CSE1400 - Computer Organisation

Self-Study: Week 4 - Instruction Set Architecture

Delft University of Technology

2025/2026 Q1

Special thanks to Sára Juhošová, Ana Băltărețu, Mara Coman, Alexandra Marcu, Alexandru Postu,
Kiril Vasilev and Rareș Toader
for helping with the compilation of this set of questions.

Important information:

1. If any question is unclear please consult Answers EWI. For more specific questions you can use the Queue during lab hours.
2. The average time for solving this self study is **3** hours, and **1** hour is allocated to giving feedback. Timings are included for each exercise to give you a more clear overview of how much time you should be spending on them.
3. The maximum amount of points for this self study is 100 points. To get the points you should submit a serious attempt on Peer and **properly review** your peers' submissions (50 points per full cycle, including review evaluation).
4. **Make sure that your submission is anonymous so please do not share personal information such as name or student number in the solution, review, or file name of your attachment on Peer!**
5. Answers will be provided during the weekly tutorial sessions.

Show your work for all assignments. Make sure to highlight your final answer.

1. (3 mins) Name the components of the following Von Neumann Architecture:

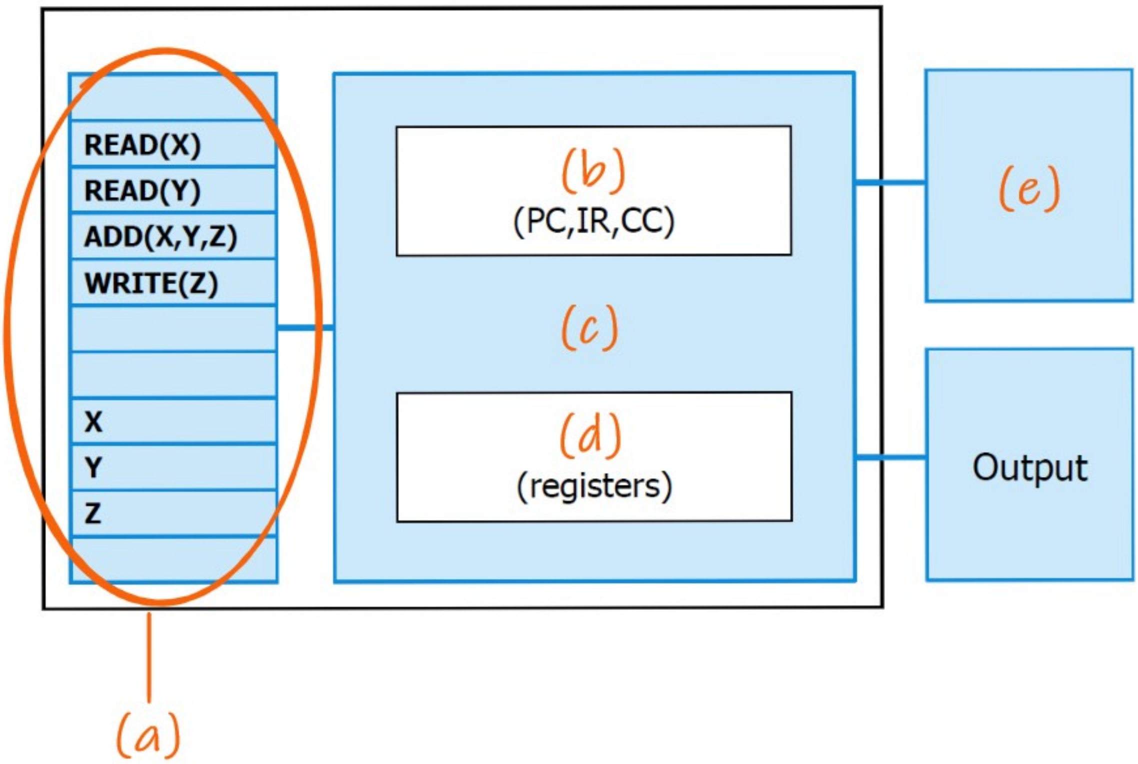

Figure 1: Von Neumann Architecture

(a) Memory

(b) Control unit

(c) CPU

(d) Logic unit

(e) Input

2. (8 mins) Consider a 64 MiB main memory. How many bits are required to be able to individually address each of the following:

(a) every bit

64MiB is 67 108 864 bytes, that times 8 is the amount of bits.
log(67108864)/log(2) (base-2 log also works) tells us we need 26 bits.
Since we need 8 times that and every bit we add gives us twice the values: 27 gives us X2, 28 gives x4, 29 gives x8, so **29 bits** (or just 26+3 because 8 is 2^3)

(b) every byte

Same as A, 64Mib is 67108864 bytes, base-2 log(64Mib) gives 26 so we need **26 bits**

(c) every 64-bit quadword

In a similair vein to A we can say we need 1/8 as many adresses as byte-adressable, that means 3 bits less than the awnser to B which was 26 so **24 bits**

3. (3 mins) How many bits are required to address 73 registers?

log(73)/log(2) gives 6.1898… so we need 7 to be able to represent the number 73. If 73 was a power of 2 we would subtract 1 bit because 0 is also a value so when the highest number you can represent is say 63 you have 64 values. But because 73 is not a power of 2 this does not hold. You can verify with 2^6 which is 64 and 2^7 with is 128. therefore **7 bits**

4. (3 mins) How many different instructions can an ISA have if its opcode size is 8 bits?

