Velkommen!

Pipeline og minnehierarki

Gruppetime 8, pensum fra uke 41

(Hanne-)Kristin Schrøder Leiros

hkleiros@uio.no

Plan for dagen

- Kort repetisjon av Teknikker for hastighetsøkning
- Jobbe med ukesoppgaver i plenum
- Obligspørsmål

Oblig 3 har frist om 6 dager

Gjerne send meliding i mattermost eller mail til hkleiros@uio.no om dere har noen spørsmål eller trenger hjelp ☺

Bekreftet 2 timers hjemmeeksamen

9. Desember 09:00 – 11:00

Inspera

Fra forrige uke: regne ut frekvens

- Hvis vi har en port med 5ps delay, vil en klokkeperiode være 10ps
  - Forutsatt 50% -50% klokkesignal
- Frekvens = (1 / klokkeperiode) * 1000
- Frekvens = (1 / 10 ps) * 1000= 100 GHz

- Eventuelt bruke utregningsverktøk som:
  - https://www.sensorsone.com/period-to-frequency-calculator/?frval=10&frunit=picoseconds+%28ps%29&frfctr=psec&tounit=gigahertz+%28GHz%29&tofctr=GHz

Læringsmål

- Kunne samtlige optimaliseringer I hardware
  design, som pipeline, minnehierarki og multi-core prosessorer

Klokkesykels CPU

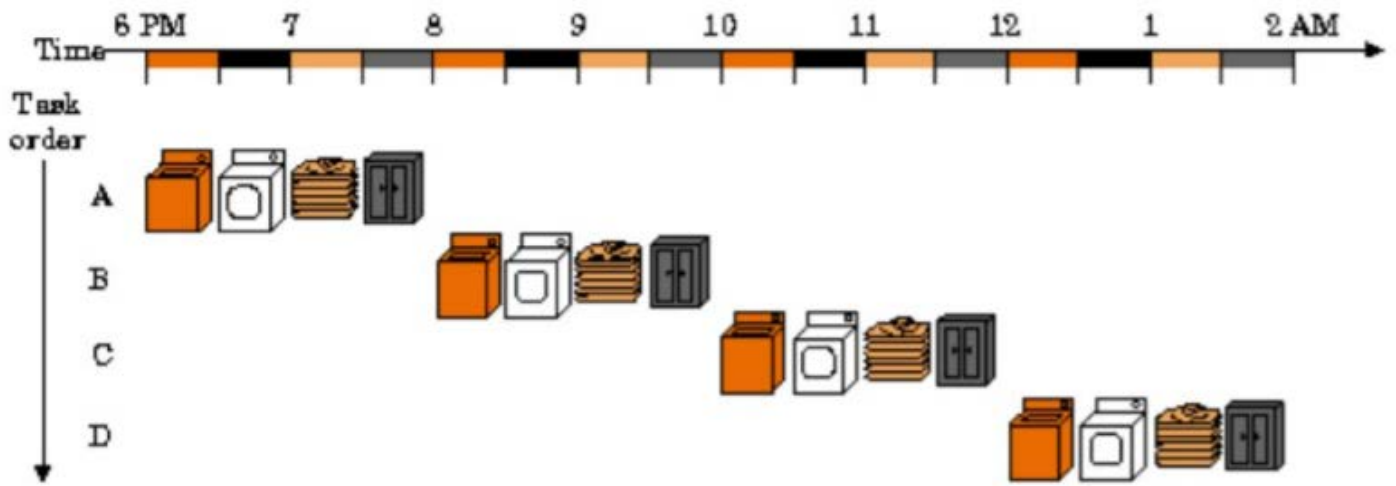
- Klokkesignalet gjør det mulig å synkronisere forskjellige operasjoner til å ta like lang tid
- Klokkeperioden må være like lang som den lengste forsinkelsen for enhver instruksjon gjennom data-pathen
- Raskere instruksjoner tvinges dermed til å bruke mer tid enn nødvendig

