# Obligatorisk øving 2 i datateknikk

Dette er den andre obligatoriske øvingen i datateknikk.

## Oppgave 1 Grunnleggende virkemåte

Den hypotetiske maskinen i leksjonen Grunnleggende Virkemåte brukte 16-bits instruksjoner der 4 bits var opkode og 12 bits var operand (enten en adresse til minnet eller en IO-adresse)

1. Hvor mange opkoder kan vi skille mellom på denne maskinen? Grunngi svaret grundig

Svar:

Man kan skille mellom 16 opkoder på denne datamaskinen. Det er fordi i binærtallsystemet har vær plass en verdi lik 2 opphøyd i sitt plassnummer, fra og med 0. med 4 bits gir dette 2^0 = 1, 2^1=2, 2^2=4 og 2^3=8. Summen av dette blir 15, men man må ikke glemme kombinasjonen 0000, så da blir det 16 kombinasjoner.

1. Hvor mange adresser kan vi skille mellom på denne maskinen? Grunngi svaret grundig

Svar:

Man kan skille mellom 4096 adresser på denne datamaskinen. Det er av samme grunnen som i oppgaven over, men denne gangen har vi 12 bits og ikke fire. Man må derfor summere 2 opphøyd i 0 til 11, pluss 1. En enklere måte å gjøre dette på er å ta 2^12.

1. Skriv bitmønsteret nedenfor på heksadesimal form (Vis fremgangsmåten tydelig):  
   0101 0000 1100 1111.

Svar:

0101 = 5 (desimal) = 5 (heksadesimal)

0000 = 0 (desimal) = 0 (heksadesimal)

1100 = 12 (desimal) = C (heksadesimal)

1111 = 15 (desimal) = F (heksadesimal)

Svaret blir: 50CF

1. Hvis bitmønsteret i oppgave c) er en instruksjon til den hypotetiske maskinen, hvilken instruksjon er det da? (Hvilken opkode og hvilken adresse)? Grunngi svaret grundig

Svar:

Opkoden er 0101. Dette betyr ADD. Den skal legge et tall fra minnet til tallet i AC-registeret, og lagre resultatet i AC-registeret.

0000 1100 1111 er adressa til tallet i minnet som skal legges til i AC.

1. Hvis bitmønsteret i oppgave c) er data til den hypotetiske maskinen (altså heltall med fortegn som står beskrevet i leksjonen), hvilken verdi representerer bitmønsteret? (Svaret skal oppgis desimalt, og svaret skal grunngis grundig)

Svar:

I data-bitmønstre er forteller den første biten om det er et positivt eller negativt tall, mens de 15 andre er verdien til heltallet. I dette tilfellet er første biten 0, altså er det et positivt tall. Resten av bitmønsteret er 101 0000 1100 1111. Det er 20687 på desimal.

## Oppgave 2 CPU, minne og buss

I leksjonen ble parallelle synkrone busser beskrevet. På moderne PCer er det en slik buss som brukes som minnebuss. Synkron buss betyr at en ny handling bare kan starte når det kommer en klokkepuls. I leksjonen så vi at det ved lesing fra minnet brukes en klokkepuls for å overføre adressen, så ventet vi en klokkepuls på at minnet skulle finne frem til rett lokasjon. Til slutt brukes en klokkepuls til å overføre data fra minnet til CPU. Se kap 1.3.4 i leksjonen om busser.

Men hva skjer hvis minnet er så tregt at det ikke rekker å hente data fra rett lokasjon i løpet av en klokkesyklus. Jo, da må vi vente flere klokkesykluser – og alltid et helt antall klokkesykluser. I denne oppgaven skal vi regne på hvor mange pulser vi må vente.

1. Anta at vi har en (synkron og parallell) buss med frekvens 800 MHz (800 millioner klokkepulser hvert sekund). Hvor lang tid går det mellom hver puls på en slik buss?

Svar:

1 / 800 millioner = 1,25 \* 10^(-9) = hvert 1,25. nanosekund.

1. Anta at DRAM-minnet som denne bussen er tilkoblet har en aksesstid på 35 ns. Hvor mange klokkepulser må vi vente?

Svar:

35 / 1,25 = 28. Man må vente 28 klokkepulser.

1. Anta at prosessoren som bruker bussen greier å utføre en milliard instruksjoner pr sekund. Hvor lang tid tar da hver instruksjon?

Svar:

1 / 1000000000 = 1\* 10^(-9). Hver instruksjon tar ett nanosekund.

*Kommentar: Hvis du synes det er noe som ikke rimer så kan du ha regnet helt rett likevel. Moderne prosessorer er mye kjappere enn DRAM-minnet, og denne oppgaven er ment som en motivasjon for å jobbe med neste modul i emnet; den som omhandler cache og systemarkitektur.*

## Oppgave 3 Disker, IO og avbrudd

I leksjonen om Sekundærlager ble harddisken Quantum Fireball 3.8 GB omtalt. Den brukte ZBR, og på den ytterste sonen ble hvert spor delt opp i 232 sektorer. Det lagres 512 Byte på hver sektor, og rotasjonshastigheten er 7200 omdreininger pr minutt.

1. Hvor mange Bytes lagres langs hvert av de ytterste sporene?

Svar:

512 B = 0,5KB.

0,5 \* 232 = 116

Det lagres 116KB på de ytterste sporene.

1. Hvor mange bytes pr sekund overføres ved fortløpende lesing langs et spor ytterst på platen?

Svar:

116\*7200 = 835200

835200 / 60 = 13920

13920KB = 13,92MB

Det overføres 13,92MB per sekund ved fortløpende lesing.

1. Hvor lang tid tar det mellom hver gang det ankommer en ny byte fra disken?

Svar:

1 / (13,92 \* 10^6) = 7,18 \* 10^(-8).

Det tar 72 nanosekunder for en ny byte å ankomme på disken.

1. Anta prosessoren fra oppgave 3. Hvor mange instruksjoner utfører denne prosessoren mellom hver gang det kommer en ny byte fra disken ved fortløpende lesing?

Svar:

Prosessoren utfører 1 instruksjon hvert nanosekund.

Prosessoren utfører 72 instruksjoner hvor hver gang det kommer en ny byte fra disken.

## Oppgave 4 Aksesstider

Data lagres på mange ulike lager-teknologier i en datamaskin. Lagerteknologiene spenner over en enorm variasjonsbredde av aksesstider: fra CPU-register (som har umiddelbar aksess) og til bånd (der aksesstiden måles i sekunder). Det er viktig å ha en ide om omtrentlig aksesstid for de viktigste teknologiene.

Hva er omtrentlig aksesstid for følgende teknologier (*Kommentar: Med omtrentlig aksesstid menes størrelsesorden, altså om aksesstiden måles i ns, ms osv.)*:

1. SRAM

Svar:

1 ns

1. DRAM

Svar:

35-70ns

1. SSD

Svar:

0,1 ms

1. Harddisk

Svar:

4 ms

1. CD/DVD

Svar:

100 ms