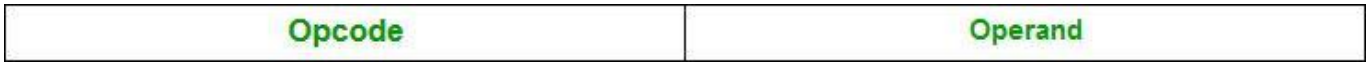
Addressing Modes

Addressing modes are techniques used by the CPU to identify the location of the operand(s) needed for executing an instruction. They provide rules for interpreting the address field in an instruction, helping the CPU fetch operands correctly.

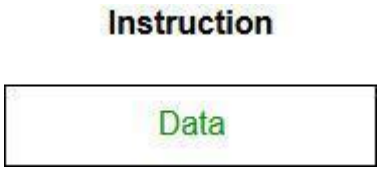
- **Opcode** – Tells the CPU what operation to perform (e.g., ADD, MOV).
- **Operands** – The data or addresses on which the operation is performed.



Addressing Modes Types

Implicit (Implied) Addressing

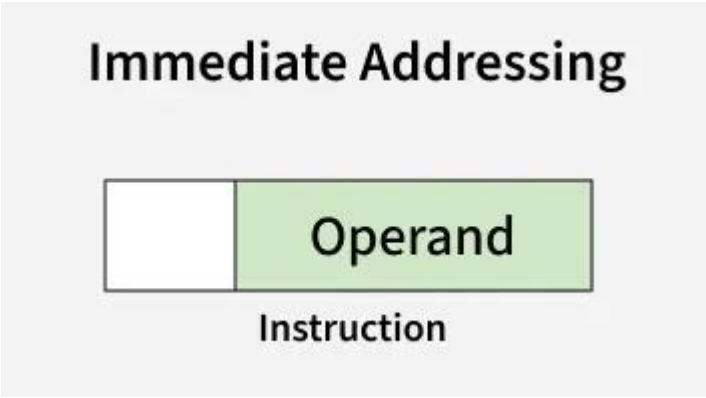
The instruction does not mention the operand directly. The CPU knows what to use from the instruction itself, usually a special register like the accumulator or the stack.



It is used for special instructions or control commands like CLA, PUSH, and RET, where the operand is automatically known from the instruction itself

Immediate Addressing

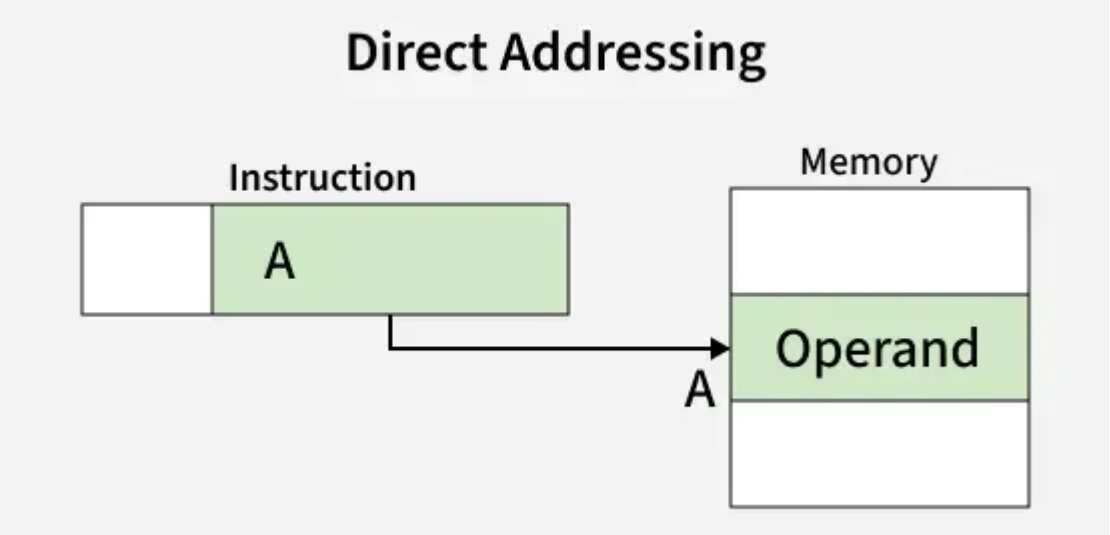
The operand is the part of the instruction itself. It is used when the value is known while writing the program.



