ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

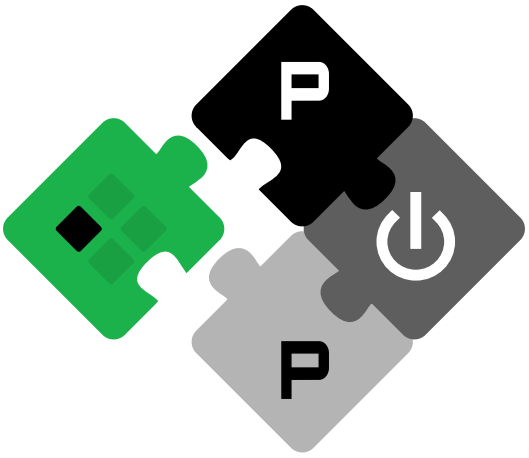
Institut für Integrierte Systeme
Integrated Systems Laboratory

Department of Information Technology and
Electrical Engineering

Autumn Semester 2015

PULPino implementation in 65nm CMOS

Semester Project



Florian Zaruba
zarubaf@student.ethz.ch

December 2015

Supervisors: Frank K. Gürkaynak (DZ), kgf@ee.ethz.ch
Michael Gautschi (IIS), gautschi@iis.ee.ethz.ch
Professor: Prof. Dr. Luca Benini, lbenini@iis.ee.ethz.ch

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abstract

PULPino is an open-source microcontroller like system, based on a small 32-bit RISC-V core that was developed at ETH Zurich. The core has an IPC close to 1, full support for the base integer instruction set (RV32I), compressed instructions (RV32C) and partial support for the multiplication instruction set extension (RV32M). It implements our non-standard extensions for hardware loops, post-incrementing load and store instructions, ALU and MAC operations. To allow embedded operating systems such as FreeRTOS to run, a subset of the privileged specification is supported. When the core is idle, the platform can be put into a low power mode, where only a simple event unit is active and wakes up the core in case an event/interrupt arrives.

The PULPino platform is available for RTL simulation, FPGA and as an ASIC in UMC 65nm (Imperio). It has full debug support on all targets. In addition we support extended profiling with source code annotated execution times through KCacheGrind in RTL simulations.

PULPino is based on IP blocks from the PULP project, the Parallel Ultra-Low-Power Processor that is developed as a collaboration between multiple universities in Europe, including the Swiss Federal Institute of Technology Zurich (ETHZ), University of Bologna, Politecnico di Milano, Swiss Federal Institute of Technology Lausanne (EPFL) and the Laboratory for Electronics and Information Technology of Atomic Energy and Alternative Energies Commission (CEA-LETI).

Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor. For a detailed version of the declaration of originality, please refer to Appendix **??**

Florian Zaruba,
Zurich, December 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Give an overview of the problem, and put your work into a bigger context. Motivate the questions addressed in this work and summarize your contributions. Related work should also be mentioned here, especially if you do not have a separate chapter for it.

1.1 First Section

1.2 Second Section

Chapter 2

Preliminaries / Background

This chapter can be skipped if the theory/algorithms are clear enough such that they can be explained without very much background information (e.g., within another chapter).

2.1 First Section

2.1.1 First Subsection

2.1.2 Second Subsection

First Subsubsection

Second Subsubsection

2.2 Second Section

2.2.1 First Subsection

2.2.2 Second Subsection

Chapter 3

Related Work

Depending on how much related work there exists, this chapter can also be merged into the introduction.

3.1 First Section

3.2 Second Section

Chapter 4

Theory / Algorithms

Describe the algorithms you evaluated. The *algorithmic* flow of your work should be clear after this chapter. Do not talk much about the resulting hardware architecture as this is a different topic (next chapter)! If you performed any number precision evaluations put them in this chapter as well.

4.1 First Section

4.2 Second Section

Chapter 5

Hardware Architecture

Describe the architecture and the architectural decisions you took. Blockdiagrams, the description of control, data flow and interfaces go in here. Note that the architecture you present here usually is more general than what you actually implemented and can even be in a parameterized form.

