RISC-V Vector寄存器[1]图解

@sunshaoce

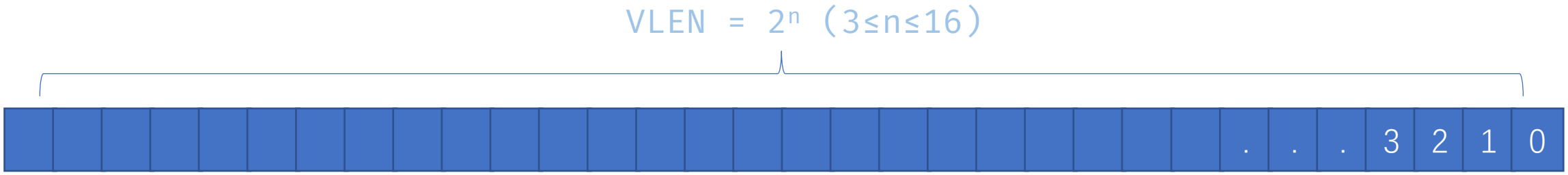
$$VLMAX = LMUL \times VLEN \div SEW$$

在一个硬件线程中（hart，hardware thread）中

一个向量寄存器（VR，Vector Register），

VLMAX = LMUL × VLEN ÷ SEW

总位数为VLEN（The number of bits in a single vector register）。



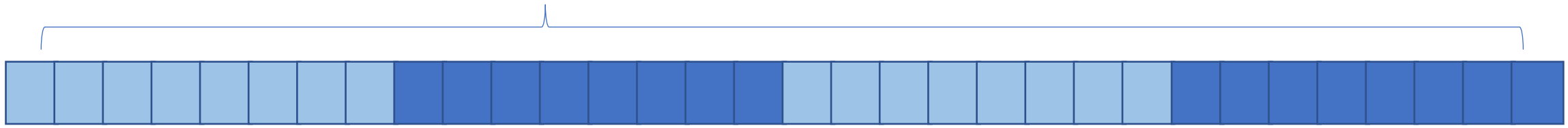
VLMAX = LMUL × VLEN ÷ SEW

一个向量元素的最大位宽为ELEN（The maximum size in bits of a vector element ），

$ELEN=2^n, 3 \le n \le Log_2 VLEN$

指定多个位存储一个元素（Element），

然后该位宽称为SEW（Vector Selected Element Width），最小值为8。



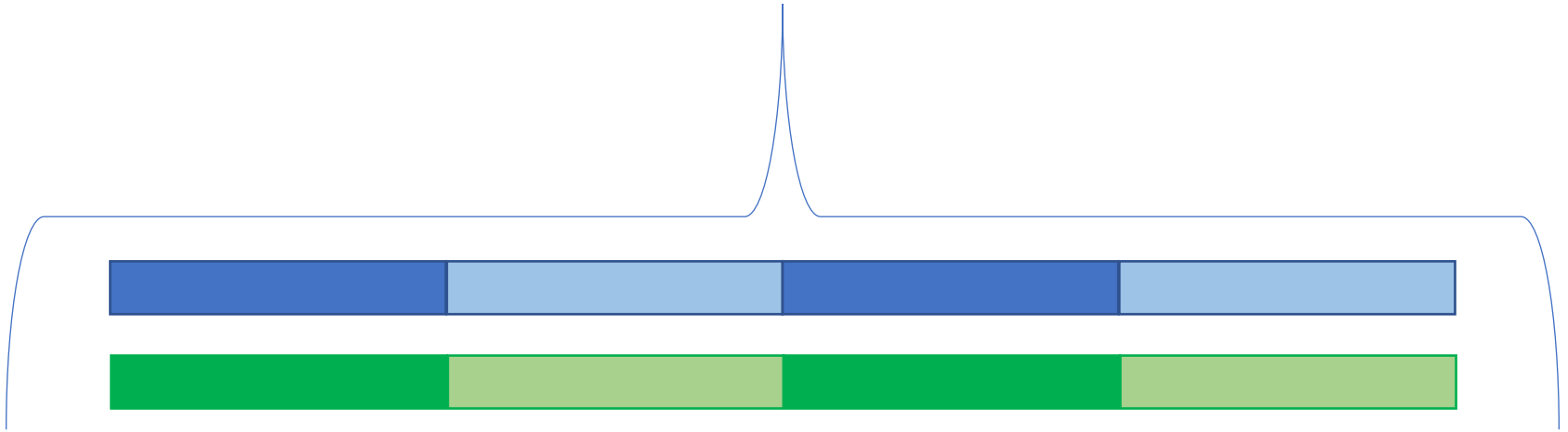
VLMAX = LMUL × VLEN ÷ SEW

有时候，一个向量寄存器不够用，
我们将多个寄存器合并为
寄存器组（Register Group），
合并的个数称为
LMUL（Vector register group multiplier ）。
可以的取值为1，2，4，8。

