SAMPLE EXAM PAPERS SOLUTIONS

Damian Gordon

Rough Work

Exam is two hours long

I need to answer three questions

So that's three questions in 120 minutes.

I'll give myself 10 minutes to read the paper at the start, and do calculations on how long things will take

I'll give myself 20 minutes to re-read my answers at the end

So that's 90 minutes to answer 3 questions

That's 30 minutes per question.

Each question is 33 marks, so that's about 55 seconds per mark.

Question 1. Rough Work

Q1(a) = 10 marks

Q1(b) = 6 marks

Q1(c) = 6 marks

Q1(d) = 12 marks

So roughly

Q1(a) = 9 minutes

Q1(b) = 5 minutes

Q1(c) = 5 minutes

Q1(d) = 11 minutes

Timing

Exam starts at 16:00

Reading until 16:10

Q1(a) 16:19

Q1(b) 16:24

Q1(c) 16:29

Q1(d) 16:40

MUST finish Question 1 by 16:40

Question 1. Rough Work

Q. 1(a) Von Neumann architecture – programs and data in same memory, and mention Von Neumann bottleneck

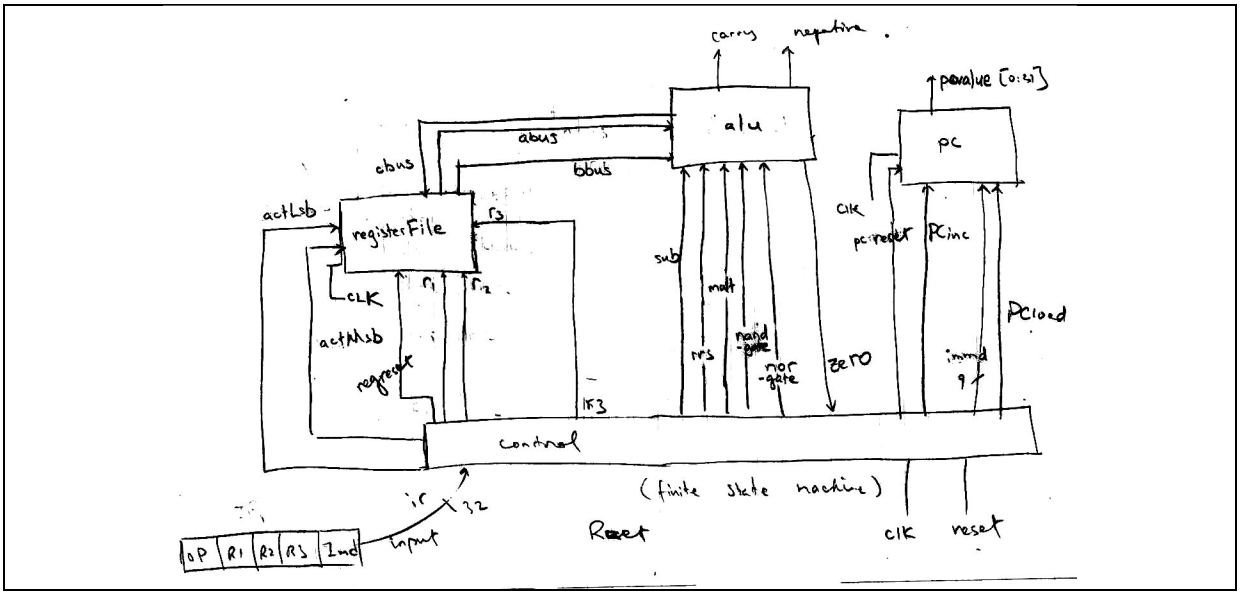
Q. 1(b) Instruction Cycle – Fetch Decode Execute - Fetch Decode INCREMENT Execute LOOP

Q1. (c) Assember Languages - low-level programming language, specific to a particular computer architecture, assembler. Advs. 1.structure 2.easier to understand 3.easier to correct errors 4.same efficiency of execution as machine level.

Q1. (d) LMC Instructions to add two numbers – get first, store, get second, add two together, output. Halt.

Q. 1(a) Von Neumann architecture

The Von Neumann architecture is a computer architecture model that describes a design architecture for an electronic digital computer with parts consisting of a processing unit containing an arithmetic logic unit and processor registers, a control unit containing an instruction register and program counter, a memory to store both data and instructions, external mass storage, and input and output mechanisms.



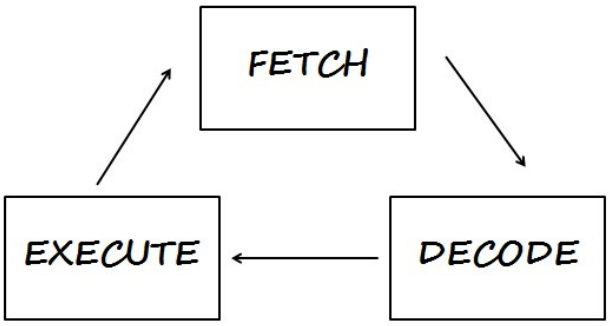
Von Neumann bottleneck

The meaning has evolved to be any stored-program computer in which an instruction fetch and a data operation cannot occur at the same time because they share a common bus. This is referred to as the Von Neumann bottleneck and often limits the performance of the system.

Q. 1(b) Instruction Cycle

The instruction cycle, or Fetch-Decode-Execute (FDE) cycle, is used in Stored-Program Computers since the separation of storage from the processing unit is implicit in this model.

1. **Fetch** the next instruction from memory at the address in the program counter
2. **Decode** the instruction using the control unit
3. Increment the Program Counter
4. The control unit commands the rest of the computer to **execute** the instruction
5. Go to step 1



Q1. (c) Assember Languages

An assember language is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions. Each assembly language is specific to a particular computer architecture, in contrast to most high-level programming languages, which are generally portable across multiple architectures, but require interpreting or compiling. Assembly language is converted into executable machine code by a utility program referred to as an assembler; the conversion process is referred to as assembly, or assembling the code.

Advantages over Machine Code

1. It creates programming structure.
2. It's easier to understand and saves a lot of time and effort of the programmer.
3. It is easier to correct errors and modify program instructions.
4. Assembly Language has the same efficiency of execution as the machine level language. Because this is one-to-one translator between assembly language program and its corresponding machine language program.

CONTINUED, ONCE EVERYONE'S

ASSIGNMENT IS HANDED UP