## Oppgave 1. Tallsystemer og datarepresentasjon (vekt: 15%)

I denne oppgaven brukes følgende to bitmønstrene:

Bitmønster A: 00010111

Bitmønster B: 10101110

1. Skriv hvert av de to bitmønstrene på både heksadesimal og desimal form. Vis fremgangsmåten nøye.
2. 000101112

=0\*27+0\*26 +0\*25+1\*24+0\*23+1\*22+1\*21+1\*20

=16+4+2+1

=2310

=(0001)2(0111)2

=1716

1. 101011102

=1\*27+0\*26 +1\*25+0\*24+1\*23+1\*22+1\*21+0\*20

=128+32+8+4+2

=17410

=(1010)2(1110)

=AE16

## Oppgave 2. Busser (vekt: 15%)

De tre hovedtypene av busser som er omtalt i kurset er: parallell buss, seriell buss og svitsjet

buss.

1. Beskriv oppbyggingen til hver av disse hovedtypene.

* . **Parallell buss:** Denne type buss består av et sett med parallelle ledninger, noe som gjør at vi kan overføre flere bits samtidig. Det er vanlig at de består av mellom 50-100 ledninger. Disse ledningene deles inn i tre kategorier: databuss, adressebuss og kontrollbuss. På databussen overføres binær data, på adressebussen overføres (binære) adresser og på kontrollbussen skjer overføringen av signaler som styrer og synkroniserer overføringene.
* **Serielle busser:** Denne type buss overfører kun en og en bit i slengen (altså etter hverandre). Dette gjør at vi ikke trenger like lange ledninger som en parallell buss, siden bare en bit kan reise gjennom den i slengen.
* **Svitsjede busser:** Denne typen buss har en rekke tilkoblingskomponenter der hvert punkt er koblet til hver sin komponent. Svitsjen skal sette disse komponentene i kommunikasjon med hverandre etter behov, og kan koble flere komponenter parvis sammen om gangen.

1. Nevn fordeler og ulemper med hver av disse hovedtypene.

* **Parallelle busser:**
  + **Fordeler:**
    - Rask overføring av data ved korte lengder på ledningene.
    - Kan sende en større mengde data om gangen.
  + **Ulemper:**
    - Om vi skal øke båndbredden må vi enten utvide de fysiske dimensjonene til bussen, og av praktiske årsaker økes ikke denne forbi 64-bits, eller vi må øke frekvensen. Om vi øker klokkefrekvensen vil de elektroniske signalene påvirke hverandre og det vil oppstå individuelle forskjeller på tiden signalene bruker.
* **Serielle busser:**
  + **Fordeler:**
    - Mindre plasskrevende, siden vi ikke trenger like mange ledninger.
    - Kan bruke høyere hastighet enn parallelle busser ved hjelp av økt frekvens.
    - Kan bruke små dimensjoner på koblingsmatriell, lange ledninger og god båndbredde.
  + **Ulemper:**
    - Krever høyere frekvens for at overføringen av bitsen skal skje raskt.
* **Svitsjede busser:**
  + **Fordeler:**
    - Uansett hvor mange sammenkoblinger som er gjort, kan en ny sammenkobling skje, forutsatt at mottakeren er ledig.
  + **Ulemper:**
    - Teknologien tilknyttet en svitsjede buss er mer komplisert og forutsetter at alle komponentene er kompatibel med bussen.

1. Moderne datamaskiner bruker gjerne alle disse hovedtypene av busser. Nevn minst et eksempel på moderne busser fra hver av hovedtypene.

* **Eksempel på moderne parallelle busser:**
  + Minnebus på datamaskinen, dette er den hurtigste bussen vi har i datamaskinen vår (sett bort i fra bussene som er inni prosessoren).
* **Eksempel på moderne serielle busser:**
  + Denne type buss brukes som busser for IO-utstyr, også kalt IO-busser. Et eksempel på dette er USB (universal serial bus).
  + Vi bruker SATA (Serial ATA) som buss til harddisken i moderne datamaskiner.
* **Eksempel på moderne svitsjede busser:**
  + I vår moderne datamaskin bruker vi PCI-ekspress som er en svitsjet buss, og en del av brikkesettet til PCen. Dette blir omtalt som hovedbussen til datamaskinen.