可以合并，也可以分开，
LMUL也可以是分数，用以表示
将寄存器一分为几份，
可以的取值为：1/2, 1/4, 1/8。
LMUL应确保$LMUL \geq SEW_{MIN}/ELEN$，
使元素能够被正常存取。

VLMAX = LMUL × VLEN ÷ SEW

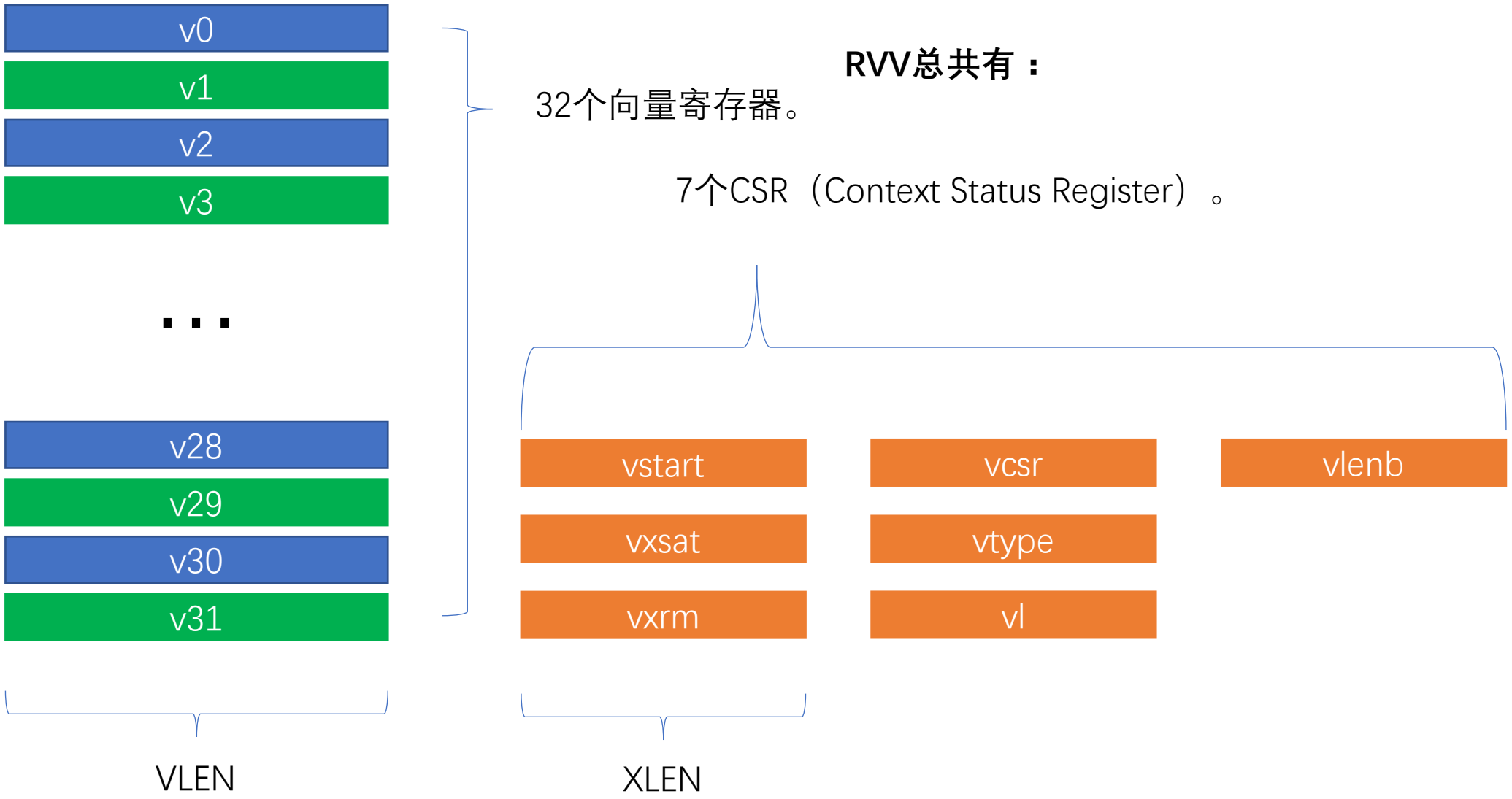
然后，向量个数为vl（Vector Length），
而向量寄存器（组）中能存储的最大向量个数vlmax（Vector Length Maximum）



