Instruction Decoding

2 HANDOUTS
(1 you might already have ...)

Today's lecture

- **Instruction Encoding**
  - R-type & I-type encodings
- **Instruction Decoding**
  - Operands
  - Sign-extending the immediate
  - Decoding the ALU operation

MIPS Instruction Review

- MIPS provides arithmetic and logical operations:

```
add   sub   mul   div   and   or   nor   xor
```

- These are three register instructions; for example:

```
add  $14, $18, $3      # $14 = $18 + $3
mul  $22, $22, $11     # $22 = $22 x $11
```

- For many instructions, 2nd source can be a constant:

```
addi $14, $18, 3       # $14 = $18 + 3
ori  $22, $22, 0xff    # $22 = $22 | 0xff
```

Writing an arithmetic program

- Write MIPS code to compute the following expression?

$$z = 4 + x*y - z;$$

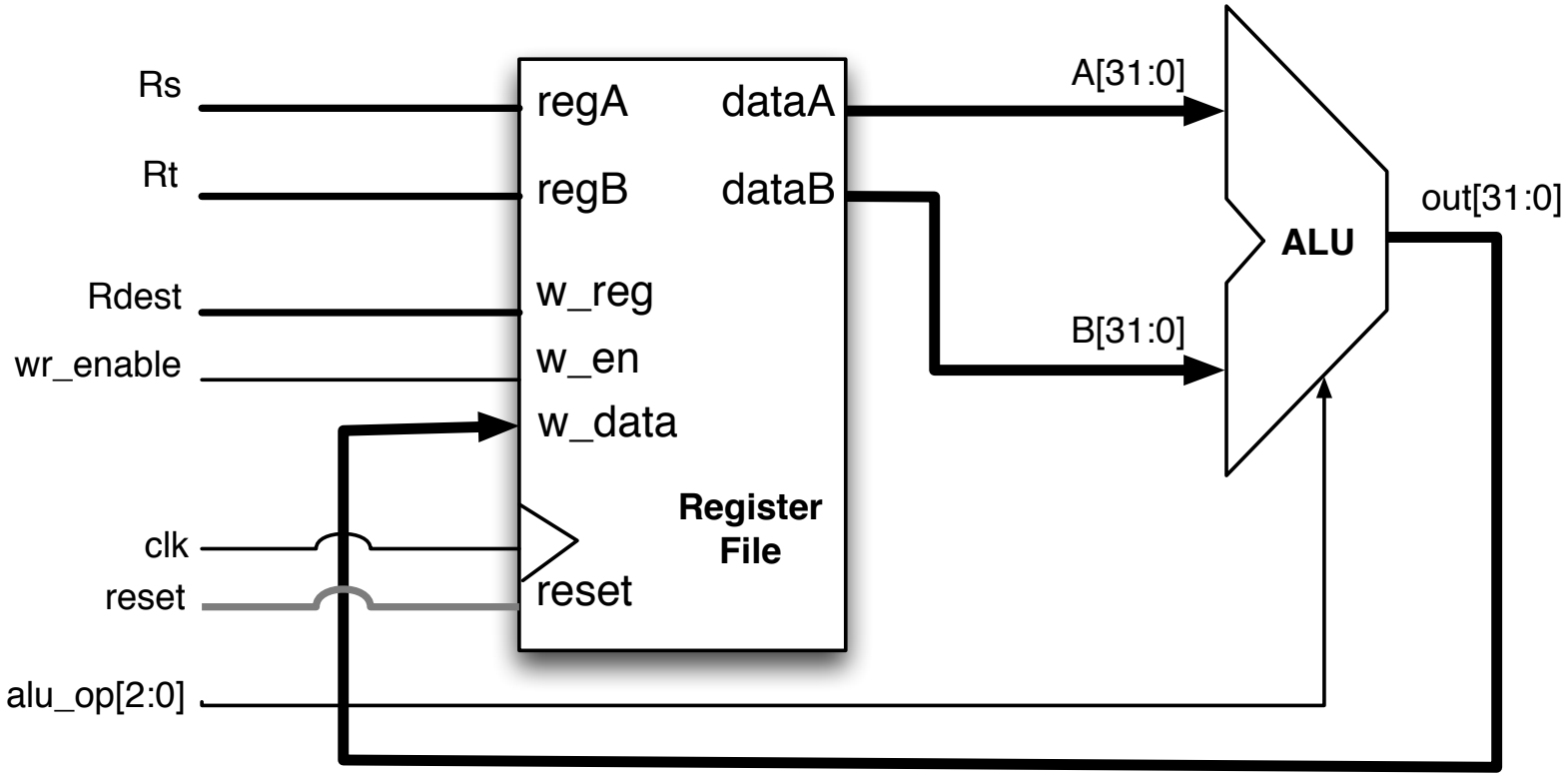
- Assume the following register allocation:
  - $13 = x, $20 = y, $15 = z

```
mul  $14, $13, $20
addi $14, $14, 4
sub  $15, $14, $15
```

