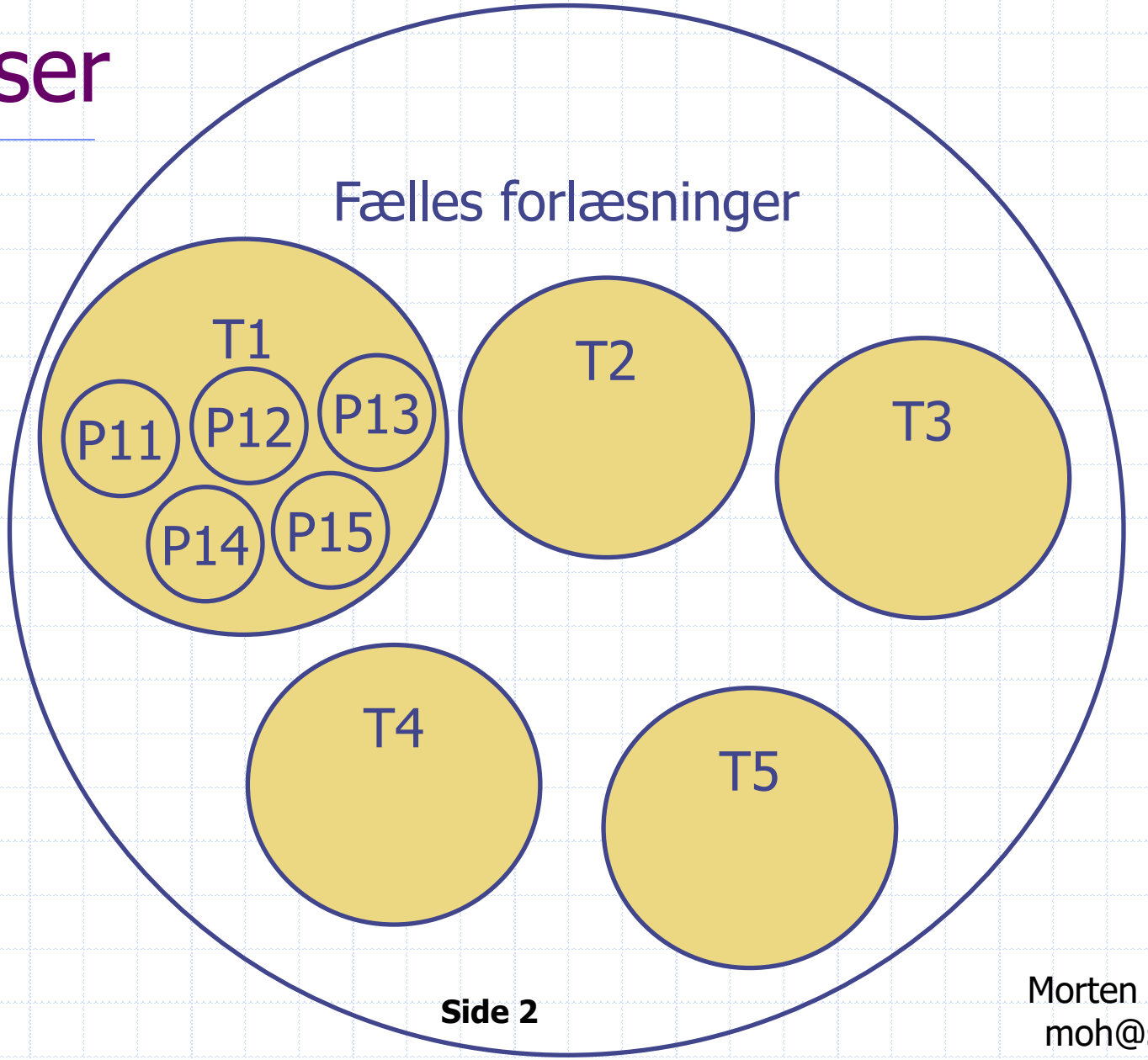
Velkommen til COS Computersystemer

Lektion #1

Morten Hansen
moh@sdu.dk

Klasser



Morten Hansen
moh@sdu.dk

Undervisere

- Morten Hansen, moh@mmmi.sdu.dk

- Der er 6 instruktorer til kurset.
- Hver instruktor er tilknyttet en klasse.

