Implementations for Dynamic SHA

Zijie Xu
E-mail: xuzijiewz@gmail.com

Abstract. This paper specifies the implementations of Dynamic SHA algorithms.

Key words: SHA, Dynamic SHA

1 Introduction

This paper specifies the implementations of Dynamic SHA algorithms. The implementations include ANSI C implementation and hardware implementation.

2. Implementation

2.1 Platforms

Dynamic SHA has been implemented at follow platform :

1. Intel 80/87c58.
2. Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 32-bit (x86) Edition.
3. Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 64-bit (x64) Edition.

This ANSI C code file and implementation result is in CD.

2.2 Estimation

2.2.1 8-bit processor

Dynamic SHA has been implemented at the simulation “Keil uVision” , the processor is Intel 80/87c58. The parameter of Intel 80/87c58 is:

MCS-51 CHMOS single-chip 8-bit microcontroller with 32 I/O lines, 3 Timers/Counters, 6 Interrupts/4 priority levels, 32K Bytes On-Chip ROM/EPROM, 256 Bytes on-chip RAM, Programmable Serial Channel with Frame Error Detection, 24 MHz crystal oscillation.

2.2.2 Reference Implementation

The result of reference implementation as table 2.1show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	1.38×10^{-2}	4.79×10^{-1}	5.04×10^{-1}	1.17	9.50	9.30×10
Dynamic SHA-256	1.38×10^{-2}	4.80×10^{-1}	5.05×10^{-1}	1.17	9.50	9.30×10
Dynamic SHA-384	2.66×10^{-2}	1.06	1.09×10^{-1}	1.34	1.06×10	1.04×10^{2}
Dynamic SHA-512	2.66×10^{-2}	1.06	1.09×10^{-1}	1.35	1.06×10	1.04×10^{2}

Table 2.1 The seconds of reference implementation of Dynamic SHA on Intel 80/87c58

The Program Size is as table 2.2 show:

	Data (bytes)	Xdata (bytes)	Code (bytes)	Total (bytes)
Dynamic SHA-224	34.0	10605	6478	17117
Dynamic SHA-256	34.0	10605	6478	17117
Dynamic SHA-384	34.0	11045	8943	20022
Dynamic SHA-512	34.0	11045	8943	20022

Table 2.2 The Program Size of reference implementation of Dynamic SHA on Intel 80/87c58

2.2.3 Optimized Implementation

The result of optimized implementation as table 2.3show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	1.38×10^{-2}	4.16×10^{-1}	4.41×10^{-1}	1.04	8.49	8.30×10
Dynamic SHA-256	1.38×10^{-2}	4.17×10^{-1}	4.42×10^{-1}	1.04	8.49	8.31×10
Dynamic SHA-384	2.66×10^{-2}	0.962×10^{-1}	0.98×10^{-1}	1.24	9.73	9.54×10
Dynamic SHA-512	2.66×10^{-2}	0.956×10^{-1}	0.982×10^{-1}	1.24	9.72	9.54×10

Table 2.3 The seconds of optimized implementation of Dynamic SHA on Intel 80/87c58

The Program Size is as table 2.4 show:

	Data (bytes)	Xdata (bytes)	Code (bytes)	Total (bytes)
Dynamic SHA-224	34.0	10609	13724	24367
Dynamic SHA-256	34.0	10609	13724	24367
Dynamic SHA-384	34.0	11051	40650	51735
Dynamic SHA-512	34.0	11051	40650	51735

Table 2.4 The Program Size of optimized implementation of Dynamic SHA on Intel 80/87c58

From table 2.1, 2.2, 2.3 and 2.4. it is known that the program size of optimized implementation is about 1.42(resp. 2.58) times of the program size of reference implementation. And optimized implementation has a higher speed. The contrast as table 2.5 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	1	86.8%	87.5%	88.9%	89.4%	89.2%
Dynamic SHA-256	1	86.9%	87.5%	88.9%	89.4%	89.4%
Dynamic SHA-384	1	90.8%	89.9%	92.5%	91.7%	91.7%
Dynamic SHA-512	1	90.2%	90.1%	91.9%	91.7%	91.7%

Table 2.5 The contrast of speed optimized implementation and reference implementation

In table 2.5, it is known that optimized implementation is speedyer about 10% than reference implementation.

