Chapter #12-13
Instruction Sets

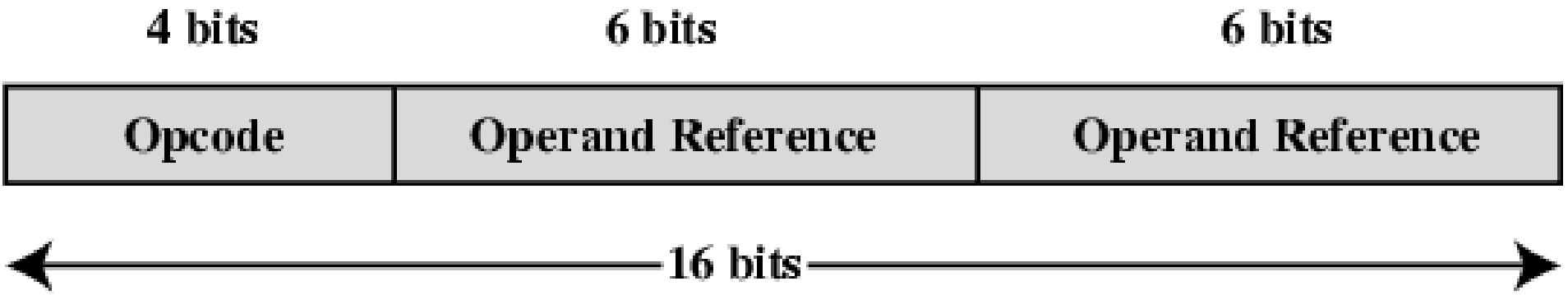
Instruction Set Details

- Complete collection of instructions understood by CPU
- Elements of an instruction:
  - —Opcode
    - type of instruction
  - —Operand(s)
    - Source, destination of instruction
    - Ex: registers, constants, addresses, labels

Instruction Types

- Data Transfer
  - —Move, Store, Load, Swap, Clear, Set Push, Pop
- Arithmetic
  - —Add, Subtract, Multiply, Divide, Absolute, Negate, Increment, Decrement
- Logical
  - —AND, OR, NOT, XOR, Compare, Shift, Rotate
- Program Control
  - —Jump (branch), Return, Skip, Halt, Wait, No operation
- I/O
  - —Read, Write

Instruction Format



Number of Operands

- 3 operands (not common)
  - —Ex: ADD R1,R2,R3 {R3=R1+R2}
- 2 operands
  - —Ex: MOVE R1, R2 {R1=R2}
- 1 operand
  - —Ex: LSH R2 {shift the contents of R2 one position to left}
- 0 operands
  - —Ex: POP {Remove top item from stack}

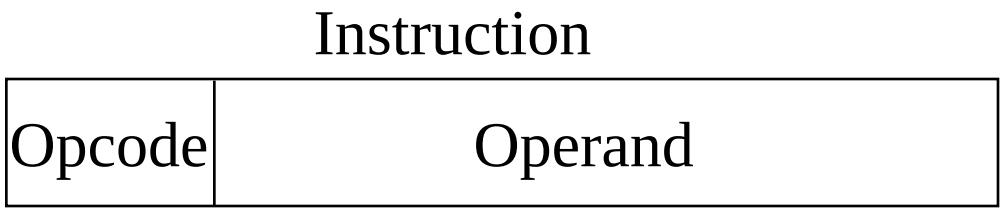
Addressing Modes

- Immediate
- Direct
- Indirect
- Register
- Register Indirect
- Displacement (Indexed)
- Stack

Immediate Addressing

- Address field is actual operand
- Ex:
  - —MOVE R1, #5
    (Copy value 5 to R1)
- Adv:
  - —No memory reference (fast)
- Disadv:
  - —Limited operand magnitude

