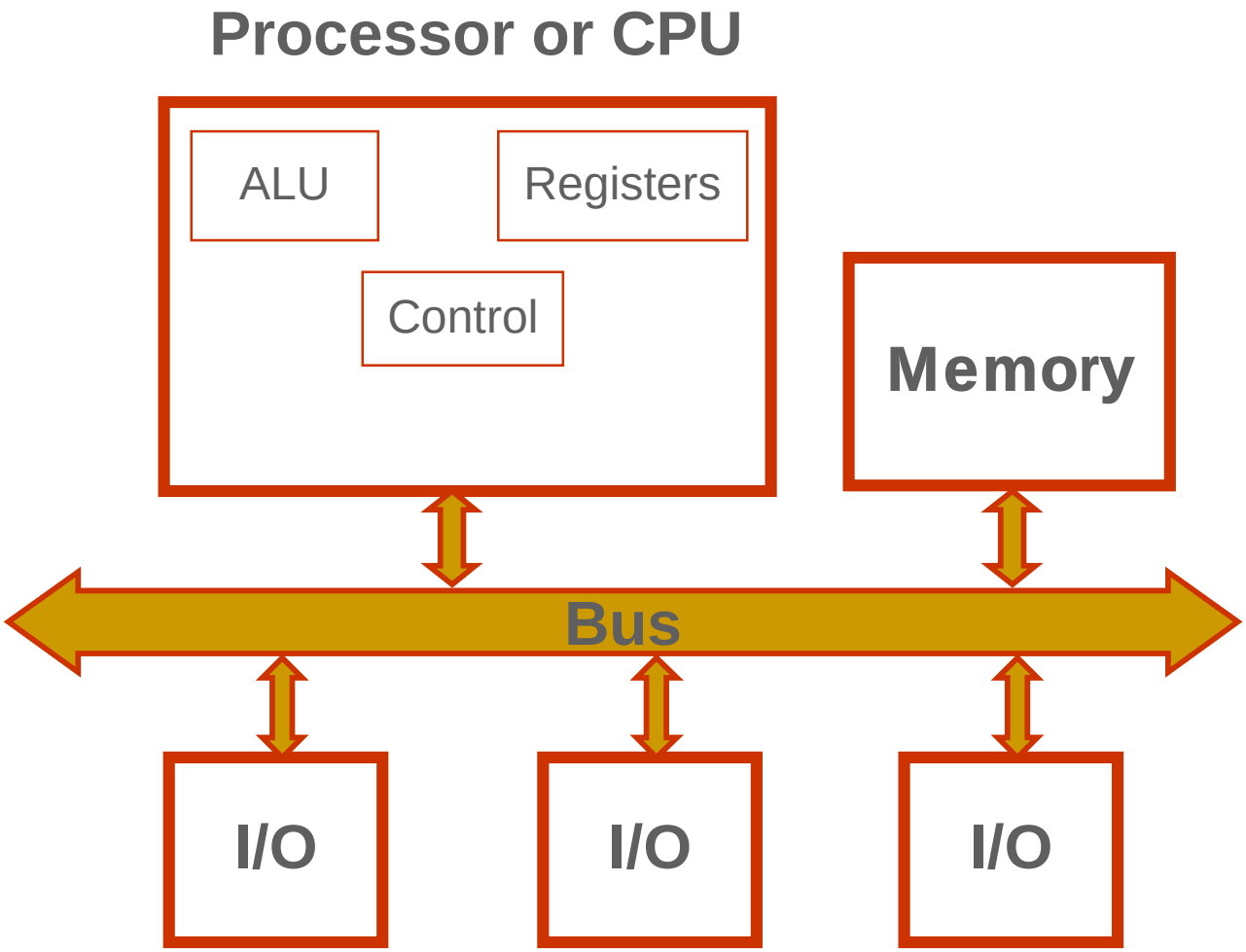
Basic Computer Organization

- Main parts of a computer system:
  - Processor: Executes programs
  - Main memory: Holds program and data
  - I/O devices: For communication with outside
- Machine instruction: Description of primitive operation that machine hardware is able to execute e.g. ADD these two integers
- Instruction Set: Complete specification of all the kinds of instructions that the processor hardware was built to execute

