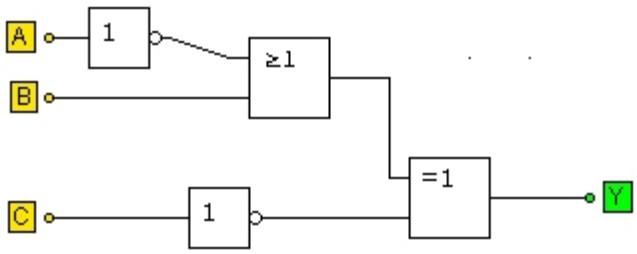
baiCOSY06 2e jaar bachelor AI, 2e semester 23 september 2013 13u-15u

IWO 4.04A (blauw), Academisch Medisch Centrum, Meidreef 29, Amsterdam ZuidOost

Het is niet toegestaan communicatieapparatuur zoals tablets en telefoons te gebruiken: zet de mobiele telefoon uit!

Gebruik van een rekenmachine en de boeken behorende bij dit vak (Computer Systems, en Van 0 en 1 tot pipeline processor) is toegestaan. Succes!

Vraag 1



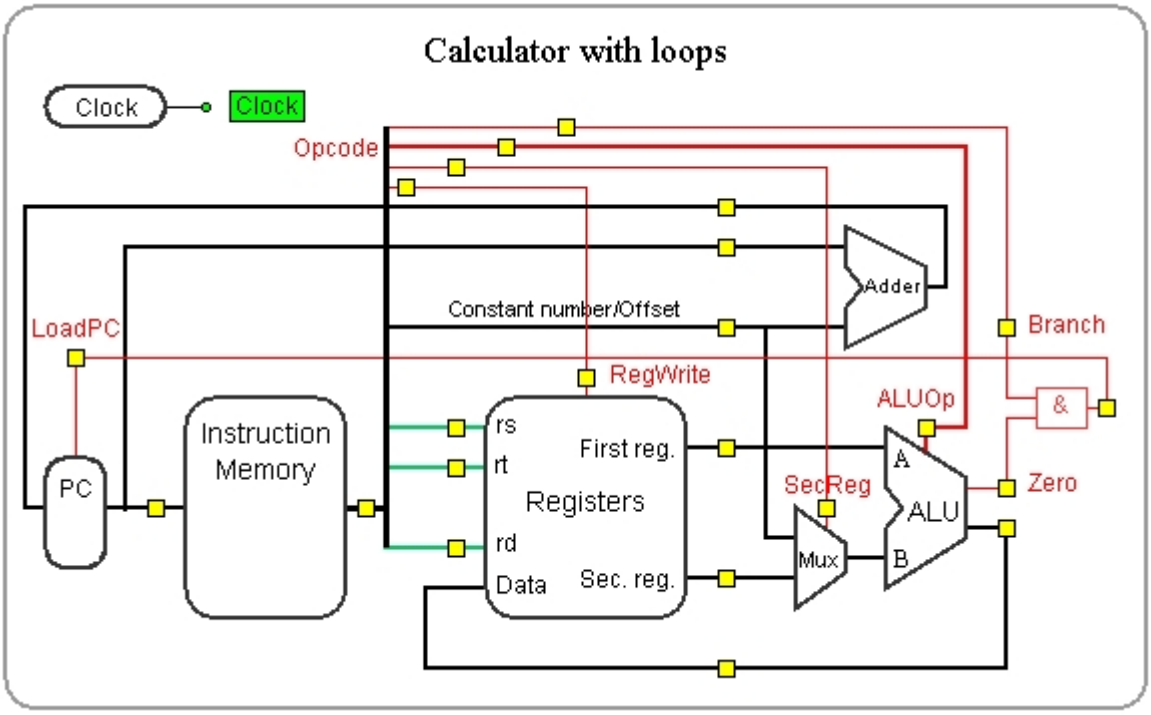
a. Geef de formule hoe Y afhangt van de ingangen A,B,C.
b. Geef de waarheidstabel van bovenstaande schakeling.
c. De afzonderlijke poorten hebben een t_{pd}=10 ns. Wat is van de schakeling de maximale t_{pd}?

Vraag 2

a. Wat is het verschil tussen een half-adder en een full-adder?
b. Geef de 8-bits binaire representatie van de (decimale) getallen 107 en 83.
c. Deze twee getallen worden bij elkaar opgeteld. Welke bits (nummering bit#0..7) krijgen er een carry-bit bij?
d. Wat is de 8-bits two's complement code van het getal -7 ?
e. Wat is de hexadecimale code van het (decimale) getal 1000 ?

