RISC V Architecture

Mahesh Awati
Department of Electronics and Communication Engg.

RISC V ARCHITECTURE

UNIT 2 – Instructions: The Language of Computer

Mahesh Awati
Department of Electronics and Communication Engineering

Instructions – Language of Computer

Representing Instructions in the Computer

I-type Instruction Format

Syntax: mnemonics rd,rs1,imm

- In these Instructions data is part of Instruction i.e., Immediate data.
- What maximum size of immediate data can be used ?????
  - Can it be 32 bit size ?
  - Can it be 5 bit size ?
- Ideally, RISC-V would have **only one instruction format** (for simplicity): unfortunately, we need to compromise
- Define **new instruction format that is mostly consistent with R-format**
- **Immediate data can be as large as possible i.e., 32 bits but there is restriction as the instruction format is of maximum 32 bit size. It should be including fields Opcode, rdes, rsrc1, funct and field for immediate data also**
- Notice if instruction has immediate, then uses at most 2 registers (one source, one destination) Ex: **addi rd, rs1, imm**

- **5-bit field only represents numbers up to the value 31: immediate may be much larger than this.**

Compare:

- ✓ **add rd, rs1, rs2**
- ✓ **addi rd, rs1, imm**

Instructions – Language of Computer

Representing Instructions in the Computer

How I-Format Instruction are Encoded ? Syntax: mnemonics rd,rs1,imm

- As Opcode field defines Instruction type and there are limited instructions which support immediate operand func3 field is sufficient.
- Therefore funct7 and rs2 fields of r-type can be used immediate field in i-type at maximum. Therefore, immediate data can be of maximum 12 bit size.
- The 12-bit immediate is interpreted as a two's complement value, so it can represent integers from -2^{11} to $2^{11}-1$.

0	000	0000	0000
0	000	0000	0001
0	111	1111	1111
1	000	0000	0000
1	000	0000	0001
1	111	1111	1111

+ve

-ve

Instructions – Language of Computer

Representing Instructions in the Computer

How I-Format Instruction are Encoded ?

Syntax: mnemonics rdes,rsrc1,imm

imm[11:0]	rs1	000	rd	0010011	addi
imm[11:0]	rs1	010	rd	0010011	slti
imm[11:0]	rs1	011	rd	0010011	sltiu
imm[11:0]	rs1	100	rd	0010011	xori
imm[11:0]	rs1	110	rd	0010011	ori
imm[11:0]	rs1	111	rd	0010011	andi
0000000 shamt	rs1	001	rd	0010011	slli
0000000 shamt	rs1	101	rd	0010011	srli
0100000 shamt	rs1	101	rd	0010011	srai

All these are different Instructions supporting I Format

One of the higher-order immediate bits is used to distinguish “shift right logical” (SRLI) from “shift right arithmetic” (SRAI)

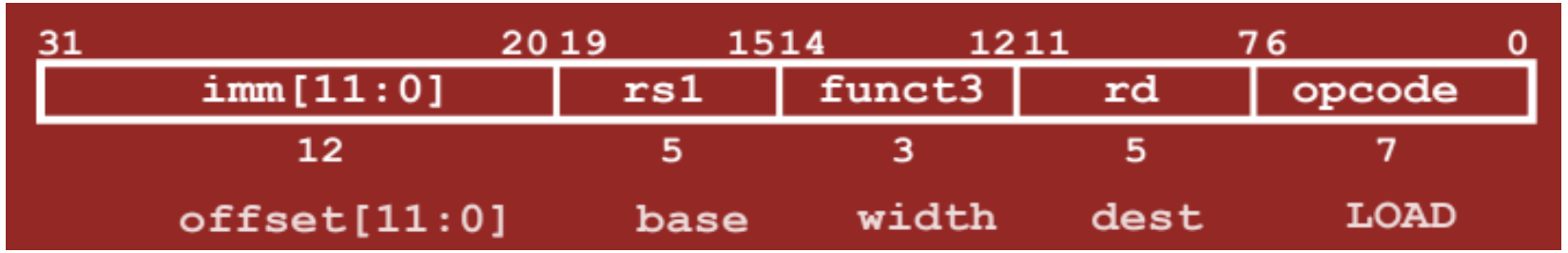
“Shift-by-immediate” instructions only use lower 5 bits of the immediate value for shift amount (can only shift by 0-31 bit positions)

Instructions – Language of Computer

Representing Instructions in the Computer

How I-Format Instruction used for load

Syntax: mnemonics rd, imm(rs1)



Syntax: mnemonics rd, imm (rs1)

lw

rs1 holds the base address of Memory location

Destination Register in which data from Memory address=base address + immediate offset

