VLSI Implementation With Double Cipher and Media Processing for Ad-Hoc Network

Masa-aki Fukase[1], Non-member

ABSTRACT

Ubiquitous network is really one of remarkable trends of next generation information and communication technologies. However, the rapid increase of ubiquitous technologies has given rise to serious concerns about security issues which are getting worse, especially in the case of ad-hoc network because it is resource constrained. Both transience and secureness are characteristic features of ad-hoc network. In order to provide a sophisticated processor accompanied with such characteristics, we have so far developed HCgorilla. This is a ubiquitous processor that unifies basic aspects of PC processors, mobile processors, Java CPUs, cryptography processors, etc. The unification of these processors has shown effectiveness to achieve not only power and chip area saving but also secureness for ubiquitous environment. The secureness provided by HCgorilla is temporal with practically enough cipher strength and without relying on permanent network infrastructure. This is really one of qualitative conditions required for ad-hoc networks. Considering the ever growing ubiquitous network, the target of this study is the total optimum design of HCgorilla chips with particular emphasis on small resource and secure implementation. The improved version, HCgorilla.7 is implemented in a CMOS standard cell chip. Specific features of HCgorilla.7 are described regarding occupied area, power consumption, throughput, and cipher strength. Judging from these results, HCgorilla is one of most profitable candidates for the next generation ubiquitous environment.

Keywords: Ubiquitous Network, Ad-Hoc Network, Ubiquitous Security, Double Cipher, Media Processing, Processor Chip

1. INTRODUCTION

Ad-hoc network is really an emerging technology for the next generation ubiquitous computing. It is contributive to the development of ubiquitous environment in view of cost performance, simplicity, functionality, usability, etc. However, ubiquitous community has left fundamental issues unsolved. Some of them are digital divide, information flood, etc. For these issues, software approach such as dependable computing is usually taken. But it is really time and power consuming in computing huge amount of multimedia data. Actually, massive data is used for the expression of multimedia information. This is crucial for the interaction between ubiquitous devices and human being. The massive quantity of multimedia information is very difficult for regular techniques like embedded software to satisfy various demands for not only security but also usability, speed, and power consciousness.

More fundamental issue is notorious security threat due to the expansion or diversity of ubiquitous platforms. Although the diversity is inevitable for the usability and functionality of ubiquitous network, it also causes notorious security issues like insecurity, security threat or illegal attack such as tapping, intrusion, pretension. The worldwide diversity vs. security threat is the two-faced characteristics of ubiquitous network. In order to keep the security of such a transient network, a practical solution is not to develop an extremely strong cipher scheme, but to explore a temporary security with practically enough cipher strength and without relying on permanent network infrastructure. Not only transience but also secureness is one of most characteristic features of ad-hoc network. The proliferation and standardization of ubiquitous technologies have given rise to serious concerns about security and privacy issues which are exacerbated by the fact that the majority of these technologies are resource constrained in terms of area and energy budgets. This has led to increased interest in efficient implementations of cryptographic primitives.

An overall strategy for contribution to ad-hoc network will be power conscious hardware approach that develops a sophisticated single VLSI chip processor, which is able to treat multimedia date with practical security over ubiquitous network. It is really worthwhile to exploit the hardware integration of possible key technologies. We have unified the role of PC processors, mobile processors, Java CPUs, cryptography processors, etc. into a ubiquitous processor named HCgorilla [1].

An ad-hoc cipher implemented in HCgorilla is a double cipher scheme that is a microarchitecture-based, software-transparent hardware cipher [2]. The double cipher scheme offers the security of the whole data with negligible hardware cost and moderate per-

Manuscript received on July 15, 2012 ; revised on November 28, 2012.

[1] The author is with Graduate School of Science and Technology, Hirosaki University, Hirosaki 036-8561, Japan, E-mail:slfuka@eit.hirosaki-u.ac.jp

formance overhead by combining two cipher algorithms. The one is RAC (random addressing cryptography). It is transposition cipher devised from the direct connection of a built-in RNG (random number generator), register file, and a data cache. RAC requires no additional chip area and power dissipation. A random store based on the direct connection scrambles or transposes a series of multimedia data at random without any special encryption operation. The other is a data sealing algorithm implemented during the data transfer from register file to data cache. This complements RAC's shortcoming and enhances the security of data information as a whole.