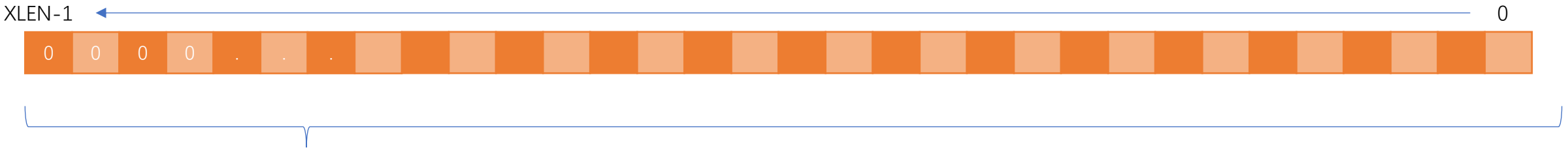
其中计算公式为：

$$VLMAX = LMUL \times VLEN \div SEW$$

vstart, vxsat, vxrm, vcsr,
vtype, vl, vlenb



vstart, vxsat, vxrm, vcsr, vtype, vl, vlenb

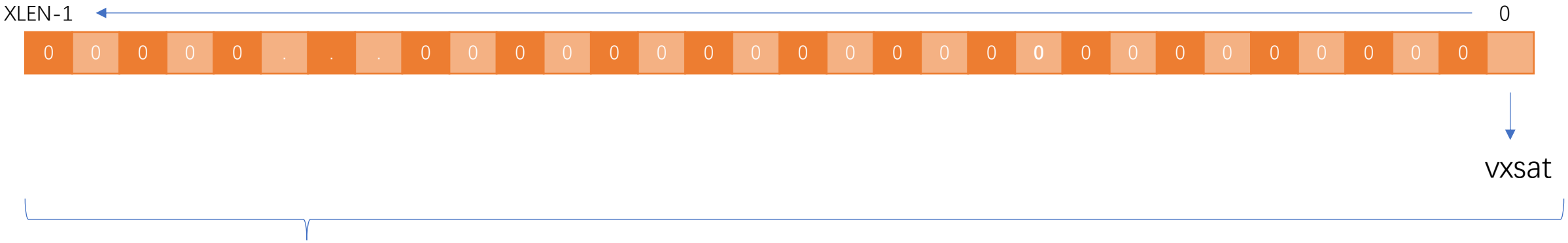


vstart（Vector Start）寄存器，
存储向量指令中被执行的第一个元素的索引值。

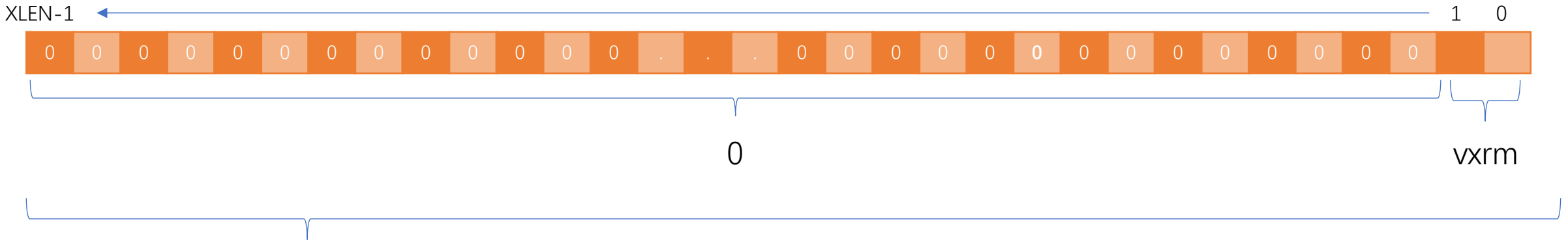
vstart可读写的位数，我们根据极端情况来计算，
即$LMUL_{MAX}=8$，$SEW_{MIN}=8$的情况：

