University of Engineering and Technology (UET), Peshawar, Pakistan

Lecture 8

CSE-304: Computer Organization and Architecture

BY:

Dr. Muhammad Athar Javed Sethi

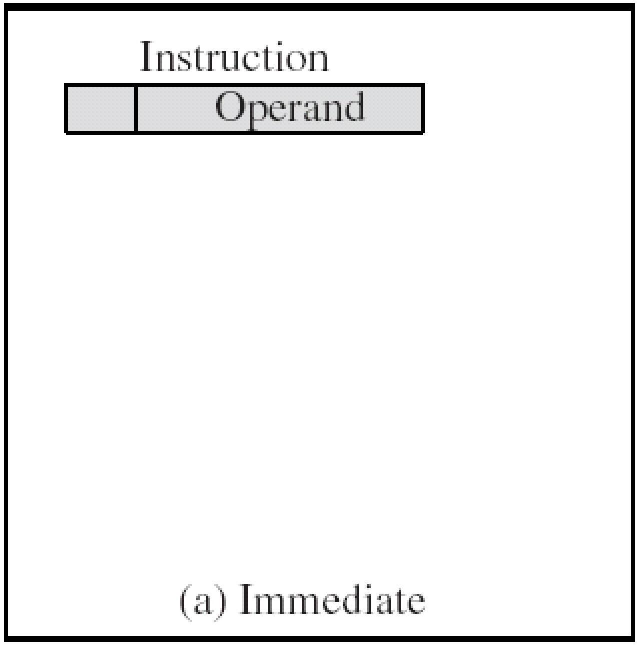
Slides are taken from student resources of William Stallings website

Addressing Modes

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

Immediate Addressing

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



(a) Immediate

Direct Addressing

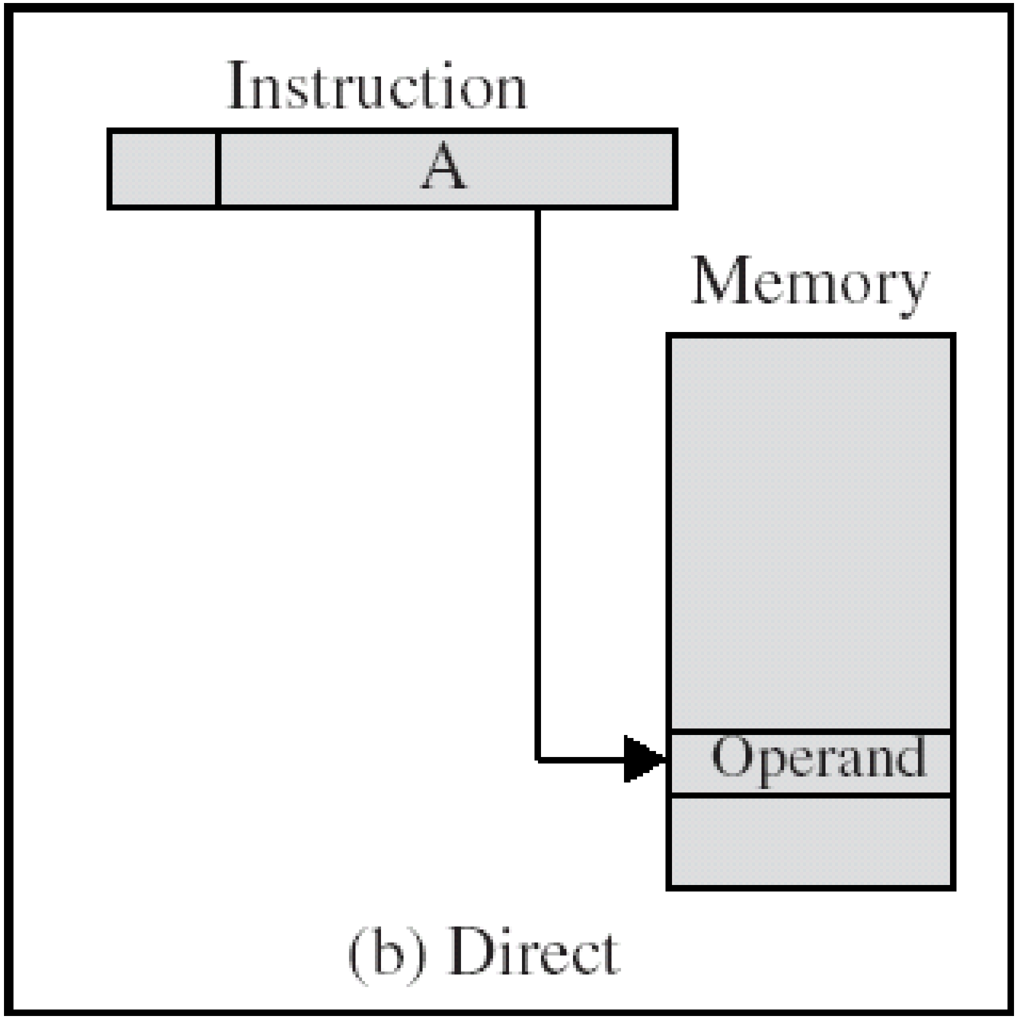
- Address field contains address of operand

