American Journal of Computer Architecture 2012, 1(1): 6-11
DOI: 10.5923/j.ajca.20120101.02

Double Cipher Implementation in a Ubiquitous Processor Chip

Masa-aki Fukase[1,*], **Tomoaki Sato**[2]

[1]Graduate School of Science and Technology, Hirosaki University, Hirosaki, 036-8561, Japan
[2]Computer and Network Systems Center, Hirosaki University, Hirosaki University, Hirosaki, 036-8561, Japan

Abstract This paper focuses on improving the cipher strength of a particular ubiquitous processor HCgorilla. The reason why this is called the ubiquitous processor is due to its specific features for ubiquitous computing. Ubiquitous computing is really a leading edge trend of next generation information and communication technologies. One of the most promising solutions for ubiquitous computing applied to HCgorilla is exploiting not higher speed but parallelism. It has progressed the overall status of ubiquitous and processor techniques. The basic organization of HCgorilla follows multicore and multiple pipelines. These are Java compatible media pipelines (shortened to pipes hereafter) with sophisticated structures and cipher pipes. The cipher pipe implements transposition cipher called RAC (random number addressing cryptography) by using a hardware RNG (random number generators). Although RAC is excellent at software-transparency, it is not always sufficient for practical security. Since emerging ubiquitous environment requires strong security as well as high performance, it is also a crucial issue to enhance cipher strength. Thus, the improved HCgorilla in this study embeds two RNGs. These are used for double cipher, that is, the one for RAC and the other for a substitution cipher by data sealing. This approach promises strong cipher strength without any overhead for hardware cost, power dissipation, throughput, etc. Various aspects of the improved HCgorilla are evaluated.

Keywords Ubiquitous Processor Chip, Hardware Cryptography, Double Cipher, Power Dissipation, Throughput

1. Introduction

The advent of ubiquitous environment still demands power conscious strong security, high performance, high precision, and real time responsibility for processors. Thus, next generation processors for ubiquitous environment are desired to be developed. Strategy for the development of ubiquitous processors is to achieve higher throughput, stronger security, and lower power dissipation for large quantity of multimedia data[1-4]. 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[5-7]. This has led to increased interest in efficient implementations of cryptographic primitives[8].

The authors have given challenges for the development of ubiquitous processors as follows. In order to satisfy the demand for data quantity and performance, the adoption of block ciphers is inevitable. Then, hardware cryptography is profitable to achieve power conscious strong security[9].

With respect to this view point, a particular ubiquitous processor, HCgorilla has been really eligible[10]. The basic organization of HCgorilla follows multicore and multiple pipelines[11]. They are Java compatible media pipes and hardware cryptography embedded cipher pipes. By using a waved MFU (multifunctional unit) in the execution stage, the media pipe issues arithmetic instructions free from complicated scheduling[12]. This is a very sophisticated instruction level parallelism to achieve power conscious high performance[13]. The cipher pipe implements transposition cipher called RAC (random number addressing cryptography) by using a built-in hardware RNG.

Although RAC is excellent at software-transparency, it is not always sufficient for practical security. Since emerging ubiquitous environment requires strong security as well as high performance, it is also a crucial issue to enhance cipher strength. Thus, this paper focuses on increasing the cipher strength of previous HCgorilla. In general, the cipher strength can be improved by increasing both the key length and kinds of bit operations, which can be seen in the improvement from DES (Data Encryption Standard) to AES (Advanced Encryption Standard)[14]. This is based on the general rule of deciphering a secret key cryptography that seeks an unknown key or password, assuming plaintext, cipher text, and encryption algorithms are open[15].

The authors propose in this study a double cipher scheme with two RNGs[16]. The double cipher approach promises practically strong cipher strength by providing double kind

* Corresponding author:
slfuka@eit.hirosaki-u.ac (Masa-aki Fukase)
Published online at http://journal.sapub.org/ajca

of bit operations. The additional RNG is used for a substitution cipher by data sealing. Consequently, this increases the key length. Another advantage of this approach is power consciousness with high area efficiency and high throughput due to microarchitecture level hardware mechanism.

