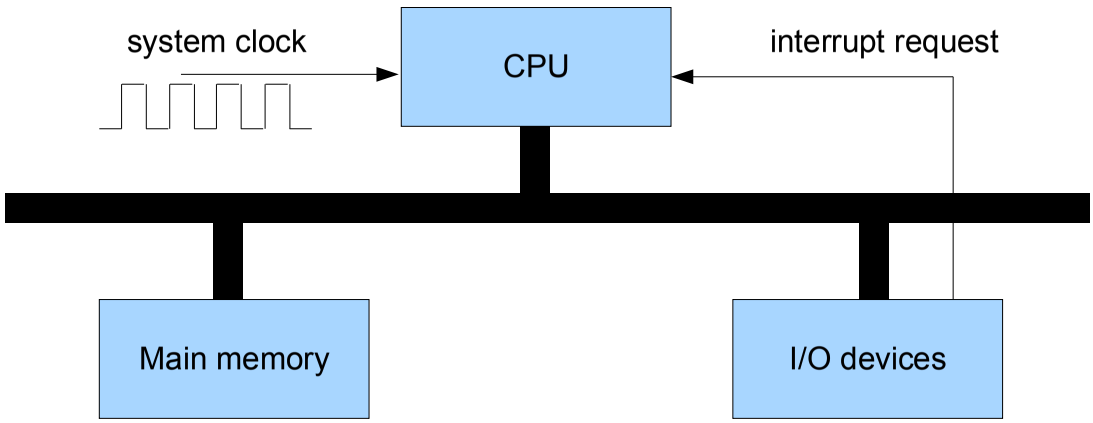
The main parts of a computer

- The main parts of a computer, whether it is a desktop computer, a mobile phone or your digital clock are:
  - Central Processing Unit (CPU)
    - This is the “brains” of the computer where all the calculations take place.
  - Main memory
    - This is where the computer stores the information that the CPU uses.
  - Input/Output devices (I/O devices)
    - These are all the devices that the user interacts with; the keyboard, display, communication devices (wired and wireless), hard disk, DVD drive, etc.

System bus

- The main parts of the computer are connected by the system bus.



System bus (2)

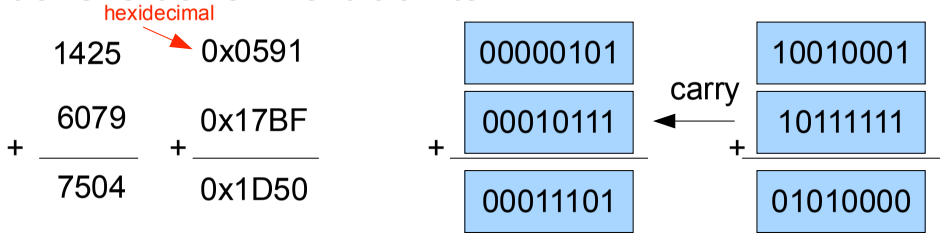
- The system bus contains 3 buses
  - Address bus
    - Each main memory location and each I/O device has its own address, so we can differentiate between different memory locations and I/O devices.
  - Data bus
    - Carries data between the CPU, the main memory and the I/O devices. Typically the data flows between the CPU and main memory, or between the CPU and the I/O devices.
  - Control bus
    - Consists of special signals, mainly to synchronise the address bus and data bus.

Address bus

- Each location in memory has a unique address.
- The size of the address bus determines the maximum number of memory locations that the CPU can access.
  - A 3-bit address bus would have 3 signals, A0, A1 and A2.
    - The maximum number of different addresses for a 3-bit address bus is 2^3=8 (i.e. 000,001,010,011,100,101,110,111)
  - A 16-bit address bus has 2^{16} (=65,536) different addresses.
  - A 32-bit address bus has 2^{32} (=4,294,967,296) different addresses.
    - The bigger the address bus the more memory that the CPU can access.
  - But more signals are need! Harder to manufacture.

