

Development of a Next Generation Ubiquitous Processor Chip

Masa-aki Fukase¹, Kohei Ichinohe², Kazuki Narita³, Tatsuya Takaki⁴,
Naomichi Mimura⁵, Non-members,
Tomoaki Sato⁶, Member, and Atsushi Kurokawa⁷, Non-member

ABSTRACT

Power conscious PC processors, mobile processors, cryptography processors, RFID tags, etc. have obviously supported the trend of ubiquitous network and computing. The authors have developed a ubiquitous processor named HCgorilla under the strategy to unify these devices. The unification of those devices has been effective to achieve power consciousness for Green IT and secureness for ubiquitous environment. The target of this study is the shift of HCgorilla to SoC, considering the exploitation of adaptability to SoC and resource-constrained implementation is cutting edge microprocessor tendency. Since this requires the total improvement of HCgorilla, the complicated clock schemes are optimally unified together. One of most important clock schemes applied to HCgorilla is a waved MFU (multifunctional unit). Although gated clock and scan path have been supported by standard CAD tools, the wave-pipelining has been mainly done by manual tuning because it has not always been so popular. Thus, these clock schemes are totally introduced. Specific features of the improved HCgorilla and its implementation in a CMOS standard cell chip are described in this article. The HCgorilla.7 chip developed in this study is the best in view of clock speed, occupied area, power consumption, and hardware security.

Keywords: Ubiquitous Processor, SoC, CMOS Chip, Clock Scheme, Waved MFU

1. INTRODUCTION¹

Most remarkable trends of next generation information and communication technologies are ubiquitous network, computing, environment, etc. However, the rapid increase of recent years ubiquitous technologies have given rise to serious concerns about

power dissipation and security issues which are getting worse by the fact that the majority of these technologies are resource constrained in terms of chip area and battery energy available. For example, LCD and processors consume similar power in running mobile devices. While LCD is turned on only when it is used to display some information, processors are always in standby state to receive calling anytime. Thus, the restriction for the power saving of mobile devices is inevitably imposed on processors.

In order to solve the issue described above, the authors have unified the role of PC processors, mobile processors, cryptography processors, RFID tags, etc. into a ubiquitous processor architecture named HCgorilla [1]. Strategy for the development of HCgorilla has been to achieve power consciousness for Green IT and secureness for ubiquitous environment. In order to attain power conscious high performance, HCgorilla has promptly followed the paradigm shift of PC processors from the speed up of clock to the introduction of parallelism. One of most important processor techniques applied to HCgorilla is a waved MFU (multifunctional unit) that is the combination of multifunctionalization and wave-pipelining [2]. This is very effective for scheduling free ILP (instruction level parallelism). In addition, the wave-pipelining has potential features for power conscious high speed [3].

Considering cutting edge microprocessor tendency, the adaptability of HCgorilla to SoC and resource-constrained implementation are crucial features. Thus, the shift of HCgorilla to SoC is very important. In order to exploit the adaptability to SoC, the total improvement of HCgorilla is required. With respect to this viewpoint, it is really effective to improve complicated clock schemes. Although gated clocking, scan test, and wave-pipelining are known, the situation of most microprocessor techniques is as follows.

- They are dedicated to individual fields. The gated clocking has been so far applied to regular pipelines. To the best of our knowledge, it has not yet applied to the waved MFU in conjunction with parallel stack.
- Individual technique is not so enough by itself, but is used in the form of hybrid techniques.
- While gated clocking and scan test have been supported by regular CAD tools, the wave-pipelining has been mainly done by manual tuning because it

Manuscript received on March 1, 2012 ; revised on June 20, 2012.

¹ This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

^{1,2,3,3,4,5} The authors are with Graduate School of Science and Technology, Hirosaki University, Hirosaki 036-8561, Japan. , E-mail: slfuka@eit.hirosaki-u.ac.jp

^{6,7} The authors are with C&C Systems Center, Hirosaki University, Hirosaki 036-8561, Japan. , E-mail: tsato@cc.hirosaki-u.ac.jp

has not always been so popular.

