Optimization Implementation of SM3 Algorithm Based on 64 Rounds Grading Calculation

Optimization Implementation of SM3 Algorithm Based on 64 Rounds Grading Calculation

Peifeng Chen

Jilin Radio and TV University Jilin, China

quake0490@sina.com

Abstract. This electronic document analyzes the calculation process of SM3 algorithm and finds that the iterative compression process is suitable for optimizing and improving with the characteristics of hardware. Therefore, a parallel implementation strategy of running water line and computing is proposed during the iterative compression process. This strategy makes full use of the hardware's parallel features, making the SM3 algorithm the most complex and time-consuming.In the 64-wheel calculation project, it is possible to allow successively entered data to be calculated in different rounds at the same time. In the case of continuous large traffic data, the computational efficiency is greatly improved, and the delay is reduced, and there is no case where the calculation module waits for data. This allows computing resources to be used efficiently. The delay can be as low as 0 at 125 MHz clock. 008μs. in addition, the parallel mode of multiple algorithm kernel modules is used to make the algorithm kernel become a relatively independent module, which greatly increases the scalability of the performance.

1. Introduction

With the development of network information technology, information security and Hash letter Numbers have become an increasingly important research topic. Hash function can map any finite length input message to a fixed-length output value a class of functions.

Based on the excellent security performance of the Hash function, it is widely used and is also an important part of a variety of information security chips. Typical SHA-1 algorithm and SHA-256 / 384/512 algorithm have the advantages of high non-linearity of compression functions, message filling and strong grouping, but it also has the disadvantage of threatening security. The National Cryptographic Administration has issued aSM3 hash algorithm [5]. However, SM3 has many iterations and many logical operations, which often can not meet the needs of high throughput. Based on the field programming gate Array (FPGA) device, this paper optimizes the SM3 algorithm, optimizes the compression function part by running water classification to reduce its time delay, and passes the running water classification algorithm kernel of multiple multiplexing compression functions.To improve the scalability of its performance.

2. SM3 algorithm Introduction

The SM3 algorithm is a hash algorithm based on packet iteration. This algorithm uses message word processing method combined with message word to realize message rapid diffusion and chaos in a local area. The SM3 hash algorithm is filled and compressed iteratively to generate a length of 256 bitsThe hash value. The SM3 algorithm is mainly divided into four parts.

2.1. Message fill preprocessing:
Add "1" to the end of the message and add K 0, where K is the minimum non-negative integer (L is the message length) that satisfies L +1 + K <S> 488mod 512. Then add a 64-bit bit bit string represented by the binary of length L. The filled message M 'is an integer multiple of 512. And group the message M 'by 512 bits: M' = B (0) B (1) ... B (N-1). Where N = (L + K +65) / 512.

2.2. Message extension:
Expand B (I) to generate 132 32-bit string W0, W1, ..., W67, W0 ', W1', ..., W63 'as follows. 1 Divide message group B (I) into 16 32-bit series W0, W1, ..., W15.

```
2 FORj = 16 TO 67
Wj ← P1 (Wj-16(W1-3)) (Wj-13) 7) Treatment of Wj-6
ENDFOR3
FORj = 0 TO 63
W'j = Wj <UNK> Wj +4
ENDFOR
```

2.3. Compression function CF calculation:
8 32-bit registersA, B, C, D, E, F, G, H, Initialized as A = 7380166f, B =4914b2b9, C = 172442d7, D = da 8a0600, E =

A 96f 30bc, F = 163138aa, G = e38 dee4d, H =B0fb0e4e. SS1, SS2, TT1.

TT2 is an intermediate variable, the compression function V (I +2) = CF (V(I), B(I)), the calculation process of $0 \leq I \leq N-1$ is described as follows:

```
FOR j = 0 TO 63
SS1← ((A <<< 12) + E + (Tj <<< j)) <<< 7
SS2←SS1 (A <<< 12)
TT1←FFj
(A, B, C) + D + SS2 + W'j
TT2←GGj
(E, F, G) + H + SS1 + Wj
D←C
C←B <<< 9
B←A
A←TT1
H←G
G←F <<< 19
F←E
NDFOR
P0
(X) = X(X <<< 9) (X <<< 17)
P1
(X) = X(X <<< 15) (X <<< 23)
FF1
(X, Y, Z) =
XYZ 0≤j≤15
(X∧Y)V (X∧Z)V(Y∧Z) 16≤j≤63
GGj
(X, Y, Z) =
```

XYZ 0≤j≤15
(X∧Y)V (－ X∧Z) 16≤j≤ 63
Tj =79cc4519 0≤j≤15
7a879d8a 16≤j≤ 63
V (i + 1)←ABCDEFGHV(i)
FOＲ i = 0 TO n － 1
V (i + 1) = CF(V(i) － B(i))

CF is a compression function, V (0) is 256 bits initial value IV =7380166f 4914b9 172442d7da 8a0600 a96f30bc163138aa e38 dee4d b0fb0e4e. When a message of all after the 512bit packet is processed, the last 512bit packet loses Outgoing is the message summary value.

