Speed Improvement of AES Encryption using hardware accelerators synthesized by C Compatible Architecture Prototyper(CCAP)

Hiroyuki KANBARA* Takayuki NAKATANI† Naoto UMEHARA†
Nagisa ISHIURA‡ Hiroyuki TOMIYAMA§

* ASTEM RI, Shimogyo, Kyoto, 600-8813, JAPAN
† Graduate School of Science&Engineering, Ritsumeikan University, Kusatsu, 525-8577, JAPAN
‡ School of Science&Technology, Kwansei Gakuin University, Sanda, 669-1337, JAPAN
§ Graduate School of Information Science, Nagoya University, Chigusa, Nagoya, 464-8603, JAPAN

hls@ksc.kwansei.ac.jp

Abstract— The authors are developping a high-level synthesizer called C Compatible Architecture Prototyper(CCAP). CCAP compiles ANSI C program which is a part of embedded software and generates an application specific hardware accelerator in HDL. Synthesized accelerator has an ability to read/write main memory and executes calculation faster than an embedded processor. CCAP offers an arbiter circuit which makes it possible for the synthesized accelerator and a processor to access main memory in parallel. In this paper we report the speed improvement of AES Encryption using design methodology of CCAP synthesizer.

I. Introduction

Recently handheld devices like mobile phones or personal digital assistant(PDA)s need powerful computing power for capturing movie or internet browsing. Sensor network devices with signal-processing and communication capability is one of the future trends of ubiquitous computing. Requirements to these embedded systems like handheld devices or sensor network devices are increasingly 'high-volume' and 'low-cost'.

A common solution for high-volume and low-cost embedded systems production is the sytem on a chip(SoC), which includes an embedded processor, a memory and I/O peripherals. Usually clock rates of the embedded processors are one-tenth of processors for personal computer(PC)s. Because an energy source for the handheld devices or the sensor network devices is commonly a battery and power consumption of the embedded processor must be under several watts. Performance requirements to the embedded systems have become much higher, sometimes almost the same computation ability compared with PCs. Hardware accerelation is one of the techniques to improve perfomance of the embedded systems.

†Takayuki NAKATANI is currently with SHIMADZU Corporation and Naoto UMEHARA is currently with KAWASAKI MICROELECTRONICS, Inc.

Hardware accelerators are designed for computationary intensive software code. For example, Gaussian or Sobel filter functions are used for signal processing. Block cipher encryption/decryption is used for secure communication on internet. These functions can be five-to-ten times faster when its calculation is accelerated by an application specific hardware. Hardware accerelation is very useful approach but design efficiency of the hardware accerelator has not been improved. Because design steps for the hardware accrelerator as follows are difficult and time-consuming.

- design hardware accelerator using Hardware Descrption Language
- develop device driver program for invoking the hardware accelerator from software
- verification of the hardware accelerator including the driver program

The authors are developping a high-level synthesizer called C Compatible Architecture Prototyper (CCAP) [1][2] . CCAP compiles ANSI C program which is a part of embedded software and generates an application specific hardware accelerator in HDL. Synthesized accelerator executes faster than a cpu and accesses to main memory.

CCAP offers an arbiter circuit which makes it possible for the synthesized accelerator and a cpu to access main memory in parallel. These features of CCAP synthesizer make it possible for a software designer to get much faster calculation result by the synthesized hardware accelerator, without hardware designer's assistance.

In this paper we report the speed improvement of AES Encryption using CCAP synthesizer.

AES encryption mainly consists of 5 stages : KeyExpansion, AddRoundKey, SubBytes, ShiftRows and MixColumns. AES encryption software is written in ANSI C and above mentioned 5 stages are implemented in separate 5 functions. Hardware accelerators for 3 functions : SubBytes, ShiftRows and MixColumns are designed by human designer from ANSI C functions.

