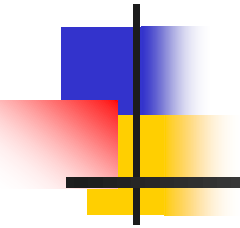


Emulator program of microcontroller

Assignment 1

Deadline: 12PM, Friday of Week 6

Microcontroller – instruction set

History

B. Pascal
(1623-1662)

C. Babbage
(1791-1871)

Von Neumann
(1903-1957)

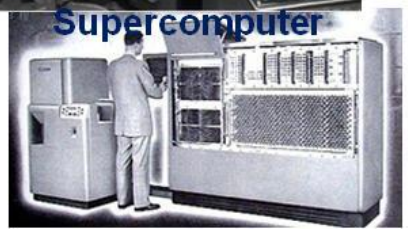
Alan Turing
(1912-1954)

Bill Gates
(1955)

?

Abacus
(China)
500 B.C

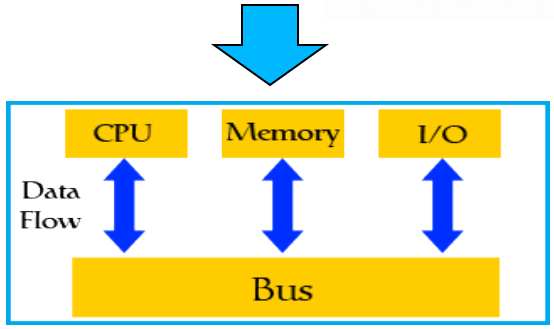
Babbage
Machine



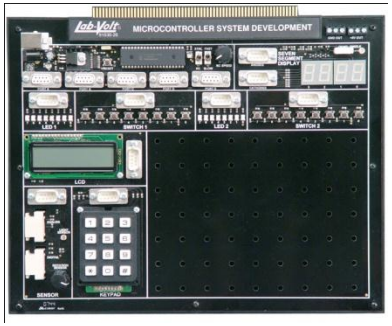
Supercomputer

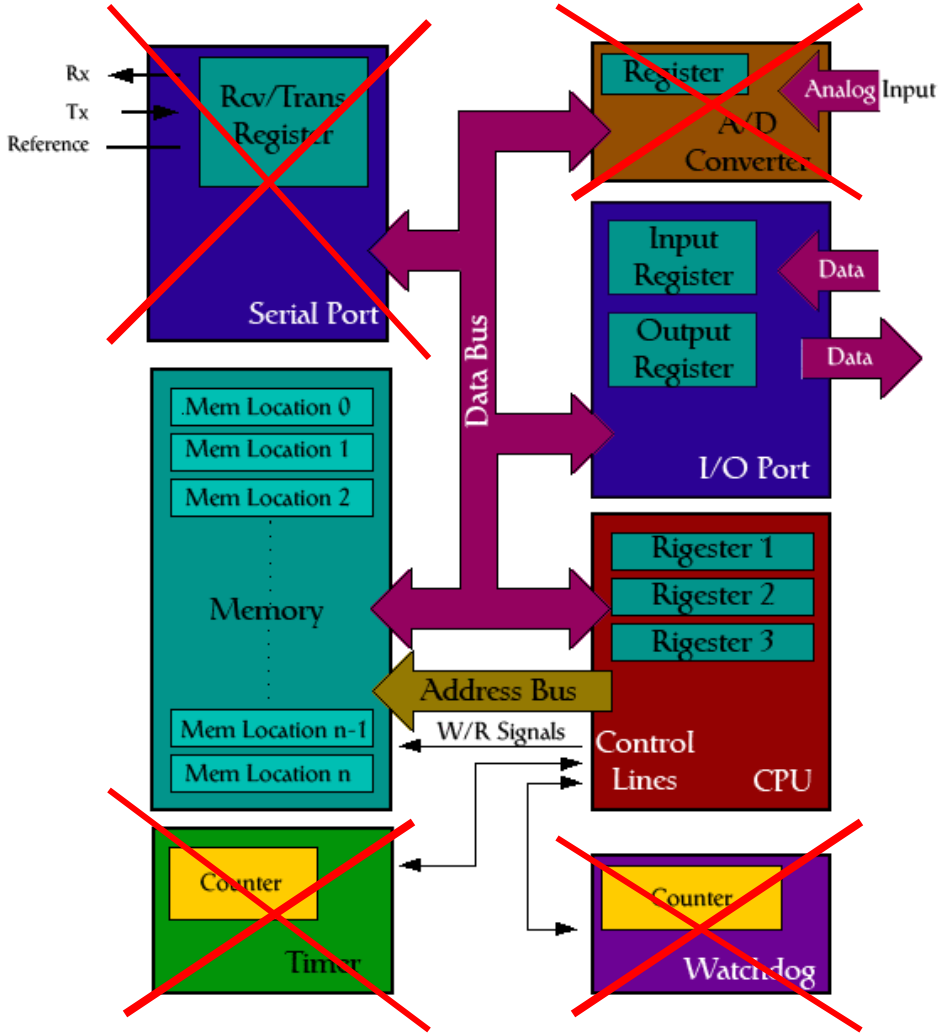
Personal computer

Microcontroller - Von Neumann Architecture



Von Neumann architecture

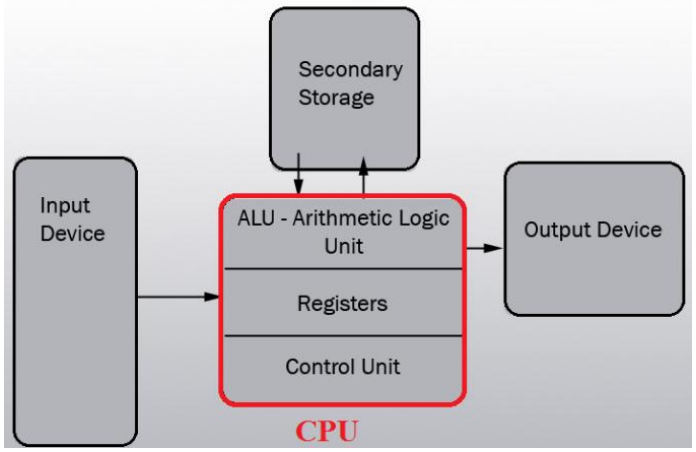