Thus, the aim of the study is the further improvement of HCgorilla chips for the next generation ubiquitous environment with particular emphasis on the total optimum design of the HCgorilla's clock scheme [4]. The improved HCgorilla is implemented in a CMOS standard cell chip. Specific features of the HCgorilla chip are also described.

2. ARCHITECTURE

2.1 Architecture and Instruction Set

Fig. 1 illustrates the architecture of HCgorilla that takes symmetric multicore and multiple pipeline structure. Following HW/SW codesign approach, HCgorilla receives the benefit of software support such as parallelizing compilers and language interface [1]. Symmetric cores run multiple threads in parallel. Each core has two arithmetic media pipes and a cipher pipe.

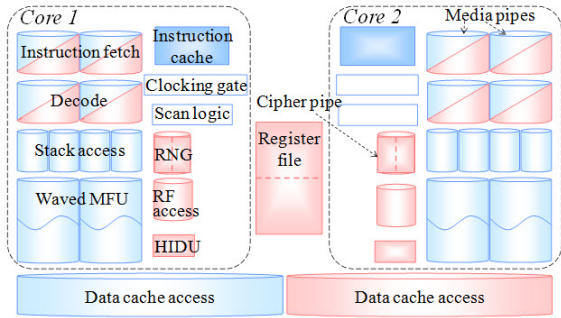


Fig. 1: Architecture of HCgorilla

Table 1: Instruction Set of HCgorilla

Operation	Instruction	Operation	Instruction	Operation	Instruction	
Cipher	rsw	Integer type	Stack 0	Integer type	Stack 0	
	rlw		Stack 1		Stack 1	
	nop		bipush		load	lload
	ifeq		sipush		load_0	lload_0
	ifne		pop		load_1	lload_1
	iflt		pop2		load_2	lload_2
	ifge		dup		load_3	lload_3
	ifgt		dup_x1		store	lstore
	ifle		dup_x2		store_0	lstore_0
	Jump		if_icmpeq		dup2	store_1
if_icmpne		dup2_x1	store_2	lstore_2		
if_icmplt		dup2_x2	store_3	lstore_3		
if_icmpge		swap	iadd	liadd		
if_icmpgt		iconst_m1	isub	lisub		
if_icmple		iconst_0	imul	limul		
goto		iconst_1	ineg	lineg		
		iconst_2	ishl	lishl		
		iconst_3	ishr	lishr		
		iconst_4	iushr	liushr		
	iconst_5	iand	liand			
	ficonst_0	ior	lior			
	ficonst_1	ixor	lixor			
	ficonst_2	fadd	lfadd			
		fsub	lfsub			
		fmul	lfmul			

Table 1 summarizes the instruction set of HCgorilla. It is composed of 2 cipher instructions executed by a cipher pipe and 58 Java compatible instructions executed by a media pipe. The cipher instructions

are SIMD (single instruction stream multiple data stream) mode. Cipher streaming is possible due to SIMD mode instructions. Excepting jump instructions, sequential media instructions are related to stack operation because the HCgorilla's media pipe basically follows Java CPU and is a stack machine. The stack-related media instructions are divided into integer and floating point types. Floating point arithmetic instructions is provided to cover Java compatibility and the capability of precise control, which are fundamental aspects of SoC applications.

The reason why stack-related media instructions are provided with a set of two is due to the existence of two stacks per media pipe. The meaning of the parallel stack is described at the end of next section. Arithmetic instructions are also provided with each stack. A pair of arithmetic instructions, for example, imul and lmul, are alternately issued by one clock.

2.2 Waved MFU

The media pipe's execution (EX) stage or waved MFU is described more in detail. Although floating point arithmetic is inevitable for SoC, floating point arithmetic instructions take longer latency in the execution by a general EX stage. This is composed of IU (integer unit) and FPU (floating point unit) as shown in Fig. 2 (a). This requires instruction scheduling in the execution of the mixed sequence of floating point and integer type instructions, which frequently occurs in SoC applications. Then, the instruction scheduling causes pipeline disturbance that surely degrades the throughput in instructions per second.

