
Computer Architecture

Addressing Modes

Moniruzzaman

Adjunct Lecturer
North Western University, Khulna-9000

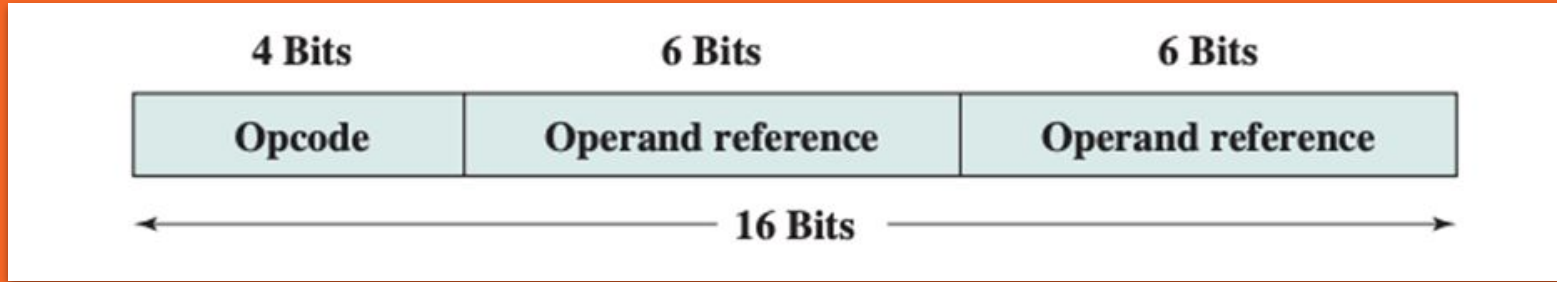
105



Confession

- ❖ Most of the materials have been collected from Internet.
- ❖ Images are taken from Internet.
- ❖ Various books are used to make these slides.
- ❖ Various slides are also used.
- ❖ References & credit:
 - Atanu Shome, Assistant Professor, CSE, KU.
 - Computer Organization and Design: the Hardware/Software Interface - Textbook by David A Patterson and John L. Hennessy.
 - Computer Organization and Architecture - Book by William Stallings

A Simple Instruction Format



Addressing Modes

Immediate

Direct

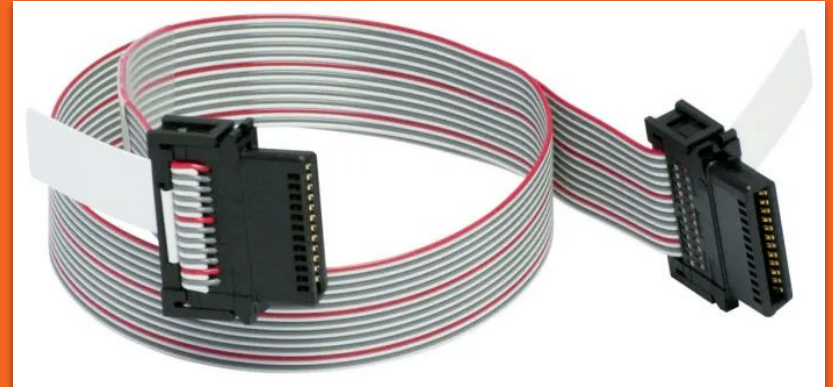
Indirect

Register

Register Indirect

Displacement (Indexed)

