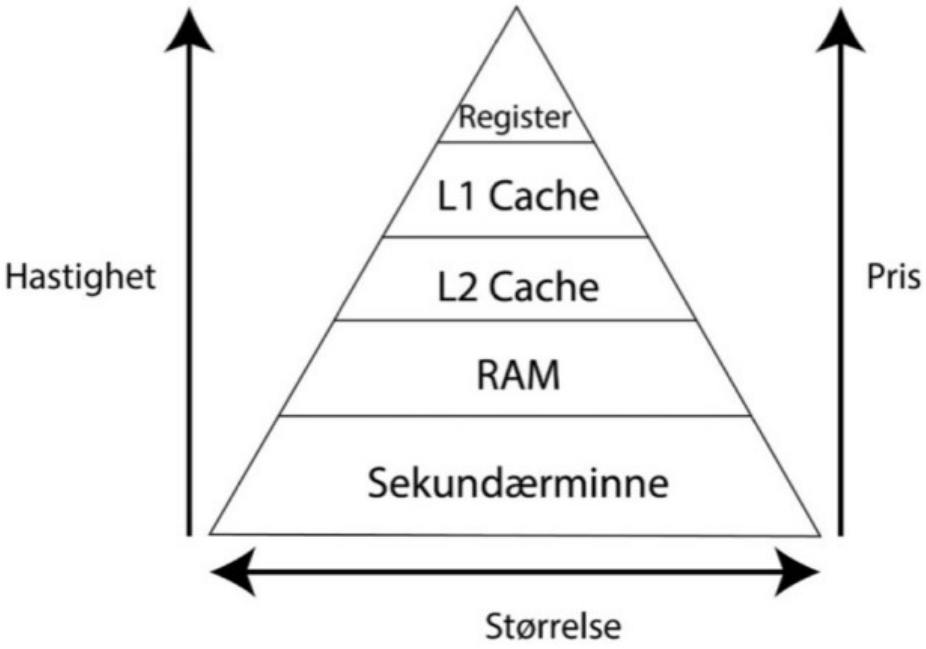
Minnehiearki

- All data er lagret i en form for minne.
- De har ulik hastighet (i forhold til CPUen), pris og størrelse(kapasitet).
- Øverst i hierarkiet har vi det som ligger nærmest CPU-en og som er raskest, men disse koster også mest å lage
- Nederst i hierarkiet er det de som har størst lagringskapasitet



Minnehierarki – aksesshastighet

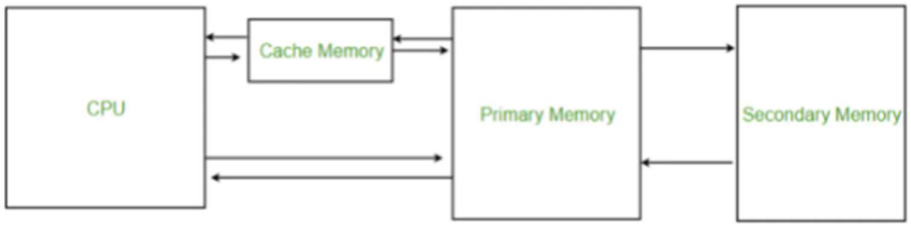
Registers	< 1ns	≈ 100 Byte
L1 (på CPU) cache	≈ 1ns	≈ 10 KB
L2,L3 (utenfor CPU) cache	2-10ns	≈ 1 MB
Hovedminne (RAM)	20-100ns	≈ 1 GB
SSD/Flash	100ns-1us	≈ 1 TB
Harddisk	1ms	≈ 1 TB

Registere:

- Mini minne for å lagre instruksjonen, minneadresse (pasifisert av instruksjonen), lagring for en bit med data, osv.
    - I LMC: 4 registere: "Instruction register", "adress register", "program counter ", "accumulator"
- Bruk: I de fleste moderne datamaskinarkitekturer følger man prinsippet om at data/instruksjoner leses fra minnet og inn i registere før bruk, og deretter lagre resultat tilbake til minnet fra et register.
- Registre er raskt å nå for CPUen, høy aksesshastighet. Ligger jo inni CPUen. Veldig dyre.

Cache

- Mellomlager mellom cpuen(prosessoren) og Minne(RAM).
- Er for å gjøre prosessoren raskere.



Cache hit og cache miss

- Når en prosessor skal hente en instruksjon eller lese/skrive data vet den ikke om den skal hente fra RAM eller Cache.
- Hvis det prosessoren ber om ligger i cache får vi cache hit.
- Hvis dataen ikke ligger i cache får vi cache miss.

Ukesoppgave 3 om cache hit og cache miss:

Anta at en prosessor vil prosessere en instruksjon per klokkesykel og skal totalt prosessere 1.000.000 instruksjoner som typisk vil ha behov for å hente data fra minne for 60% av

instruksjonene, hvor det vil uheldigvis være 50% cache-miss. Anta videre at det typisk vil være en penalty (økt forsinkelse) for cache-miss på 10 klokkesykler. Hva mange klokkesykler vil det hele ta?

