Role and Components of CPU

Role

The CPU carries out the FDE cycle

- Controls the movement of data and instructions
- Fetch data and instructions from memory
- Decode and execute instructions
- Perform arithmetic operations
- Perform logical operations

Components

- Registers are memory that hold any instructions, data and/or memory addresses that are about to be used by the CPU.
- Buses - Parallel wires connecting two or more components of the CPU.

Main Components

- Arithmetic logic unit - Completes arithmetic and logic operations such as adding or subtracting.
- Control Unit - Sends signals to control how the processor runs, and controls how data is moved around parts of the CPU and memory.
- Cache - Very fast memory in the CPU that is filled with frequently / next to be accessed instructions that are built into the CPU or placed on the motherboard.

Registers

- Program Counter - Stores address of next instruction
- Memory Address Register - Stores address of data to be fetched from or sent to memory.
- Memory Data Register - Stores data to be fetched from or sent to memory, like a middle man.
- Accumulator - Stores result of calculations by the arithmetic logic unit.
- Current Instruction Register - Stores actual instruction being decoded or executed.
- Interrupt Register - Checks to see if any interrupts require servicing.

Buses

- Address Bus - unidirectional, carries the address of data that is in the memory address register from the processor to memory.
- Data Bus - Bidirectional, carries data between the processor and memory.
- Control Bus - Bidirectional, carries control signal between processor and the components.

Control Unit

- Decodes instructions.
- Sends control signals down the control bus to coordinate the movement of data through the processor and execution of instructions.
- Controls and coordinates the activates of the CPU directing the flow of data between CPU and other devices.
- Controls buses.

Performance of a CPU

Main Factors

- Clock speed
- Cache size
- Number of cores
- Width of buses
- Pipelining

Clock Speed

- Controlled by the control unit.
- The speed of the FDE cycle is controlled by the clock speed.
- A CPU with a high clock speed will carry out more instructions per second and therefore have a higher performance.

Cache

- Small, very fast memory.
- Filled with frequently accessed and next to be accessed instructions.
- Built into the CPU or placed on the motherboard.
- Faster to use cache than RAM.
- Generally, the larger the cache, the higher the performance.
- Most modern CPUs have three caches (L1, L2, L3).

Number of Cores

- A multicore processor is made up of two or more independent cores.
- Each core is a distinct processing unit on the CPU with its own individual components and cache.
- A multicore CPU will have higher performance than a single-core CPU with the same clock speed.
- The biggest performance gain when using a multicore processor is when the software has been specifically written to run on multicore processors simultaneously.
- When multitasking different cores can run different applications or even run on the same one.

Width of Buses

Word

- A word is a group of bits that refers to the number of bits used to store data in the CPU's registers and cache.
- A larger word size means that the CPU will be able to process larger numbers in a single operation.
- A size or width of a bus is how many bits it carries in parallel.
- Modern CPUs have a 64bit word size.

Width of Buses

- The width of the address bus determines the maximum possible memory capacity of the new system.
- A wider data bus can transmit larger values or more bites per instruction, and so as the data bus transmits the data held in memory, the word size affects the total amount of memory a computer system can handle.
- The largest value that can be held in a word is related to the size of the operating system.

Pipelining

- Using pipelining, the computer architecture allows:
  - The next instructions to be fetched at the same time as the processor is performing arithmetic or logical operations.
  - Holding them in a buffer close to the processor until the instruction can be performed.
- Limitations:
  - It is not always possible to accurately predict what instruction needs to be fetched and decoded next. E.g. an instruction that acts differently depending on the result of a calculation.
  - Pipelining will choose one of the instructions to at least have a 50-50 chance of having the correct one.
  - But if the wrong instruction is chosen, we must "flush" all instructions from the pipe.
  - The more flushing that happens, the less benefit pipelining gives us.

Fetch-Decode-Execute Cycle

Fetch

- The program counter points to the next instruction to be fetched.
- The contents of PC is sent / copied into memory address register.
- The address (contents of MAR) is sent along the address bus.
- A signal is sent from the control unit on the control bus.
- The content of address in memory is accessed, and can now be transferred using the data bus into the MDR.
- If it is an instruction, the contents of the location in memory is sent to the current instruction register using the data bus.

Decode

- Instruction in the CIR is decoded by the decode unit.
- Opcode gives the command, operand gives the data / address.