Various aspects of the improved HCgorilla are studied in this paper. Metrics related to the evaluation are occupied area, power consumption, cipher strength, running time, and throughput. According to evaluation, the double cipher promises practically secure strength without any overhead. A new knowledge to be drawn from this study is to apply the temporary security of the double ciphe to ad-hoc cipher streaming without permanent network infrastructure.

2. Processor Design

Fig. 1 shows the hardware organization of the improved version, HCgorilla.6. This has two symmetric cores. Each core is composed of two media pipes and a cipher pipe.

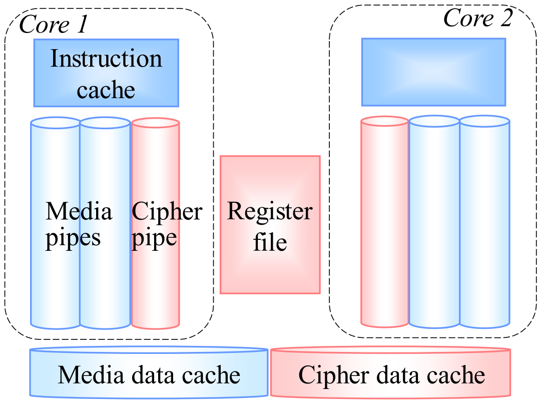

Figure 1. Organization of HCgorilla.6

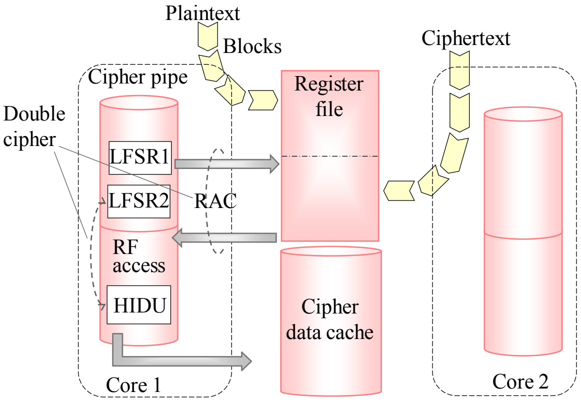

Figure 2. Double cipher mechanism during data transfer

2.1. Cipher Pipe

The cryptographic defect of the previous HCgorilla originates in embodying only one hardware RNG. Transposition cipher like RAC cannot hide data structure. Thus, the HCgorilla.6's cipher pipe is built in two RNGs. Fig. 2 illustrates the structural aspect, internal behavior, and algorithm of the cipher pipe. LFSR (linear feedback shift register) is used as RNG to achieve longer cycle with negligibly small overhead. This is because LFSR does not consume so much hardware cost, area, power consumption, running time, throughput, etc.[17]. The one of two LFSRs covers the transposition cipher of RAC and the other is used for substitution cipher. The substitution cipher is implemented by a hidable unit, HIDU.

The external data of plaintext or cipher text is divided into blocks. The register file plays the role of streaming buffer. It buffers a block of external data. The transfer of the block to the register file is assumed to be DMA mode, though it is not our concern in this study.

The cipher pipe executes a SIMD (single instruction stream multiple data stream) mode instruction. This occupies the cipher pipe as long as the corresponding data stream continues. The SIMD mode sequence forms double cipher streaming. RAC is carried out by making LFSR1's output specify a register file address, synchronizing a data cache address with the current clock count, and transposing the specified register file's content to the synchronized data cache address. Then, LFSR2 makes HIDU on data lines work for the substitution of transferred data. The resultant content stored in data cache is the double encryption of the register file's content. Such a microarchitecture-based, software-transparent mechanism offers the protection of the whole data with negligible hardware cost and moderate performance overhead. Similarly, core 2 does double cipher decryption or encryption.

2.2. Media Pipe

The media pipe shown in Fig. 1 is a sort of an interpreter type Java CPU[18]. The media pipe uses a waved MFU in the execution stage. This is the combination of wave-pipelining and multifunctionalization of arithmetic logic functions for media processing. Since the latency of the waved MFU is constant independently on arithmetic logic operations, the media instructions are free from scheduling[10].

