This exemplar Student Learning Record was produced by a teacher to give Year 12 students an example of the quality expected when producing assessed work at the end of a topic.

SLR 1 – "Structure and function of the processor"

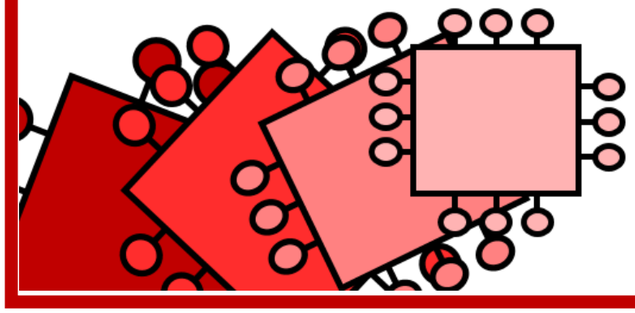
This SLR was graded at an A*.

Remember an A* does not mean the SLR was perfect.

It simply means the student has produced / handed in a piece of work which is of the maximum quality we would expect from a student at this stage of academic study.

Even with A* graded work it is possible to find areas the student can improve or enhance.

NOTE: Students should be encouraged to produce the evidence for their Student Learning Record in any format they wish. This could be directly in the word document as electronic evidence, in traditional pen and paper format, or indeed in any other format such as a Power Point, Prezi or even a video! The important thing here is to assess the quality and depth of the evidence presented, not prescribe a set format.



Examination Questions

1. Describe the stages of the fetch-decode-execute cycle.

……

……

……

……

……

…… **[4]**

2. Increasing the number of cores affects the performance of the CPU. Explain why this increase in performance is not linear

……

……

……

…… **[2]**

3. State the purpose of the address bus

……

…… **[1]**

4. Name one register other than the PC found in the CPU and describe its purpose

……

……

……

…… **[2]**

5. Explain why the Little Man Computing (LMC) instructions BRA / BRP would cause the contents of the PC to change

……

……

…… **[2]**

A LEVEL COMPUTER SCIENCE

1 - Structure and function of the processor (AS / A Level)

UNIT 1

Specification Points / Learning Objectives:

Core text book page ref: 124-131, 133-134

A Level	AS Level	Specification point description
1.1.1a	1.1.1a	The arithmetic logic unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Busses: data, address and control: How this relates to assembly language program
1.1.1b	1.1.1b	The fetch-decode-execute cycle, including its effect on registers
1.1.1c	1.1.1c	The factors affecting the performance of CPU, clock speed, number of cores, cache
1.1.1d		The use of pipelining in a processors to improve efficiency
1.1.1e	1.1.1d	Von Neumann, Harvard and contemporary processor architecture

Expectations / Learning Outcomes:

- Terms 1-21 from your **A Level Key Terminology** PowerPoint should be included and underlined.
- You must include at least one diagram which depicts the fetch-decode-execute cycle.
- You must include at least one diagram which shows the direction and connections of the 3 busses.
- You must include at least one diagram which illustrates how the various registers interact during a typical fetch-decode-execute cycle.

Grade	TG.	Breadth	Depth	Presentation	Understanding
A/A*		ALL	**LINK / FORMULATE** Create, Generate, Hypothesis, Reflect, Theorise, Consider	Quad Core	Quad Core
B/C		MOST	**EXPLAIN / ANALYSE** Apply, Argue, Compare, Contrast, Criticise, Relate, Justify	Dual Core	Dual Core
D/E		SOME	**DESCRIBE / IDENTIFY** Name, Follow Simple Procedure, Combine, List, Outline	Single Core	Single Core
U		FEW	Very little depth of understanding shown		
			MY ASSESSMENT GRADE IN THIS TOPIC IS:		

How To Improve:

My Response Is: (Set yourself specific targets / objectives as to how you will achieve your HTI)

-
-
-

The 'Fetch' part of the cycle

1) The PC holds the address of the next instruction to be carried out.
2) This address is copied into the MAR.
3) The contents of the address in the MAR are copied into the MDR.
4) The contents of the MDR are copied into the CIR.
5) The contents of the PC are incremented.