5.1 First Section

5.2 Second Section

Chapter 6

Design Implementation and Results

This chapter is about the architecture variant you actually implemented and its resulting performance; e.g., SNR, image quality, peak throughput, required bandwidth ... (whatever quality and performance metrics apply). In an ASIC or FPGA project you would also specify the key figures of your design; e.g., area/lut usage, timing figures, interface widths... In an ASIC project you would also talk about backend specific things such as the floorplan of your chip, design for test (and test coverage), power simulation, special clocking circuitry and pad/bonding diagrams.

6.1 First Section

6.2 Second Section

6.3 Verification

6.3.1 Functional

Do not forget to include information about how you managed to do the functional verification (golden model, testbench, etc.). Figure **??** illustrates an example setup.

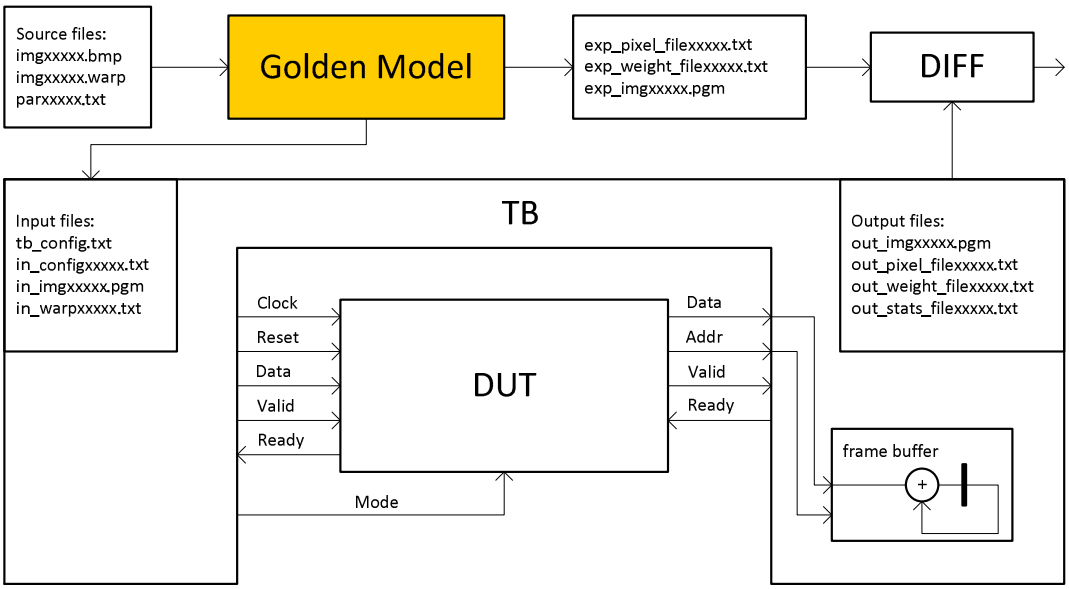


Figure 6.1: Functional verification setup.

6.3.2 Design for Testability (DFT)

Automated Testpattern Generation

6.4 Results

If you only have very few results, it might be a better approach to insert them into this chapter (instead of putting the results into a separate one).

Chapter 7

Results

If you have a large amount of results you can move them to this separate chapter.

7.1 First Section

7.2 Second Section

Chapter 8

Conclusion and Future Work

Draw your conclusions from the results you achieved and summarize your contributions. Comparisons (e.g., of hardware figures) with related work are also appropriate here. Point out things that could or need to be investigated further.

8.1 First Section

8.2 Second Section

Appendix A

Task Description

Include the task description **pdf** you got from your assistant(s) with the `\includepdf` command.

