Prosessoren og abstraksjonsnivåer

Av: Elise Eknes

Gruppe 15: Introduksjon til datateknologi

Viktig!

1. Ikke ferdig med oblig? Se forelesning 15.10.25 for tips
2. Sattelit frist: 27.10.25
3. Maskinvare: Siste pensum del! Kanskje den mest tyngste



Maskinvare

Kompendium: https://www.uio.no/studier/emner/matnat/ifi/IN1020/h25/ressurser/2025-10-15-kompendium.pdf , Les før timen!
Motivasjonspitch: Les de før timen!

Hvert kapittel i kompendiet er hver forelesning. Fin forberedelse til forelesningen!

Læringsmål:

1. **Kjenne til de forskjellige abstraksjonsnivåene:**
    a. Oppnå kjennskap til og kunne beskrive de ulike abstraksjonsnivåene i et datamaksinsystem, inkludert maskinvare, mineherarki, maskinkode og programmeringsspårk
    b. Lær hvordan abstraksjonsnivåene påvirker design og implementasjon av programvare og maskinvare komponenter

Snart eksamenstid

- Allerde nå burde dere begynne å tenke på eksamen
- Tre eksamen kort etterhverandre, disponer tiden godt!
- MAKI
  - IN1000 (5.12)
  - IN1020 (8.12)
  - MAT1080 (12.12)
- DESIGN
  - IN1050 (28.11)
  - IN1000 (5.12)
  - IN1020 (8.12)

Generasjoner av prosesser

- Hvis vi kan kontrollere strøm kan vi bygge en datamaksin - vi ser på hvordan vi styrer den (raskere og mindre fysisk)
- I IN1020 regner vi med at vi har 4 generasjoner med prosesser

1. Radiorørteknologi, tiden da “en datamaskin = et helt rom”
2. Transistorer
3. Integrerte kretser
4. Mikroporsessorer

tiden da “en datamaskin = et helt rom” → Overganger til integrasjon og miniatyrisering → (NÅ) Mikroprosessorer, som er raskere