The study described in this article focuses on the further improvement of HCgorilla chips with particular emphasis on low resource and secure implementations. The total improvement of HCgorilla requires complicated clock schemes. The improved version, HCgorilla.7 is implemented in a CMOS standard cell chip. Specific features of HCgorilla.7 are described regarding occupied area, power consumption, throughput, and cipher strength. Judging from these results, HCgorilla is one of most profitable candidates for the next generation ubiquitous environment.

2. ARCHITECTURE

The majority of recent years ubiquitous technologies are resource constrained in terms of chip area and battery energy available. For example, LCD (liquid crystal display) and processors consume similar power in running mobile devices. While LCD is turned on only when it displays some information, processors are always in the standby state to receive calling. Thus, the restriction for the power saving of mobile devices is inevitably imposed on processors. Strategy in developing HCgorilla is also to achieve power consciousness for Green IT and secureness for ubiquitous environment.

Fig. 1 shows the basic architecture of HCgorilla.7 developed in this study. Power conscious resource-constrained implementation is achieved by the design steps in the following.

a. Parallelism:
- · The parallelism at architecture level takes multicore and multiple pipeline structure. Following HW/SW codesign approach, two symmetric cores run multiple threads in parallel.
- · ILP (instruction level parallelism) is carried out in each core. It has two arithmetic media pipes and a cipher pipe.

b. Register file as the shared memory of the double core. The register file plays the role of a streaming buffer.

c. LFSR (linear feedback shift register) is used as RNG built in the cipher pipe due to restricted hardware quantity. It achieves longer cycle with negligible additional area. Random number generation is one of potential topics.

HCgorilla achieves secureness by using the cipher pipe built in two LFSRs. The cipher pipe does double encryption during the transfer of an image data from the register file to data cache. While an LFSR controls the transposition cipher, RAC, another LFSR controls a substitution cipher or data sealing implemented by the hidable unit, HIDU (hidable data unit). The double cipher executes cipher streaming by SIMD (single instruction stream multiple data stream) mode cipher and decipher codes. They do not attach operands, but repeat instances to transfer byte structured data from a register file to a data cache.

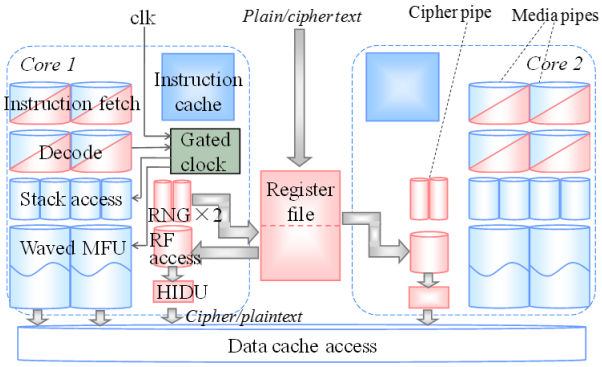

Fig.1: *Architecture of HCgorilla.7.*

HCgorilla is applicable to the three major fields of SoC (system-on-chip) in the following.

a. Communication: The double core covers bidirectional communication.

b. Multimedia:
- · HCgorilla's media pipe is Java compatible fostering a ubiquitous community of participation and transparency.
- · The media pipe includes two arithmetic pipes. Then, the arithmetic pipe has a double stack and a two-waved MFU. Since the parallel degree of the media pipe is two, each core is able to execute four stack-related instructions in parallel.

c. Control: The waved MFU executes floating point arithmetic instructions. This is effective for precise NC (numerical control) machining, GPS (global positioning system) navigation, etc.

2.1 Structure Level Power Saving

Processor systems have saved power by the control of supply voltage and clock. The supply voltage is sometimes scaled down [4] and sometimes gated [5]. Then, the clock is also scaled [6] and gated [7]. More sophisticated clock systems are clocks variable [8], cycle variable or adaptive clock [9]. However, these are accompanied with tradeoff against throughput. This in turn causes considerable overhead.

More effective strategy for power saving should be formulated at a lower level. Wave-pipelining is a

promising candidate for structure level power saving [3]. It has been so far applied to arithmetic logics, circuit blocks, pipeline stages, etc. The wave-pipelining uses the delay of mainly combinational elements instead of using intermediate registers, while conventional pipelining uses registers to divide the circuit into shorter paths. Therefore, the wave-pipelining is more effective in achieving both power saving and speed up without throughput degradation (Table 1).

