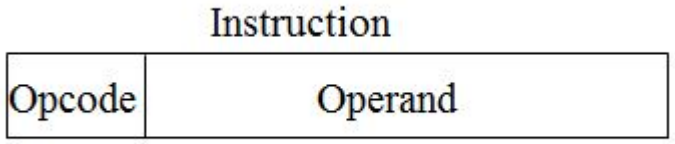
# Addressing Modes

# Addressing Modes

- The term ***addressing modes*** refers to the way in which the operand of an instruction is specified.
- An addressing mode specifies how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction.
- Types of Addressing Modes:
  1. Immediate
  2. Direct
  3. Indirect
  4. Register
  5. Register Indirect
  6. Displacement
  7. Stack
  8. Auto Increment Mode
  9. Auto Decrement Mode

# Immediate Addressing Mode

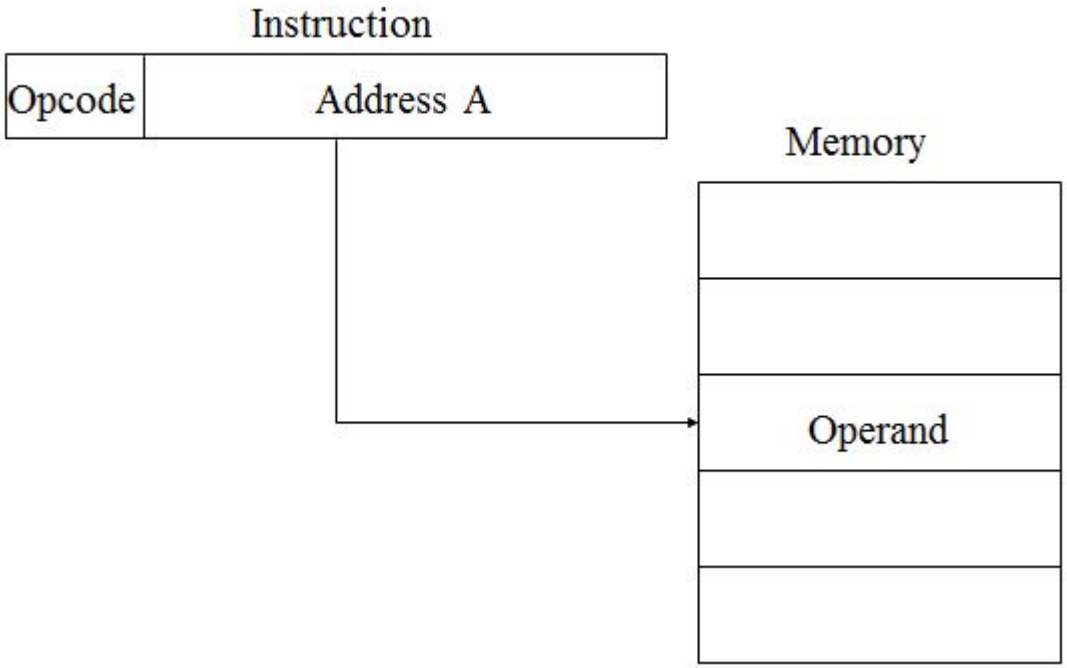
- Operand is specified in the instruction itself i.e. the instruction has operand field rather than address field.
- Example: ADD 5
  - Add 5 to contents of accumulator
  - 5 is operand
- No memory reference to fetch data
- Fast
- Limited range



# Direct Addressing Mode

- Address field contains address of operand
- Effective address (EA) = address field (A)
- e.g. ADD A
  - Add contents of cell A to accumulator
  - Look in memory at address A for operand
- Single memory reference to access data
- No additional calculations to work out effective address
- Limited address space