ADD A, value

  - Add contents of cell value to accumulator A
  - Look in memory at address value for operand

- Single memory reference to access data
- No additional calculations to work out effective address

Direct Addressing Diagram



(b) Direct

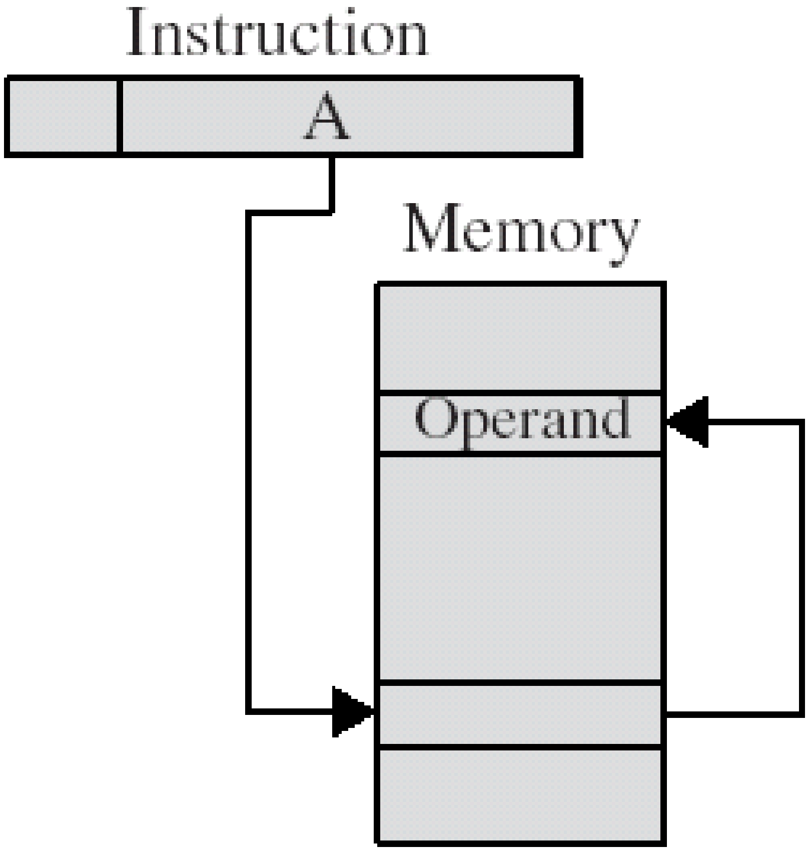
Indirect Addressing (1/2)

- 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, (A)
  - Add contents of cell pointed to by contents of A to accumulator

Indirect Addressing (2/2)

- Large address space
- May be nested, multilevel, cascaded
  - e.g. EA = (((A)))
- Multiple memory accesses to find operand
- Hence slower