Example: MOV R1, #5 moves the value 5 into register R1, where #5 is the immediate value.

Direct Addressing

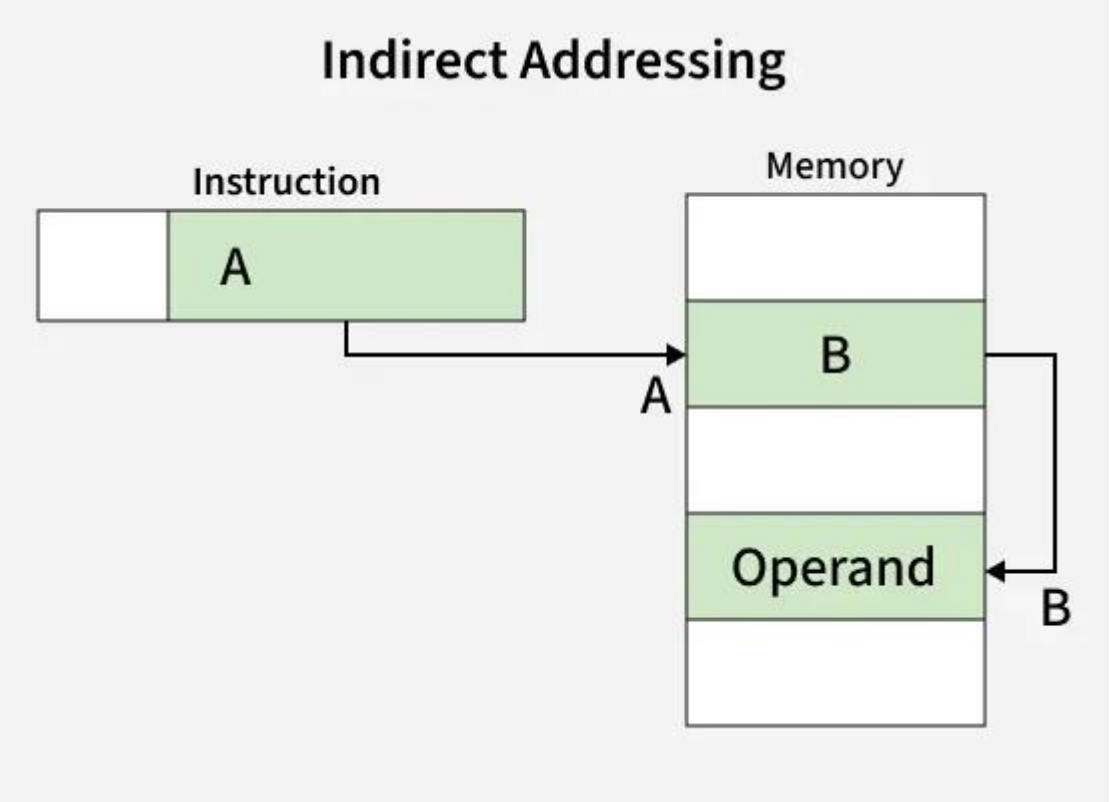
The instruction contains the memory address of the operand. The CPU accesses the data directly from that address.



Example: LOAD R1, 1000 loads data from memory address 1000 into register R1.

