# Obligatorisk øving 4 i datateknikk

Dette er den fjerde obligatoriske øvingen i datateknikk.

## Oppgave 1 Parallellitet på instruksjonsnivå

1. Tidligere var det «god skole» at kompilatorer samlet instruksjoner som er avhengige av hverandre i kontinuerlige blokker. På moderne prosessorer er ikke dette nødvendigvis særlig lurt lengre.

* Hvorfor var dette lurt på eldre prosessorer?

Svar:

Det var lurt på eldre prosessorer siden de kjørte instruksjoner sekvensielt og hadde ikke like mange registere som nyere CPUer, dermed ble det viktig å gruppere instruksjoner som er avhengig av hverandre for å unngå at CPUen går tom for registere.

* Hvorfor er det ikke lurt på nyere prosessorer?

Svar:

I moderne CPUer kjøres flere instruksjoner samtidig og derfor kan det potensielt skje at en instruksjon trenger data fra en tidligere instruksjon som ikke er ferdig enda, derfor er det lurere å la CPUen arbeide med instruksjoner som ikke trenger å vente på data mens andre instruksjoner gjør seg ferdig.

1. Gode kompilatorer tar hensyn til egenskaper med prosessoren, og genererer kode som er optimalisert for den prosessoren. Anta at følgende instruksjoner skal utføres:

a = 1  
b = 1  
a = a + b  
c = a + 2  
p = 10  
q = 10  
p = p + q  
r = q + 10

Her er a, b, c, p, q, r variabler, og vi antar at utrykkene er så enkle at hver linje utgjør én instruksjon.

* Hvorfor kan dette være en uheldig rekkefølge på en superskalar prosessor?

Svar:

Dette er en uheldig rekkefølge for en superskalar prosessor siden instrukser som er avhengig av svar fra tidligere instruksjoner må vente på at instruksjon skal skrive svaret i et register, som dermed vil utsette alle de videre instruksene.

* Foreslå en instruksjonsrekkefølge som ikke endrer sluttresultatene av program­sekvensen, men som er optimalisert for en superskalar prosessor av grad 2. Vær nøye med å begrunne det du gjør.

Svar:

----- (a = 1)

----- (b = 1)

----- (p = 10)

----- (q = 10)

-- --- (a = a+b)

-- --- (p = p+q)

- ---- (r = q+10)

- - --- (c = a+2)

Totalt 11 pulser i stedet for 15 pulser som den forrige brukte

## Oppgave 2 Minneaksess på moderne prosessorer

*Denne oppgaven krever at du husker stoffet om Systemarkitektur i tillegg til stoffet om moderne prosessor­arkitektur. Om nødvendig må du lese de aktuelle leksjonene på nytt.*

Anta en prosess med flere superskalare kjerner. En slik prosessor kan fint utføre én instruksjon pr klokkesyklus over relativt lang tid hvis man unngår minneaksess.

Anta at prosessoren har en (intern) klokkefrekvens på 3 GHz.

Anta at minnet er av typen DDR3 1600 med følgende timing 8-8-8-24.

Vi skal nå sammenligne prosessorytelse og minneytelse.

1. *Prosessoren:*

* Hvor lang tid tar hver instruksjon når prosessoren utfører en instruksjon pr klokkesyklus?

Svar:

1 / 3 GHz= 0.33 ns

1. *Minne:*

* Hvor lang tid (i ns) går det mellom hver klokkesyklus på minnebussen?

Svar:

1 / 800 MHz = 1.25 ns

* Hvor lang tid (i ns) går det mellom hver overføring (hvert ord) i en *burst* med denne minnetypen?

Svar:

1 / 1600 MHz = 0.625 ns

* Hvor lang tid (i ns) tar det å hente ut nye data når rett rekke allerede er aktivert?   
  (Enkelte artikler kaller denne tiden for aksesstiden til minnet. Det stemmer ikke med det vi kalles aksesstiden. Den tiden som beregnes i dette tilfellet krever jo at rett rekke allerede er aktivert)

Svar:

8 \* 1/800 MHz = 10 ns

* Hvor lang tid (i ns) tar en minne­aksess i verste fall?   
  (I verste fall betyr at minnet må begynne med å aksessere rett bank, deretter rett rad og så videre. Det er dette som er aksesstiden til minnet, altså tiden det tar å aksessere en vilkårlig plass i minnet. )

Svar:

(8+8+8) \* 1/800 MHz = 30 ns

* Sammenlign aksestiden du fant i forrige spørsmål med aksesstiden som kompendiet opererer med for DRAM. Er det overensstemmelse?

Svar:

Minnet i oppgavene over er en del kjappere enn det brukt i kompendiet

1. *Sammenligninger:*

* Hvor mange instruksjoner utføres mellom hver enkelt overføring (hvert ord) i en *burst*?

Svar:

0.625 / 0.33 ≈ 2 instruksjoner

* Hvor mange instruksjoner kan prosessoren utføre i løpet av den tiden en minne­aksess i verste fall tar (i verste fall betyr at minnet må begynne med å aksessere rett bank, deretter rett rad og så videre).

Svar:

30 / 0.33 ≈ 91