Data bus

- The data bus carries data from one device to another.
- Generally the CPU is involved.
  - From main memory to the CPU.
  - From CPU to main memory.
  - From I/O device to CPU
  - From CPU to I/O device.
- An 8-bit data bus can represent numbers from 0 to 255.
  - If the calculations require numbers greater than 255, then the calculations is done in 8-bit units.



Control bus

- The control bus contains signals that are used to synchronise the transfer data to and from the CPU.
  - It contains:
- Signals that say when the address bus is valid.
- Signals that control whether data is read from or written to a memory location or I/O device.
- Interrupt Signals (requests and acknowledgements)
- Reset and timing signals.

Main memory

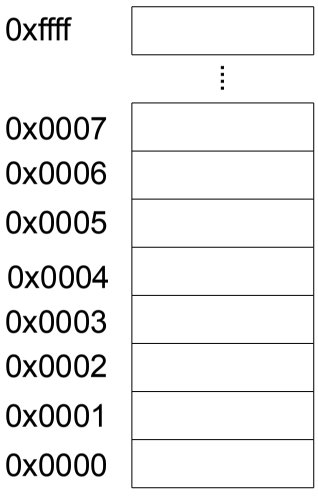
- Each memory location has a unique address.
- The number of unique addresses is determined by the size of the address bus.

#bits	Address range	#unique addresses
16	0x0000 - 0xffff	65,536
20	0x00000 - 0xfffff	1,048,576 (1M)
24	0x00000 - 0xffffff	16,777,216
32	0x00000000 - 0xffffffff	4,294,967,296 (4G)

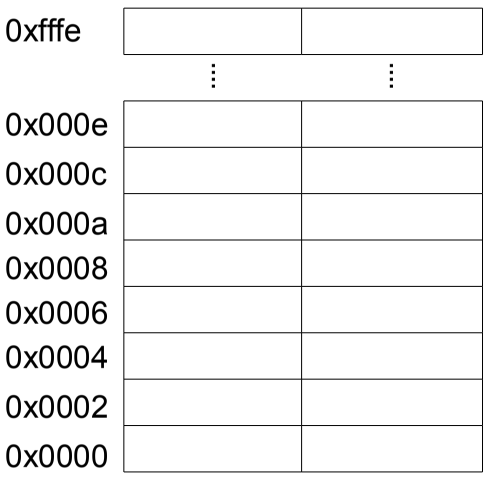
- Each memory location contains a 8-bit (byte) value.
- A 16-bit data bus requires 2 bytes of data.
  - Each 16-bit data value is at an even address.
- Main memory contains the program and its data.

Main memory (2)

16-bit address bus
and an 8-bit data bus

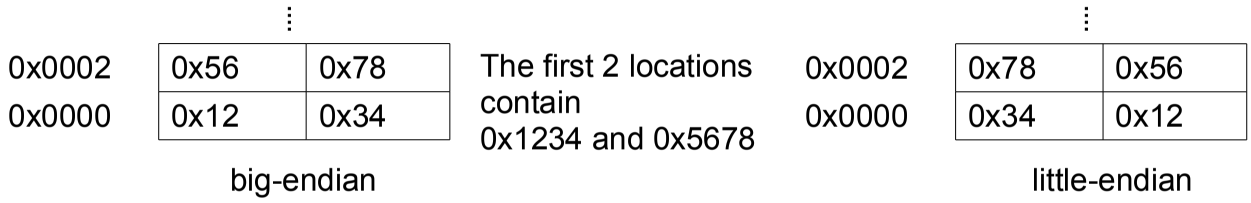


16-bit address bus
and a 16-bit data bus



Big-endian and Little-endian

- With a 16-bit data bus, the order is which the 2 bytes that form the 16-bit word can either be:
  - Big-endian
    - Most-significant (MS) byte first then least-significant (LS) byte next.
    - Convention adopted by Motorola
  - Little-endian
    - Least-significant (LS) byte first then most-significant (MS) byte next.
    - Convention adopted by Intel



What happens if data is transferred from an Intel-based machine to a Motorola-based machine?