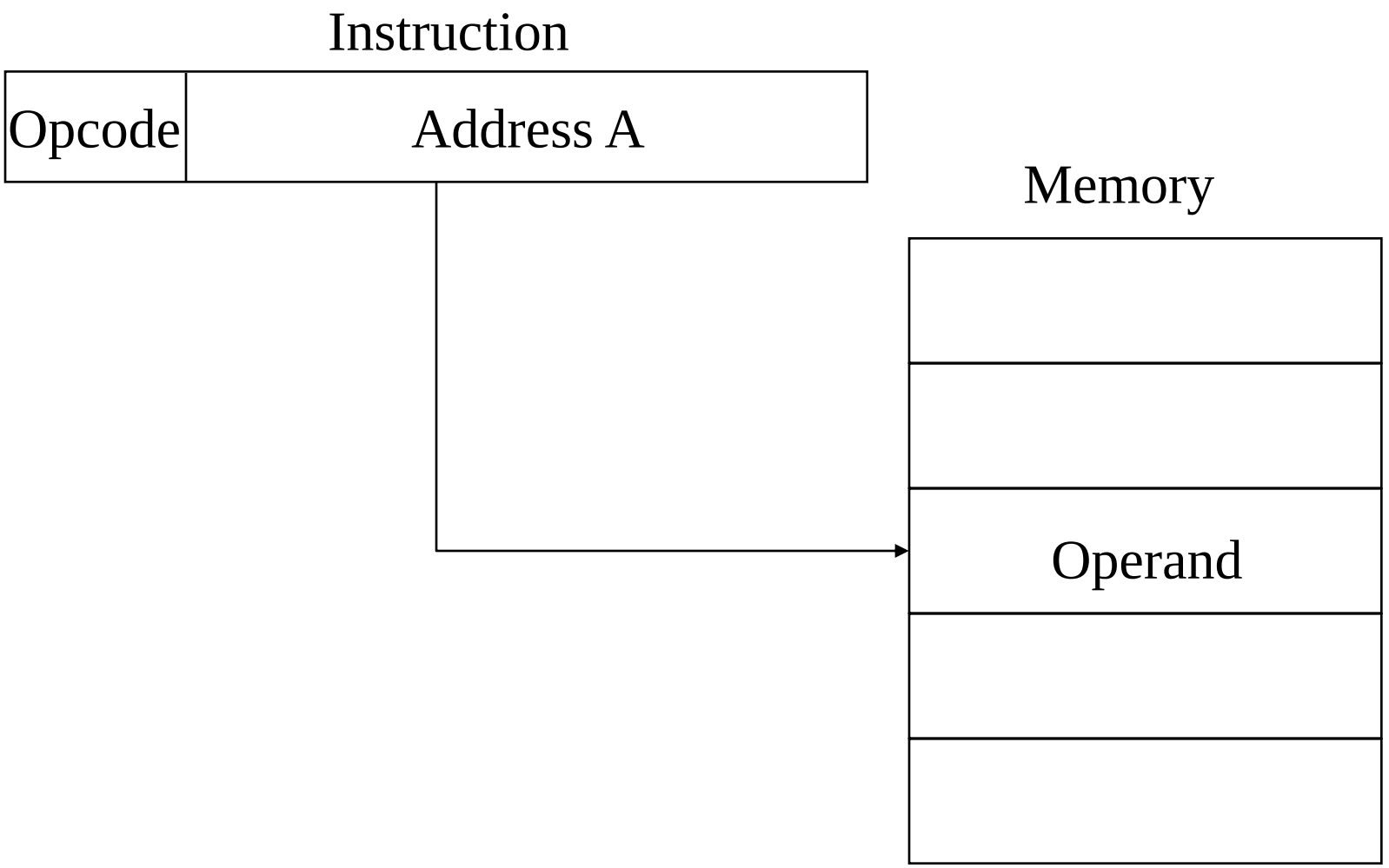
Immediate Addressing Diagram



Direct Addressing

- Address field contains address of operand
- Ex:
  - —MOVE R1, 1024
    (Copy contents of address 1024 to R1)
- Adv:
  - —Simple to reference (fast)
- Disadv:
  - —Limited address space

