UPPSALA
UNIVERSITET

1DT109 Accelerating Systems with Programmable Logic Components

1 Aim of the project

After the project students shall be able to design hardware accelerators; know how to integrate these into an embedded system consisting of both hardware and software; estimate performance, circuit complexity and power consumption of the components; and be able to determine which solution is the best from the system perspective.

2 Platform

You are required to use the provided MiniZed board as the default hardware platform. Xilinx Vivado is the default EDA tool. Please read the MiniZed board hardware user guide[1] before starting the development task. Please specify the version of Vivado you are using in the report submission (we recommend Vivado HLx Edition 2018.3 or 2019.1). A tutorial of how to install and use Vivado can be found at the Xilinx website. We also provide a tutorial for the installation in the course materials of 1DT109. See the student portal for more recommended resources (might get updated after the project start).

3 What to do

In the project, the students (working in pairs of two) will develop three versions of an algorithm with 1) C/C++ programming language, 2) Verilog HDL, and 3) high level synthesis (HLS) design methodology. The students should also evaluate and compare the performance of the three versions of the algorithm.

We do not have restrictions on the algorithm you will implement. However, we recommend you implement an FFT algorithm: you first implement a software version of FFT and later compare it with the FFT accelerators you implement in HDL and HLS. However, it is also OK that you choose to implement something else. Please ensure that the topic you choose involves a compute-intensive design (a design that processes large amounts of data). For example, a FIR algorithm is a good candidate, graphic analysis algorithm is another good one. But the traffic light control algorithm shown in the course is not a viable one. If you choose to define your own project, please state clearly

[1] http://zedboard.org/sites/default/files/documentations/MiniZed-HW-UG-v1-0-V1_0.pdf

in the report what algorithm you will implement, what the algorithm is doing, why is it important to accelerate such an algorithm. If you choose another algorithm than FFT, you also need to **get confirmation** from us as soon as possible.

4 Tasks

The core tasks of the project are 1) developing different implementations of the algorithm you choose; 2) exploring ways to optimize your designs; 3) evaluating and comparing the implementations; 4) answering the following questions when developing/optimize/evaluating your designs and conveying the answers in your final report.

You can utilize the various versions of your design to cross-check each other. Be sure that they yield the same outputs given the same inputs.

1 Software Implementation

1.1 Data types

What data types will you use in your design? Integer, fixed point numbers, or floating-point numbers? Please be consistent with data types across your designs, *i.e.*, if you choose to use integer, all the three versions of your design should use integer for fair performance comparison. For a thorough evaluation, you are encouraged to evaluate the performance difference using different data types. [**Note!** The ARM A9 processor on your MiniZed board include a double-precision FPU[2]].

1.2 On-chip vs off-chip memory

Discuss in which way it is beneficial to add separate on-chip memories and place one or both of RAM or/and ROM data structures of your code on them, *i.e.*, what data should be stored in the on-chip memory? What data should be stored in the SDRAM?

1.3 Evaluate your design in MiniZed

Run your C/C++ implementation of the FFT code on the MiniZed board ARM A9 core. Record the program execution time. *[Optional]* Find a way to estimate/measure the power consumption and energy efficiency of the core running your program.

1.4 Testbenches

Please state clearly in the report what testbenches you use to evaluate your design. Be sure to consistently use the same testbenches/input files to test all your design versions. If you choose to implement the FFT algorithm, you can find testbenches on the student portal, but you will need to add further tests yourself. Please use that file to test all your designs.

2 Hardware Implementation

[**Note!** You are encouraged (but not required) to separate control logic and data-path components in your HDL design. If you have not been familiar with the control logic + data-path design methodology, please refresh your knowledge with the slides from 1DT093 Computer Architecture on processor data-path and pipelining. You can find them in the student portal of 1DT109 Accelerating Systems with Programmable Logic Components, under resources/tutorials.]

2.1 Number Representation

[2] https://static.docs.arm.com/ddi0408/i/DDI0408I_cortex_a9_fpu_r4p1_trm.pdf

What and how do you represent numbers in your hardware accelerator? What is the bit width of a number? What types of numbers will you use, integer, fixed point numbers, or floating-point numbers, or all of them?

2.2 CFG

Draw the control flow graph (CFG[3]) of your design

2.3 Show the controller and data-path components

If you choose to separate controlling logic and data-path, draw a block-level diagram illustrating the connections among the controller and data-path components. The figure does not need to be overly detailed. Just a brief architectural-level figure illustrating how you connect the controller to the data-path components. Show the inputs/outputs of the controller and data-path components.

2.4 Data-path Design

What are the data-path components (such as adders, multiplexers, multipliers, registers, *etc.*)? How do you connect them together?

2.5 Controller Design

Has your design a controller/controlling logic? How do you design it and connect it with the data-path components?

2.6 Pipelined Design

Is your design pipelined or non-pipelined? What components are pipelined? If so, please illustrate the pipeline stages and the functionality of each stage.

