INTERVAL ARITHMETIC REQUIREMENTS FOR DIGITAL SIGNAL PROCESSORS

W. Edmonson[a], **W. Alexander**[b], **C. Gloster**[c] and **E. Hughes**[d]

[a]Department of Electrical Engineering
Hampton University
Hampton, Virginia 23668
wedmonson@ieee.edu

[b]Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

[c]Department of Electrical and Computer Engineering
Howard University
Washington, DC 20059

[d]Department of Electrical Engineering
Virginia Commonwealth University
Richmond, VA 23284

Interval methods are important to the signal processing community because the interval method results for signal processing algorithms are valid in that the results of the computations are sets of real numbers that include the infinite precision results. These results take into account both computational errors and data uncertainty. Interval methods also provide beneficial results for optimization algorithms. Optimization is an important procedure in many signal processing and control problems related to estimation and identification.

We present a set of system and architectural design requirements for the implementation of an interval arithmetic logic unit (IA-ALU). This IA-ALU can be used as the core of any digital signal processor. Digital signal processors operate on algorithms that can be characterized as being repetitive and numerically intensive. General requirements of a digital signal processor are that it has a fast multiply-accumulate instruction, multiple-access memory, specialized program control for interrupt handling and I/O, and fast and efficient access to peripherals. The IA-ALU needs to be able to compute the arithmetic operations of addition, subtraction, multiplication and the combined multiply/accumulator instruction in one clock cycle to be practical for a digital signal processor. The set operators are required for many of the interval method based optimization algorithms and therefore they will also be implemented. The initial design will be formulated as a fixed point IA-ALU because of its faster computational speed, simplicity and low power as compared to a comparable floating-point unit. We will provide a comparative analysis of the proposed interval arithmetic-ALU with general purpose ALU's.