moviCompile: An LLVM based compiler for heterogeneous SIMD code generation

Erkan Diken*, Roel Jordans*, Martin J. O'Riordan†
*Eindhoven University of Technology
Eindhoven, The Netherlands
Email: e.diken@tue.nl, r.jordans@tue.nl
† Movidius Ltd., 1^{st} Floor, O'Connell Bridge House, D'Olier St, Dublin 2, Ireland
Email: Martin.ORiordan@movidius.com

Abstract

Numerous applications in communication and multimedia domains show significant data-level parallelism (DLP). The amount of DLP varies between applications in the same domain or even within a single application. Most architectures support a single vector-, SIMD-width which may not be optimal. This may cause performance and energy inefficiency. We propose the use of multiple (heterogeneous) vector-widths to better serve applications with varying DLP. The SHAVE (Streaming Hybrid Architecture Vector Engine) VLIW vector processor shown in Figure 1 is an example of such an architecture. SHAVE is a unique VLIW processor that provides hardware support for native 32-bit (short) and 128-bit (long) vector operations. Vector arithmetic unit (VAU) supports 128-bit vector arithmetic of $8/16/32-bit$ integer and $16/32-bit$ floating point types. Scalar arithmetic unit (SAU) supports 32-bit vector arithmetic of $8/16-bit$ integer and $16-bit$ floating point types.

The moviCompile compiler is an LLVM based commercial compiler targeting code generation for SHAVE processor family. The moviCompile compiler is capable of SIMD code generation for 128-bit (long) and 64-bit vector operations. This work focuses on compiler backend support for 32-bit (short) vector operations. More specifically, this work aims to generate SIMD code for short vector types (e.g. 4 x i8, 2 x i16, 2 x f16) that can be executed on 32-bit SAU next to the 128/64-bit SIMD code. As a result, moviCompile is able to generate heterogeneous assembly code consisting of both short and long vector SIMD operations. Currently, we are testing the compiler using TSVC (Test Suite for Vectorizing Compilers) and intend to measure the performance improvements.

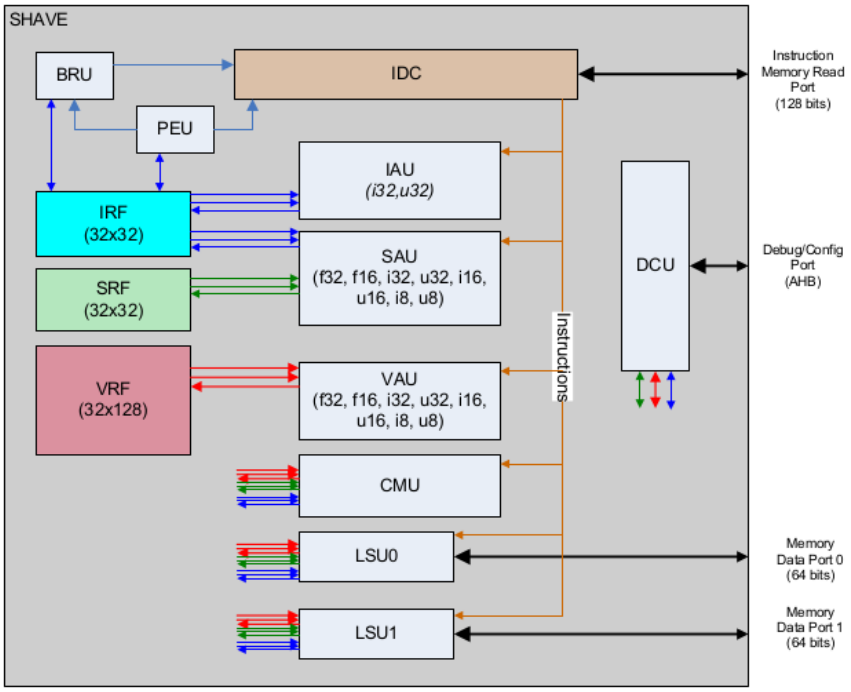

Fig. 1: SHAVE VLIW vector processor