Design of High-Throughput SHA-256 Hash Function based on FPGA

Shamsiah binti Suhaili
Department Electrical and Electronic Engineering
Faculty of Engineering, Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak, Malaysia
sushamsiah@unimas.my

Takahiro Watanabe
System LSI
Graduate School of IPS, Waseda University
Kitakyushu-shi, Fukuoka, 808-0135 Japan
watt@waseda.jp

Abstract—**Nowadays, security has become an important topic of interest to researchers. Different types of cryptography algorithms have been developed in order to improve the performance of these information-protecting procedures. A hash function is a cryptography algorithm without a key such as MD5, RIPEMD160, and SHA-1. In this paper, a new SHA family is developed and designed in order to fulfil the cryptographic algorithm performance requirement. Thus, SHA-256 design and SHA-256 unfolding design based on reconfigurable hardware have been successfully completed using Verilog code. These designs were simulated and verified using ModelSim. The results showed that the proposed SHA-256 unfolding design gave better performance on Arria II GX in terms of throughput. The high throughput of SHA-256 unfolding design was obtained at a data transfer speed of 2429.52 Mbps.**

Keywords— Cryptography algorithm; FPGA; SHA256 Hash Function; Unfolding transformation.

I. INTRODUCTION

NIST (The National Institute of Standards and Technology) standard specifies the adoption secure hash algorithms such as SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 [1]. Hash Function algorithms are used during data transmission to produce the message digest. Therefore, it becomes an essential tool for embedded security in e-mail, internet banking, and other applications. A hash function takes an arbitrary-length message input to produce a fixed-length output. A hash function is a one-way hash function; it is difficult to invert a hash value to a message input. Furthermore, it is computationally infeasible to find a message that produces the same hash value. These properties become an important aspect to ensure that a hash function can work properly.

The purpose of this paper is to provide a high-speed hardware implementation for the SHA-256 algorithm. This algorithm is synthesised and implemented based on Arria II GX. The motivation of this design is to increase the performance of SHA-256 algorithm. The organisation of this paper is as follows: Section 2 describes the SHA-256 Algorithm; Section 3 presents the proposed SHA-256 Algorithm. The implementation results are discussed in Section 4 together with a comparison with other SHA-256 designs. Finally, the last section provides the conclusions of this project.

II. SHA-256 ALGORITHM

SHA-2 consists of four different types of hash functions such as SHA-224, SHA-256, SHA-384, and SHA-512. The output length of these hash algorithms depends on the SHA-2 length ranging from 256 to 512-bit. In this paper, the SHA-256 hash function has been designed. This section describes the SHA-256 algorithm together with the block diagram of this algorithm. Each SHA-256 algorithm can be divided into two stages: pre-processing and hash computation. Pre-processing involves padding a message and parsing the padded message into m-blocks. Initialisation values are set to be used in the hash computation. Hash computation produces a message schedule from the padded message. The output hash value generated by hash computation is used to determine the message digest. Hash computation comprises message schedule, functions, constants and word operations that are generated iteratively in order to obtain a hash value. Table 1 shows the characteristics of the SHA-256 hash function. The security of SHA-256 hash function depends on the size of the hash value.

TABLE I. CHARACTERISTICS OF SHA-256 HASH FUNCTION

Hash Function	**SHA-256**
Size of hash value (n)	256
Constants Kt number	64
Message block size (m)	512
Word size	32
Numbers of words	8
Digest rounds number	64

The first step of the SHA-256 hash function is pre-processing; the input message is padded. The process of padding the message starts after getting the message input, and a single 1-bit is added at the end of the message. Then, it is followed by n 0-bit until the length of the message is congruent to 448 modulo 512. The last 64-bit is reserved for calculating the length of the message. Thus, the overall message input is 512-bit.

Figure 1 shows the message schedule of a SHA-256 algorithm. The message, W_t of SHA-256, is computed by the message scheduler as shown in Figure 1. For $0 \le t \le 15$, a

This project is supported by Universiti Malaysia Sarawak (UNIMAS) under grant code L18403/F02/00/Osaka/Research

978-1-5386-0475-5/17/$31.00 ©2017 IEEE