## Oppgave 3. Cache og systemarkitektur (vekt: 20%)

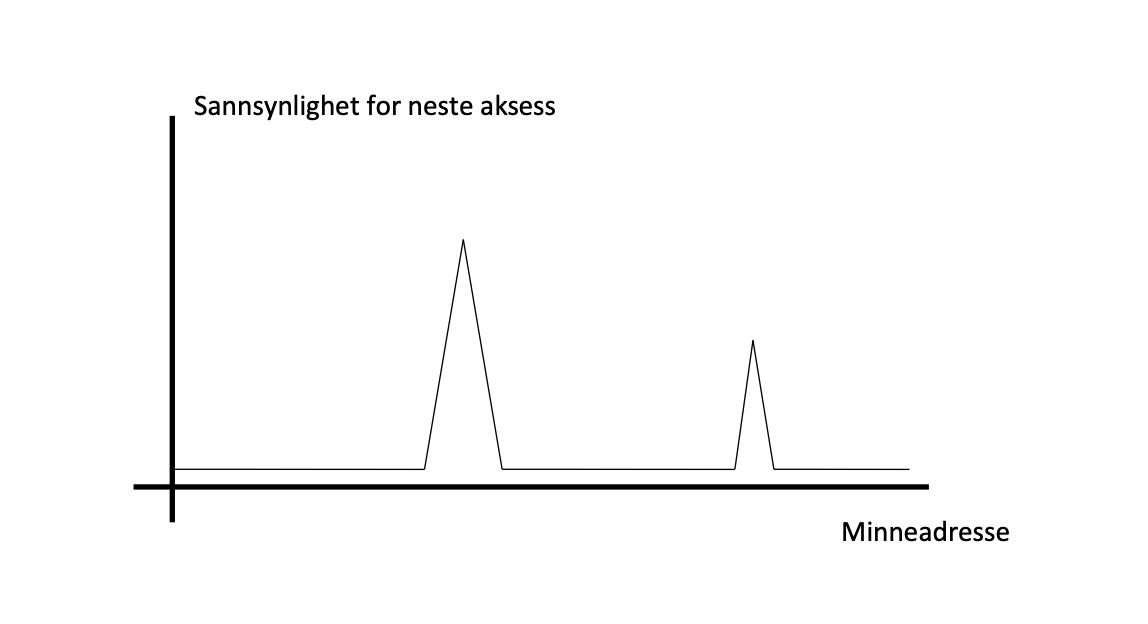
1. Hva sier prinsippet om lokalitet? Begrunn hvorfor prinsippet gjelder.

* Selve prinsippet sier at dersom en minnelokasjon er benyttet en gang, er det svært sannsynlig at den eller en lokasjon i nærheten vil bli benyttet en gang til. Grunnen til at dette prinsippet gjelder er at instruksjoner som regel ligger sekvensielt, i tillegg inneholder mange program løkker som gjentas flere ganger. VI vet også at mange programkonstruksjoner er iterative (et lite antall instruksjoner gjentas mange ganger).

1. L1-cachen til en prosessor oppgis til å være 8-veis sett-assosiativ cache med linjestørrelse på 64 byte. Forklar nøye begrepene som er understreket, og hvorfor det ofte er en hensiktsmessig oppbygging.

* At det er en 8-veis sett-assosiativ cache betyr at cachen har 8 linjer pr sett, altså 8 konkurrerende blokker kan ligge i cachen samtidig, hver av disse linjene har en størrelse på 64 byte. Vi bruker som regel sett-assosiativ cacher for å unngå trashing. Det sett assosiative cacher gjør er å gi konkurrerende blokker flere linjer, slik at flere av blokkene kan være i cachen samtidig uten å trashe hverandre, for dette reduserer hastigheten til cachen. Denne type cache brukes for den gir en rimlig avveining mellom enkelhet og fare for trashing og vil som regel ha høyest treffrate.

1. Dersom man tar et normalt program og ser hvilke minnelokasjoner som i praksis brukes av programmet, vil man se at det bruker to områder i minnet. Dette er skjematisk fremstilt i figuren nedenfor. Plasseringen av disse områdene i minnet vil bestemmes av operativsystemet.



Svar på følgende spørsmål:

- Hvorfor er det to slike områder?

- Grunnen til at vi får to slike områder er fordi operativsystemet allokerer dataområde og instuksjonsområde til prosesser. Det er dette vi ser på figuren.

- Har denne egenskapen med programmene noe å si for at sett-assosiativ cache er

den vanligste mapping-funksjonen?

- Siden instruksjonene og dataen har to separate adresser vil sannsynligheten for at de er i cachen samtidig være større, siden sannsynligheten for at de liggger i samme sett er mindre.

d) Moderne minneteknologier er spesiallaget for å utføre såkalte «burst».

- Hva er en «burst»?

-Med utgangspunkt i det vi har lært om systemarkitektur: Forklar detaljert hvordan en «burst» utføres.