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[19]. 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. While registers occupy large area, the wave-pipelining does not. Therefore, the wave-pipelining is more effective in view of hardware cost, power saving, speed up, and throughput degradation.

Power saving of processor systems has been done by the control of supply voltage and clock. The supply voltage is sometimes scaled down[20] and sometimes gated[21]. Then, the clock is also scaled[22] and gated[23]. More sophisticated clock systems are clocks-variable[24], cycle-variable or adaptive clock[25]. However, these are accompanied with tradeoff against throughput. This in turn causes considerable overhead. Wave-pipelining is a more effective strategy for power saving at a lower level. It has been so far applied to arithmetic logics, circuit blocks, pipeline stages, etc.

3. Results and Discussion

Basic features of the improved HCgorilla are studied. Fig. 3 illustrates a test program and internal behavior in the evaluation of HCgorilla.6's throughputs focusing on the core 1 for the sake of simple representation. A test program is composed of the double cipher and media processing. The plaintext used in the double cipher processing is a 240×320-pixel QVGA format data.

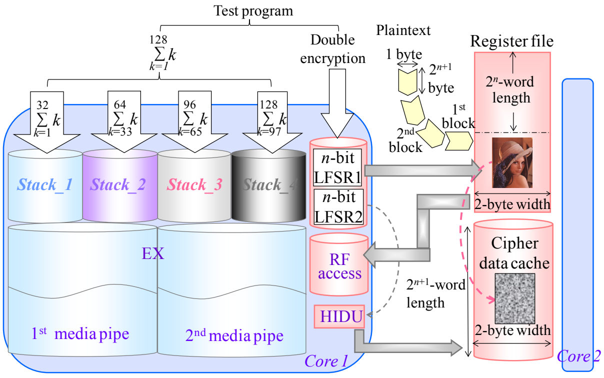

Figure 3. A test program and internal behaviour in the evaluation of HCgorilla.6's throughput

The media processing shown in Fig. 3 is a simple summation. This is used to validate the two effects, that is, instruction issues free from scheduling and hardware parallelism of the execution stage of HCgorilla.6. The instruction scheduling free effect is examined by carrying out the summation in three ways or the three routines *A*, *B*, and *C*. These are distinguished by that the variable *k* and loop count are integers or floating point numbers. The routine *A* uses only integers and the routine *B* does floating point numbers. Then, the routine *C* uses both integer and floating point numbers. The hardware parallelism is utilized by dividing the summation into four threads and assigning them into four stacks in order to fully make use of the two waved MFUs.

3.1. Evaluation of the Double Cipher

The HCgorilla.6's cipher pipe is evaluated in view of cipher strength and throughput. Fig. 4 shows how to measure the double cipher strength through the experiment of rough-and-ready guess or round robin attack in a ubiquitous environment where HCgorilla built-in platforms are used. The cipher strength is the degree of enduringness against attack by a malicious third party. The attack is the third party's irregular action to do decipher, break, or crack. This is clearly distinguished from decryption that is the right recipient's regular process to recover the plaintext by using the given key.

According to a normal scenario, the rules applied to deciphering a secret key cryptography in Fig. 4 are as follows.

(a) A plaintext, cipher text, and the cipher algorithm are open to third parties.

(b) A key or the initial value of RNG used in encryption is secret from third parties, though it is open to a right recipient.

What is sought out in deciphering is a true key. Sometimes it is called a password. The reason why a plaintext and a cipher text are open is because they are various which in turn their quantity is beyond the protection. In addition, it is reasonable that the cipher algorithm or its specification is open because its value is usability in communication stages. This demands the spread of the algorithm in some community.

