CPU Organisation

Table of contents

Computer organisation 1
Parts of a CPU 1
Registers 2
See also 2

Where the digital logic of the first half of the course ends up: the blocks a processor is made of, and what each one holds. See arithmetic-logic-unit[1] for the ALU in detail and fetch-decode-execute-cycle[2] for what the control unit does with all of it.

Computer organisation

A simplified view of a whole machine: a **CPU** connected by **external buses** to main memory, secondary storage and IO devices, with **internal buses** wiring together the CPU's own blocks.

Main memory holds both the program and the data: the instructions to be fetched and the values to be operated on.

Parts of a CPU

Block	Role
Control Unit	Fetches instructions from memory, and makes the ALU and the registers perform the instruction
ALU	Performs arithmetic and logical operations

[1] courses/csse2010/arithmetic-logic-unit.pdf
[2] courses/csse2010/fetch-decode-execute-cycle.pdf

Block	Role
Registers	High-speed memory — stores temporary results and control information

How many registers, and how wide, varies enormously with the machine: a small CPU may have 8 × 8-bit registers, a large one 64 × 32-bit registers. The AVR used later in this course has 32 registers (R0–R31) and an 8-bit word, so it also has an **8-bit ALU**.

Registers

Registers are **temporary storage locations inside the CPU**. Two types:

General purpose registers hold the data to be operated on (e.g. data read from memory) and the results of operations. Collectively they are the **register file**, and **their width is the CPU word size**.

Special purpose registers are used by the system rather than by the program's arithmetic:

Register	Holds
Program Counter (PC)	the address of the **next** instruction to be fetched from memory
Instruction Register (IR)	the **current** instruction

A third special purpose register, the **status register**, records what the last ALU operation did — see fetch-decode-execute-cycle[3]. PC and IR are the two the fetch step manipulates directly.

See also

- arithmetic-logic-unit[4] · fetch-decode-execute-cycle[5]
- binary-number-representations[6] — word sizes and two's complement, which fix what the registers can hold
- avr-architecture[7] — the same three blocks as an actual chip
- memory-hierarchy-and-cache[8] — the registers at the top of the memory hierarchy, and the cache between the CPU and main memory

[3] courses/csse2010/fetch-decode-execute-cycle.pdf
[4] courses/csse2010/arithmetic-logic-unit.pdf
[5] courses/csse2010/fetch-decode-execute-cycle.pdf
[6] courses/csse2010/binary-number-representations.pdf
[7] courses/csse2010/avr-architecture.pdf
[8] courses/csse2010/memory-hierarchy-and-cache.pdf

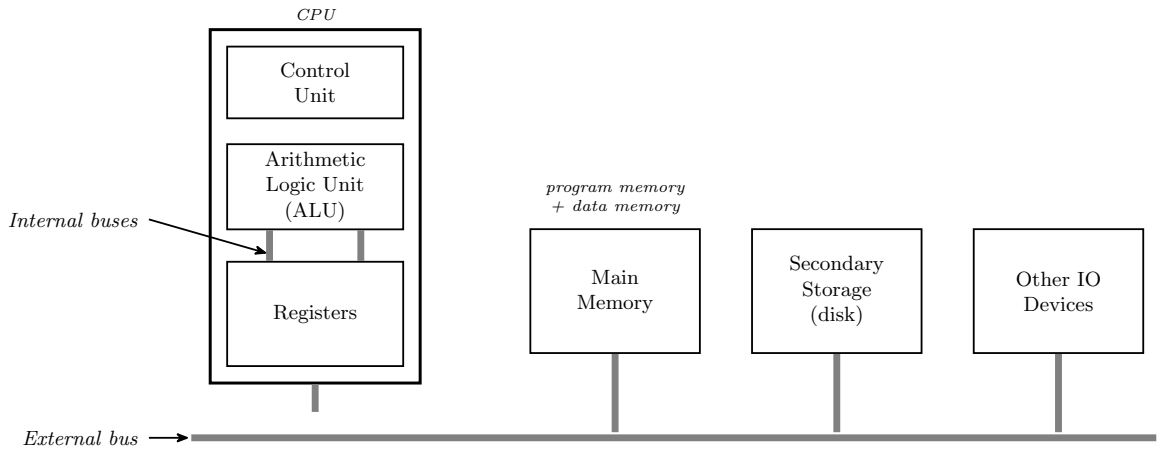


Figure 1: Simplified computer organisation. The CPU's three blocks are joined by internal buses; a single external bus carries everything between the CPU and the rest of the machine. Main memory holds both the program and the data operated on, so it is both program memory and data memory.

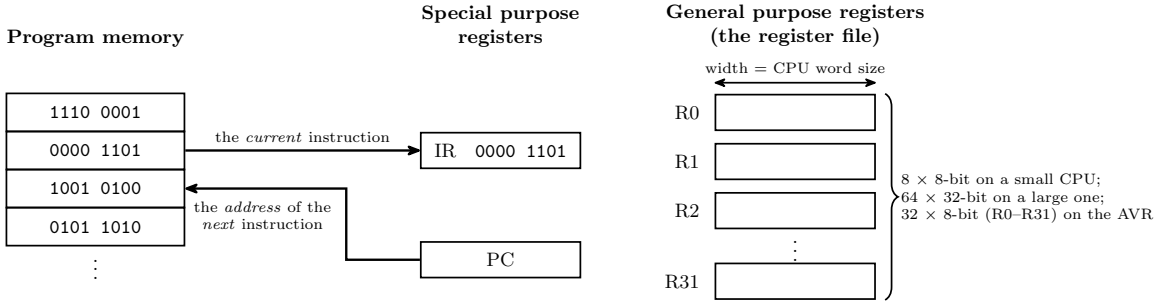


Figure 2: The two types of register, and why the distinction matters. A general purpose register is one anonymous row of the file, as wide as the machine word and picked by number; a special purpose register is named, and what it holds is interpreted — the PC as an address into program memory, the IR as an instruction to be decoded. The particular bit patterns shown in memory are illustrative only.

- 2026-08-24-alu-and-control-unit[9] — Lecture 9

[9] courses/csse2010/2026-08-24-alu-and-control-unit.pdf