



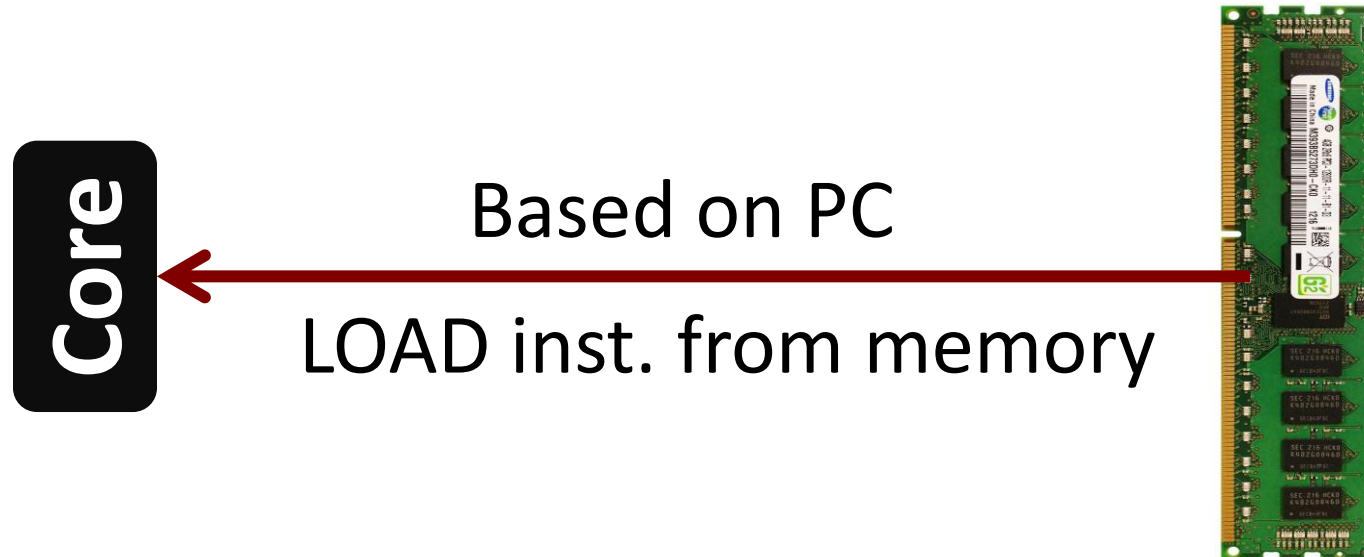
CS305: Computer Architecture

Instruction decoding

<https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html>

<https://www.cse.iitb.ac.in/~biswa/>

Why instruction decoding?