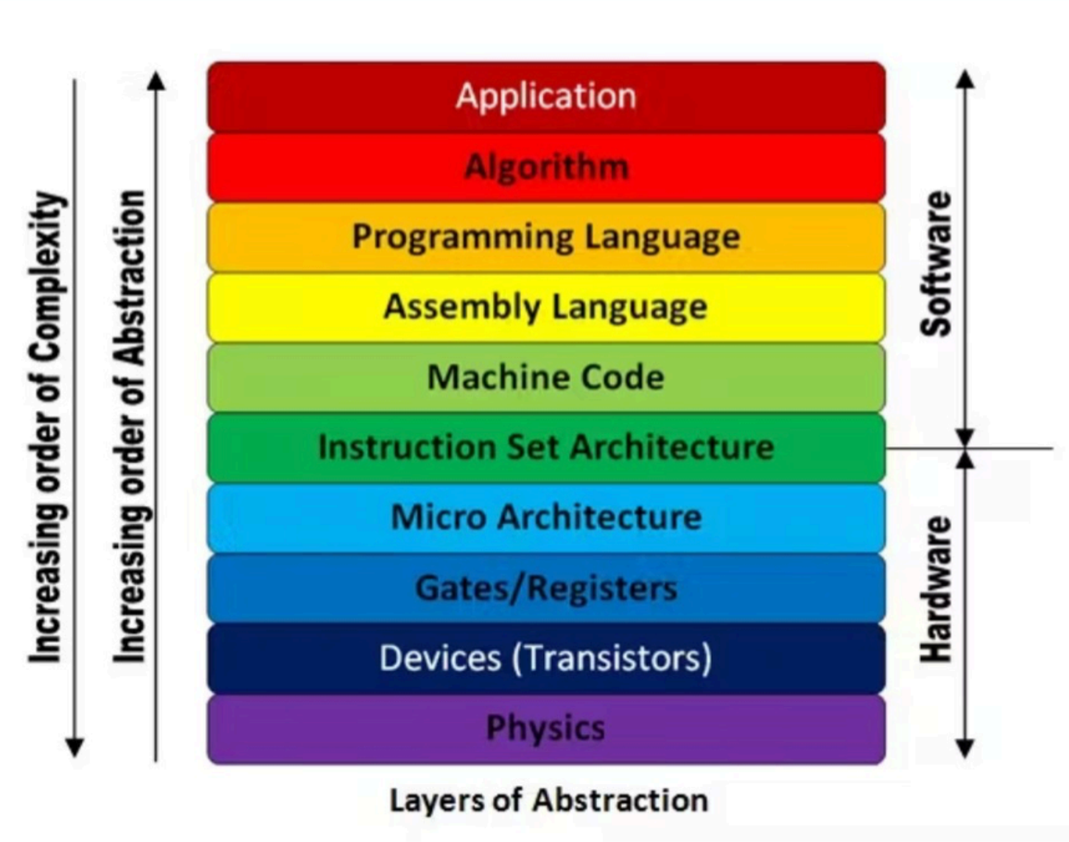








Abstraksjonsnivåer

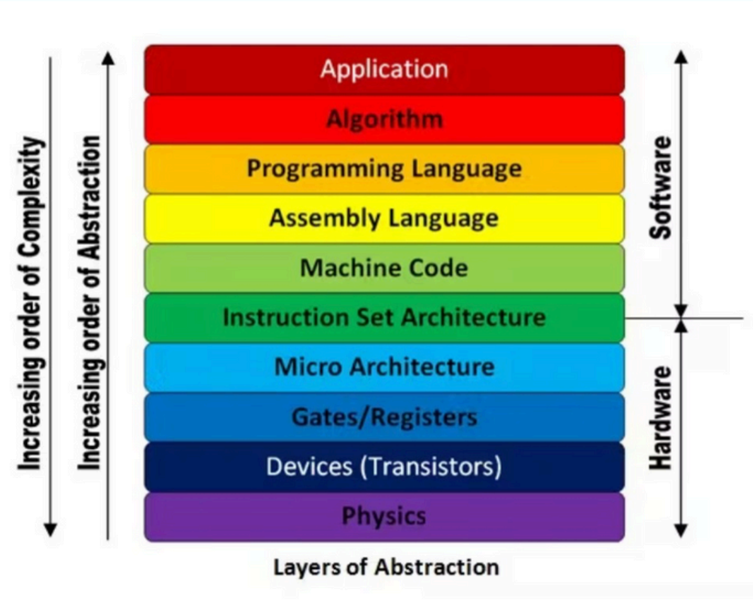


Ryggraden for hvordan vi kan forstå og jobbe med komplekste datamaskinsystemer

Hvofor trenger vi abstraksjon?

- Verden er full av komplekse systemer: fra smarttelefoner til datasentre
- Abstraksjon = verktøy for å håndtere kompleksitet
- Gjør det mulig å:
  - fokusere på det viktige
  - samarbeide på tvers av nivåer
  - bygge videre uten å forstå alt

Spørsmål: Hva betyr egentlig abstraksjonsnivå?



Abstraksjonsnivå

- Et nivå i et hierarki som skjuler kompleksiteten under
- Hvert nivå:
  - Bygger på nivået under
  - Tilbyr et grensesnitt oppover
  - Beskytter mot unødvendige detaljer
- **Eksempel:** Når du kjører bil så bruker du rattet og pedaler, du signaliserer ikke til motorstyringsenheten