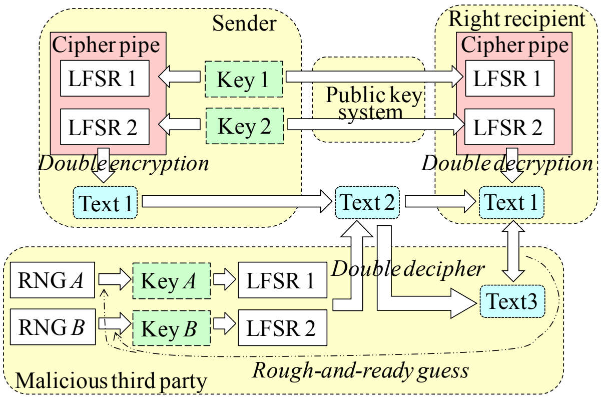

Figure 4. Round robin attack against the secret key cryptography system including HCgorilla built-in ubiquitous platforms

LFSR 1 and LFSR 2 in Fig. 4 are RNGs for the double cipher built in the cipher pipes of a sender and a right recipient. They are also used by third parties according to the rule (a). Key 1 and key 2 are the initial values of LFSR 1 and LFSR 2 issued by the sender. Further encryption of these secret keys is conventionally applied to keep the confidentiality of themselves that are the target of attacks. For example, WEP (wired equivalent privacy) cipher keys are encrypted by RC4 cipher. In Fig. 4, a public key system is available in exchanging the key between a sender platform and a right recipient platform.

Text 1 is a plain/cipher text and Text 2 is a cipher/plain text derived by applying the Key 1 and Key 2 to Text 1. Key *A* and Key *B* are the guess of Key 1 and Key 2 and are the initial values of the third partie's LFSR1 and LFSR 2. RNG *A* and RNG *B* that are completely independent on LFSR1 and LFSR 2 are used for rough-and-ready guess or random guess by the third party. Text 3 is the guess of Text

1 by the third party. When Text 3 disagrees with Text 1, the one of RNG *A* and RNG *B* is made to proceed to a next stage. If the disagreement continues by the end of the cycle of random number generation, the comparison of Text 3 and Text 1 is repeated by using the other RNG. Thus, the round robin attack against the double cipher undergoes nested loops.

From the discussion described above, the cipher strength is given by

Cipher strength=time for the round robin attack

=no. of round robin attacks×clock cycle time

$$\leq 2^{\text{LFSR 1 size+LFSR 2 size}} \times \text{clock cycle time} \quad (1)$$

Since

LFSR 1 or LFSR 2 size

$$\geq \log_2\{\text{register file length}/2\} \quad (2)$$

holds from Fig. 3, the register file length is a critical factor of cipher strength. However, enlarging memory size surely causes the increase of power dissipation, the deterioration of clock speed, throughput, etc. Thus, the demand of cipher strength is inevitably limited.

Fig. 5 shows the register file length dependency of HCgorilla.6's throughput. in running the test program shown in Fig. 3. The register file length is swung backward and forward from the HCgorilla.6's register file length, 128 words.

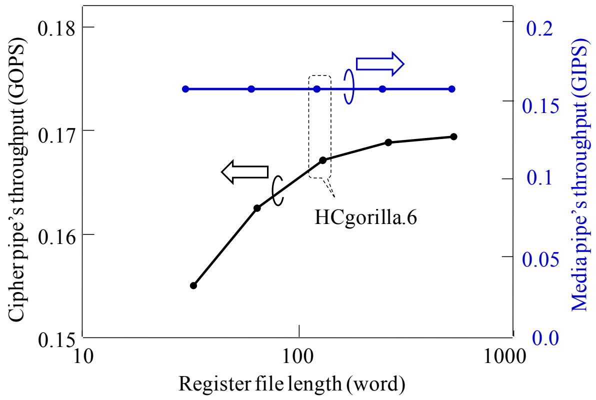

Figure 5. Throughput vs. register file length

The cipher pipe's throughput that is the mean value of the number of double cipher operations per unit time is derived by using

$$\text{Throughput [OPS]} = \frac{\text{no.of double cipher operations}}{\text{running time [sec]}} \quad (3)$$

Since the running in (3) is the repetition of block transfer, rewriting the register file, and double cipher operation, the running time is derived from

$$\text{Running time}=m(t_1+t_2)+t_3 \quad (4)$$

Here, *m* is the number of blocks. The block width is usually fixed to byte because ubiquitous media like pixels takes the form of byte structured stream. As a consequence, the register file width is evaluated by byte. On the other hand, the register file length is measured expedientially by word as shown in Fig. 3. t_1 is block access time or the latency taken to transfer a block to the register file. t_2 is the time of a SIMD mode cipher operation. t_3 is latency taken to transfer a block from data cache. As for t_1 and t_3, let the memory access speed of cellar phones be 208 to 532 Mbytes/s and the mean value is adopted.