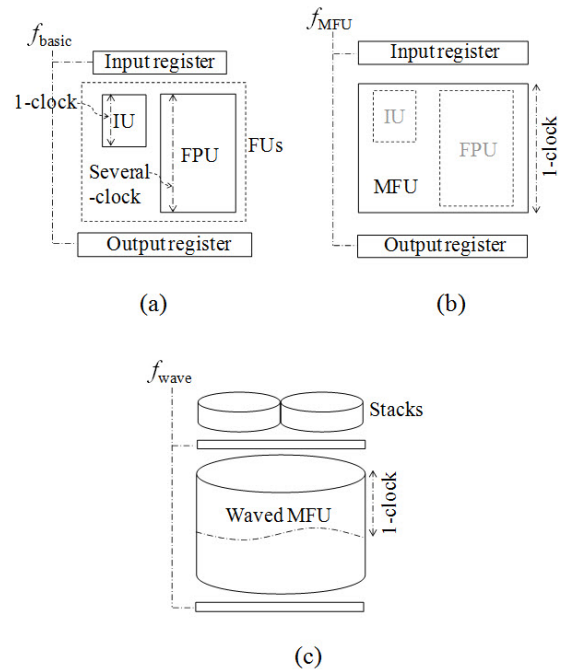


Fig. 2: Organization of (a) a basic EX stage (b) MFU (c) a waved MFU

A possible way to avoid the instruction scheduling required by the basic EX stage shown in Fig. 2 (a) is to merge IU and FPU into MFU as shown in Fig. 2 (b). This surely executes every function with the same latency. However, the MFU's clock speed f_{MFU} is slower than that of the basic EX stage, f_{basic} because 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 FUs (functional units) does not always promise the total enhancement of processor performance.

In order to completely unify FUs without deteriorating clock speed, wave-pipelining shown in Fig. 2 (c) is really promising. While a regular pipeline depends on pipeline registers and the tuning of maximum delay, a wave-pipeline does not use pipeline registers [5]. Since the clock speed f_{wave} of the waved MFU is not related to logic depth, but determined by the difference between the critical path delay and the minimum path delay of the waved circuit, f_{wave} is improved as follows.

$$f_{wave} < f_{MFU} \quad (1)$$

Thus, the employment of a waved MFU in the EX stage of the HCgorilla's media pipe releases instruction scheduling without any overhead and thus overcomes the fundamental issue of EX stage described above.

The meaning of providing the waved MFU with parallel stack, as is illustrated in Fig. 1 and Fig. 2 (c), is to adjust the timing between the 1-clock stack stage and the two-clock waved MFU. The timing constraint for the media pipe is held by making the two stacks alternatively supply data to the waved MFU. The parallel stack is prepared corresponding to the following relation.

$$\text{Wave degree} = \text{waved MFU's latency} / f_{wave}. \quad (2)$$

Although clocks-variable [6] and cycle-variable or adaptive clock [7] might be more sophisticated clock systems than wave-pipelining, these are accompanied with trade-off against throughput. With respect to this point, the waved MFU in combination with parallel stacks is a more effective strategy for clock systems.

2.3 Overall Behaviour

Fig. 3 shows the internal behaviour of HCgorilla operating at full capacity. The core 1 is focused for the sake of simple representation and data flow is illustrated within both media and cipher pipes. Media data is divided into two threads and assigned into the media pipes. Then, each thread is further divided into the two stacks, which alternatively supply data to the waved MFU.