Applikasjoner

↑

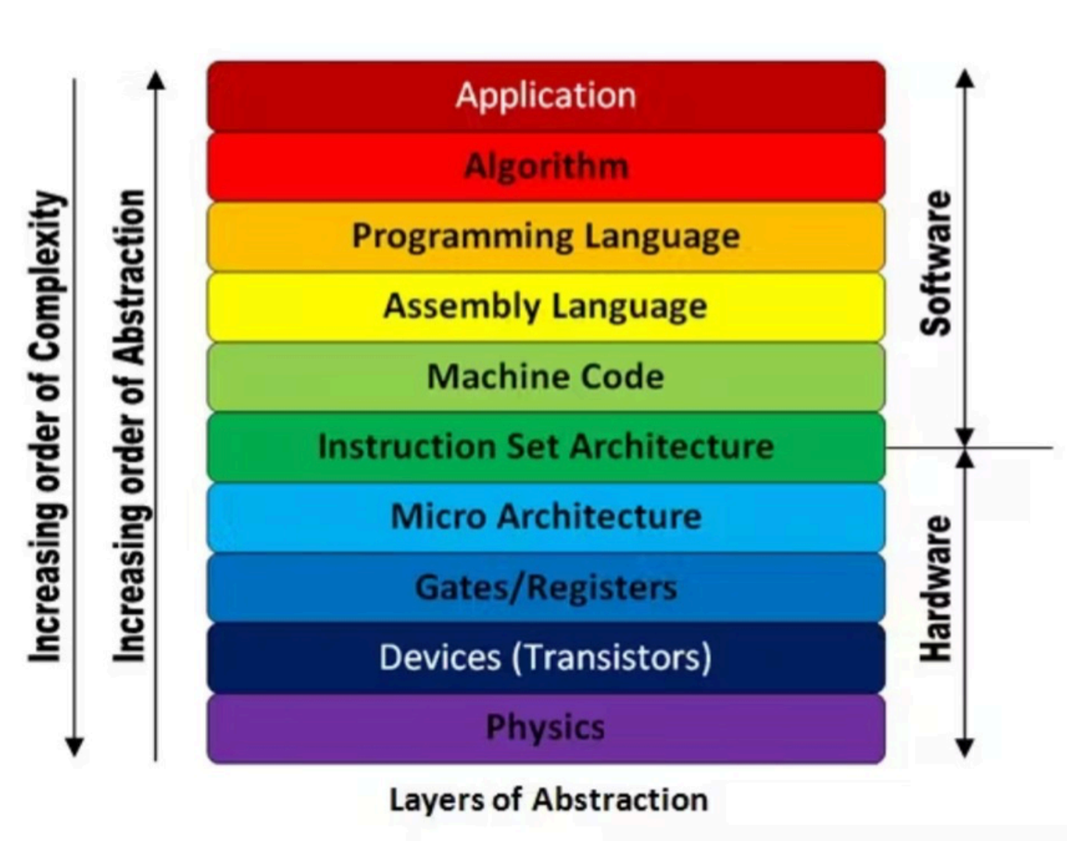
Operativsystem

↑

Maskinvare

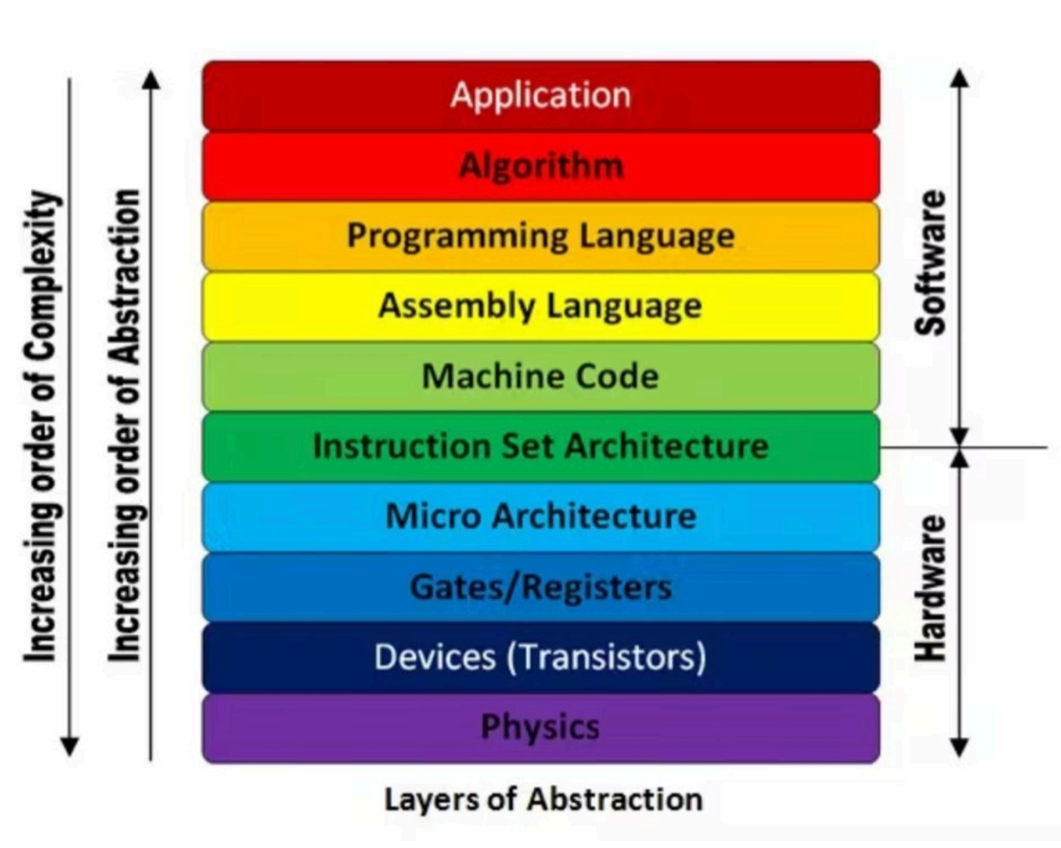
↑

Transistorer

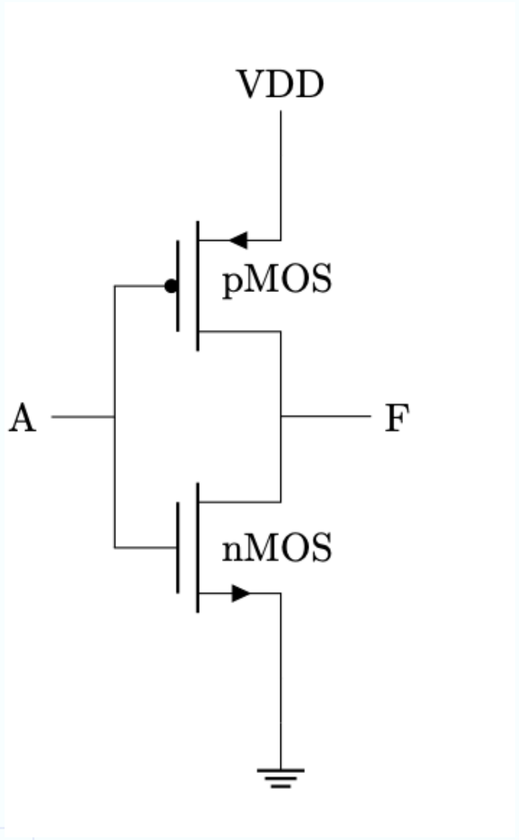


Lagene

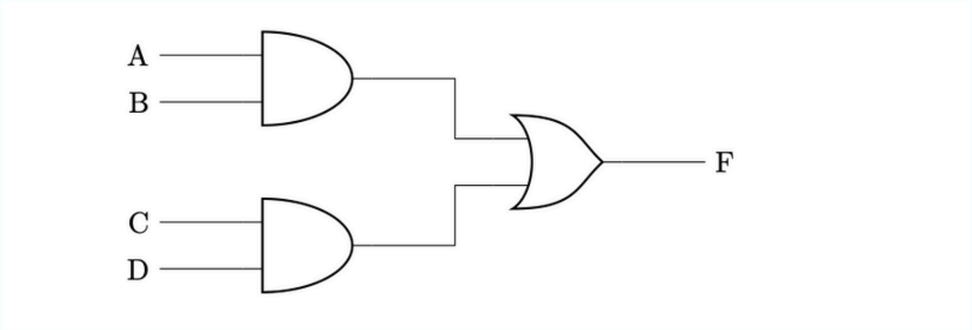
- **Fysikk:** Elektroner som flyter i materialer
- **Transistorer:** Små brytere som styrer strøm (1 og 0)
- **Logiske porter / mikroarkitektur:** Hvis strøm her -> strøm der, altså grunnlogikken