$$VLMAX_{MAX} = LMUL_{MAX} \times VLEN \div SEW_{MIN} = VLEN$$

即需要$\log_2 VLEN$位来存储，而其他没有用到的位编码为0。

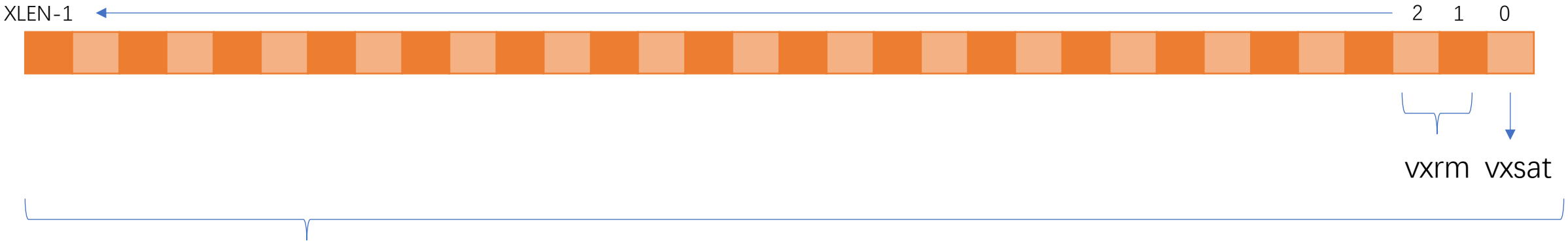


vsat（Vector Fixed-Point Saturation Flag）向量定点饱和标志

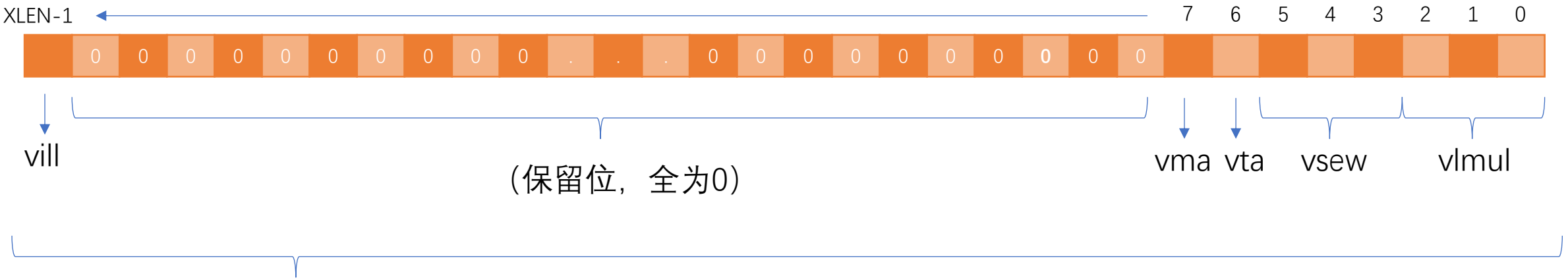


vxrm（Vector Fixed-Point Rouding Mode）向量定点取整模式

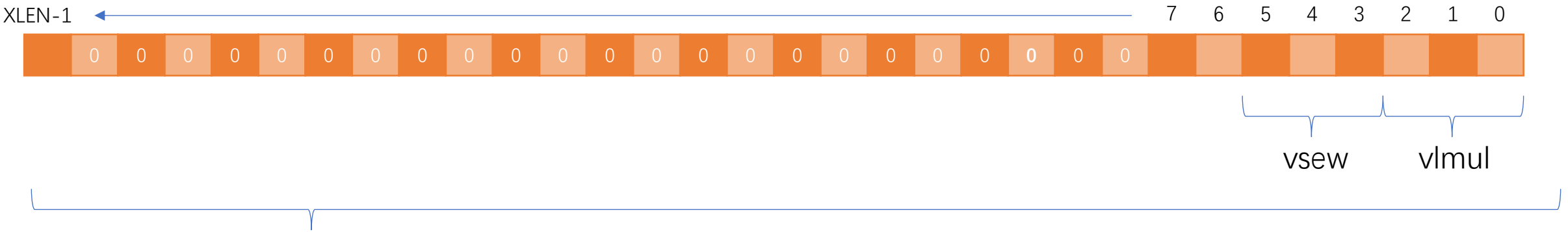
1	0	缩写	取整模式
0	0	rnu	向上取整
0	1	rne	取到最近的偶数
1	0	rdn	截断
1	1	rod	取到最近的奇数