01111

rd rs rt

How do instructions control the datapath?



- First step is to learn how instructions are encoded

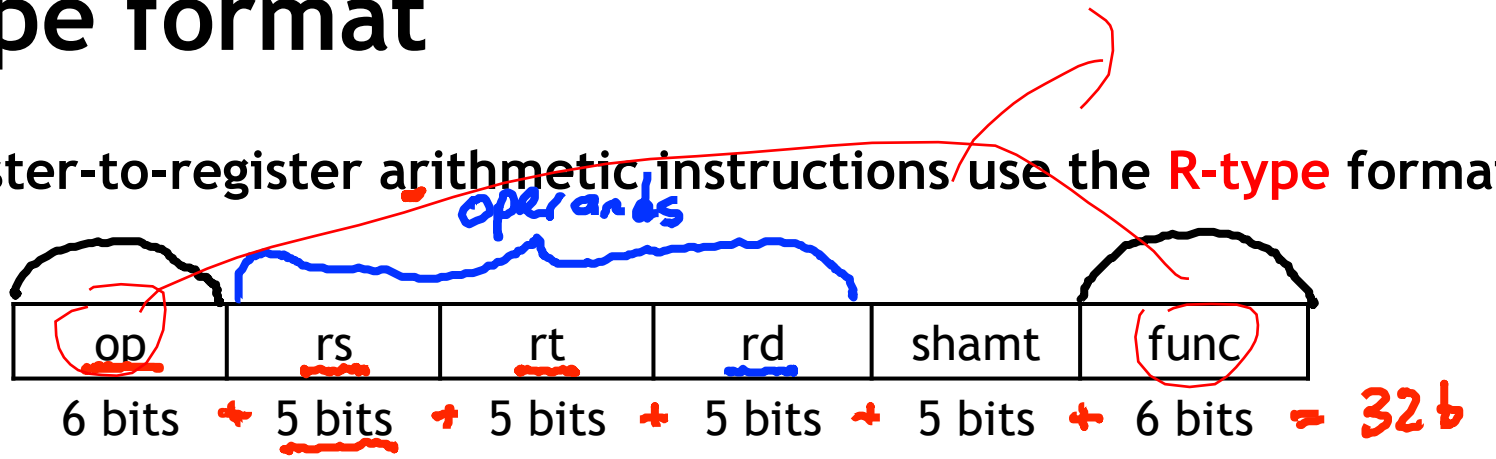
Machine Language

- **Machine language is a binary format that can be stored in memory and decoded by the CPU.**

- **MIPS machine language is designed to be easy to decode**
  - Each MIPS instruction is the same length, 32 bits. = 4B
  - There are only three different instruction formats, which are very similar to each other.
    - We'll see two of them today

R-type format

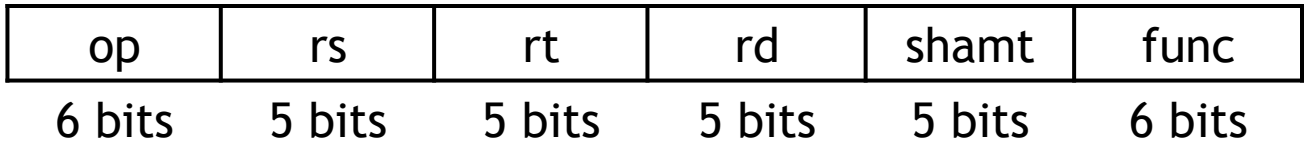
- **Register-to-register arithmetic instructions use the R-type format.**