De studerende

- Hvem er I?
- Er I her?

Morten Hansen
moh@sdu.dk

Computersystemer:

Indhold

Fagligheden giver den studerende en kortfattet introduktion til computerarkitektur, operativsystemer, netværk, algoritmer og programmerings paradigmer.

Fagligheden omfatter datalagring, computerarkitektur, maskinsprog, operativsystemer, koordinering af aktiviteter på computeren, processer, netværk, protokoller, Internettet, sikkerhed og algoritmeteori.

Morten Hansen
moh@sdu.dk

Viden

Ved faglighedens afslutning forventes den studerende at kunne:

- Beskrive hvordan data lagres på en computer
- Beskrive hvordan en computer manipulerer data og kommunikerer med perifere enheder og herunder beskrive grundlæggende elementer i maskinarkitektur
- Beskrive hvad et operativsystem er og hvordan operativsystemet koordinerer computerens aktiviteter
- Beskrive hvordan computere kan knyttes sammen i netværk og herved dele informationer og resurser
- Beskrive simple algoritmer

Morten Hansen
moh@sdu.dk

Færdigheder:

Ved faglighedens afslutning forventes den studerende at kunne:

- Konvertere tal fra decimal til binær eller floating point repræsentation, og omvendt
- Lave simple logiske kredsløb
- Programmere i et simplificeret maskinsprog
- Redegøre for situationer hvor deadlock og race condition kan opstå
- Udføre simpel kommunikation mellem forskellige processer over socket forbindelser (TCP og UDP)
- Designe simple algoritmer vha. pseudokode

Morten Hansen
moh@sdu.dk

Kompetencer:

Ved faglighedens afslutning forventes den studerende at kunne:

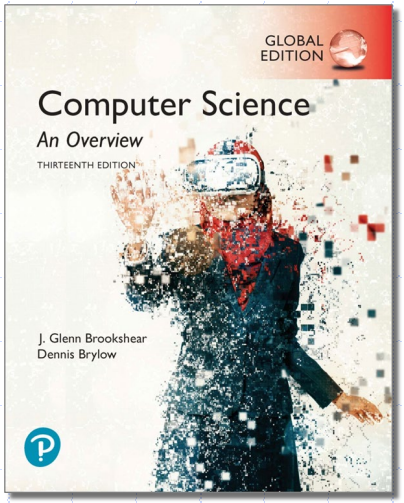
- Eksemplificere dele af ovenstående viden og færdigheder vha. et programmeringssprog

Morten Hansen
moh@sdu.dk

Materialer:

Litteratur:

- Brookshear & Brylow: *Computer Science, An Overview*
- *Diverse noter og links*



Ugens cyklus:

- Fredag 10:15-12:00
  - Forelæsninger
- Fredag 12:15-14:00
  - Øvelser
- Hjemme arbejde
  - Færdiggør øvelser
  - Læs litteratur til næste forelæsning (se BB/Assignments)

Morten Hansen
moh@sdu.dk

Lektionsplan & Lektionsbeskrivelser

Se: itslearning/Resources

Morten Hansen
moh@sdu.dk

Computersystemer
Computer Science

Computer Science er læren om algoritmer.

- **Algorithm:** A set of steps that defines how a task is performed
- **Program:** A representation of an algorithm
- **Programming:** The process of developing a program
- **Software:** Programs and the algorithms they represent
- **Hardware:** The machinery

Morten Hansen
moh@sdu.dk

Figure 0.1
An algorithm for a magic trick

Effect: The performer places some cards from a normal deck of playing cards face down on a table and mixes them thoroughly while spreading them out on the table. Then, as the audience requests either red or black cards, the performer turns over cards of the requested color.