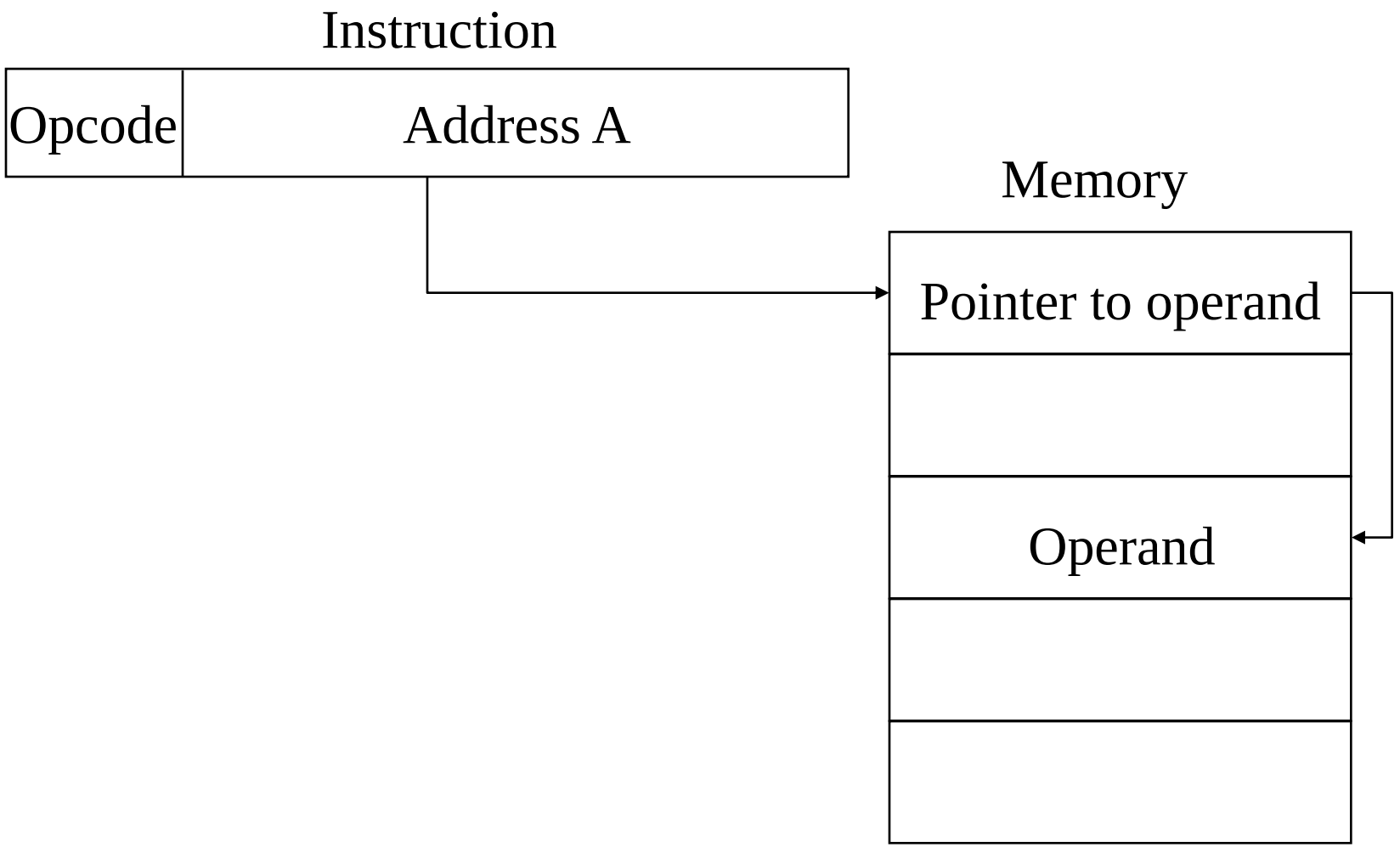
Direct Addressing Diagram



Indirect Addressing

- Memory cell pointed to by address field contains the address of (pointer to) the operand
- Ex:
  - —MOVE R1, (1024)
    (Copy contents of address contained in 1024 to R1)
- Adv:
  - —Large address space
- Disadv:
  - —Multiple memory references (slow)