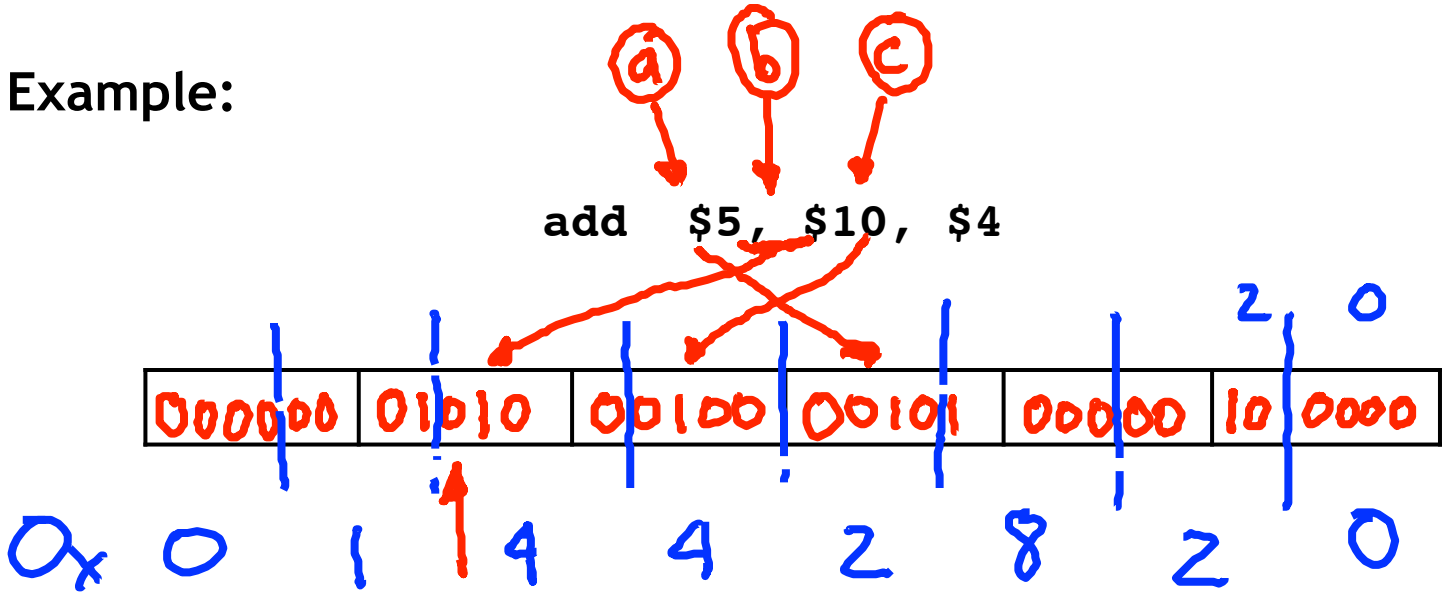
- **This format includes six different fields.**
  - op is an operation code or opcode that selects a specific operation.
  - rs and rt are the first and second source registers.
  - rd is the destination register.
  - shamt is only used for shift instructions.
  - func is used together with op to select an arithmetic instruction.

R-type format

- Register-to-register arithmetic instructions use the R-type format.



- Example:



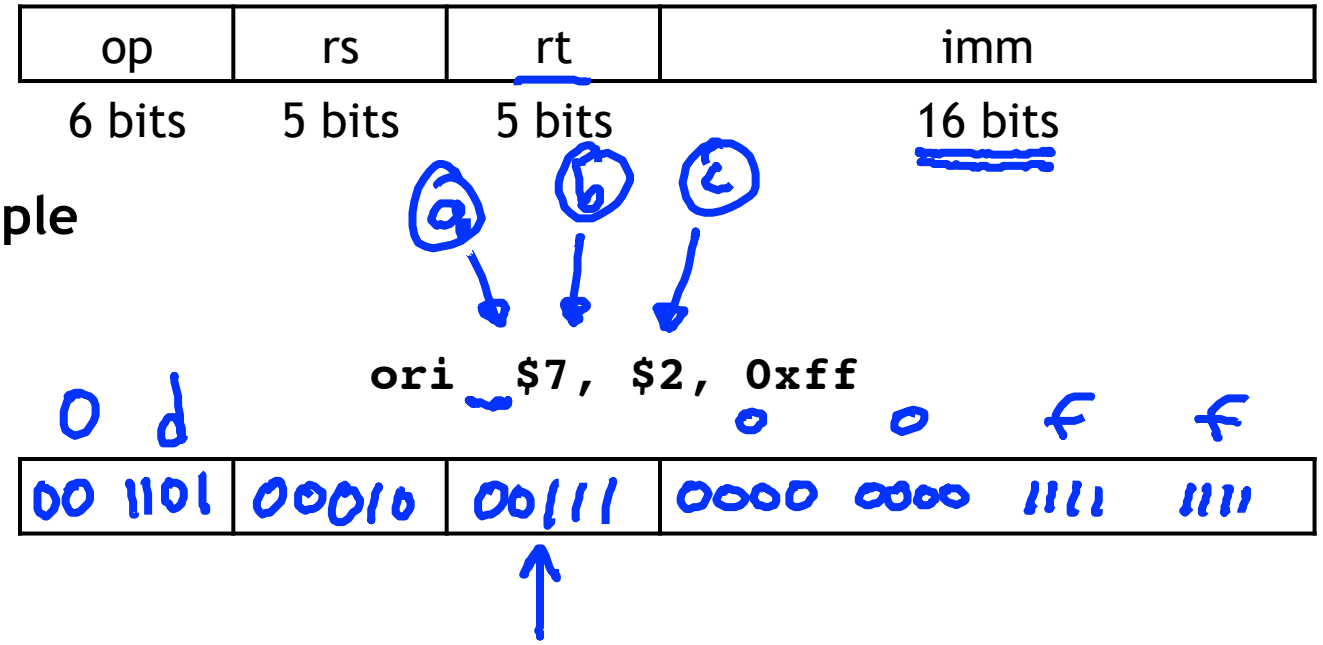
I-type format

- Instructions with immediates all use the I-type format.

- For uniformity, op, rs and rt are in the same positions as in R-type
- The meaning of the register fields depends on the exact instruction.
  - For arithmetic instructions, rt is the destination and rs a source.
- The imm field is a 16-bit signed two's-complement value.
  - It can range from -32,768 to +32,767.

