# 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:

Med 4 bits vil man ha 24 verdier tilgjengelig. Grunnen til dette er at opkoden oppgis i binær kode i bits, som er i totallsystemet. Derfor vil den aktuelle verdien til 4 bits være 24 = 16 ulike adresser. Den laveste 00002 og høyeste 11112 i det binære tallsystemet.

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

Svar:

På denne maskinen har vi 12 bits som operand eller adresser å benytte oss av. Dette er også i totallsystemet, derfor vil antall adresser være 212 = 4096 adresser.

1. Skriv bitmønsteret nedenfor på heksadesimal form (Vis fremgangsmåten tydelig):  
   0101000011001111.

Svar:

Hexadesimal er et 16-tallssystemet, eller 4 bits (24 = 16). Derfor kan vi dele opp den binære koden i 4 bits hver, og deretter konvertere den binære koden til hexadesimal kode.

01012 = 516

00002 = 016

11002 = C16

11112 = F16

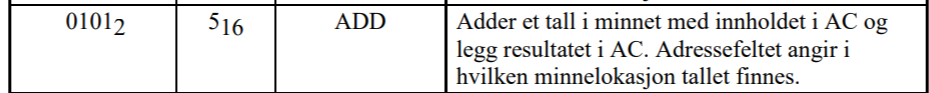
0101 0000 1100 11112 = 50CF16

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:

Opkode: 0101: Er en instruks om å legge til tallet i minnet med tallet i AC, deretter skal den legge dette resultatet i AC.

*Hentet fra kompendium 1.4 virkemåte*



Adresse: 0000 1100 11112. Dette er adressen som opkoden skal bruke når den skal hente tallet i minnet. Adressen i titallsystemet er: 256 + 128 + 8 + 4 + 2 + 1 = 399.

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:

0101 0000 1100 11112

Første tallet angir om tallet er positivt eller negativt.

0 = positivt

1 = negativt

Resten av den binære koden angir verdien.

1\*214 + 1\*212 + 1\*27 + 1\*26 + 1\*23 + 1\*22 + 1\*21 + 1\*20

= 16 384 + 4096 + 128 + 64 + 8 + 4 +2 +1

= 20 68710

## 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:

1Hz = 1 svinging per sekund.

Frekvens = antall hendelser /tid 🡪 f = n/t

T = 1/f

T = 1 / 800MHz = 1 / 800 \* 106 = 1.25 \* 10-9 = 1,25ns

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

Svar:

Antall = aksesstid / pulstid

Antall = 35ns / 1,25ns = 28 klokkepulser.

*Hvis vi hadde fått et desimal tall her må vi runde opp siden vi må vente en hel klokkepuls før vi kan sende på neste.*

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

Svar:

Antall instruksjoner = 1\*109 pr sekund

F = n / T 🡪 T = n / F = 1 / 1\*109 = 1\*10-9 = 1ns

*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:

Total byte = antall sektorer \* antall byte per sektor

Total byte = 232 \* 512B =118 784B = ca. 119kB

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

Svar:

7200 omdreininger pr minutt = 7200/60 omdreininger per sekund =

120 omdreininger per sekund.

Bytes pr sekund = omdreininger per sekund \* total byte

Bytes pr sekund = 120 \* 118 784B =14 254 080B/pr S = ca. 14,3MB/pr S

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

Svar:

Det kommer 14 254 080 Byte per sekund fra disken.

F = 1 / T

T = 1 / F = 1 / 14 254 080 = 7,0 \* 10-8 = 70 \* 10-9 = 70ns

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:

Antall instruksjoner = tid per Byte fra disk / tid per instruksjon

Antall instruksjoner = 70ns / 1ns = 70 instruksjoner.

## 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:

Aksesstiden er tiden minnet bruker på å fremskaffe ønskede data.

Typisk aksesstid for SRAM er 1 – 10 ns (nano-sekund)

1. DRAM

Svar:

Typisk aksesstid for DRAM er 35 – 70 ns (nano-sekund).

1. SSD

Svar:

Typisk aksesstid for SSD er 35 – 100 ms

1. Harddisk

Svar:

Typisk aksesstid for HDD er 5000 – 10000 ms

1. CD/DVD

Svar:

Typisk aksesstid for CD er 100 – 300 ms (nano-sekund).

Typisk aksesstid for DVD er 110 – 170 ms (nano-sekund).