Floating Point Arithmetic

Jiaming Xu

2023-9-20

<1997 Extended-precision Format

- IEEE 754 (1985)
- IA32 (1985), x86-64 (1999), Itanium (2001), Motorola 6888x, FPA10 (ARM), x87

- 80-bit extended-precision format floating point register
- 1-bit sign, 15-bit exponent, 63-bit fraction

- Everything works well?
- NO! Something frustrating happened!

<1997 IEEE 754 Extended-precision Format

- Unconsistency between register and memory

```
1  volatile int rcnt = 0;  /* Used to create side effects */
2
3  double recip(int denom) {
4      rcnt++;   /* Side effect to prevent optimization */
5      return 1.0/(double) denom;
6  }
7
8  int dequal(double x, double y) {
9      return x==y;
10 }
11
12 void test1(int denom) {
13     double r1, r2;
14     int t1, t2;
15
16     r1 = recip(denom);    /* Stored in memory              */
17     r2 = recip(denom);    /* Stored in register            */
18     t1 = r1 == r2;        /* Compares register to memory   */
19     t2 = dequal(r1,r2);   /* Compares memory to memory     */
20     printf("test1 t1: r1 %f %c= r2 %f\n", r1, t1 ? '=' : '!', r2);
21     printf("test1 t2: r1 %f %c= r2 %f\n", r1, t2 ? '=' : '!', r2);
22 }
```

```
test1 t1: r1 0.100000 != r2 0.100000
test1 t2: r1 0.100000 == r2 0.100000
```

⩾ 1997 Streaming SIMD Extensions

- Pentium MMX (1997) SSE (single-precision FP arithmetic)
- Pentium 4 (2001) SSE2 (single/double-precision FP arithmetic)

- All processors capable of executing x86-64 code support SSE2 or higher
- 8x (with IA32) or 16x (with x86-64) XMM registers of 128 bits each

⩾ 1997 SSE

- Logic circuit diagrams

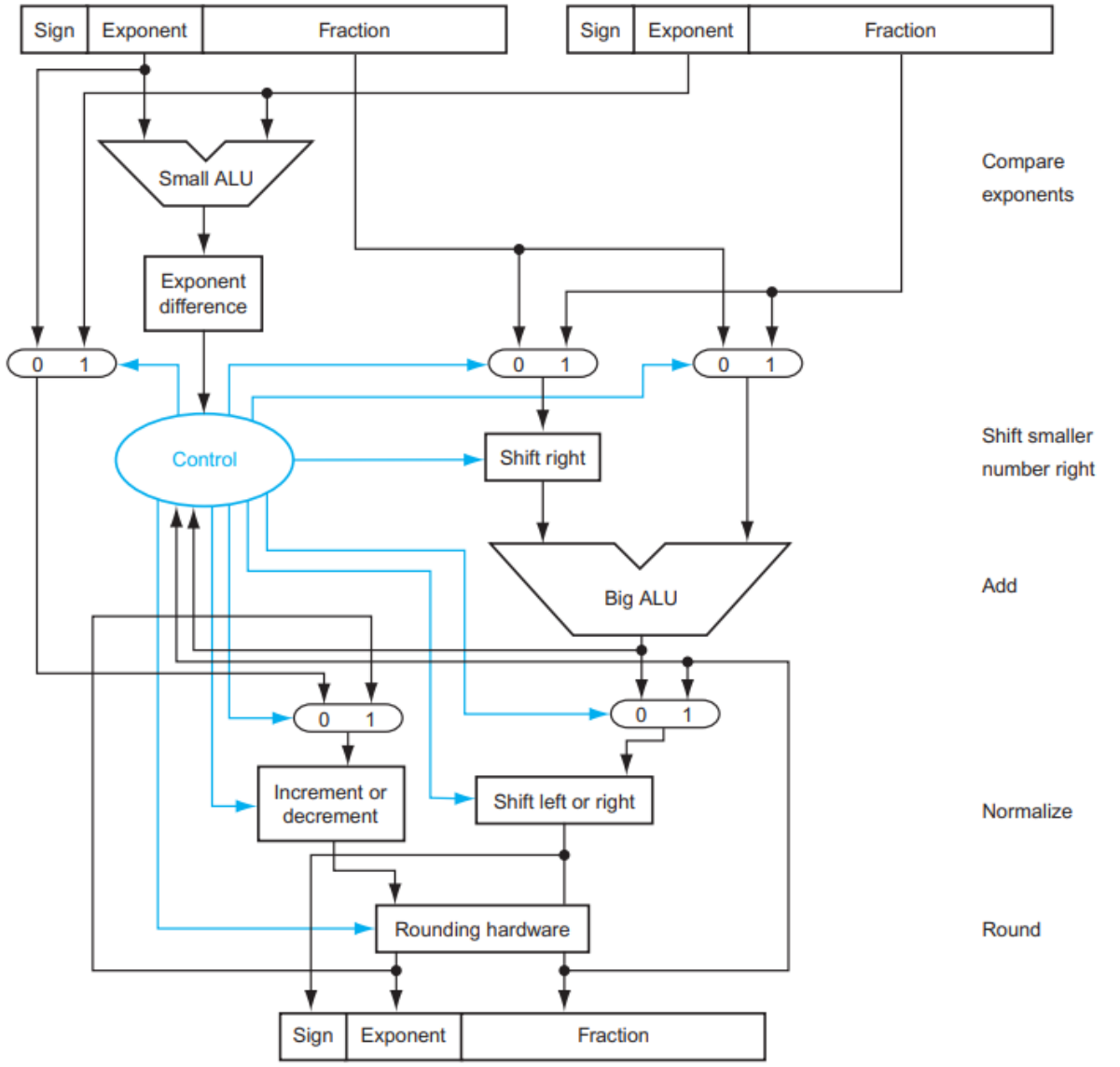

FIGURE 3.15 Block diagram of an arithmetic unit dedicated to floating-point addition. The steps of Figure 3.14 correspond to each block, from top to bottom. First, the exponent of one operand is subtracted from the other using the small ALU to determine which is larger and by how much. This difference controls the three multiplexors; from left to right, they select the larger exponent, the significand of the smaller number, and the significand of the larger number. The smaller significand is shifted right, and then the significands are added together using the big ALU. The normalization step then shifts the sum left or right and increments or decrements the exponent. Rounding then creates the final result, which may require normalizing again to produce the actual final result.

⩾1997 Streaming SIMD Extensions

- Integer-like action
  - Friendly for evaluating expressions containing a mixture of data types
- Better performance
  - move data back and forth between registers and memory less
- Probability of parallel operations on packed data
  - More on Chapter 5

Reference

- Extended precision - Wikipedia
- CS:APP Web Aside DATA:IA32-FP: Intel IA32 Floating-Point Arithmetic
- CS:APP2e Web Aside ASM:SSE: SSE-Based Support for Floating Point
- Computer Organization And Design: The Hardware/Software Interface (David a. Patterson, John L. Hennessy) Chapter 3