Addressing Modes in RISC-V.

1) Register Addressing

* Definition → Both source operands and destinations are in registers.
* The instruction specifies register numbers in its fields. No memory access is needed except for fetching the instruction itself.

ex :- add x5, x6, x7 # x5 = x6 + x7

* Fast execution since it avoids memory access.

2) Immediate Addressing.

* Definition → One operand is constant embedded in the instruction other comes from a register
* The constant is sign extended before use.

ex :- addi x5, x6, 10 # x5 = x6 + 10

* No extra memory fetch for the constant value

3) Base + offset Addressing

* Definition → the effective address is calculated as.

E.A = (B.R) + S.O

* The B.R. typically holds a pointer to memory and the offset is encoded in the instruction.

ex :- lw x5, 8(x6) # Load word from address x6+8 to x5
sw x5, 12(x6) # store word from x5 into address x6+12

* Efficient for accessing near a known pointer

4) PC-Relative Addressing

* Definition → The E.A is obtained by adding a signed offset to the current program counter.
* The offset is usually word aligned and encoded within the instruction.

ex :- beq x5, x6, label # Branch to label if x5 = x6
auipc x5, lw # x5 = pc + (lw << 12)
jal x1, label # Jump to label and save return address in x1

* Code can run from any memory location without absolute addresses

5) Register Indirect Addressing

* Definition → E.A is taken directly from the contents of a register (offset = 0)
* Register contains the exact memory addresses to be accessed.

ex :- lw x5, 0(x6) # Load word from address in x6 to x5

* useful when addresses are computed at runtime