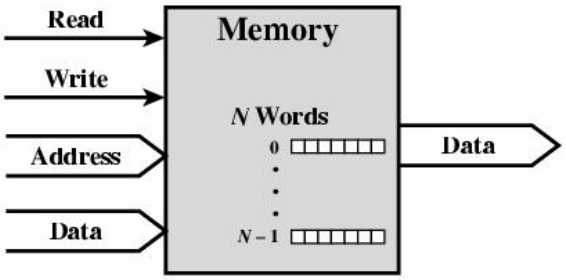


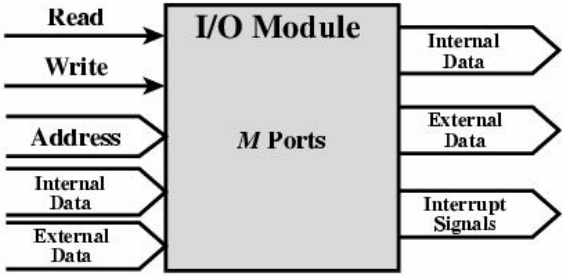
Microcontroller – Von Neumann Architecture

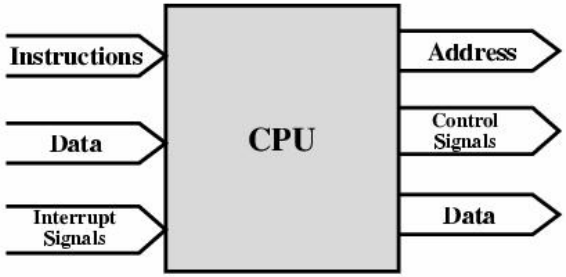
Main CPU's components:

- **ALU:** performs arithmetic operations: addition (ADD), subtraction (SUB), multiplication (MUL) and division (DIV) & logical operations: Boolean functions such as AND, OR and NOT (used during conditional branching).
- **Control unit**: Decodes op-codes and controls instruction data flow.
- **Program counter & registers**: 8 bits
- **Instruction set:**
  - ✓ **Instruction set** & **address modes**
  - ✓ **Data format**: (sign/unsigned) Integer (8-bit).