- **Maskinkode:** 1- og 0-instruksjoner CPU forstår
- **Assembly:** Mer lesbart- symbolske navn for instruksjoner (LMC)
- **Programmeringsspråk:** Python, Java- lettere for mennesker å lese
- **Algoritmer og applikasjoner:** Programmer, spill, apper

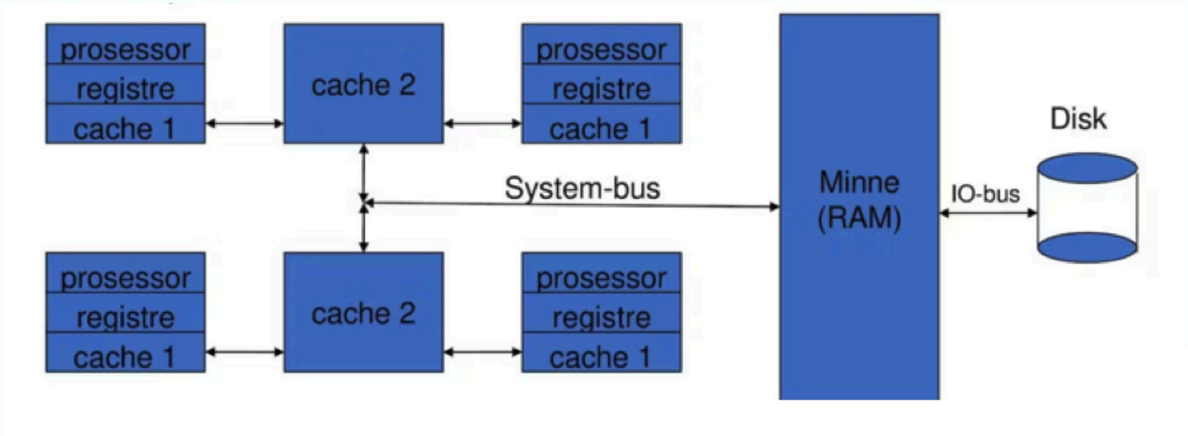


Teorietisk men praktisk arbeid!



A	B	F
0	0	1
0	1	0
1	0	0
1	1	0





Quiz

Ukesoppgaver, oblig