Indirect Addressing Diagram



(c) Indirect

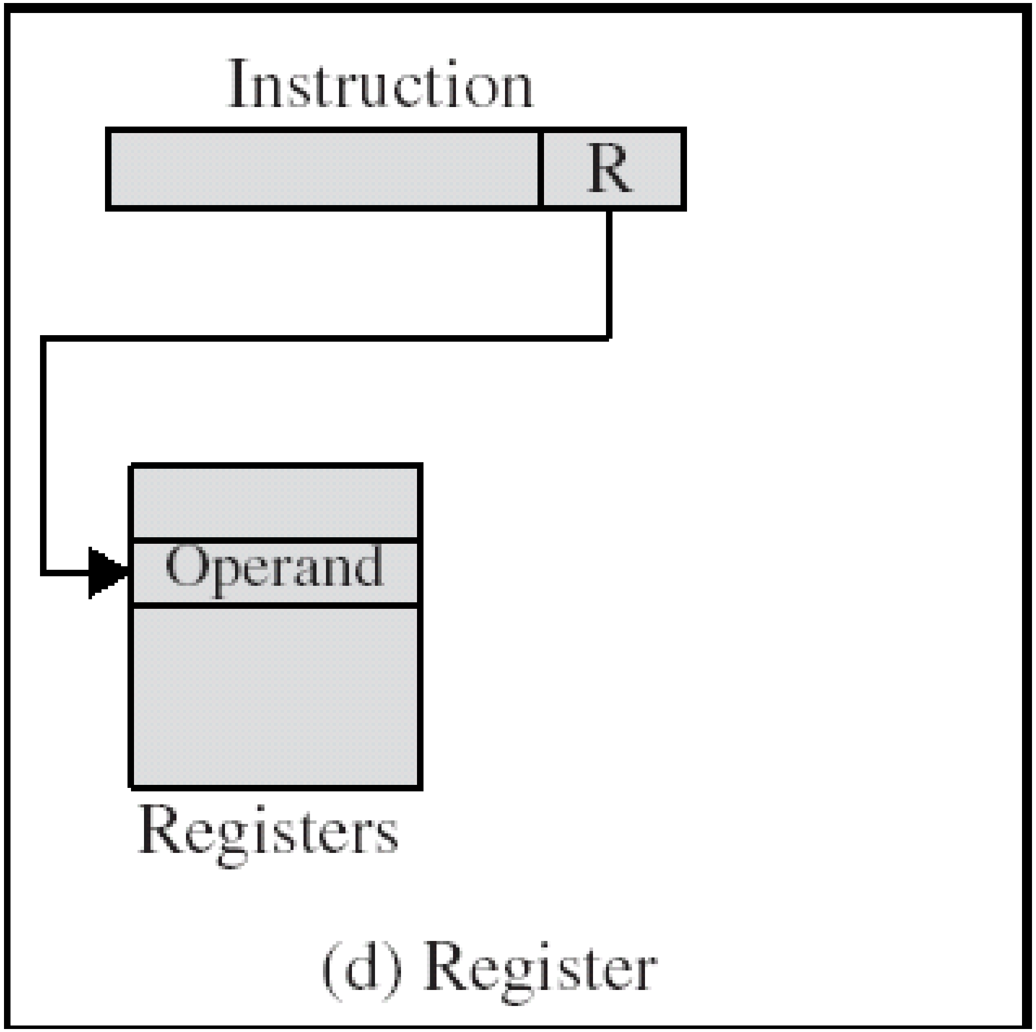
Register Addressing (1/2)

- 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
    - MOV A, B
    - ADD A, B

Register Addressing (2/2)

