|  |  |
| --- | --- |
| **Describe classic Von Neumann architecture, identifying the need for, and the uses of, special registers in the functioning of a processor** | |
| **ALL** | **Describe Von Neumann architecture in simple terms** |
| **MOST** | **Explain the need and uses of special registers** |
| **SOME** | **Analyse the limitations of this architecture** |

**TASK**:

Create in interactive image map that shows the main area of a CPU. When you mouse-over each one an explanation should appear.

Follow the instructions in the help sheet. If you can think of another way of doing this then go for it!

Useful links:

[Link 1](https://www.image-maps.com/) [Link 2](https://en.wikipedia.org/?title=Control_unit) [Link 3](https://en.wikipedia.org/wiki/Arithmetic_logic_unit) [Link 4](https://en.wikipedia.org/wiki/Memory_address_register) [Link 5](https://en.wikipedia.org/wiki/Memory_data_register)

Include:

* + Memory – what can be stored there?
  + Control Unit
  + Arithmetic Logic Unit
  + Input / Output
  + Registers:
    - Current Instruction Register
    - Memory Address Register
    - Memory Data Register
    - Program Counter
    - Accumulator

**EXTENSION**: Investigate the “Von Neumann bottleneck”. Add to your webpage a blog post that explains what it is and discusses its potential impact in modern CPU design. You should include an explanation of Harvard architecture in your answer (minimum 500 words).

|  |  |
| --- | --- |
| **Describe, in simple terms, the fetch/decode/execute cycle, and the effects of the stages of the cycle on specific registers** | |
| **ALL** | **Explain the purpose and order of the fetch / decode / execute cycle** |
| **MOST** | **Describe the order in which data is switched to and from the CPU registers** |
| **SOME** | **Discuss the fetch / decode / execute cycle in the context of modern CPU design** |

**Videos:**

<https://www.youtube.com/watch?v=jFDMZpkUWCw>

<https://www.youtube.com/watch?v=xfJbpCJSpd8>

Create an animated .gif or .swf that could be added to your theory website.

It should describe the flow of data through all the registers during the fetch, decode, execute cycle.

Other registers:

Find out what is meant by:

* Status register
* Stack pointer register
* General purpose registers

|  |  |
| --- | --- |
| **Discuss co-processor, parallel processor and array processor systems, their uses, advantages and disadvantages** | |
| **ALL** | **Know some modern CPU architectures** |
| **MOST** | **Explain the advantages and disadvantages of a range of modern CPU architectures** |
| **SOME** | **Analyse the best CPU architecture for a given application** |

Make notes on the following using the teacher presentation. Then create a page for your site and add the information to it.

**Single Instruction Single Data (SISD)**

***Pipelining***

<http://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/pipelining/>

**Single Instruction Multiple Data (SIMD)**

***Array / Vector processor***

**Multiple Instruction Multiple Data (MIMD)**

***Multiple processor***

***Coprocessor***

Complete this table and add it to your site:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technology** | **Class** | **Where it might be used** | **Advantages** | **Limitations** |
| Pipelining | SISD | CPU | Can allow for scheduling of other instruction to be fetched and decoded while another instruction is being processed | Sometimes has to wait for the instructions to be completed to fetch new instructions |
| Array processing | SIMD | CPU with lots of instructions | Allows for more instruction to be fetched and decoded per cycle | Limited to one instruction at a time |
| Multiple cores | MIMD | CPU that needs a lot of processing power | Allows for multiple instructions to be processed | Needs an OS that can support multiple cores. Costs more. |
| Coprocessor | MIMD | GPU to render graphics | Takes the load from the CPU to load and display the graphics with a dedicated system | Only certain instructions can be processed by the coprocessor |

Coprocessor

Parallel processing

<http://www.c-jump.com/CIS77/CPU/VonNeumann/lecture.html>

**Task**:

* Read the lecture above – go over again some of the work we have looked at so far
* Pay particular note to the bottom of the lecture and the history of CPUs
* Using software of your choice create a timeline of Intel CPUs.
* You should start with the 8088 and finish with the latest Core-i7
* For each iteration you should state the new technology that has been included. Is there some examples of the types of processing we have already learned about?

EXTENSION:

<https://en.wikipedia.org/wiki/Titan_(supercomputer)>

Read the article above. Which chips have been chosen to operate in the Titan supercomputer? What are the advantages of this configuration bearing in mind the type of processing this computer is expected to do?

|  |  |
| --- | --- |
| **Describe and distinguish between Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) architectures** | |
| **ALL** | **Remember some facts about RISC and CISC** |
| **MOST** | **Be able to compare RISC and CISC** |
| **SOME** | **Confidently identify the best applications for RISC and CISC** |

Create a graphic to compare RISC and CISC architecture.

Add this to your website.

Which is better? RISC or CISC? How has thinking changed over time?

You should discuss the advantages and disadvantages of each architecture and suggest applications for their use.

Answer this question in no less than 500 words.