**Overview**

This scholarship relates to a project that will deliver a radically new technique for performing the basic computer arithmetic operations, including add, subtract, multiply, divide and exponentiate. The current 'floating point' technique is slow and bulky, and liable to rounding errors that accumulate unpredictably during a long computation, the result of which therefore offers no guarantee of correctness. The new technique will dramatically improve the speed of these operations, while using less silicon and dissipating less power. It will also permit a user to mandate a specified degree of accuracy and will deliver bounded results within this limit. This combination of properties will render a successful implementation attractive to developers of custom silicon devices in a range of numerical application areas, to developers of supercomputing systems, and to IP houses. The proposal combines two strong and established lines of research, both of which independently have already delivered very significant results, but which complement each other in a way that is entirely novel and synergistic.

**Methodology**

One of these themes involves the replacement of the floating point arithmetic system with the logarithmic number system. We showed in [Col1] that this would deliver dramatic gains in the speed and accuracy of multiplication and division, although the performance of addition and subtraction would be largely unchanged. A microprocessor device incorporating these techniques was fabricated and extensively tested, and the results confirmed these predictions [Col2]. In a further development of the mathematical theory, we have now shown that the speed of addition and subtraction too can be substantially improved [Col3], and that a logarithmic arithmetic unit will occupy around the same area as a floating point one. Ongoing work will reduce the area still further. In simultaneous developments, an approach to the problem of accuracy has been proposed [Gus1]. This is based on the development of a 'universal number' system, an extension of floating point techniques that tracks the accuracy of each step of a computation and automatically adjusts the precision in order to maintain the accuracy of the result within user-specified limits. It has also been shown that, as this system will always set the accuracy to the minimum required to meet a given specification, power savings can frequently be achieved by reducing the wordlength and thus eliminating unnecessary memory and bus traffic.

It will be apparent that if these two approaches could be combined, they would precisely complement each other. A 'universal logarithmic' number system would offer extensive improvements in speed, accuracy and silicon area for all operations, at reduced levels of power dissipation, and would guarantee that its results lie correctly within known bounds. As an incidental benefit, programmers would be relieved of the need to work at exaggerated levels of precision in order to guard against error accumulation. Such a system would be a major advance on any arithmetic techniques in existence and would be of significant interest in supercomputing, DSP and numerically intensive areas such as graphics. We are therefore proposing a programme of research that will develop the theoretical aspects of this concept, simulate its performance, and then, time permitting, build demonstration hardware that could either be general.
purpose or based on a specific application of interest to the student.

**Timeline**

**Year 1**

- Familiarisation with the universal number (unum) system.
- Visit to A*Star.
- Development of a unum simulator, possibly in low-level format (e.g. C language) to facilitate eventual transcription into hardware design.
- Using the simulator, studies of execution of algorithms from a variety of application areas using standard floating point (FP) arithmetic and unum arithmetic.
- Quantification of gains over FP, and selection of appropriate demonstration applications.
- Estimate of silicon requirements and operational speed of unum processing hardware.
- Familiarisation with the logarithmic number system (LNS).
- Write-up of first-year report.

**Year 2**

- Development of an LNS simulator (largely from existing code).
- Further work on this, e.g. to allow the representation of integer values and automatic transition between integer and log.
- Review of silicon requirements and operational speed of LNS processing hardware.
- Using parameters determined above, comparative study of FP, unum and LNS processing hardware; hence proposal for universal logarithmic (ulog) processing system combining strengths of unums and LNS.
- Visit to A*Star to discuss proposal.

**Year 3**

- Development of ulog simulator combining elements from both existing simulators.
- Using the simulator, studies of execution of previously selected demonstration algorithms, now in ulog format.
- Quantification of gains over FP.
- Development of VLSI implementation of a ulog arithmetic unit, much of which will use existing VHDL code.
- Determination of speed and area parameters for the above.
- Enhancement of simulation results using above parameters, and thus comprehensive evaluation of gains over FP.
- Visit to A*Star to discuss results.
- Iteration and refinement of previous activities as necessary.
- (Time permitting) Integration of the arithmetic unit into a custom device in an area of interest to the student, and prototyping this on an FPGA.

**Year 4**

- Writing up.

**Training & Skills**

The student will gain familiarity with contemporary and emerging techniques in computer arithmetic, and of appropriate research methodology and technical writing. There will be extensive practical experience of VLSI design using the CADENCE suite, and possibly also of FPGA prototyping. A successful student will be exceptionally well placed to enter a career in VLSI design and development with a semiconductor company.

**References & Further Reading**

[Col3] J. N. Coleman and R. Che-Ismail, "LNS with Co-Transformation Competes with Floating-Point", accepted for publication in *IEEE Transactions on Computers*.

**Further Information**

Dr J.N. Coleman
School of Electrical & Electronic Engineering
Newcastle University
NE1 7RU
j.n.coleman@ncl.ac.uk