1.3 ADDRESSING AND ADDRESSING MODES

Each instruction of a computer specifies an operation on certain data.

The different ways in which the location of an operand is specified in an instruction is called as

Different operands will use different addressing modes. One or more bits in the instruction format can be used as mode field. The value of the mode field determines which addressing mode is to be used. The effective address will be either main memory address of a register.

The most common addressing modes are:

1. Immediate addressing mode
2. Direct addressing mode
3. Indirect addressing mode
4. Register addressing mode
5. Register indirect addressing mode
6. Displacement addressing mode
7. Stack addressing mode

Immediate Addressing:

- This is the simplest form of addressing. Here, the operand is given in the instruction.
- This mode is used to define constant or set initial values of variables.
- The advantage of this mode is that no memory reference other than instruction fetch is required to obtain operand.
- The disadvantage is that the size of the number is limited to the size of the address field because most instruction sets is small compared to word length.
- **Example:** ADD 3
- Adds 3 to contents of accumulator and 3 is the operand.

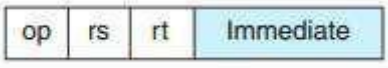


Fig 1 Immediate Mode

Direct Addressing:

- In direct addressing mode, effective address of the operand is given in the address field of the instruction.

- It requires one memory reference to read the operand from the given location and provides only a limited address space.
- Length of the address field is usually less than the word length.
- **Example :** Move P, Ro

  Add Q, Ro

  Where P and Q are the address of operand, R_o is any register.

Sometimes Accumulator (AC) is the default register. Then the instruction will look like:

Add A

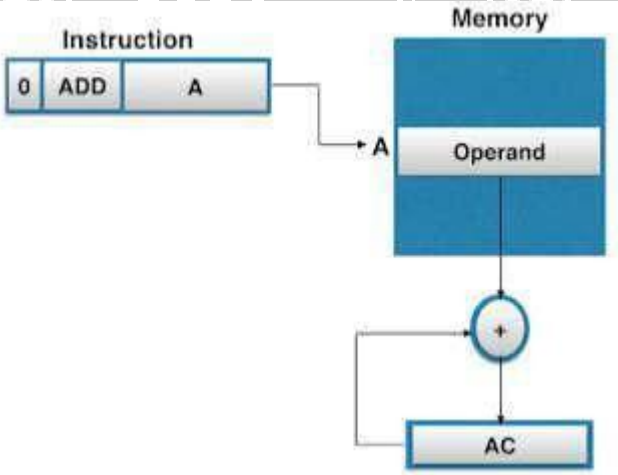


Fig 2 Direct Addressing modes

- **Indirect or Pseudo direct Addressing:**
  - Indirect addressing mode, the address field of the instruction refers to the address of a word in memory, which in turn contains the full length address of the operand.
  - The address field of instruction gives the memory address where on, the operand is stored in memory.
  - Control fetches the instruction from memory and then uses its address part to access memory again to read Effective Address.
  - The advantage of this mode is that for the word length of N, an address space

of 2N can be addressed.

- The disadvantage is that instruction execution requires two memory references to fetch the operand.
- Multilevel or cascaded indirect addressing can also be used.
- **Example:** Effective Address (EA) = (A).
- The operand will be present in the memory location A.

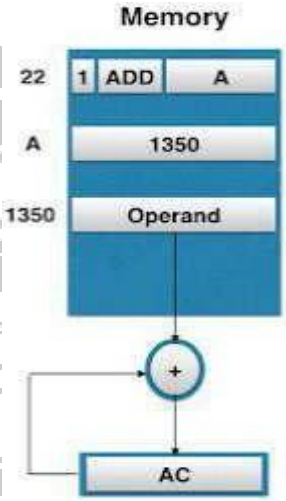


Fig 3 Indirect Addressing Modes

- **Register Addressing:**

- Register addressing mode is similar to direct addressing. The only difference is that the address field of the instruction refers to a register rather than a memory location.
- 3 or 4 bits are used as address field in the instruction to refer 8 to 16 generate purpose registers (GPR).
- The operands are in registers that reside within the CPU.
- The instruction specifies a register in CPU, which contain the operand.
- There is no need to compute the actual address as the operand is in a register and to get operand there is no memory access involved.
- The advantages of register addressing are small address field is needed in the instruction and faster instruction fetch.
- The disadvantages includes very limited address space and usage of multiple registers helps in performance but it complicates the instructions.

