HW/SW Co-Design Lab

Seminar 1

WS 2016/2017
Schedule of the Lab

Introduction:
Become acquainted with software and tool flows

Lab task:

Internet: http://mns.ifn.et.tu-dresden.de/Teaching/Courses/Pages/P-HWSW-Codesign.aspx

Work:
Autonomous work in groups of 2-3 people
during WS in our PC pools and preferably on your own PC due to limited seats in PC pool

Delivery:
During WS: until 31\textsuperscript{th} March 2017
1. Exported project workspaces including all sources (C, TIE)
2. Project report: Word-Template on Website

Hand in via e-mail: sebastian.haas@tu-dresden.de

Evaluation: Grade for 0/1/2 course, ET/IST: Lecture+Lab 7 credits, NES: 4 credits

Do not forget to register for the lab via HISQIS before examination period starts!
Tasks

1. FIR filter
   - Develop instruction set extensions for Tensilica processor (TIE)
   - Use Fusion, SIMD, FLIX
   - Compare direct form, transposed form
   - This task need **not** to take part in the final report.

2. FFT/IFFT
   - Develop instruction set extensions for Tensilica processor (TIE)
   - Use Fusion, SIMD, FLIX, knowledge from task 1
   - Compare DIT, DIF
   - Main task for the report and valuation.
Seminars

- **Today, 11th Oct., 4.DS, BAR/213/H:**
  - Introduction to Lab
  - First information on Tensilica Instruction Set Extension
  - Demo on Xtensa Xplorer

- **Tuesday, 25th Nov., 4.DS, BAR/213/H:**
  - Accounts ready to pick-up
  - Discussion/Solution to first task
  - Introduction to second task

- **Further questions at any time during semester:**
  - Make appointment via e-mail
  - If desired, discussion/consultation some weeks before the end of lab
Submission

Hand in via E-mail until 31\textsuperscript{th} of March 2017

Grading:
\begin{itemize}
  \item Archive with Tensilica workspace including optimized source code
  \item Project report in English (max. 3 pages w/o source code excerpts)
\end{itemize}

Assessment criteria:
\begin{itemize}
  \item 50\% practical work, 50\% report
\end{itemize}

(Anonymous) publication of the results via Internet:
\begin{itemize}
  \item Therefor necessary: your compliance with your signature
  \item \textit{Signature is voluntary!}
  \item \textit{There will be no disadvantage for you, if you disagree with the publication!}
  \item If you wish, your name can also be included in the publication
\end{itemize}
OVERVIEW
Benefits from this Lab

- Practical application of HW/SW Co-Design
- Get familiar with the Tensilica tool flow which is used by many companies today to create customized DSPs
- Grade for 0/1/2 course
Lab Approach

- RISC
- Enhanced RISC (ASIP)

Software

Compiler

Hot Spot

Accelerate (by creating new instructions)

Your Task!

TU Dresden
HW/SW Co-Design Lab
Slide 8
HW architecture: Xtensa LX5

- Debug Module
  - Trace Port
  - Access Port
  - On-Chip Debug
  - Performance Monitors
- Exception Support
  - Exception Handling Registers
    - Data Address Watch Registers
    - Instruction Address Watch Registers
  - Timers
  - Interrupt Control
- Designer-Defined Execution Units, Register Files, and Interfaces
  - Designer-Defined FLIX parallel execution pipelines - "N" wide
  - Base ISA Execution Pipeline
    - Loop Buffer
- Instruction Fetch / Decode
  - Instruction RAM
  - Instruction ROM
  - Inst. Memory Management & Error Protection
- Instruction Cache
- Base Register File
  - Base ALU
    - MAC 16 DSP
    - MUL 16/32
  - Integer Divider
  - Floating Point
- Designer-Defined Execution Units
  - ConnX Vectra LX / VMB DSP
  - ConnX BBE16/32/64 DSP
  - ConnX D2 DSP
  - HiFI Mini / HiFI 2 / HiFI EP / HiFI 3 Audio DSP
- Designer-Defined Data Load/Store Unit
  - Data Load/Store Unit
- Data Cache
  - Data ROMs
  - Data RAMs
- External Interface
  - Xtensa LX5 Processor Interface Control
    - Prefetch
    - Write Buffer
- PIF
- Designer-Defined Queues, Ports, and Lookups
  - Trace JTAG Debug APB
Tensilica Tool Flow

Not part of the lab

Upload Files

Download Files

Custom Processor RTL

RTL source code for complete processor hardware, EDA scripts

Compiler Tool Chain

Xtensa C/C++ Compiler
GNU Assembler, Debugger, Profiler

Simulation Models

Cycle-accurate ISS and XTMP system-modeling environment

TIE Hardware & EDA Scripts

RTL source code for TIE hardware, EDA scripts

Local Workstation/PC

 Iterate in minutes

Tensilica Processor Generator (Web-hosted)

