

Bits

DM534

Rolf Fagerberg, 2012

Resume af sidst

- ▶ Overblik over kursus
- ▶ Introduktion. Tre pointer:
 - ▶ Datalogi er **menneskeskabt** og dynamisk. Tidslinie over fremskridt mht. ideer og hardware.
 - ▶ **Algoritme** er et centralt begreb i datalogi.
 - ▶ **Abstraktion** er et stærkt værktøj.

Algoritmer

Algoritme = sekvens af operationer på en virtuel maskine.

Virtuel maskine = Samling af veldefinerede operationer (handlinger).

Algoritmer

Algoritme = sekvens af operationer på en virtuel maskine.

Virtuel maskine = Samling af veldefinerede operationer (handlinger).

Eksempler:

- ▶ Rutevejledning.
- ▶ Madopskrift.
- ▶ Strikkeopskrift.
- ▶ Noder.
- ▶ Origami-instruktioner.
- ▶ Samlevejledning for IKEA-møbler.
- ▶ Computerprogram.

Algoritmer

En algoritme skal beskrive en løsningsmetode. Med til formålet tilpas præcision.

Algoritmer

En algoritme skal beskrive en løsningsmetode. Med til formålet tilpas præcision.

Program når det skal læses af en computer.

Andre metoder når mennesker skal læse. Udelad detaljer, som kan tilføjes uden problemer senere (en vurderingssag), beskriv essensen i løsningsmetoden.

Algoritmer

En algoritme skal beskrive en løsningsmetode. Med til formålet tilpas præcision.

Program når det skal læses af en computer.

Andre metoder når mennesker skal læse. Udelad detaljer, som kan tilføjes uden problemer senere (en vurderingssag), beskriv essensen i løsningsmetoden.

Een mulighed: **pseudokode**.

Pseudokode = grundelementerne fra programmeringssprog (while, if-then-else, for, tildeling, . . .), minus streng grammatik, plus fri tekst.

Algoritmer

Eksempel: korttrick.

Algoritmer

Eksempel: korttrick.

Eksempel: Euklids algoritme for største fælles divisor af to heltal M og N .

Algoritmer

Eksempel: korttrick.

Eksempel: Euklids algoritme for største fælles divisor af to heltal M og N .

M = største tal

N = mindste tal

while N ikke går op i M :

R = rest ved division af M med N .

$M = N$

$N = R$

returner N

Algoritmer

Eksempel: korttrick.

Eksempel: Euklids algoritme for største fælles divisor af to heltal M og N .

M = største tal

N = mindste tal

while N ikke går op i M :

R = rest ved division af M med N .

$M = N$

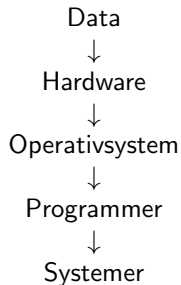
$N = R$

returner N

- ▶ Stopper?
- ▶ Hvor hurtigt?
- ▶ Svarer korrekt?

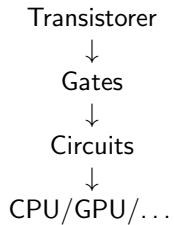
Abstraktion

Eksempel: Byg større ting af mindre ting.



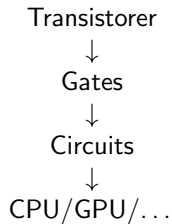
Abstraktion

Eksempel: Byg større ting af mindre ting.



Abstraktion

Eksempel: Byg større ting af mindre ting.



De underliggende tings virkemåde behøves ikke kendes, kun deres ydre virkemåde.

Abstraktion

De underliggende tings virkemåde behøves ikke kendes, kun deres ydre virkemåde.

Abstraktion

De underliggende tings virkemåde behøves ikke kendes, kun deres ydre virkemåde.

Interface, modularisering, modellering giver:

- ▶ Struktur på konstruktionen.
- ▶ Uafhængighed mellem de enkelte dele (disse kan re-implementeres uden at resten skal ændres).
- ▶ Analyserbarhed.
- ▶ Øget innovation/fremskridt/produktivitet (“stå på skuldrene af giganter”).

