The Fetch-Execute Cycle

Stored program computers are designed to fetch, decode and execute instructions in a continuous cycle. This is known as the fetch-execute cycle.

Fetch

- The Program Counter (PC) contains the address of the next instruction to be fetched
- The address contained in the PC is copied to the Memory Address Register (MAR).
- The instruction is copied from the memory location contained in the MAR and placed in the Memory Buffer Register (MBR).
- The entire instruction is copied from the MBR and placed in the Current Instruction Register (CIR)
- The PC is incremented so that it points to the next instruction to be fetched

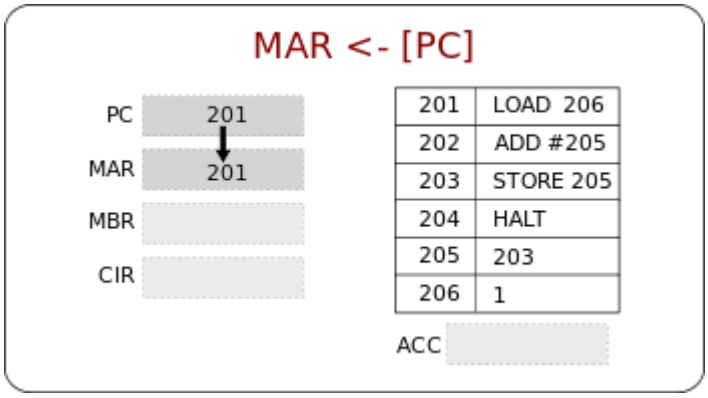
Execute

- The address part of the instruction is placed in the MAR
- The instruction is decoded and executed
- The processor checks for interrupts (signals from devices or other sources seeking the attention of the processor) and either branches to the relevant interrupt service routine or starts the cycle again.

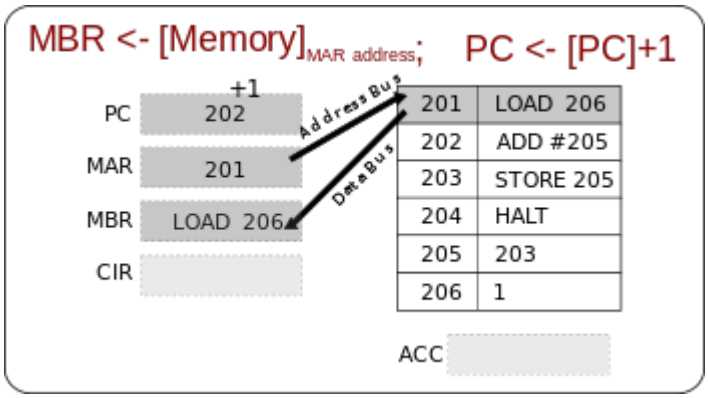
Registers/circuits involved

The circuits used in the CPU during the cycle are:

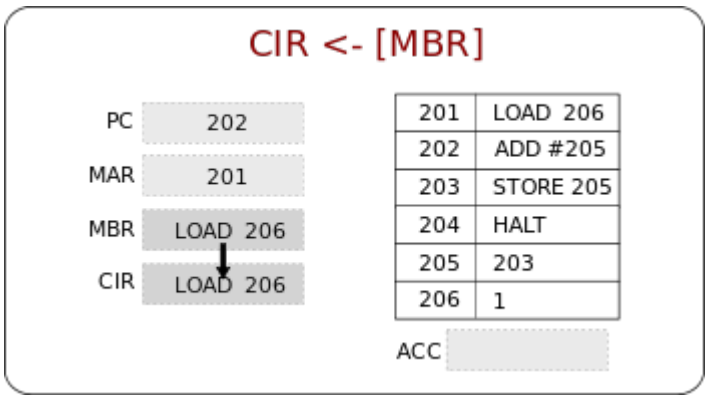
Program Counter (PC) - an incrementing counter that keeps track of the memory address of which instruction is to be executed next...
Memory Address Register (MAR) - the address in main memory that is currently being read or written
Memory Buffer Register (MBR) - a two-way register that holds data fetched from memory (and ready for the CPU to process) or data waiting to be stored in memory
Current Instruction register (CIR) - a temporary holding ground for the instruction that has just been fetched from memory
Control Unit (CU) - decodes the program instruction in the CIR, selecting machine resources such as a data source register and a particular arithmetic operation, and coordinates activation of those resources
Arithmetic logic unit (ALU) - performs mathematical and logical operations



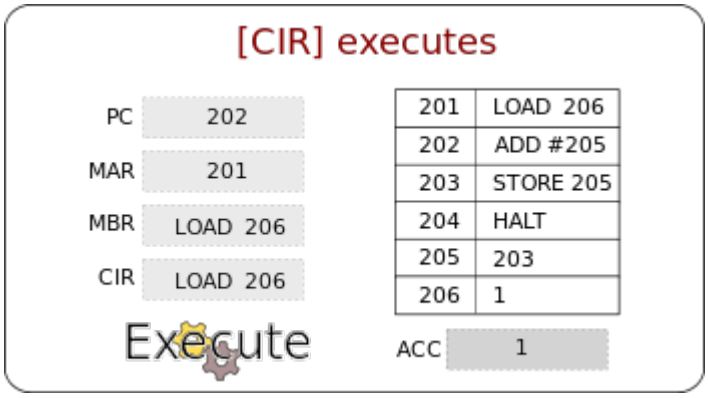
The address of the next instruction to be executed, held in the **Program Counter**, is copied to the **Memory Address Register** ...



The address is sent from the **MAR** along the **address bus** to main memory.
The instruction found at that address is returned along the **data bus** to the **Memory Buffer Register**. While this is being done, the **Program Counter** is incremented by 1, to point to the next instruction to be executed....



The **MBR** passes the instruction to be executed into the **Current Instruction Register** ...



The instruction is executed (using the **ALU** if needed) ...

... and the cycle continues!