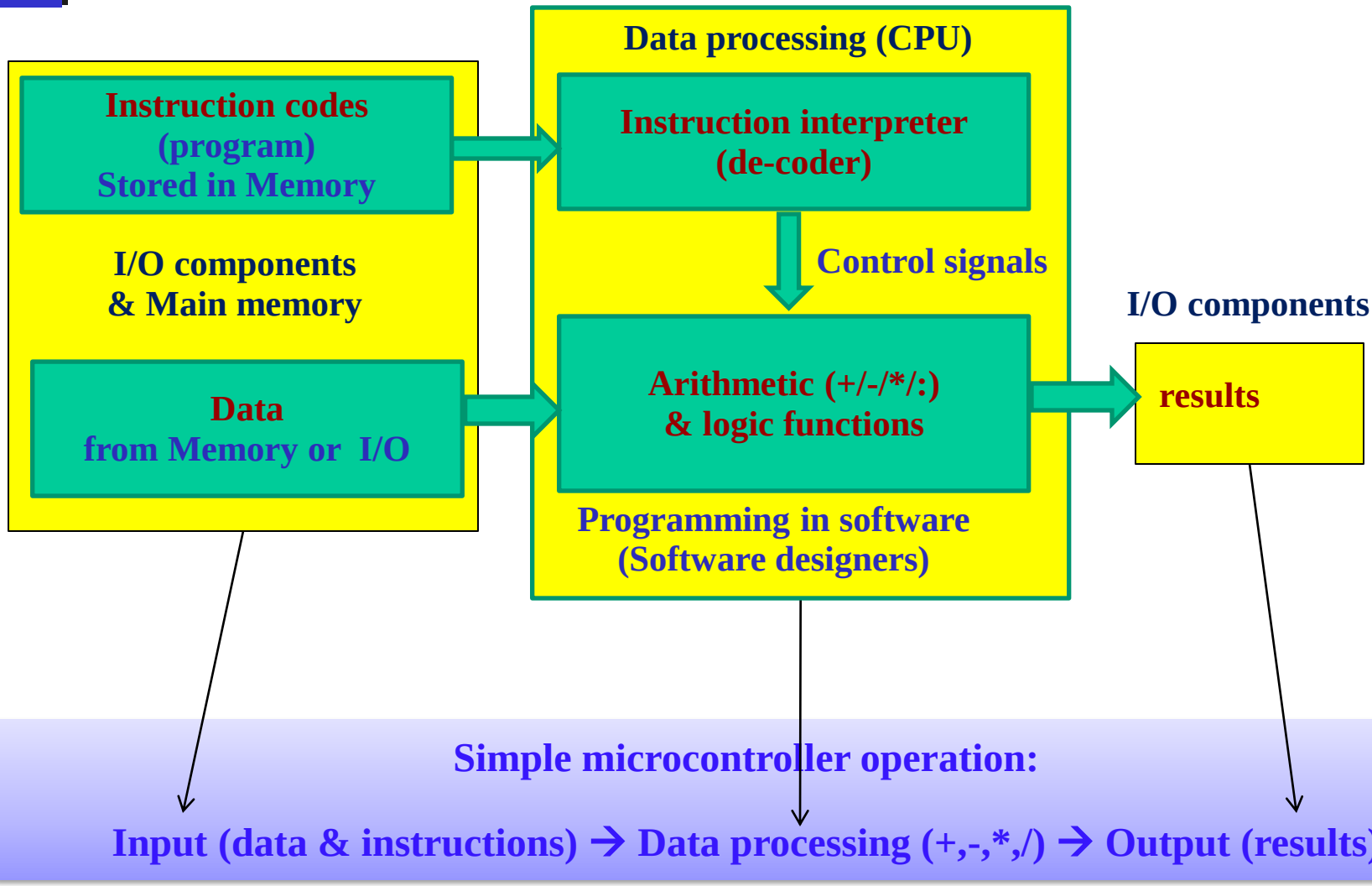




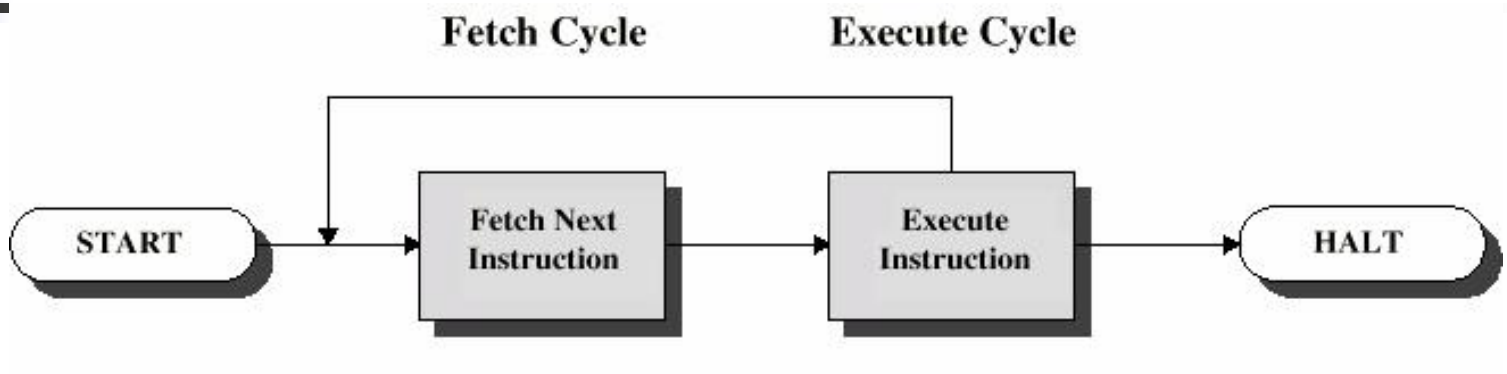




Microcontroller – instruction set



Instruction: Basic instruction cycle



The basic function performed by a computer is **execution of a program** which consists of a **set of instructions** stored in **memory**.