The 'Decode' part of the cycle

6) The contents of the CIR are then divided into the binary code standing for the operation to be carried out, and probably the address of the data that will be used by the program.
7) The control unit then interprets the operation code so that the processor knows what to do next.

The 'Execute' part of the cycle

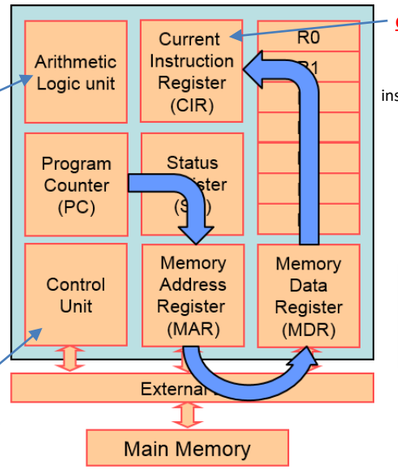
8) The address part of the instruction is copied from the CIR to the MAR.
9) The data found in the address in the MAR is copied to the MDR.
10) The data is used.

Point 10 in reality is more complex than this because the action depends entirely on the type of instruction.

Can you think of an example of what might actually happen here?

Arithmetic Logic Unit: "The part of the CPU where data is processed and manipulated. This processing and manipulation normally consists of arithmetic operations or logical comparisons allowing a program to make decisions."

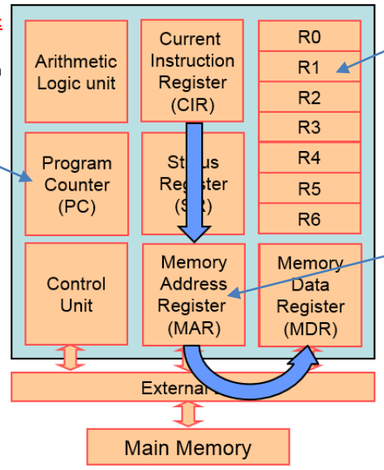
Control Unit: "The part of the CPU that manages the execution of instructions. The control unit fetches each instruction in sequence, and decodes and synchronises it before executing it by sending control signals to other parts of the computer."



Current Instruction Register: "A register in the control unit that stores the address of the next instruction currently being executed and decoded."

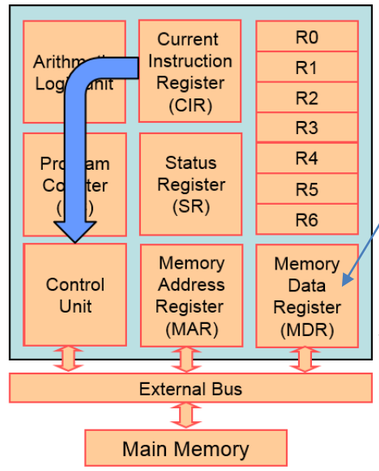
Program Counter: "A register in the control unit which holds the address of the next instruction to be executed."

Register: "Tiny areas of extremely fast memory located in the CPU normally designed for a specific purpose, where data or control information is stored temporarily."



Accumulator: "A special register within the ALU. It is used to hold the data currently being processed by the central processor. Any data to be processed is stored temporarily in the accumulator, the results ending up back in the accumulator being stored in the memory unit."

Memory Address Register: "A register in the CPU that stores the address of the memory location currently in use. In the fetch phase, this would be the address of the instruction being loaded; in the execute phase, it would be the address of the data being used."