2.3 Windows Vista Ultimate 32-bit (x86) Edition.

The platform of this implementation is :Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 32-bit (x86) Edition.

Compiler: The ANSI C compiler in the Microsoft Visual Studio 2005 Professional Edition.

2.3.1 Reference Implementation

The result of reference implementation as table 2.6 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	104	1720	1797	3954	31056	302788
Dynamic SHA-256	100	1730	1833	4003	31224	303001
Dynamic SHA-384	133	5148	4995	5860	52064	520520
Dynamic SHA-512	130	5053	4966	5902	52113	522434

Table 2.6 The numbers of processor clock cycles on Windows Vista Ultimate 32-bit (x86) Edition

The memory requirement is as table 2.7 show:

	File size (bytes)	Message words (bytes)	work variables (bytes)	temporary words (bytes)
Dynamic SHA-224	69632	64	32	4
Dynamic SHA-256	69632	64	32	4
Dynamic SHA-384	73728	128	64	8
Dynamic SHA-512	73728	128	64	8

Table 2.7 The memory requirement of reference implementation of Dynamic SHA on Windows Vista Ultimate 32-bit (x86) Edition

2.3.2 Optimized Implementation

The result of optimized implementation as table 2.8 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	104	1596	1665	3732	28685	279994
Dynamic SHA-256	100	1623	1697	3769	28736	279297
Dynamic SHA-384	135	4295	4543	5284	45611	474191
Dynamic SHA-512	131	4368	4405	5367	44938	472848

Table 2.8 The numbers of processor clock cycles on Windows Vista Ultimate 32-bit (x86) Edition

The memory requirement is as table 2.9 show:

	File size (bytes)	Message words (bytes)	work variables (bytes)	temporary words (bytes)
Dynamic SHA-224	69632	64	32	4
Dynamic SHA-256	69632	64	32	4
Dynamic SHA-384	73728	128	64	8
Dynamic SHA-512	73728	128	64	8

Table 2.9 The memory requirement of optimized implementation of Dynamic SHA on Windows Vista Ultimate 32-bit (x86) Edition

From table 2.6, 2.7, 2.8 and 2.9. it is known that the file size of optimized implementation is same as the file size of reference implementation. And optimized implementation has a higher speed. The contrast as table 2.10 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	1	92.8%	92.7%	94.4%	92.4%	92.5%
Dynamic SHA-256	1	93.8%	92.6%	94.2%	92.0%	92.2%
Dynamic SHA-384	1	83.4%	91.0%	90.2%	87.6%	91.1%
Dynamic SHA-512	1	86.4%	88.7%	90.9%	86.2%	90.5%

Table 2.10 The contrast of speed optimized implementation and reference implementation

In table 2.10, it is known that optimized implementation is speedyer about 10% than reference implementation.

2.4 Windows Vista Ultimate 64-bit (x64) Edition.

The platform of this implementation is :Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 64-bit (x64) Edition.

Compiler: The ANSI C compiler in the Microsoft Visual Studio 2005 Professional Edition.

2.4.1 Reference Implementation

The result of reference implementation as table 2.11 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	103	1718	1792	3952	31011	302305
Dynamic SHA-256	100	1729	1832	3997	31043	300940
Dynamic SHA-384	128	5149	4969	5832	52016	520110
Dynamic SHA-512	130	5069	4949	5939	52029	522007

Table 2.11 The numbers of processor clock cycles on Windows Vista Ultimate 64-bit (x64) Edition

The memory requirement is as table 2.12 show:

	File size (bytes)	Message words (bytes)	work variables (bytes)	temporary words (bytes)
Dynamic SHA-224	69632	64	32	4
Dynamic SHA-256	69632	64	32	4
Dynamic SHA-384	73728	128	64	8
Dynamic SHA-512	73728	128	64	8

Table 2.12 The memory requirement of reference implementation of Dynamic SHA on Windows Vista Ultimate 64-bit (x64) Edition