Table 1: *Sophisticated Clocking Schemes.*

Techniques	Power saving	Speed up	Throughput enhancement	Usage
Wave-pipelining	Effective	Effective	Effective	Circuit block
Clocks variable	Effective	Ineffective	Ineffective	Local clock
Cycle variable	Effective	Ineffective	Ineffective	Local clock
Asynchronous	Effective	Ineffective		Global clock

We apply wave-pipelining to the execution stage of HCgorilla as shown in Fig. 1. The execution stage is a waved MFU that is the combination of wave-pipelining and multifunctionalization of the media pipe's arithmetic functional units. The waved MFU is conceived as follows. A possible way to release instruction scheduling in running processors is to merge the parallel structure of regular pipelines and to make them completely multifunctional. This surely executes every function with the same latency. However, the increase of circuit scale accompanied with the multifunctionalization elongates the critical path. This results in the degradation of clock speed. Thus, the simply merging of regular pipelines does not always promise the total enhancement of processor performance. In order to completely unify hardware units without deteriorating clock speed, wave-pipelining is really promising.

In view of switching probability, the stack access stage and the waved MFU stages are the target of gated clocking as shown in Fig. 1. Gated clocking is a cell-based approach for power saving at microarchitecture level. It stops the clocking of such circuit blocks with low activity that waste switching power. Since leakage power is extremely small in the case of the 0.18 m CMOS standard cell process used in this study, the gated clock is very effective for power saving.

In addition, scan logic for DFT (design for testability) is introduced. Since the scan logic is applied to pipeline registers, a clock scheme to merge gated clocking and scan logic has been developed in this study. Although gated clock and scan path have been supported by regular CAD tools, the wave-pipelining has been mainly done by manual tuning. The clock scheme is implemented in the context as follows. The pipeline registers are made to play the role of a shift register. The role is determined by a scan control signal. The serial mode of the shift register is used in the validation of the HCgorilla chip. On the other hand, a clocking gate unit is placed after an instruction decode stage produces a gated clock.

2.2 Internal Behavior

Fig. 2 illustrates the internal behavior of HCgorilla, focusing on the core 1 for the sake of simple representation. The media data cache and the cipher data cache are logical divisions of the data cache shown in Fig. 1. A test program is composed of media processing and the double cipher. The media processing is divided into four threads and assigned into four arithmetic media pipes in order to fully make use of the parallelism of the waved MFU.

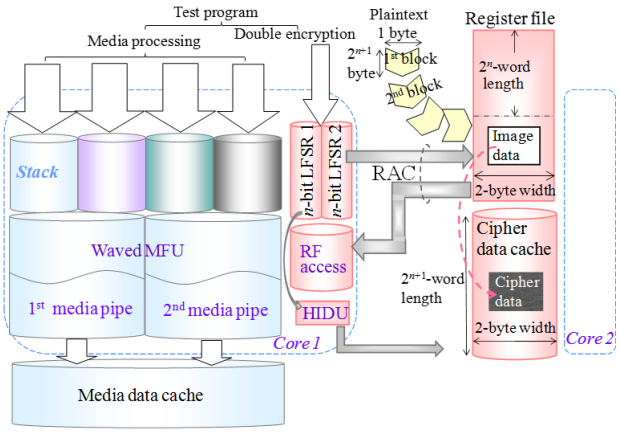

Fig.2: *Internal Behavior of HCgorilla.*

On the other hand, the cipher pipe does double encryption during the transfer of a plaintext like image data from the register file to data cache. Register file and data cache is word structured. Each word is 2-byte width. Therefore, a pixel occupies one and a half word. Register file stores a block divided from the plaintext. Plain and cipher texts are assumed to be divided into blocks as is similar to AES (advanced encryption standard) and DES (data encryption standard). Since the block size influences the performance and the secureness of cryptography, further expanding the register file and data cache is crucial.

The double cipher encryption is implemented in HCgorilla according to the following algorithm [2].
Input: a plaintext
Output: a ciphertext
(*i*) Divide the input into blocks in order.
(*ii*) Store the top block in the register file.
(*iii*) Make the random numbers output from LFSR 1 specify a register file address.
(*iv*) Synchronize a data cache address with the current clock count.
(*v*) Transfer the specified register file's content to the synchronized data cache address.
(*vi*) Make a hidable function work for the block during the transfer. Be the responsibility of the hidable function to perform substitution cipher for

the block by using the random numbers output from LFSR 2.