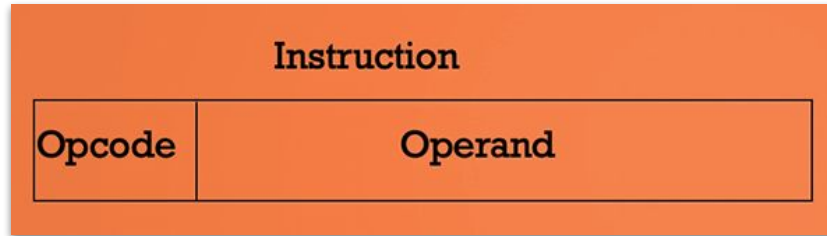
Stack



Comments

- One or More mode in same architecture
- Effective Address

Immediate Addressing



Operand is part of instruction

Operand = address field

e.g. ADD 5

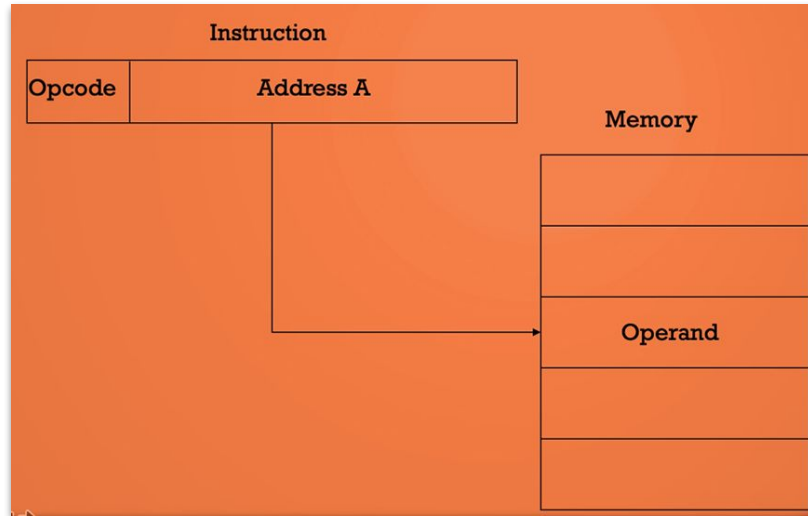
- Add 5 to contents of accumulator
- 5 is operand

No memory reference to fetch data

- Fast

Limited range

Direct Addressing



Address field contains address of operand

Effective address = address field

$$E.A = A$$

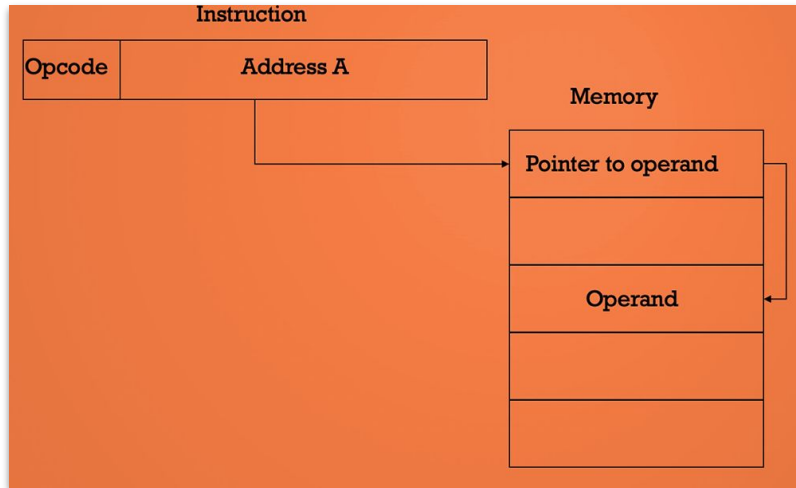
e.g. ADD A

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

Indirect Addressing



Memory cell pointed to by address field contains the address of (pointer to) the operand