Svar:

- Instruksjoner som leser/skriver data fra/til minnet(minneaksessering):
    - Om vi får en cache miss vil det ta + 10 klokkesykler, altså 11 klokkesykler
    - Om vi får en cache hit tar det 1 klokkesykel
- Instruksjoner som ikke krever minne aksessering:
    - 1 klokkesykel

- 40 % av instruksjonene trenger ikke minneaksessering -> 400 000 -> 400 000 * 1 klokkesykel = 400 000 klokkesykler
- 60 % av instruksjonene trenger mineaksessering -> 600 000
    - 50 % cache-miss
        - 300 000 -> cache hit -> 300 000 * 1 klokkesykel = 300 000 klokkesykler
        - 300 000 -> cache miss -> 300 000 * 11 klokkesykler = 3 300 000 klokkesykler

- 400 000 + 300 000 + 3 300 000 = 4 000 000 klokkesykler totalt

Eksamensoppgave 2019 om cache hit og cache miss (modifiserte tall)
Anta at prosessoren har 360 instruksjoner igjen å utføre. Det tar én klokkesykel pr instruksjon, og man regner med å ha en minneaksessering på 70% og med cache-miss på 50%. En cache-miss fører til en total tidsbruk på 20 klokkesykler. Cache-hit bruker totalt én klokkesykel. Hva er totalt antall gjenstående klokkesykler?

Svar:

- Instruksjoner som leser/skriver data fra/til minnet(minneaksessering):
    - Om vi får en cache miss tar det 20 klokkesykler
    - Om vi får en cache hit tar det en klokkesykel

- Instruksjoner som ikke krever minne aksessering:
    - 1 klokkesykel

- 30% av instruksjonene trenger ikke minneaksessering -> 360*0,3 = 108 klokkesykler
- 70 % av instruksjonene trenger mineaksessering -> 360*0,7 = 252 klokkesykler
    - 50 % cache-miss
        - 126 -> cache-hit -> 126 * 1 = 126 klokkesykler
        - 126 - > cache-miss -> 126 * 20 = 2 520 klokkesykler

- 108 + 126 + 2520 = 2754 klokkesykler totalt

RAM:

- Radom access memory
- Primærminne
- Ting lagret i RAM forsvinner når vi skrur av datamaskinen.

Sekundærminne:

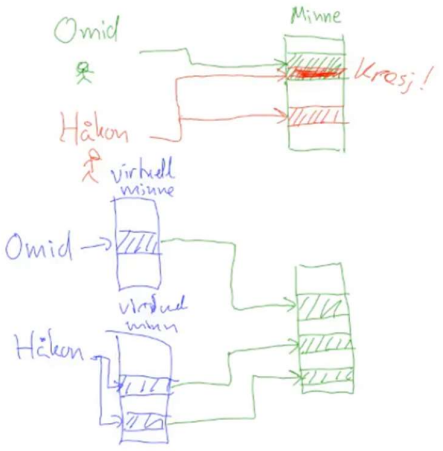
- Hard disk drive (HDD) og Solid State Drive (SDD) er eksempler på sekundærminne.
- På for eksempel harddisken på pcen, lagres hele programmer og data, som ikke forsvinner om pcen skrus av, i motsetning til RAM
- Har stor kapasitet til lagring.

Andre typer minne:

1. Virtuelt minne

- Ikke et eget «minne-sted»
- To brukere kan ha hver sin virtuelle minneblokk. De to kan lagre noe på samme plass i hver sin virtuelle minneblokk. Men når disse to brukerne skal lagre dette fysisk på for eksempel en PC, så oversetter operativsystemet den virtuelle minneadressen(der de har lagret noe begge to) til to forskjellige fysiske minneadresser. Dette gjør at når

vi skal lagre noe fysisk på for eksempel PCen forstyrrer ingen minneblokker hverandre.



- (Fra IN2110 Informasjonsikkerhet)(ikke IN1020 pensum, bare her for forståelse). X86-64 har støtte for virtuelt minne; for overføring av data mellom Minnet(RAM) og sekundærminne. Virtuelle minneadresser oversettes av operativsystemet til fysiske minneadresser. Dette gjør at om vi kjører en prosess som inneholder skadevare(for eksempel virus), har den ikke har tilgang til fysiske adresser for andre data og prosesser, bare sit eget virtuelle minneområde.

2. Flip flop

- Flip Flop er en hardware minne del som holder på 1 bit data.
- Den er den raskeste enheten og byggesteinen til alle andre minne-enheter.

Annet:

BUS

- BUS sørger for kommunikasjon mellom komponenter.
    - For eksempel mellom CPUen og RAM
- Det finnes flere typer BUS som sørger for at forskjellig type informasjon blir kommunisert mellom forskjellige typer komponenter. I CPUen:
    - Address BUS: Overfører minneadresser fra prosessoren til andre komponenter slik som primærminne og input/output komponenter
    - Data BUS: Overfører data mellom prosessoren og andre komponenter. (går begge veier)

- Control BUS. Overfører kontrollsignaler(klokkesignaler) fra prosessoren til andre komponenter.