EDA385
A Parallelized Hash Generator System

Niklas Aldén
ael10nal@student.lu.se

Gabriel Jönsson
ael10gjo@student.lu.se

Jonathan Sönnerup
ael10jso@student.lu.se

September 15, 2014

Department of Computer Science
Lund University, Faculty of engineering

1 Concept

This project will be a proof of concept of how the parallelism in an FPGA can be used to compute a large number of password hashes simultaneously in order to reduce time to crack passwords. The concept can easily be scaled up by increasing the the size of the FPGA.

2 Implementation

The project will be implemented on a *Spartan 6* FPGA–board, using a *Microblaze* processor to run the software and communicate with the custom hardware IP blocks.

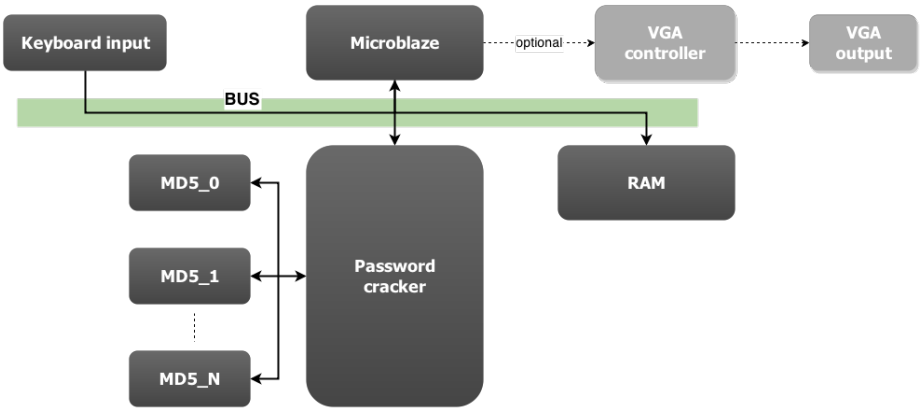

Figure 1: System block diagram

Microblaze is a softcore microprocessor that can be implemented on programmable logic, e.g. FPGA:s. It will run the software and handle the communication between the blocks.

Password cracker uses the MD5 hash function to generate a hash from a random generated character sequence. The outputted hash is compared to a given hash (the one which is supposed to be cracked) to determine what the original password was. By doing this repeatedly in parallel and in hardware the calculations are done rapidly. The numbers of MD5:s is only limited by the size of the board.

MD5 is a hash function which maps an N-bit input to a fixed length output, 128 bits. The reason for using a hash function is because it is a non-invertible function. When storing passwords in a database, it is preferable not to store it in clear text, therefore the use of hash functions is evident.

Keyboard input allows the user to interact and choose what task the system should run. Either create a new RT or crack passwords.

VGA controller and VGA output is an optional implementation that will be added if there is time, but it is not essential for the functionality of the system.

3 Schedule

A draft of the schedule is shown in figure 2.

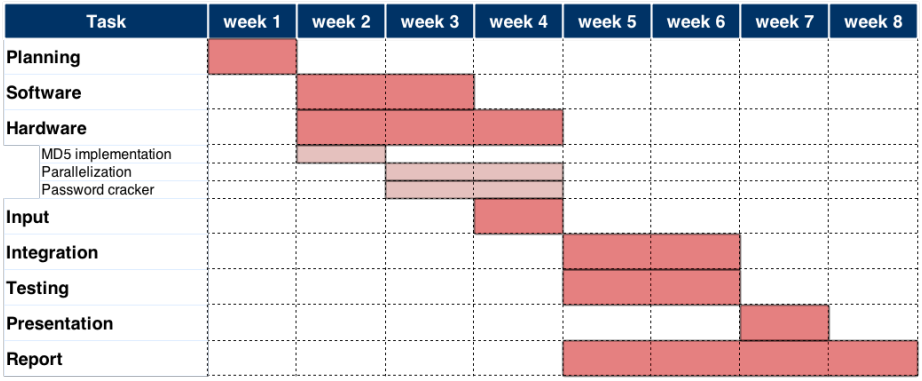

Figure 2: Proposed schedule