The CPU

- Modern central processor units (CPUs) consist of 4 major subsystems.
  - Control Unit
    - This subsystem controls the executing of the CPU.
  - Arithmetic Logic Unit (ALU)
    - This subsystem performs the arithmetic and logical operations.
  - Registers
    - Local, and very high speed, memory that includes the status flag register.
  - Cache
    - Specialised high speed memory

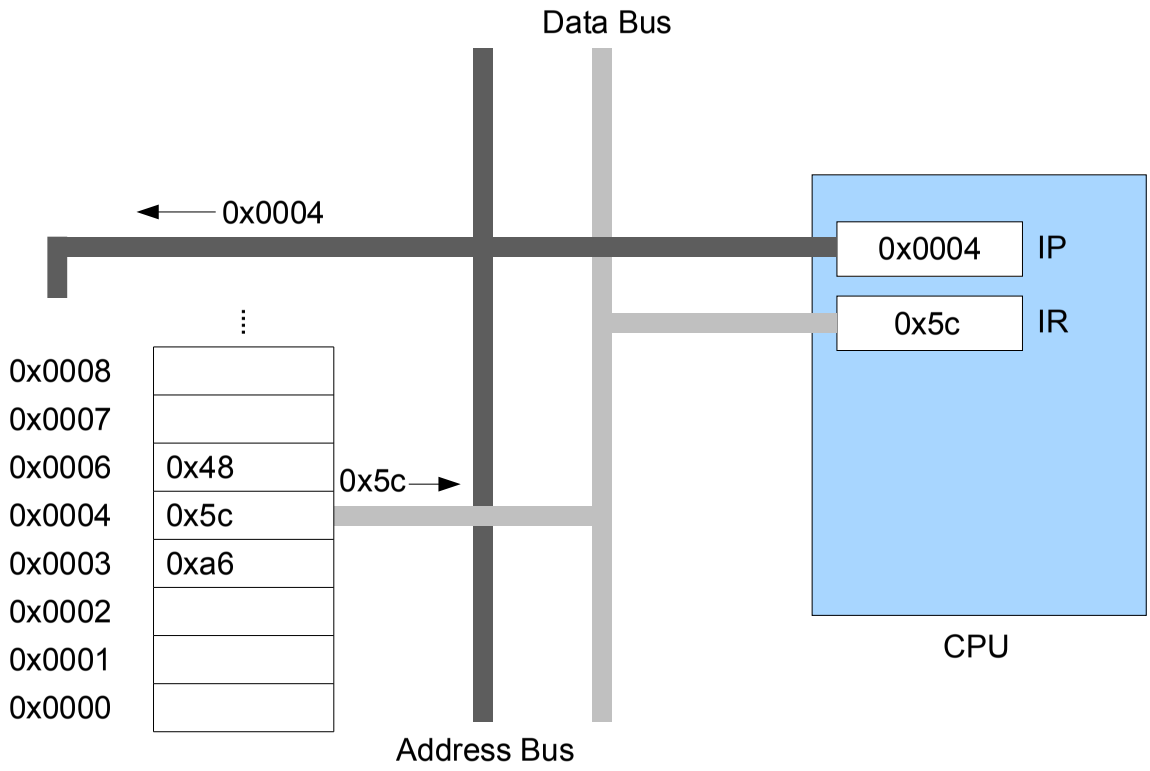
Control Unit

- The Control Unit (CU) controls the operation of the CPU, and in particular, the fetch-decode-execute cycle.
  - Fetch
    - Retrieve the next instruction to execute.
  - Decode
    - Decode the instruction and set-up the necessary data paths to allow the execution of the instruction.
  - Execute
    - Execute the instruction.