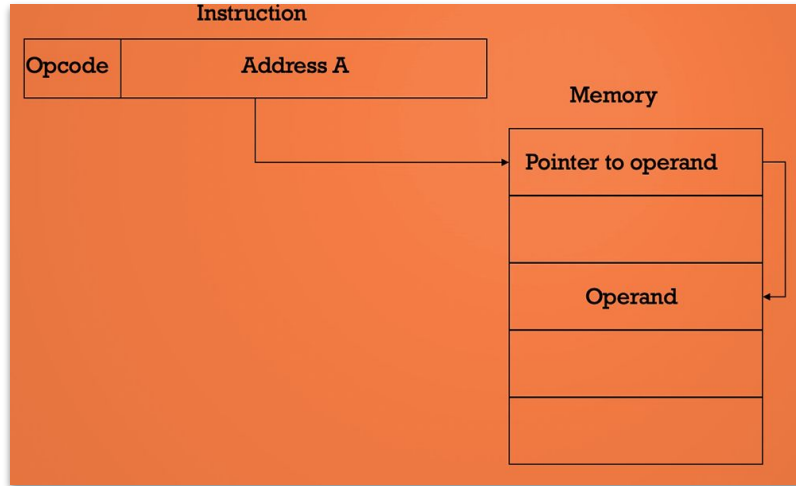
EA = (A)

