RISC V Architecture

Dr. Santhameena.S

Department of Electronics and Communication Engineering

RISC V Architecture

UNIT 3: Instructions: The Language of Computer

Dr. Santhameena.S
Department of Electronics and Communication Engineering

Instructions – Language of Computer

Synchronization

- Two processors sharing an area of memory
  - P1 writes, then P2 reads
  - Data race if P1 and P2 don’t synchronize
    - Result depends of order of accesses
- Hardware support required
  - Atomic read/write memory operation
  - No other access to the location allowed between the read and write
- Could be a single instruction
  - E.g., atomic swap of register ↔ memory
  - Or an atomic pair of instructions

Instructions – Language of Computer

Synchronization in RISC-V

- Load reserved: lr.d rd,(rs1)
  - Load from address in rs1 to rd
  - Place reservation on memory address
- Store conditional: sc.d rd,rs2,(rs1),
- Store from rs2 to address in rs1
  - Succeeds if location not changed since the lr.d
    - Returns 0 in rd
  - Fails if location is changed
    - Returns non-zero value in rd

Instructions – Language of Computer

Synchronization in RISC-V

- Example 1: **atomic swap** (to test/set lock variable)

```
again:      lr.w x10,(x20)
            sc.w x11, x23,(x20) // X11 = status
            bne  x11,x0,again  // branch if store failed
            addi x23,x10,0     // X23 = loaded value
```

- Example 2: **lock**

```
            addi x12,x0,1              // copy locked value
again:      lr.w x10,(x20)             // read lock
            bne  x10,x0,again          // check if it is 0 yet
            sc.w x11, x12,(x20)        // attempt to store
            bne  x11,x0,again          // branch if fails
```

- **Unlock:**

```
            sw   x0,0(x20)             // free lock
```

THANK YOU

Dr. Santhameena.S
Department of Electronics and Communication Engineering
santhameena.s@pes.edu