The material the quiz will cover is converting between decimal (base 10), binary (base 2) and hexadecimal (base 16) representation using two's complement signed values as well as addition/subtraction with signed hexadecimal values indicating whether signed overflow occurs.

# Binary Numbers

Binary Representation and Logic

* Bit = Binary Digit: 0/1, Off / On
* Mathematical Diversion:
  + Assume n-state device where cost of device is linear
    - N = KN
  + Number of devices required for representing an arbitrary number ‘x’ is logn(x)
  + Thus, total cost to represent x = kn \* logn(x)
    - Find n that minimuzes cost
    - N = e = 2.718
  + N bits have 2n permutations

Memory unit terminology

* Word
  + Unit of memory access and/or size of registers
    - 16 / 32 / 64 bits
* Byte
  + Unit for character representation
    - 8 bits
* Nibble
  + Unit for binary-coded decimal digit
    - BCD
    - 4 bits

Terminology for bases:

* Decimal (Base 10, 0-9)
* Binary (Base 2, 0-1)
* Octal (Base 8, 0-7)
* Hexadecimal (Base 16, 0-F)

Conversions

* Decimal -> Binary
  + While(number > 0)
    1. Divide number by 2, getting quotient and remainder
    2. Remainder is next binary digit
    3. Number = quotient
* Binary -> Decimal
  + Positional representation
    1. 10110111 =
    2. 1\*27 + 0\*26 + 1\*25 + 1\*24 + 0\*23 + 1\*22 + 1\*21 + 1\*20 =
    3. 128 + 0 + 32 + 16 + 0 + 4 + 2 + 1 =
    4. 128 + 32 + 16 + 4 + 2 + 1 =
    5. 183 = 10110111
* Binary -> hexadecimal
  + Using a table
    - 1001 0010 1110 0001 1010
    - 9 2 E (14) 1 A (10)
  + Binary -> octal is same, just use groups of 3

Signed binary representation

* Sign-magnitude
  + Put a 0 for positive, 1 for negative at very left of byte
  + Computationally awkward
* Ones-complement
  + If negative, flip all values
* Twos-complement
  + If negative, flip all values and add 1

# Arithmetic