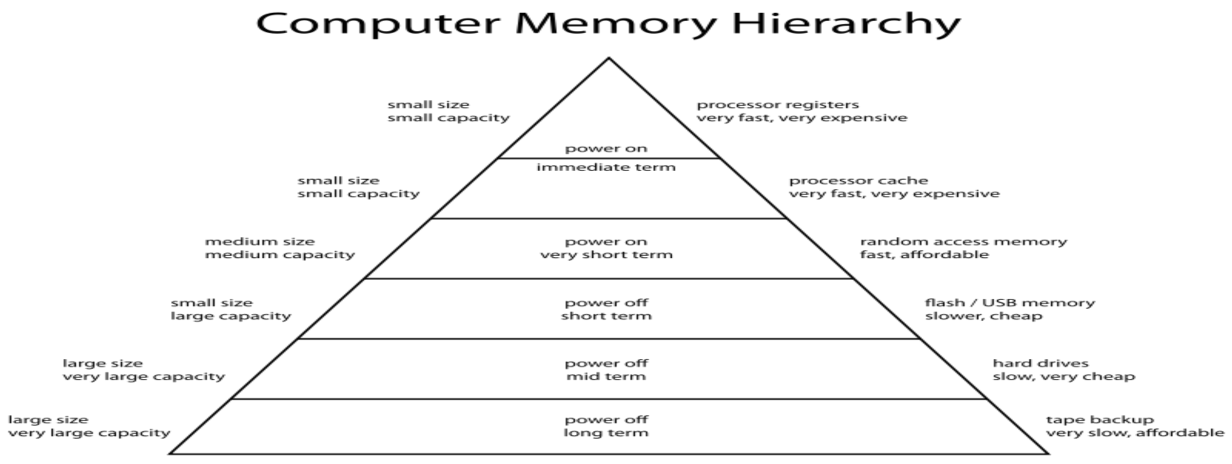
Definition: It is a special, high-speed storage area within the CPU. All data must be represented in a register before it can be processed. For example, if two numbers are to be multiplied, both numbers must be in registers, and the result is also placed in a register. (The register can contain the address of a memory location where data is stored rather than the actual data itself.)

The number of registers that a CPU has and the size of each (number of bits) help determine the power and speed of a CPU. For example a 32-bit CPU is one in which each register is 32 bits wide. Therefore, each CPU instruction can manipulate 32 bits of data.

Usually, the movement of data in and out of registers is completely transparent to users, and even to programmers. Only assembly language programs can manipulate registers. In high-level languages, the compiler is responsible for translating high-level operations into low-level operations that access registers.

In computer architecture, a **processor register** is a small amount of storage available as part of a CPU or other digital processor. Such registers are (typically) addressed by mechanisms other than main memory and can be accessed more quickly. Almost all computers, load-store architecture or not, load data from a larger memory into registers where it is used for arithmetic, manipulated, or tested, by some machine instruction. Manipulated data is then often stored back in main memory, either by the same instruction or a subsequent one. Processor registers are normally at the top of the memory hierarchy, and provide the fastest way to access data.

A "memory hierarchy" in computer storage distinguishes each level in the "hierarchy" by response time. Since response time, complexity, and capacity are related, the levels may also be distinguished by the controlling technology.



Categories of registers

Registers are normally measured by the number of bits they can hold, for example, an "8-bit register" or a "32-bit register". A processor often contains several kinds of registers, that can be classified accordingly to their content or instructions that operate on them:

User-accessible registers – The most common division of user-accessible registers is into data registers and address registers.
Data registers can hold numeric values such as integer and floating-point values, as well as characters, small bit arrays and other data. In some older and low end CPUs, a special data register, known as the accumulator, is used implicitly for many operations.
Address registers hold addresses and are used by instructions that indirectly access primary memory.

- Some processors contain registers that may only be used to hold an address or only to hold numeric values (in some cases used as an index register whose value is added as an offset from some address); others allow registers to hold either kind of quantity. A wide variety of possible addressing modes, used to specify the effective address of an operand, exist.
- The stack pointer is used to manage the run-time stack. Rarely, other data stacks are addressed by dedicated address registers, see stack machine.

Conditional registers hold truth values often used to determine whether some instruction should or should not be executed.
General purpose registers (**GPR**s) can store both data and addresses, i.e., they are combined Data/Address registers and rarely the register file is **unified** to include floating point as well.
Floating point registers (**FPR**s) store floating point numbers in many architectures.
Constant registers hold read-only values such as zero, one, or pi.
Vector registers hold data for vector processing done by SIMD instructions (Single Instruction, Multiple Data).
Special purpose registers (**SPR**s) hold program state; they usually include the program counter (aka instruction pointer) and status register (aka processor status word). The aforementioned stack pointer is sometimes also included in this group. Embedded microprocessors can also have registers corresponding to specialized hardware elements.

- **Instruction registers** store the instruction currently being executed.

In some architectures, **model-specific registers** (also called *machine-specific registers*) store data and settings related to the processor itself. Because their meanings are attached to the design of a specific processor, they cannot be expected to remain standard between processor generations.
Control and status registers – There are three types: program counter, instruction registers and program status word (PSW).
Registers related to fetching information from RAM, a collection of storage registers located on separate chips from the CPU (unlike most of the above, these are generally not *architectural* registers):

- Memory buffer register (MBR)
- Memory data register (MDR)
- Memory address register (MAR)
- Memory Type Range Registers (MTRR)

Hardware registers are similar, but occur outside CPUs.