Appendix B

Declaration of Originality

Include the declaration of authorship with the `\includepdf` command (sign it and scan it). For more information about plagiarism, please visit `https://www.ethz.ch/students/en/studies/performance-assessments/plagiarism.html`

- **English version:** `https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf`
- **German version:** `https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiat-eigenstaendigkeitserklaerung.pdf`

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

This is a sample title

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
First	Student
Second	Student

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 01.01.2000	First Student Signature
	Second Student Signature

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Appendix C

File Structure

Describe how the project directories/files are organized, e.g.:

```
/
├── README ..... A README with some general information about the project.
├── 01_report ..... The source files of the project report.
├── 02_presentation ..... The source files of the presentation.
└── 03_designflow ..... Some designflow-specific files.
```

What needs to be done to run an RTL simulation (stimuli generation, compilation...)?

Appendix D

Datasets

If you have a data set comprising several test images, you could depict and describe them here. Use a simple naming scheme such that you can easily refer to certain elements of this data set in the text.

Appendix E

More Evaluation Results

If you conducted an extensive evaluation you could move surplus graphs/results to the appendix.

Appendix F

Algorithms / Tables

Large algorithm boxes and tables may clutter your chapters and impair the readability. If they are not very important, consider moving them to the appendix as well.

Appendix G

ASIC Datasheet (<Chipname>)

If you have designed an Application-Specific Integrated Circuit (ASIC) during your work, you should include a datasheet for your chip into the report. As soon as you start testing your fabricated chip, you will be glad to have such a datasheet. An example structure of such a datasheet is given in the following. For more inspirations on what you may include in your datasheet, have a look at the datasheet of a commercial Integrated Circuit (IC).

G.1 Features

- Lorem ipsum dolor sit amet, ...
- Lorem ipsum dolor sit amet, ...
- Lorem ipsum dolor sit amet, ...
- Lorem ipsum dolor sit amet, ...