Control Unit (2)

- There are 2 special registers in the CPU that are involved in the fetch operation.
  - The Instruction Pointer (IP)
    - Also known as the Program Counter (PC).
    - A register that contains the address of the memory location that contains the next instruction to execute.
    - The IP register is large enough to reference any address in the memory.
  - The Instruction Register
    - During the fetch operation the next instruction to execute is copied into the Instruction Register.

Control Unit (3)



Control Unit (4)

- An instruction is made up of an operation code (opcode) and an arbitrary number of operands.
  - In the x86 processors the instruction add AL,#01 is stored as 0x04 0x01 in memory

Opcode for add AL,# Operand

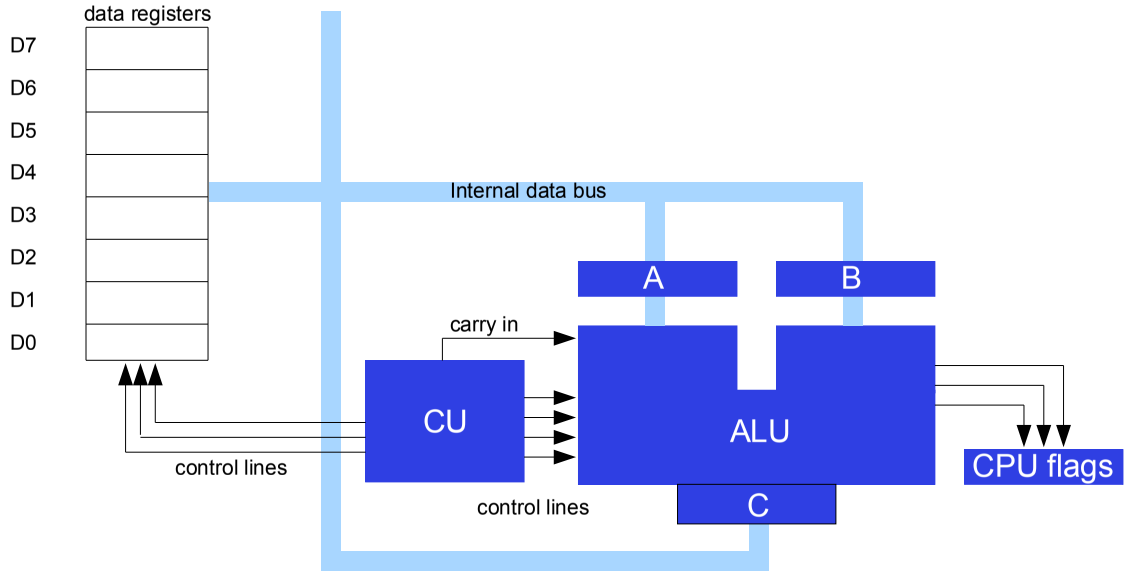
- Once the opcode is loaded in to the IR, it is decoded.
  - Sets up the data paths between the registers, ALU and memory.
  - The decoding is highly dependent on the internal structure of the CPU.

Control Unit (5)

- In a CISC (Complex Instruction Set Computer) each instruction has its own little program to execute it. This program is called ***microcode*** and is stored internally in the CPU.
- In a RISC (Reduced Instruction Set Computer) each instruction is typically larger and encodes all the information required to execute the instruction. In RISC systems the control unit is typically busier than a CISC system.

ALU

- The Arithmetic Logic Unit (ALU) performs the arithmetic and logical operations as determined by the Control Unit.



Registers

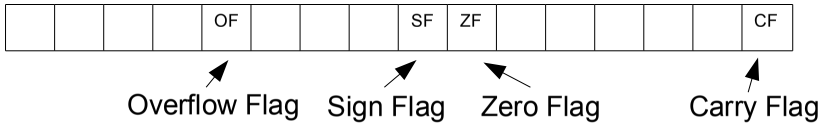
- Registers are local memory that is connected to the CPU's internal data bus.
  - Accessing data in a register is significantly faster than accessing data in main memory because:
    - There is no need to put an address on the system address bus before reading the data.
    - The CPU's internal data bus is faster than the external system data bus.
- Different CPUs have a different number of registers and different sizes of registers.
- Some registers are general purpose and some have specific purposes.