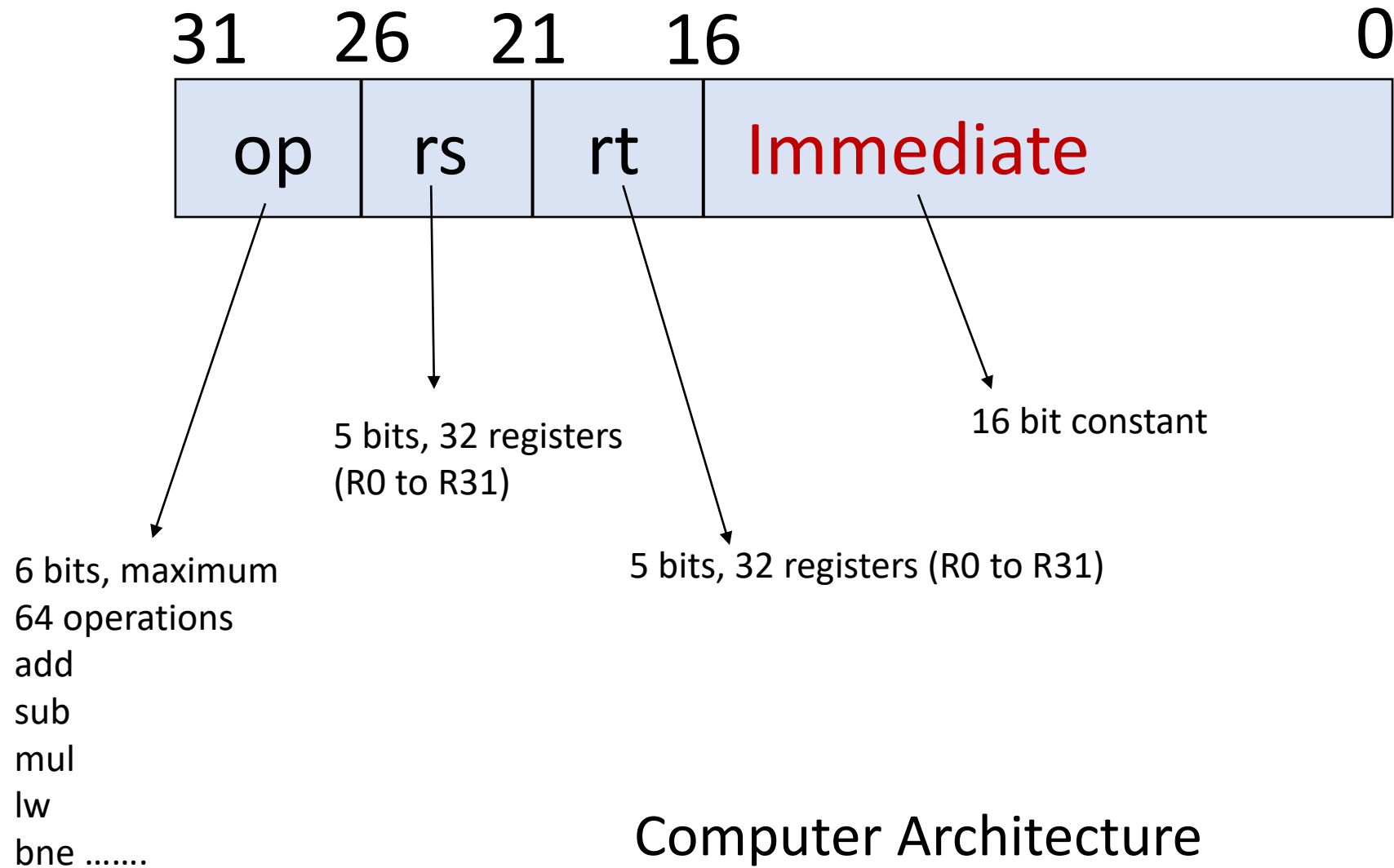
Instruction received then what?

Core

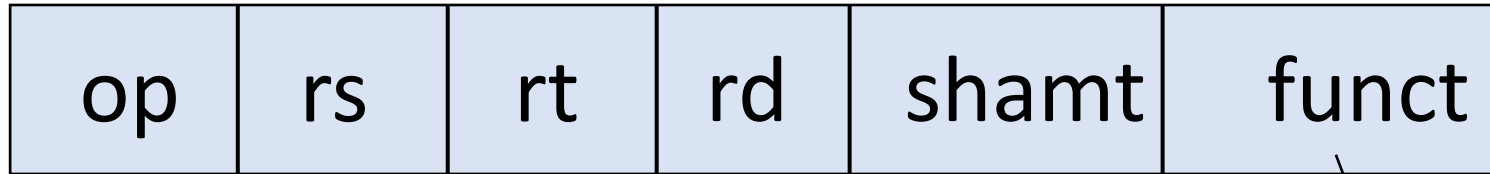
Remember instructions are of 32-bit size (in MIPS),
so PC+4

How will the processor know what to infer from these 32 bits?
Simple: Have a decoder 😊