Indirect Addressing

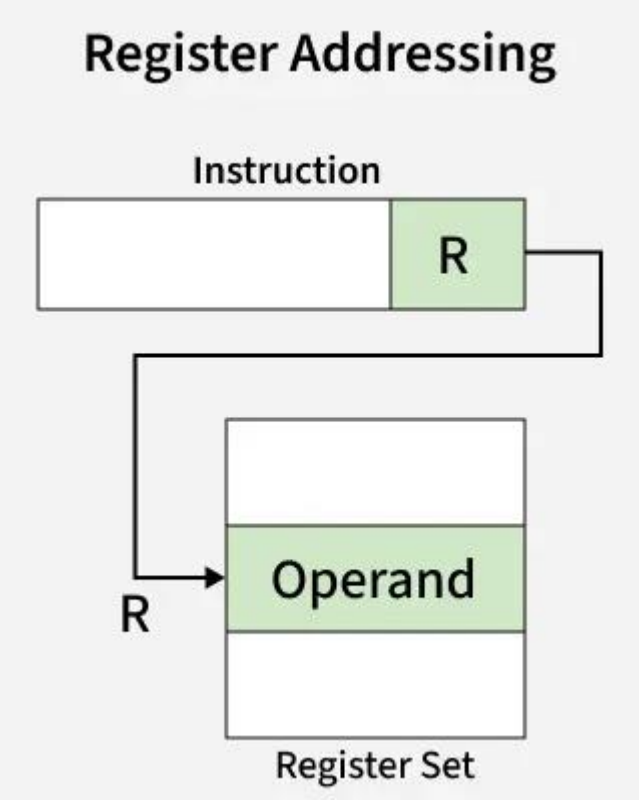
The instruction contains the address of a register or memory location that holds the actual address of the operand. The CPU first fetches this address, then accesses the operand.



Example: LOAD R1, (R2) loads data from the memory location whose address is in register R2.

Register Addressing

The operand is located in a CPU register specified by the instruction.



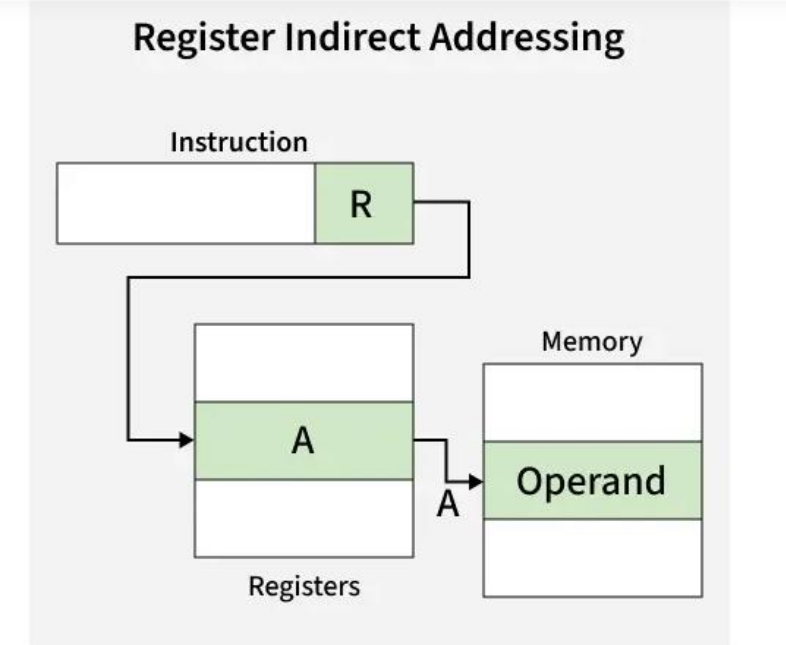
Step:

- The instruction specifies a register (R).
- The CPU takes operand directly from register R.

Example: MOV A, B operates between registers A and B.

Register Indirect Addressing

The register specified in the instruction contains the memory address of the operand.



