🧾 RISC-V Vector Cheat Sheet — `vsetvli` Example

🔧 Instruction Format

```
vsetvli rd, rs1, sew
```

- **`rd`** : destination register to hold **VL** (vector length)
- **`rs1`** : application vector length (AVL) — number of elements requested
- **`sew`** : Standard Element Width (e.g., `e8` , `e16` , `e32`)

📚 Acronyms and Terms

Term	Meaning
VLEN	Max bits in a hardware vector register (e.g., 128, 256, 512 bits)
SEW	Standard Element Width in bits (`e8` = 8-bit, `e32` = 32-bit, etc.)
LMUL	Register grouping factor (e.g., `m1` , `m2` , `m4`); default is `m1`
VL	Vector Length: number of elements that can be processed in one op
AVL	Application Vector Length: number of elements the program wants to use

🔢 Worked Example

Assume:

- **VLEN = 128 bits**
- **SEW = e8 (8-bit elements)**
- **LMUL = m1 (default)**
- **a2 = 50** (bytes to copy)

Step-by-step:

1. Max elements supported by hardware:

   ```
   Max Elements = VLEN / SEW = 128 / 8 = 16
   ```

2. AVL (application vector length) = 50
3. Final VL = `min(AVL, Max Elements) = min(50, 16)` = **16**

So this:

```
vsetvli t0, a2, e8
```

sets:

- `VL = 16`
- `t0 = 16`

💡 Summary Table

`a2` (AVL)	SEW	VLEN	VL (Result)
50	e8	128	16
5	e8	128	5
30	e16	128	8

100	e32	128	4