The cipher pipe's transfer rate that is the mean value of the amount of transferred data per unit time is given by

$$\text{Transfer rate [bps]} = \frac{\text{full text size [b]}}{\text{transfer time [sec]}} \quad (5)$$

Identifying the transfer time in the denominator of (5) with the running time in (4), following relation is derived.

Transfer rate [Mbps]

$$=\text{throughput [GOPS]}\times\text{register file width [b]}\times 10^3 \quad (6)$$

The media pipe's throughput is almost constant in Fig. 5. This is because the clock speed is kept constant in varying the length of the register file. Then, the media pipe's throughput differs little between the routines *A*, *B*, and *C* as shown in Fig. 6. This is due to the instruction scheduling free media pipes.

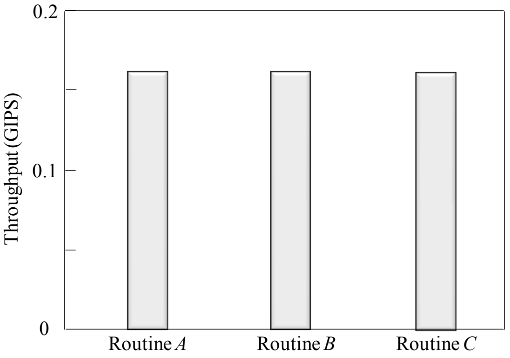

Figure 6. Media pipe's throughput

Table 1. Prospective Specifications and Potential Aspects of HCgorilla Chips

			HCgorilla.3	HCgorilla.5	HCgorilla.6
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
	Core		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
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		

3.2. Total Evaluation of HCgorilla

Considering (i) a secret key cryptography uses an algorithm that repeats simple operations like EXOR and shift, (ii) the algorithm is open to a third party, (iii) the cipher strength is determined by the length of a key, the point in the evaluation of the hardware implementation of a secret key cryptography is not cipher strength but hardware specifications. Table 1 summarizes chip parameters and hardware specifications of HCgorilla.6 and previous derivatives. The

HCgorilla family uses the same 0.18-μm CMOS standard cell technology. Clock frequency is derived from timing analysis by using a Synopsys design compiler. Power dissipation is derived from static analysis. HCgorilla.6 and HCgorilla.5 have almost the same structure except the number of built in RNG. Since RNG occupies negligibly small area, the same clock frequency is reasonable.

One of the most important metrics of ubiquitous devices is power dissipation. Fig. 7 shows the register file length dependency of power dissipation and chip area. The register file length is swung similarly to Fig. 5. Power dissipation in this figure is rough approximation derived from static analysis, which summarizes the mean value of every gate. It does not take into account of switching condition. Fig. 8 shows the breakdown of HCgorilla.6's power consumption more in detail. In addition, the effect of double cipher on power dissipation is shown by comparing HCgorilla.6 and HCgorilla.5.