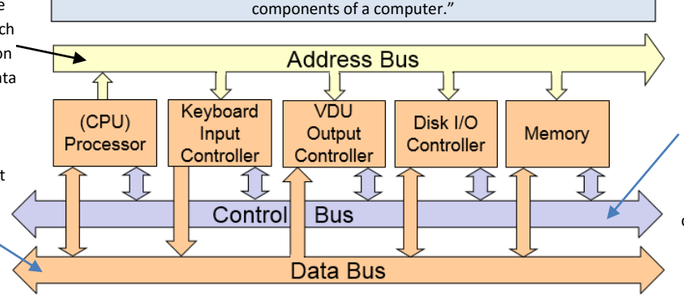
Memory Data Register: "A register in the CPU that stores data being transferred to and from the immediate-access store. It acts as a buffer, allowing the central processor and memory unit to act independently without being affected by minor differences in operation. A data item will be copied to the MDR ready for use at the next clock pulse, when it can either be used by the central processor or be stored in main memory."

Busses: "A common physical pathway shared by signals to and from several components of a computer."

Address Bus: "The part of the bus which carries identification about where the data is being sent."

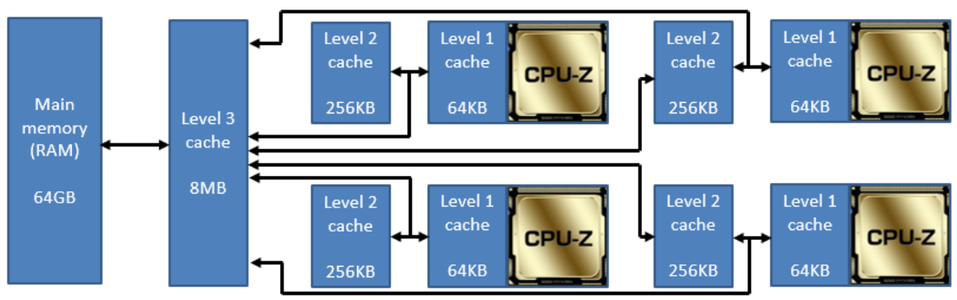
Data Bus: "The part of the bus which carries the actual information."

Control Bus: "This bus carries command and control signals to and from every other component of a computer."



Cache: "A part of the main store between the central processor and the rest of the memory. It has extremely fast access, so sections of a program and its associated data are copied there to take advantage of its short fetch cycle."

A quad-core i7 Haswell processor with dedicated Level 1 and Level 2 cache for each core and a shared Level 3 cache across all cores.

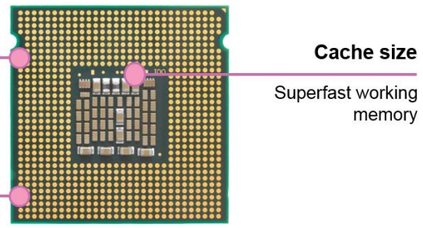


Level 1 Cache: Often located directly on the CPU itself, very low capacity, very expensive, typically runs at the same speed as the CPU.

Level 2 Cache: Often part of the CPU module, still very expensive, larger capacity, but still less than 1MB, runs at CPU or close to CPU speed

Level 3 Cache: Often located further away on motherboard, less expensive, larger capacity, now in MBs, typically shared between on cores.

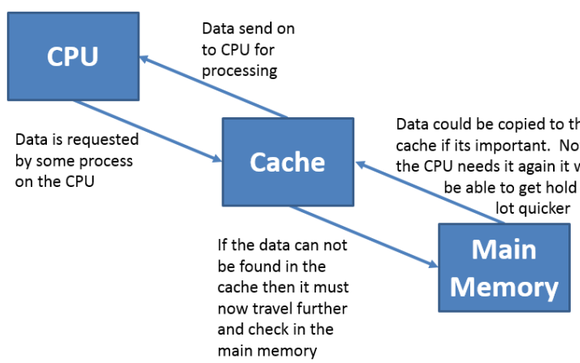
Cores: "A part of a multi-core processor. A multi-core processor is a single component with two or more independent actual CPUs, which are the units responsibly for the fetch-decode-execute cycle."