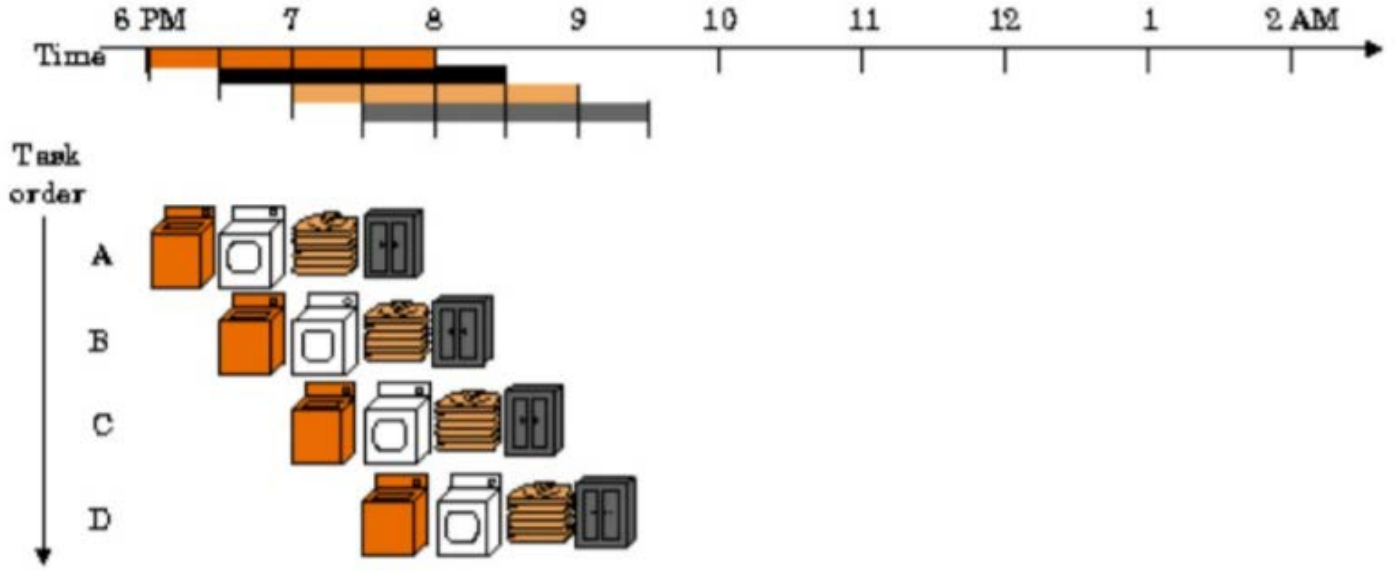
Bus

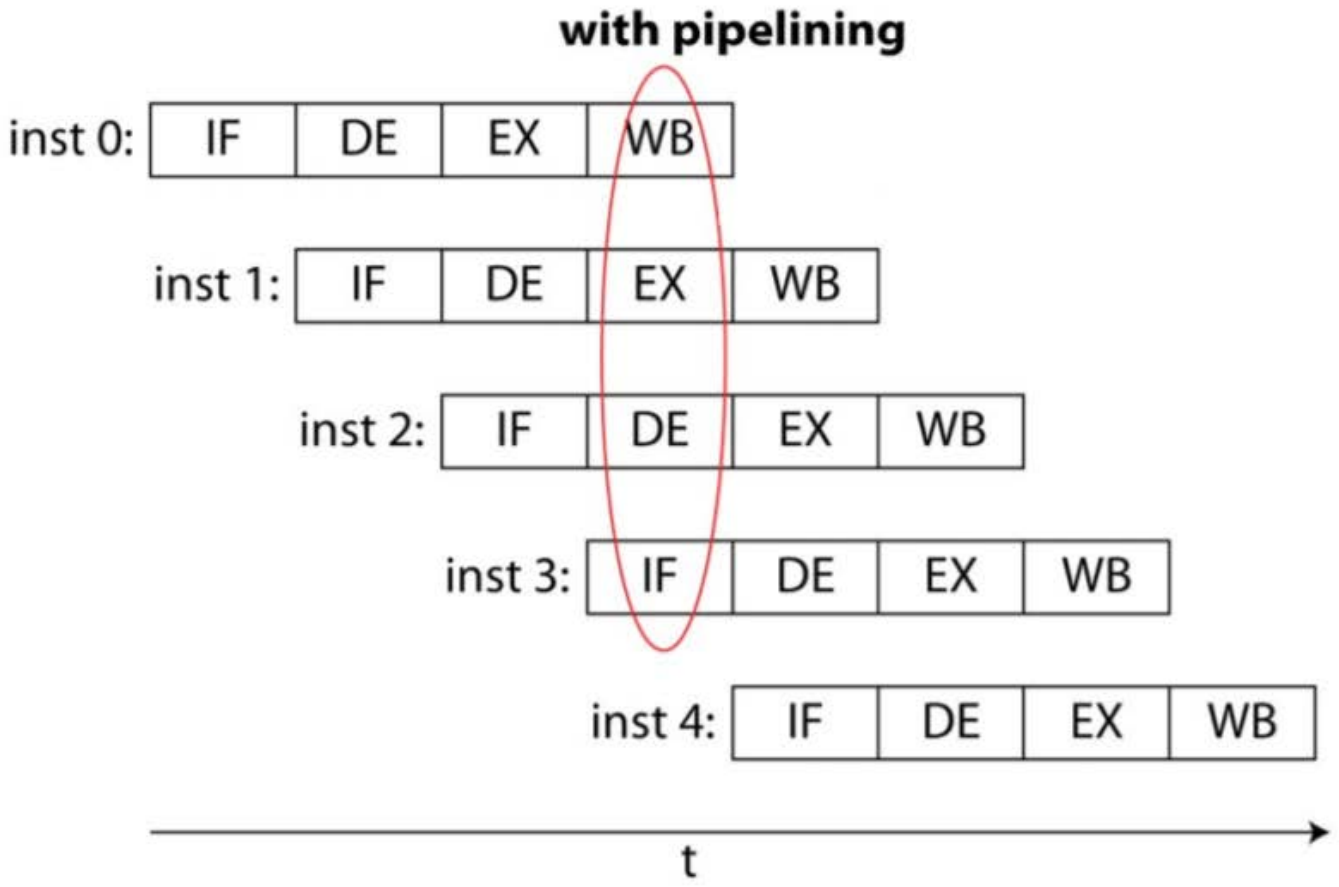
- Kommunikasjonskanal mellom registre, ALU, minne og I/O enheter
- Kan deles mellom flere enheter, men kun én kan sende om gangen
- Internt i en CPU er det en eller flere bus(er) som overfører data mellom interne registre