Instruction Decoding



10K Feet View of MIPS encoding



Why this field?
Wastage of space 😞

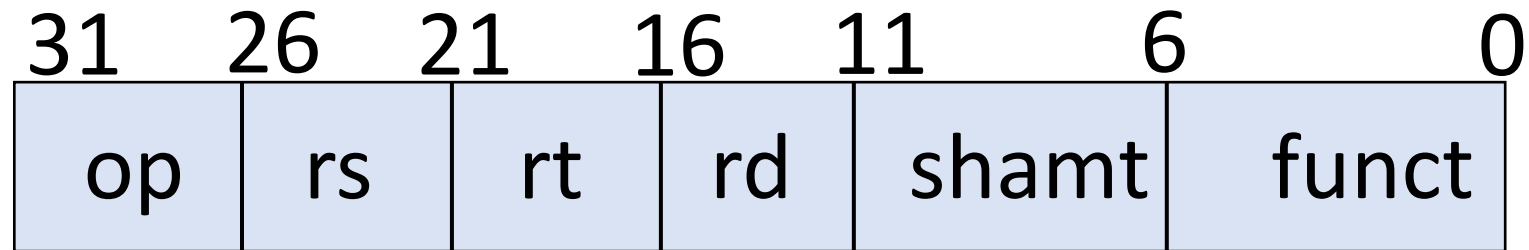
Good design demands good compromises

Instruction	Format	op	rs	rt	rd	shamt	funct	address
add	R	0	reg	reg	reg	0	32	n.a.
sub	R	0	reg	reg	reg	0	34	n.a.
addi	I	8	reg	reg	n.a.	n.a.	n.a.	constant
lw	I	35	reg	reg	n.a.	n.a.	n.a.	address
sw	I	43	reg	reg	n.a.	n.a.	n.a.	address

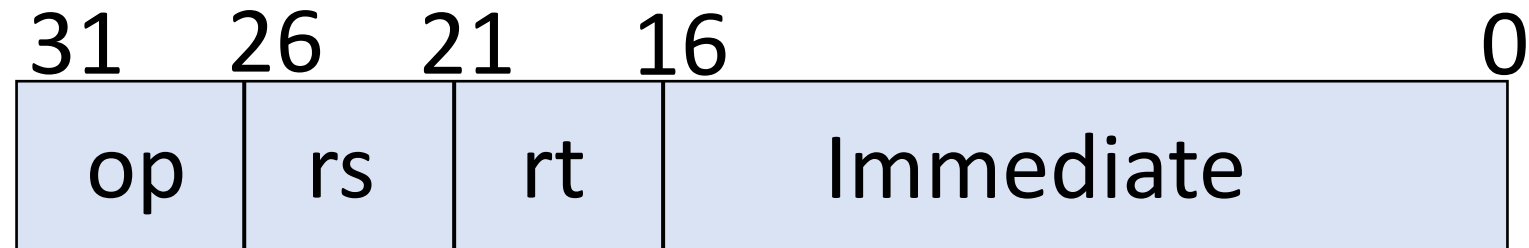


tells how to treat the last set of fields:
three fields or one field, still why funct 😞

