

## Lab DM534 Uge 38

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

### Uge 37

Disse lab-timer handler om programmering i maskinsprog. Vi skal bruge en webside, som simulerer og visualiserer en simpel CPU. CPU'ens struktur og kommandosæt er beskrevet i slides om CPUer.

Linket til simulatoren findes på kursets webside. I simulatoren vises et vindue med en CPU med registre og program counter til venstre og en række hukommelsesceller (RAM) til højre. 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 Play) for at udføre det indtastede program. Både bitmønstre i celler og numre på celler og registre angives med hexadecimal notation (se slides om repræsentation af tal). Bemærk: hukommelsescellerne er opstillet parvis, fordi en kommando fylder to ord (dvs.  $2 \times 8 \text{ bit} = 2 \times 2$  hexadecimale cifre).

Man kan også skrive programmer linie for linie i en tekstboks, hvilket ofte er hurtigere. Vælg 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 (dobbelt) hukommelsescelle, så skriv fire hexadecimale cifre pr. 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. Bemærk at websiden gemmer dit aktuelle program, så du kan lukke siden og åbne den igen uden tab.

Brug `Help` til f.eks. af få liste over alle kommandoer i maskinsproget. For at køre et program igen skal man bruge `Reset CPU` (som nulstiller registre og program counter). For at slette et helt program skal man bruge `Reset RAM`.

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. Hvad er den hexadecimal notation for kommandoerne til at gøre følgende (numre på registre og RAM celler er nedenfor angivet hexadecimalt):

- Kopiere indholdet af register C til RAM celle 0A.
- Lægge bitmønstret 10110011 ind i register 2.
- Addere register 3 og 4, og lægge resultatet i register 5.
- Lave bit-wise XOR af register B og C.
- Hoppe til instruktionen i RAM celle 14 hvis indholdet i register C er større ( $>$ ) end indholdet i register 0.

2. 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).

3. Forklar hvad følgende program gør:

```
1110
1212
5112
1214
5112
3118
C000
```

4. 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.]
5. 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.)
6. Lav et program som læser to heltal fra RAM celle 16 og 18, og som skriver det største af dem i celle 14.
7. 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.
8. 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?).
9. 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.
10. 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.