Secret and Patter:

Step 1. From a normal deck of cards, select ten red cards and ten black cards. Deal these cards face up in two piles on the table according to color.

Step 2. Announce that you have selected some red cards and some black cards.

Step 3. Pick up the red cards. Under the pretense of aligning them into a small deck, hold them face down in your left hand and, with the thumb and first finger of your right hand, pull back on each end of the deck so that each card is given a slightly *backward* curve. Then place the deck of red cards face down on the table as you say, "Here are the red cards in this stack."

Step 4. Pick up the black cards. In a manner similar to that in step 3, give these cards a slight *forward* curve. Then return these cards to the table in a face-down deck as you say, "And here are the black cards in this stack."

Step 5. Immediately after returning the black cards to the table, use both hands to mix the red and black cards (still face down) as you spread them out on the tabletop. Explain that you are thoroughly mixing the cards.

Step 6. As long as there are face-down cards on the table, repeatedly execute the following steps:

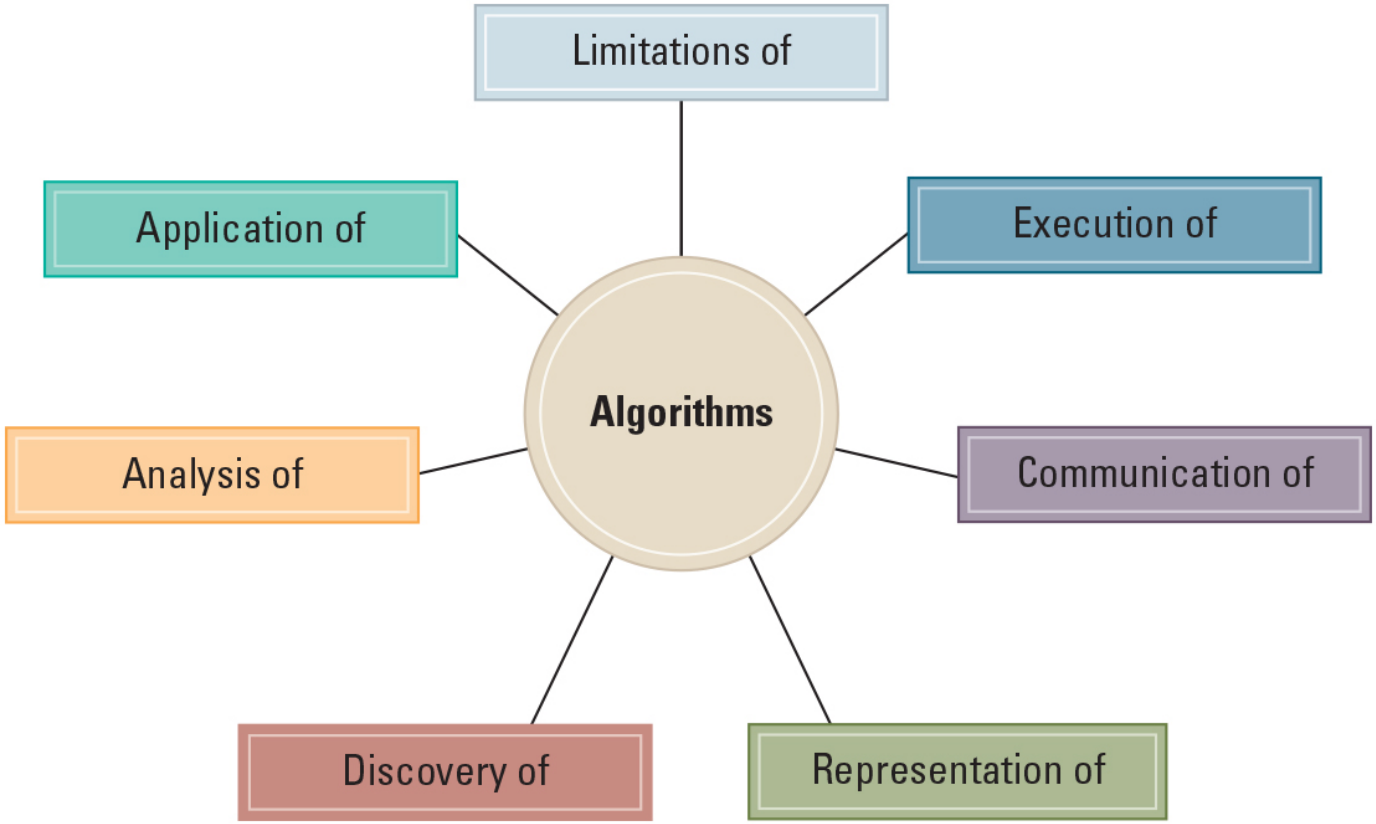
6.1. Ask the audience to request either a red or a black card.