I-type format

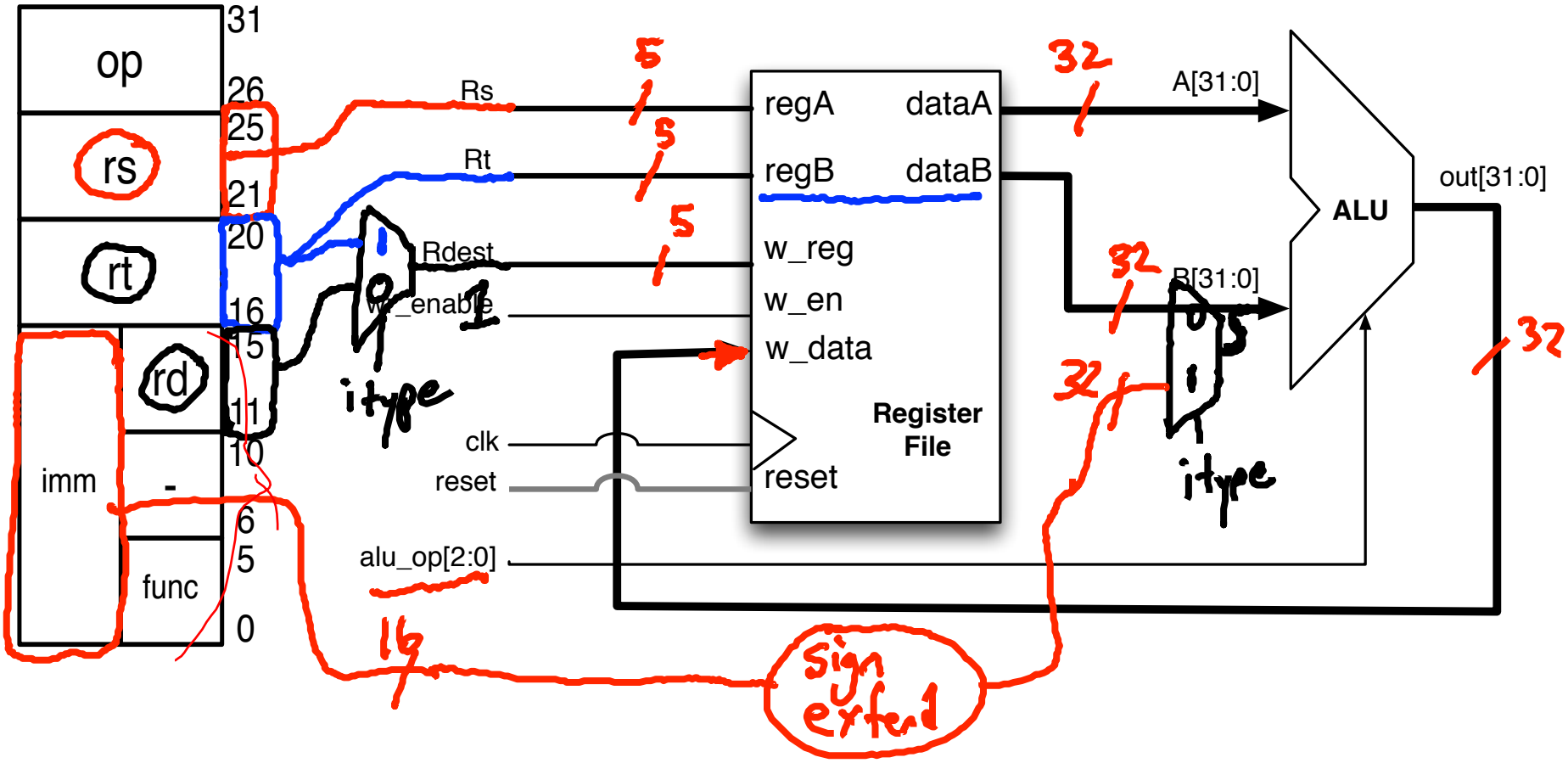
- Instructions with immediates all use the I-type format.



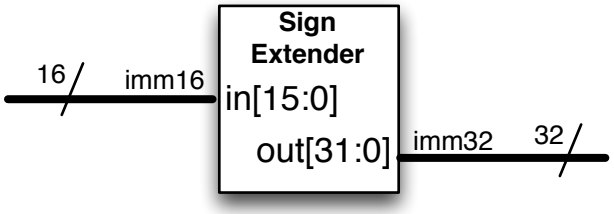
- Example

How do instructions control the datapath?

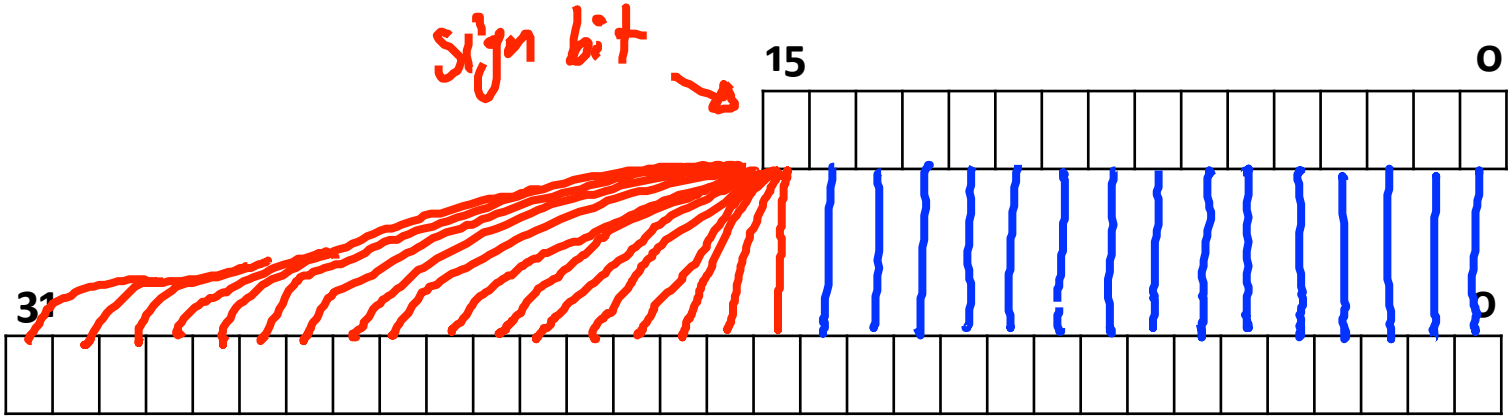
- Some of the fields come directly from instructions.



