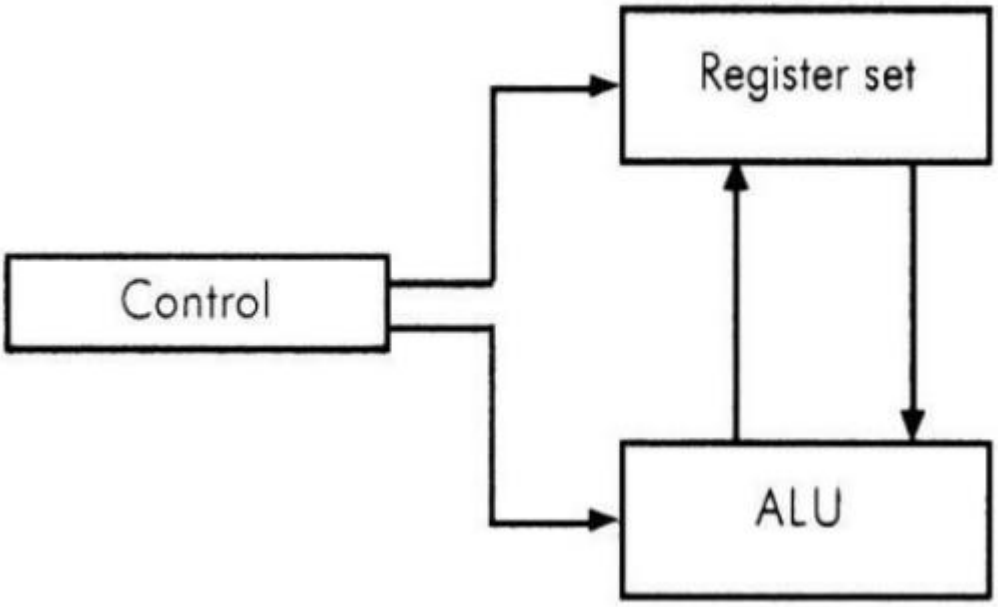
General Register Organisation: ·

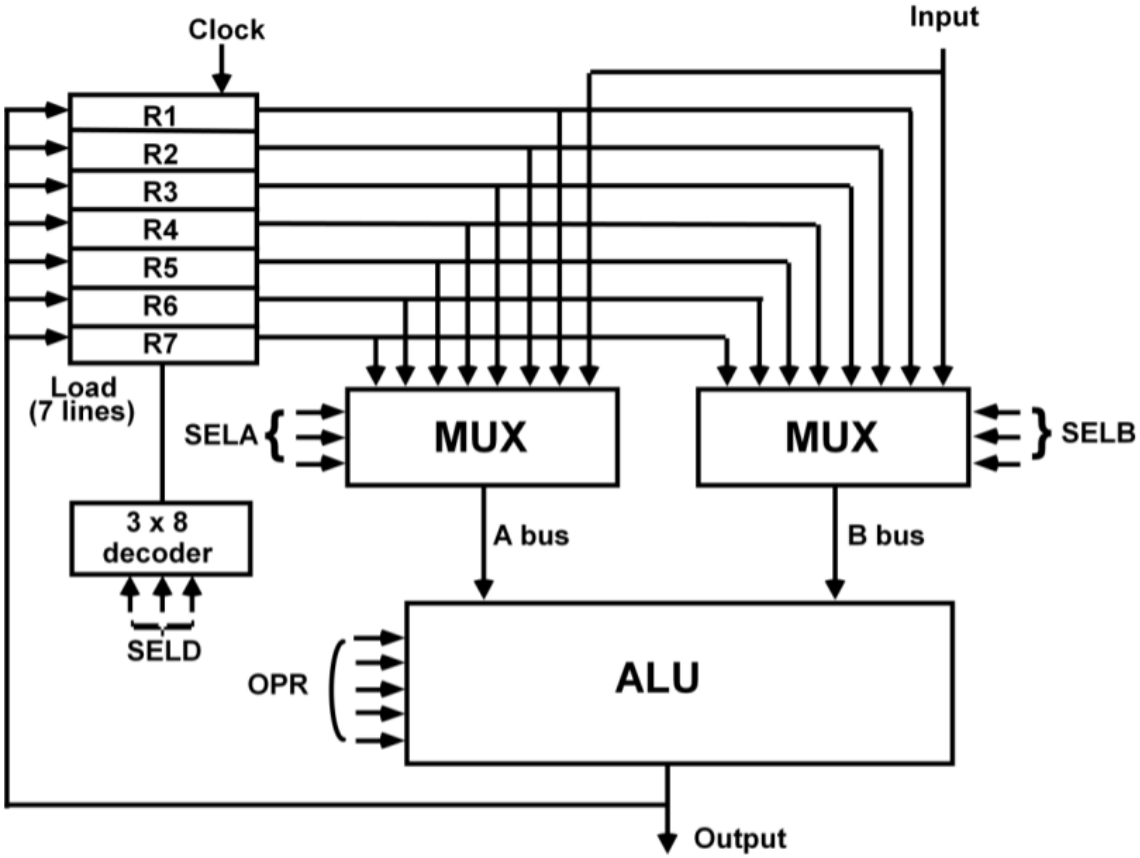
The Central Processing Unit (CPU) is called the brain of the computer that performs data processing operations. Below figure shows the three major parts of CPU.



Intermediate data is stored in the register set during the execution of the instructions. The microoperations required for executing the instructions are performed by the arithmetic logic unit whereas the control unit takes care of transfer of information among the registers and guides the ALU. The control unit services the transfer of information among the registers and instructs the ALU about which operation is to be performed. The computer instruction set is meant for providing the specifications for the design of the CPU. The design of the CPU largely, involves choosing the hardware for implementing the machine instructions.

The need for memory locations arises for storing pointers, counters, return address, temporary results and partial products. Memory access consumes the most of the time off an operation in a computer. It is more convenient and more efficient to store these intermediate values in processor registers.

A common bus system is employed to contact registers that are included in the CPU in a large number. Communications between registers is not only for direct data transfer but also for performing various micro-operations. A bus organization for such CPU register shown below figure, is connected to two multiplexers (MUX) to form two buses A and B. The selected lines in each multiplexers select one register of the input data for the particular bus.



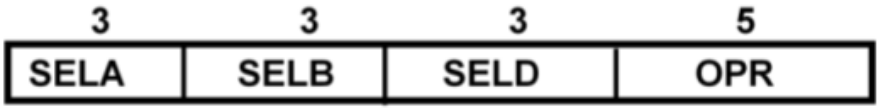
OPERATION OF CONTROL UNIT: The control unit directs the information flow through ALU by: -

- Selecting various Components in the system
- Selecting the Function of ALU

Example: R1 <- R2 + R3

1. MUX A selector (SELA): BUS A ← R2
2. MUX B selector (SELB): BUS B ← R3
3. ALU operation selector (OPR): ALU to ADD
4. De-MUX destination selector (SELD): R1 ← Out Bus

Control Word:



Encoding of register selection fields

Binary Code	**SEL A**	**SEL B**	**SEL D**
000	R0(Input)	R0(Input)	R0(Input)
001	R1	R1	R1
010	R2	R2	R2
011	R3	R3	R3
100	R4	R4	R4
101	R5	R5	R5
110	R6	R6	R6
111	R7	R7	R7

Encoding of ALU operations

OPR Select	Operations	Symbol
000	OR A and B	OR
001	Increment A	INCA
010	Increment B	INCB
011	ADD A + B	ADD
100	Subtract A-B	SUB
101	Decrement A	DECA
110	Shift Left A	SHRA
111	Shift Right A	SHLA

Circuit: