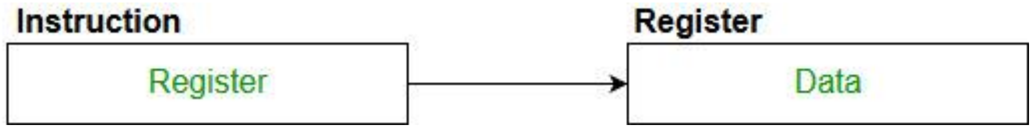
**Addressing Modes** – The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

**Various Addressing Modes:**

- **Immediate addressing mode:** In this mode data is present in address field of instruction using # symbol. It is designed like one address instruction format.

  Example: MOVE #200, R1

- **Register mode:** In register addressing the operand is placed in one of 8 bit or 16 bit general purpose registers. The data is in the register that is specified by the instruction.



- **Auto Indexed (increment mode):** Effective address of the operand is the contents of a register specified in the instruction. After accessing the operand, the content of this register are automatically incremented to point to the next consecutive memory location ((R1) +). This mode is useful for stepping through arrays in a loop.

```
Add R1, (R2) +
R1 = R1 + M [R2]
R2 = R2 + d

R2 – Start of the array
d – Size of an element
```

**Auto indexed (Decrement mode):** Effective address of the operand is the contents of a register specified in the instruction. Before accessing the operand, the contents of this register are automatically decremented to point to the previous consecutive memory location (– (R1)).

Auto decrement mode is similar to auto increment mode. Both can also be used to implement a stack as push and pop operations. Auto increment and auto decrement modes are useful for implementing “Last In First Out” (LIFO) data structures.

```
Add R1, - (R2)
R2 = R2 - d
R1 = R1 + M [R2]

R2 – Start of the array
d – Size of an element
```

- **Direct addressing/ Absolute addressing Mode:** The operand’s offset is given in the instruction as an 8 bit or 16 bit displacement element. In this addressing mode the 16 bit effective address of the data is the part of the instruction.

```
        ADD AL, [0301]
//add the contents of offset address 0301 to AL
```

- **Indexed addressing mode:** The operand’s offset is the sum of the content of an index register SI or DI and an 8 bit or 16 bit displacement.

```
        MOV AX, [SI + 5]
```

- **Based Indexed Addressing:** The operand’s offset is sum of the content of a base register BX or BP and index register SI or DI.

```
        ADD AX, [BX + SI]
```

**Advantages of Addressing Modes:**

- To give programmers to facilities such as Pointers, counters for loop controls, indexing of data and program relocation.
- To reduce the number bits in the addressing field of the Instruction.