OCR Computer Science A Level

1.1.1 Structure and Function of the Processor

Concise Notes

Specification:

1.1.1 a)

- The Arithmetic and Logic Unit
- The Control Unit
- Registers:
  - Program Counter
  - Accumulator
  - Memory Address Register
  - Memory Data Register
  - Current Instruction Register
- Buses:
  - Data Bus
  - Address Bus
  - Control Bus
- How these relate to assembly language programs

1.1.1 b)

- The Fetch-Decode-Execute Cycle
- Its effect on registers

1.1.1 c)

- The factors affecting the performance of the CPU:
  - Clock Speed
  - Number of Cores
  - Cache

1.1.1 d)

- The use of pipelining in a processor to improve efficiency

1.1.1 e)

- Von Neumann architecture
- Harvard architecture
- Contemporary processor architecture

Components of a Processor

The Arithmetic and Logic Unit

- The ALU (Arithmetic and Logic Unit) completes the arithmetical and logical operations.

The Control Unit

- The Control Unit is the component of the processor which directs operations inside the CPU. It has the following jobs:
  - Controlling and coordinating the activities of the CPU
  - Managing the flow of data between the CPU and other devices
  - Accepting the next instruction
  - Decoding instructions
  - Storing the result back in memory

Registers

- Registers are small memory cells that operate at very high speeds,
- They are used to temporarily store data,
- All arithmetic, logical or shift operations occur in these registers

Registers	Purpose
Program Counter (PC)	Holds the address of the next instruction to be executed.
Accumulator (ACC)	Stores the results from calculations
Memory Address Register (MAR)	Holds the address of a location that is to be read from or written to.
Memory Data Register (MDR)	Temporarily stores data that has been read or data that needs to be written.
Current Instruction Register (CIR)	Holds the current instruction being executed, divided up into operand and opcode.

Buses

- Buses are a set of parallel wires which connect two or more components inside the CPU together
- The collection of the data bus, control bus, and address bus is called the system bus
- The width of the bus is the number of parallel wires the bus has.

Data Bus

- This is a bi-directional bus (meaning bits can be carried in both directions) used for transporting data and instructions between components.

Address Bus

- Used to transmit the memory addresses specifying where data is to be sent to or retrieved from
- Adding a wire to the address bus doubles the number of addressable locations

Control Bus

- This is a bi-directional bus used to transmit control signals between internal and external components.
- The control signals include:
    - Bus request: shows that a device is requesting the use of the data bus
    - Bus grant: shows that the CPU has granted access to the data bus
    - Memory write: data is written into the addressed location using this bus
    - Memory read: data is read from a specific location to be placed onto the data bus,
    - Interrupt request: shows that a device is requesting access to the CPU
    - Clock: used to synchronise operations

Relationship with assembly language

- Assembly code uses mnemonics to represent instructions.
- It is a simplified way of representing machine code.
- The instruction is divided up into operand and opcode
- Opcode is used to determine the type of instruction and what hardware to use to execute it.
- The operand is the address of where the operation is performed.

Operation Code								Operand							
Machine Code Operation						Addressing mode									
0	1	0	0	0	1	0	1	0	0	1	1	0	1	0	1

Fetch-Decode-Execute Cycle and Registers.

- The fetch-decode-execute cycle is the sequence of operations that are completed in order to execute an instruction.

Fetch Phase:
- Address from the PC is copied to the MAR,
- Instruction held at that address is copied to MDR by the data bus, simultaneously the contents of the PC are increased by 1,
- The value held in the MDR is copied to the CIR.

Decode Phase:
- The contents of CIR are split into operand and opcode

Execute Phase:
- The opcode is executed on the operand.

Factors affecting CPU performance

Clock Speed:
- Clock speed is determined by the system clock
- All processor activities begin on a clock pulse
- Each CPU operation starts as the clock changed from 0 to 1
- The clock speed is the number of clock cycles completed in a second.

Number of Cores:
- A core is an independent processor that is able to execute its own fetch-execute cycle
- A computer with multiple cores can complete more than one fetch-execute cycle at any given time
- Some programs aren't optimised for the use of more than one core.

Amount and type of Cache Memory
- Cache memory is the CPU's onboard memory
- Instructions fetched from main memory are copied to the cache, so if required again it can be accessed quicker.
- As cache fills up, unused instructions get replaced.

Cache Type	Properties

Level 1 Cache	Very fast memory cell, however it also has a small amount of capacity. (2-64KB)
Level 2 Cache	Relatively fast memory cell, with medium sized capacity. (256KB-2MB)
Level 3 Cache	Much larger and slower memory cell.

Pipelining

A-Level only

- Pipelining: The process of completing the fetch, decode, and execute cycles of three separate instructions simultaneously
- Data is held in a buffer in close proximity to the CPU until it's required
- Pipelining is aimed to reduce the amount of the CPU which is kept idle.

Computer Architecture

Von Neumann Architecture:

- Von Neumann includes a single control unit, ALU, registers and memory units
- Shared memory and data bus used for both data and instructions.

Harvard Architecture:

- Physically separate memories for instructions and data
- More commonly used with embedded processors

Advantages of Von Neumann Architecture	Advantages of Harvard Architecture
Cheaper to develop since the control unit is easier to design.	Quicker since data and instructions can be fetched in parallel.
Programs can be optimised in size.	Both memories can be different sizes,

Contemporary Processing

- The combination of Harvard and Von Neumann architecture
- Uses Von Neumann when working with the data and instructions in main memory
- Uses Harvard when working with cache.
- There is an instruction cache and data cache.