

Lab DM534

Til labtimerne (TL) behøver man ikke forberede sig, men skal blot arbejde med opgaverne undervejs i timerne. Dette er anderledes end til eksaminatorietimer (TE), hvor det som sagt forventes at man har forsøgt at løse opgaverne inden timerne.

Uge 37

Disse labtimer handler om programmering i maskinsprog. Vi skal bruge et program `BrookshearMachine.jar`, som simulerer og visualiserer en simpel CPU, med et kommandosæt beskrevet i slides om CPUer.

Klik på linket til simulatoren på kursets webside for at downloade det. Du skal have Java installeret for at køre det på din egen maskine. Du kan også køre det på maskinerne i IMADAs computerrum.

Afhængigt af din maskines sikkerhedsindstillinger kan du enten køre det direkte fra download (normalt via ikon i bunden af browser, vælg “Run” eller “Open”, mens du ignorerer eventuelle sikkerhedsadvarsler), eller du kan køre det gemte program i et terminalvindue med kommandoen `java -jar BrookshearMachine.jar`. (Hvis du ikke kan få lov at gemme filen via Google Chrome, så prøv en anden browser, eller download den direkte fra kommandolinie med kommandoen `wget urlTilFil`).

Når simulatoren er startet, vises et vindue med en CPU med registre og program counter til venstre og en række hukommelsesceller (RAM) til højre. Maksimer evt. vinduet for en bedre oplevelse. Simulatorens virkemåden er rimeligt ligetil - fyld blot hukommelse med program (og data) ved direkte indtastning. Værdien af program counter kan også ændres. Tryk på **Step** gentagne gange (eller **Run**) for at udføre det indtastede program. Både bitmønstre i celler og numre på celler og registre angives med hexadecimal notation (se sidste side i slides om repræsentation af tal). Bemærk: i “Memory List” fane er hukommelsescellerne slået sammen parvis (så der

skal skrives fire hexadecimalle cifre ad gangen i stedet for to), og kun hver anden celleadresse er angivet.

Man kan også skrive programmer linie for linie i en tekstboks, hvilket ofte er hurtigere. Vælg menupunkt **File->Import/Export** for at få en tekstboks, hvor man kan få udskrevet (export) og indlæst (import) indhold i hukommelsescellerne. Formatet er simpelt: hver linie i tekstboksen svarer til en (dobbel) hukommelsescelle, så skriv fire hexadecimalle cifre per linie. Tomme linier giver tomme celler. Man kan flytte programmer mellem denne tekstboks og `.txt`-filer åbnet i en editor ved hjælp af copy-paste (Ctrl-c og Ctrl-v), og man kan dermed gemme programmer og bruge dem igen.

Brug menupunkt **Help** for f.eks. af få liste over alle kommandoer i maskinsproget. For at køre et program igen skal man reset'e registre og program counter via knappen nederst til venstre. For at slette et helt program, vælg menupunkt **File->Clear**.

RAM cellerne indeholder både programmer og data. Læg gerne programmerne først (startende i celle 00) og data senere (gerne adskilt fra programmet af et par tomme celler).

Lav følgende opgaver. Arbejd meget gerne i par.

1. På kursets webside er der to `.txt`-filer med eksempelprogrammer. Download disse, indlæs dem i simulatoren ved hjælp af copy-paste metoden ovenfor, og kør dem.
Forstå deres virkemåde ved at læse kommentarerne til dem på slides om CPU.
Følg også med i listen over instruktionssættet (enten i slides eller under **Help** i simulatoren).
2. Lav et program som læser to heltal fra RAM cellerne 10 og 12, finder deres sum, og skriver resultatet i RAM celle 14. [Hint: det er en let forandring af det første eksempelprogram.]
3. Lav et program som læser et heltal k fra RAM celle 18 og som skriver summen $1 + 2 + 3 + \dots + (k - 1)$ i RAM celle E1. [Hint: det er en let forandring af det andet eksempelprogram.] Da man med 8 bits heltal i two's complement kun kan repræsentere heltal op til 127, skal vi have $k \leq 16$ for at kunne repræsentere resultatet.
4. Lav et program som læser et heltal k fra RAM celle 20 og som skriver bitmønsteret 11111111 (hexadecimalt: FF) i RAM celle 22 hvis k er

forskellig fra 0, og skriver bitmønsteret 01010101 (hexadecimalt: 55) i RAM celle 22 hvis k er lig 0.

5. Lav et program som læser et bitmønster fra RAM celle 10, laver de første fire bits om til 0'er, og skriver svaret i RAM celle 12. Hint: det kan gøres med bit-wise AND med et bestemt bitmønster (hvilket?).
6. Lav et program som læser to bitmønster x og y fra RAM cellerne 20 og 22, laver et nyt bitmønster, som består af de første fire bits fra x efterfulgt af de sidste fire bits fra y , og skriver svaret i RAM celle 22. Hint: brug ideen fra sidste opgave to gange, samt bit-wise OR.
7. Lav et program som læser to bitmønster x og y fra RAM cellerne 20 og 22, laver et nyt bitmønster, som består af de sidste fire bits fra y efterfulgt af de første fire bits fra x , og skriver svaret i RAM celle 22. Hint: brug ideen fra sidste opgave, samt cyklisk rotation af bits.
8. Følgende metode skifter fortegn på heltal repræsenteret i two's complement: inverter alle bits i tallet og læg derefter 1 til (man kan vise at dette gør det samme som metoden angivet på slides om repræsentation af tal). Implementer et program, som læser et tal x fra RAM celle 20 og skriver tallet $-x$ i celle 22. Hint: bits i x kan inverteres ved bitwise XOR af x med et bestemt bitmønster (hvilket?).