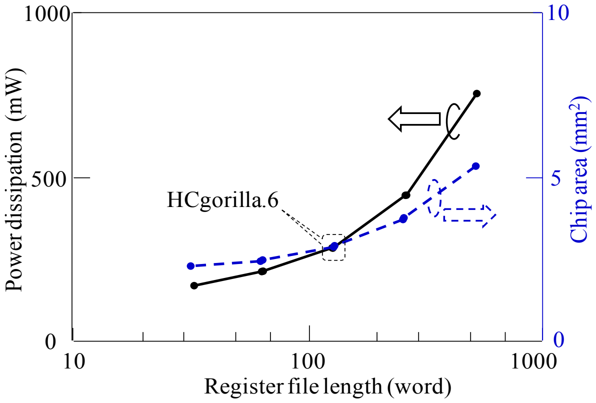

Figure 7. Register file length dependency of power dissipation and chip area

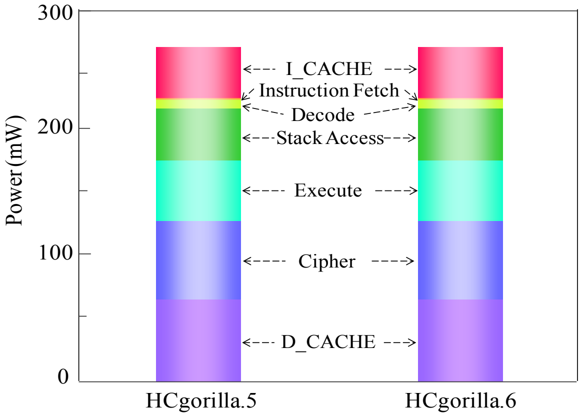

Figure 8. The effect of double cipher on power dissipation

3.3. Discussion

The cost of increasing RNGs is negligible small compared with the HCgorilla.6 chip. The hardware cost of each cipher pipe is 270 cells that occupy only 0.1 mm square. The power dissipation of each cipher pipe is 30 mW from Fig. 8. Since this value is based on rough approximation, more practical value based on dynamic analysis or experimental measurement is smaller than this. Accordingly, HCgorilla.6 implemented in a 0.18-μm CMOS chip is provided with hardware specifications suited to ubiquitous devices.

While the cipher pipe's transfer rate is almost enough because it is comparable to ATM's one, the throughput in GOPS should be improved in view of CPU performance. Actually, the cipher pipe's throughput almost saturates at 128-word length from Fig. 5. In order to enhance GOPS value, which directly affects the increase of Mbps value, running time should be decreased from (3). This is possible according to following strategies.

(a) Reducing t_1 and t_3 by using memory buffer with faster access speed.

(b) Reducing the summation of block access and transfer times, $\sum(t_1+t_3)$ by increasing the register file size. Expanding the register file length leads to the increase of cipher strength. However, it needs to take into account of tradeoff between throughput and power dissipation. Increasing the register file size causes more power dissipation judging from (3). In fact, the power dissipation rapidly increases from 128-word length from Fig. 7.

(c) Reducing t_2 by the speed up of the cipher pipe's clock. Increasing the number of pipeline stages is also useful for this aim.

4. Conclusions

In order to solve the cryptographic issue of the previous versions of the ubiquitous processor, HCgorilla family, a double cipher scheme has been implemented in this study. While the double cipher is clearly stronger than the single cipher of RAC, other metrics like occupied area, power consumption, cipher strength, running time, and throughput are almost independent on the number of RNGs. The optimum buffer size of the latest version, HCgorilla.6 is 128-word length judging from throughput and power tendencies.

The benefits of the improved HCgorilla.6 are very profitable from the new knowledge drawn from this study. The double cipher progressively offers the temporary protection of the whole data. The temporary security of the double cipher is really applicable for ad-hoc cipher streaming without permanent network infrastructure.

Next steps of this study are as follows. Firstly, the throughput in GOPS should be improved in view of CPU performance. Secondly, the double cipher scheme should be implemented in the media pipe, though the cipher pipe and the media pipe are explicitly distinguished each other in this study. The microarchitecture level mixing of instruction scheduling free media processing with cipher processing will contribute power conscious security in ubiquitous network.

ACKNOWLEDGEMENTS

This work is supported by Grant-in-Aid for Scientific Research (C) (21500048) from Ministry of Education, Culture, Sports, Science and Technology, Japan. In addition, this work is 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. Who, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "AnySP: Anytime anywhere anyway signal processing," IEEE micro, Vol. 30, No. 1, pp. 81-91, Jan./Feb. 2010.

[2] M. Murata, "Towards ambient information society," The Jour. of IEICE Vol. 93, No. 3, pp. 233-238, Mar. 2010.

[3] H. Lee, C. Chakrabarti, and T. Mudge, "A low-power DSP for wireless communications," IEEE Trans. on VLSI Syst., Vol. 18, No. 9, pp. 1310-1322, Sept. 2010.

[4] A. M. Caulfield, L. M. Grupp, and S. Swanson, "Gordon: An improved architecture for data-intensive applications," IEEE micro, Vol. 30, No. 1, pp. 121-130, Jan./Feb. 2010.