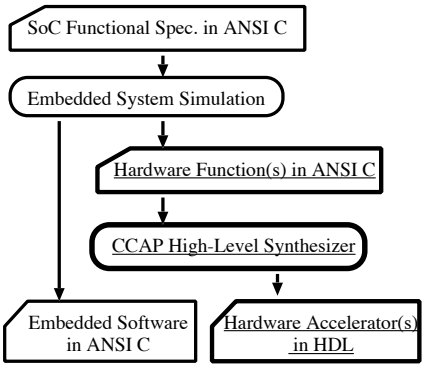

Fig. 1. Embedded System Design Flow using CCAP Synthesizer

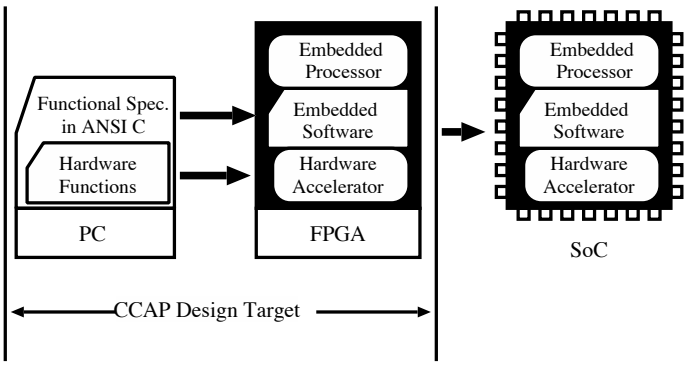

Fig. 2. Design Target of CCAP Synthesizer

These 3 hardware accelerators speed up the AES encryption almost 5.0 times faster, compared with software execution by an embedded processor which is Mips R3000 compatible. Circuit size of the 3 accelerators with the arbitor circuit is only 59% of the embedded processor. This design result shows that solution using CCAP hadware synthesizer is enough efficient compared with multi-processors solution.

II. High-Level Synthesizer CCAP

Embedded system design flow using high-level synthesizer CCAP is show in Fig. 1. In this flow, bottleneck functions of 'SoC Functional Spec. in ANSI C' are found in the 'Embedded System Simulation' step and 'CCAP High-Level Synthesizer' generates 'Hardware Accelerator(s) in HDL' from the bottleneck function(s) called 'Hardware Function(s) in ANSI C'.

Embedded system designer describes all requirements to a SoC in ANSI C. ANSI C is a commonly used programming language for embedded software. The described ANSI C program can be compiled by ANSI C compiler and executed on PCs. Using embedded sytem simulator, execution speed of the software by an embedded software, or speed improvement by hardware accereletors synthesized by CCAP is estimated.

The system designer finds out bottleneck function(s) and generates hardware accelerator(s) using CCAP. We call these ANSI C functions for hardware synthesis 'hardware functions'.

CCAP synthesizes hardware accelerators in HDL from the hardware functions in ANSI C. Driver software for each synthesized accelerator is generated by CCAP. The system designer compiles the orignal ANSI C program and the generated driver software by a cross compiler for an embedded processor.

Fig. 2 shows our prototyping scheme that FPGA will be used for verification and performance estimation. The synthesized hardware accelerators and the embedded processor will be implemented within the FPGA logic. The compiled ANSI C program will be executed by the FPGA prototype embedded system and the system designer will find out that

- the embedded processor and the accelerators give same functions done by the software execution
- the embedded processor and the accelerators offer enough performance

Following these design steps, the system designer co-designs hardware accelerators in HDL and embedded software in ANSI C. The system designer completes prototyping of the embedded system without help of hardware designers. To realize this hardware/software co-design methodology, CCAP high-level synthesizer has features as follows.

- CCAP handles ANSI C software as it is. The system designer does not have to change ANSI C hardware functions for synthesizing hardware accelerators.
- CCAP compiles 'pointer' and 'external variables' of ANSI C. This makes it possible for synthesized hardware accelerators and an embedded processor to share same memory address space.

III. AES Encryption Argorithm

Advanced Encryption Standard(AES) is a block cipher adopted as an encryption standard by the U.S. government. [3][4]. Fig. 3 shows procedure of the AES encryption. The AES has a fixed block size input(IN) of 128bits(16bytes) and a key(KEY) size of 128, 192, 256bits. Rounds of 'loop' change depending on the KEY size, 9 rounds for 128bits, 11 rounds for 192bits and 13 rounds for 256bits.