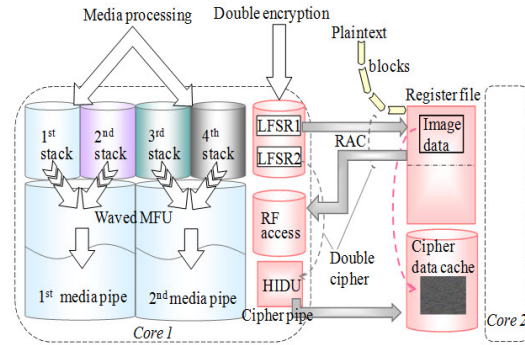


Fig. 3: Internal behaviour of HCgorilla

Cipher data is buffered in the register file that is the shared memory of the double core. Register file and data cache is word structured to deal with plain and cipher texts. These are divided into blocks as is similar to AES (advanced encryption standard) and DES (data encryption standard). Since each word is 2-byte width, a pixel, for example, occupies one and a half word. The plaintext of an image data, exemplified as cipher data in Fig. 3, is divided into blocks whose size corresponds to the size of buffer or register file.

Due to restricted hardware quantity in designing HCgorilla, a cipher pipe uses LFSR (linear feedback shift register) as RNG (random number generator). The cost of LFSR is negligible small. The cipher pipe does double encryption during the transfer of the image data from the register file to data cache [4]. While an LFSR controls the transposition cipher called RAC (random number addressing cryptography), another LFSR controls a substitution cipher implemented by the hidable unit, HIDU.

2.4 Fundamental Aspects

The fundamental aspects of HCgorilla are three-fold. The first one is power conscious resource-constrained implementation for Green IT. In general, power saving of processor systems has been done by the control of supply voltage, clock, and parallelism. The supply voltage is sometimes scaled down [8] and sometimes gated [9]. Then, the clock is also scaled [10] and gated [11]. 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 not a critical factor in the case of the 0.18- μm CMOS standard cell process used in this study, the gated clock is very effective for power saving. The parallelism has been applied at various levels from program to microarchitecture. Wave-pipelining at a lower level is a more effective strategy for power saving as well as high speed clocking [2]. It has been so far applied to arithmetic logics, circuit blocks, pipeline stages, etc.

Fig. 4 shows the total clock scheme applied to the HCgorilla's media pipe. The targets of gated clocking are the stack access and EX stages where switching probability is higher. In addition, scan logic is introduced for DFT (design for testability). The scan logic is applied to pipeline registers. Although gated clock and scan path have been supported by regular CAD tools, the wave-pipelining has been mainly done by manual tuning. Thus, the clock scheme shown in Fig. 4 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 the instruction decode stage.

Considering (i) scan test repeats FF switching more frequently than normal operation, (ii) the test clock is not slower than the system clock because detecting delay faults is one of most important target of scan test, scan test consumes large power. This often causes large voltage drop and results in the reverse of signals. In addition, generated heat even causes structural damages to Si, bonding wire, and package [12]. In view of power conscious testing, more efficient cells than MUX, for example, MUX-scan D-FF, is reasonable in exchanging between scan test and normal operation [13].

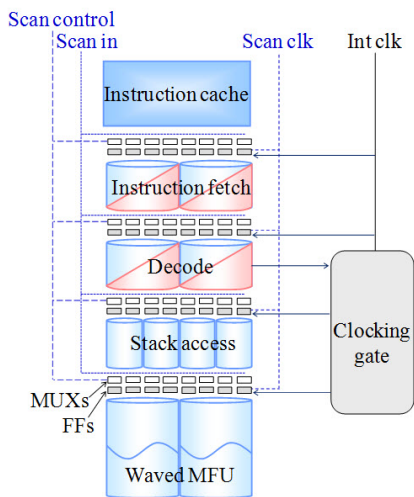


Fig.4: Clock scheme of HCgorilla

The second aspect of HCgorilla is secureness for ubiquitous environment. 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 to AES. 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.

