What is claimed is:

1. A method for performing failsafe computation, the method comprising the steps of:

performing a first calculation to generate a first result using a single math library or using a single math co-processor;

performing a second calculation using a scalar and the first calculation to generate a second result using the single math library or using the single math co-processor, the second calculation including multiplying the first calculation by the scalar to generate a scaled result, and dividing the scaled result by the scalar to generate the second result; and

indicating whether the first result and the second result are equivalent.

2. The method of claim 1, further comprising the step of:

performing a safe operation if the first result and the second result are not equivalent.

3. The method of claim 1, wherein the first calculation includes one or more operations, the one or more operations including a final operation, and wherein the second calculation includes multiplying the scalar before the final operation.

4. The method of claim 1, wherein the first result has a first binary representation and the scaled value has a second binary representation, and the scalar is selected to maximize the difference between the first binary representation and the second binary representation.

5. The method of claim 1, wherein the scalar is an integer, a floating point, or a double.

6. The method of claim 1, wherein the scalar is an irrational number.

7. The method of claim 1, wherein a difference between the scalar and 1.0 is minimized to preserve resolution of the scaled result.

8. The method of claim 1, further comprising the steps of:

adding an offset to the first result if the first result is determined to be zero.

9. A device for performing failsafe computation to operate a hardware (102), the device comprising:

an electronics (104) configured to:

perform a first calculation (212) to generate a first result (214) using a single math library or using a single math co-processor;

perform a second calculation (218) using a scalar (216) and the first calculation (212) to generate a second result (220) using a single math library or using a single math co-processor, the second calculation (218) including multiplying the first calculation (212) by the scalar (216) to generate a scaled result, and dividing the scaled result by the scalar (216) to generate the second result (220); and

indicate whether the first result (214) and the second result (220) are equivalent.

10. The device of claim 9, wherein the electronics (104) is further configured to perform a safe operation if the first result (214) and the second result (220) are not equivalent.

11. The device of claim 9, wherein the first calculation (212) includes one or more operations, the one or more operations including a final operation, and wherein the second calculation (218) includes multiplying the scalar (216) before the final operation.

12. The device of claim 9, wherein the first result (214) has a first binary representation and the scaled value has a second binary representation, and the scalar (216) is selected to maximize the difference between the first binary representation and the second binary representation.

13. The device of claim 9, wherein the scalar (216) is an integer, a floating point, or a double.

14. The device of claim 9, wherein the scalar (216) is an irrational number.

15. The device of claim 9, wherein a difference between the scalar (216) and 1.0 is minimized to preserve resolution of the scaled result.

16. The device of claim 9, wherein the electronics (104) is further configured to add an offset to the first result (214) if the first result (214) is determined to be zero.

17. A method for compiling a code to operate a hardware in a failsafe manner comprising the steps of:

identifying a first calculation configured to generate a first result using a single math library or using a single math co-processor;

generating a second calculation including the first calculation and a scalar using the single math library or using the single math co-processor, the second calculation generating a second result, the second calculation including multiplying the first calculation by the scalar to generate a scaled result, and dividing the scaled result by the scalar to generate the second result;

generating an indication of whether the first result and the second result are equivalent; and

including the first calculation, the second calculation, and the indication in an executable program.

18. The method of claim 17, wherein generating the second calculation including the first calculation and the scalar and generating the indication of whether the first result and the second result are equivalent includes defining a class.

19. The method of claim 17, further comprising the steps of:

storing the first result and the second result in distant sections of memory.

20. The method of claim 17, further comprising the steps of:

storing the first calculation and the second calculation in distant sections of memory.

21. The method of claim 17, further comprising the step of:

pretesting the single math library for computation errors.