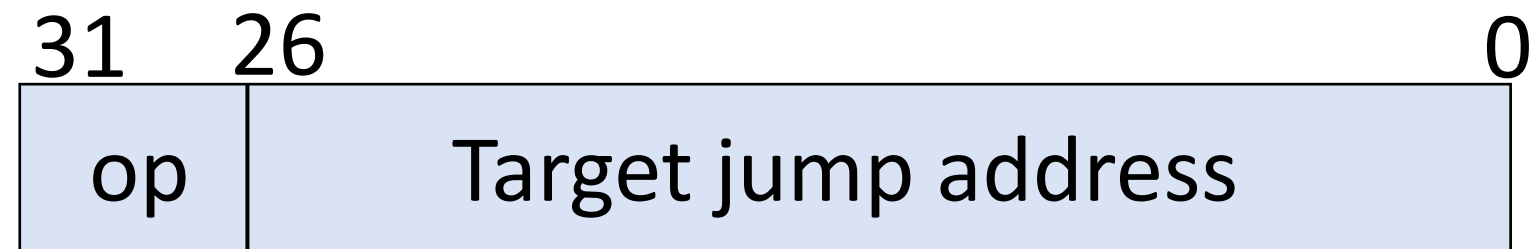
Let's have a look



R-type

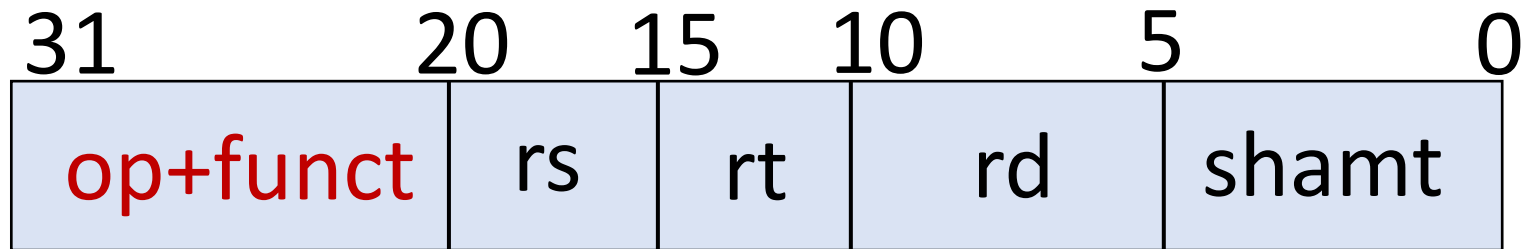


I-type

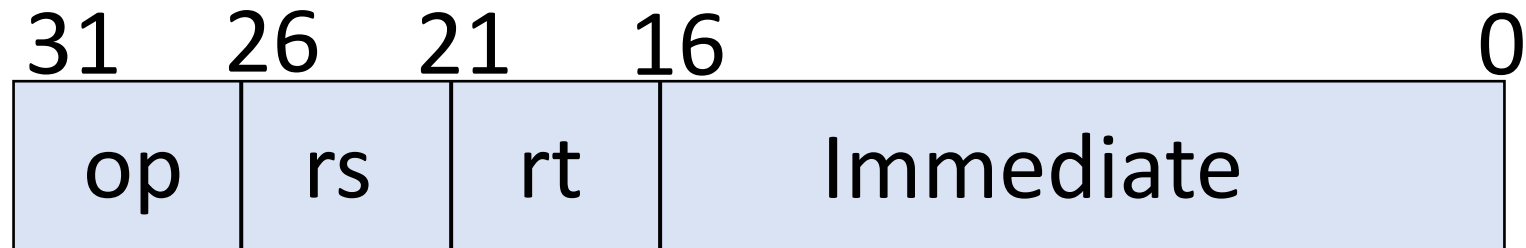


J-type

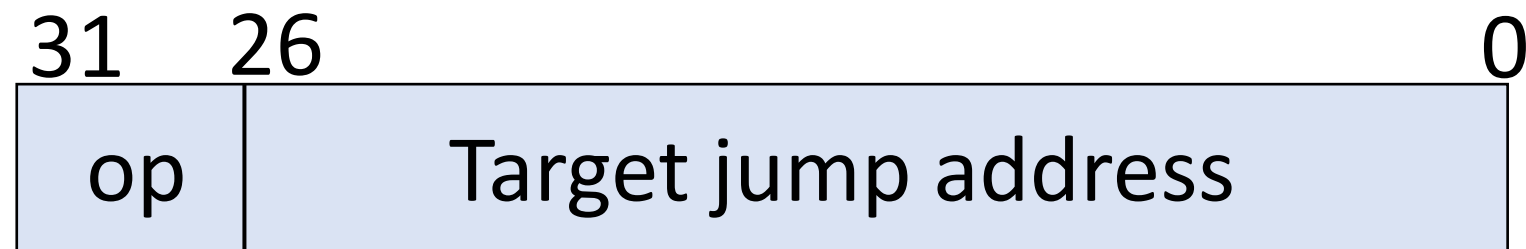
Why not?



R-type



I-type



J-type

What is a good compromise?

- **Fixed length** instructions 😊 32-bit irrespective of ops
- Fields are at the *same* or almost same location
- All formats look *similar*

Xièxiè