Revision Notes on Addressing Modes

1. Definition

Addressing modes are the different ways in which the location of an operand is specified in an instruction. They define how the CPU should interpret the data or memory address field of an instruction.

2. Types of Addressing Modes

Immediate Addressing

Operand is directly specified in the instruction.

Example: MOV A, #5 (Move value 5 into register A).

Advantage: Fast execution.

Disadvantage: Limited operand size.

Direct Addressing

Instruction specifies the memory address where the operand is stored.

Example: MOV A, 5000 (Move content of memory address 5000 to register A).

Advantage: Simple to understand.

Disadvantage: Memory address is fixed.

Indirect Addressing

Instruction specifies a register/memory location that contains the effective address of the operand.

Example: MOV A, (R1).

Advantage: Increases flexibility.

Disadvantage: Slower (extra memory reference required).

Register Addressing

Operand is located in a CPU register.

Example: MOV A, R1.

Advantage: Very fast (no memory access needed).

Disadvantage: Limited by the number of CPU registers.

Register Indirect Addressing

Register contains the memory address of the operand.

Example: MOV A, (R2).

Advantage: Useful for accessing arrays and pointers.

Disadvantage: Requires extra memory reference.

Auto-Increment & Auto-Decrement Addressing

Auto-Increment: Operand is accessed from the address in a register, then register is automatically incremented.
Example: MOV A, (R1)+

Auto-Decrement: Operand is accessed from the address in a register, then register is automatically decremented.
Example: MOV A, -(R1).

Advantage: Useful for accessing arrays/lists sequentially.
Disadvantage: More complex hardware.

Relative Addressing

Effective address = Program Counter (PC) + Offset.

Example: JMP LABEL.

Advantage: Useful for branching in programs.

Disadvantage: Limited range of addressing.

Base and Index Addressing

Base Addressing: Effective address = Base Register + Displacement.

Index Addressing: Effective address = Index Register + Displacement.

Base + Index (Combined): Effective address = Base Register + Index Register + Displacement.

Use: Accessing arrays, tables, and structures.

3. Summary Table

Mode	Description	Example	Use
Immediate	Operand in instruction	MOV A, #5	Constants
Direct	Address given directly	MOV A, 5000	Fixed data access

Indirect	Address field points to operand address	MOV A, (5000)	Flexible addressing
Register	Operand in register	MOV A, R1	Fast operations
Register Indirect	Register holds memory address	MOV A, (R1)	Arrays/pointers
Auto-Inc/Auto-Dec	Register auto modified after/before access	(R1)+ , -(R1)	Stack, sequential data
Relative	PC + displacement	JMP LABEL	Branching/jumps
Base/Index	Base + Index + displacement	MOV A, (BX+SI)	Arrays, tables, structs