(*vii*) If the current block is not the last, be the second block top one and go back to (*ii*), else stop.

According to this algorithm, the sequential random addressing store of the plaintext blocks results in the formation of a ciphertext in the data cache. The double cipher is symmetric. The double cipher decryption similarly proceeds by exchanging the input and output.

Dominant reasons why the double cipher algorithm described above is used are as follows.

a. By the adequate setting of the bit width of LFSR, that is, key length, the double cipher algorithm promises temporary security with practically enough cipher strength and without relying on permanent network infrastructure. Such quality of secureness really satisfies conditions required for ad-hoc networks.

b. Since the processing of plain and cipher texts is accompanied with the addressing of the register file and data cache, this algorithm falls into the category of block cipher.

c. Thus, this is applicable to any multimedia data (image, audio, text) that are byte-structured. For example, image data is expressed by PPM (Portable PixMap), JPEG (Joint Photographic Expert Group), BMP (Bit MaP), etc.

Fig. 3 illustrates the microoperation within the core 1 according to the double cipher algorithm described above. Here, $d_1d_2d_3d_4d_5$ exemplifies a plaintext block. The transfer of the block to the register file is not our concern. 30241 is corresponding key or LFSR 1 output. $h(d_2)h(d_5)h(d_3)h(d_1)h(d_4)$ is a cipher text that is the resultant of the double encryption. In the execution of RAC, the plaintext and LFSR 1's output are synchronized according to their sequence. For example, the first data "d_1" and the first random number "3" are synchronized. During the storage to the 3rd location of the data cache, a hidable function h works for the plaintext block. The sequence of random addressing store like this results in the formation of a cipher in the data cache. The sequence of such microoperations is practiced by a simple wired logic, which is effective to keep usability, speed, and power consciousness.

3. VLSI CHIP IMPLEMENTATION

HCgorilla.7 is implemented in a CMOS standard cell chip. The design environment is summarized in Table 2. Fig. 4 focuses on the chip implementation of HCgorilla.7. The floor planning shown in Fig. 4 (a) corresponds to Fig. 1. The actual layout is shown in Fig. 4 (b).

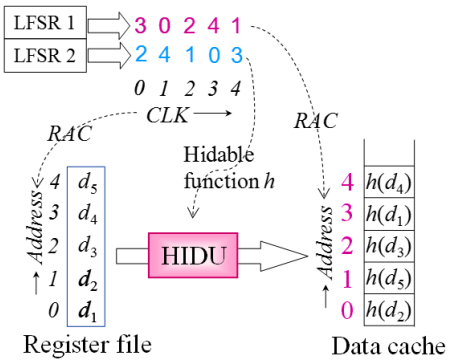

Fig.3: *Microoperation of the Double Cipher.*

Table 2: *Design Environment.*

Software	
OS	Red Hat Linux 4/CentOS 5.4
Synthesis tool	Synopsys - Design Compiler D-2010.03
Simulation tool	Synopsys - VCS version Y-2006.06-SP1
Physical Implementation tool	Synopsys - IC Compiler C-2009.06
Verification tool	Mentor - Calibre v2010.02_13.12
Equivalent verification tool	Synopsys – Formality B-2008.09-SP5
Static Timing analysis tool	Synopsys – Primetime pts,vA-2007.12-SP3

Language	
Synthesis	VHDL
Simulation	Verilog - HDL

Technology
ROHM 0.18-μm CMOS Kyotouniv Standard Cell Library

4. EVALUATION

Fig. 5 gives the overall evaluation of HCgorilla.7. Fig. 5 (a) demonstrates the sharing of occupied area corresponding to Fig. 1. The portions denoted by "Stack access" and "waved MFU" show the sum of four media pipes. The portion denoted by "Cipher pipes" shows the sum of four RNGs, register file, and two HIDUs. The portion of "D cache" is the sum of the media data cache and the cipher data cache in Fig. 2 and Fig. 4 (a). Similarly, Fig. 5 (b) demonstrates the distribution of power dissipation that is derived from static evaluation.

Table 3 summarizes prospective specifications and potential aspects of HCgorilla chips we have so far developed. HCgorilla.7 and HCgorilla.6 have almost the same architecture. Yet, the chip areas of these are different. This is due to whether floor planning is done or not. Since the floor planning takes more area, it contradicts low resource implementation. In addition, the clear layout often takes side channel attacks, tampering, etc. Nevertheless, floor planning is indispensable for local and global clocks separation, effective gated clocking, etc.

