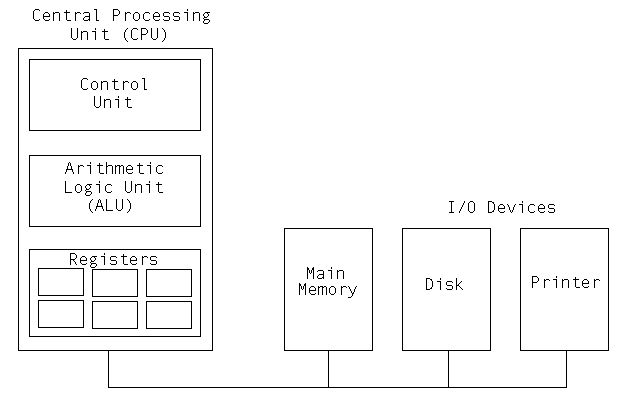
**CPU Architecture**

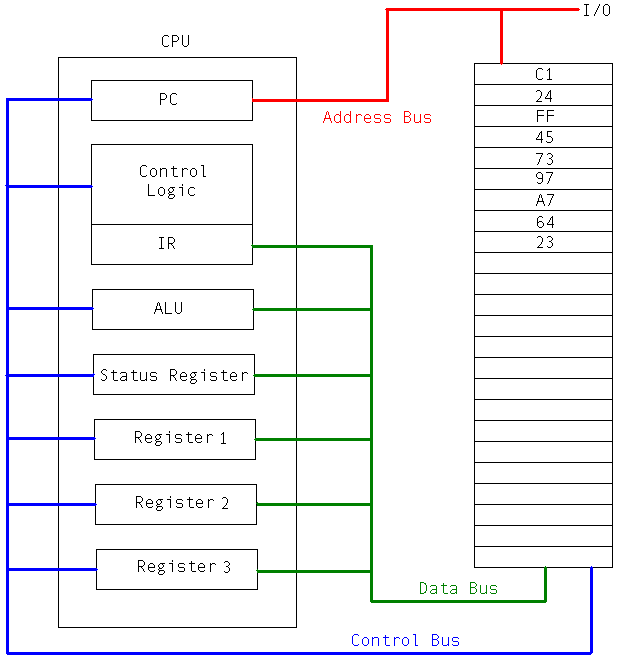
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| http://staffweb.cms.gre.ac.uk/~sp02/fetchexecute/images/cartoon.gif | Algorithm | Assembly Code | Machine Code | http://staffweb.cms.gre.ac.uk/~sp02/fetchexecute/images/computer.gif |



**The Fetch Execute Cycle**

The CPU executes each instruction in a series of small steps:

1. Fetch the next instruction from memory into instruction register.
2. Change the program counter so that it points to the following instruction.
3. Determine the type of instruction just fetched.
4. If the instruction uses data in memory determine where they are.
5. Fetch the data, if any into internal CPU registers.
6. Execute the instruction.
7. Store the result in the proper place.
8. Go to step 1 to begin executing the following instruction.



**Translation Programs**

* Compiler
* Assembler

**Languages**

* Assembly
  + Reduced Instruction Set Computer (RISC)
  + Complex Instruction Set Computer (CISC)
* High Level
  + Visual Languages
  + Object Orientated
  + 3GL

**Speed Increase**

* Increase Clock Speed
* Architectures
* Physics

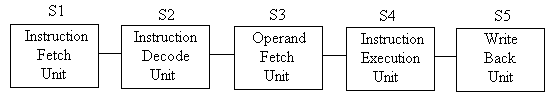
**Architectures**

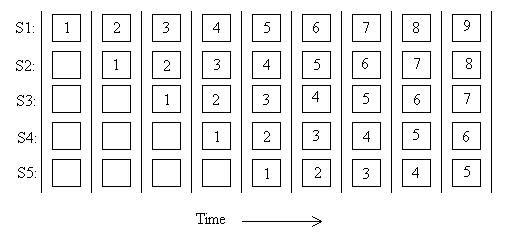
* Scalar
* Pipelined
* Superscalar
* Vector - Multiple processors undertake multiple operations simultaneously with a single intruction.

**Pipelined**

Fetching the instruction from memory is a major bottleneck in instruction speed

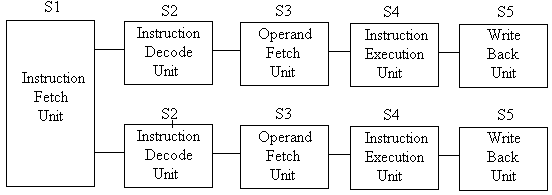
Pipelines can increase speed by a factor of about 10

A five stage pipeline with nine clock cycles illustrated



**Multiple pipelines**

Single instruction fetch unit fetches pairs of instructions together and puts each one in its own pipeline

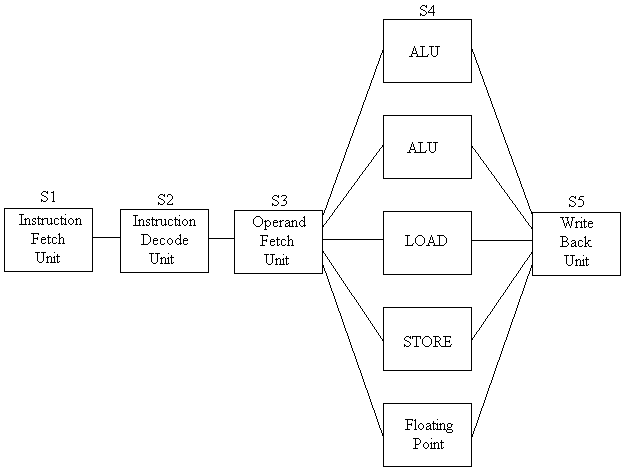


1 and 2 pipelines are common max 4, often up to 10 stages in pipeline, however requires Liner code, branch prediction, efficient compiler to work effectively.

**Superscalar**

Single pipeline multiple functional units.

Implicit in the idea of a superscalar processor is that stage 3 can issue instructions considerably faster than stage 4 is able to execute them.



**Parallelism**

* Processor Parallelism
  + Single computer with multiple processors.
  + Shared memory, often each processor has both individual and shared memory.
  + Difficult after 64 processors.
* Multi Computers
  + Large number of interconnected computers with no common memory

**Semiconductor Types**

Used to make transistors/intergrated circuits

CMOS (Complementary Metal-Oxide Semiconductor) 0V to 2.4V, Low power consumption, slower than TTL

TTL (Transistor Transistor Logic) 0V to 5V Average Speed, Average Power Consumption

ECL (Emitter Coupled Logic) -1.025V to -1.325V, Fast, High Power Consumption

**Limiting factors**

* Speed of light,
* Heat

**Next generation**

* Carbon tubes (Bucky Balls)

**The future**

* Quantum computing