

kdlObligatorisk øving 4 i datateknikk

Dette er den fjerde obligatoriske øvingen i datateknikk.

Oppgave 1 Parallellitet på instruksjonsnivå

- a) 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 lenger.

- Hvorfor var dette lurt på eldre prosessorer?

Svar:

Mellomregningen kunne lagres i register istedenfor å måtte flyttes til minnet og tilbake igjen. Dette sparer tid og plass.

- Hvorfor er det ikke lurt på nyere prosessorer?

Svar:

Nyere prosessorer er superskalare som betyr at instruksjoner ofte er avhengige av hverandre. Dette kan føre til en data-hasard.

- b) 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 uttrykkene er så enkle at hver linje utgjør én instruksjon.

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

Svar:

Fordi det kan gi en data-hasard ved at f.eks. c på linje 4 ikke har verdien til a på linje 3 fordi den kun er en instruksjon bak. Dette er fordi de to instruksjonene utføres nesten/helt parallelt med hverandre.

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

Svar:

En superscalar prosessor av grad 2 vil utføre instruksjoner parallellt parvis, som betyr at en ordning der instruksjonene parvis ikke er avhengig av hverandre vil optimalisere eksekveringen.

$a = 1$

$b = 1$

$p = 10$

$q = 10$

$a = a + b$

$p = p + q$

$c = a + 2$

$r = q + 10$

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.

a) Prosessoren:

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

Svar:

3.33ns

b) Minne:

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

Svar:

1.25ns

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

Svar:

0.625ns

- 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:

$$8+8+24 = 40 \text{ klokkesykluser}$$

$$40 * 1,25 \text{ ns}$$

$$= 50 \text{ ns}$$

- 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:

$$8+8+8+24 = 48 \text{ klokkepulsar}$$

$$48 * 1,25 \text{ ns}$$

$$= 60 \text{ ns}$$

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

Svar:

Ja, 60ns er innenfor 35-70ns.

c) Sammenligninger:

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

Svar:

Ingen, fordi overføringene skjer automatisk fra 1.adresse. Minnet trenger kun å vite hvilken adresse den skal starte på.

- 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:

$$60 \text{ ns} / 3,33 \text{ ns}$$

$$= 18 \text{ instruksjoner}$$