

COMP3211/9211 Computer Architecture 2011 Project

Encryption/Tagging Processor Design

Part 2 Pipelining

This part of project is to be completed by each project group. Your design, implementation and results are to be presented in week 10.

Deliverables for Part 2 of the project contribute to the group report that is due on Friday, 27 May (week 12).

Problem Overview

Refer to the specification of Part 1 for an overview of the project.

Objectives of Part 2

- To pipeline your design from Part 1 in order to obtain a faster, pipelined solution to the problem;
- To describe your pipelined processor in VHDL;
- To simulate and verify your design by running the encryption/tagging function to encrypt a string on your processor core; and
- To analyze the performance gain of your pipelined processor and any area overhead as compared to your single cycle processor design.

Part 2 Requirements

- **Examine your performance as a team member in the Project Part 1 and any issues you encountered. Take a positive step to resolve or alleviate such problems that may jeopardize or hinder the success of Project Part 2;**
- As in Part 1, each project group is required to make a detailed project plan which may include
 - tasks that need to be performed in order to complete this project part,
 - the schedule of the tasks,
 - the role of each member for these tasks,
 - the test methods for each task,
 - a project management strategy to ensure that the project work is carried out smoothly and completed on time with as good quality as possible, and
 - strategies for project design presentation based on your experience from part 1.

- Your design must meet the following specific requirements:
 - It can encrypt a plain text of up to 100 characters;
 - Your VHDL model must take the component delays into account. The related timing information for some functional components is as follows:
 - i. PC register propagation delay: 0.5ns
 - ii. Adder: 8-bit adder 1ns; 16-bit adder 1.5ns;
 - iii. 8-bit ALU: 1.5ns; 16-bit ALU: 2ns;
 - iv. Memory/Register file latencies (note the delay formula is used just for easy of specification and may not true for large memories)
 - 1. 8k x 8-bit: $(0.3 + \log_2 k \times 0.4)$ ns, where $k = 1, 2, 4, 8, \dots$
 - 2. 8k x 16-bit: $(0.5 + \log_2 k \times 0.5)$ ns, where $k = 1, 2, 4, 8, \dots$
 - v. 2-to-1 MUX: 0.2ns;
 - vi. Logic gates with fanins of 4 or below: 0.2ns
 - vii. Control unit: 1.5ns
 - viii. Stage register set-up time: 0.25ns

The following serves as an example of some tasks (and suggestions) that you may consider for this project part.

1. Using the longest delay of the functional components in Part 1 as a guide to split the data and control paths of the single-cycle processor into processing stages that allow the delays to be distributed as evenly as possible across the stages.
2. Determining extra hardware resources required and the maximal clock rate of the pipelined processor.
3. Analyzing potential hazards (Note: your code must not be rescheduled to eliminate hazards that are present), and determining how to handle such hazards.
4. Completing the pipelined processor design with a block diagram.
5. Using an execution-timing diagram to illustrate the instruction execution flow of your encryption program. **Note:** Repeated sequences may be omitted from the illustration, but the hazards related events should be specially identified in the illustration diagram.
6. Developing a VHDL model for the pipelined processor and verify your design.
7. Analyzing the performance improvement and related area overheads of your design as compared to your single cycle processor design. The area can be measured in components.

Part 2 Presentations

Refer to the specification and assessment guidelines for Part 1 of the project for an overview.

Guidelines for assessing Part 2 presentations will be made available by the end of Week 8.

Project Deliverables

Please refer to the specification for Part 1 of the project for a description of the requirements.