2.7 Timing Sequence

Please show the timing sequence reported by Vivado behavioral simulation in the final report. Please make sure that your design does not suffer timing error.

2.8 System integration

How do you plug the HDL accelerator into the whole system via the AXI bus? How to invoke your customized hardware accelerator from high level code?

2.9 Prototyping and evaluating your design in MiniZed

Make sure that the design is synthesizable. Show the synthesis results you get from Vivado. Report execution time of your design and compare it with the results you get from the software version. *[Optional]* Estimate/measure the power consumption and energy efficiency of the HDL accelerator. Compare it with the power consumption of the software version.

3 HLS Implementation

3.1 There are several mapping patterns for HLS to map the software statement into the corresponding hardware structure. Please examine how each of the following statements in your design is mapped from HLS to the corresponding hardware structure.

- Function arguments
- Global variables

[3] https://en.wikipedia.org/wiki/Control-flow_graph

- Simple and nested loops
- Sub-function calls

3.2 System integration
How do you plug the HLS accelerator into the whole system via the AXI bus? How to invoke your customized hardware accelerator from high level code?

3.3 Examine the schematics of HDL and HLS implementations. Is there any difference? What is the cause of those differences?

3.4 Evaluate the design with your MiniZed board.
Report execution time of your design and compare it with the results of the software and hardware versions. *[Optional]* Estimate/measure the power consumption and energy consumption of the HLS accelerator. Compared it with the power consumption of the software and HDL versions.

4 [Optional] Deeper design space exploration
You can do more design space exploration to test your designs from different angles. For example, you can choose different windows in the FFT algorithm to test its sensitivity to different window lengths. In the project so far, your design has been implemented as either a total software or hardware solution. In this task, you can manipulate your design to be a software/hardware co-design, *i.e.*, you will decide which components are better to be implemented in software and what should be implemented in hardware.

5 Deadlines and final submission

The project starts on **13th May** and ends on **3rd June** (sharp 3 weeks). By the end of the deadline, each team should submit a report representing the answers to those tasks listed in Section 4. Organize your report into the following structure.

1. **Project description.**
   The report should clearly define the project. If you choose to implement algorithms other than the suggested FFT algorithm, please define the new topic in the report. What is the functionality you choose to implement? Why is it important to do that?

   Optimization method.
   This section should reveal the insights of why/how you optimize your design. For example, Task 1.2 should be discussed in this section. You can also discuss how your design is pipelined, and then show the pipeline stages of your design in this section. Compare the results you get before/after you apply those optimizations.

   [Hints. Design optimizations can include but are not restricted to loop unrolling, strength reduction, pipeline balancing, *etc*.]

2. **Design.**
   The report should state clearly how the designs are implemented. All tasks listed in Section 3 (except the ones that are relative to design optimizations and evaluations) should be reported within this section. Do not articulate too much on each task. Keep the answer simple and concise!

3. **Evaluation.**
   Report the evaluation results in this section. Execution time comparison of the three versions of the algorithm is mandatory. Power consumption and energy efficiency results are optional.

4. **Deliverable.**
   Please upload your code to the student portal. For this, export your project from Vivado (File/Project/Archive), and upload the created zip-archive. Please document clearly in the report on the instructions to replicate the results. [optional] You can also upload a video to the student portal demonstrating your design.

Please keep your report concise. Directly and closely follow the example report structure and answer all evaluation questions with concise words. The length of a report should be **less than 10 pages**.

6 Administration

Please read the following instructions before you start the project.

1. Start to draft the report at day 0 of your project. Writing partial results down is not only helpful to record your progress but also very important to give you a hindsight on what you have achieved and what you can achieve later. **The project work and the quality of the report are factors influencing your course grade.**
2. Provide two weekly progress reports on **19th and 27th May**, correspondingly. Those reports should be uploaded on the student portal. Each weekly report should be at most two pages, and be an intermediate milestone towards your final report. The weekly report should give a summary of the tasks you worked on in the past week, highlight issues you faced, and describe how you solved them. You can reuse text from the weekly reports in the final project report.
3. Students are working in groups of two. We recommend that each group works on a private repository on bitbucket or github.
4. Each group should work independently without cooperation with other groups; each group should develop its own implementations of the chosen algorithm, and not copy code from public repositories, example projects, *etc*. While the project is ongoing, groups must not upload their implementations on public repositories. Cheating is strictly forbidden throughout the whole project. Please notify us on suspected cheating.
5. Each group is sent one MiniZed board for this project. You should send back the board using the provided envelope at the end of the project.

7 Contacts

There are two help sessions for the project:

- 13:15 – 17:00 20th May,

- 13:15 – 17:00 28th May.

You are encouraged to use these to discuss your progress with us, get feedback, and ask any questions you might have about the project. You are also welcome to reach us using the course forum on the student portal, or using the Slack workspace. The project help sessions are going to take place on Zoom (same link with the lab sessions).
Good Luck!