BBC Bitesize

The fetch decode execute cycle – GCSE Computer Science video

NARRATOR: Welcome to the central processing unit, otherwise known as CPU city.

I'll show you how the city carries out the fetch-decode-execute cycle.

Everything begins when a program has to run. These buildings are called registers. They're like government administration offices.

All information flowing through the city passes through one of these buildings. Data, addresses, program instructions, all registered here.

The first register to get going is called the program counter. As you can see it's set to zero, which represents an address. But before we visit that address, we need to know how information moves around the city.

These pathways are called buses. They carry digital signals around CPU city. There are different types of buses and each type has a different purpose.

The Address bus carries addresses. The Data bus carries data. And the control bus carries control instructions issued by this place, the Control Unit. It's the centre of operations. It receives and decodes instructions, then sends out commands to the other components to get those instructions executed.

So the first thing CPU city needs, are those instructions. It gets them from outside of the city, from this place; the memory.

The memory is the library where data is kept in the form of a computer program. CPU city can store data here as well as fetch from it.

As you can see, there are only a few lines of written code currently being stored here. Bear in mind that a modern computer can store billions of lines of code.

The fetch decode execute cycle begins when the memory address held in the program counter is sent on the address bus from the program counter to the Memory Address Register. This is where we hold the memory address of the next data value or instruction which is going to be fetched or stored.

At the moment, the memory address is address zero. The zero is then sent to the memory on the address bus with the aim of fetching some data that CPU city can work on. When it reaches address zero, the program instruction stored there is copied onto the data bus, and carried back to CPU city. Because the instruction is data, and not an address, it's loaded into the Memory Data Register, which holds values and instructions until they are ready to be used by the CPU. It also holds values while they are waiting to be sent to the memory. And as LOAD 3 is an instruction, it's automatically copied from the Memory Data Register into the current instruction register which holds instructions ready to be executed. The instruction is finally copied over to the control unit.

Before it completes the fetch stage, the control unit sends out an instruction to increment the address in the program counter, moving it up one digit. So that once the fetch-decode-execute cycle is complete, the program counter will be loaded and ready with the next memory location it needs to start the next cycle. And with that, the fetch stage is complete.

Now starts the decode stage.

The control unit decodes the instruction LOAD 3 to mean LOAD the value at memory location 3 into this place, the Accumulator.

The accumulator is another special purpose register.

It's like a warehouse where data is temporarily stored while mathematical and logical operations are performed by this place ... the Arithmetic Logic Unit, it is the factory of the city, and uses data stored in the accumulator to process calculations.

So now that the Control Unit has decoded the instruction, the decode stage is complete. The control unit can now tell the other components to execute the program instruction, and we're into the execute stage of the cycle.

The 3 in LOAD 3 is an address, so the control unit first updates the Memory Address Register ... then a signal is sent to retrieve the data stored at memory address 3 … before the data there is sent back to the Memory Data register ... And finally makes its way to the Accumulator. Where that number 13 is stored, bringing the execute stage to an end, and completing the first fetch-decode-execute cycle.

The next cycle begins and the whole process proceeds in much the same way.

The next instruction, SUB 4, is fetched by the control unit.

Now if LOAD 3 means load whatever value is at location 3 in the memory, it follows that SUB 4 means to subtract whatever value is being held at location 4 in the memory. The value at location 4 is fetched and we can see it's the number 8. It gets loaded into the arithmetic logic unit.

Now the full instruction, to subtract 8 from 13, can be executed.

That number 13 gets sent to the arithmetic logic unit, which does the required calculation, before the result, 5, is sent back and stored in the accumulator.

Ending the execute stage, and completing the second fetch decode execute cycle.