

Opgavenserie 6: Assembly

Systeemarchitectuur 1

25 maart 2011

De uitwerkingen van deze opgaven graag inleveren uiterlijk op 29 maart 2011 om 10.45 uur, per e-mail aan Wouter Geraedts (w.geraedts@student.ru.nl). Ik corrigeer alleen uitwerkingen in **platte tekst** of **PDF** die in een e-mail met onderwerp „[PnP] Opgave 6” verstuurd worden. Ik word extra vrolijk van uitwerkingen die met L^AT_EX zijn gemaakt.

Als je drie van de vijf opgaven goed hebt beantwoord, telt jouw uitwerking mee voor de bonus bij het tentamen.

Maak zonodig zinvolle aannames. Beredeneer je antwoord; laat tenminste zien dat je het antwoord niet hebt gegoogelt.

1. **(2 punten)** Tijdens het werkcollege heb ik een programma laten zien om te vermenigvuldigen. Deze code definiëren we nu als de macro-instructie MUL. (Zie werkcollege-sheets) Schrijf een programmafragment voor de practicum-processor dat $R_0^{R_1}$ (machtsverheffing) berekent en in R_0 opslaat.

2. **(2 punten)** Geef het principe aan hoe je een switch-statement naar assembly vertaalt. Vertaal volgens dat principe het volgende switch-statement:

```
switch( i ) {  
  case 'A': j = 1;  
  case 'B': k = 2;  
             break;  
  case 'C': j = 2;  
             break;  
  default: k = 3;  
             break;  
}
```

De variabelen i , j en k staan op adressen 3, 5 en 8.

3. We hebben een willekeurig programma, waarbij 2% van de code tijdens executie 50% van de totale executietijd nodig heeft. Vergelijk de volgende drie strategieën en kijk hoeveel tijd er nodig is voor het programmeren en hoeveel tijd er nodig is voor het uitvoeren van het programma. Neem aan dat voor dit programma 100 man-maanden nodig zijn om het in C te schrijven. Assemblycode is 10 keer langzamer om te schrijven en vier keer meer efficiënt.

- (a) Het gehele programma in C.
- (b) Het gehele programma in assembler.
- (c) Alles is eerst in C geschreven, waarbij later het belangrijke stukje van 2% herschreven is in assembler.