

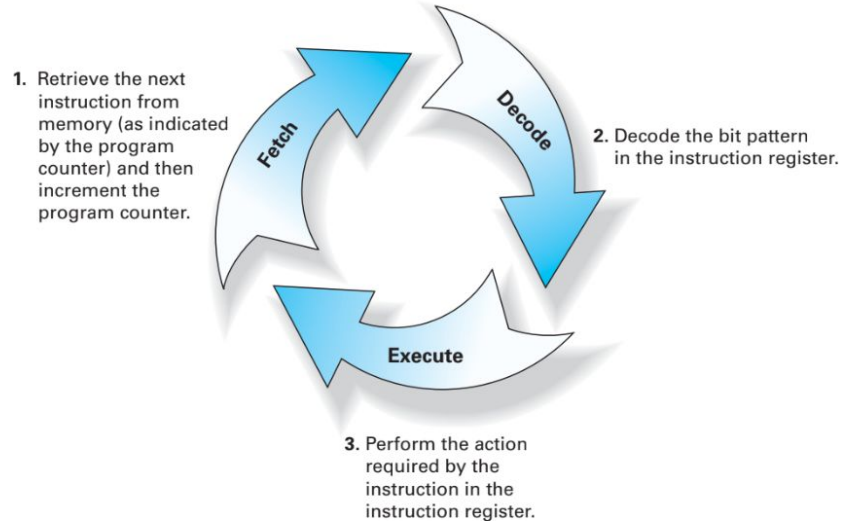
A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

# An Example of Program Execution

By Joseph Koop

# The Machine Cycle

**Figure 2.8** The machine cycle



Machine Cycle: Fetch, Decode, Execute

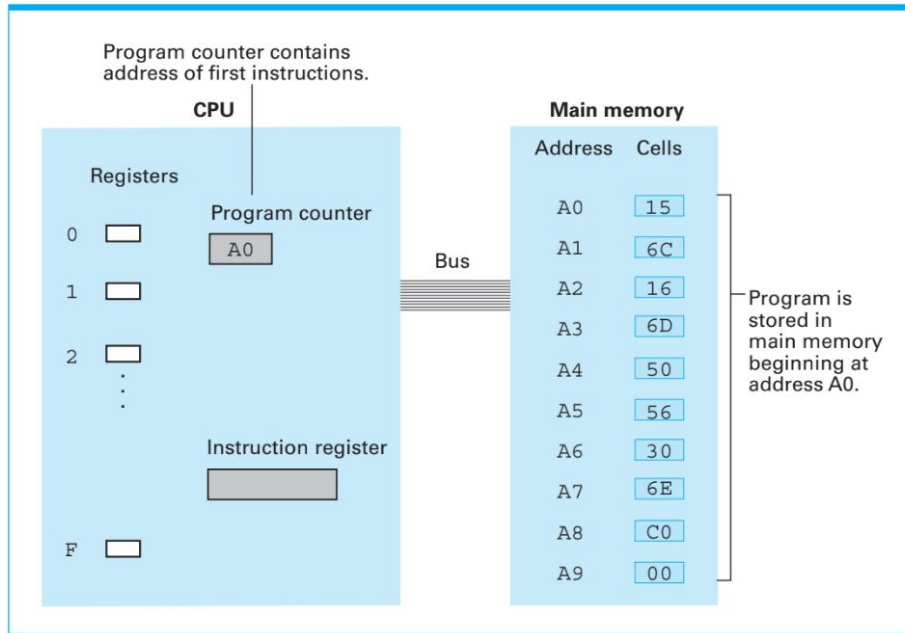
Fetch step brings instructions from the main memory into instruction register, then increases program counter by 2

Decode step translates the instruction into machine code

Execute step activates the right circuitry to perform the instruction

# Program Counter and Instruction Register

**Figure 2.10** The program from Figure 2.7 stored in main memory ready for execution



The program counter holds the address of the next instruction to carry out

The instruction register holds the current instructions that are being carried out

# A program that adds two numbers

## 2.2 Adding values stored in memory

- Step 1. Get one of the values to be added from memory and place it in a register.
- Step 2. Get the other value to be added from memory and place it in another register.
- Step 3. Activate the addition circuitry with the registers used in Steps 1 and 2 as inputs and another register designated to hold the result.
- Step 4. Store the result in memory.
- Step 5. Stop.

Figure 2.7 An encoded version of the instructions in Figure 2.2

| Encoded instructions | Translation  |
|----------------------|--|
| 156C                 | Load register 5 with the bit pattern found in the memory cell at address 6C.   |
| 166D                 | Load register 6 with the bit pattern found in the memory cell at address 6D.   |
| 5056                 | Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0. |
| 306E                 | Store the contents of register 0 in the memory cell at address 6E.   |
| C000                 | Halt.  |

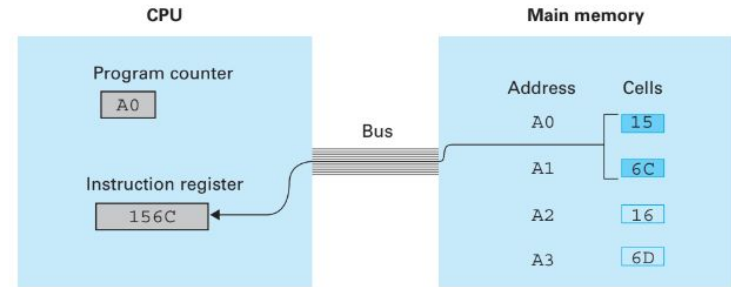
# Fetch Step

Starts at A0

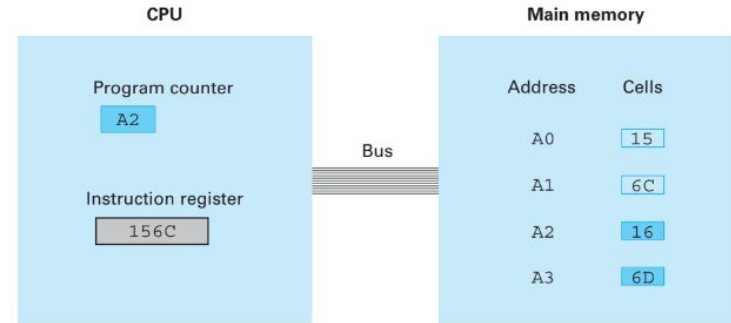
Puts the contents of A0 and A1 into the instruction register

Changes program counter to A2

**Figure 2.11** Performing the fetch step of the machine cycle



a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.



b. Then the program counter is incremented so that it points to the next instruction.

# Decode Step

**Figure 2.7** An encoded version of the instructions in Figure 2.2

| Encoded instructions | Translation  |
|----------------------|--|
| <b>156C</b>          | Load register 5 with the bit pattern found in the memory cell at address 6C.   |
| 166D                 | Load register 6 with the bit pattern found in the memory cell at address 6D.   |
| 5056                 | Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0. |
| 306E                 | Store the contents of register 0 in the memory cell at address 6E.   |
| C000                 | Halt.  |

## 156C

The first number indicates the operation

1 = Load

The second number indicates the CPU register to work with

5 = Load into register 5

The last two numbers indicate the address of the memory cell from which to take the number.

6C = Take the bit pattern from cell 6C

156C = Load the bit pattern from memory cell 6C into register 5.



# Execute Step

Carries out the instruction by performing the required operation (Add, Subtract, Load, Move, Store, etc.)

Goes back to fetch step

Repeats the process until the end of the program



The End