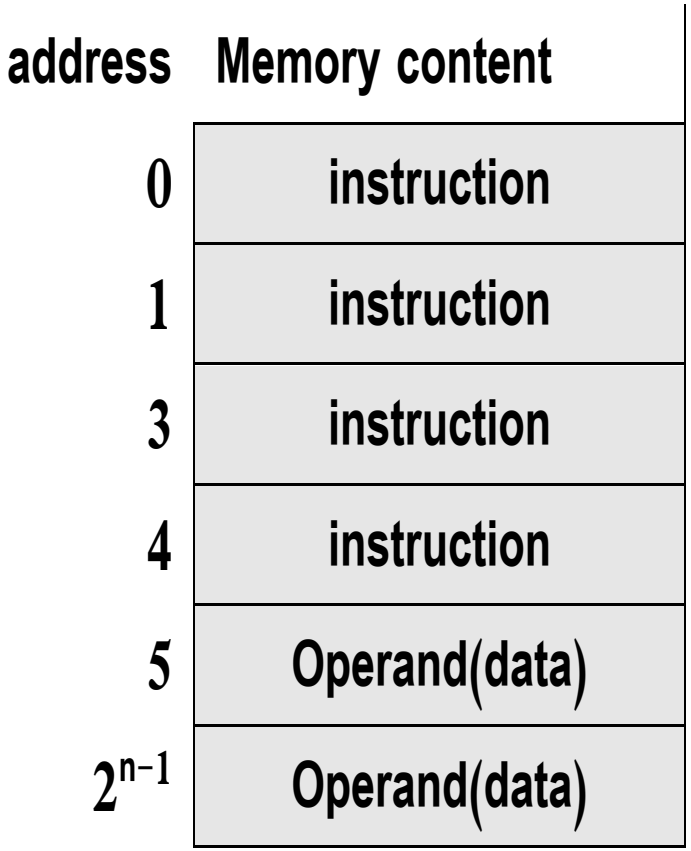
Addressing mode

Uses the address field of the instruction to determine the address of the memory location or one of the registers in the CPU to get the required operand in a process called the addressing in which get operand A .The address that appears directly in the code is called the **stated address**. And address of the memory location that contains the operand is called the **Effective Address**. Computer memory stores both instructions and data. See figure1

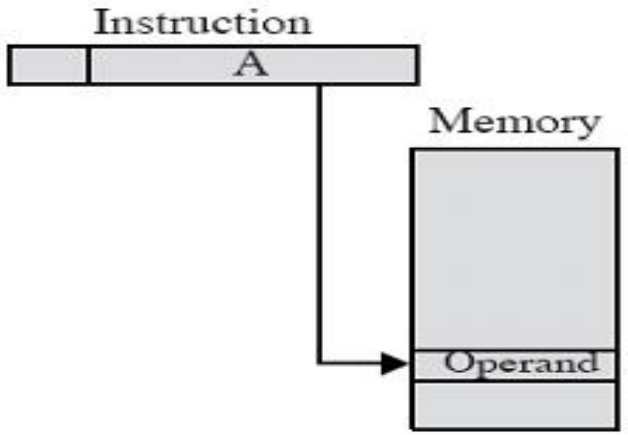

address
Memory content
0
instruction
1
instruction
3
instruction
4
instruction
5
Operand(data)
2^{n-1}
Operand(data)

Immediate Addressing

- Operand is part of instruction
- Operand = address field
- e.g. ADD 5
  - Add 5 to contents of accumulator
  - 5 is operand
- No memory reference to fetch data
- Fast
- Limited range

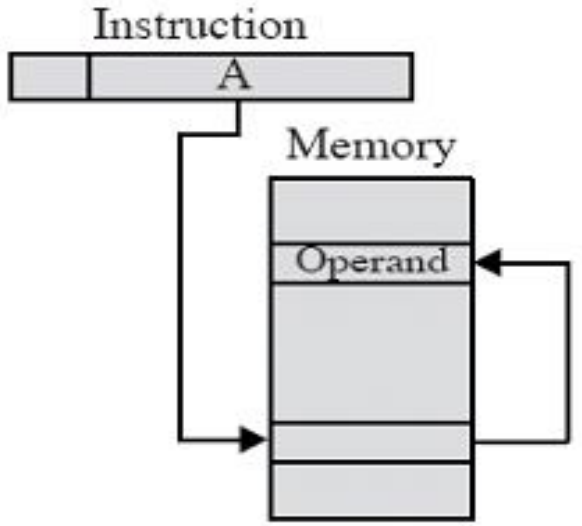
Direct Addressing

- 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
- Limited address space



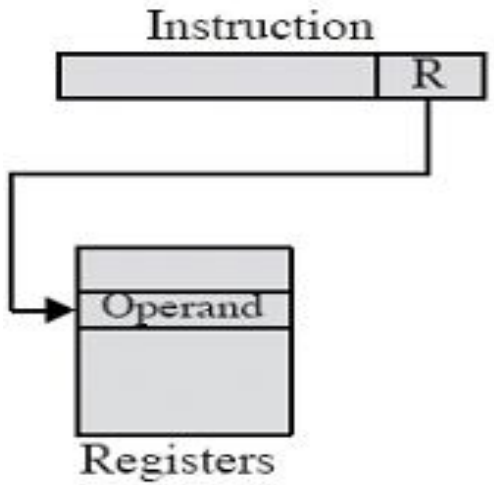
Indirect Addressing

- Memory cell pointed to by address field contains the address of (pointer to) the operand
  - EA = (A)



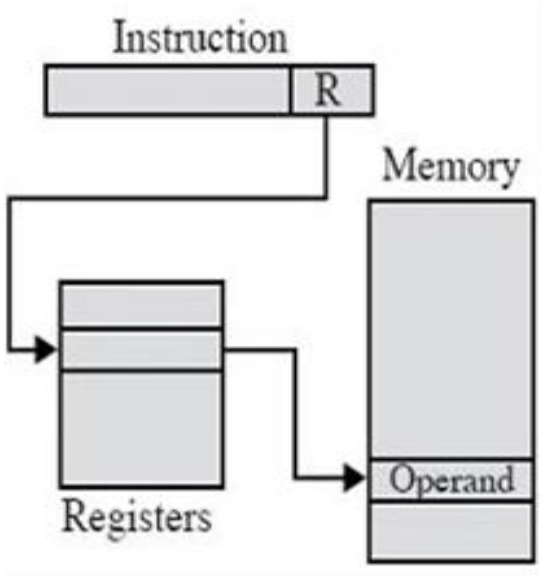
Register Addressing

- Operand is held in register named in address filed
- EA = R
- Limited number of registers
- Very small address field needed
  - —Shorter instructions
  - —Faster instruction fetch
- No memory access
- Very fast execution
- Very limited address space



Register Indirect Addressing

- Similar to memory-indirect addressing; much more common
- EA = (R)
- Operand is in memory cell pointed to by contents of register R
- Large address space (2n)
- One fewer memory access than indirect addressing



Displacement Addressing

- EA = A + (R)
- Combines register indirect addressing with direct addressing
- Address field hold two values

— A = base value

— R = register that holds displacement