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

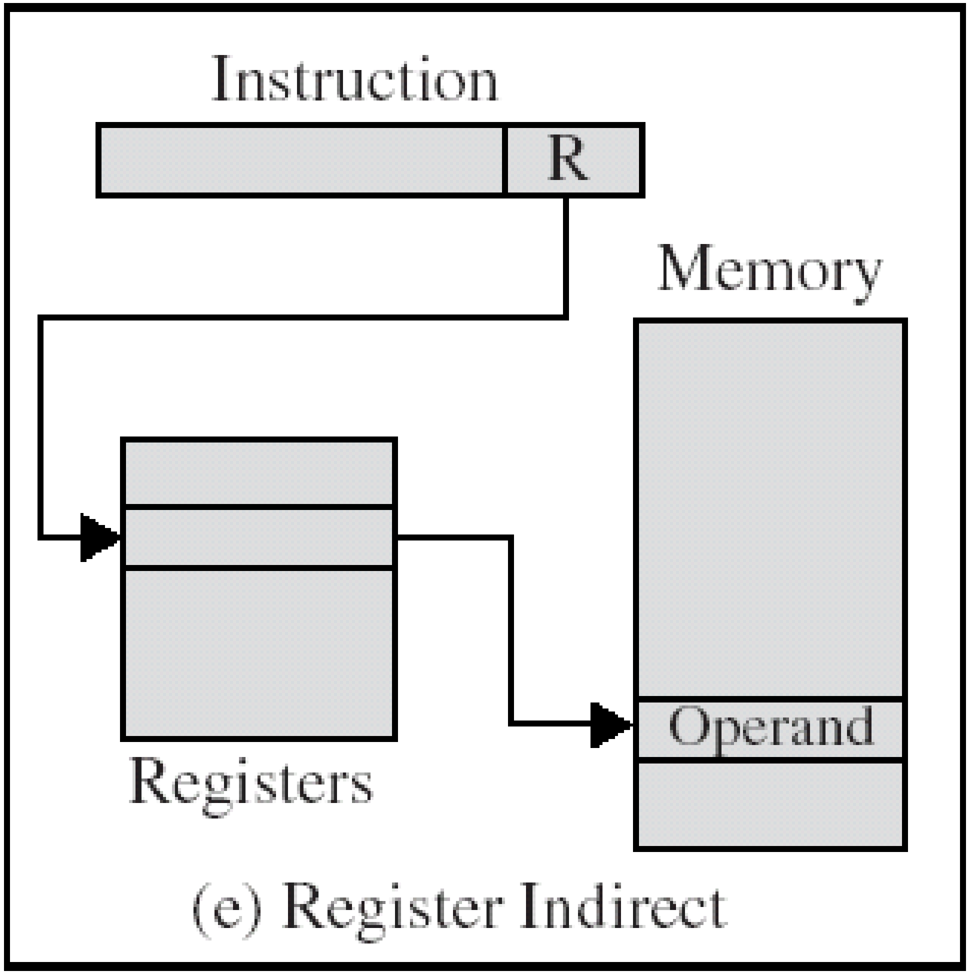
Register Addressing Diagram



Register Indirect Addressing

- EA = (R)
- Operand is in memory cell pointed to by contents of register R

Register Indirect Addressing Diagram

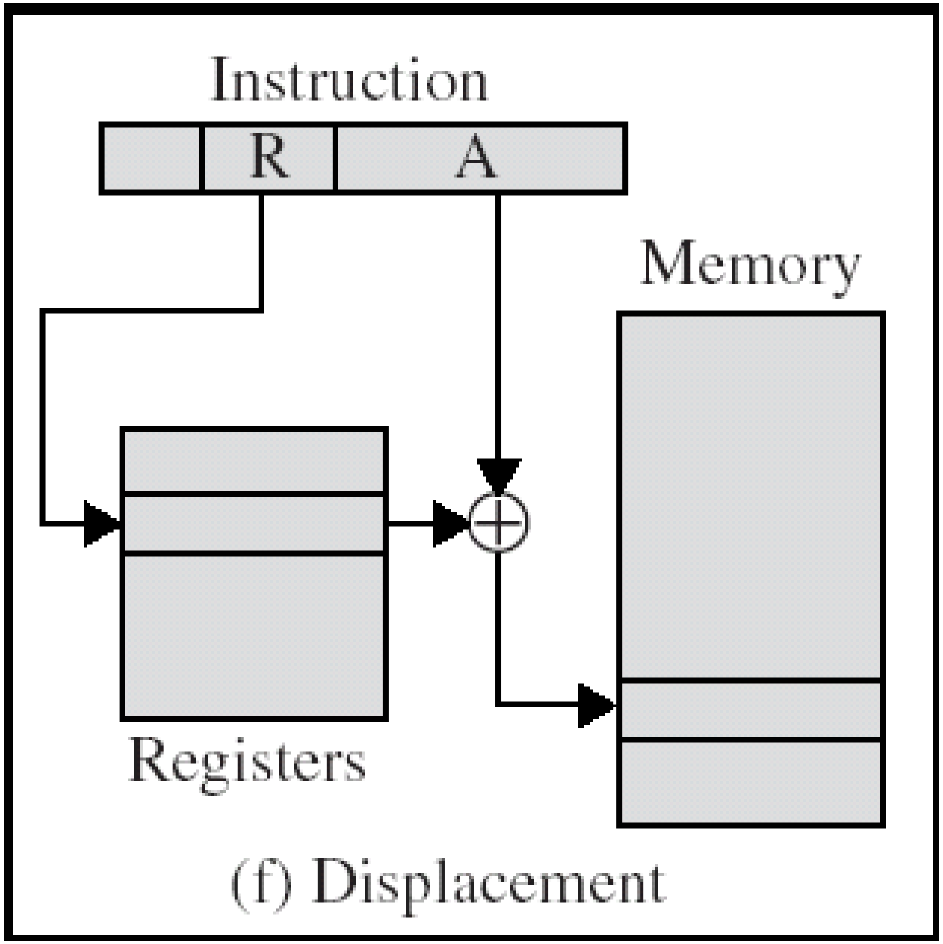


(e) Register Indirect

Displacement Addressing

- EA = A + (R)
- Effective address=start address + displacement