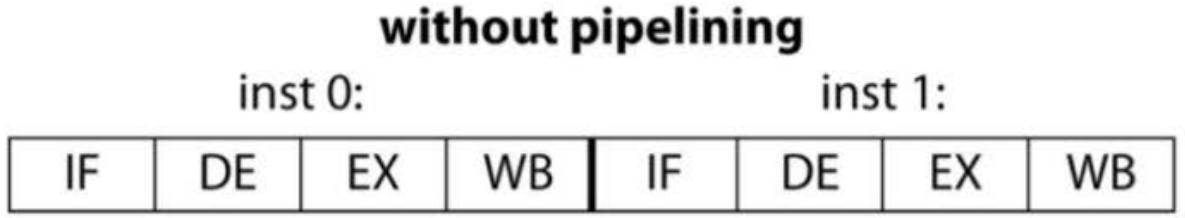
Pipeline

- Samlebåndsprinsipp
  - Algoritme / metode for å effektiviser gjennomføringen av instruksjoner
  - Nesten alt som foregår i en datamaskin kan effektiviseres med en pipeline
- Dele opp instruksjonen i uavhengige deler
- Neste instruksjon settes i gang før forrige instruksjon er helt ferdig
  - Hver instruksjon tar like lang tid å utføre
  - Krever bestemt rekkefølge på instruksjonene