From the above description, the cipher strength of HCgorilla is given by

$$\text{Cipher strength} \leq 2^{\text{LFSR 1 size} + \text{LFSR 2 size}} \times \text{clock cycle time.} \quad (3)$$

Since

$$\begin{aligned} \text{LFSR 1 size} &= \text{LFSR 2 size} \\ &\geq \log_2 \{\text{register file length}/2\} \end{aligned} \quad (4)$$

holds from Fig. 3, the register file length and the block size are critical factors of cipher strength. However, further expanding the register file and data cache surely causes the increase of power dissipation, the deterioration of clock speed, throughput, etc. Thus, the demand of cipher strength is inevitably limited.

Owing to the two aspects described above, HCgorilla is provided with the third aspect of adaptability to SoC. SoC is used in the three major fields of communication, multimedia and control. The double core covers bidirectional communication. Since HCgorilla's media pipe is Java compatible, it fosters multimedia processing in a ubiquitous community. Then, HCgorilla is suited to control because the waved MFU executes floating point arithmetic instructions. This is effective for precise NC (numerical control) machining, GPS navigation, etc.

3. CHIP IMPLEMENTATION

3.1 Design

The chip implementation of HCgorilla is focused on in this section with particular emphasis on the total optimum design of the HCgorilla's clock scheme. Although the clock is wired in the physical layout design after the logic design, HCgorilla's clock scheme is taking into account of in various design steps as shown in Fig. 5.

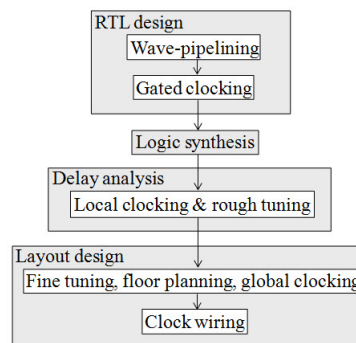


Fig.5: Implementation of the clock scheme

Since the wave-pipelining does not use pipeline registers, its outline is described in the RTL (Register Transfer Level) design step whose role is the allocation of the registers. Gated clocking is also taken into

account of by the RTL design because it is carried out by a control circuit. This is described by the RTL design. Rough tuning in the delay analysis step and fine tuning in the layout design step are the delay tuning required for the wave-pipelining. Although the local clock and global clock are the same in this study, the implementation of the clock scheme shown in Fig. 5 allows further development of HCgorilla.

Table 2 summarizes the design environment of HCgorilla chips. The technology of ROHM 0.18- μ m CMOS Kyotouniv Standard Cell Library is basically available for the development of HCgorilla chips. What we have devised is the script of power planning in Perl. Fig. 6 shows the die photo of HCgorilla.7 developed in this study.

3.2 Evaluation

The evaluation is based on simulation that employs netlist extracted from chip layout. The simulation is performed basically by using CAD tools shown in Table 2. Then, the factors to be evaluated are usual ones of microprocessor techniques, that is, speed, occupied area, and power consumption. Clock frequency is achieved from timing analysis by using Synopsys Design Compiler. Power dissipation is rated from power analysis by using a circuit simulator, Synopsys VCS, and a DUV (device under verification) simulator written in Verilog.

Table 2: Design Environment

<i>Software</i>	
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 - Primitime pts,vA-2007.12-SP3
<i>Language</i>	
Synthesis	VHDL
Simulation	Verilog - HDL
<i>Technology</i>	
ROHM 0.18 μ m CMOS Kyotouniv Standard Cell Library	

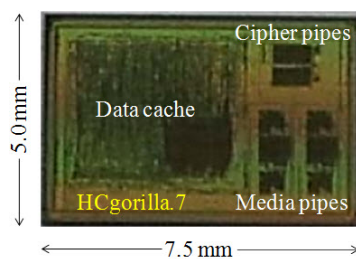


Fig. 6: Die photo of HCgorilla.7

Table 3 summarizes the basic evaluation of HCgorilla.7 compared with previous derivatives we developed. These employ the same 0.18- μ m CMOS standard cell technology. Overall aspects, chip parameters, and hardware specifications are also shown in this table. Comparing Fig. 6 and Table 3, a media pipe is made by 24,367 cells and occupies 2.5 mm². The hardware cost of a cipher pipe is 270 cells that occupy only 0.1 mm square. Then, the data cache employs 20,625 cells and occupies 18 mm².