The processor does this work by executing instructions specified in the program. In its simplest form, instruction processing consists of two steps:

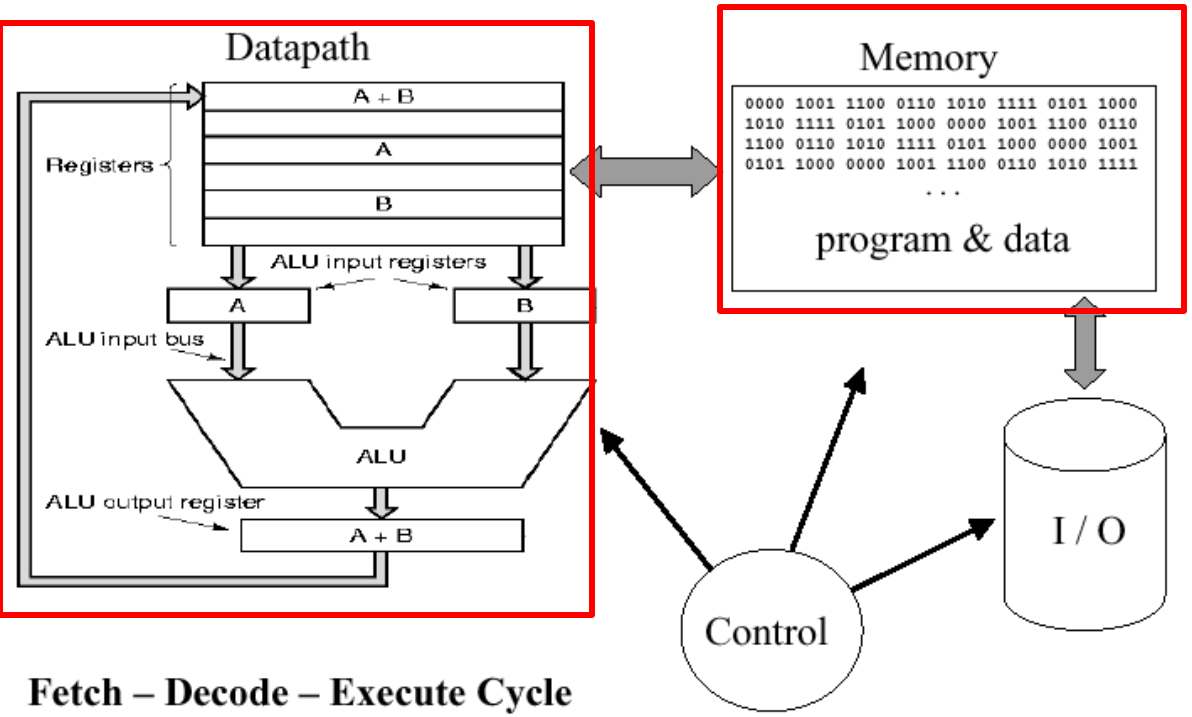
(1) Instruction fetch: the processor reads (fetches) instructions from memory one at a time.