Basic Computer Organization



Inside the Processor…

- Control hardware: Hardware to manage instruction execution
- ALU: Arithmetic and Logical Unit (hardware to do arithmetic, logical operations)
- Registers: small units of memory to hold data/instructions temporarily during execution

Aside: About Memory

- What is memory?
  - Something that can remember things
- There are different kinds of memory in a computer system
  - Some remember by the state an electrical circuit is in e.g., SRAM
  - Others remember by the amount of electrical charge stored in a capacitor e.g., DRAM – “Memory”
  - Yet others remember by magnetic or optical properties e.g., Hard disk drive/Mag Tape, VCD/DVD
- They can vary substantially in their speed and capacity

Inside the Processor…

- Control hardware: Hardware to manage instruction execution
- ALU: Arithmetic and Logical Unit (hardware to do arithmetic, logical operations)
- Registers: small units of memory to hold data/instructions temporarily during execution
- There are 2 kinds of registers in a CPU
  1. Special purpose registers
  2. General purpose registers

Special Purpose Registers

- These are used for specific purposes by the control hardware
- Program Counter (PC): used to remember the location in memory of the instruction currently being executed
- Instruction Register (IR): used to remember that instruction
- Processor Status Register: used to remembers status information about current state of processor, e.g., whether an arithmetic overflow has occurred

General Purpose Registers

- Available for use by the programmer
- Useful for remembering frequently used data
- Why is it a good idea to do this?

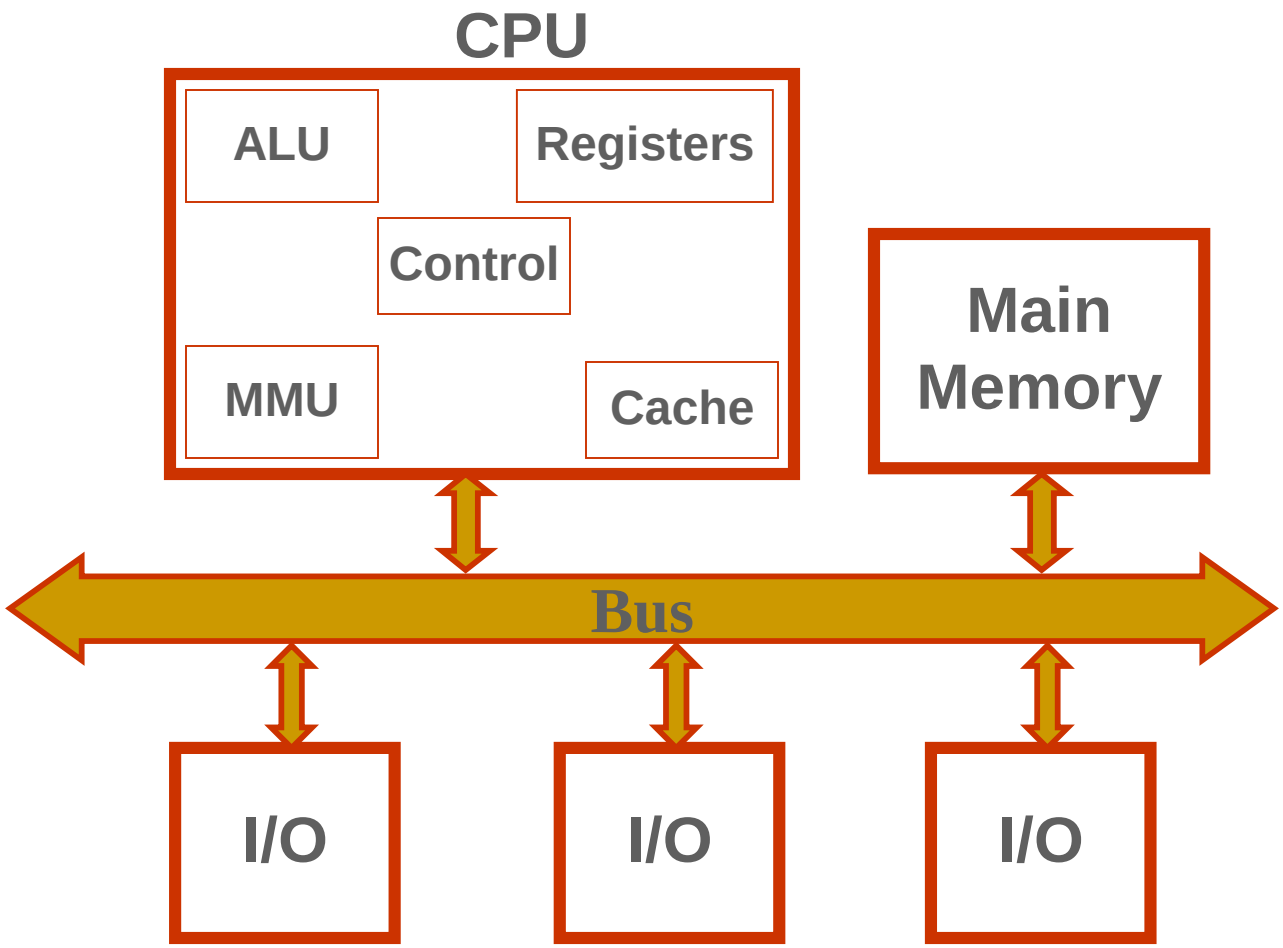
Why use General Purpose Registers?