- Effective address=Offset + (Segment Register)
- Address field hold two values
  - A = base value
  - R = register that holds displacement
  - or vice versa

Displacement Addressing Diagram



(f) Displacement

Relative Addressing (PC-Relative)

- A version of displacement addressing
- R = Program counter, PC
- EA = A + (PC)

Base-Register Addressing

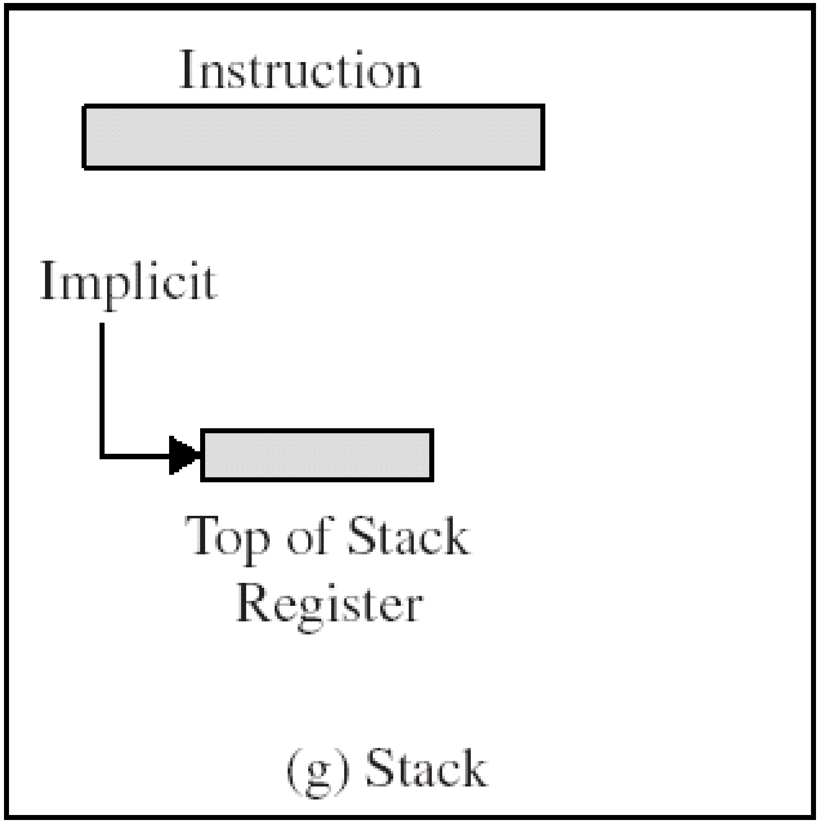
- A holds displacement
  - EA = (CS) + A
- CS holds pointer to base address