3. Hardware Optimization of 2 SM3 Algorithm

3.1. Overall algorithm optimization

According to the rules of the SM3 algorithm, after the message is preprocessed, each512Bit groups can be calculated in parallel. Introducing the SM3 algorithm (2),(3) The budget is used as the algorithm kernel. First pair of generated 512 bitsGroup to schedule whether the first level of the algorithm core is idle, Select the kernel module that you want to input. Use multiple kernel modules Block to flexibly increase throughput and reduce latency. This article uses the Optimizer shown in Figure 1.

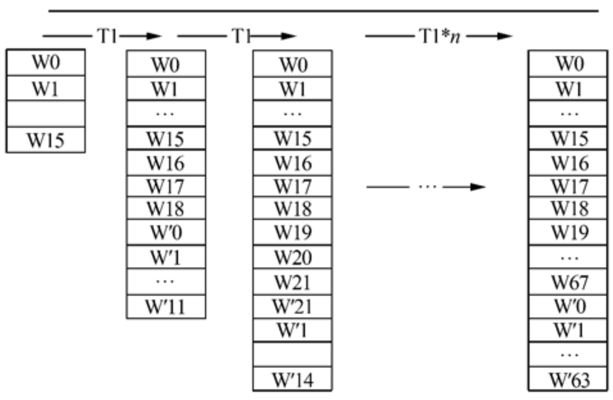

Figure 1. Message Extension Optimization Diagram

3.2. Optimization of Algorithm Core

Iterative compression calculation of 512 bits after grouping. This one. Part of it is the place that consumes the most resources. 132 outreach students in progress. Formed W0, W1, ...W67, W0 ', W1', ... W63 'and 64 rounds The process of substituting ABCDEF GH is the main factor that affects performance.

Optimization of message extensio. In the calculation of 132 extended words, each T1Time lost. Into a group of 512bit messages, each level of running water handles different messages at the same time one round of calculations. It takes T1 to calculate a set of Wj, Wj 'values. According to the principle of parallelism, T1The shortest is 7clk1 (clk1 is a message extension the working clock). But in order to match the subsequent 64 compression iteration. Combine, can do the appropriate delay and increase the water series.

Optimization of compressed iterations. For compressed functions, increase the number of ABCDEF GH registers Number, and use a 64 pipeline operation. Each Matching TimeT2 Run a round of calculations of 512 bits of data, 64 streams at the same time, One round of calculations for processing different data, every passing time T2.Water down Level 1, so that in continuous input, after calculating the first data Time required (64 * T2) After that, you can use every T2 The speed of time lost Out of a set of register ABCDEF GH values, and then perform XOR operations Output the last hash value. In single-level calculations, the longest delay calculation process is:

SS1←((A <<< 12) + E + (Ti <<< j)) <<< 7

SS2←SS1 (A <<< 12)

TT1 — GGj

(E, F, G) + H + SS1 + W'j

TT2 — GGj

(E, F, G) + H + SS1 + Wj

Last time.Addition operation, multiple addition operations executed in parallel to reduce addition operations. In this way, five levels of operations are performed in turn, that is, five clock cycles can be to calculate the values for SS1, SS2, TT1, TT2. A similar method is used for P0 (TT2) calculations. That is, P0 (TT2) can be calculated after three clock cycles Value. Calculate SS1, SS2, TT1, TT2 take five cycles and calculate P0 (TT2) takes 3 cycles, of which 1 cycle can overlap, so A total of 7 clock cycles. The clock cycle here is the working clock of the compressed iteration moduleclk2, IE T2 = 7clk2. As shown in Figure 2.

