# Eksamen 2021

Af Jesper Bertelsen

## Theoretical Questions

### Question 1. As we go further from zero, how does the distribution of representable numbers change when using the IEEE floating point standard? Why? (4 points)

IEEE floating point standard.

Cases:

1. Normalized values ( exponent bits ≠ det mindste eller det største som den kan være ):
2. Denormalized values ( exponent bits = 0 ):
3. Infinity ( exponent bits = største værdi muligt, frac = 0 ): Ingen reelt tal at få ud af det
4. NAN ( exponent bits = største værdi muligt, fract ≠ 0): Ikke et reelt tal.

Normalized exp range for single precision:

Values =

Eksempel:

0 0000 010 (Denormalized) =>

0 0010 010 ( Normalized) =>

0 0111 010 ( Normalized) =>

As we go further away from away from zero, the case of the IEEE floating point standard comes closer and closer to the infinity case. This will happen if the exponent bit comes closer to being all ones, whilst the fractional bits comes closer to being all zeros.   
This can result in overflows.

Why the must be zero, I don’t quiet understand. To me it seems as, with the

### Question 2. What is the “word size” of a computer? What are the typical values? (3 points)

The word size of a computer indicates how large the transfer size can be.   
In early days of computers, systems would be run with word sizes of 32 bits, said as a 32bit operating system.

Know a days modern computers run with 64 bits.  
Hardware for other applications might run with 32 bits or less.

Microcontrollers doesn’t have that large of word sizes, I think they are usually about 8bits to 16 bits.

### Question 3. How do compilers speed up multiplications? (3 points)

Instead of multiplying, which is an expensive operation, compilers left shift bits. Left shifting bits is easy, when some value is multiplied with something that can be translated into power of 2. Ex.

This is written as resulting in: , adding 3 zeroes at the end.

With numbers not able to be directly translated into power two, addition can be used for a combination of shifts.

Depended on how many combinations the compiler has to do, it might use two’s compliment for multiplying instead, as these operations can add up.

### Question 4. What is the Arithmetic/Logic unit? Does it have memory elements? (3 points)

The Arithmetic / Logic Unit is also known as the ALU an is a part of the CPU and serves the purpose as to perform Add, Subtract, And or Xor operations for the CPU, working as a multiplexer taking in two inputs as values and a selector to select between the 4 operations.

The ALU does NOT have a memory unit built withing it, but rather writes/overwrites it to the CPU’s register.

### Question 5. What are the two types of hazards when talking pipelining? why do they happen? (4 points)

The two types of hazards when talking about pipelining is:

1. Data hazards, which occur when a value required in cycle *x*, has not yet been updated until cycle *x + 1 or more*. The operations made in cycle *x* then uses outdated data, which might give unexpected results.
2. Control hazards, which I am not sure about, but it has something to do with jumps. Say stage 3 of the processor returns a condition resulting in a jump to stage 6. Then the newly started stage 2 and stage 1 processors will have wasted resources, as these should just have started in stage 6 instead, as the jump happened. It’s something like that.

### Question 6. When a HDD needs to read an address, what takes up most of the time and what is relatively inexpensive? (4 points)

With a HDD one thing concerning the speed is it’s physical capacity, meant in the way, that it’s spinner reading the data, can only speed so fast, without breaking of making failures. If the data needed is in a sector, that has just recently been passed by the reader, the fetch must then wait until the spinner has cycled the plate a whole round. I think it’s about 3ms for one spin or so.

It’s inexpensive as to other much higher speed storage technologies as with these. Platters can just be stacked on top of each other, making room for multiple layers of data. Higher speed storage technologies takes advantages of other technologies, which are more advanced and complicated, thus making it more expensive.

Question 6.