G.2 Applications

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.3 Description

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.4 Packaging

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.5 Bonding Diagram

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.6 Pin Map

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

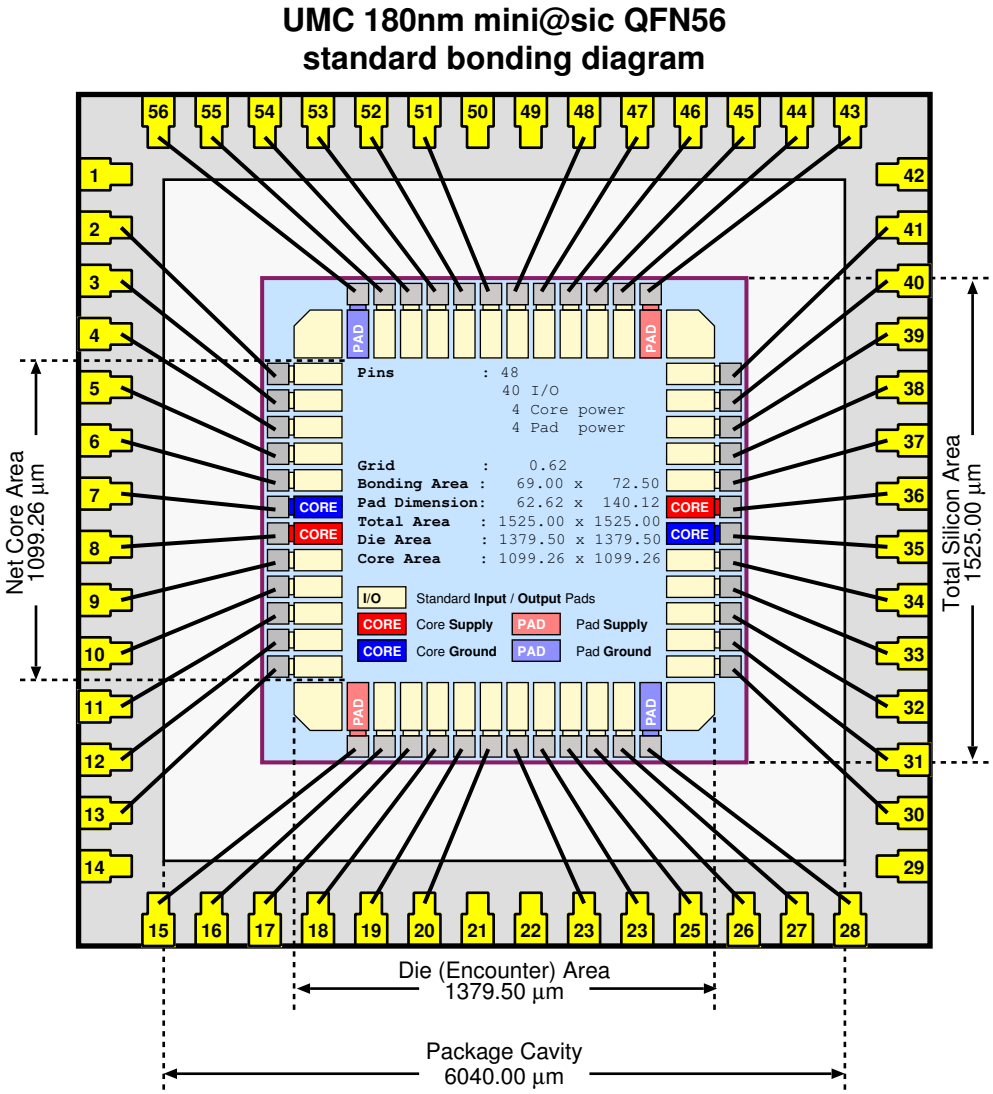


Figure G.1: Bonding diagram.