- There is a large speed disparity between the processor (CPU) and the memory where instructions and data are stored
- Consider a 1 GHz processor
  - This frequency corresponds to a 1 nanosecond time scale
  - milli (10^{-3}), micro (10^{-6}), nano (10^{-9})
  - G (giga) 2^{30} for memory; 10^{9} for frequency, disk size
- Memory: ~ 100 nanosecond time scale
- Aside: More on nanosecond
  - Speed of light: ~ 300,000 km/sec or ~ 0.3 m/nsec

General Purpose Registers.

- Available for use by the programmer
- Useful for remembering frequently used data
- A typical processor today has 32 GPRs, say R0, R1,…, R31
- The operands to an instruction could come either from registers or from main memory

Basic Computer Organization



General Purpose Registers

- Integer Registers
- FP Registers

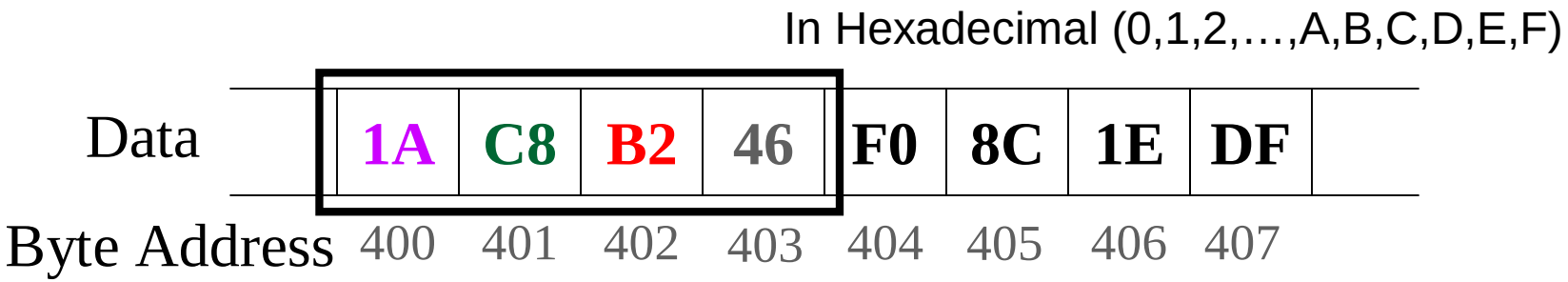
Special Purpose Registers

- PC
- IR
- Status register

Main Memory

- Holds instructions and data
- View it as a sequence of locations, each referred to by a unique **memory address**
- If the size of each memory location is 1 Byte, we call the memory **byte addressable**
- This is quite typical, as the smallest data (character) is represented in 1 Byte
- Larger data items are stored in contiguous memory locations, e.g., a 4Byte float would occupy 4 consecutive memory locations

