PRMDL: a Machine Description Language for Clustered VLIW Architectures

A.S. Terechko, E.J.D. Pol, J.T.J. van Eijndhoven Philips Research Eindhoven {andrei.terechko, evert-jan.pol, jos.van.eijndhoven}@philips.com

Abstract

PRMDL is a format of the central machine description file that contains parameters of the whole retargetable compiler-simulator framework. The format features separate software and hardware views on the processor and defines a wide scope of the framework retargetability, enabling platform-based processor design and vast design space exploration for clustered VLIW architectures.

1. Physical and virtual machines in PRMDL

Modern embedded processors deploy rich and powerful sets of custom operations. However, these custom operations hamper retargetability of the compilers and C compatibility of the applications for the embedded processors. To overcome this obstacle PRMDL features explicitly separate software and hardware views on the processor (see Figure 1).



Figure 1. Separate software and hardware views

The *physical machine*, constituting the hardware view, accommodates all parameters of the processor hardware architecture, such as register file and hardware operation parameters. The *virtual machine*, constituting the software view, contains the C programming model of the processor. The application programmer can call the custom operations using C functions. The C functions corresponding to the processor custom operations are described in the virtual machine and referred to as software operations. The retargetable compiler recognises these functions in C code and maps them on the custom hardware operations from the

physical machine. The operation *mappings*, denoted by arrows in Figure 1, are also described in PRMDL. Hardware changes solely influence the physical machine and the mappings, leaving the compiler, the virtual machine and, consequently, the applications invariable. Figure 1 shows stability of the virtual machine across multiple hardware generations, insuring the source-level compatibility.

2. PRMDL capabilities

The *physical machine* describes the processor state, the distribution of functional units among VLIW issue slots, the processor datapaths and the hardware operations. The description of the processor state can include diverse memory structures (e.g. stacks, queues, random access register files, etc.). An explicit specification of the processor datapaths allows to describe clustered VLIW architectures with incomplete resource interconnect (e.g. partial bypass network). The physical machine also contains hardware operation signatures, combining operation names, guards, arguments, results, properties, and side-effects.

The *virtual machine* accommodates software operation signatures, containing operation names, argument and result types, and operation properties.

The PRMDL mapping sections define operation transformations, capable of driving parameterised code selection in all compilation stages. PRMDL supports two mapping types: ordinary and conditional. An ordinary mapping defines a transformation of a source operation into one set of target operations. In the conditional mapping an operation is mapped onto different groups of operations depending on a condition imposed on the operation immediate argument. The possible condition types are as follows: fitting a given range, fitting the range of an argument of a physical operation, and matching a given pattern. In order to ensure a strong processor family compatibility PRMDL supports mappings across architectures with different datapath sizes (e.g. from a 128bit CPU onto a 64-bit one). PRMDL enables this by allowing to address fractions of the arguments and results in the mapping definitions.