Reconfigurable Embedded Processing: A Compiler Approach

Wong Weng-Fai

Dept. of Computer Science
National University of Singapore
http://www.comp.nus.edu.sg/~wongwf
## The Niche

<table>
<thead>
<tr>
<th>Speed</th>
<th>Flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>Performance/Cost</td>
</tr>
<tr>
<td>ASIC</td>
<td>Very High</td>
</tr>
<tr>
<td>ASIP/DSP</td>
<td>High</td>
</tr>
<tr>
<td>Custom</td>
<td>High</td>
</tr>
<tr>
<td>Generic</td>
<td>Low-Medium</td>
</tr>
</tbody>
</table>

[Faraboschi’00]
Reconfigurable SOC processors that can be rapidly customizable to be application-specific will enable quick time to market and rapid prototyping at low non-recurrent engineering cost.
The Research Challenge

- Specification for Reconfiguration
- Compiling for Reconfiguration
- Reconfigurable Processors
  – Architectural Space Exploration
Assumptions

- We can analyze an application to death
- Reconfigurable processors on the horizon
  - Tensilica, Triscend
- VLIW / EPIC Core Processor
  - Adaptive EPIC (AEPIC)
Our Approach

Compiler-centric

– High level analysis of application
– Retargeting different processor configuration
– Walk the architectural space for the best cost-performance solution
– Automatic hardware-software partitioning with interaction with architectural space exploration
Turning the Table

Application A

Application B

Application C

Compiler

Processor

Program Code
Turning the Table

Application

Compiler

Processor Config A

Processor Config B

Processor Config C

Program Code

H/W Config Specs
Our Research Agenda

- Application-specific processor customization
- Dynamic reconfiguration
- Application-specific memory hierarchy
- Application-specific runtime
The Baseline AEPIC

Profile and analyze - Tools

Hardwired component

Adaptive component

AEPIC executable
Research Statement:
Given an application, what is the best processor configuration
- number of functional units
- amount and type of registers
- datapaths
- instruction set
that achieves the optimal balance of
- performance
- power
- cost
Dynamic Reconfiguration

Research Problems:

1. How can dynamic reconfigurability be expressed to a compiler?

2. What can of analysis needs to be done to compile for a dynamically reconfigurable processor?

3. What kind of microarchitecture is needed to support dynamic reconfiguration?
Application-specific Memory Hierarchy

Research Statement:
Given an application, what is the best memory configuration
- level of caching
- cache parameters
- memory bussing
- prefetch mechanism
that achieves the optimal balance of
- performance
- power
- cost
Research Statement:
Given an application, what is the best runtime system

- resource allocation
- scheduling / multithreading
- I/O support

that achieves the optimal balance of
- performance
- power
- cost
Trimaran

C program

K&R/ANSI-C Parsing
Renaming & Flattening
Control-Flow Profiling
C Source File Splitting
Function Inlining

Classical Optimizations
Code Layout
Superblock Formation
Hyperblock Formation
ILP Transformations

Elcor/CAR

Dependence Graph Construction
Acyclic Scheduling
Modulo Scheduling

Region-based Register Allocation
Post-pass Scheduling

ReaCT-ILP

Machine Description

Simulator

Execution Statistics
State of Research

- Implementation of a simulator for AEPIC in the Trimaran framework
  – Polytechnic University of Bucharest

- Conversion of Trimaran/HP code to Handel-C

- Architectural description language for AEPIC
  – Wayne Luk (Oxford University)
March 28, 2002
ICSOC Presentation - Taiwan

\textbf{C \rightarrow Rebel \rightarrow Handel-C}

\textbf{Original}

- 2534 FF
- 620 cycles

\textbf{Our version}

- 3740 FF
- 651 cycles
The Collaborators

- Tulika Mitra, (Co-PI) NUS
- Center for Research on Embedded Systems and Technology, Georgia Institute of Technology
- Wayne Luk, Oxford University
- Asian Institute of Technology, Thailand
- Anna University, India
- St. Petersburg State University (initial contact)
Pilot project funding ≈ US$20K
– Agency for Science, Technology and Research, Singapore

Embedded and Hybrid Systems Programme
– A*STAR programme, call expected July 2002
Plug into a global consortium of experts on SOC
  – Especially at the physical design layer

Interchange of know-how and applications
  – Joint research
operands var="li1;"
if (var.is_reg())
  Reg_file file;
int vr_num = v;
LSReg vreg(file);
vr_liveouts +=
  if (glob_pref_
    blk_pref_fo;
  )
  if (var.alloca
    int ac_num;
    if (ac_num < 4)