

### In More Depth: The Power PC's Multiply-Add Instruction

The matrix multiply on page 210 relied on a multiply operation and an add operation, which is typical of many matrix and vector operations. Hence, the PowerPC has a “fused” multiply-add instruction: a single instruction reads three operands, multiplies two operands and adds the third to the product, and writes the sum in the result operand. Hence, the two MIPS floating-point instructions in the matrix multiply example would be replaced by one in PowerPC. This instruction can increase peak floating-point performance.

Fused multiply-add also performs the two operations and *then* rounds, unlike separate multiply and add instructions, which would round after each operation. The instructions also calculate extra bits for intermediate results to improve accuracy. Besides being potentially faster, the extra accuracy of fused multiply-add can also be helpful for calculating divide and square root, and in software libraries that calculate at higher precision than 64 bits. In fact, PowerPC hardware uses fused multiply-add hardware to calculate divide, and accurate division was the motivation for skipping the round between the two operations.

**3.25** [10] <§§3.5, 3.4> Suppose you have a multiply-add instruction that performs the two operations back to back: `fmadd $f0, $f2, $f4, $f6` --->  $\$f0 = \$f2 * \$f4 + \$f6$ . Convert the following MIPS code into MIPS code with the fused multiply add instruction:

```
l.d    $f0, -8($gp)
l.d    $f2, -16($gp)
l.d    $f4, -24($gp)
mult.d $f0, $f0, $f2
add.d  $f0, $f0, $f4
s.d    $f0, -8($gp)
```

**3.26** [20] <§3.5> One of the advantages of the fused multiply-add instruction is its increased precision. This is because it does not round between the multiply and add step. When rounding, use a guard, round, and a sticky bit, and round using round to nearest. Consider the following IEEE 754 single precision numbers with  $x = 0100\ 0000\ 0110\ 0000\ 0000\ 0000\ 0010\ 0001_{\text{two}}$ ,  $y = 0100\ 0000\ 1010\ 0000\ 0000\ 0000\ 0000\ 0000_{\text{two}}$ , and  $z = 0011\ 1100\ 1110\ 0000\ 0000\ 1010\ 1100\ 0000_{\text{two}}$ .

Perform  $x * y + z$ :

- With round between the multiply and the add.
- Without rounding between the multiply and the add.