Terms: Byte ordering

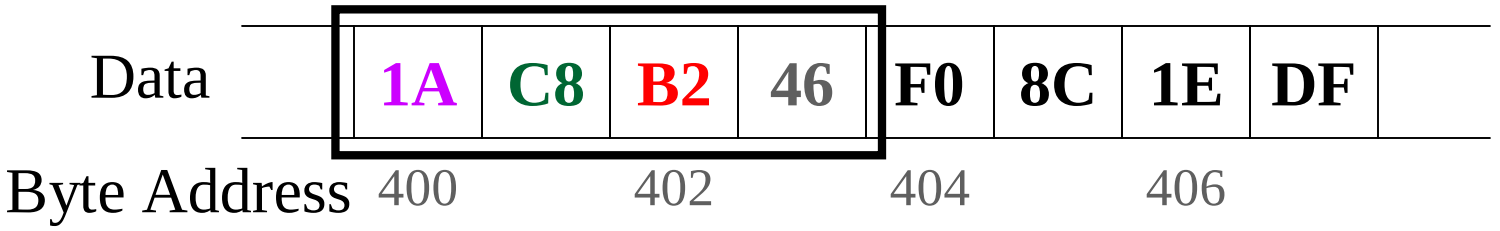


Q: Value of the integer (4 byte data) at Address 400?

A: There are a few possibilities!

Depending on how significant the bytes are

Byte ordering



Value of the integer (4 byte data) at Address 400?

Possibility 1: `1A' is the most significant byte

1 A C 8 B 2 4 6

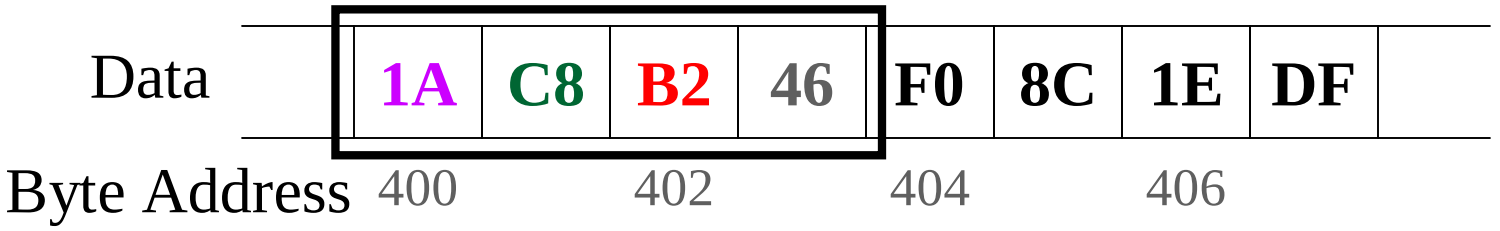
0001 1010 1100 1000 1011 0010 0100 0110

Unsigned int value: 449,360,454

$$\sum_{i=0}^{n-1} x_i 2^i$$

This convention is called Big-endian byte ordering

Byte ordering.



Value of the integer (4 byte data) at Address 400?

Possibility 2: If `46' is the most significant byte

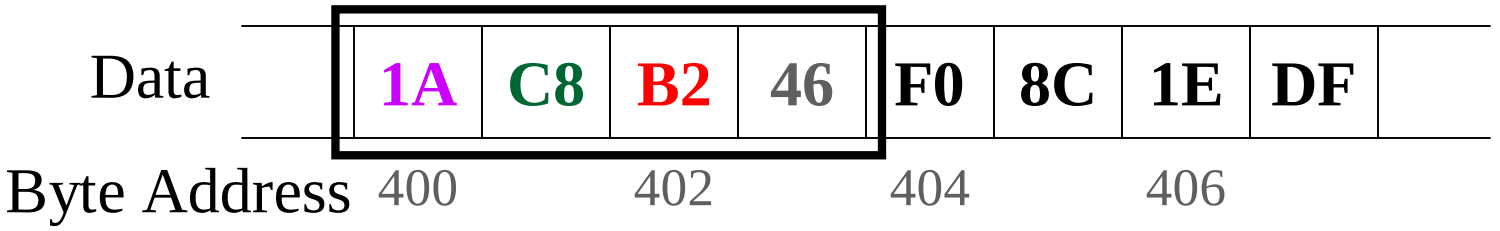
4 6 B 2 C 8 1 A

0100 0110 1011 0010 1100 1000 0001 1010

Unsigned integer value: 1,186,121,754

This convention is called Little-endian byte ordering

Byte ordering..



Value of the integer (4 byte data) at Address 400?

Big-endian ordering Little-endian ordering

449,360,454 1,186,121,754

Some machines are built to use big-endian byte ordering and others are designed to use little-endian byte ordering

This can be relevant to the programmer