vcsr（Vector Control and Status Register）寄存器，
有vxrm和vsat的镜像表示



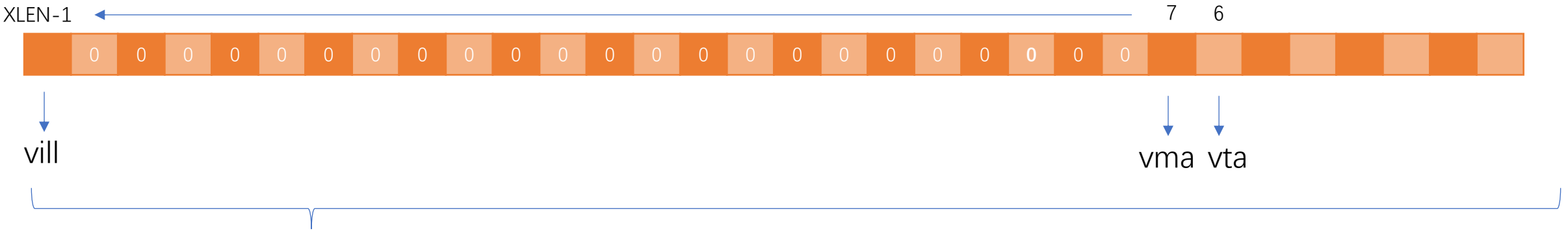
vtype（Vector Type）寄存器，只能被 `vset{i}vl{i}` 指令修改值。



vtype（Vector Type）寄存器，vsew和vlmul为位域（filed）。$SEW=2^{3+vsew[2:0]}$。$LMUL=2^{vlmul[2:0]}$

5	4	3	SEW
0	0	0	8
0	0	1	16
0	1	0	32
0	1	1	64
1	x	x	Reserved

2	1	0	LMUL	#groups	VLMAX
1	0	0	-	-	-
1	0	1	1/8	32	VLEN/SEW/8
1	1	0	1/4	32	VLEN/SEW/4
1	1	1	1/2	32	VLEN/SEW/2
0	0	0	1	32	VLEN/SEW
0	0	1	2	16	2*VLEN/SEW
0	1	0	4	8	4*VLEN/SEW
0	1	1	8	4	8*VLEN/SEW



vtype（Vector Type）寄存器。

vma：Vector mask agnostic
vta ：Vector tail agnostic

vill	含义
1	非法指令

vma	vta	Inactive Elements	Tail Elements
0	0	保持（undisturbed）	保持
0	1	保持	模糊
1	0	模糊（agnostic）	保持
1	1	模糊	模糊

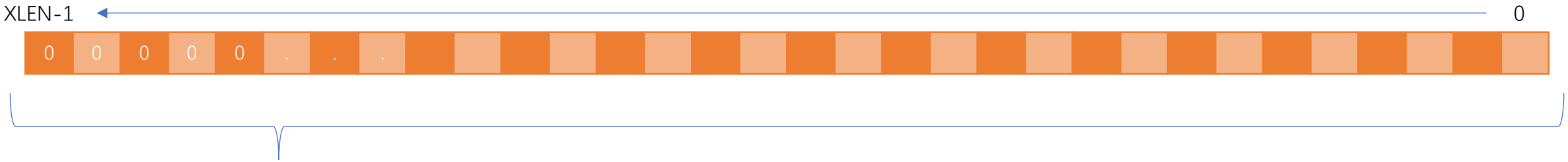
vstart, vxsat, vxrm, vcsr, **vtype,** vl, vlenb

vl（Vector Length）寄存器，记录一天向量指令修改多少向量。

vl的位宽计算，例如：

VLEN=32，SEW=8，LMUL=8时，VLMAX=32

所以，vl需要保存0~32中的某一个数，就需要能存下33个数，因而至少需要6位。



vlenb（VLEN/8， vector register length in bytes）寄存器。
只读常量，存储的是VLEN/8的值，也就是一个向量寄存器的最大字节数。