- **Example:** MOV AX, BX

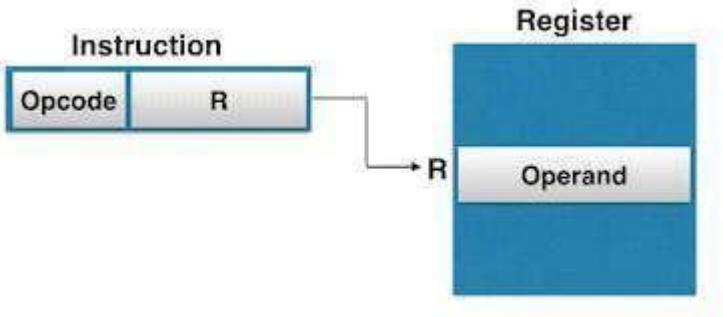


Fig 4 Register Mode

- **Register Indirect Addressing:**
  - This mode is similar to indirect addressing. The address field of the instruction refers to a register.
  - The instruction specifies a register in CPU whose contents give the operand in memory.
  - The selected register contain the address of operand rather than the operand itself.
  - The register contains the effective address of the operand. This mode uses one memory reference to obtain the operand.
  - Control fetches instruction from memory and then uses its address to access Register and looks in Register(R) for effective address of operand in memory.
  - The address space is limited to the width of the registers available to store the effective address.
  - **Example: MOV AL, [BX]**

    Code example in Register:

    MOV BX, 1000H

    MOV 1000H, operand
  - The instruction (MOV AL, [BX]) specifies a register [BX] which contain the address of operand (1000H) rather than address itself.

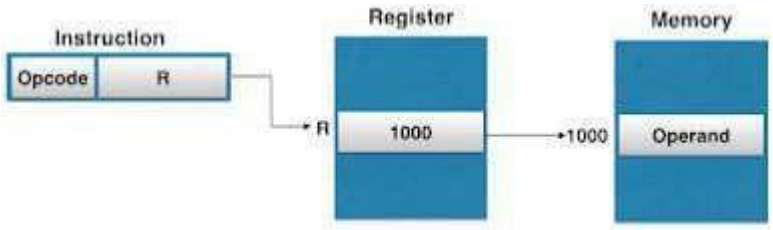


Fig 5 Register Indirect Mode

- **Displacement Addressing:**
  - It is a combination of direct addressing or register indirect addressing mode.
  - Displacement Addressing Modes requires that the instruction have two address fields, at least one of which is explicit means, one is address field indicate direct address and other indicate indirect address.
  - Value contained in one addressing field is A, which is used directly and the value in other address field is R, which refers to a register whose contents are to be added to produce effective address.

Example: EA=A+(R)