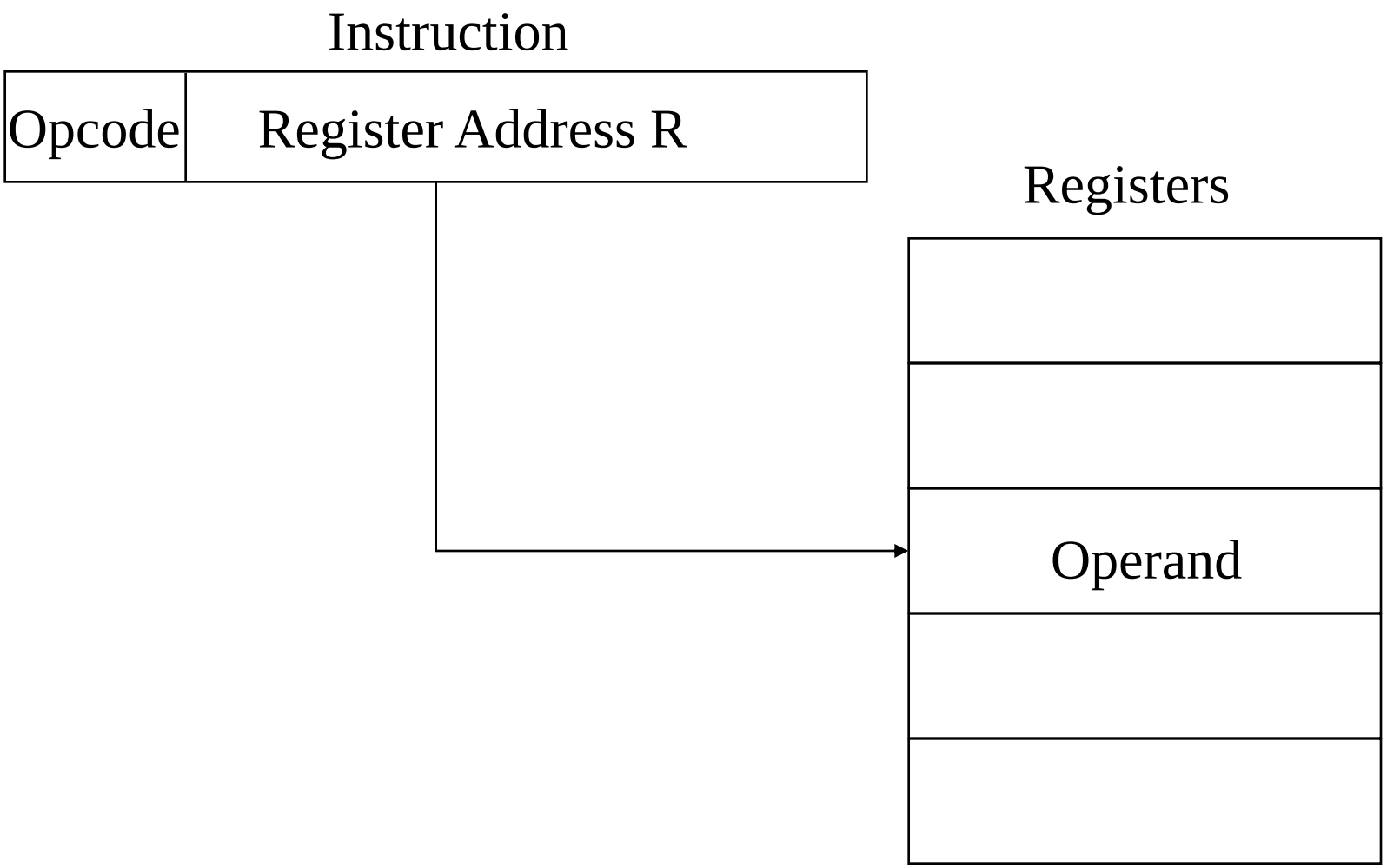
Indirect Addressing Diagram



Register Addressing

- Operand is held in register named in address field
- Ex:
  - —MOVE R1, R2
    (Copy contents of R2 into R1)
- Adv:
  - —No memory reference
- Disadv:
  - —Limited number of registers