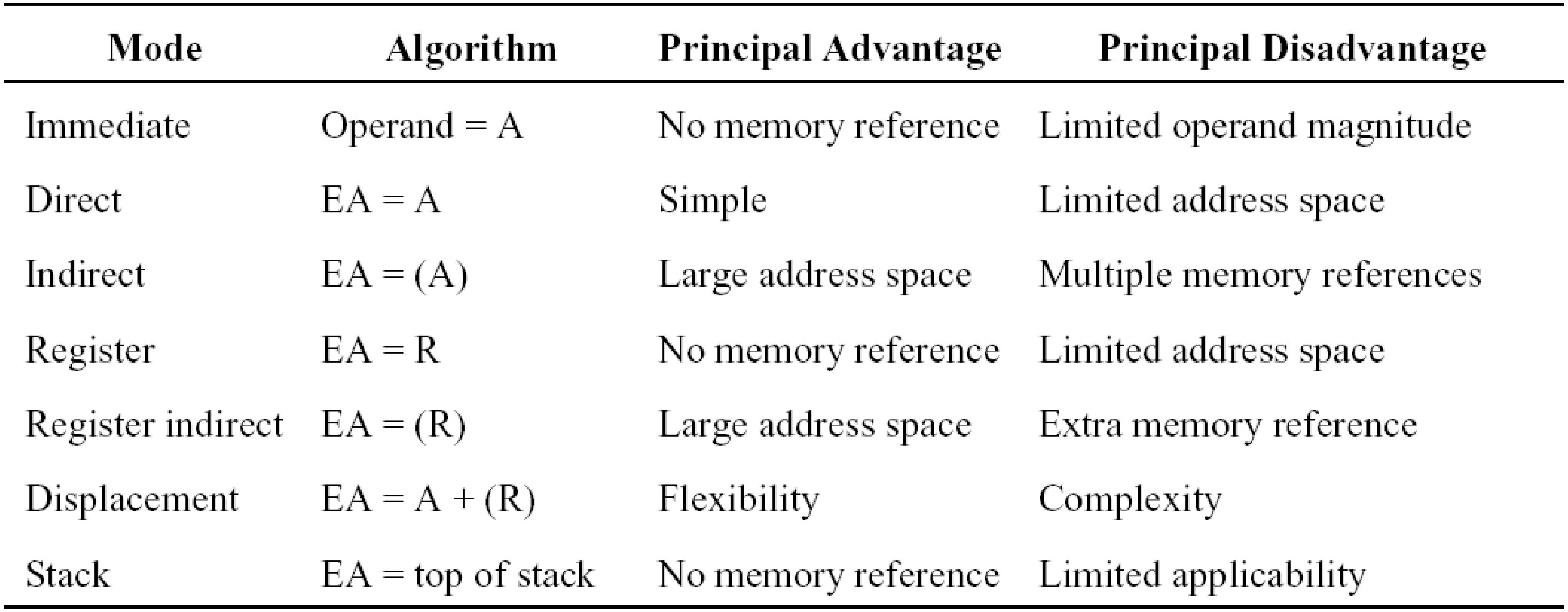
Indexed Addressing

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

Stack Addressing

- Operand is (implicitly) on top of stack





Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	EA = R	No memory reference	Limited address space
Register indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA = A + (R)	Flexibility	Complexity
Stack	EA = top of stack	No memory reference	Limited applicability