Sign Extension



- Remember how to do sign-extension?

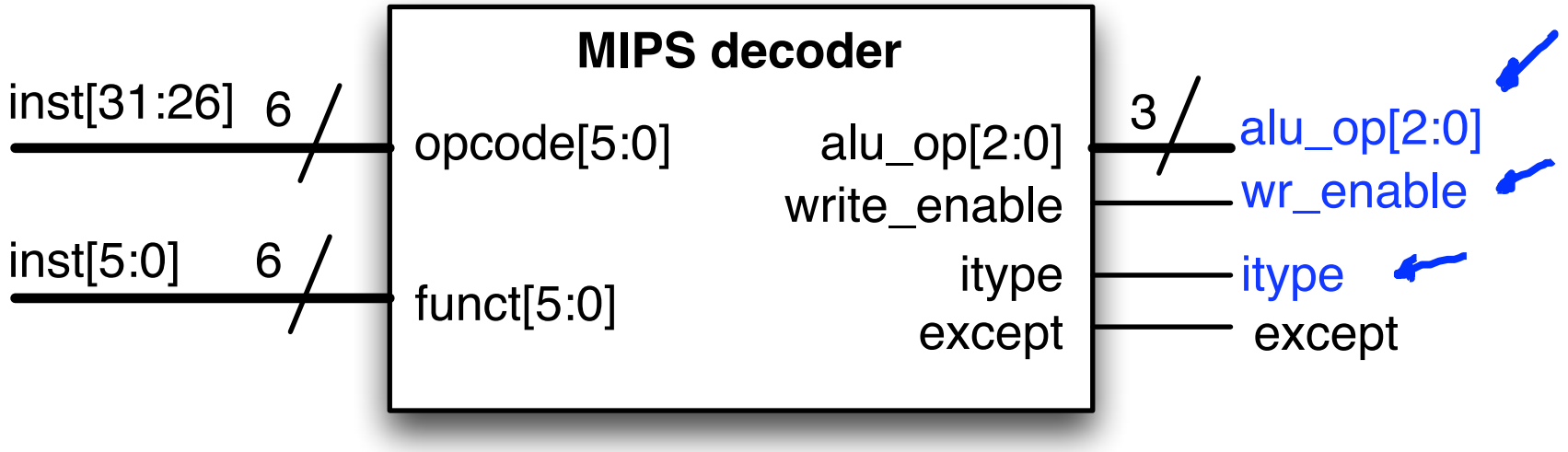


What about the alu_op?

- It is a function of the opcode / func field.

Instruction	opcode	func	alu_op	itype	wr_enable
add	0x00	0x20	add (2)	0	1
sub	0x00	0x22	sub (3)	0	1
and	0x00	0x24	and (4)	0	1
or	0x00	0x25	or (5)	0	1
xor	0x00	0x26	xor (7)	0	1
nor	0x00	0x27	nor (6)	0	1
addi	0x08	—	add (2)	1	1
andi	0x0c	—	and (4)	1	1
ori	0x0d	—	or (5)	1	1
xori	0x0e	—	xor (7)	1	1

Instruction Decoder



Arithmetic Machine Datapath