6 PM
7
8
9
10
11
12
1
2 AM
Time
Task order
A
B
C
D


6 PM
7
8
9
10
11
12
1
2 AM
Time
Task order
A
B
C
D

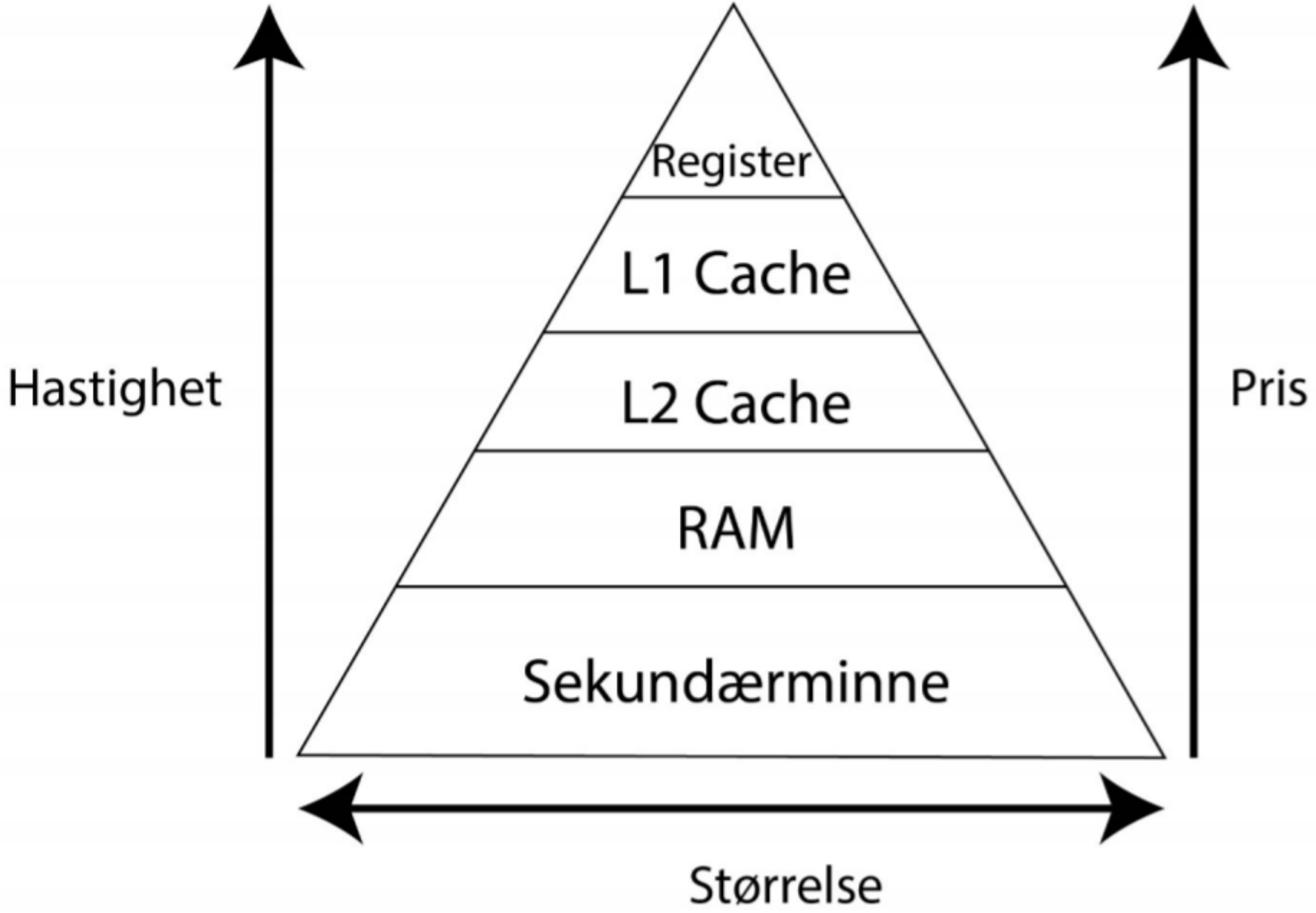

with pipelining
inst 0:
IF
DE
EX
WB
inst 1:
IF
DE
EX
WB
inst 2:
IF
DE
EX
WB
inst 3:
IF
DE
EX
WB
inst 4:
IF
DE
EX
WB
t


without pipelining
inst 0:
inst 1:
IF
DE
EX
WB
IF
DE
EX
WB

Husk!