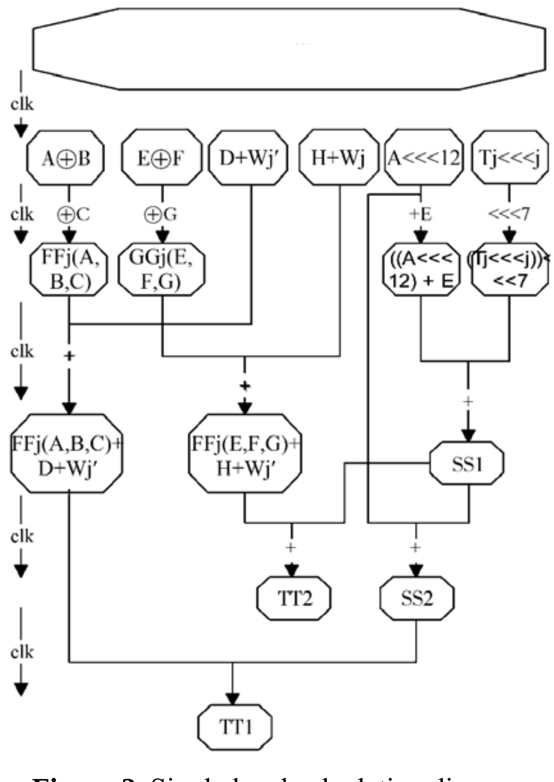

Figure 2. Single level calculation diagram

3.3. Optimization analysis

By parallel processing at the same time as message extension and compression iteration and the design of the pipeline structure can significantly improve the SM3The operating speed of the algorithm kernel is reduced while the delay is reduced. After Max {T1, T2} can perform a set of ABCDEF GH calculator calculations. In After each algorithm completes the calculation of the first-level iterative register, it is sufficient Receive new data input and call through multiple algorithms Obviously improving performance, it can also be based on the actual performance requirements and The corresponding clock frequency to add or reduce the number of kernel modules, To achieve the corresponding performance targets. Achievement and results

4. Achievement and results

This scheme uses Verilog language to implement theSM3 algorithm implementation method, using Vivado 15.4 software, modelsemSimulation software, and Xilinx's K7 series FPGA corePiece. To verify what is described above.

Without running water, for continuous 512bit input Enter, only after 64 rounds of calculations have been completed can the next one be carried out 512Bit 64 rounds of calculations. After 448(64 * 7) clkIn order to get the results of each 64 rounds of calculations. In the case of the same clock, using a pipeline structure for Continuous input can greatly reduce its delay.Comparison of single algorithm kernel and multiple algorithm cores of pipeline structure. Conduct simulation analysis. Unifies the working clocks of all modules to clk. From the simulation results, it can be seen that for a single algorithm core, the input link Continued 512bit string, obtained after the initial 448(64 * 7) clk the first 64 round calculation result (and for clk0 $\geq$ 7 clk Calculations error occurred). After that, every seven clk outputs a 64. Wheel calculation results. For multiple algorithm cores(this example uses 7 algorithm cores), after passing through After the first 448 cycles output the first 64 rounds of calculations, each Clock cycle clk can output a set of 64 round calculations, compared to The single algorithm kernel has a significant increase in time delay.

In this paper, we use the 64 wheel calculation grading model and multiple algorithms. The hardware optimization of SM3 algorithm is realized by parallel multiplexing. Through Over-optimization of key calculation paths and the classification design of 64 wheel calculations, large the delay is greatly reduced and the performance is improved. And through multiple algorithms Multiplexing makes performance break through the constraints of the performance of iterative compression algorithms, in real terms. Inter packet testing is also easy to swallow with a low delay of up to 10Inhalation, and further increase the space is very large, compared with single-level SM3 Can increase 50 times.

References

[1] Galvez-Lopez Dorian, Tardos, Juan D. Bags of Binary Words for Fast Place Recognition in Image Sequences [J]. IEEE Conf. Comput. Vis. Pattern Recognit, 2012, 28 (5): 1188 — 1197.

[2] Roberto Arroyo, Alcantarilla Pablo F, Bergasa Luis M, et al. Bidirectionalloop closure detection on panoramas for visual navigation [C]. 2014 Intelligent Vehicles Symposium Proceedings: 1378 — 1383.

[3] Mur-Artal R, Tardos J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras [J]. IEEE Trans. Robot. , 2017, 33 (5): 1255 — 1262.

[4] Singh R P P, Kumar P, Singh B. Performance Analysis of 32-bitArray Multiplier with a Carry Save Adder and with a Carry-look-ahead Adder[J]. International Journal of Recent Trends in Engineering, 2009, 2 (6): 83 — 86.

[5] Tim Caselitz, Bastian Steder, Michael Ruhnke, et al. Matching Geometry for Long-term Monocular Camera Localization [C]. Workshop:AI for Long-term Autonomy, IEEE International Conference of Robotics and Automation (ICRA), Stockholm, Sweden, 2016.

[6] Thomas Whelan, Michael Kaess, John J, et al. Deformation-based loop closure for large scale dense RGB-D SLAM [C]. 2013 IEEE/RSJ Int. Conf. Intell. Robot. Syst. , 2013: 548 — 555.

[7] Sivic, Zisserman. Video Google: a text retrieval approach to object matching in videos [J].Proceedings Ninth IEEE International Conference on Computer Vision, 2003, 2:1470 — 1477.