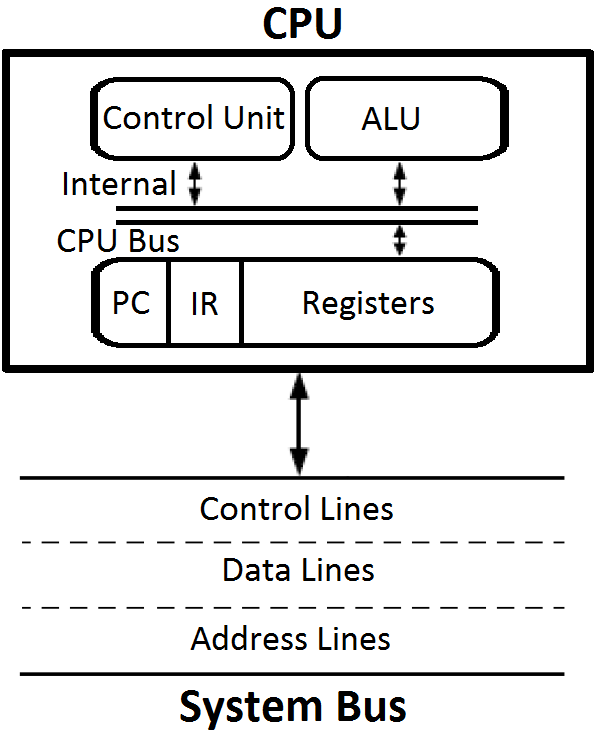
**THE CPU**



* Stores Intermediate e data during execution of instruction
* Performs the required microoperations for executing the instructions
* Supervises the transfer of info among the registers and instructs the ALU as to which operation to perform

**CPU FUNCTIONS**

* CPU functions depend on computers instruction set.
* Computer Architecture = Computer Structure and Behavior as seen by the low level programmer
* Instruction formats
* Addressing modes
* The instruction set
* The general organization of CPU
* The instruction set provides the specifications for the CPU design and this know how is essential to be able to program efficiently.

**REGISTER ORGANIZATION**

* Memory access is the most time consuming operation in a computer
* More efficient to use registers connected by a bus system.
* The set of registers within the CPU represents the top level of the memory hierarchy inside the computer system
* User visible registers: can be accessed by assembly language programmers.
* Control and Status registers: used by the Control Unit to control the operation of the CPU; not directly accessible by the programmer.

**USER VISIBLE REGISTERS**

* Some architecture provides a set of registers which can be used without restrictions as operands for any opcode and as address registers; these are so called general-purpose registers.
* Often the architecture creates a separation between:

***Data registers***

Data registers can be used to hold only data. Some architecture impose restrictions to the use of data registers: for example there can be disjoint sets of registers for integer and for floating point computation.

***Address registers***

Address registers used only for address representation and computation: for example base registers, index registers, stack pointer, etc. In some architecture address registers can be specialized for some of the previous functions.

**SOME TRADE OFFS**

* A large number of general purpose registers means large number of bits for encoding register operands; specialization of registers reduces this need.
* Too small number of registers creates problems to the programmer and leads to an increased memory traffic.
* The number of general-purpose or data registers is often between 8 - 32.
* RISC processors often have a very large number of registers (~ 100).

**CONTROL AND STATUS REGISTERS**

|  |  |  |
| --- | --- | --- |
| Program Counter | PC | holds the address of the instruction to be fetched |
| Instruction Register | IR | holds the last instruction fetched |
| Memory Address Register | MAR | holds the address of a memory location that is to be read or written |
| Memory Buffer Register | MBR | holds the data to be written to memory or the data most recently read |
| Program Status Word | PSW | Condition Code Flags + other bits defining the status of the CPU (interrupt enabled/disabled, supervisor, etc.) |

**GENERAL REGISTER ORGANIZATION**



**Timing of Control Word**

* The 4 control selection variables are generated in the control unit
* They must be available at the beginning of a clock cycle
* The data from the two source registers propagates through the gates in the MUX and the ALU, to the O/p bus, and into the i/p of the destination register, all during the clock cycle interval.
* During the next clock transition, the information is transferred from bus to R1.