In the KeyExpansion step, the round key for the AddRoundKey step is generated from the original key through a process called 'Key Scheduling'. The KeyExpansion step is usually done once because the same round key will be used for encryption of one input stream. In the SubBytes step, each input byte of 4 !_4 array of bytes is replaced by another byte, from look-up tabel called S-Box. This 4 !_4 array of bytes for the AES encryption is callled the State matrix. In the ShiftRows step, each byte of the State matrix is shifted cyclically certain number of steps. In the

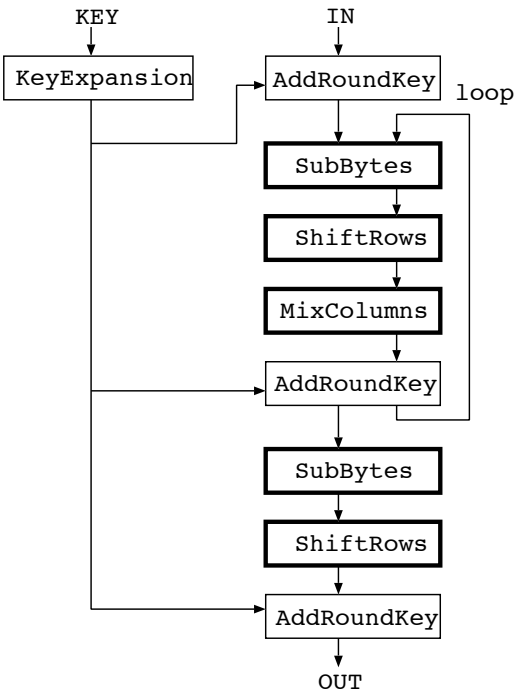

Fig. 3. AES Encryption Procidure

MixColumns step, each column of the State matrix is multiplied with a fixed polynomial. In the AddRoundKey step, each byte of the state is combined with a byte of the round key using the XOR operations.

The following is speed improvement of the AES Encryption based on CCAP synthesizer design methodology.

IV. Speed Improvement of AES Encryption

Five different configurations as follow are used to estimate speed improvement of the AES encryption.

- All steps in the AES encryption are software(executed by an embedded processor)
- 'ShiftRows' step is executed by hardware accelerators and other steps are software(executed by an embedded processor)
- 'SubBytes' step is executed by hardware accelerators and other steps are software(executed by an embedded processor)
- 'MixColumns' step is executed by hardware accelerators and other steps are software(executed by an embedded processor)
- 'ShiftRows"SubBytes' and 'MixColumns' steps are executed by hardware accelerators and other steps are software(executed by an embedded processor)

For each configuration, we estimated circuit size and clock cycles taken for 128 bits input stream to be encrypted by 128bits length user key.

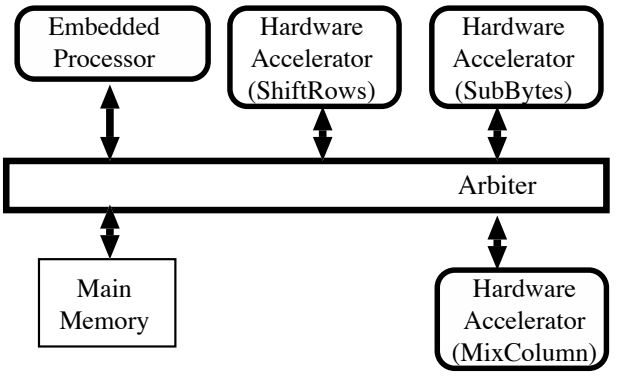

Fig. 4. Hardware Configuration for AES Speed Improvement

The KeyExpansion step remains software for every configuration, assuming that the KeyExpansion step does not become bottleneck. Becuase the KeyExpansion step is done only once and other 'ShiftRows', 'SubBytes' and 'MixColumns' steps are repeated.

RT-level simulation using ModelSim-XE Verilog Simulator is used to count clock cycles for encrypting 128bits length input stream by 128bits length user key. MIPS R3000 compatible processor in RTL level description is prepared for the embedded processor. This embedded processor does not include FPU, MMU and cache memory for instructions and data that original R3000 processor has. AES encryption software is compiled by GNU C Compiler for R3000.

Hardware architecture for an embedded processor and hardware accelerators is shown in Fig. 4.

The embedded processor, the arbiter circuit and the hardware acclerators for 'ShiftRows','SubBytes' and 'MixColumns' are synthesized for XILINX Vertex series FPGA by ISE Foundation tools. Circuit size of each configuration is shown in 'slices' to be used for configuring Vertex series FPGA. 'Main memory' in Fig. 4 is not included for estimating the circuit size. 'Arbiter' in Fig. 4 is not a target of high-level synthesis by CCAP. We prepared the arbiter circuit in Verilog HDL which allows 3 hardware acclerators and an embedded processor to read/write the main memory in parallel.