6.2. If the color requested is red and there is a face-down card with a concave appearance, turn over such a card while saying, "Here is a red card."

6.3. If the color requested is black and there is a face-down card with a convex appearance, turn over such a card while saying, "Here is a black card."

6.4. Otherwise, state that there are no more cards of the requested color and turn over the remaining cards to prove your claim.

Figure 0.5
The central role of algorithms in computer science



The seven "Big Ideas" of computer science:

- Algorithms,
- Abstraction,
- Creativity,
- Data,
- Programing,
- Internet
- Impact

Morten Hansen
moh@sdu.dk

Bits

- **Bit:** Binary Digit (0 or 1)
- Bit Patterns are used to represent information
  - Numbers
  - Text characters
  - Images
  - Sound
  - And others

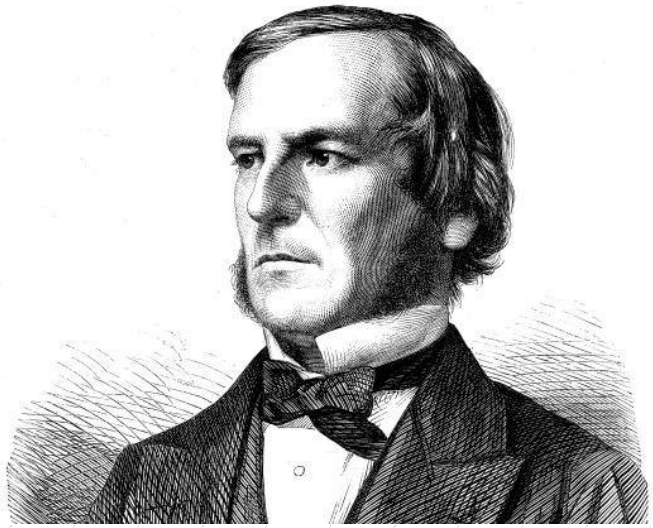
Morten Hansen
moh@sdu.dk

George Boole (1815-1864).

Boolsk algebra.

- De 4 postulater (Aksiomer)
- Afledte formler (Algebra)

Se note på BB.



Morten Hansen
moh@sdu.dk

1. Aksiom

- En variabel: X kan antage én og kun én af to værdier:
- X = 1, hvis X ≠ 0. X = 0, hvis X ≠1.