Registers (2)

- Some special purpose registers:
  - Accumulator
    - On older CPUs this was directly connected to the ALU and arithmetic or logical operations involving the accumulator where faster than other registers. On modern CPUs the accumulator is still faster for some specialised operations.
  - Index Registers
    - Contain the address of a memory location and can be used to read or write to that location. Typically index registers have fast increment and decrement operations.

Registers (3)

- Flag Register
  - The individual bits in this register are typically updated by the ALU to reflect to result of the most recent operation.



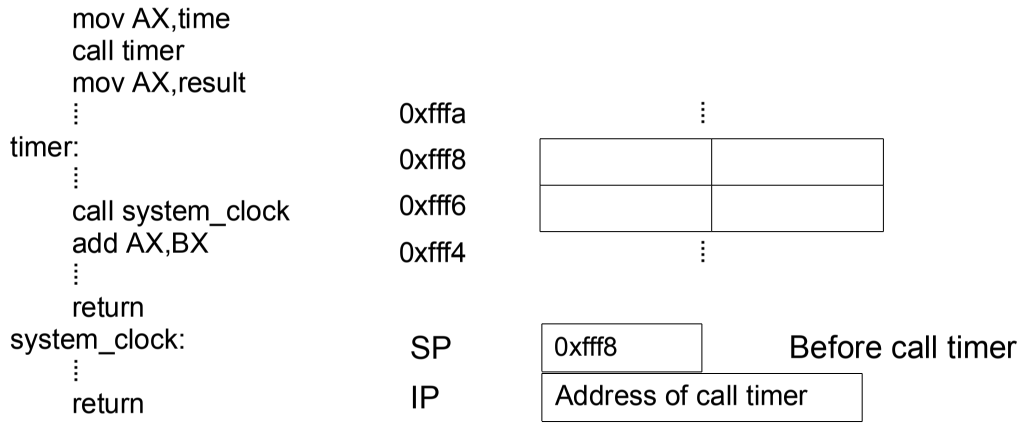
  - Carry Flag
    - The Carry Flag is set to 1 if an ALU operation results in a carry out from the MS bit, else it is 0.
  - Sign Flag
    - The Sign Flag is set to 1 if the MS bit is 1, else it is 0.

Registers (4)

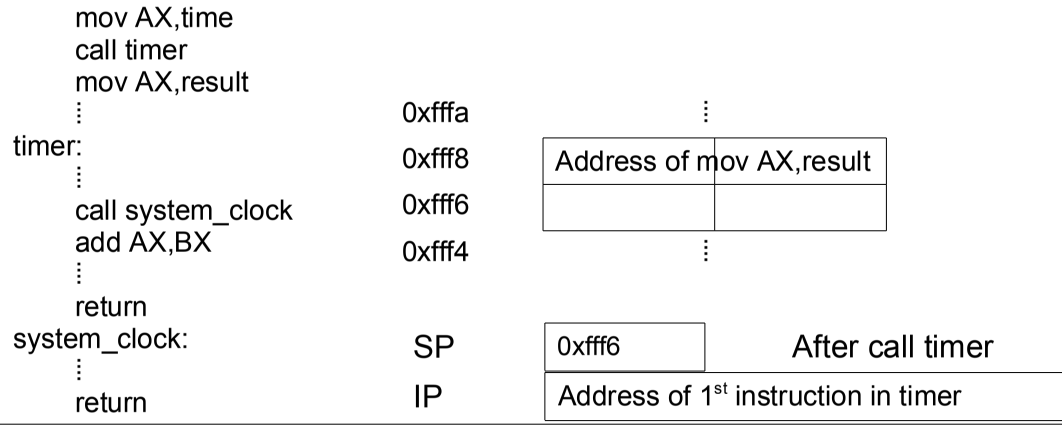
  - Zero Flag
    - The Zero Flag is set to 1 if the ALU operation resulted in ZERO, else it is 0.
  - Overflow Flag
    - This Overflow Flag is set to 1 if the addition of 2 positive numbers results in a negative number, of the addition of 2 negative numbers results in a positive number, else it is 0.