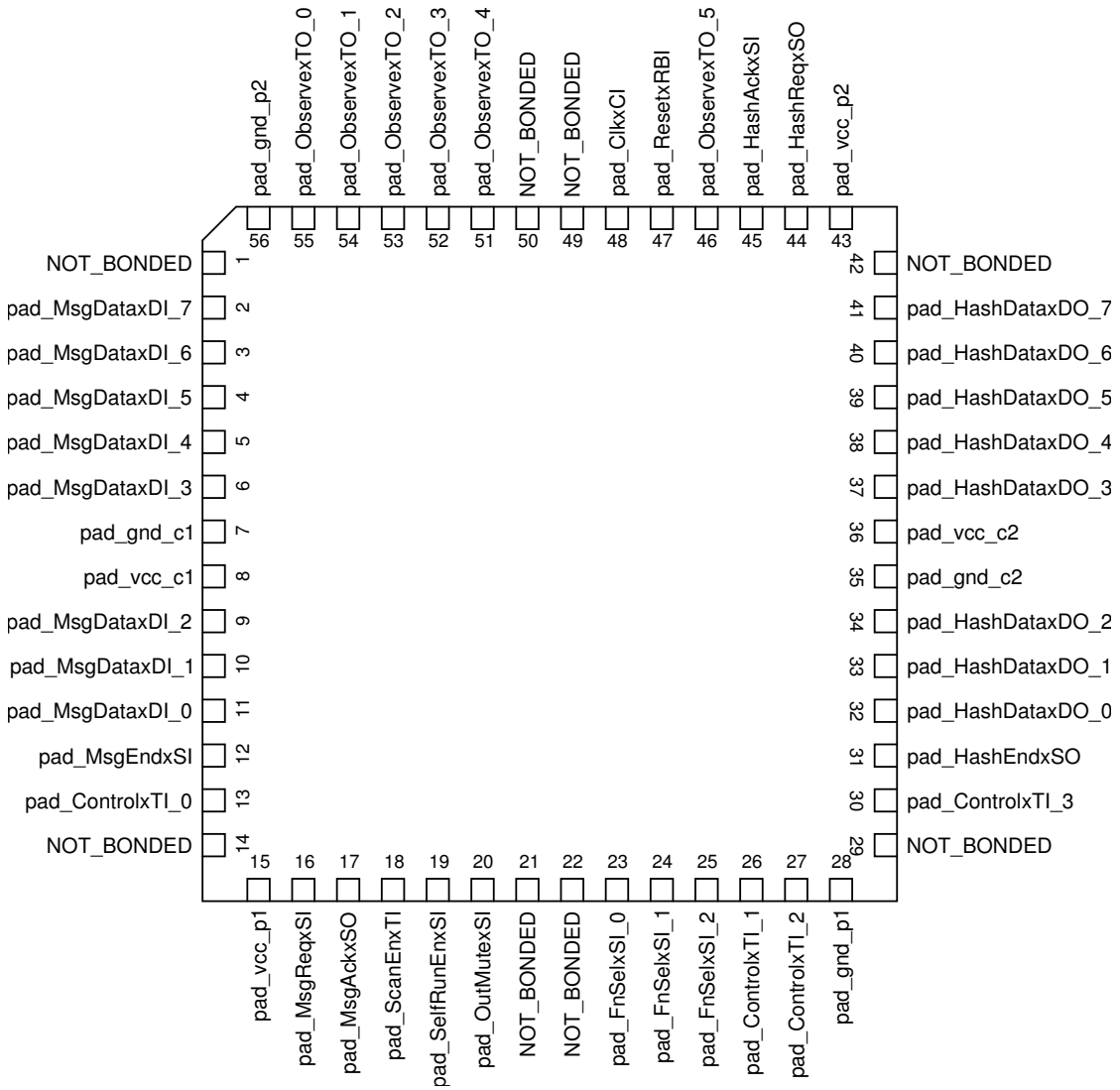


Figure G.2: <Chipname> pinout.

G.7 Pin Description

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.8 Interface Description

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.9 Register Map

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.10 Operation Modes

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.10.1 Functional Modes

G.10.2 Test Modes

G.11 Electrical Specifications

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

G.11.1 Recommended Operating Regions

G.11.2 Absolute Maximum Ratings

Appendix H

The Template Directory Structure

This LaTeX framework suitable for creating reports spreads over various directories and files. In order to give you a short overview of this structure, the respective directories and the contained files are described in the following:

- `/`
  - `README` ..... README file with a quick start guide.
  - `Makefile` ..... Makefile with some LaTeX related build targets.
  - `report_template.tex` ..... The main LaTeX file of the report document, which further loads other (content) files.
  - `bib` ..... Contains bibliography related files.
    - `main.bib` ..... Bibliography file.
  - `content` ..... Contains the actual source files of your report.
    - `*.tex` ..... Here, multiple content files are provided.
  - `figures` ..... Contains the images which are loaded during your report.
    - `eth_logo.*` ..... ETH logo in Encapsulated PostScript (EPS) and Portable Document Format (PDF) format.
    - `titlepage_logo.*` ..... Titlepage logo in EPS and PDF format.
    - `asic_pinout.*` ..... Sample pinout of an ASIC in EPS and PDF format.
  - `figures_raw` ..... Contains the raw sources of your figures.
    - `titlepage_logo.obj` ..... Tgif titlepage logo source.
  - `glossaries` ..... Contains glossaries.
    - `glossaries.tex` . The glossaries file containing both the entries of the list of acronym entries and the entries of the main glossary.
  - `preamble` ..... Contains preamble information of the document.
    - `preamble.tex` ..... Preamble of the report document.

Appendix I

LaTeX Tips

Writing a report with LaTeX may not be as intuitive as it is the case with What You See Is What You Get (WYSIWYG) editors. Especially if you are using LaTeX (more or less) the first time, some problems with the syntax will occur. In general, the present document should already serve as a good starting point for your report and in the best case you only have to insert the content of your project based on this framework.

Nevertheless, I will try to give some useful tips with regard to LaTeX throughout the next sections, which may help you to increase the quality of your documents even further. If you want to use any of the presented ideas, simply copy the LaTeX source code of the appropriate section to your on document and adapt it accordingly.