Abstraktion

De underliggende tings virkemåde behøves ikke kendes, kun deres ydre virkemåde.

Interface, modularisering, modellering giver:

- ▶ Struktur på konstruktionen.
- ▶ Uafhængighed mellem de enkelte dele (disse kan re-implementeres uden at resten skal ændres).
- ▶ Analyserbarhed.
- ▶ Øget innovation/fremskridt/produktivitet (“stå på skuldrene af giganter”).

Gennemgående tema i datalogi

Abstraktion

Modeller tillader at fokusere på det væsentlige.

Abstraktion

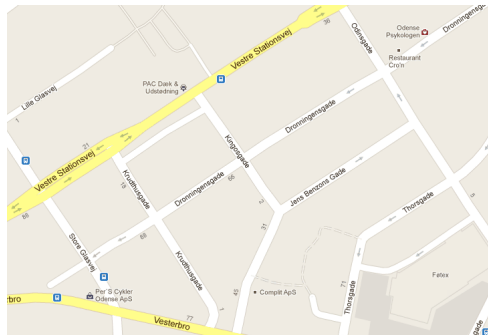
Modeller tillader at fokusere på det væsentlige.

Eksempel: Vejkort og repræsentation som graf.

Abstraktion

Modeller tillader at fokusere på det væsentlige.

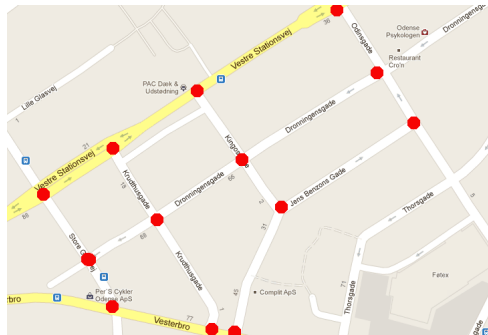
Eksempel: Vejkort og repræsentation som graf.



Abstraktion

Modeller tillader at fokusere på det væsentlige.

Eksempel: Vejkort og repræsentation som graf.



Abstraktion

Modeller tillader at fokusere på det væsentlige.

Eksempel: Vejkort og repræsentation som graf.



Bits

Bit = 0 eller 1.

Fundamental dataenhed.

Samles ofte i større grupper. Eg. 8 bits = 1 byte.

Behov for at **manipulere/ændre** og for at **gemme**.

Manipulere bits

AND



Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

OR



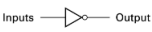
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

XOR



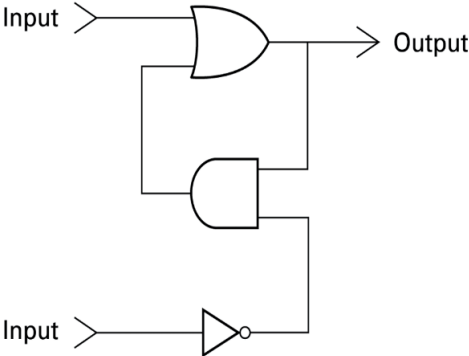
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

NOT



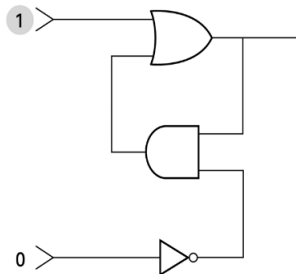
Inputs	Output
0	1
1	0

Gemme bits



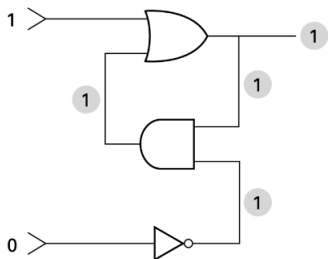
Gemme bits

a. 1 is placed on the upper input.



Gemme bits

- b. This causes the output of the OR gate to be 1 and, in turn, the output of the AND gate to be 1.



Gemme bits

c. The 1 from the AND gate keeps the OR gate from changing after the upper input returns to 0.