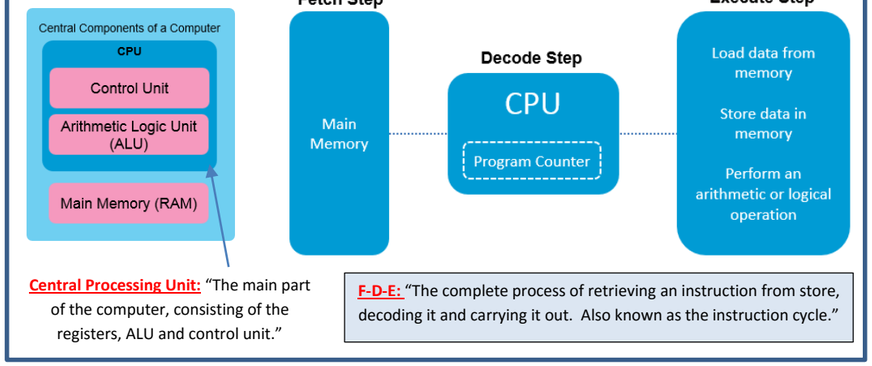
Clock Speed: "Measured in Hertz, the clock speed is the frequency at which the internal clock generates pulses. The higher the clock rate, the faster the computer may work. The "clock" is the electronic unit that synchronises related components by generating pulses at a constant rate."

PROCESSOR CLOCK SPEEDS

One cycle per second = 1 Hertz (Hz) = 1 instruction carried out each second
1 Kilohertz (KHz) = 1024 cycles per second
1 Megahertz (MHz) = 1,048,576 cycles per second
1 Gigahertz (GHz) = 1,073,741,824 cycles per second (Approximately 1 billion!)

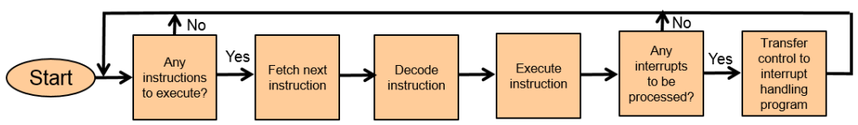


Von Neumann Architecture: "Traditional computer architecture that forms the basis of most digital computer systems. A single control unit manages program control flow following a linear sequence of "fetch-decode-execute"

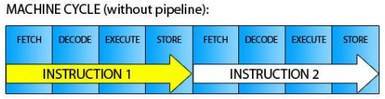


Central Processing Unit: "The main part of the computer, consisting of the registers, ALU and control unit."

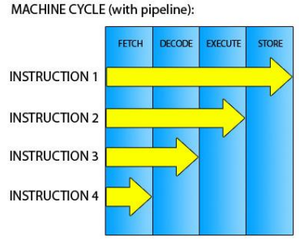
F-D-E: "The complete process of retrieving an instruction from store, decoding it and carrying it out. Also known as the instruction cycle."



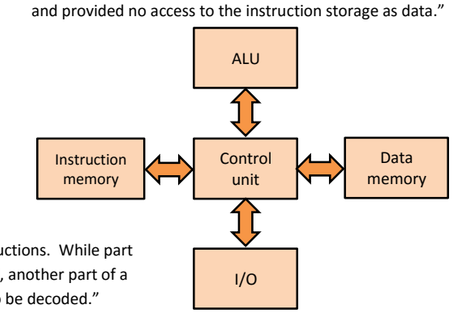
MACHINE CYCLE (without pipeline):



MACHINE CYCLE (with pipeline):



Harvard Architecture: "A computer architecture with physically separate storage and signal pathways for instructions and data. These early machines had data storage entirely contained within the central processing unit, and provided no access to the instruction storage as data."



Pipelining: The concurrent decoding of two or more machine instructions. While part of one instruction (for example, an address field) is being decoded, another part of a second instructions (for example, an operation code) may also be decoded."