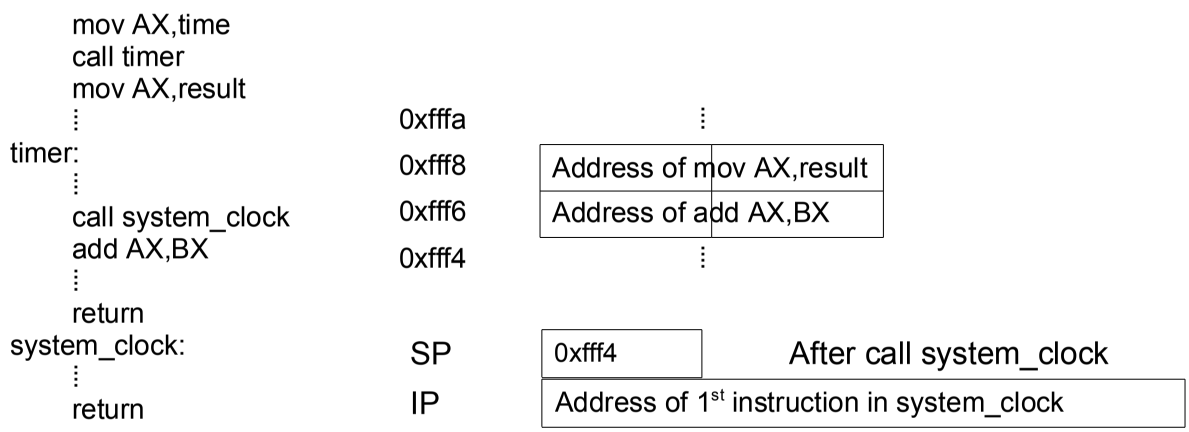
Registers (5)

- Stack Pointer (SP)
  - This register holds the address of the next vacant location in the stack.
  - The stack is an area of main memory that is primarily used to hold the the return addresses from subroutine calls.
  - CALL and RETURN instructions automatically update the stack and the stack pointer.

