Computer Organization

Douglas Comer

Computer Science Department
Purdue University
250 N. University Street
West Lafayette, IN 47907-2066

http://www.cs.purdue.edu/people/comer

VI

Operand Addressing And Instruction Representation

Number Of Operands Per Instruction

- Four basic architectural types
  - 0-address
  - 1-address
  - 2-address
  - 3-address

0-Address Architecture

- No explicit operands in the instruction
- Operands kept on stack in memory
- Instruction removes top *N* items from stack
- Instruction leaves result on top of stack

Example 0-Address Instructions

```
push X
push 7
add
pop  X
```

- Increments *X* by 7

1-Address Architecture

- One explicit operand per instruction
- Second operand is implicit
  - Always found in hardware register
  - Known as *accumulator*

Example 1-Address Instructions

load X
add 7
store X

2-Address Architecture

- Two explicit operands per instruction
- Result overwrites one of the operands
- Operands known as *source* and *destination*
- Works well for instructions such as memory copy

Example 2-Address Instructions

add 7, X

- Computes X $\leftarrow$ X + 7

3-Address Architecture

- Three explicit operands per instruction
- Operands specify *source*, *destination*, and *result*

Example Of 3-Address Instructions

add X, Y, Z

- Computes $Z \leftarrow X + Y$

Example Operand Types

- Operand that specifies a source
  - Signed constant
  - Unsigned constant
  - Contents of a register
  - Value in a memory location
- Operand that specifies a destination
  - Single register
  - Pair of contiguous registers
  - Memory location

Example Operand Types
(continued)

- Operand that specifies a constant is known as *immediate* value
- Memory references usually much more expensive than immediate or register access

Von Neumann Bottleneck

On a computer that follows the Von Neumann architecture, the time spent performing memory accesses can limit the overall performance. Architects use the term Von Neumann bottleneck *to characterize the situation, and avoid the bottleneck with techniques such as restricting most operands to registers.*

Operand Encoding

- Implicit type encoding
  - For given opcode, the type of each operand is fixed
  - More opcodes required
  - Example: opcode is *add_signed_immediate_to_register*
- Explicit type encoding
  - Operand specifies type and value
  - Fewer opcodes required
  - Example: opcode is *add*, operands specify *register* and *immediate*

Example Of Implicit Encoding

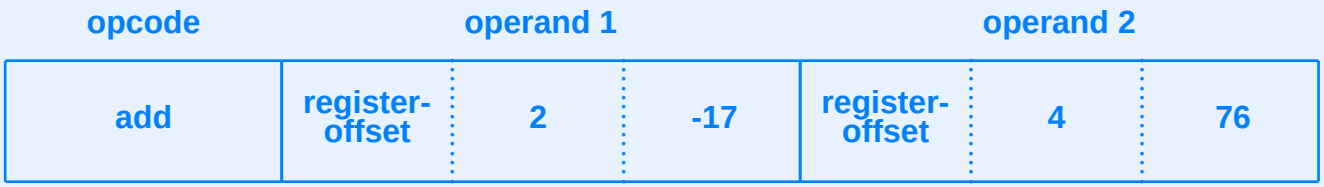
Opcode	Operands	Meaning
Add register	R1 R2	R1 $\leftarrow$ R1 + R2
Add immediate signed	R1 I	R1 $\leftarrow$ R1 + I
Add immediate unsigned	R1 UI	R1 $\leftarrow$ R1 + UI
Add memory	R1 M	R1 $\leftarrow$ R1 + memory[M]

Example Of Explicit Encoding

Combinations

- Operand specifies multiple items
- Processor computes final value from individual items
- Typical computation: sum
- Example
  - *Register-offset* specifies register and immediate value
  - Processor adds immediate value to contents of register

Illustration Of Register-Offset



Operand Tradeoffs

- No single style of operands optimal for all purposes
- Tradeoffs among
  - Ease of programming
  - Fewer instructions
  - Smaller instructions
  - Larger range of immediate values
  - Faster operand fetch and decode
  - Decreased hardware size