[5] R. Oppliger, "Security and privacy in an online world," Computer Magazine, Vol. 44, No. 9, pp. 21-22, Sept. 2011.

[6] A. Stavrou, J. Voas, T. Karygiannis, and S. Quirolgico, "Building security into off-the-shelf smartphones," Computer Magazine, Vol. 45. No. 2, pp. 82-84, Feb. 2012.

[7] F. Burns, A. Bystrov, A. Koelmans, and A. Yakovlev, "Security evaluation of balanced 1-of-n circuits," IEEE Trans. on VLSI Syst., Vol. 19, No. 11, pp. 2135-2139, Nov. 2011.

[8] S. O'Melia and A. J. Elbirt, "Enhancing the performance of symmetric-key cryptography via instruction set extensions," IEEE Trans. on VLSI Syst., Vol. 18, No. 11, pp. 1505-1518, Nov. 2010.

[9] T. Good and M. Benaissa, "692-nW advanced encryption standard (AES) on a 0.13-μm CMOS," IEEE Trans. on VLSI Syst., Vol. 18, No. 12, pp. 1753-1757, Dec. 2010.

[10] M. Fukase and T. Sato, "A ubiquitous processor built-in a waved multifunctional unit," ECTI-CIT Trans. Vol. 4, No. 1, pp. 1-7, May 2010.

[11] M. Levy and T. M. Conte, "Embedded multicore processors and systems," IEEE micro, Vol. 29, No. 3, pp. 7-9, May/Jun. 2009.

[12] A. Kurokawa, T. Takaki, and M. Fukase, "Efficient delay cells for wave pipelined multifunctional unit," Proc. of SASIMI 2012, pp. 121-126, Mar. 2012.

[13] B. Catanzaro, A. Fox, K. Keutzer, D. Patterson, B.-Y. Su, M. Snir, K. Olukotun, P. Hanrahan, and H. Chafi, "Ubiquitous parallel computing from Berkeley, Illinois, and Stanford," IEEE micro, Vol. 30, No. 2, pp. 41-55, Mar./Apr. 2010.

[14] M.-Y. Wang, C.-P. Su, C.-L. Horng, C.-W. Wu, and C.-T. Huang, "Single- and multi-core configurable AES architectures for flexible security," IEEE Trans. on VLSI Syst., Vol. 18, No. 4, pp. 541-552, Apr. 2010.

[15] M. Matsui, "Survey of the research and development of MISTY cryptography," Jour. of Digital Practice, Vol. 2, No. 4, pp. 282-289, Oct. 2011.

[16] H. Uchiumi, T. Ishihara, M. Fukase, and T. Sato, "Development and evaluation of a next generation ubiquitous processor chip," 2010 Tohoku-Section Joint Convention Record of Institutes of Electrical and Information Engineering Japan, p. 282, Aug. 2011.

[17] S. Wang and S. K. Gupta, "DS-LFSR: A BIST TPG for low switching activity," IEEE Trans. on CAD of IC and Syst., Vol. 21, No. 7, pp. 842-851, Jul. 2002.

[18] K.-Y. Chen, J.M. Chang, T.-W. Hou, "Multithreading in Java: performance and scalability on multicore systems," IEEE Trans. on Computers, Vol. 60, No. 11, pp. 1521-1534, Nov. 2011.

[19] J. Xu, W. Wolf, and W. Zhang, "Double-data-rate, wave-pipelined interconnect for asynchronous NoCs," IEEE micro, Vol. 29, No. 3, pp. 20-30, May/Jun. 2009.

[20] 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.

[21] 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.

[22] 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.

[23] T. Mudge, "Power: A first-class architectural design constraint," Computer Magazine, Vol. 34, No. 4, pp. 52-58, April, 2001.

[24] Y-S. Su, D.-C. Wang, S.-C. Chang, and M. M.-Sadowska, "Performance optimization using variable-latency design style," IEEE Trans. on VLSI Syst., Vol. 19, No. 10, pp. 1874-1883, Oct. 2011.

[25] 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.