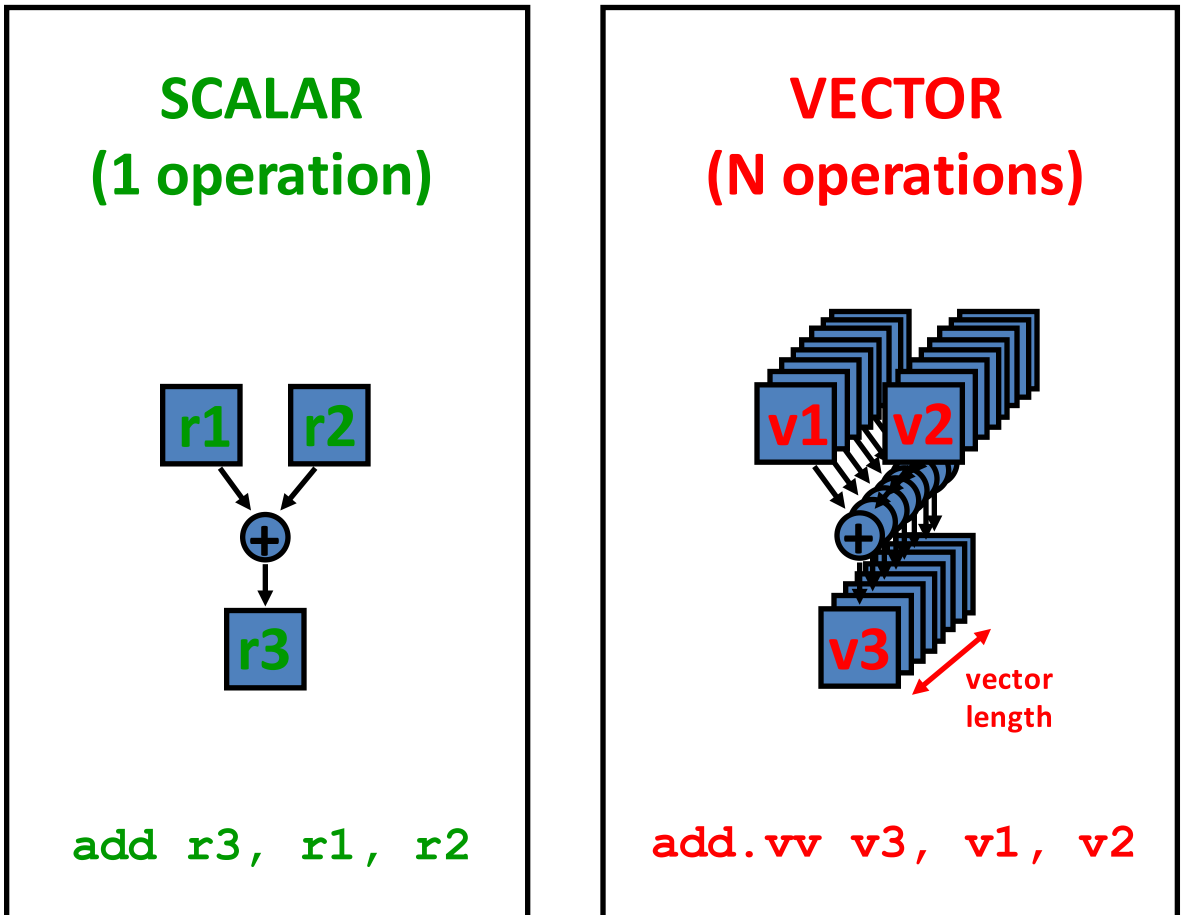
Vector Processing

Alternative Model:Vector Processing

- Vector processors have high-level operations that work on linear arrays of numbers: "vectors"



3.4.1 Characteristics of Vector Processing

A vector operand contains an ordered set of n elements, where n is called the *length* of the vector. Each element in a vector is a scalar quantity, which may be a floating-point number, an integer, a logical value, or a character (byte). Vector instructions can be classified into four primitive types:

$$
\begin{aligned}
f_1 &: V \rightarrow V \\
f_2 &: V \rightarrow S \\
f_3 &: V \times V \rightarrow V \\
f_4 &: V \times S \rightarrow V
\end{aligned}
\tag{3.20}
$$

where V and S denote a vector operand and a scalar operand, respectively. The mappings f_1 and f_2 are unary operations and f_3 and f_4 are binary operations. As

Vector Instruction

- Given V- a Vector(operand)
  - An ordered set of n elements(n : the *length* of vector)
    - Elements are scalar : floating point number, integer, logical value, character(byte)
- Given S- a scalar (operand)

- Unary Operation- f1 and f2:
  - f1: $V \longrightarrow V$ example

f_1 VSQR Vector square root : $B(I) \leftarrow \sqrt{A(I)}$

Vector Instruction contd….

- Unary Operation- f1 and f2:
  - f2: V $\longrightarrow$ S example:

f_2 VSUM Vector summation: $S = \sum_{I=1}^{N} A(I)$

- Binary Operation- f3 and f4:
  - f3: V x V $\longrightarrow$ V example:

f_3 VADD Vector add: $C(I) = A(I) + B(I)$

- f4: V x S $\longrightarrow$ V example:

f_4 SADD Vector-scalar add: $B(I) = S + A(I)$

Table 3.5 Some representative vector instructions

Type	Mnemonic	Description ($I = 1$ through N)	
f_1	VSQR	Vector square root:	$B(I) \leftarrow \sqrt{A(I)}$
	VSIN	Vector sine:	$B(I) \leftarrow \sin(A(I))$
	VCOM	Vector complement:	$A(I) \leftarrow \overline{A(I)}$
f_2	VSUM	Vector summation:	$S = \sum_{I=1}^{N} A(I)$
	VMAX	Vector maximum:	$S = \max_{I=1,N} A(I)$
f_3	VADD	Vector add:	$C(I) = A(I) + B(I)$
	VMPY	Vector multiply:	$C(I) = A(I) * B(I)$
	VAND	Vector and:	$C(I) = A(I)$ and $B(I)$
	VLAR	Vector larger:	$C(I) = \max(A(I), B(I))$
	VTGE	Vector test $>$:	$C(I) = 0$ if $A(I) < B(I)$
			$C(I) = 1$ if $A(I) > B(I)$
f_4	SADD	Vector-scalar add:	$B(I) = S + A(I)$
	SDIV	Vector-scalar divide:	$B(I) = A(I)/S$

VECTOR LENGTH

- vector length of 64.
  - 1. In real world applications vector lengths are not exactly 64.
    - adding just first n elements of a vector ,Vector Length register(VLR) used for this
    - VLR controls the length of any vector operation by defining their length.
    - value cannot be greater than the length of the vector registers. (64 in this case)
  - 2. In real world applications, data in vectors in memory can be greater than the MVL of the processor.
    - we use a technique called Strip Mining

STRIP MINING

- Splitting data : each vector operation is done for a size less than or equal to MVL.
  - Done by a simple loop with MOD operator as control point.

STRIP MINING continued…

```
low = 0;
 VL = (n % MVL); /*find odd-size piece using modulo op
    % */
 for (j = 0; j <= (n/MVL); j=j+1)
 { /*outer loop*/
for (i = low; i < (low+VL); i=i+1) /*runs for length VL*/
    Y[i] = a * X[i] + Y[i] ; /*main operation* /
low = low + VL; /*start of next vector*/
 VL = MVL; /*reset the length to MVL*/ }
```