

A common design methodology for embedded DSP systems is the integration of one or more digital signal processors (DSPs), program memory, and ASIC circuitry onto a single IC. Program memory size being limited in embedded DSP systems, the criterion for optimality is that the embedded software must be very dense. In order to improve the instruction cycle time, it's becoming more common to design DSPs with multiple functional units. Recently, DSPs have been designed to include support for SIMD instructions. Such a design with support for instruction level parallelism (ILP) poses significant challenges to developing optimizing DSP compilers, because existing DSP compilers provide little for exploiting ILP. In this work we show how we can modify a retargetable VLIW compiler infrastructure to efficiently develop an optimizing compiler for DSP with ILP. Our thesis is that using a VLIW compiler framework is a better approach to develop an optimizing compiler for a DSP with ILP/SIMD than a DSP-specific compiler that is enhanced to exploit ILP/SIMD. We use the IMPACT VLIW framework to develop a compiler for the second generation Fujitsu Hiperion, a fixed-point DSP.