Referring to Fig. 2, the evaluation of the cipher pipe and the media pipe is described more in detail. By using the test data of QVGA (quarter video

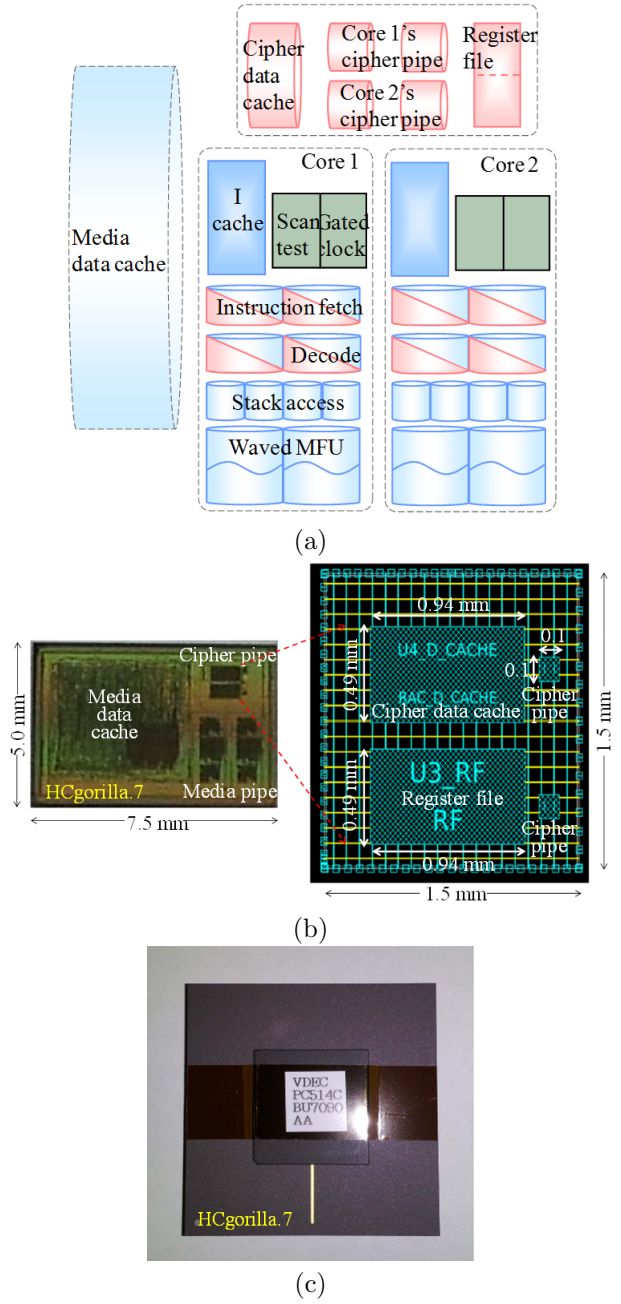

Fig.4: *VLSI Implementation (a) Floor Planning (b) Die Photo (c) Package.*

graphic array) format shown in Fig. 6 (a), Fig. 6 (b) gives the running time of a cipher pipe. This is derived by counting the time taken by the repetition of block transfer, rewriting the register file, and doubling cipher operation. The block access time is assumed to be 370 Mbytes/s, which is a typical value of the memory access speed of cellar phones. The double cipher progresses according to SIMD mode cipher operation. In Fig. 6 (b), "Two RNGs" is the result of HCgorilla.7 and "One RNG" is that of HCgorilla.5. The throughput shown in Fig. 6 (c) is derived from the running time shown in Fig. 6 (b) and the number of double cipher operations.

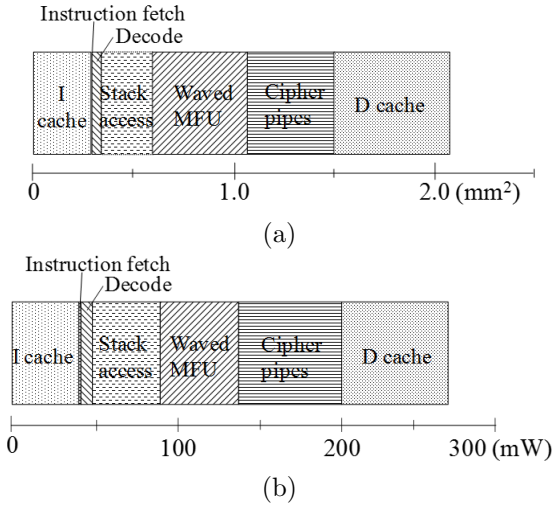

Fig.5: *Overall Evaluation of HCgorilla.7 (a) Occupied Area (b) Power Dissipation.*

