

2001 Paper 8 Question 10

Comparative Architectures

An important application spends a large proportion of its running time executing a particular loop. The loop is responsible for summing two arrays containing unsigned eight-bit values packed in memory into a similar third array. Saturating arithmetic is used, whereby results that overflow are “clipped” to the maximum representable value. For example, for the unsigned eight-bit case, 250 plus 20 would result in 255. In this particular application, such overflows are rare in practice, and this fact may be exploited to optimise the implementation.

- (a) Write pseudo code for a simple implementation of the inner loop for a 32-bit processor with a RISC-like instruction set. [Hint: It is possible to use the CPU’s 32-bit ALU to perform four eight-bit additions with a single add instruction, and then use further code to detect if overflow occurred and correct it. You may assume the arrays start on word-aligned boundaries.] [10 marks]
- (b) Assuming the arrays are present in the CPU’s L1 data cache, estimate the number of cycles required to sum arrays of length N on a statically-scheduled two-way super-scalar processor. State any assumptions you make. [5 marks]
- (c) Many instruction set architectures have been augmented with SIMD (Single Instruction Multiple Data) instructions to enhance processors’ performance when dealing with packed arrays such as those used by this application. Demonstrate how SIMD instructions could be used to optimise the loop. Assuming that 50% of the running time of the application was spent executing your previous loop implementation, estimate the program’s speedup when using the SIMD optimised loop. [5 marks]