Table I shows circuit size and maximum delay time for each hardware module. The maximum delay time of the embedded R3000-compatible processor is longer than any other ones of hardware modules. This result means that adding hardware accleretors does not affect clock rates of the embedded processor. The sum of the circuit size of the arbitor, hardware accleretors for ShiftRows, hardware accleretors for MixColumn and hardware accleretors for ShiftRows is 59% of the size of the processor.

In Table II clock cycles for the AES encryption and circuit size of each configuration are shown. Hardware acceleration of the SubBytes step is the most effective for speed improvement. The hardware accelerelator for SubBytes has 4 lookup tables for the S-Box and circuit size of the SubBytes acclerelator becomes bigger than any other accelera-

tor This 4 lookup tables makes it possible to replace input byte to another byte in parallel in the SubBytes step.

Taking increase in circuit size, hardware acceleration for MixColumns step is the most efficient. In the MixColumns step, each input byte of the 4 !_4 State matrix is replaced by other byte. When the MixColumns step is software implementation, this replacement is done sequentially by the embedded software. In the hardware acclerelator for MixColumns, only hard-wired logic is needed to execute this replacement in parallel.

When 3 steps of ShiftRows, SubBytes, MixColumns are implemented as hardware accelerators, execution speed of the AES encryption becomes almost 5.0 times faster than software execution. With this configuration, circuit size including the embedded processor and the arbitor is 1.58 times bigger than the embedded processor. System design approach with CCAP synthesizer is more efficient than multi-processor solution.

V. Conclusion

We have presented an embedded system design approach using high-level synthesizer CCAP. AES encryption embedded software is accelerated by converting bottleneck software functions into hardwares based on the proposed design approach. We confirms that CCAP high-level synthesizer approach is efficient to synthesize hardware from embedded software.

Acknowledgement

We would like to thank Prof. Yamazki at Ritsumeikan University for making AES encryption software open. We would also like to thank Mr. Sugihara of Kyoto University, Mr. Nishimura and the memebers of Ishiura Laboratory of Kwansei Gakuin University for their discussion and suggestions. This work is in part supported by KAKENHI 19700040.

References

[1] M.Nishimura, K.Nishiguchi, N.Ishiura, H.Kanbara, H.Tomiyama, Y.Takatsukasa, M.Kotani,
"High-Lebel Synthesis of Variable Accesses and Function Calls in Software Compatible Hardware Synthsizer CCAP",
in Proc. Workshop on Synthesis And System Integration of Mixed Information Technologies(SASIMI) 2006, pp.29-34, 2006

[2] M.Nishimura, N.Ishiura, Y.Ishimori, H.Kanbara, H.Tomiyama,
"Calling Software Functions from Hardware Functions",
in Proc. Workshop on Synthesis And System Integration of Mixed Information Technologies(SASIMI) 2007, 2007

[3] Daemen,J. and Rijmen, "AES Proposal : Rijndael",
http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf

[4] National Institute of Standards and Technology(NAIST), "Advanced Encryption Standard(AES)", FIPS Publication 197, 2001,
http://csrc.nist.gov/encryption/aes/index.html

TABLE I
CIRCUIT SIZE AND MAXIMUM DELAY OF EMBEDDED PROCESSOR, ARBITER, AND HARDWARE ACCLERELATORS

Hardware Module	Circuit Size (Slices)	Maximum Delay (ns)
Embedded Processor	2825	12.4
Arbiter	528	5.71
Hardware Acclerelator (ShiftRows)	210	1.95
Hardware Acclerelator (SubBytes)	470	1.97
Hardware Acclerelator (MixColumn)	367	2.01

TABLE II
SPEED IMPROVEMENT OF AES ENCRYPTION AND INCREASE IN CIRCUIT SIZE

Configuration	Execution Speed		Circuit Size	
	Clock Cycles	Improvement Ratio	Slices	Increase Ratio
Embedded Processor(AES Software)	26794	1.00	2825	1.00
Embedded Processor and Arbitor	-	-	3353	1.18
Embedded Processor, Arbitor and Hardware Accleretor(ShiftRows)	24964	1.07	3563	1.26
Embedded Processor, Arbitor and Hardware Accleretor(SubBytes)	21964	1.21	3823	1.35
Embedded Processor, Arbitor and Hardware Accleretor(MixColumns)	12047	2.22	3720	1.32
Embedded Processor, Arbitor Hardware Accleretor(ShiftRows) Hardware Accleretor(SubBytes) and Hardware Accleretor(MixColumns)	5387	4.97	4400	1.58