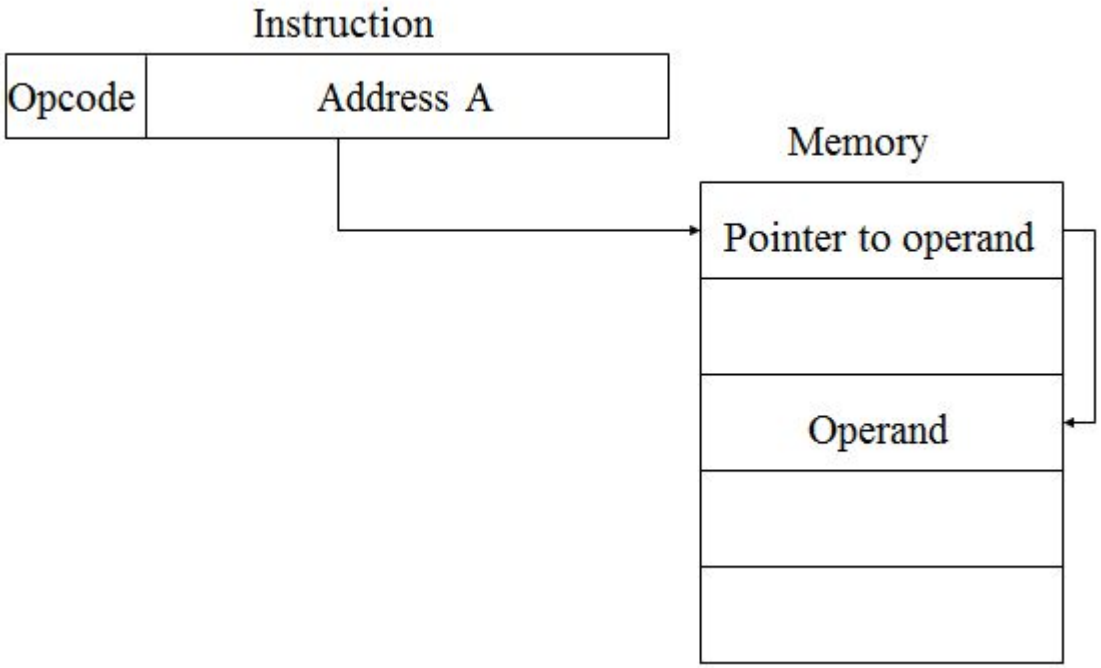
# Direct



# Indirect Addressing Mode

- Memory cell pointed to by address field contains the address of (pointer to) the operand
- EA = (A)
  - Look in A, find address (A) and look there for operand
- e.g. ADD (A)
  - Add contents of cell pointed to by contents of A to accumulator

