Abstract

In order to meet the increasing challenges concerning the performance and power demands of embedded applications, a processor is now embedded with the Application-specific functional units. Customized Functional Units both as hardware and the corresponding instructions are embedded to the base processor in order to improve the computational efficiency for a target application. During this process of generating the complex instructions and also for the code generation on this extended processor, one of the critical challenges for the compiler is to automatically perform fast and efficient instruction matching and selection. In this project, we developed a novel and efficient algorithm for matching the multiple-output complex Functional Units (FU’s). We will also illustrate that the assumption, which is the basis of the most of the current covering methodologies, may not always hold true. Current covering algorithms, generally aim to find the optimal cover within each basic block that minimizes the number of selected matches. Fewer matches translate to fewer operations for the schedule, and it is expected that the increased scheduling freedom leads to better (shorter) schedule. We provide some examples showing that this assumption need not necessarily achieve the goal of minimizing the execution time.
Multiple Output Complex Instruction Matching Algorithm for Extensible Processors

References

- N. Pothineni et. al., Exhaustive Enumeration of Legal Custom Instructions for Extensible Processors, 21st Int&apos;s Conf. on VLSI design 2008.

Index Terms

Computer Science

Embedded

Keywords

Matching algorithms Instruction Set Architecture