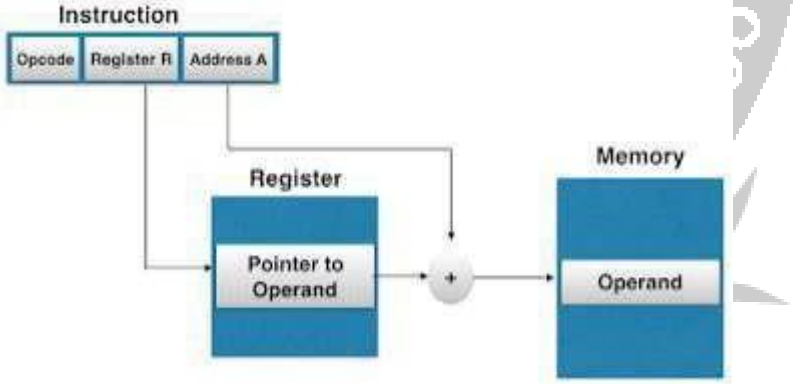


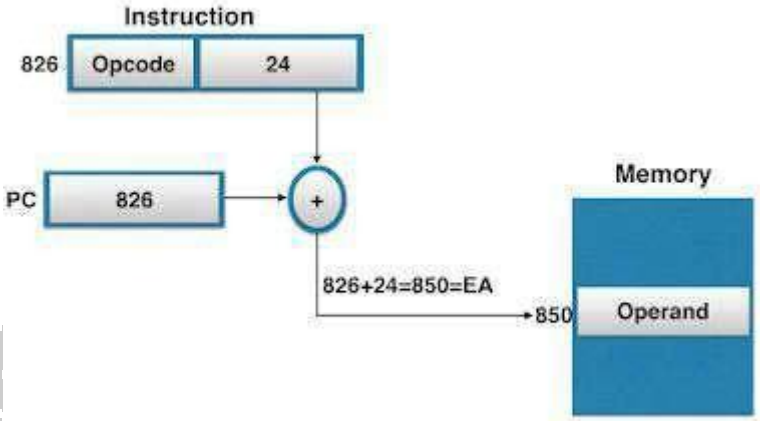
Fig 5 (a) Displacement Addressing Modes

- In displacement addressing mode there are 3 types of addressing mode.
- **Relative addressing:**

  The contents of program counter is added to the address part of instruction to obtain the Effective Address. The address field of the instruction is added to implicitly reference register Program Counter to obtain effective address.

  Example: EA=A+PC

Assume that PC contains the value 825 and the address part of instruction contain the value 24, then the instruction at location 825 is read from memory during fetch phase and the Program Counter is then incremented by one to 826. Here both PC and instruction contains address. The effective



address computation for relative address mode is 826+24=850

Fig 5 (b) Relative addressing

- **Base register addressing**

The content of the Base Register is added to the direct address part of the instruction to obtain the effective address. The address field point to the Base Register and to obtain EA, the contents of Instruction Register, is added to direct address part of the instruction. This is similar to indexed addressing mode except that the register is now called as Base Register instead of Index Register.

Example: EA=A +Base

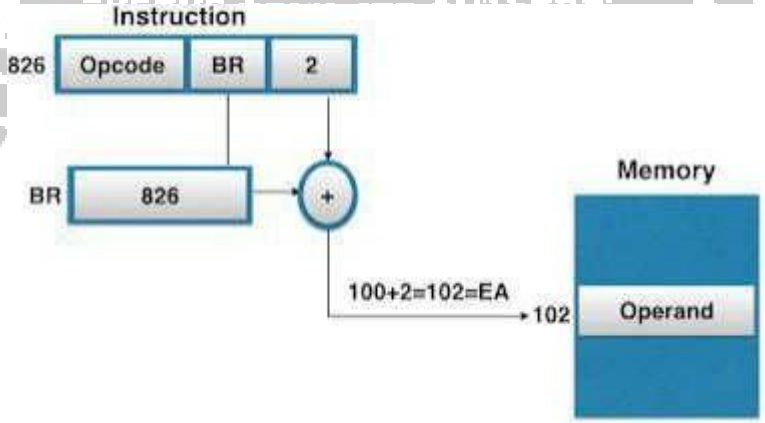


Fig 5 (c) Base Register Addressing Mode

Indexed addressing:

The content of Index Register is added to direct address part of instruction to obtain the effective address. The register indirect addressing field of instruction point to Index Register, which is a special CPU register that contain an Indexed value, and direct addressing field contain base address.

The data array is in memory and each operand in the array is stored in memory relative to base address. The distance between the beginning address and the address of operand is the indexed value stored in indexed register.

Any operand in the array can be accessed with the same instruction, which provided that the index register contains the correct index value i.e., the index register can be incremented to facilitate access to consecutive operands.

Example: EA=A+ Index

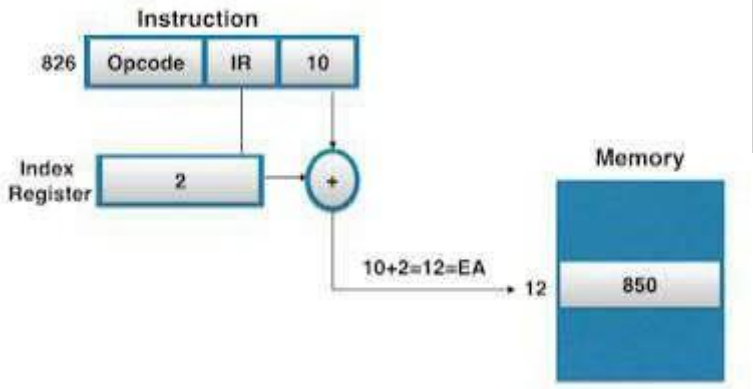


Fig 5(d) Indexed Addressing

Stack Addressing:

- Stack is a linear array of locations referred to as last-in first out queue.
- The stack is a reserved block of location, appended or deleted only at the top of the stack.
- Stack pointer is a register which stores the address of top of stack location.
- This mode of addressing is also known as **implicit addressing**.
- Example: Add
- This instruction pops two items from the stack and adds.