Execute

- Address of the data is sent to the MAR which then sends it to the RAM.
- CU sends a signal down the control bus to read the data.
- Contents of memory is sent down the data bus to the MDR.
- Contents of the MDR normally then sent to the ACC depending on the instruction.
- After each execute the cycle is repeat.

CPU Architecture

Von Neumann

Components

- Control Unit
- Arithmetic Logic Unit
- Memory Unit
- Registers
- Input / Outputs

Design

- Instructions and the program data are stored in the same memory.
- It follows a linear fetch-decode-execute cycle.
- One instruction at a time.
- In particular, the same memory and bus are used for both data and instructions.
- In particular, the same memory and bus are used for both data and instructions.
  - A program must be placed in main memory to be executed.
  - The machine code instructions are fetched from memory one at a time, decoded and executed in the processor.
- This design is still used in most computers produced today.
- The same data bus is used to transfer both data and instructions.
- Similarly, a single address bus is used to transfer the address of data and instructions.
- The same word length is used for all memory, whether it holds data or instructions.

Von Neumann Bottleneck

- Time required to move instructions from the CPU to the main memory and back can cause latency
- The CPU has to wait a certain time which can be quite long. It could be shorter to access registers.
- To overcome this one possibility is to use cache, another is to use the Harvard architecture.

Harvard Architecture

Data and instructions are stored in:

- Separate memory units
- With separate buses
- Reading (fetching) data and writing data can be done at the same time as reading an instruction
- Tends to be used in RISC processors.

Modern high performance CPU chips incorporated aspects of both Von Neumann and Harvard architectures.

Multicore and Parallel Systems

Multicore Processor

- Single chip that contains two or more independent processing units (cores).
- Each core can execute ordinary CPU instructions on their own.
- Most manufacturers integrate these cores onto a single physical chip.
- This chip is known as a chip multiprocessor (CMP).
- They can be further enhanced by features such as sharing a local cache of memory and having some sort of inter-core communication system.
- Doubling the cores rarely equates to doubling the processing speed.
- Improvements on performance depend largely on the individual software programs being run.
- Typically, the gains in performance will be limited to the parts of the program that have been designed to run in parallel over the multiple cores.
- This is known as Amdahl’s law.

Parallel Processing

- Parallel processing of program instructions by dividing them amongst multiple processors or processor cores, with the objective of running the overall program in less time.
- Often uses pipelining.
- There are more than one processor / core in a complex operating system.
- The different processors will work together to execute a single job split into tasks.
- Each task can then be performed by any processor.
- Gains depend on the task and whether a program has been designed to make use of parallel.

CISC and RISC Processors

Instruction Sets

- At the core of all computers are their instruction sets.
- This is the complete set of all instructions written in machine code that are recognisable and thus executable by the CPUs.
- Different CPU designs will have a different core instruction set.
- These belong to two main categories, RISC and CISC.

Complex Instruction Set Computing

- Machine manufacturers offer larger and more complex sets of machine instructions available on their CPUs.
- The manufacturer needs to add more complex circuitry. Therefore more complex hardware is needed but it can support simpler coding.
- Each instruction requires multiple machine cycles.
- Most desktops and laptops use a CISC architecture.
- Produced by intel and AMD.

Reduced Instruction Set Computing

- Hardware is often simple but software needs to be more complicated.
- Each instruction can be carried out in a single machine cycle.
- Pipelining can be used to speed up processing. It can only be used on RISC processors as each instruction only takes one machine cycle.
- Most smartphones and tablets use RISC CPUs (based on arm).
- Cheaper (less silicone required).
- Lower clock speed, but performs simpler tasks better than CISC CPUs.
- Lower energy requirements.

Co-Processors

- A co-processor is any additional processor used for a specialised task.
- The purpose is to improve overall performance of the computer system by executing concurrently with the main CPU but with a highly bespoke set of electronics.

Graphical Processing Unit

- A co-processor.
- Originally used for rendering graphics.
- Due to the high number of cores they are not just used for rendering graphics.
- Each core works slower than a CPU core and has fewer features.
- Slower than CPUs, but highly specialised.

What They're Best At:

- **CPU's**
  - Complex operations, on small data sets.
- **GPU's**
  - Simple manipulations but to much larger data sets.
  - Better at any operation where a larger data set is used.