Considering the effect of the clock schemes shown in Fig. 4 on the practical execution of programs, we give higher priority to gated clocking and wave-pipelining than scan test. Since wave-pipelining is carried out in combination stack access stage as is described in Section 2.2, the effect of gated clock on the waved-MFU and the stack access stage is evaluated more in detail.

Regarding clock speed, it is not affected by the clocking gate unit shown in Fig. 1 because it takes slower delays than other units and pipeline stages. On the other hand, the effect of wave-pipelining on clock speed is clear from Eq. (1), that is, wave-pipelining is basically effective for speed-up. In addition, the wave-pipelining is practically effective by employing the analogue design of buffers dedicated for the fine tuning of the waved MFU [14]. Since these buffers take longer delay than standard buffers provided by technology files, they are very useful to achieve faster f_{wave} as is described in Eq. (1).

Table 3: Specific Features HCgorilla Chips

		HCgorilla.3	HCgorilla.5	HCgorilla.6	HCgorilla.7
Design Rule		ROHM 0.18 μ m CMOS			
Wiring		1 polySi, 5 metal layers			
Area	Chip	5.0 mm \times 7.5 mm		2.5 \times 5-mm	5.0 mm \times 7.5 mm
	Core	4.28 mm \times 6.94 mm			4.28 mm \times 6.94 mm
Assembly	Pad	105		158	
	Signal	48		32	
	VDD/VSS	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 \times 32 word \times 2		16 bit \times 64 word \times 2	
Data cache		16 bit \times 128 word		16 bit \times 128 word \times 2	
Stack memory		16 bit \times 8 word \times 4		16 bit \times 16 word \times 8	
Register file		16 bit \times 64 word		16 bit \times 128 word	
RNG/cipher pipe		4 bit \times 1	6 bit \times 1		6 bit \times 2
No. of cores		2			
ILP degree		2	4		
Clock	Wave-pipeline	Introduced			
	Gated clock	Unintroduced		Introduced	
	Scan test	Unintroduced		Introduced	
	Frequency	330 MHz	200 MHz		
Throughput	Media pipe	0.17 GIPS			
	Cipher pipe	0.1-0.2 GOPS			
Transfer rate		160-320 Mbps			

As to area, delay buffers based on analogue design occupy less area than standard buffers. This also results in the superiority of wave-pipelining. On the other hand, the clocking gate unit shown in Fig. 1 occupies 0.036 mm² composed of 29 cells. This is only 1.4% of a media pipe. Thus, the combination of gated clock and wave-pipelining is preferable in view of occupied area.

The power consciousness of HCgorilla.7 is evalu-

ated from dynamic power and the effect of gated clocking is investigated as shown in Fig. 7. The DUV simulator simulates the media processing of the core that repeats a pair of arithmetic instructions 50 times. The instruction pair is issued alternately by one clock to the media pipes of a core. Thus, the number of repetitions by a core is doubled, that is, 50×2 times. Considering applications, for example, various processing of pixels, the repetition of 50×2 times might be not so large. Yet, it is enough from Fig. 7 to show the effect of gated clocking.

The one of two columns noted ‘‘Gated clock’’ in Fig. 7 shows the power consumption of the HCgorilla.7’s core excepting the cipher pipe. The other column noted ‘‘Nongated clocking’’ shows the power consumption of the HCgorilla.5’s core. Since HCgorilla.7 and HCgorilla.5 are implemented in the same chip area by using the same technology, the effectiveness of gated clocking is clear from Fig. 7.

Although it is not obvious from Fig. 7, power reduction of the waved MFU is 50-90% and that of stack access stage is 40-65%. Thus, the effect of gated clocking is large for the target stages of gated clocking, that is, the stack access stage and the waved MFU. 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 power in the stack access stage.

