LLVM compiler for in flight SW development and validation process

The LLVM LEON backend resulted from a joint project between LERO - the Irish Software Research Center and the ESTEC division of ESA. The project targeted the development of a compiler that adapts and extends the LLVM toolchain to the LEON family of processors. The LLVM LEON backend project run as a software engineering endeavor with the emphasis put on compiler correctness. All the traditional software process stages were performed in a rather spiral manner where multiple iterations led to the final product. During these iterations, various artifacts were produced, including documentation, backend implementation, and a test suite for integration testing. The multiple-iteration software process helped to refine the final results and come up with a comprehensive, well-structured and well-tested, yet well-integrated in the LLVM framework, backend for LEON processors. Most of the initially planned features were fully implemented and tested, including support to the planned variants of LEON 2 (AT697E and AT697F), LEON 3 (UT699 and GR712C), and LEON 4 (GR740) processors, software floating- point instructions, support to instruction latencies for LEON 2, LEON 3, and LEON 4, erratum fixes for LEON 2 and LEON 3, and optimizations. As part of the project, the LLVM LEON backend code was uploaded into the LLVM code repository.

The latest test results obtained from the Cobham Gaisler test suite clearly demonstrate that the LLVM LEON backend is a complete solution and good competitor of the GCC LEON compiler. The LLVM system can therefore be used to generate working production code with equivalent confidence to that code generated by GCC. Further, the use of itineraries, the LEON 2-specific errata fixes, and the clean separation of all the errata fixes, along with support for LEON-specific instructions, CASA, UMAC and SMAC, means that the LLVM LEON backend now surpasses the capabilities of GCC. Moreover, it provides a basis for further extension of the LLVM SPARC backend in a way that would be difficult to achieve on the relatively aged GCC codebase.