🧾 RISC-V Vector Instruction Suffix Cheat Sheet

🔍 Understanding the Suffixes

The suffixes like `.x.s`, `.s.x`, `.v.x`, `.v.v`, etc., describe the **types of operands** used in a vector instruction. They tell you:

- Whether each operand is a **vector register**, **scalar integer**, **scalar float**, or **immediate**.
- The **order of suffixes matches** the order of operands.

📘 Suffix Table

Suffix	Operand Type	Meaning Example
`.x`	Scalar **integer** register (from x-register)	`vmv.s.x` — scalar → vector
`.f`	Scalar **float** register (from f-register)	`vfmv.s.f` — float → vector
`.s`	Scalar from **vector element 0**	`vmv.x.s` — vector → scalar
`.v`	Full **vector register**	`vadd.vv` — vector + vector
`.vx`	Vector + scalar **integer**	`vslideup.vx`
`.vf`	Vector + scalar **float**	`vfslide1up.vf`
`.vi`	Vector + **immediate**	`vslideup.vi`
`.vv`	Vector + vector	`vrgather.vv`
`.vm`	Uses a **mask register**	`vcompress.vm`

📌 Examples

- `vmv.x.s rd, vs2` → `x[rd] = vs2[0]`
- `vmv.s.x vd, rs1` → `vd[0] = x[rs1]`
- `vslideup.vx vd, vs2, rs1` → slides elements up by `x[rs1]`
- `vrgather.vi vd, vs2, 5` → all lanes gather `vs2[5]`

The **suffix guides your understanding of operand types** and is essential for correct instruction usage.