# Obligatorisk øving 3 i datateknikk

Dette er den tredje obligatoriske øvingen i datateknikk.

## Oppgave 1 Cache

1. I kurset har vi sett at den effektive aksesstiden ved bruk av cache er:

Te = Tc + (1-H)T

hvor  
 Te er effektiv aksesstid   
 H er treffraten  
 Tc er cachens aksesstid  
 T er aksesstid til lageret vi leser fra (f. eks primærminne)

Svar på spørsmålene nedenfor. På noen av spørsmålene kan du gjerne ta utgangs­punkt i formelen (men alle spørsmålene kan besvares med bruk av sunn fornuft hvis du ikke liker matematikk! Og husk: ikke gjør spørsmålene vanskeligere enn de er):

1. Hva skjer med effektiv aksesstid dersom vi over lang tid bare har bom (altså at treffraten er null i lang tid)?

Svar:

med treffrate på null over lang tid går effektiv aksesstid til maksverdi da cachen ikke inneholder det vi trenger så vi alltid må i primærminnet og hente data.

1. Hva skjer med effektiv aksesstid dersom vi over lang tid bare har treff (altså at treffraten er lik 1 i lang tid)?

Svar:

Med bare treff over lang tid får vi laveste mulig aksesstid siden alt vi trenger alltid er i cachen, så vår får dataen vi trenger veldig raskt hele tiden.

1. Hvorfor er det lite sannsynlig at treffraten er null i lang tid?

Svar:

Siden data som hører til de samme tingene som regel er lagret samlet på primærminnet, dermed blir det nesten alltid med riktig informasjon i blokkene som blir tatt inn i cachen.

1. Treffraten ER null med det samme vi starter et nytt program. Hvorfor?

Svar:

Fordi det ikke har blitt hentet inn riktige blokker fra primærminnet enda siden data fra forrige program fortsatt ligger der.

1. Hva skjer med treffraten når et nystartet program har fått kjøre en stund?

Svar:

Treffraten går jevnt oppover ettersom mer blokker blir hentet inn i cachen og går mot sin normale 95-98% treffrate.

1. Hva er det som bestemmer hvor høy treffraten må være for at cachen skal ha positiv effekt? Samme spørsmål på en annen måte: «Hvis treffraten er for lav vil vinningen gå opp i spinningen. Hva er det som bestemmer hvor høy treff­raten må være for at dette ikke skal skje?»

Svar:

Treffraten må være høy nok til at du fortsatt sparer tid på å hente informasjon og legge den i cachen så lete gjennom den, fremfor å bare hente fra primærminnet til enhver tid. Så jo lavere aksesstid primærminnet har, jo høyere treffrate må du ha for å få en positiv effekt av cachen.

## Oppgave 2 Systemarkitektur. Synkrone teknologier

1. Gi en beskrivelse av hva hvert av begrepene CL, tRCD, tRP og tRAS betyr.

Svar:

1. Et minne som er bygget opp av ordinært DDR SDRAM kan brukes på flere ulike klokkefrekvenser. I spesifikasjonene til et minne får du oppgitt følgende tabell:

|  |  |  |  |
| --- | --- | --- | --- |
| Frekvens: | 266 MHz | 333 MHz | 400 MHz |
| CL: | 4 | 5 | 6 |

Vi ser at dess høyere frekvens vi bruker, dess flere pulser må vi vente.

Regn ut ventetiden i nanosekund til minnet, for hver av de tre bussfrekvensene med tilhørende CL.

(NB: Hvis dette er vanskelig bør du ta en titt på den frivillige øvingen i Systemarkitektur)

Svar:

1. Som forrige deloppgave viste, så kan en og samme minnebrikke brukes på ulike frekvenser. Men forsinkelsen (latency) blir større dess høyere frekvens bussen bruker.

Til tross for at forsinkelsen (latency) øker ved høye frekvenser, så ønsker vi å bruke høyest mulig frekvens. Når er det vi har bruk for den høye frekvensen?

Svar:

## Oppgave 3 Din egen PC

I denne oppgaven skal vi finne en del nøkkelinformasjon om egen datamaskin. Vi bruker programvare for å finne denne informasjon. For Windows finnes mange velegnede program. Selv foretrekker jeg programmet CPU-Z fra http://cpuid.com.

Hva skal du gjøre hvis du bruker MAC eller Linux? Denne oppgaven tar utgangspunkt i den informasjonen du finner med CPU-Z. Jeg tror det beste er om du finner en Windows-maskin der du kan kjøre dette programmet. Eventuelt kan du gjøre øvingen i samarbeid med en i klassen som bruker Windows.

Det finnes sikkert alternativer for Mac og Linux, så du kan jo google «zpu z alternative mac» eller «zpu z alternative linux». For MAC har jeg ikke funnet noe fullgodt alternativ, men programmet CPU X har LITT av den samme funksjonaliteten. På Linux finnes programmene CPU G og I NEX. De som ønsker det kan jo prøve disse eller andre alternativer.

**Oppgave**Last ned og installer programmet CPU-Z fra https://cpuid.com. Start programmet og bruk resultatene fra programmet til å svare på spørsmålene nedenfor.

1. Se under fanen CPU, og svar på følgende:
   1. Hva er navnet (Name) på prosessoren?

Svar:

* 1. Hva er prosessorens klokkehastighet (Core Speed) oppgitt til?

Svar:

* 1. Forklar hva som menes med klokkehastigheten til en prosessor.

Svar:

1. Se under fanen Caches, og svar på følgende:
2. Hvor stor (Size) er L1-cachen (antagelig står det L1 D-cache)?

Svar:

1. Det står også hvor mange-veis sett-asossiativ cachen er, og hva linje-størrelsen er. Hva står det her på din PC?

Svar:

1. Hvor stor (Size) er L2-cachen? Hvor mange-veis sett-asossiativ er den? Hva er linjestørrelsen?

Svar:

1. Se under fanen Mainboard, og svar på følgende:
2. Hva står angitt som grafisk grensesnitt (Graphic Interface)?

Svar:

1. Hva vet du om båndbredden til dette grensesnittet? (Søk om nødvendig i læremateriell eller på Internett).

Svar:

1. Se under fanen Memory, og svar på følgende:
2. Hva slags type RAM bruker du, og hvor stort er minnet (lagringskapasitet)?

Svar:

1. Hvor mange kanaler (channels #) har du?

Svar:

1. Hva er frekvensen til bussen (DRAM Frequency)?

Svar:

1. Angi henholdsvis: CL-tRCD-tRP-tRAS

Svar:

1. Se under fanen SPD, og svar på følgende:
2. Sjekk hver Slot og se om det står data for alle. Hva betyr det hvis det finnes slots der det ikke står noe data?

Svar:

1. Angi for hver slot: Størrelse (Module Size) og Maks båndbredde

Svar:

1. Gransk Timing Table og svar på spørsmålene:
   * Hva skjer med antall pulser forsinkelse (Latency) etter hvert som frekvensen øker?
   * Hvorfor betyr dette at minnet likevel ikke blir tregere når vi øker frekvensen?

Svar:

Forviss deg om at du forstår alle begrepene ovenfor.