(2) Instruction execution: may involve several operations and depends on the nature of the instruction.

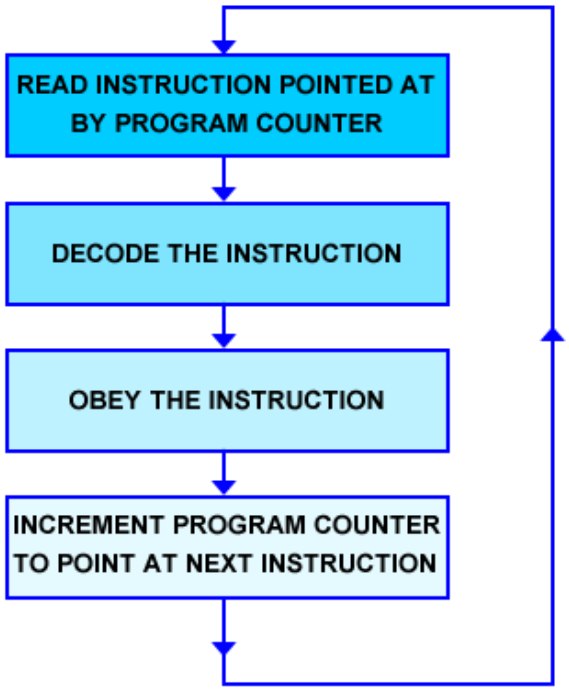
Program execution consists of repeating the process of instruction fetch and instruction execution. Program execution halts only if the machine is **turned off**, some sort of unrecoverable **error** occurs, or **a program instruction that halts** the computer is encountered.

The **processing** required for a single instruction is called an **instruction cycle**.

Microcontroller – instruction set



Fetch – Decode – Execute Cycle



fetch-decode-execute cycle.

Mops R500 Microcontroller

Features
- Big endian, 2 byte memory addressing, 1 byte opcodes
- 1024 bytes of memory

Reset

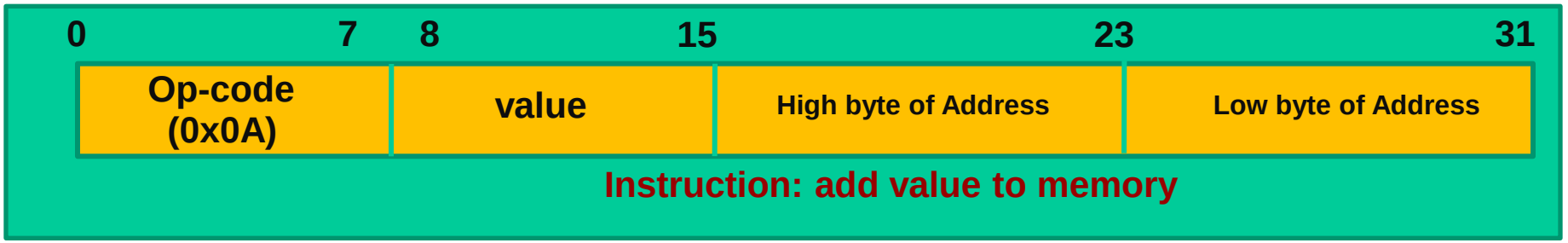
Upon reset, the microcontroller sets the program counter (PC) to location 0. The microcontroller clears all internal memory and initialises it to 0. This provides the programmer with a clean slate.

