



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

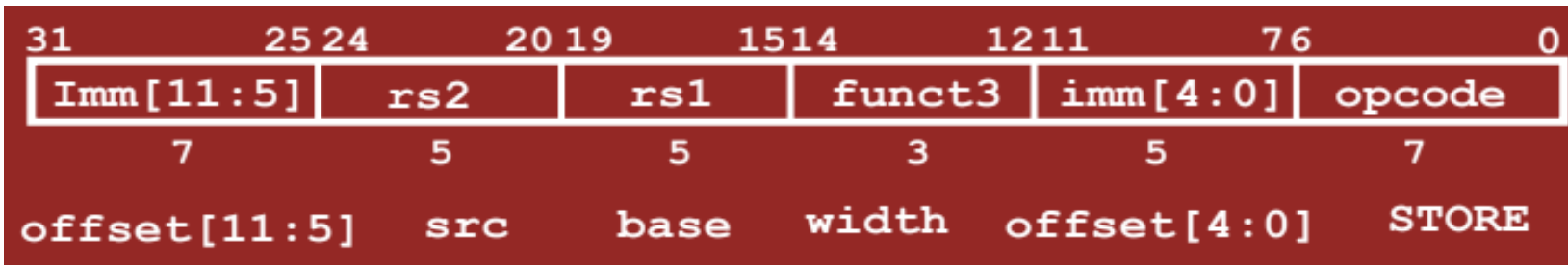
Instructions – Language of Computer

Representing Instructions in the Computer

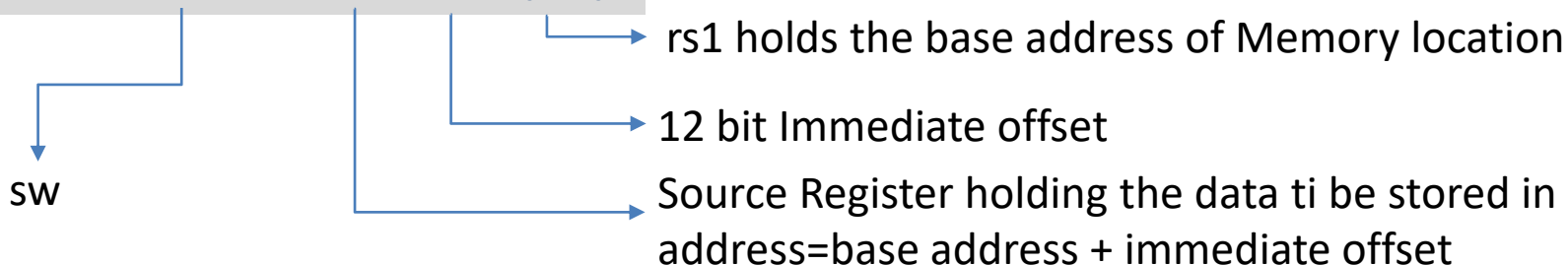
S-type Instruction Format

Syntax: mnemonics rs2,imm(rs1)

- Store needs to read two registers, rs1 for base memory address, and rs2 for data to be stored, as well immediate offset!
- Can't have both rs2 and immediate in same place as other instructions!
- Note that stores don't write a value to the register file, no rd!
- RISC-V design decision is to move low 5 bits of immediate to where rd field was in other instructions – keep rs1/rs2 fields in same place



Syntax: mnemonics rs2, imm (rs1)

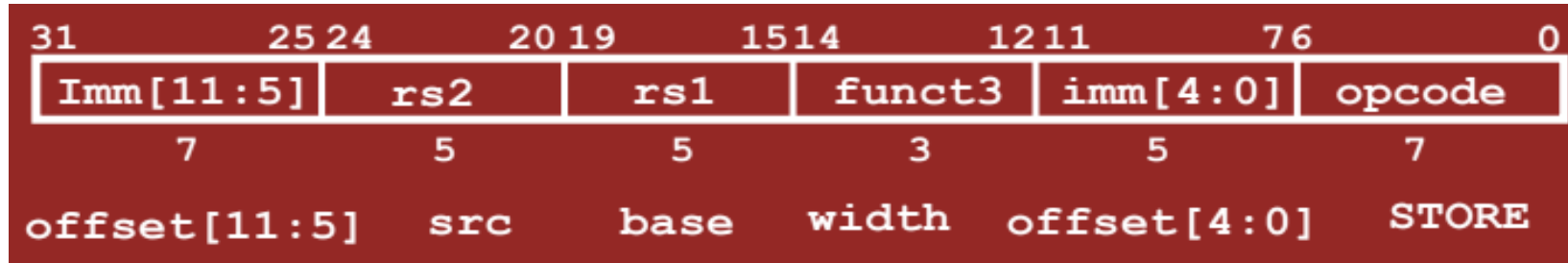


Instructions – Language of Computer

Representing Instructions in the Computer

S-type Instruction Format

Syntax: mnemonics rs2,imm(rs1)



sw x14, 8(x2)



Imm₁₂: 0000 000 0 1000 (08)

Hexadecimal Representation: 0x00E12423

Instructions – Language of Computer

Representing Instructions in the Computer

S-type Instruction Format

Syntax: mnemonics rs2,imm(rs1)

- Store byte, half-word, word

Imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	sb
Imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	sh
Imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	sw
width						



Instructions – Language of Computer

Representing Instructions in the Computer

- Translating RISC-V Assembly Language into Machine Language
- If x10 has the base of the array A and x21 corresponds to h, the assignment statement

$A[30] = h + A[30] + 1$; is compiled into

```
lw x9, 120(x10)           // Temporary reg x9 gets A[30]
add x9, x21, x9            // Temporary reg x9 gets h+A[30]
addi x9, x9, 1             // Temporary reg x9 gets h+A[30]+1
sw x9, 120(x10)           // Stores h+A[30]+1 back into A[30]
```

What is the RISC-V machine language code for these three instructions?

Since $120_{\text{ten}} = 0000011\ 11000_{\text{two}}$, the binary equivalent to the decimal form is:

immediate	rs1	funct3	rd	opcode
000011110000	01010	010	01001	0000011

funct7	rs2	rs1	funct3	rd	opcode
0000000	01001	10101	000	01001	0110011

immediate	rs1	funct3	rd	opcode
000000000001	01001	000	01001	0010011

immediate[11:5]	rs2	rs1	funct3	immediate[4:0]	opcode
0000011	01001	01010	010	11000	0100011



THANK YOU

Mahesh Awati

Department of Electronics and Communication

mahasha@pes.edu

+91 9741172822