3

$$\begin{vmatrix} 1 \\ 3 \\ 5 \\ 7 \end{vmatrix} + \begin{vmatrix} 0 \\ 2 \\ 4 \\ 8 \end{vmatrix} = \begin{vmatrix} 1 \\ 5 \\ 9 \\ 15 \end{vmatrix}$$

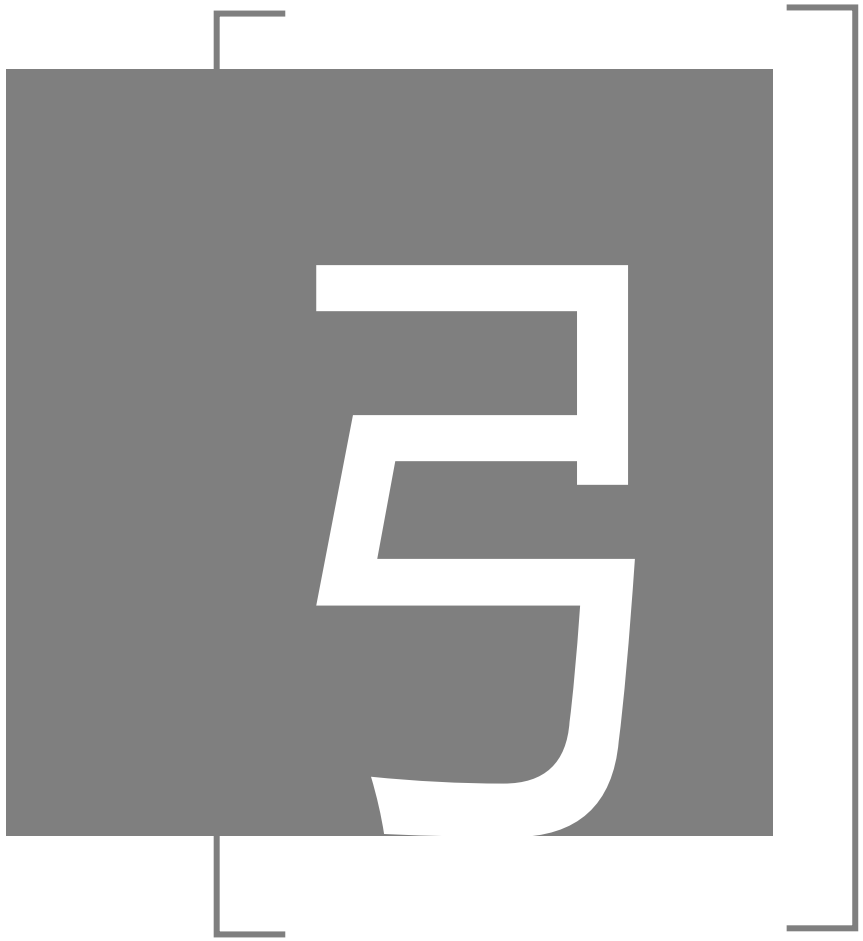
```c
#include <assert.h>
#include <riscv_vector.h>
#include <stdio.h>
#define N 4
int main() {
  vint32m1_t va, vb, vc;
  int vlmax = vsetvlmax_e32m1();
  assert(N <= vlmax);

  int32_t ia[N] = {1, 3, 5, 7};
  int32_t ib[N] = {0, 2, 4, 8};
  int32_t ic[N] = {0, 0, 0, 0};
  va = vle32_v_i32m1(ia, N);
  vb = vle32_v_i32m1(ib, N);
  vc = vadd_vv_i32m1(va, vb, N);
  vse32_v_i32m1(ic, vc, N);

  for (unsigned i = 0; i < N; ++i)
    printf("%d ", ic[i]);
  printf("\n");
  return 0;
}
```

int ; sew = 32 ; lmul = 1

vsetvli s0, zero, e32, m1, ta, mu

int ; sew = 32 ; lmul = 1 ; vl = N = 4

$$\begin{vmatrix}1\\3\\5\\7\end{vmatrix} + \begin{vmatrix}0\\2\\4\\8\end{vmatrix} = \begin{vmatrix}1\\5\\9\\15\end{vmatrix}$$



[1] 参考自RISC-V V Spec *(Version 1.0),*
https://github.com/riscv/riscv-v-spec/releases/tag/v1.0

[2] riscv-v-spec-1.0（矢量指令） 学习理解（1-5 & 18 segment）
https://blog.csdn.net/weixin_43348382/article/details/113736632