# M2. Data and program representations

How are data and programs represented? Keywords: binary numbers, opcode, operands

Binary encoding of numbers

* Signed and unsigned integers encoding.
  + Sign-magnitude (gem sign som separat bit)
  + One's complement (bits bliver reversed for at markere et negativt tal)
  + Two's complement (start med et positivt tal, træk en fra og reverse bitsne)
    - Smart because the lowest binary number is also the lowest decimal
    - You can easily calculate what -2 + 4 is since you just increase -2 in binary with 4.
* Range limits.
  + Bytes kan kun holde tal op til (2^k)-1 hvor k er antallet af bits i en byte.
* Hexadecimal notation.
  + 0x er hex, 0b er binær
  + 16, er deleligt med to gør konversion til binær nem
* Precision limits, number density.
  + Precision depends on number size. 64 bit floats are more precise than 32 bits floats.

**Overflow & underflow**

**Overflow** is when a result requires more bits than allocated.  
**Underflow** is when there is being read faster from the buffer than there is being written to it.

Instruction set is the set of intructions understood by a processor.  
Instruction format is the binary representation of the instructions understood by the processor.

Opcode is the numeric value assigned to an instruction  
Operands are the values taken by the instruction  
Result refers to the place for the result to be placed (not used by all processors)