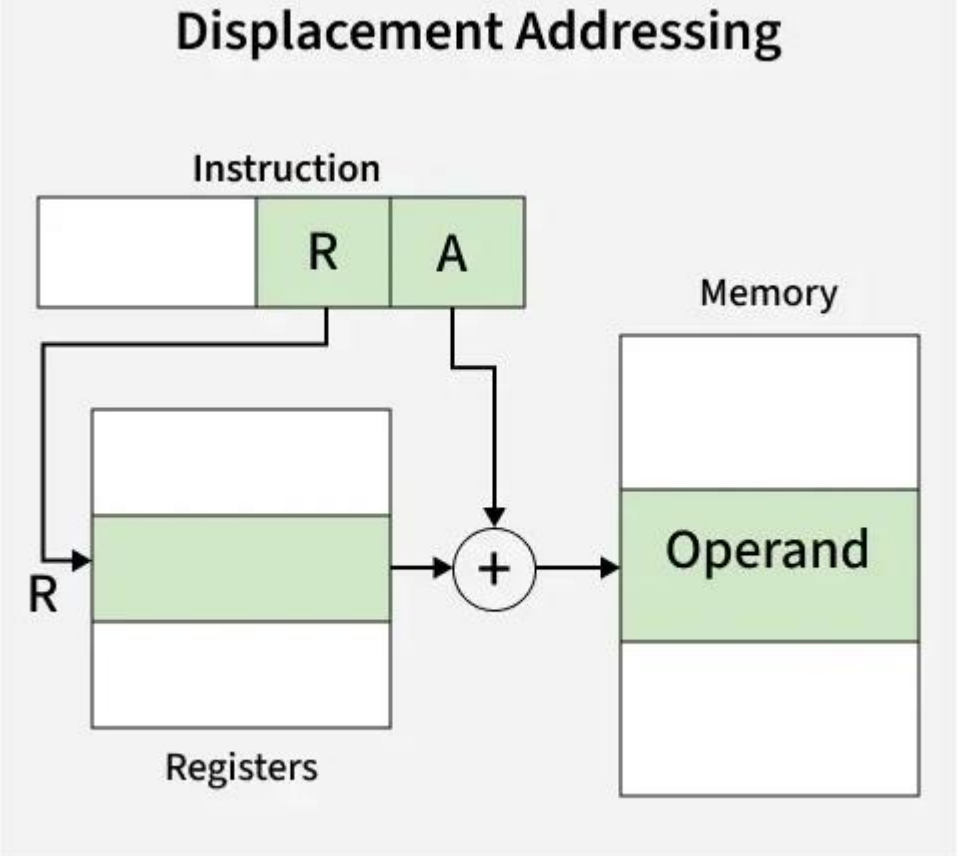
Step:

- The instruction specifies a register.
- This register holds the address (A).
- The CPU fetches the operand from memory location A.

Example: MOV A, [R1] uses content of R1 as memory address.

Displacement Addressing (Indexed, Base-Register, Relative)

The operand's effective address is calculated by adding a constant value (displacement) to the contents of one or more registers.



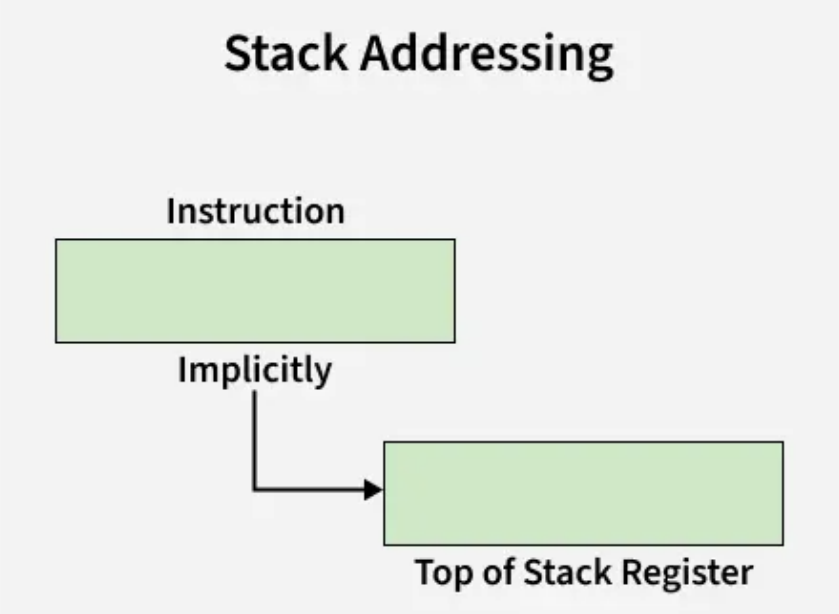
Step:

- The instruction provides a base register (R) and an address part (A).
- CPU adds the value of R and A to get the effective operand address.
- Operand is fetched from the calculated address in memory.

Example: Used for arrays, accessing an element at a position relative to a base.

Stack Addressing

The operand is implicitly taken from the top of the stack, without being mentioned in the instruction.



Step:

- Operation is performed using the value at the stack's top (implied by instruction).
- No need for explicit operand field; CPU refers to stack pointer register by default.

Example: POP and PUSH operations.