Bitslice Vector Computation

MCS Dissertation Abstract

Student: John Lennon - 10705273
Award: Master in Computer Science
Supervisor: Dr. David Gregg
May 2015

This work explores an alternative way of representing arrays of numbers in computers. A bitslice representation is considered. This approach takes advantage of bitwise-level parallelism. In the past, bitslicing approaches have served to produce more efficient cryptographic implementations, such as the Data Encryption Standard (DES) symmetric-key algorithm. In general however, bitslicing is rarely seen outside of the realm of cryptography. To help bridge the gap into more mainstream fields, this dissertation focuses on the development of a bitslice arithmetic library. Functions to add, subtract, and multiply bitslice values are built from the ground up, using only bitwise operations. Conversion algorithms are developed to allow conversion from standard arrays to bitslice arrays, and vice versa. Logical and arithmetic shifting routines are also developed. Support for signed bitslice numbers and fixed-point bitslice numbers are added. Interesting properties arise from developing the library, such as the inherent support for arbitrary-sized (or unusual-sized) bitslice values (where computing 9-bit values is no more complex than computing 8-bit values). This work examines the practical challenges of developing such a library. It examines the performance of bitslice computation compared to that of conventional approaches, by measuring and comparing relative clock cycles for each operation. It is found that in the case of 8-bit values, bitslice computation performs faster than conventional computation when the functionality is mapped from gate-level logic.