The Design of the SHA1 Processor

ECE 111 Final Project

Winter Quarter 2016

Paul Do

Surya Manchikanti

03/15/16

Introduction

Cyber security is increasingly becoming an important field to be aware of. With estimated damages of a break in costing anywhere in the millions to billions of dollars, every step counts for a company with sensitive assets. The SHA-1 algorithm was used to verify a file's integrity to predict whether a Man In The Middle (MITM) attack occurred. Designed by the NSA in 1993, it has several variants that make it harder to compute the hash but also to make it more resistant to forging. Although SHA-1 is no longer considered secure for opponents with adequately funded opponents.

Example of SHA algorithm being used: Tor, Tails OS, Cent OS, Bittorrent clients, etc.

Description of the SHA-1 Algorithm

The SHA-1 algorithm uses a sequence of rounds to parse through the data by blocks generating a hash value every 512 bits. While processing these bits in 512 bit blocks, circular shifts and logic combinations of the input bits ensure that a single changed bit changes the hash value by a compounded large margin.

Design Details

Our design process started off with a straight-forward approach. We allocated clock cycles for separate steps of the SHA-1 algorithm (reading in data, padding the message with zeros, computing new hash values, etc.). This approach gave us a first design that worked, but was very slow. (Average 180 clock cycles per message block)

The biggest optimization we did was interleaving the reading of the message and all the necessary hash value computations into one clock cycle. We ended up with a design that performed much faster (81 clock cycles per message block), at the expense of some area.

Working in Teams

Working together was an absolute pleasure, we pair programmed with Paul in the driver seat and Surya in the coder seat. Paul researched syntax and other conventions to ensure Surya had all relevant information he needed to code the program.

Paul and Surya met up regularly to code out significant chunks of code. Constant pondering of design decisions led us to create a rather efficient design. By keeping true to our roots as a computer engineer and electrical engineer, we tackled this problem as a combination of challenges. One being electrical engineering design and the other one being computer science and optimizing for speed.

We've learned that expectations must be set for every meeting to ensure steady progress. Functioning in a team is highly effective especially when motivated individuals are brought together to solve a challenge.

We think that the instructors can make us solve smaller challenges along the way to keep our coding skills sharp throughout the quarter in lieu of larger less-frequent assignments.

Summary of Results

1. #ALUTS: 1842
2. #Registers: 936
3. Area = 2778
4. Clock Period: 8.48ns
5. #Cycles for testbench_v6: 244
6. Delay: 2067.97ns
7. Area x Delay: 5.74 x 10^-3

References

- Wikipedia: https://en.wikipedia.org/wiki/SHA-1
- IETF Specification: http://www.ietf.org/rfc/rfc3174.txt
- SHA-1 Project PPT (finalprojectv2.ppt) by Professor Bill Lin: http://cwcserv.ucsd.edu/~billlin/classes/ECE111/final_project.php