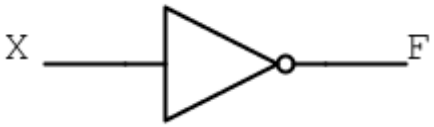
X
0
1

Morten Hansen
moh@sdu.dk

2. Aksiom

- ”Ikke” funktionen. Negationen. (NOT).
- $F = \overline{X}$

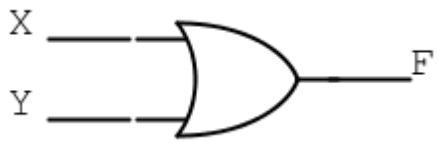
X	F
0	1
1	0



Morten Hansen
moh@sdu.dk

3. Aksiom

- ”Eller” funktionen. (OR).
- $F = X + Y$

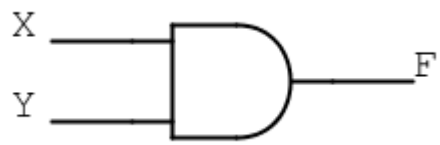


X	Y	F
0	0	0
0	1	1
1	0	1
1	1	1

Morten Hansen
moh@sdu.dk

4. Aksiom

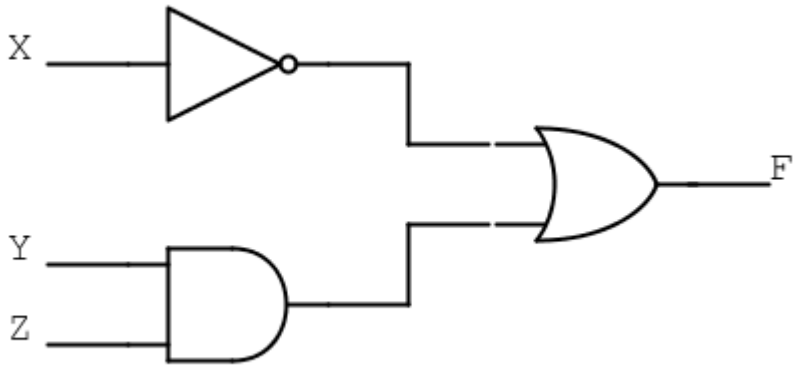
- ”Og” funktionen. (AND).
- $F = X \cdot Y$