I.1 Compiling a LaTeX Document

Basically, either `latex` or `pdflatex` can be used in order to generate the document output in Device Independent File Format (DVI) or PDF format, respectively. Throughout this section I will solely use the `pdflatex` command for demonstration purposes (if you prefer a DVI document, just replace the `pdflatex` command by `latex`0).

Compiling a latex document at the Integrated Systems Laboratory (IIS) computers is, in general, as simple as executing the following command in a UNIX terminal window:

```
pdflatex <document_name>
```

Currently[1] a TeX Live version from the year 2008 is the default distribution at the IIS. In order to use the present LaTeX framework for your report, you have to use a more up-to-date version of TeX Live, because the framework uses some LaTeX packages which are

[1] State: July 2012

not part of the 2008 version. I suggest using the 2011 version of TeX Live. The simplest way to check that you can build the report template successfully, is by executing:

```
pdflatex-2011 report_template.tex
```

This should (re)generate the PDF output of the report template, i.e., the file you are currently reading through. If typing in the `-2011` postfix becomes annoying for you, you may add aliases into your `.cshrc` as follows:

```
alias latex 'latex-2011'
alias pdflatex 'pdflatex-2011'
```

If you also want to (re)build the glossaries (maybe you have added some acronyms or the like), you have to compile your report together with the glossaries as follows:

```
pdflatex-2011 your_report.tex
makeglossaries-2011 your_report
pdflatex-2011 your_report.tex
```

Furthermore, when you modify the references of your report (within the bibliography file), you also have to (re)run BibTeX in order to update your bibliography, i.e.:

```
pdflatex-2011 your_report.tex
bibtex-2011 your_report
pdflatex-2011 your_report.tex
pdflatex-2011 your_report.tex
```

I.2 Figures

In order to include an image into your report (as it has been done within in the previous sample chapters), you may use the `figure` floating environment. With that, LaTeX will take care of placing them nicely and you can focus on the actual content of your document. Figure **??** shows an example of how to insert a single figure.

Figure I.1: Standard ETH logo.

If you want to place multiple figures side-by-side, you can do this with the use of `minipages`. Figure ?? and ?? illustrates an example.

Figure I.2: Left ETH logo.

Figure I.3: Right ETH logo.

In order to create a single figure with multiple subfigures, you can do this as presented in Figure ??

(a) Left ETH logo.

(b) Center ETH logo.

(c) Right ETH logo.

Figure I.4: Multiple ETH logos as subfigures.

I.3 Tables

Tables in LaTeX allow you to present your results quite nicely. Table ?? shows a standard table.

Table I.1: Standard table.

Row 1 - Column 1	**Row 1 - Column 2**	**Row 1 - Column 3**
Row 2 - Column 1	Row 2 - Column 2	Row 2 - Column 3
Row 3 - Column 1	Row 3 - Column 2	Row 3 - Column 3
Row 4 - Column 1	Row 4 - Column 2	Row 4 - Column 3

Sometimes you may want to add a table which stretches one of its columns in order to reach the full width of the document. Such an example is shown in Table ??.

If you need to place two tables next to each other, you may use an approach based on `minipages` as shown in Table ?? and Table ??.

I.4 Creating Glossaries

In order to generate a glossary within your report (e.g., a list of acronyms or an actual glossary), take a look into the file `glossaries.tex`. There, you will find some examples

Table I.2: Stretched table.

Row 1 - Column 1	**Row 1 - Column 2**	**Row 1 - Column 3**
Row 2 - Column 1	Row 2 - Column 2	Row 2 - Column 3
Row 3 - Column 1	Row 3 - Column 2	Row 3 - Column 3
Row 4 - Column 1	Row 4 - Column 2	Row 4 - Column 3

Table I.3: Left table.

Row 1 - Column 1	Row 1 - Column 2
Row 2 - Column 1	Row 2 - Column 2
Row 3 - Column 1	Row 3 - Column 2
Row 4 - Column 1	Row 4 - Column 2

Table I.4: Right table.