Let us briefly summarize the evaluation of a cipher pipe. The area related to the cipher pipes is 2.25 mm^2 , where 16,597 cells are included. The power dissipation of each cipher pipe is 30 mW. As for cipher strength, HCgorilla.7 is superior from Eq. (3). HCgorilla.7’s cipher pipe doubles the number of RNG. The effect of additional area and power due to duplicating RNG is negligibly small.

4. CONCLUSION

HCgorilla has been improved for the next generation ubiquitous environment with particular emphasis on the total optimum design of the clock scheme. This covers gated clock, scan path, and waved MFU carried out various design steps. The improved HCgorilla is implemented in a $0.18\text{-}\mu\text{m}$ CMOS standard cell chip. In view of overall aspects of clock speed, occupied area, power consumption, and hardware security, the HCgorilla.7 chip developed in this study is the best from specific features and quantitative evaluation.

The next step of this study is to further improve clocking as follows.

- Extension of the total clock scheme to the cipher pipe excepting scan test in view of security.
- Distinction between global clock and local clock.
- Introduction of PLL for the local clock.

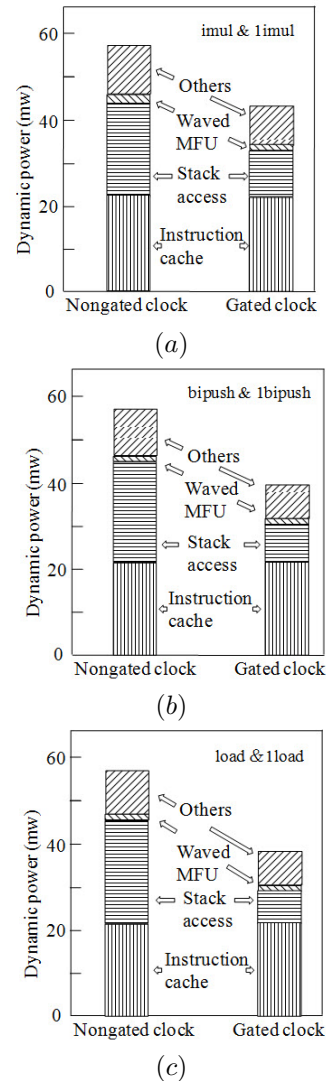


Fig. 7: Power dissipation of an HCgorilla.7’s core (a) Integer multiplication (b) Push (c) Load

- Analogue design of buffers for the fine tuning of the waved MFU [14].

5. ACKNOWLEDGEMENT

This work is supported in part by Grant-in-Aid for Scientific Research (C) (21500048) from Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

- [1] M. Fukase, R. Murakami, and T. Sato, ‘‘Design and Chip Implementation of an Instruction Scheduling Free Ubiquitous Processor,’’ *Proc. of ASP-DAC*, pp. 375-376, Jan. 2010.
- [2] 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.
- [3] M. J. Flynn, P. Hung, and K. W. Rudd, M.