X	Y	F
0	0	0
0	1	0
1	0	0
1	1	1

Morten Hansen
moh@sdu.dk

Et logisk kredsløb

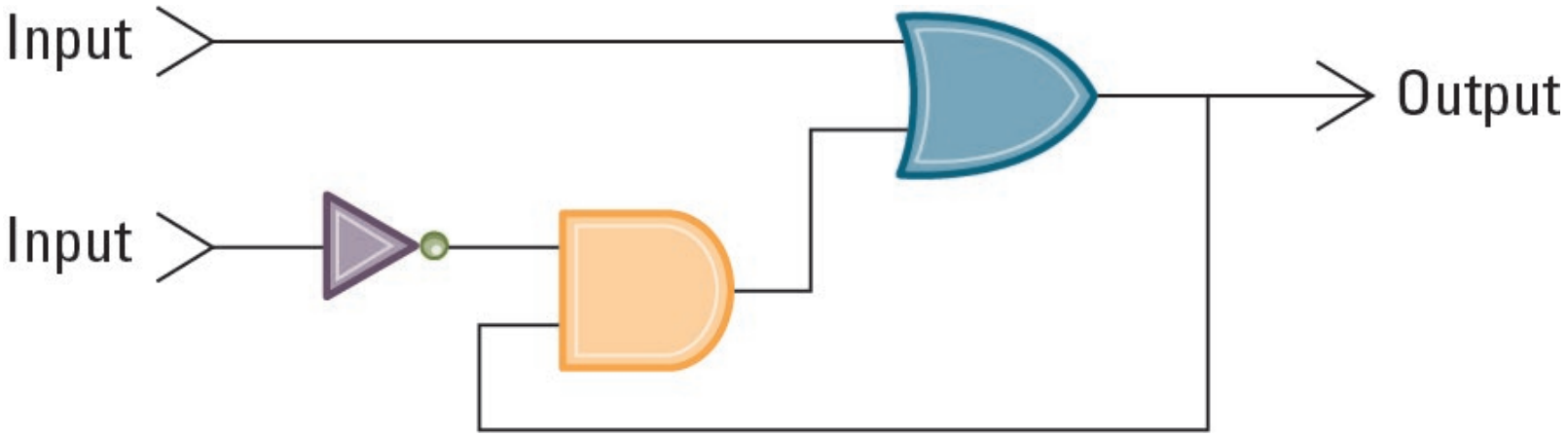


$$F = \overline{X} + Y \cdot Z$$

X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

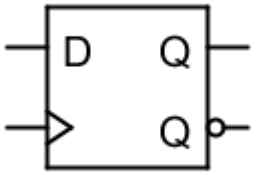
Morten Hansen
moh@sdu.dk

Figure 1.3 A simple flip-flop circuit

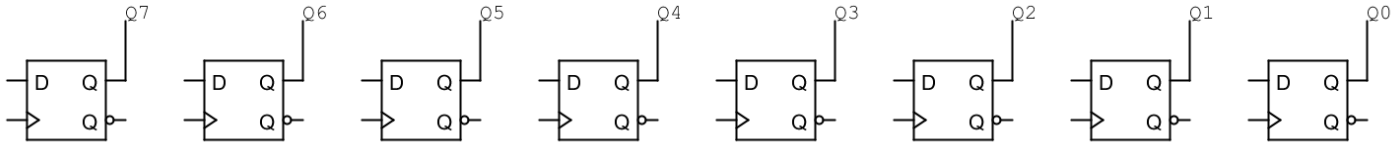


D-flip-flop

En abstraktion

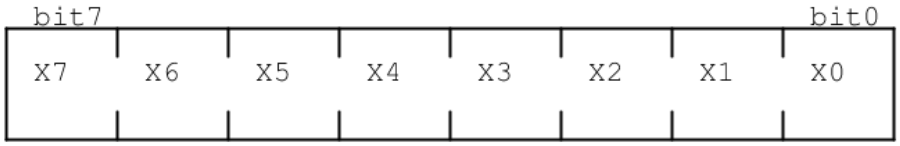


8 flip-flop's – 8 bit



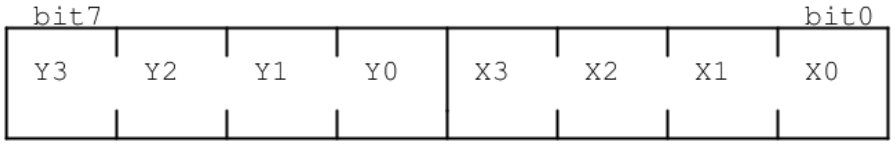
Morten Hansen
moh@sdu.dk

Byte



Eksempel:
01101010_2

2 nibbles



0110_2 1010_2

Hexadecimal notation

$6A_{16}$ = 0x6A

Morten Hansen
moh@sdu.dk

Hexadecimal Notation

- **Hexadecimal notation:** A shorthand notation for long bit patterns
  - Divides a pattern into groups of four bits each
  - Represents each group by a single symbol
- Example: 10110101 becomes 0xB5

Hexadecimale cifre

Bit				Hex
3	2	1	0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Morten Hansen
moh@sdu.dk