Table 3: *Specifications and Aspects of HCgorilla Chips.*

			HCgorilla.3	HCgorilla.5	HCgorilla.6	HCgorilla.7
Design Rule			ROHM 0.18 μm CMOS			
Wiring			1 polySi, 5 metal layers			
Area	Chip		5.0 mm × 7.5 mm		2.5 × 5-mm	5.0 mm × 7.5 mm
	Core		4.28 mm × 6.94 mm			4.28 mm × 6.94 mm
Assembly	Pad	Signal	105	158		
		VDD/VSS	48	32		
	Package		QFP208 (Ceramic)	PGA257		
Power supply			1.8 V (I/O 3.3 V)			
Power consumption			241 mW	274 mW	275 mW	274 mW
Instruction cache			16 bit × 32 word × 2	16 bit × 64 word × 2		
Data cache			16 bit × 128 word	16 bit × 128 word × 2		
Stack memory			16 bit × 8 word × 4	16 bit × 16 word × 8		
Register file			16 bit × 64 word	16 bit × 128 word		
RNG			4 bit × 1	6 bit × 1	6 bit × 2	
No. of . cores			2			
ILP degree			2	4		
Clock frequency			330 MHz	200 MHz		
Throughput	Media pipe			0.17 GIPS		
	Cipher pipe		0.1-0.2 GOPS			
Transfer rate			160-320 Mbps			

Fig. 7 describes the measurement of the cipher strength of HCgorilla.7. Fig. 7 (a) shows the measurement of the double cipher strength that follows so called round robin attack. A result is achieved every round robin attack. j is the number of block. The reason why measurement steps are distinguished in case of $j = 0$ and $j > 1$ is because the same random number sequence is issued for the all blocks from Fig. 2. k is the number of RAC trial attacks. l is the number of HIDU trial attacks. Counting k and l through the experiment, the double cipher strength is derived from the number of nested loops or the time needed in decipher. This evaluates the degree of enduringness or the strength. Actually, the cryptographic strength is the number of attack trials multiplied by the time for decryption. Each of the nested loops guesses a key at random, decrypts ciphertext by using the key, and judges if decipher is successful.

The cipher strength shown in Fig. 7 (b) is achieved

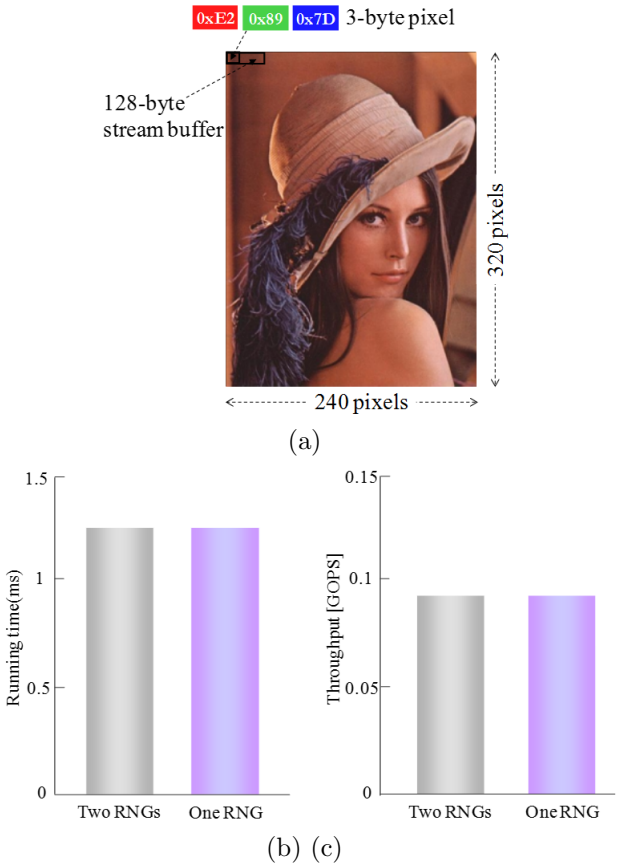

Fig.6: *Performance of a Cipher Pipe (a) Test data (b) Running Time (c) Throughput.*

