

Von Neumann Architecture – Fetch decode execute cycle

Fetch

- 1) The **program counter** stores the **address** of the **next instruction to be fetched**. This is **copied to the Memory address register (MAR)** so the MAR is now loaded with **the address of the instruction that is being fetched**.
- 2) The memory address register (**MAR**) **places the address** to be used in the **'Address bus'** (**locating the memory location** of the address).
- 3) The **control unit then sends a signal** over the **control bus** to **read the memory**. I.e. The **MAR has triggered the read signal**. So the **memory chip in main memory** places the **contents** of the **address (instruction/data being asked for)** into the **data bus**.
- 4) **The data from the address** is transferred back to the **CPU over data bus** where it is loaded into the **MDR (memory data register)**. **Simultaneously the PC increments by one (holding the address of next instruction to be executed)**.
- 5) The contents of the **MDR** is copied into the **CIR (current instruction register)**. This consists of the **instruction (opcode)** and the **data part (operand)**.

Decode

The next step for the CPU to interpret the data just fetched is to decode it. This instruction in the CIR is split into the opcode and operand.

- 1) **The decode unit** of the CPU **analyses the instruction (opcode)** in the **CIR and decodes it**. The decode unit has an **instruction set** that **defines the opcodes** and so what commands to carry out. Every CPU will have its **own instruction set defining legitimate** commands. The opcode determines what part of the hardware is needed for execution.
- 2) A **series of micro-signals** are sent to different areas to **prepare the CPU** for readiness of the next step (**by the Control unit**).
- 3) The data to be operated on (**operand**) may be: **passed onto the ALU** OR the **address of the data** may be used with the operation which is then **copied to the MAR** OR the data to be operated on is **copied to the MDR**.

Execute

- 1) The appropriate **instruction** is carried out **upon the data (operand)** in the **MDR or accumulator** by the **execute unit** (containing an **arithmetic logic unit** that carries out calculations on the data). This is the executing of the instruction
- 2) The **results** of the arithmetic logic unit (ALU) are **output into another register called the accumulator**.
- 3) The **control unit assigns new signals** to **reset the CPU** to **prepare for the next cycle**.

Note: the CPU's own internal memory areas are called registers, the main memory, however, stores instructions/data of programs in memory addresses or locations.

A branch instruction in the CIR will cause the program counter to be loaded with the address of the instruction being branched to in main memory. This will occur when the decode unit analyses the instruction in the CIR. It will restart the whole fetch-decode-execute cycle.

What is each bus and what is each bus needed for?

- ❖ **A bus consists** of a series of paralleled wires each capable of transferring only 1 bit (logic 1 true or logic 0 false). They are present in a computer system connecting the CPU, memory and other hardware or peripheral devices.

The address bus = *Identifies the memory address locations within the main memory. The contents of the MAR (memory address register) are loaded onto this bus containing details on address of the instruction to be fetched.*

It allows the CPU to define where the content of the data bus (or memory address) is going to or coming from.

- ❖ **The control bus** = *The control unit transfers control signals along this bus which manages the activity of the memory and peripheral devices. I.e. it coordinates the flow of data. Transfers command, timing and status information signals.*

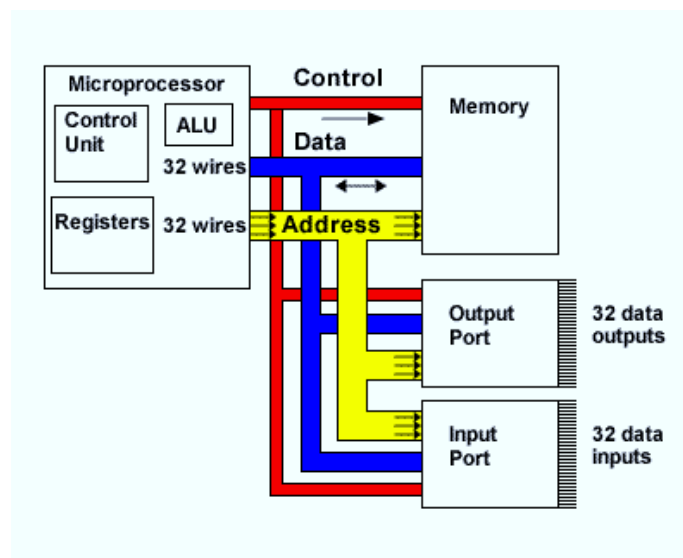
The CPU needs to receive and send control information between other parts of the computer to manage the flow of data. Many parts make this up but an example is the control line which joins the CPU and Power supply, sending signals to confirm the CPU to only boot when the power is steady.

- ❖ **The data bus** = *Transfers data and instructions between the CPU, main memory and peripheral devices.*

The data bus in a standard (Von Neumann) CPU handles both data and instructions and its purpose is to allow data/instructions to pass from one area to another.

Describing the buses

- The width of the bus determines how many bits the CPU can handle at once.
- Typical widths are 8 bit, 16 bit, 32 bit and 64 bits (refer to bus width).
- Each wire in an address bus or data bus can transfer only 1 bit, logic 0 or 1. A 16-bit computer has buses that are 16 wires wide so they can transfer 2^{16} locations (with regards to the address bus) or 2^{16} bits of data/instructions (with regards to the data bus).



The arithmetic logic unit

The arithmetic logic unit (ALU) will be made up of two parts: an arithmetic part which carried out the mathematical operations (+, -, /, X) and a logic part which carries out other logic functions such as comparisons (<, >, = e.c.t.)

The results are stored in the accumulator in the CPU (a type of register in the Cache / CPU's local memory) as this allows for faster access than if it were to be stored in the main memory. The transfer speeds across buses between the CPU and memory decrease efficiency.