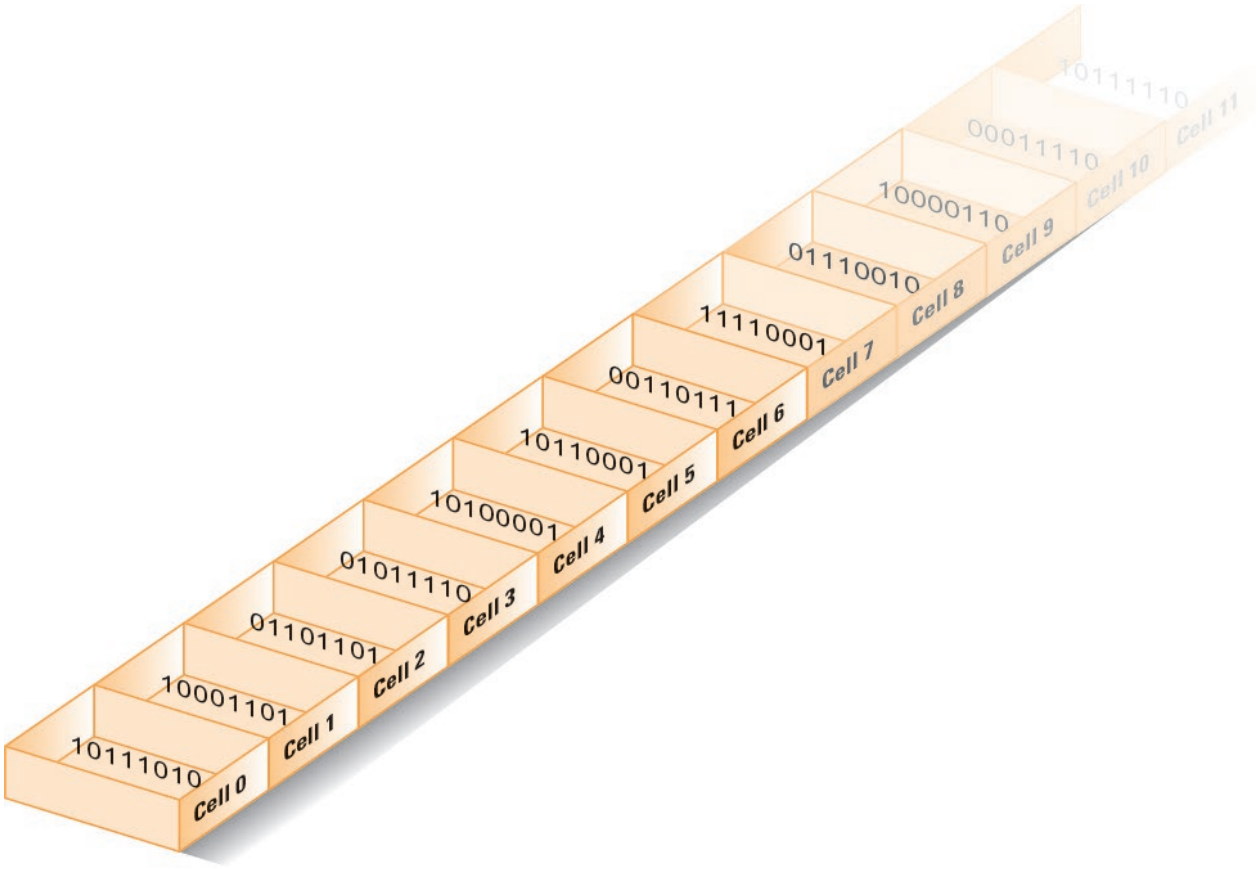
1.2 Main Memory

- **Cell:** A unit of main memory (typically 8 bits which is one **byte**)
  - **Most significant bit:** the bit at the left (high-order) end
  - **Least significant bit:** the bit at the right (low-order) end

Main Memory Addresses

- **Address:** A “name” that uniquely identifies one cell in the computer’s main memory
  - The names are actually numbers.
  - These numbers are assigned consecutively starting at zero.
  - Numbering the cells in this manner associates an order with the memory cells.

Figure 1.8 Memory cells arranged by address



Instruktorer

- Der er 6 instruktorer til kurset. Hver instruktor er tilknyttet en klasse:

- Klasse T1 i U165: Mads Munch Christensen, madch19@student.sdu.dk
- Klasse T2 i U166: Nichlas Daniel Boraso, nibor19@student.sdu.dk
- Klasse T3 i U167: Jonas Ahwazian, joahw18@student.sdu.dk
- Klasse T4 i U171: Mads Hundevad Jensen, madsj18@student.sdu.dk
  Mikkel Daugaard Høyberg, mihoe18@student.sdu.dk
- Klasse T5 i U172:Frederik Alexander Hounsvad, frhou18@student.sdu.dk

Instruktorerne vil være til stede og være behjælpelige med løsning af opgaver fredage kl.: 12:15-14:00 i klassens lokale.

Morten Hansen
moh@sdu.dk