# Indirect


Instruction
Opcode
Address A
Memory
Pointer to operand
Operand

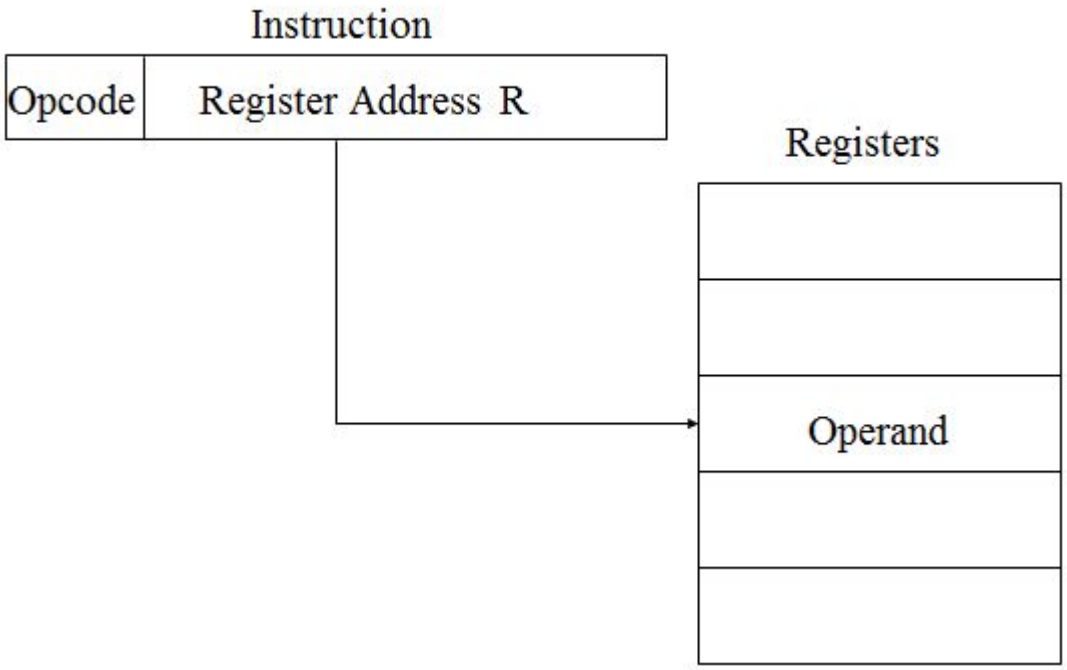
# Indirect

- Large address space
- $2^n$ where n = word length
- May be nested, multilevel, cascaded
  - e.g. EA = (((A)))
    - Draw the diagram yourself
- Multiple memory accesses to find operand
- Hence slower

# Register Addressing Mode

- Operand is held in register named in address filed
- Limited number of registers
- Very small address field needed
  - Shorter instructions
  - Faster instruction fetch

# Register



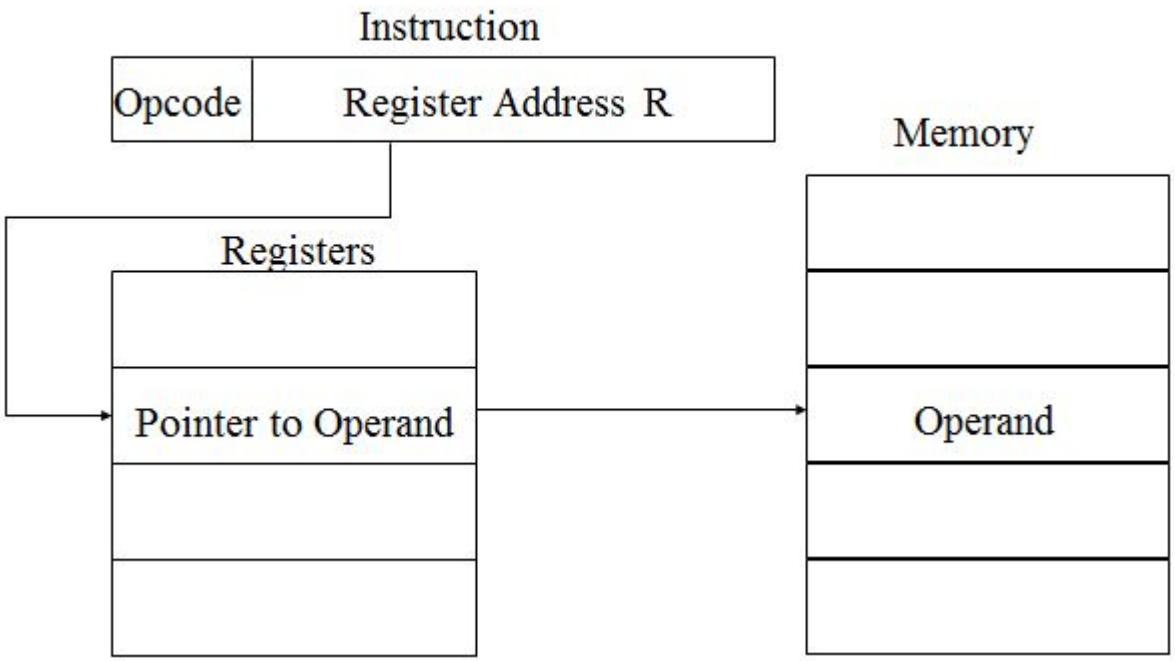
# Register

- No memory access
- Very fast execution
- Very limited address space
- Multiple registers helps performance
  - Requires good assembly programming or compiler writing

# Register Indirect Addressing Mode

- EA = content of R
- Operand is in memory cell pointed to by contents of register R
- Large address space ($2^n$)
- One fewer memory access than indirect addressing

# Register Indirect



# Displacement Addressing Mode

- A very powerful addressing mode that combines the capabilities of both direct addressing and register indirect addressing mode.
- EA = A + (R)
- Address field hold two values
  - A = base value
  - R = register that holds displacement
  - or vice versa

