Development of the Retargetable Tool Suite for Embedded Software

Kwangman KO

School of Computer and Information Engineering, SangJi University, KOREA
kkman@sangji.ac.kr

Abstract

In order to construct the most suitable embedded software development environment, we design and implements the Retargetable Tool Suite for Embedded Software (RTS-ES) composed of target code generator, low power/energy optimizer, system simulator and debugger through the proposed Embedded Processor Description Language (EPDL).

1. Introduction

Recently, together with a new advent of embedded processor developed to support specific application area, and it evolution, a new study of software development to support the embedded processor and its commercial use has been revitalized. Especially for software development environment for embedded system which is sensitive to Time-to-Market issue, the research and development of efficient code generation technique to generate good quality of target code, efficient management of limited resources and power, and simulator/debugger to verify execution and performance evaluation of the processor in terms of software side prior to the actual development have been recognized as the core issues.

In this paper, we proposed Embedded Processor Description Language (EPDL) base on the novel ALDs. And then we generate the toolkit generator, such as target specific compiler, low power/energy optimizer, and simulator through the EPDL. Finally, we proved the excellence in the verification of execution, and the various performance evaluations through the benchmarking.

In section 2, we describe related work on ADLs and compare them. Section 3 presents a brief overview of RTS-ES using EPDL, while Section 4 concludes this paper.

2. Related works

Architecture Description Languages (ADL) have been classified into two categories depending on whether they primarily capture the Instruction-Set (IS) or the structure of the processor

EXPRESSION, a language architecture design space exploration for embedded System-on-Chip (SoC) and automatic generation for embedded of a retargetable compiler/simulator toolkit. Key features of EXPRESSION include: a mixed behavioral/structural representation supporting a natural specification of the architecture; explicit specification of the memory subsystem allowing novel memory organizations and hierarchies; clean syntax and ease of modification supporting
architectural exploration; a single specification supporting consistency and completeness checking of the architecture; and efficient specification of architectural resource constraints allowing extraction of detailed reservation tables for compiler scheduling[1].

nML[2] and ISDL[3] are examples of IS ALDs. In nML, the processor’s IS is described as an attributed grammar with the derivations reflecting the set of legal instructions. nML has been used by the retargetable code generation environment CHESS[4] to describe DSP and ASIP processors. ISDL also describes the processor in terms of its IS, with the goal of deriving a code generator, assembler and simulator.

MIMOLA[5] is an example of an ADL which primarily captures the structure of the processor wherein the net-list of the target processor is described in a HDL like language. One advantages of this approach is that the same description is used for both processor synthesis and code generation. MIMOLA descriptions are generally very low-level, and laborious to write. It is not clear how they generate a cycle accurate simulator using the MIMOLA description.

More recently, languages which capture both the structure and the behavior of the processor, as well as detailed pipeline information (typically specified using Reservation Tables) have been proposed. LISA [6] is one such ADL whose main characteristic is the operation-level description of the pipeline. LISA seems to have been designed primarily for retargeting simulators. Also, it does not support specification of detailed constraints information needed for compiler instruction scheduling.

Dynamic voltage scaling (DVS) is recognized as one of the most effective power reduction techniques. It exploits the fact that a major portion of power of CMOS circuitry scales quadratically with the supply voltage[7]. As a result, lowering the supply voltage can significantly reduce power dissipation. For non-interactive applications such as movie playing, decompression, and encryption, fast processors reduce device idle times, which in turn reduce the opportunities for power savings through hibernation strategies. In contrast, DVS techniques are still beneficial in such cases, i.e., DVS reduces power even when these devices are active. However, DVS comes at the cost of performance degradation. An effective DVS algorithm is one that intelligently determines when to adjust the current frequency-voltage setting(scaling points) and to which frequency-voltage setting(scaling factors), so that considerable savings in energy can be achieved while the required performance is still delivered.

3. Overview of the RTS-ES

Not only to ensure the core technique to cope with software development environment research for the embedded system but also to improve the current reality for inefficiency and staggering development time due to absolute dependency on base of GNU gcc, this latest unique technique enables to construct the most suitable software development environment for the embedded system. For which, we proposes and implements the Retargetable Tool Suite for Embedded Software(RTS-ES) composed of target code generator, low power/energy optimizer, system simulator and debugger through the proposed Embedded Processor Description Language(EPDL) as following Figure 1.
First, we designed and implemented originative Embedded Processor Description Language (EPDL) based on novel ADLs. EPDL compiler reads a processor specific EPDL (such as AMR.epdl, MIPS.epdl, SPARC.epdl) file and then produces Code Generator Description, Low Power/Energy Optimizer Generator Description, and Simulator & Debugger Generator Description. Each Description file stores information for the processor-specific code generator, low power/energy optimizer, and simulator & debugger generator. Second, we developed a target code generator producing good quality target code and low power/energy optimizer for the target code through the descriptions. Lastly, we developed a simulator and a debugger for verification of processor’s execution and performance evaluation. For the input of the whole system, a new intermediate language (*.ic) improved with intermediate representation of DAG [8] is used.

EPDL employs a simple C-like syntax to ease specification and enhance readability and description is composed of 4 sections (operations section, instruction selection section, processor specification section, memory section). Operation section describes the Instruction Set (IS) of the processor and organized into various operation groups. Each group contains a set of operations having some common characteristics as following Figure 2.

```
(OPCODE ADD
   (OP_TYPE DATA_OP)
   (OPERANDS (_Rn_int_any) (_Rm_int_all) (_Rd_int_any))
)```
Instruction Selection section described information needed by intermediate code to target instruction ion mapping as following rules.

```
(Instruction Selections
  IC_instruct (iadd Src1 Src2 Dest)
  TARGET (add Src1 Src2 Dest)
)
```

In this instruction selection, we uses a tree pattern matching algorithm [9][10] and design algorithm for the low power/energy managements. Detaild low power/energy strategy and performance results can be published/presented final paper. Processor specification section contains architecture components and memory section describes the types and attributes of various storage components as like the EXPRESSION.

4. Conclusion

In this paper, we present the RTS-ES model and EPDL, a new ADL used for retargeting a low power/energy optimized compiler/simulator toolkit. To support fast retargeting of the low power/energy optimized compiler and simulator is necessary. So we use the EPDL and RTS-ES model to retarget the low power/energy optimized compiler through a set of toolkit generation techniques. In many respects the EXPRESSION is similar to our approach. However, for ADL descriptions, EXPRESSION approach provides a very huge rules and limited MIPS R4000 processor.
So far, we designed an EPDL and implement an EPDL compiler. Practically, we describes ARM processor description (ARM.epdl) and then generates a compiler with a low power/energy optimizations. In our on-going research and work, we targets more detailed low power/energy optimizations by inserting various algorithms and providing a refine simulator for architecture explorations.

References


Authors

I am an Associate Professor at the School of IT Engineering, Sang-Ji university. I joined the faculty in Sep. 2001 having previously worked as a full-time lecturer at Kwang-Ju Women's University (March 1998 to August 2001). I graduated a BS (WonKwang University, 1991), and then MS and Ph.D (Computer Engineering from DongGuk University in 1993 and 1998), respectively. My research focuses on compilation techniques and efficient programming language environments. Currently, my research focuses on the retargetable tool suite (low power/energy optimized compiler, simulator) for the embedded systems, virtual machine, and Architecture Description Language.