Instruction Set Classification: RISC vs CISC

1. CISC: Complex Instruction Set Computer

- Large number of instruction formats in the Instruction Set Architecture (ISA).
- Multiple addressing schemes and instruction types.
- Varying instruction sizes.
- Instructions can take multiple cycles to execute.

2. RISC: Reduced Instruction Set Computer

- Small number of fast and simple instructions.
- Fixed instruction format and fixed instruction length.
- Most instructions execute in a single clock cycle.
- Uses a large number of general-purpose registers.
- Typical of load/store architectures (only load/store instructions access memory).

3. Examples of Architectures

- **CISC**: VAX, Intel x86, Motorola 680x0
- **RISC**: MIPS, DEC Alpha, Sun SPARC, IBM 801, ARM6

4. RISC vs CISC Comparison

Feature	**CISC**	**RISC**
Instruction Length	Variable	Fixed (single word)
Instruction Format	Many formats	Few formats
Memory Access	Operands can be memory locations	Load/store only
Operations	Complex	Simple
Execution Time	Multiple cycles	Mostly single cycle

5. Instruction Format

General Format

An instruction specifies:

- The operation to be performed (opcode).
- The set of operands involved in the operation:

  - Input data
  - Destination/result

In assembly language, this format is generally:

```
op X1, X2, ..., Xn
```

Where:

- `op`: operation code (opcode)
- `X1, X2, ..., Xn`: operands (registers or memory addresses)

6. RISC Instruction Format (Example: RISC1)

- Memory access is restricted to `load` and `store` instructions.
- Most instructions use register operands only, making it possible to encode them in a single word.

Register-to-Register Operation

```
Rd := F(Rs, S2)
```

Where:

- `Rd`: Destination register
- `Rs`: First source register
- `S2`: Second source (register or immediate value)

Note: If bit 13 of the instruction is 1, `S2` is treated as an immediate constant.