2^8 = 256 so **256 instructions**

5. (15 mins) Consider an ISA with 128 possible instructions, 32 registers and 64-bit instructions. Each instruction consists of an opcode, one register operand and two direct memory access operands. How much memory (in MiB) can it address if the memory is of the following type:

(a) bit addressable memory

Because 2^7 = 128 we need 7 bits for the opcode
Because 2^5 = 32 we need 5 bits for the register
We had 64 bits, we ‘spent’ 12 of those so we have 52 left for 2 equally sized memory adresses
So 26 bits per memory address. Which gives 67 108 864 separate addresses, divide by 8 to get the amount of bytes (67108864/8 = 8388608), divide that by 1024 to get the about of KiB (8388608/1024 = 8192), divide that by 1024 for the amount of Mib (((2^26)/8)/1024)/1024 = **8 MiB**

(b) byte addressable memory

8 bits per byte so just the awnser to A times 8 (8*8 = 64MiB) or the same final calculation without dividing by 8: ((2^26)/1024)/1024 = **64MiB**

(c) 32-bit word addressable memory

Take the awnser of A, multiply by 32 → 8*32 = **256MiB**

6. (4 mins) Consider an ISA with 32 registers and instructions which are each one word long and the word size is 16 bits.Each instruction has an opcode and two register operands.How many bits does the opcode have?

32 regs means log(32)/log(2) = 5 so 5 bits to represent 32 values.
Instructions are 16 bits long, 2 registers and 1 opcode
16-(2*5)=6, **the opcode has 6 bits**

7. (4 mins) Consider an ISA that has 64 registers and 15 different instructions which are all composed of an opcode and two register operands.How many bits are used for each instruction?

64 regs means log(64)/log(2) = 6 so 6 bits to represent 64 values
Instructions are 1 opcode and 2 regs of 6 bits
1 opcode need 15 or more unique values so ceil(log(15)/log(2)) = 4 (wasting 1 opcode without associated instruction)
Instructions are 1 opcode (4 bits) and 2 regs (2*6) long so (2*6)+4 = **16 bits**

8. (7 mins) Consider an ISA with 64-bit instructions, 64 registers, and a 32 MiB byte-addressable main memory.42 of these instructions have two direct memory addresses and one register operand. Considering all instructions have the same opcode length, how many other instructions can be created within the same ISA?

6 bits needed to represent 64 regs,
32MiB byte addressable memory is 33 554 432 bytes, log(33554432)/log(2) gives 25 so we need 25 bits for memory addresses.
Weird way of writing the question but i understand they want to know how many diffrent opcodes we can represent and then subtract 42.
Instructions are 64 bits, consisting of 1 opcode, 1 register operand (6 bits), 2 memory operands (2*25 bits)
Opcode length = (64-(1*6))-(2*25) = 8, conclusion: opcodes are 8 bits long, therefore there are 256 different opcodes (2^8), therefore there are 256-42=**214 other instructions**

9. (7 mins) Consider a 64-bit processor with 36 registers and 3 addressing modes (immediate, direct and register). This processor supports 144 instructions and a byte addressable memory of up to 2 TiB. Every instruction has the following operands: two registers, one direct memory access and one immediate value. If all of the instructions are 128 bits long, how many bits does the immediate value operand have?

36 registers need 6 bits to be represented since 36 is between 2^6 (64) and 2^5 (32)
144 insructions need 8 bits to be represented since 144 is between 2^7 (128) and 2^8 (256)
2TiB = 2 199 023 255 552 Bytes, the base 2 log of that is 41 so we need 41 bits per address
Each instruction has 128 bits, 1 opcode (8 bits), 2 registers (2*6 bits), 1 memory address (41 bits) and 1 immediate value (whatever is left)
Question: how much is left when we get to the immediate value.
128-(8+(2*6)+41)=**67 bits**

10. (7 mins) Consider an ISA with 102 bits per instruction and a total number of 256 possible instructions. This ISA supports 42 registers and can access 32 MiB of byte-addressable main memory. Each instruction has an opcode, two register operands, two direct memory access operands and one immediate value. How many bits does the immediate value operand have?

Instructions: log(256)/log(2)=8 so 8 bits per opcode
Registers: ceil(log(42)/log(2))=6 so 6 bits per register
Memory: 32 MiB = 33 554 432 bytes; log(33554432)/log(2)=25 so 25 bits per address
Total=102 bits; 1x opcode (8 bits), 2x register (2*6 bits), 2x memory address (2*25), 1x immediate value (x bits)
X=102-(8+12+50)=**32 bits**

11. (8 mins) Consider having an ISA that takes zero-address instructions that uses implicit reference to the stack for ALU operations. You are asked to write a program that calculates the following expression: $(A \cdot (B + C)) \cdot (A + B) \cdot C$

```
Push a
Push b
Push c
Add
Mul
Push a
Push b
Add
Push c
Mul
mul
```