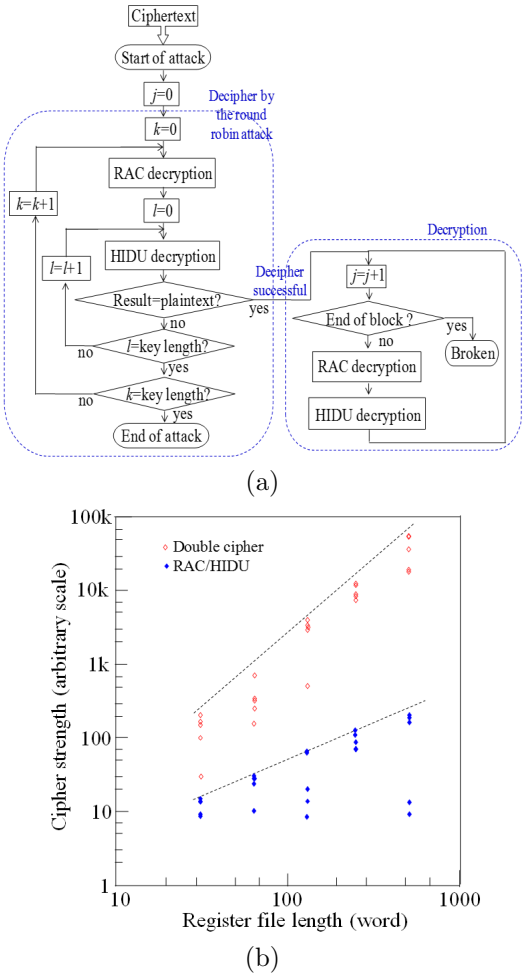

Fig.7: *Cipher Strength (a) Measurement (b) Cipher Strength vs. Register File Length.*

by practicing the method shown in Fig. 7 (a) and by using the test data derived from Fig. 6 (a). Note that the cipher text does not affect the cipher strength from Fig. 2 and Fig. 7 (a). It depends on entirely the block size or the half size of the register file because the blocks after the success of the first attack are simply decrypted by the known key. The abscissa is notched by the full length of the register file and the half size indicates a logical space. Although the HCgorilla.7's register file length is 128 words from Table 3, the register file length is varied from 32 to 512 words to understand the dependency of the cipher strength. Correspondingly, the size of LFSR is varied from 5 to 9 bits. The measurements are done five times for each length. The dotted lines show the upper limit of the cipher strength. The maximum strength of the double cipher is proportional to $2^{\text{LFSR 1 size+LFSR 2 size}}$ from Fig. 7 (a). Similarly, the single cipher reaches at $2^{\text{LFSR 1 size}}$.

Fig. 8 shows the effect of gated clocking on switching power in running a simple integer summation from 1 to 20 on the two media pipes of a core. The effect of gated clocking in the stack access stage is very effective. This is because HCgorilla's media pipe is Java compatible and thus it follows a stack machine. Since the stack machine repeats stack access even in processing a simple calculation, it is really reasonable that the stack machine wastes large switching power in the stack access stage.

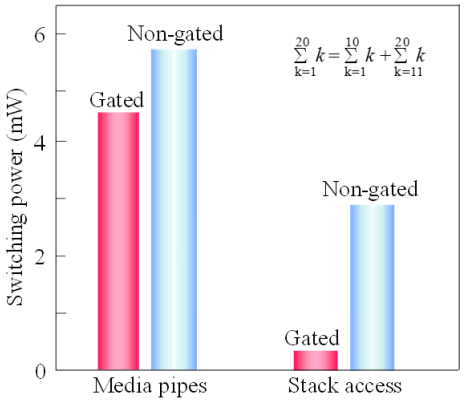

Fig.8: *Power Dissipation of the Media Pipes.*

5. CONCLUSION

Considering the ever growing ubiquitous network, both Java compatible media processing and power conscious temporal cipher have been implemented in the ubiquitous processor HCgorilla. In this article, the total optimum design of HCgorilla has been studied for the next generation ubiquitous environment. The improved version HCgorilla.7 has been implemented in a 0.18-?m CMOS standard cell chip.

The general evaluation of the HCgorilla.7 chip is as follows. Although floor planning takes more area and contradicts low resource implementation, it is indispensable for effective clock schemes. The effect of waved MFU and gated clock for power conscious media processing has been made clear.