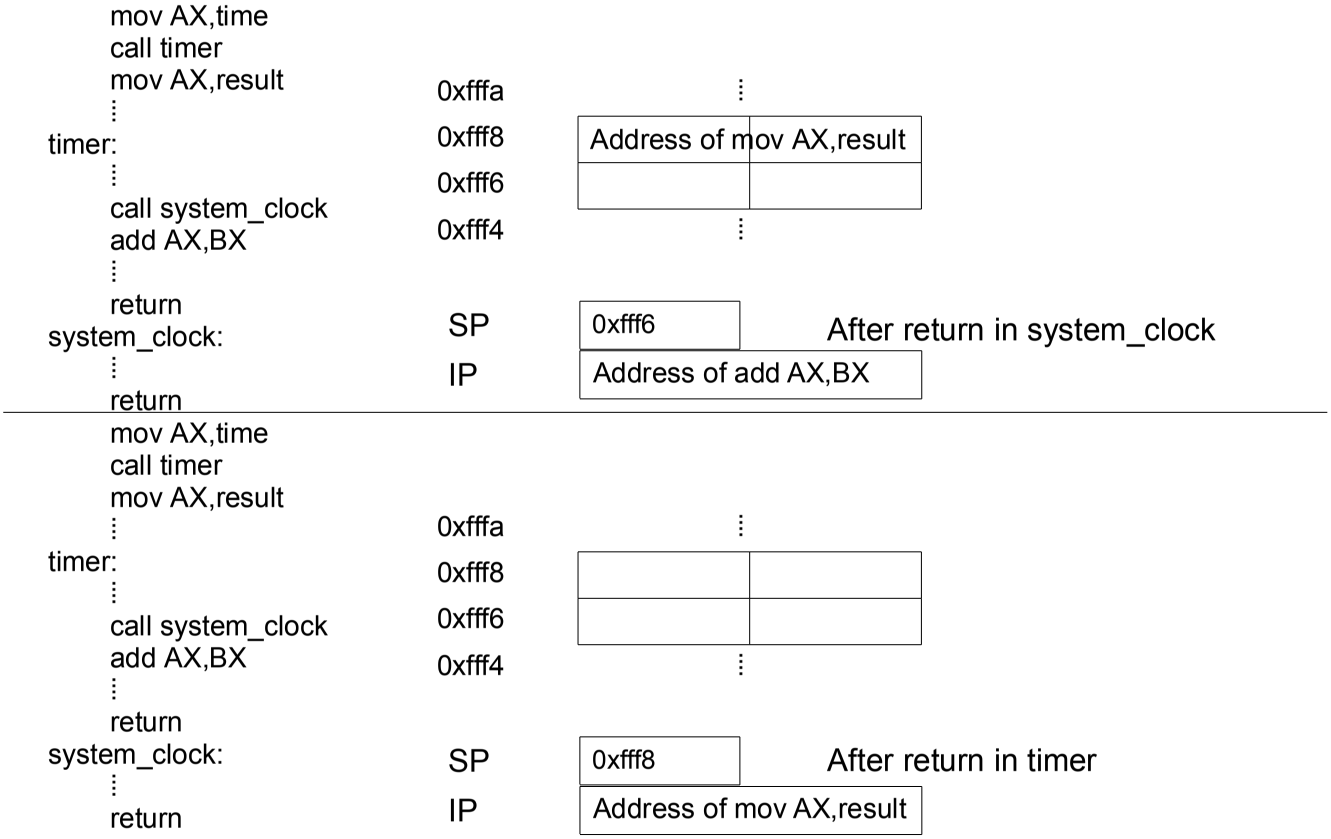


Registers (6)





Registers (7)



Cache

- Programs have loops, even machine code programs.
- Good programs have small functions.
- This gives rise to two affects.
  - Any instruction that is about to be executed was most likely executed recently.
  - Any memory location to be accessed was most likely accessed recently.
- Since accessing main memory is slower than memory in the CPU it would be more efficient to keep a copy recently execute instructions and accessed memory in the CPU.

Cache (2)

- A ***Cache*** is specialised memory for storing recently executed instructions and accessed memory.
  - On-chip cache (level 1 cache) is built into the CPU and is connected to the CPU's internal data bus.
  - Level 2 cache is not in the CPU but is high speed memory located close to the CPU.
- Cache memory is smaller than main memory and it only holds a copy of a fraction of main memory.
  - The most recently used fraction.
- Cache memory is organised differently to main memory. Rather than being a linear collection of memory locations, it is a pool of memory locations.
  - Each entry in the cache has a tag and data value.

Cache (3)

- When an instruction or data value is to fetched, its address is used to calculate a tag.
- The cache is searched for an entry with a matching tag.
- If a matching tag is found this is called a **cache hit** and the data attached to the tag is returned.
- If a matching tag is not found this is called a **cache miss** and main memory is accessed.
  - The data from main memory is then added to the cache along with its tag.

Cache (4)

- A cache is designed to enable an efficient search for tags in order to determine if there is a cache hit or cache miss without searching the whole cache.
- A cache is determined by:
  - Its size.
  - How it is organised.
  - A replacement policy: When a cache is full and a new entry is added, which value do we lose. The oldest? The least used?
  - A write policy: If a data value is modified in the cache, is the main memory updated immediately (***write-through***) or when the entry corresponding to the data value is removed from the cache (***write-back***)?