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 Addressing



Large address space

2^n where n = word length

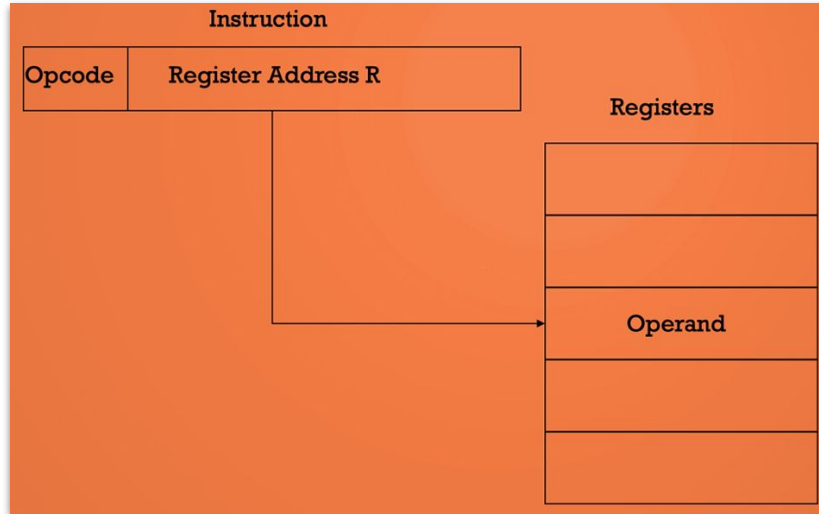
May be nested, multilevel, cascaded

e.g. $EA = ((A))$

Multiple memory accesses to find operand

Hence slower

Register Addressing



Operand is held in register named in address field

$$EA = R$$

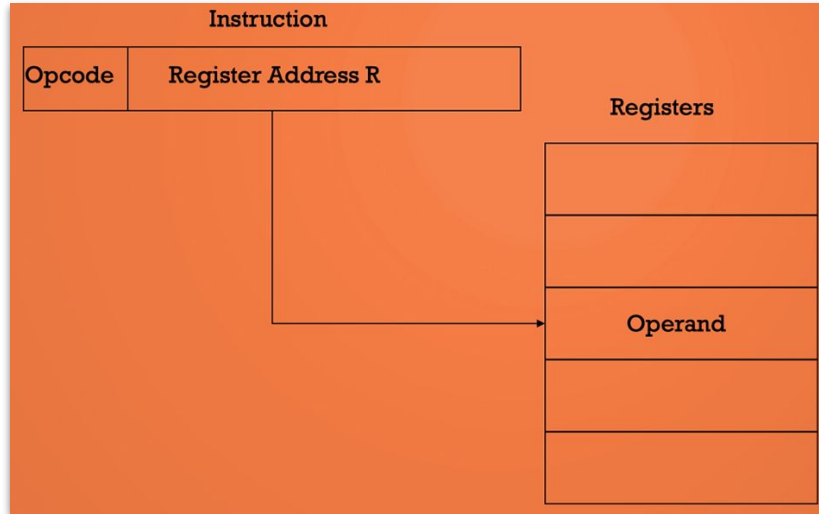
Limited number of registers

Very small address field needed

Shorter instructions

Faster instruction fetch

Register Addressing



No memory access Very fast execution

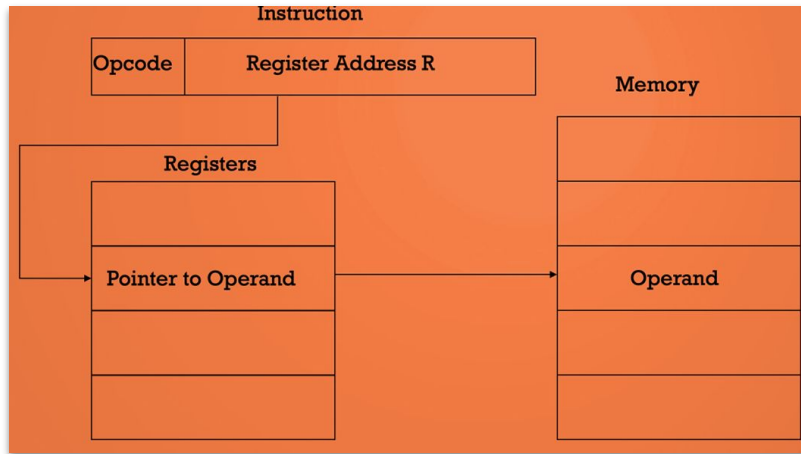
Very limited address space

Multiple registers helps performance

Requires good assembly programming or compiler writing

c.f. Direct addressing

Register Indirect Addressing



Indirect addressing

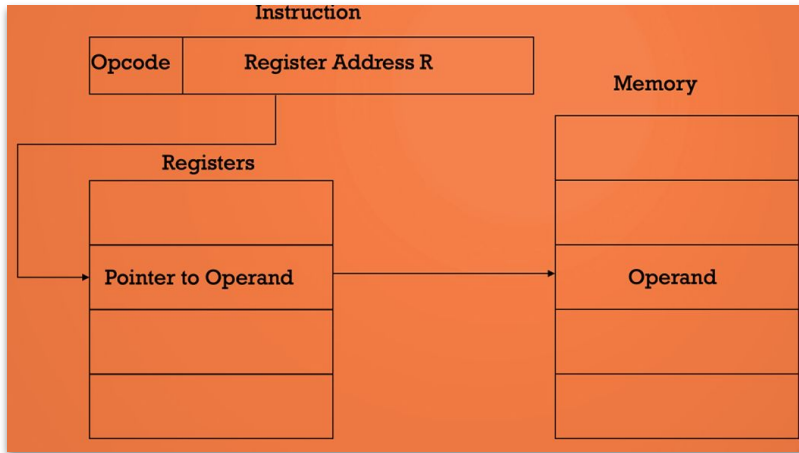
$$EA = (R)$$

Operand is in memory cell pointed to by contents of register R

Large address space (2^n)