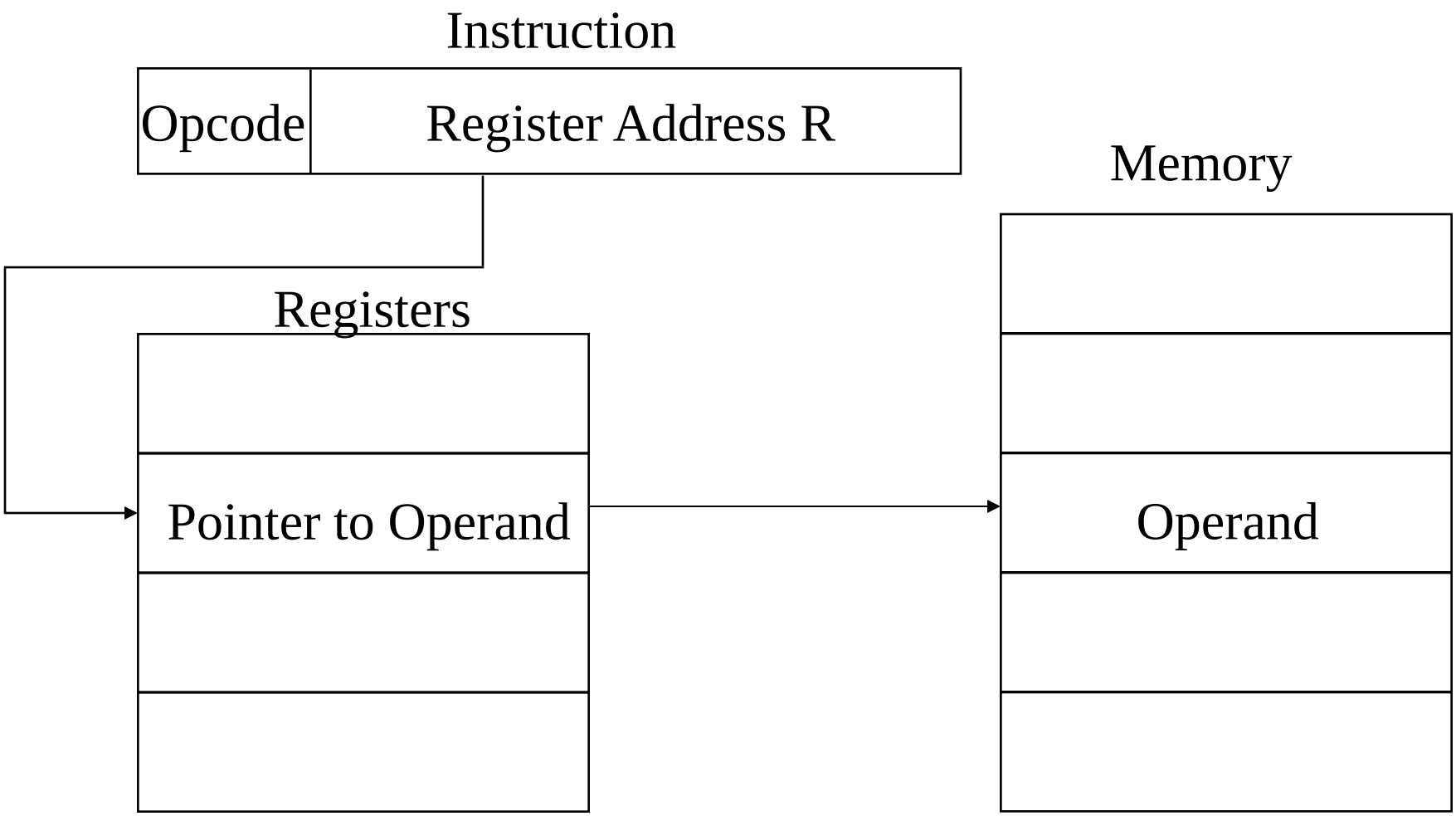
Register Addressing Diagram



Register Indirect Addressing

- Operand is held in address contained within register
- Ex:
  - —MOVE R1, (R2)
    (Copy contents of address contained in R2 to R1)
- Adv:
  - —Large address space
- Disadv:
  - —Extra memory reference

