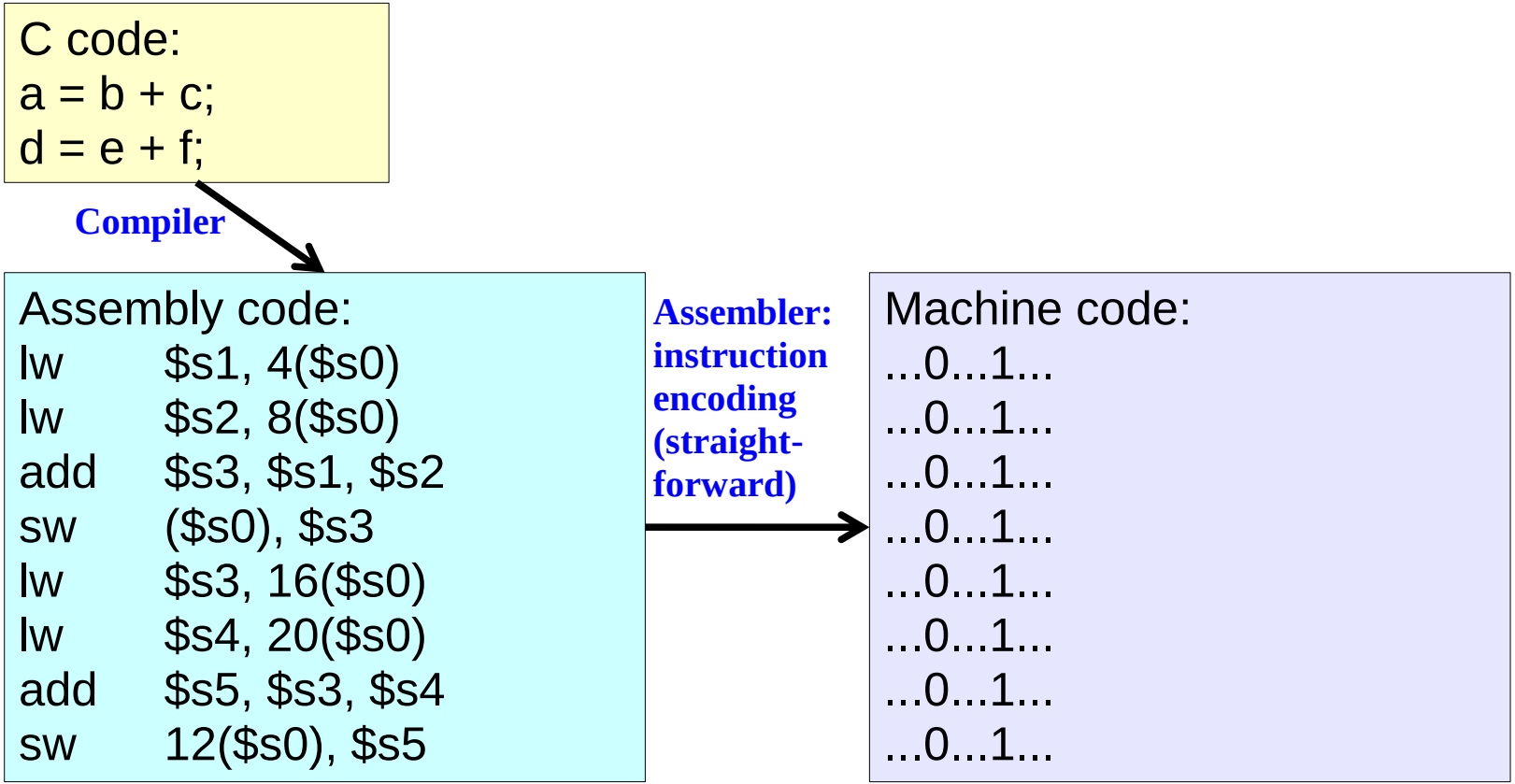
CS305
Computer Architecture

Instruction Encoding

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

A Simple Example



Instruction Encoding

- Encoding: representing instructions as numbers/bits
  - Recall: instructions are also stored in memory!
  - Encoding == (assembly language → machine language)
- MIPS: all instructions are encoded as 32 bits (why?)
- Also, all instructions have *similar* format (why?)

Regularity ⇒ simplicity ⇒ efficient implementation

MIPS Instruction Format

add
sub

opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)

R-type instruction: register-register operations

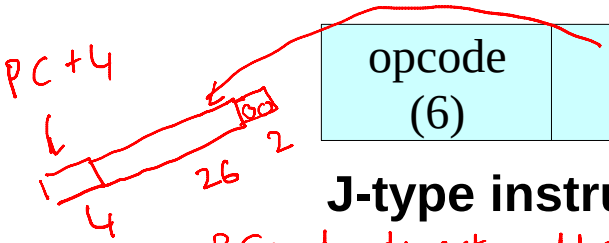
addi

opcode (6)	rs (5)	rt (5)	immediate/constant or offset (16)

PC+4

I-type instruction: loads, stores, all immediates, *conditional branch, jump register, jump and link register*

PC+4



opcode (6)	offset relative to PC+4 word (26)

J-type instruction: *jump, jump and link, trap and return*

Pseudo-direct addressing

Test Your Understanding…

- What format is used by the slt instruction?
- What instruction format is used by beq ?