Row 1 - Column 1	Row 1 - Column 2
Row 2 - Column 1	Row 2 - Column 2
Row 3 - Column 1	Row 3 - Column 2
Row 4 - Column 1	Row 4 - Column 2

on how to define an acronym as well as a glossary entry. If you want to reference one of the acronyms within your report, you can do it the same way as I did it with the Light-Emitting Diode (LED) right here (just take a look into the source code).

As already mentioned in Section **??**, you have to rebuild your glossaries in order to display changes. For that, you first have to build your document using `latex-2011` or `pdflatex-2011` in a shell window, or the *build-button* in your preferred LaTeX editor GUI. Next, you have to call `makeglossaries-2011 <file_name>` in a shell window[2], followed by another build process of your main source file, i.e.:

```
pdflatex-2011 your_report.tex
makeglossaries-2011 your_report
pdflatex-2011 your_report.tex
```

I.5 Creating Algorithm Boxes

Algorithm boxes in LaTeX allow you to present your algorithms in pseudo code as shown in the following example:

[2]The `makeglossaries` script is a Perl script available at the IIS computer system and should also be part of most TeX distributions.

Algorithm 1: disjoint decomposition

input : A bitmap Im of size $w \times l$
output: A partition of the bitmap

```
1  special treatment of the first line;
2  for i ← 2 to l do
3  |  special treatment of the first element of line i;
4  |  for j ← 2 to w do
5  |  |  left ← FindCompress(Im[i, j − 1]);
6  |  |  up ← FindCompress(Im[i − 1, ]);
7  |  |  this ← FindCompress(Im[i, j]);
8  |  |  if left compatible with this then // O(left,this)==1
9  |  |  |  if left < this then Union(left,this);
10 |  |  |  else Union(this,left);
11 |  |  end
12 |  |  if up compatible with this then                    // O(up,this)==1
13 |  |  |  if up < this then Union(up,this);
   |  |  |  // this is put under up to keep tree as flat as possible
14 |  |  |  else Union(this,up);                          // this linked to up
15 |  |  end
16 |  end
17 |  foreach element e of the line i do FindCompress(p)
18 end
```

Appendix J

General Writing Guidelines

As soon as you get familiar with the syntax of LaTeX (and I can promise you, you will get familiar with it quite quickly as soon as you start writing your reports with LaTeX), some more general writing tips might become of interest for your. Therefore, I collected a few general writing guidlines in the following sections, some of them with regard to LaTeX, some of them not.

Placement of Floating Environments Figures and tables are the two most prominent examples for floating environments. Although the figure examples presented in Section **??** use `[htbp]` to tell LaTeX how to place them, you should normally only use the `h` parameter if you really require it. Since LaTeX then at first tries to place the figure at the same position as its source code, this somehow contradicts with the actual purpose of the `figure` environment. So, in general, try to place floating environments using one of the following parameters:

t Place the floating environment on **t**op of a page.

b Place the floating environment on the **b**ottom of a page.

p Puts the floating environment on a single *floating **p**age* with other floating environments.

Positioning of Figure and Table Captions Captions of figures are, in general, placed below the actual figure, whereas captions of tables should be placed on top of them. Section **??** and **??** contain some examples for figures and tables, including correct placement of captions.

Avoid Unneccessary LaTeX Packages Although there are so many "cool" LaTeX packages available everywhere on the Internet, try to use only those, which you really require. The main problem with loading too many, more or less unknown, packages is that some of them might redefine some commands, etc., which are used by another package which asumes that command to be the original one. Keeping track of these changes and the relations between different packages, is quite annoying and takes quite a lot of time. Hence, keep your preamble simple with regard to packages.

Make Use of Vector Drawings Since LaTeX handles vector drawings pretty good and their scalability allows you to print them in any resolution, prefer them compared to their pixel counterparts and use them whenever possible.