Upload Files

Xtensa Xplorer

TIE Source File

TIE Compiler

Configuration Options

Download Files

TU Dresden
HW/SW Co-Design Lab
Slide 10
RISC Pipeline

IF
ID
EX
M
WB

IMEM
If Logic

Logic

128-bit interface for SIMD

64 bit VLIW

4x 32-bit integers

Extended Functional Accelerator Units
Functional Unit: Example

Example: Bitmap Logical Operation Instruction
Creating New Instructions

**Pure C code:**

```c
short a, b;
int z1, z2;
z1 = a*b;
z2 = z1 >> 16;
```

**TIE code:**

```c
operation MUL_SRL_16 {out AR z, in AR a, in AR b} {} {
    wire [31:0] m = TIEmul(a[15:0], b[15:0], 1);
    assign z = {16'b0, m[31:16]};
}
```

**C code with new instructions:**

```c
#include <xtensa/tie/tie.h>
short a, b;
int z;

z = MUL_SRL_16(a, b);
```
Creating Pipelined Instructions

operation MUL_SRL_16 {out AR z, in AR a, in AR b} {}
{
    wire [31:0] m = TIEmul(a[15:0], b[15:0], 1);
    assign z = {16'b0, m[31:16]};
}

schedule ms {MUL_SRL_16}
{
    use a 1;
    use b 1;
    def z 2;
}
operation MUL_SRL_16 {out AR z, in AR a, in AR b} {
    wire [31:0] m1 = TIEmul(a[31:16], b[31:16], 1);
    wire [31:0] m0 = TIEmul(a[15:0], b[15:0], 1);
    assign z = {m1[31:16], m0[31:16]};
}
LOGIN PROCEDURE
Log in to remote computer with web browser

https://entmrda.ifi.et.tu-dresden.de/RDWeb
You will be asked for your credentials. Enter ifn\username and your password from the account information. After you logged in the first time, you will be requested to change your password.
Once logged in, click on Xtensa Xplorer 6.0.2
Xtensa Xplorer will open with a connection via Remote Desktop Protocol (RDP). Your operating system has to provide an RDP client.

**Hint:**

- Windows automatically provides the RDP client
- For Linux or Mac users, install an RDP client suitable for your operating system

If you cannot log in anyway, work at PC pool in FAL 250.
Xtensa Xplorer: C/C++ Perspective

Build, run, profile and debug the complete C code

Select compile and optimization options:
Right click → “Build Properties”

Attach TIE file to coreLX5_hwsword:
Right click → “Attach TIE…”

Compile only TIE code:
Right click into TIE code window → “Compile TIE”

TIE area report
Xtensa Xplorer: TIE area report

Total area estimation

Operation area

States

Tables

Functions
Xtensa Xplorer: Benchmark Perspective

- Total number of cycles per code line
- Total number of cycles per instruction
- Total number of cycles per function
- Processor pipeline to identify hazards
Xtensa Xplorer: Debug Perspective

- Watch C variables and TIE wires
- Debug control
- Set breakpoints
- Watch memory content
Delivery: Export Xtensa Workspace (1)

Export Xtensa workspace: Right click → “Export”

Export as “Xtensa Xplorer Workspace”
Browse and save the *.xws file to your local hard drive. Usually “C on your_PC_name”
Select your desired project

Select C Projects to Export
To show a description of a project while importing it, create a README.txt file in the project.

- HelloWorld

- Add required libraries
- Export contents of Linked Folders
- Do not export the bin folder while exporting a Xtensa Managed Project

Select All  Deselect All

< Back  Next >  Finish  Cancel
Do **not** export any
- Launches
- Configurations
- Tools
- Builds
- User-defined formats

but **do** export your TIE file(s)
Summary ➔ Click on “Finish”

![Export Xtensa workspace summary]

- C projects selected:
  1) Hello World
     - linked folder content not included (preserved as links)
     - project bin folder contents are excluded

- TIE and TDB files selected:
  1) fin_tie.tie
TASK 1: FIR FILTER
Task 1: FIR Filter

- **FIR filter:** \[ y_n = \sum_{i=0}^{N-1} b_i x_{n-i} \]

- **Hot-Spots**
  - Multiply/Accumulate Operation (MAC)
  - Load input values/coefficients
  - Store output values

- **Different approaches:**
  1. Performing MAC on one input element with one coefficient
  2. Performing MAC on multiple elements in parallel \( \Rightarrow \) SIMD
  3. Separated instructions for Load, MAC and Store

```c
void fir_C(short *in, int *out, int len) {
    int n, i;
    for (n = 0; n < len+7; n++) {
        for (i = 0; i < 8; i++)
            out[n] += in[n+7-i]*coeff[i];
    }
}
```
SUMMARY
Summary

Your next TODOs:

- Register on list coming around
- Confirm your account
  - Pick up your account information at next introduction seminar or come to FAL 236
- Join next introduction session on 25th October
Contact & Rooms

Contact for Questions:
Sebastian Haas
sebastian.haas@tu-dresden.de
Room: FAL 236
Emil Matus
matus@ifn.et.tu-dresden.de
Room FAL 231

Website:
http://www.vodafone-chair.com
→ Teaching → Courses → „Praktikum HW/SW Co-Design”
http://mns.ifn.et.tu-dresden.de/Teaching/Courses/Pages/P-HWSW-Codesign.aspx

Questions?

Rooms
Admins: FAL 241
PC-Pool: FAL 250