Student: **Collegekaartnummer:**

Vraag 3



a. Wat is de functie van de component Adder?

b. Welke functionaliteit is toegevoegd bij de uitbreiding van Rekenmachine-II naar dit model; welke instructies zijn erbij gekomen?

c. Waarom heeft het blok 'Registers' er de lijn RegWrite bij gekregen?

d. Beschouw het programma-fragment:

```
LOADI $6,3
Loop:
ADDI $0,$0,8
SUBI $6,$6,1
BZ $6,End
BRA Loop
End:
HALT
```

Wat is de offset bij BZ instructie; wat is de offset bij de BRA instuctie ? (16-bits two's complement)

Hoe lang duurt de uitvoering van bovenstaand programmaatje? De processor is Single-Cycle en de klokfrequentie is 100 MHz.

Student: **Collegekaartnummer:**

vraag 4

De geleerde Archimedes probeerde te bewijzen dat er veel zandkorrels in het heelal zijn, maar dat dit aantal de zandkorrels in het heelal aftelbaar is en niet oneindig. Dat is een aardige uitdaging, zeker als men bedenkt het grootste getal met een Griekse naam μυριάς was; 10,000 oftwel 10^4. Archimedes had in eerder (helaas verloren werk[1]) een numeriek systeem met basis 10^8 gecreerd. Deze basis μυριάς μυριάς was 10.000 maal 10.000. Getallen tot 10^8 noemde Archimedes de 'eerste' serie getallen. Tussen 10^8 en 10^{16} was de 'tweede' serie getallen. Hij kon hiermee doortellen tot de μυριάς μυριάς serie getallen (tussen $(10^8)^{9999999}$ en $(10^8)^{100000000}$). De getallen tot $10^{800000000}$ noemde hij getallen van de eerste periode. Doortellend kwam hij tot de μυριάς μυριάς perioden van getallen, waarmee zijn grootste getal een 1 met

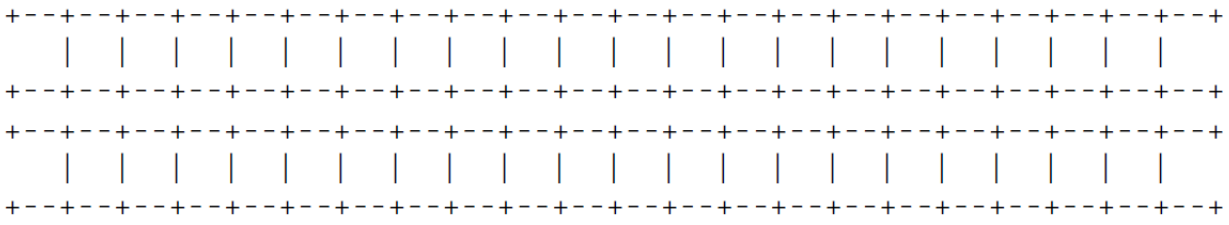
80 biljard nullen (

Student: **Collegekaartnummer:**

Het concept van series van getallen van Archimedes introduceert wel de exponent van de wetenschappelijke notatie, zonder dat het gebruik maakt van fracties of een drijvende komma. Archimedes getallen blijven dus gehele getallen, maar dan wel hele grote (10^{63} heet bijvoorbeeld in het Nederlands deciljard). De bedoeling is nu dat je een ontwerp maakt voor Archimedes numeriek systeem een data type `series` equivalent met IEEE Standard 754, maar zonder *sign bit s*, een *significand M* die niet fractional is, maar loopt van 1 tot 10^8, en een *exponent E* die loopt van 1 tot 10^8 zonder *sign bit s*.

b. Waarom heb je geen *denormalized values* nodig voor het data type `series`?

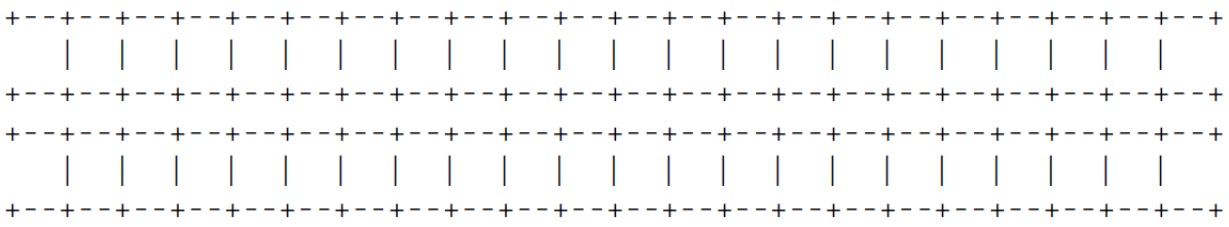
c. Hoeveel bits heb je nodig voor het data type `series`? Geef in onderstaande template de locatie aan van de *significant M* en de *exponent E*.