2.4.2 Optimized Implementation

The result of optimized implementation as table 2.13 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	101	1597	1675	3773	28693	279357
Dynamic SHA-256	100	1602	1688	3763	28649	279100
Dynamic SHA-384	129	4278	4516	5225	45492	474331
Dynamic SHA-512	129	4375	4402	5289	44830	472010

Table 2.13 The numbers of processor clock cycles on Windows Vista Ultimate 64-bit (x64) Edition

The memory requirement is as table 2.14 show:

	File size (bytes)	Message words (bytes)	work variables (bytes)	temporary words (bytes)
Dynamic SHA-224	69632	64	32	4
Dynamic SHA-256	69632	64	32	4
Dynamic SHA-384	73728	128	64	8
Dynamic SHA-512	73728	128	64	8

Table 2.14 The memory requirement of optimized implementation of Dynamic SHA on Windows Vista Ultimate 64-bit (x64) Edition

From table 2.6, 2.7, 2.8 and 2.9. it is known that the file size of optimized implementation is same as the file size of reference implementation. And optimized implementation has a higher speed. The contrast as table 2.15 show:

	Run time for set up	Bytes processed				
		1	10	100	1000	10000
Dynamic SHA-224	1	93.0%	93.5%	95.5%	92.5%	92.4%
Dynamic SHA-256	1	92.3%	92.1%	94.1%	92.3%	92.7%
Dynamic SHA-384	1	83.1%	90.9%	89.6%	87.5%	91.2%
Dynamic SHA-512	1	86.3%	88.9%	89.1%	86.2%	90.4%

Table 2.15 The contrast of speed optimized implementation and reference implementation

In table 2.15, it is known that optimized implementation is speedyer about 10% than reference implementation.

3 Hardware Implementation

The top-level architecture for Dynamic SHA implementation is as Figure 1:

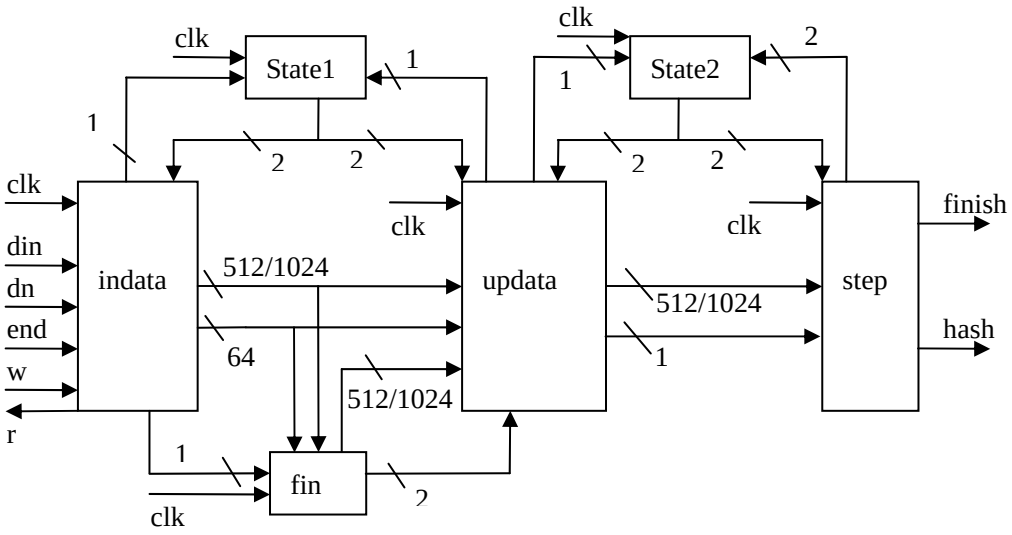

Figure 1: Atop-level block diagram of Dynamic-SHA

Implementation results of Dynamic-SHA are presented in Table 2.1.

	SHA-224/256	SHA-384/512
Equivalent gate count	40,891	85,066
Slices	2,332	5,671

Table 2.1: Implementation results of Dynamic-SHA

4. Conclusions

Form the implementation results of Dynamic-SHA2, it is known that the optimized implementation need more memory and are speedlyer than reference implementation at different platform.

And form the implementation results of Dynamic-SHA, it is known that Dynamic-SHA can be used in low power, constrained memory environments, such as: 8-bit processors (e.g., smartcards), voice applications, satellite applications.