Additional Modes:

There are two additional modes. They are:

- Auto-increment mode
- Auto-decrement mode

These are similar to Register indirect Addressing Mode except that the register is incremented or decremented after (or before) its value is used to access memory. These modes are required because when the address stored in register refers to a table of data in memory, then it is necessary to increment or decrement the register after every access to table so that next value is accessed from memory.

Auto-increment mode:

- Auto-increment Addressing Mode are similar to Register Indirect Addressing Mode except that the register is incremented after its value is loaded (or accessed) at another location like accumulator (AC).
- The Effective Address of the operand is the contents of a register in the instruction.
- After accessing the operand, the contents of this register is automatically incremented to point to the next item in the list.
- **Example:** (R)

  +.
- The contents in register R will be accessed and them it will be incremented to point the next item in the list.

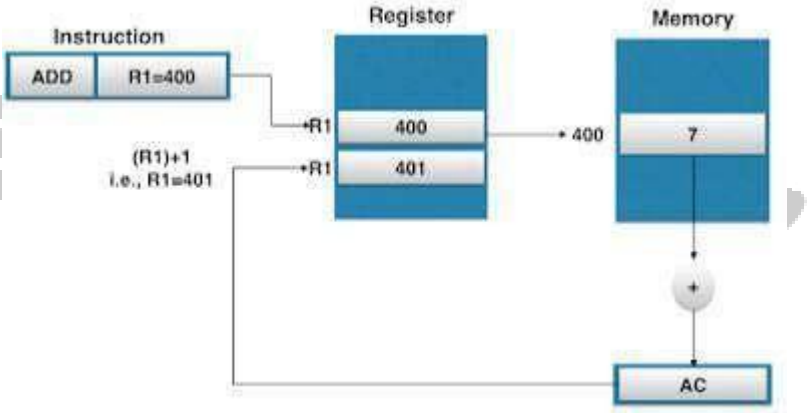


Fig 6 Auto-increment Mode

- The effective address is (R) =400 and operand in AC is 7. After loading R1 is incremented by 1, it becomes 401.

Auto-decrement mode:

- Auto-decrement Addressing Mode is reverse of auto-increment, as in it the register is decrement before the execution of the instruction.
- Effective address is equal to EA=(R)- 1
- The Effective Address of the operand is the contents of a register in the instruction.
- After accessing the operand, the contents of this register is automatically decremented to point to the next item in the list.
- **Example**: - (R)
- The contents in register R will be decremented and then it is accessed.

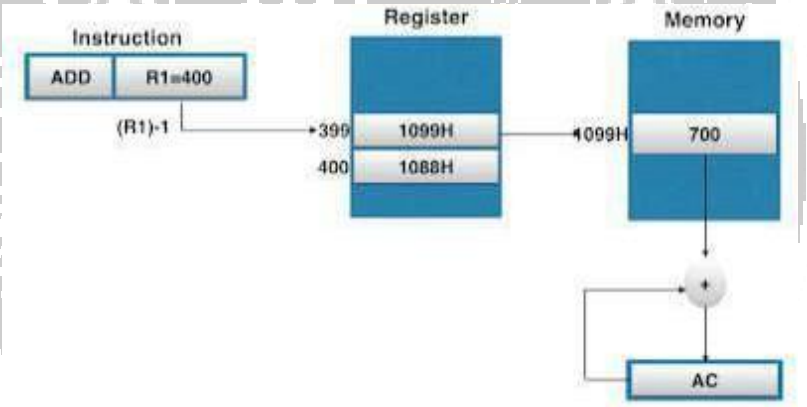


Fig 7 Auto Decrement Addressing Mode