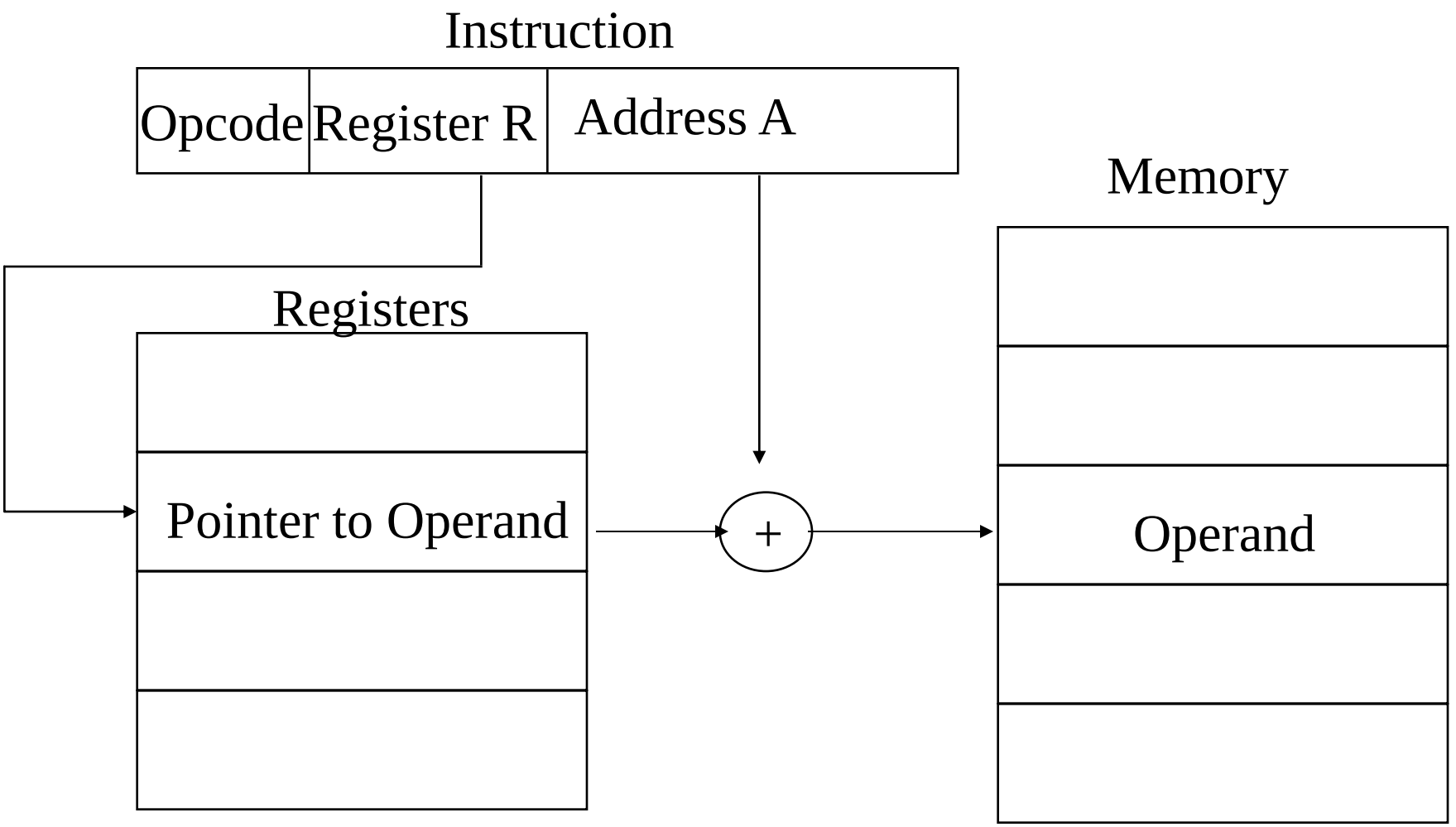
Register Indirect Addressing Diagram



Displacement Addressing

- Operand is contents of address produced as sum of base address and an offset
- Ex:
  - —MOVE R1, 1024+(R2)
    (Copy contents of address produced as sum of 1024 and address contained in R2 to R1)
- Adv:
  - —Very flexible
- Disadv:
  - —Complex

Displacement Addressing Diagram



Stack Addressing

- Destination operand is (implicitly) on top of stack
- Ex:
  - —PUSH R1
    (Push contents of R1 onto top of stack)
- Adv:
  - —No memory reference
- Disadv:
  - —Limited applicability