

The 32 vector registers feature 128 bits. The 139 new instructions include string operations, vector integer, and vector floating point operations. Each register contains multiple data elements of a fixed size. The following instructions code specifies which data format to use and the size of the elements:

- ▶ Byte (16 8-bit operands)
- ▶ Halfword (eight 16-bit operands)
- ▶ Word (four 32-bit operands)
- ▶ Doubleword (two 64-bit operands)
- ▶ Quadword (one 128-bit operand)

The collection of elements in a register is called a *vector*. A single instruction operates on all of the elements in the register. Instructions include a non-destructive operand encoding that allows the addition of the register vector A and register vector B and stores the result in the register vector A ($A = A + B$).

A schematic representation of a SIMD instruction with 16-byte size elements in each vector operand is shown in Figure 3-7.

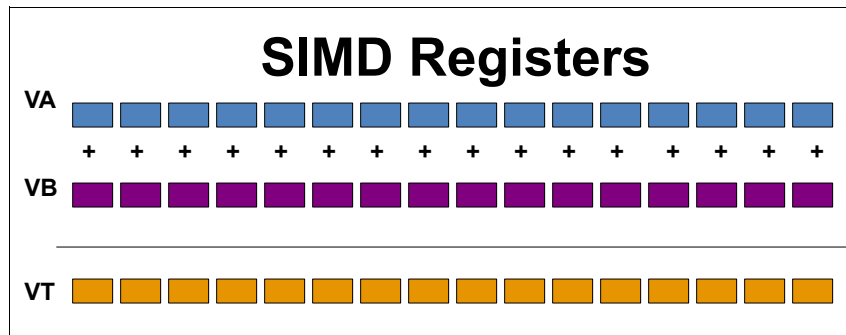


Figure 3-7 Schematic representation of add SIMD instruction with 16 elements in each vector

The vector register file overlays the floating-point registers (FPRs), as shown in Figure 3-8. The FPRs use the first 64 bits of the first 16 vector registers, which saves hardware area and power, and makes it easier to mix scalar and SIMD codes. Effectively, the core gets 64 FPRs, which can further improve FP code efficiency.

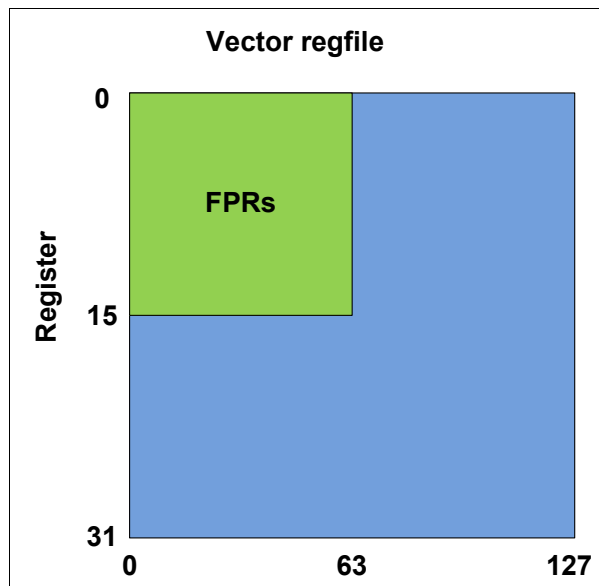


Figure 3-8 Floating point registers overlaid by vector registers