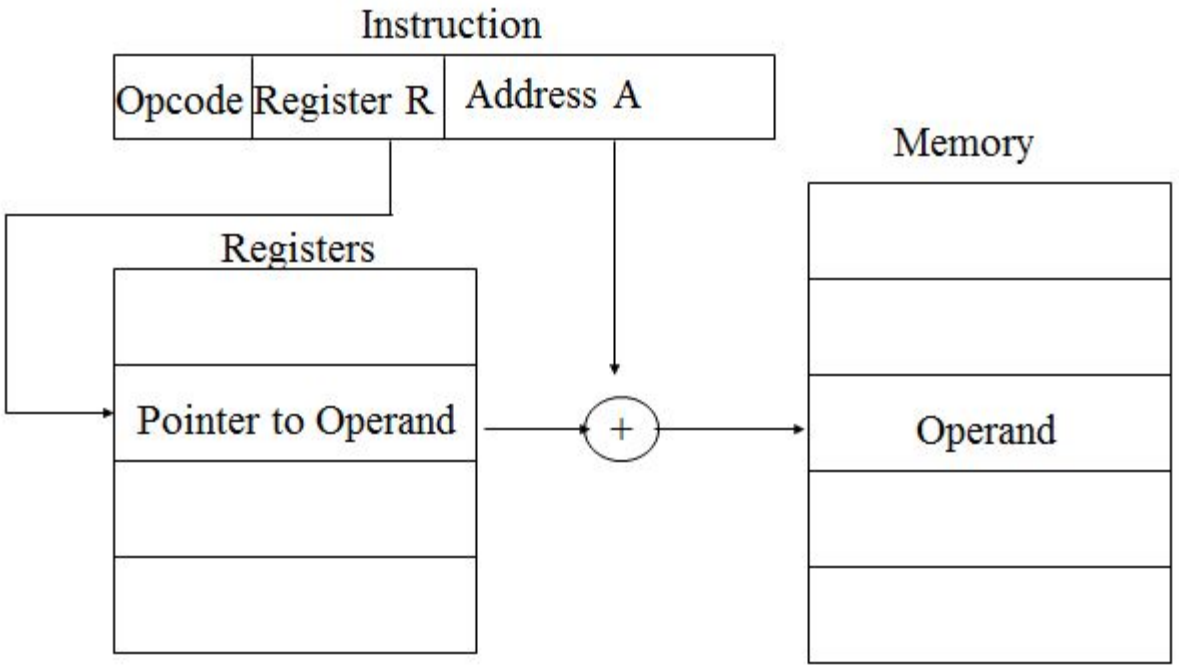
3 types

i. Relative addressing or PC Relative addressing

ii. Index-Register Addressing Mode

iii. Base –Register Addressing Mode

# Displacement



# Relative Addressing Mode

- A version of displacement addressing
- R = Program counter, PC
- EA = A + (PC)
- i.e. get operand from A cells from current location pointed to by PC

# Base-Index Addressing Mode

- A holds displacement
- R holds pointer to base address
- R may be explicit or implicit
- e.g. segment registers in 80x86

# Index Register Addressing Mode

- A = base
- R = displacement
- EA = A + R
- Good for accessing arrays
  - EA = A + R
  - R++

# Stack Addressing Mode

- Operand is (implicitly) on top of stack
- e.g.
  - ADD Pop top two items from stack and add

# Autoincrement or Autodecrement Mode

- Similar to the register indirect mode expect that the register is incremented or decremented after ( or before) its value is used to access memory.

PC = 200

R1=400

XR = 100

AC

200 LOAD, 500

| 200 | LOAD to AC | Mode |
|---|---|---|
| 201 | Address = 500 | |
| 202 | Next Instruction | |
| | | |
| 399 | 450 | |
| 400 | 700 | |
| | | |
| 500 | 800 | |
| 600 | 900 | |
| | | |
| 702 | 325 | |
| | | |
| | | |
| 800 | 300 | |

# Q. LOAD AC, 500

| Addressing Mode | Effective Address (EA) | Content of AC |
|---|---|---|
| Direct | 500 | 800 |
| Immediate | 201 | 500 |
| Indirect | 800 | 300 |
| Relative | 702 | 325 |
| Indexed | 600 | 900 |
| Register | - | 400 |
| Register Indirect | 400 | 700 |
| Auto increment | 400 | 700 |
| Auto decrement | 399 | 450 |