-Burst er kontinuerlig strøm på et antall bytes fra minnet til CPUen. Når CPUen aksesserer minnet vil den be om en adresse fra RAM og minnet vil finne fram denne lokasjonen. Når overføringen først starter vil minnet fortløpende sende et antall bytes i rekkefølge etter hverandre fra den lokasjonen som ble etterspurt og utover. I moderne DDR-SDRAM dette to ganger pr klokkesyklus (double data rate). Denne «blokken» legges så i en linje i cachen.

## Oppgave 4. Prosesser og tråder (vekt: 25%)

a) Vi snakker ofte om både prosesser og tråder. Forklar hva en prosess er og hva en tråd er

og hvilken sammenheng det er mellom prosesser og tråder.

* En prosess er kjørende program. Ved kjøring av et program oppretter operativsystemet en prosess i minnet med eget data- og instuksjonsområdet. En tråd er det som kjører i prosessene, og alle prosessene har minst en tråd. Men de kan inneholde flere tråder, der alle trådene har sin egen stakk og dataområde, men med felles instruksjoner. Alle trådene deler på prosessens dataområde noe som gjør at de kan utveksle informasjon og samarbeide. Hver prosess har sin egen PCB som inneholder PID (prosess ID), status om kjøring osv.

1. Hvordan er det mulig å kjøre flere tråder på en CPU enn det er tilgjengelige kjerner i CPU-en? Forklar.

-Pga prosessadministratoren i operativsystemet har en schduler. Denne gir hver prosess et gitt tidskvant og setter prosessen tilbake i CPU-køen etter denne tiden er oppbrukt (kvasiparallell). Dette lar CPU-en betjene mange tråder i rask rekkefølge noe som gjør at vi får simuleret simultan kjøring aka multitasking.

1. Tjenerprogrammer, f.eks på en filtjener eller på en webtjener, er som oftest såkalt

”multithreaded”, dvs kjører mange tråder. Hvorfor er det hensiktsmessig å kjøre mange tråder på f.eks en filtjener sammenlignet med å kjøre kun en prosess? Forklar.

-Om vi hadde brukt prosesser måtte vi ha ventet på at hver prosess skulle gjøre seg ferdig før neste kunne ha begynt, mens om vi bruker tråder kan vi bruker flere samtidig noe som gjør at responstiden blir kortere, og cpu-tiden blir brukt mer effektivt. Det vil derfor ta mye lenger tid om vi hadde brukt en prosess i motsetning til bruk av tråder.

d) Prosesser får tilgang til i/o-enheter via operativsystemkjernen. På hvilken måte? Forklar.

-Prosesser går i kjernemodus og spør kjernen om de får bruke i/o-utstyret (systemkall), det er dermed opp til kjernen å administrere køen. Kjernen sjekker om prosessen som etterspør i/o-utstyret har rettighetene til å bruke den og setter deretter prosessen i kø.

e) Hvorfor kan ikke prosessene gå direkte mot i/o-enhetene og gjøre sin egen i/o uavhengig

av operativsystemkjernen? Forklar.

-Operativsystemets oppgave er å administrere i/o-utstyret, uten operativsystemet ville det ikke vært noe form for system på hvilken prosess som fikk bruke i/o-utstyret og hvor lengde den fikk ha den i bruk. Dette er en viktig del for at prosessene skal gå rundt siden de vil ligge i blokkert tilstand helt til de har fått tilgang til ressursene de trenger. I tillegg til å administrere køen tilbyr kjernen gode og sikre metoder for å bruke utstyret.

## Oppgave 5. Minne (vekt: 25%)

a) I minneadministrasjon snakker man om relokering av adresser som betyr omregning fra

virtuelle adresser til fysiske adresser. Forklar hvorfor man trenger denne omregningen fra

virtuelle til fysiske adresser. Forklar også hvordan denne omregningen foregår i dagens

operativsystemer som f.eks i Windows eller Linux (ikke tekniske detaljer her, kun

prinsipper om hvordan det foregår).

* Vi trenger relokering fordi programmer ikke blir plassert på samme plass i minnet hver gang det kjører, med relokering har ikke dette noe å si for da omregner CPUen den virtuelle adressen om til en fysisk adresse som får plass på det fysiske minnet (RAM).

Denne beregningen utføres av CPUen.

b) Du har en datamaskin med 64 bits CPU. Fysisk minne er på 16 Gbyte. Men hvor stort er

det virtuelle minnet? Jeg er ikke ute etter et eksakt tall her, men en diskusjon rundt

temaet, f.eks vil jeg vite noe om både hvor stort virtuelt minne operativsystemet kan sette

av og hvor mye virtuelt minne hver prosess disponerer.

* Størrelsen på det virtuelle minnet er avhengig av størrelsen på harddisken, mens størrelse på det fysiske minnet er avhengig av størrelsen på RAMen.