As for the secureness provided by the HCgorilla.7 chip, followings have been made clear. The double cipher scheme achieves the drastic enhancement of the cipher strength without the overhead of area, power, and throughput. HCgorilla does not rely on permanent network infrastructure in running the double cipher. Thus, the quality of secureness provided by HCgorilla is probably enough for ad-hoc networks.

The next step of this study is as follows. a. Extension of the total clock scheme to the cipher pipe excepting scan test in view of security. b. Distinction between global clock and local clock. c. Introduction of DLL (delay locked loop) for the local clock [10]. d. Analogue design of buffers for the fine tuning of the waved MFU [11].

6. ACKNOWLEDGMENT

This work is supported by Grant-in-Aid for Scientific Research (C) (21500048) from Ministry of Education, Culture, Sports, Science and Technology, Japan. This work is also supported in part by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

References

[1] M. Fukase, H. Uchiumi, T. Ishihara, N. Mimura, K. Narita, T. Takaki, and T. Sato, "Double Cipher Implementation in a Ubiquitous Processor Chip," *Proc. of ECTI-CON 2011*, pp.125-128, May 2011.

[2] M. Fukase, H. Uchiumi, T. Ishihara, Y. Osumi, and T. Sato, "Cipher and Media Possibility of a Ubiquitous Processor," *Proc. of ISCIT 2009*, pp. 343-347, Sept. 2009.

[3] M. Fukase and T. Sato, "A Waved Multifunctional Unit on Account of Multimedia Mobile Computing," *Proc. of WMSCI 2009*, Vol. III, pp. 86-91, Jul. 2009.

[4] T. Austin, D. Blaauw, T. Mudge, and K. Flautner, "Making Typical Silicon Matter with Razor," *Computer Magazine*, Vol. 37, No. 3, pp. 57-65, Mar. 2004.

[5] Y. Lee, D.-K. Jeong, and T. Kim, "Comprehensive Analysis and Control of Design Parameters for Power Gated Circuits," *IEEE Trans. on VLSI Syst.*, Vol. 19, No. 3, pp. 494-498, Mar. 2011.

[6] W. Shen, Y. Cai, X. Hong, and J. Hu, "An Effective Gated Clock Tree Design Based on Activity and Register Aware Placement," *IEEE Trans. on VLSI Syst.*, Vol. 18, No. 12, pp. 1639-1648, Dec. 2010.

[7] T. Mudge, "Power: A First-Class Architectural Design Constraint," *Computer Magazine*, Vol. 34, No. 4, pp. 52-58, April, 2001.

[8] Y-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance Optimization Using Variable-Latency Design Style," *IEEE Trans. on VLSI Syst.*, Vol. 19, No. 10, pp. 1874-1883, Oct. 2011.

[9] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage Scalable High-Speed Robust Hybrid Arithmetic Units Using Adaptive Clocking," *IEEE Trans. on VLSI Syst.*, Vol. 18, No. 9, pp. 1301-1309, Sept. 2010.

[10] X. Chen, J. Yang, and L.-X. Shi, "A Fast Locking All-Digital Phase-Locked Loop via Feed-Forward Compensation Technique," *IEEE Trans. on VLSI Syst.*, Vol. 19, No. 5, pp. 857-868, May 2011.

[11] A. Kurokawa, T. Takaki, and M. Fukase, "Efficient Delay Cells for Wave Pipelined Multifunctional Unit," *Proc. of SASIMI 2012*, pp. 121-126, Mar. 2012.

Masa-aki Fukase received the B.S., M.S., and Dr. of Eng. Degrees in Electronics Engineering from Tohoku University in 1973, 1975, and 1978, respectively. He was Research staff member from 1978 to 1979 at The Semiconductor Research Institute of the Semiconductor Research Foundation. He was Assistant Professor from 1979 to 1991, and Associate Professor from 1991 to 1994 at the Integrated Circuits Engineering Laboratory of the Research Institute of Electrical Communication, Tohoku University. He has been Professor of computer engineering since 1995 at the Faculty of Science and Technology, Hirosaki University. He served as the Director of the Hirosaki University C&C Systems Center from 2004 to 2012. He has been the Representative of Hirosaki University R&DC of Next Generation IT Technologies since 2008. His current research activities are mainly concerned with the design, chip implementation, and application of power conscious highly performable VLSI processors.