

## Lab DM534

### Uge 37

Disse lab-timer handler om programmering i maskinsprog. Til bogens simple CPU og tilhørende maskinsprog (beskrevet i afsnit 2.2–4 og appendiks C) findes en simulator, der kan visualisere programmer og deres udførsel.

Klik på linket til simulator på kursets webside. Har din browser rette Java-indstillinger kan den køre simulatoren direkte, ellers kan filen gemmes og køres i en shell med kommandoen `java -jar BrookshearMachine.jar`.

Når simulatoren er startet, vises et vindue med en CPU og hukommelsesceller som på figur 2.10. Maksimer evt. vinduet for en bedre oplevelse. Simulatorens virkemåden er rimelig klar - fyld blot hukommelse med program (og data) ved direkte indtastning. Værdien af program counter kan også ændres. Tryk på **Step** (eller **Run**) for at udføre det indtastede program. Bemærk: i “Memory List” view/fane er hukommelsescellerne slået sammen parvis (så der skal skrives fire hexadecimalle cifre i stedet for to ad gangen).

Man kan også skrive programmer linie for linie i en tekstbox, hvilket er hurtigere. Vælg menupunkt **File->Import/Export** for at få en tekstbox, hvor man kan få udskrevet (export) og indlæst (import) indhold i hukommelsescellerne. Formatet er simpelt: hver linie svarer til en (dobbelt) hukommelsescelle, så skriv fire hexadecimalle cifre per linie. Tomme linier giver tomme celler.

Brug menupunkt **Help** for f.eks. af få liste over alle kommandoer i maskinsproget.

Lav flg. opgaver. Arbejd gerne i par.

1. Chapter Review Problem 14, side 117. Udtænk først svarene på spørgsmålene, efterprøv dem så via simulatoren.
2. Chapter Review Problem 15, side 117.
3. Chapter Review Problem 17, side 117.

4. Chapter Review Problem 18, side 118. Her er det nemmest at taste direkte i hukommelsescellerne (import kræver et stort antal tomme linier).
5. Chapter Review Problem 23, side 118.
6. Chapter Review Problem 33 (b, c), side 119.
7. [Udfordrende] Chapter Review Problem 41, side 120.
8. [Udfordrende] Chapter Review Problem 42, side 120. Hint: Brug negering. Hint: negeringsalgoritmen i figur 1.22 kan også udføres ved at invertere (“complement”) *alle* tallets bits, og derefter lægge 1 til. Hint: XOR.