Instruction Set Summary

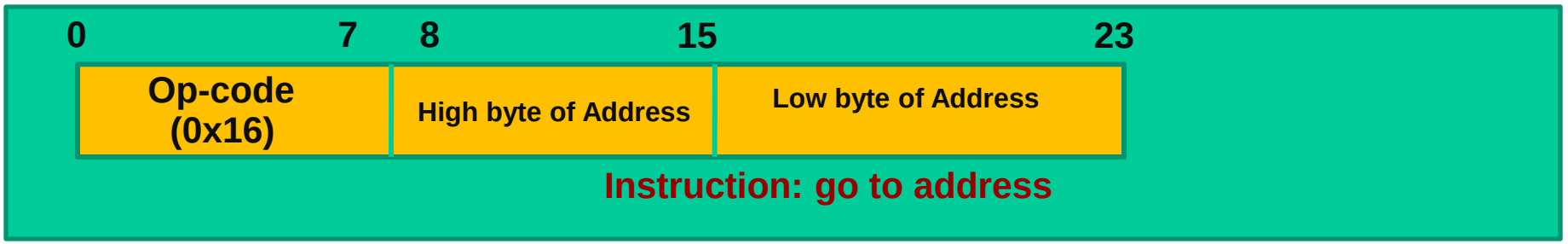
Opcode	Action
0x0A	**Add Value to Memory** The first byte after the opcode is the value that is to be *added* to memory. The second byte after the opcode is the high byte of the address, and the third byte is the low byte of the address. The memory address can be determined by (high byte << 8) | low byte. After execution, the PC points to the fourth byte after the opcode.
0x13	**Subtract Value from Memory** The format of the instruction is the same as above. The value should be *subtracted* from the memory this time.
0x16	**Go to address (always branch)** The first byte after the opcode is the high byte of the address. The second byte after the opcode is the low byte of the address. After execution, the PC points to that address.
0x17	**Branch relative** The program branches to a new location, relative to the current location. The first byte after the opcode is the value that will be added to the PC. The value must be treated as a signed value, i.e.: 128 = -128).
0xFF	Halt opcode Execution stops and the PC is not incremented.

Microcontroller – instruction set

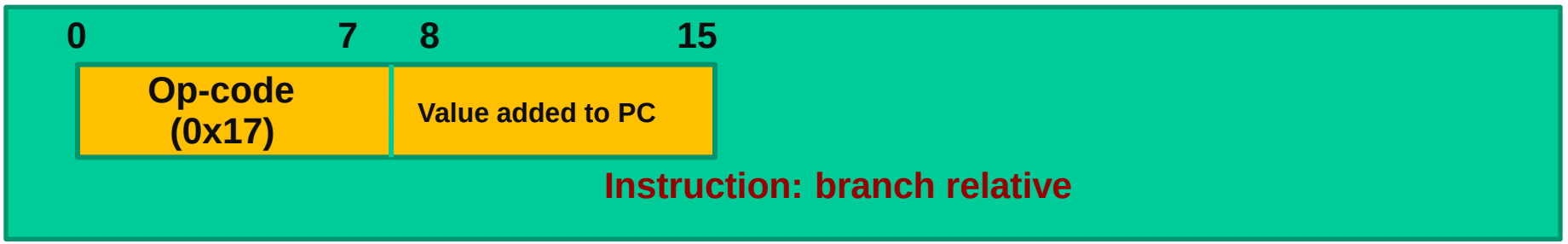
Signed Integer format



Instruction: add value to memory



Instruction: go to address



Instruction: branch relative

Microcontroller – instruction set

```
0000   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
0010   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
0020   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
...
```

Assume that: at the beginning, the PC is set to location 00x.

+ PC contains 0x1 - the address of the first instruction
+ Content of memory byte at address 0010 is 0xA1

Program 01: "Add 0x1F to the contents of memory byte at address 0010. Stop the PC"