Dette betyr ikke at det går 4 ganger raskere

Vi mister alltid noe tid til administrering av instruksjoner og komplikasjoner


Register
L1 Cache
L2 Cache
RAM
Sekundærminne
Hastighet
Pris
Størrelse

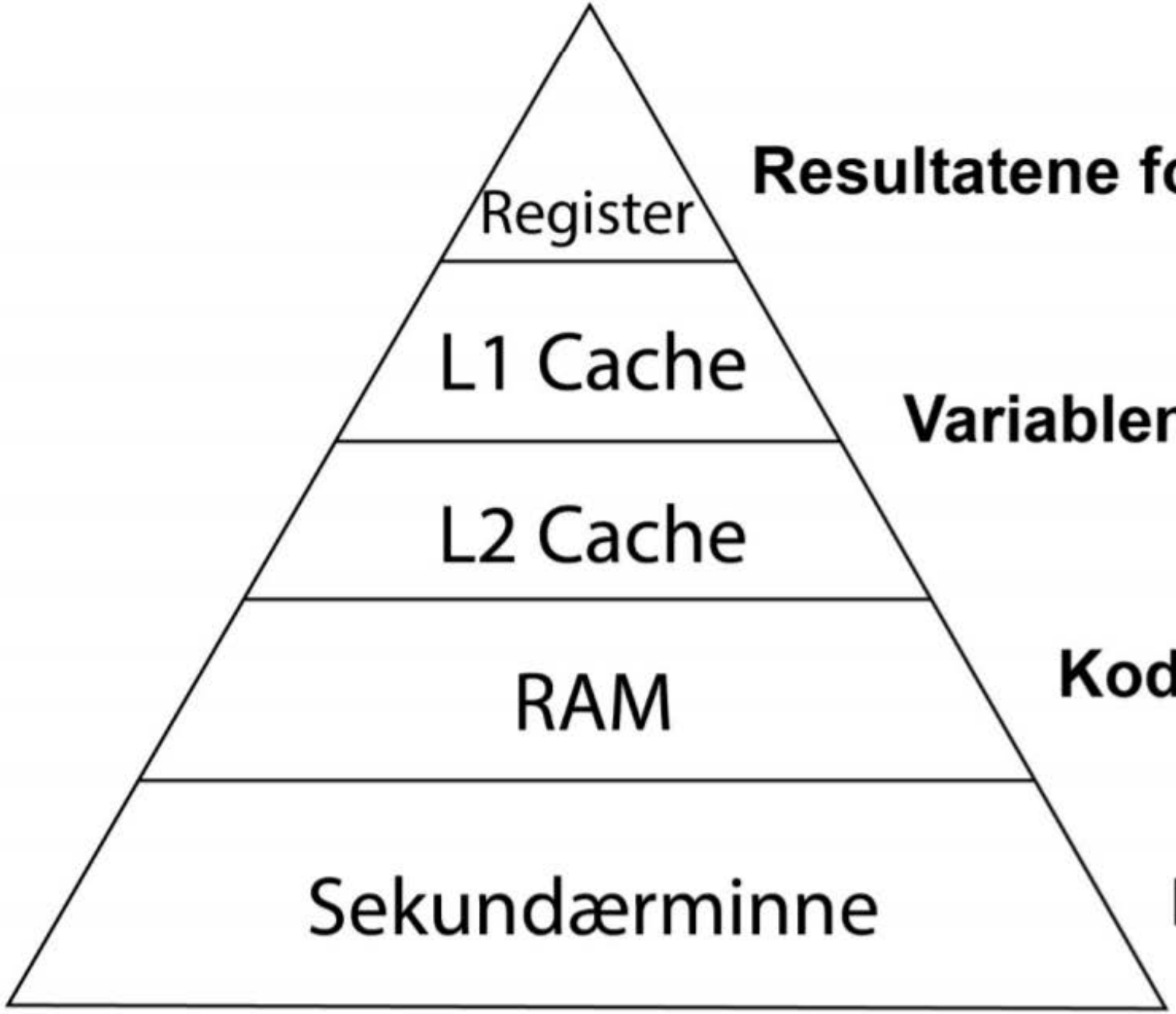
Typer minne

- Register
  - Internminnet / intern kladdeblokk i CPUen med rask aksess til innholdet (som i LMC)
  - Ca. 100 B
- Cache
  - Liten mengde hurtig mellomlager for både instruksjoner og data
  - Gjør at man ikke må skrive og lese så ofte fra RAM og man unngår flaskehalser
  - Mellomlager internt (L1) eller i nærheten (L2,L3) av CPU-en
    - Typsik kapasitet fra 10 KiloByte (L1) til 1 eller flere MegaByte (L2, L3)
- RAM
  - Buffer mellom ekstrent lagringsmedium og CPU-en med rask skrive og leseaksess
  - Internt minne på hovedkortet i nærheten av CPU-en
  - Opptil flere GigaByte med plass
- SSD/Flash/Harddisk
  - Høykapasitetsmedium for program / data
  - Ekstern eller intern lagringsenhet i maskinen med kapasitet opptil flere TeraByte

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

Mentimeter



Resultatene for hver iterasjon

Variablene og telleren

Koden til For-løkke

Hele programmet

Cache hit og cache miss

- Cache miss er når et forsøk på å lese eller skrive til cache feiler
  - Man må da hente dataen i annet minne f.eks. RAM
- Cache hit er når man finner dataen i cachen
- De fleste CPU-er har opptil 90% cache hit

- Cache minne er delt i lag eller cache hierarki
  - Nermeste er level-one cache eller L1 som er direkte på prosessoren
  - Neste er L2 og L3 som ikke er på prosessoren men veldig nært og har mer plass

Oppsummert

- Teknikker for hastighetsøkning i datamaskiner:
  - Pipeline:
    - Øker antall instruksjoner som utføres pr. sekund
  - Flere CPUer i en maskin
  - Raskere disker
  - Hurtigere og større RAM