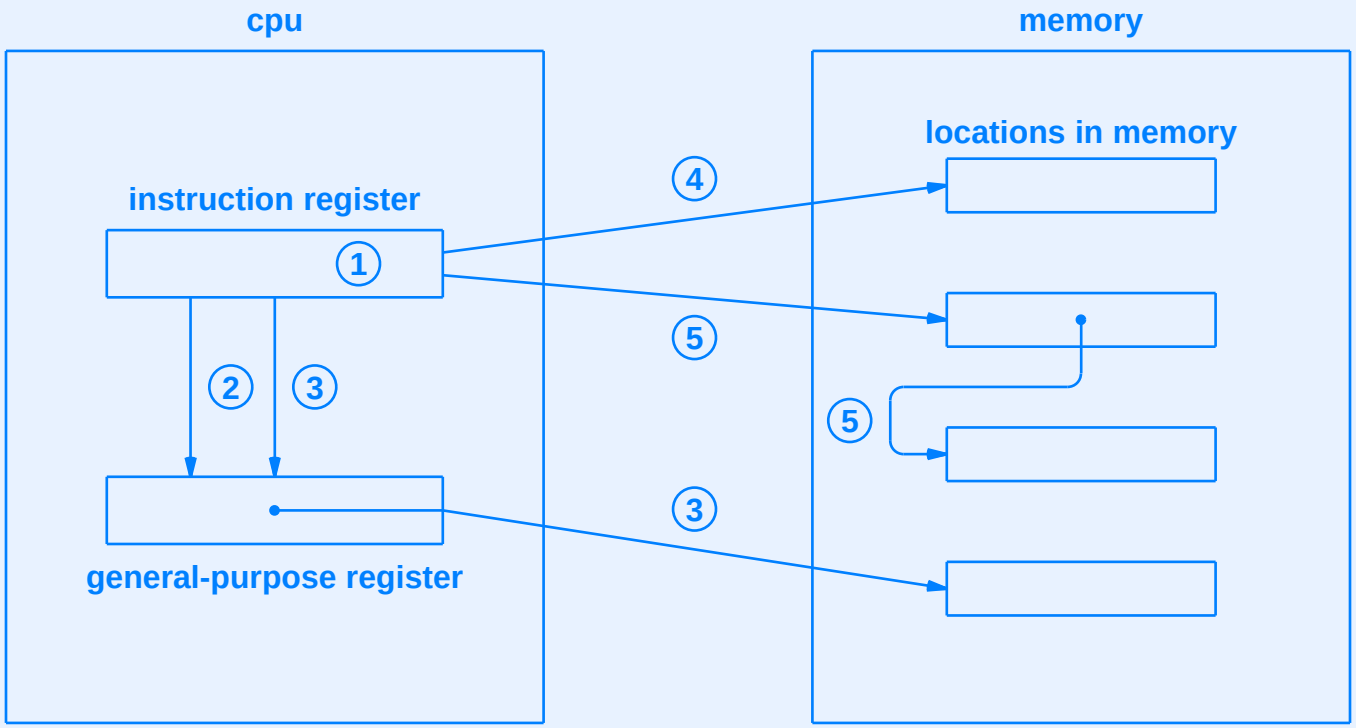
Operands In Memory And Indirect Reference

- Operand can specify
  - Value in memory (*memory reference*)
  - Location in memory that contains the address of the operand (*indirect reference*)
- Note: memory references are relatively expensive

Types Of Indirection

- Indirection through a register
  - Operand specifies register number, *R*
  - Obtain A, the current value from register *R*
  - Interpret A as a memory address, and fetch the operand from memory location A
- Indirection through a memory location
  - Operand specifies memory address, A
  - Obtain M, the value in memory location A
  - Interpret M as a memory address, and fetch the operand from memory location M

Illustration Of Operand Addressing Modes



1 Immediate value (in the instruction)

2 Direct register reference

3 Indirect through a register

4 Direct memory reference

5 Indirect memory reference

Summary

- Architect chooses the number and types of operands for each instruction
- Possibilities include
  - Immediate (constant value)
  - Contents of register
  - Value in memory
  - Indirect reference to memory

Summary
(continued)

- Type of operand can be encoded
  - Implicitly
  - Explicitly
- Many variations exist; each represents a tradeoff


Questions?