One less memory access than indirect addressing

Displacement Addressing



$$EA = A + (R)$$

Address field hold two values

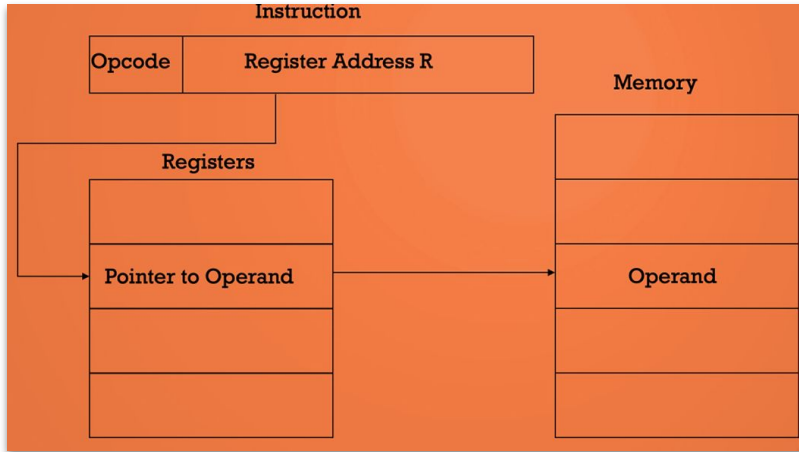
A = base value

R = register that holds displacement
or vice versa

Common uses of displacement
addressing

- Relative addressing
- Base-register addressing
- Indexing

Stack Addressing



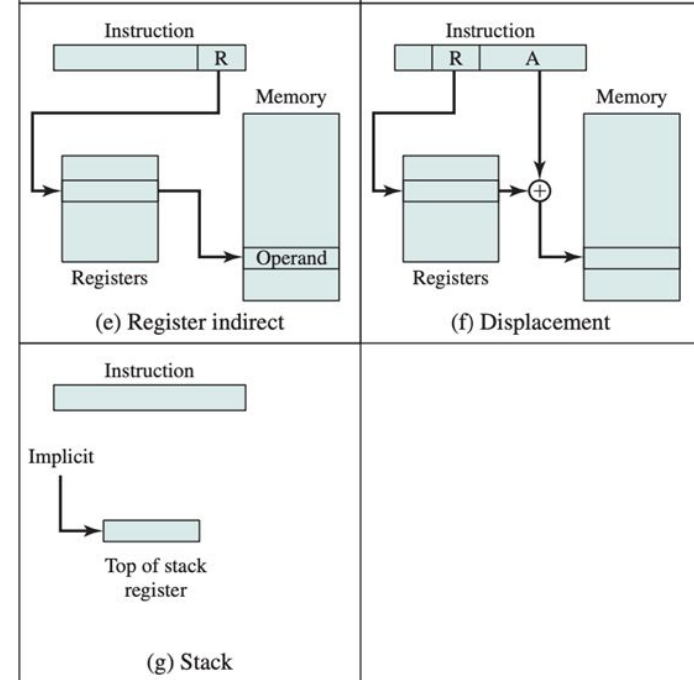
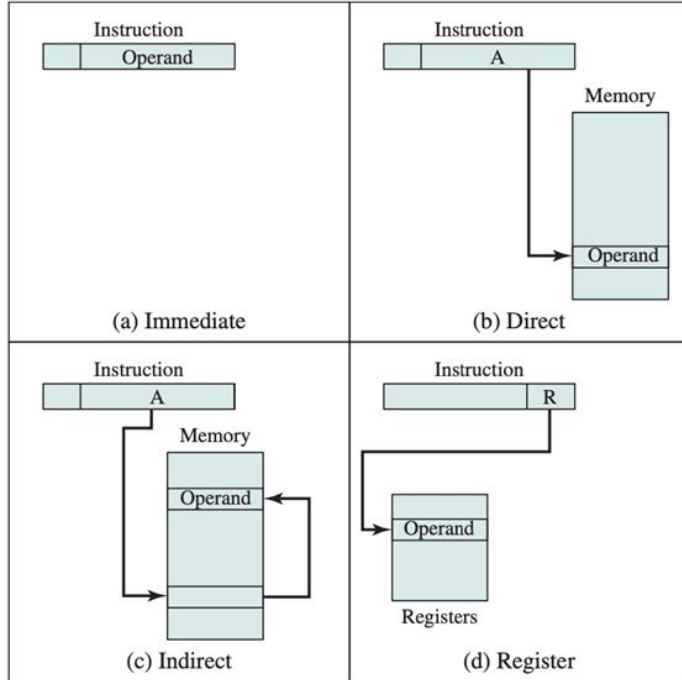
Operand is (implicitly) on top of stack

e.g.

ADD

Pop top two items from stack
and add

Addressing Modes



Addressing Modes

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

Thank You