12 bit Immediate offset

Instructions – Language of Computer

Representing Instructions in the Computer

How I-Format Instruction used for load

Syntax: mnemonics rd, imm(rs1)

imm[11:0]	rs1	000	rd	0000011	lb
imm[11:0]	rs1	001	rd	0000011	lh
imm[11:0]	rs1	010	rd	0000011	lw
imm[11:0]	rs1	100	rd	0000011	lbu
imm[11:0]	rs1	101	rd	0000011	lhu

imm[11:0]	rs1	010	rd	0000011	lw
0000 0010 0000	1011 0	010	0100 1	000 0011	

Hexadecimal Representation: 0x020B2483

lw x9, 32(x22) - Load word

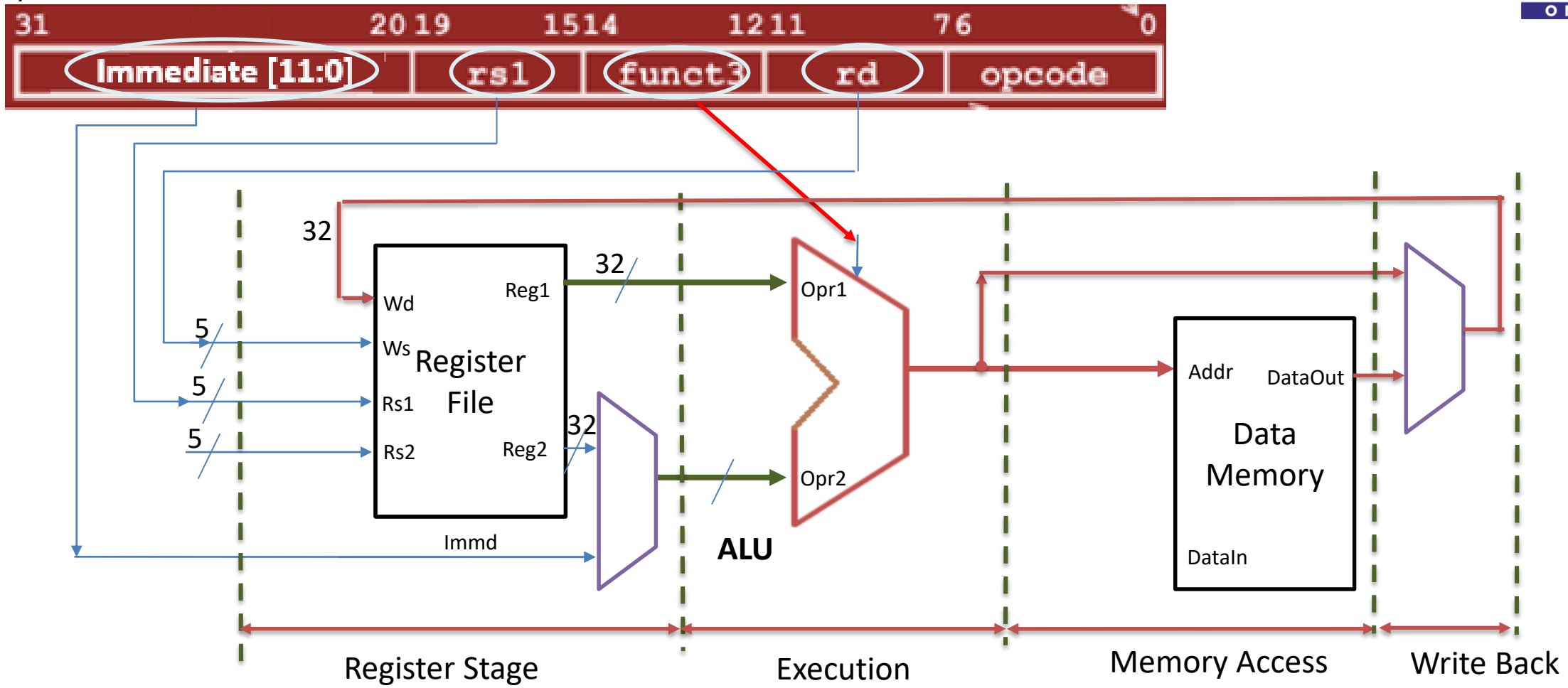
Here,

- 22 (for x22) is placed in the **rs1 field**,
- 32 is placed in the **immediate field**, and
- 9 (for x9) is placed in the **rd field.**

Instructions – Language of Computer

I-type Instruction Data Path

Anything that stores data or operates on data within a processor is called data path.



THANK YOU

Mahesh Awati
Department of Electronics and Communication

mahesha@pes.edu

+91 9741172822