- Fukase and T. Sato, "Deep-Submicron Microprocessor Design Issues," *IEEE MICRO*, pp. 11–22, Jul.-Aug., 1999.
- [4] M. Fukase, H. Uchiumi, K. Narita, T. Takaki, N. Mimura, K. Ichinohe, T. Sato, and A. Kurokawa, M. Fukase and T. Sato, "Development of a Next Generation Ubiquitous Processor Chip," *Proc. of ISPACS 2011*, pp. 225.1–225.4, Dec. 2011.
- [5] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, M. Fukase and T. Sato, "Wave-Pipelining: A Tutorial and Research Survey," *IEEE Trans. on VLSI Systems.*, vol. 6, no. 3, pp. 464–474, Sep., 1998.
- [6] Y-S. Su, D.-C. Wang, S.-C. Chang, and M. M.-Sadowska, M. Fukase and T. Sato, "Performance Optimization Using Variable-Latency Design Style," *IEEE Trans. on VLSI Syst.*, vol. 19, no. 10, pp. 1874–1883, Oct. 2011.
- [7] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, M. Fukase and T. Sato, "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.
- [8] T. Austin, D. Blaauw, T. Mudge, and K. Flautner, M. Fukase and T. Sato, "Making Typical Silicon Matter With Razor," *Computer Magazine*, vol. 37, no. 3, pp. 57–65, Mar. 2004.
- [9] Y. Lee, D.-K. Jeong, and T. Kim, M. Fukase and T. Sato, "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.
- [10] W. Shen, Y. Cai, X. Hong, and J. Hu, M. Fukase and T. Sato, "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.
- [11] T. Mudge, M. Fukase and T. Sato, "Power: A First-Class Architectural Design Constraint," *Computer Magazine*, vol. 34, no. 4, pp. 52–58, April, 2001.
- [12] J. Li, Q. Xu, Y. Hu, and X. Li, M. Fukase and T. Sato, "X-Filling for Simultaneous Shift- and Capture-Power Reduction in At-Speed Scan-Based Testing," *IEEE Trans. on VLSI Syst.*, vol. 18, no. 7, pp. 1081–1092, Jul. 2010.
- [13] W.-C. Kao, W.-S. Chuang, H.-T. Lin, J. C.-M. Li, and V. Manquinho, M. Fukase and T. Sato, "DFT and Minimum Leakage Pattern Generation for Static Power Reduction During Test and Burn-In," *IEEE Trans. on VLSI Syst.*, vol. 18, no. 3, pp. 392–400, Mar. 2010.
- [14] A. Kurokawa, T. Takaki, and M. Fukase, M. Fukase and T. Sato, "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 has been the Director of the Hirosaki University C&C Systems Center since 2004. He is also 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.



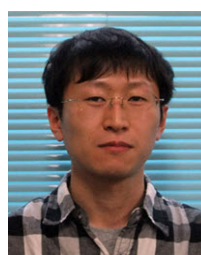
Kohei Ichinohe received the B.S. Degree in Electronics and Information Technology from Hirosaki University in 2012. He is currently working toward the M.S. degree at Hirosaki University.



Kazuki Narita received the B.S. Degree in Electronics and Information Technology from Hirosaki University in 2011. He is currently working toward the M.S. degree at Hirosaki University.



Tatsuya Takaki received the B.S. Degree in Electronics and Information Technology from Hirosaki University in 2011. He is currently working toward the M.S. degree at Hirosaki University.



Naomichi Mimura received the B.S. Degree in Electronics and Information Technology from Hirosaki University in 2011. He is currently working toward the M.S. degree at Hirosaki University.



Tomoaki Sato received the B.S. and M.S. degrees from Hirosaki University, Japan, in 1996 and 1998 respectively, and the Ph.D. degree from Tohoku University, Japan, in 2001. From 2001 to 2005, he was an Assistant Professor of Sapporo Gakuin University, Japan. Since 2005, he has been an Associate Professor of Hirosaki University. And, he has been a Visiting Associate Professor of the Open University of Japan since 2012. His research interests include VLSI Design, Computer Hardware, and Computer and Network Security.



Atsushi Kurokawa received the B.E. degree in Electrical Engineering from Seikei University, Tokyo, Japan, in 1986, and the Ph.D. degree in Information, Production, and Systems Engineering from Waseda University, Kitakyushu, Japan, in 2005. From 1986 to 2011, he was with SANYO Electric Company, Ltd. From 2002 to 2005, he went on loan to the Semiconductor Technology Academic Research Center (STARC). In 2008 and 2009, he was also a visiting associate professor in the Department of Electronic Engineering, Gunma University. He is currently an associate professor in the Graduate School of Science and Technology, Hirosaki University. His research interests include the design technology of integrated circuits in nano- and power-electronics. Dr. Kurokawa is a member of the IEEE, IEICE, IPSJ, and IEEJ.