d. Wat is de range van het data type `series`?

e. Om dezelfde range als het data type `series` te halen, heb je dan voldoende aan een `float`, een `double`, een `long double` (80 bits) of het `_float128` (128 bits) formaat van `gcc`? Beargumenteer je antwoord.

f. Representeer in jouw ontwerp voor `series` de binaire representatie van Archimedes' schatting voor het aantal zandkorrels in het heelal (10^{63} korrels ~ 10.000.000 getallen van de 'achtste' serie). Je kunt hiervoor de volgende template gebruiken:



Student: **Collegekaartnummer:**

vraag 5

Neem aan dat je de beschikking hebt tot een machine waar getallen van het type `int` 32-bits zijn. Deze gehele getallen zijn opgeslagen m.b.v. het *two's complement* formaat, en de *right shift* is aritmetisch geïmplementeerd. Reële getallen van het type `float` zijn opgeslagen in het 32-bits IEEE *floating point format*, getallen van het type `double` zijn opgeslagen in het 64-bits IEEE *floating point format*.

Er worden drie random getallen `x, y` en `z` gecreëerd en op de volgende manier geconverteerd naar andere formaten:

```
/* Create some arbitrary values */
int x = random();
int y = random();
int z = random();
/* Convert to other forms */
unsigned ux = (unsigned) x;
unsigned uy = (unsigned) y;
double dx = (double) x;
double dy = (double) y;
double dz = (double) z;
```

Controleer voor de volgende C-expressies of deze ***altijd*** waar zijn:

Expression	Always True?
`(x<y) == (-x>-y)`	Y N
`((x+y)<<4) + y-x == 17*y+15*x`	Y N
`~x+~y+1 == ~(x+y)`	Y N
`ux-uy == -(y-x)`	Y N
`(x >= 0) || (x < ux)`	Y N
`((x >> 1) << 1) <= x`	Y N
`(double)(float) x == (double) x`	Y N
`dx + dy == (double) (y+x)`	Y N
`dx + dy + dz == dz + dy + dx`	Y N
`dx * dy * dz == dz * dy * dx`	Y N

Geef voor de C-expressies die niet ***altijd*** waar zijn een tegenvoorbeeld.

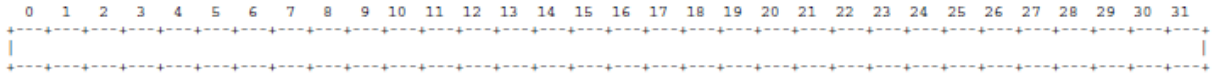
Student: **Collegekaartnummer:**

vraag 6

Beschouw de volgende datatype definities voor IA32 *(x86)* Linux machine.

```
typedef struct {        typedef union {
   char c;                 char c;
   double *p;              double *p;
   int i;                  int i;
   double d;               double d;
   short s;                short s;
} struct1;              } union1;
```

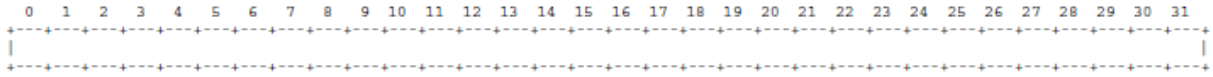
a. Teken aan de hand van de onderstaande *template* (voor maximaal 32 bytes), de allocatie van dat voor een *structure* van type `struct1`. Geef met behulp van een streep aan tot welke byte elk van de elementen (er zijn er 5) loopt en geef een *label* aan waar het element gevonden kan worden. Maak de bytes die niet gebruikt worden grijs (om de *alignment* te waarborgen).



b. Hoeveel bytes worden er gereserveerd voor een object van type `struct1`?

c. Welke *alignment* is gebruikt voor een object van type `struct1`?
Hint: als een object *aligned* is met stappen van *x*-byte, is het antwoord *x*.

d. Als we de vrijheid hebben om de elementen van `struct1` in een andere volgorde te definiëren, kunnen we het aantal ongebruikte bytes verminder. Wat is het aantal ongebruikte bytes in het optimale geval? Teken je oplossing in onderstaande *template*.



e. Hoeveel bytes worden er gereserveerd voor een object van type `union1`?

f. Welke *alignment* is gebruikt voor een object van type `union1`?

Succes!