# 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:

Opkoden til instruksjonen forteller hva CPU-en skal gjøre med instruksen. I den hypotetiske maskinen hadde vi 3 forskjellige opkoder; STORE, LOAD, ADD.

STORE: Forteller CPUen at den skal skrive til minnet fra AC. Her vil operanden (adressefeltet) oppgi til hvilken minnelokasjon det skal skrives til.

LOAD: Forteller CPUen at den skal lese data fra minnet til AC. Her vil operranden (adressefeltet) angi hvilken minnelokasjon som skal leses.

ADD: Forteller CPUen at den skal addere et tall til minnet med AC. Resultatet legges i AC. Her vil operanden (adressefeltet) angi i hvilken minnelokasjon tallet finnes.

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

Svar:

Adresser forteller hvor dataen vi skal bruke ligger. Siden vi har 12-bits adresser vil dette si at vi har 212 lokasjoner i minnet, altså 4096 lokasjoner i minnet.

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

Svar:

(0101)2 = (5)16

(0000)2 = (0)16

(1100)2 = (C)16

(1111)2 = (F)16

(0101 0000 1100 1111)2 = (50CF)16

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:

De første 4-bitsene (0101)/5 forteller oss at dette er et tall vi skal addere i minnet med AC. Adressefeltet angir i hvilken minnelokasjon tallet som skal adderes finnes, altså tallet 0000 1100 1111 eller 0CF.

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:

Fra c) ser vi at (0101 0000 1100 1111)2 = (50CF)16.

50CF16 = (5\*4096) + (0\*256) + (12\*16) + (15\*1)= 2068710

## 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:

Det vil gå 1.25 ns mellom hver puls på en slik buss.

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

Svar:

Vi 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:

Hver instruksjon tar 1 ns

*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: 232\*512 = 118784 bytes = 116KB.

Siden hvert spor er delt opp i 232 sektorer som lagrer 512 Bytes hver vil vi derfor ha 118784 bytes som lagres langs hvert av de ytterste sporene.

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

Svar:

Siden vi vet at den har 7200 omdreininger i minuttet, vil den ha 120 omdreininger i sekundet.

120\*116KB = 13.6MB/s

Da får vi at vi kan overføre 13.6MB/s ved fortløpende lesing langs et spor ytterst på platen.

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

Svar:

For å finne hvor lang tid det tar for en ny byte ankommer fra disken finner vi antall bytes pr sekund:

Vi ser da at det vil ta 70 ns.

1. Anta prosessoren fra oppgave 2. Hvor mange instruksjoner utfører denne prosessoren mellom hver gang det kommer en ny byte fra disken ved fortløpende lesing?

Svar:

Vi vet fra oppgave 2 at prosessoren bruker 1 ns på hver instruksjon, og fra c) vet vi at det tar 70 ns for en ny byte ankommer fra disken. Da får vi

Vi ser da at prosessoren får utført 70 instruksjoner mellom 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:

Nanosekunder

1. DRAM

Svar:

Nanosekunder

1. SSD

Svar:

Nanosekunder

1. Harddisk

Svar:

Millisekunder

1. CD/DVD

Svar:

Millisekunder