ASIC Implementations of Five SHA-3 Finalists

Xu Guo, Meeta Srivastav, Sinan Huang, Dinesh Ganta, Michael B. Henry,
Leyla Nazhandali and Patrick Schaumont
Center for Embedded Systems for Critical Applications (CESCA)
Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061
Email: {xuguo, meeta, shuang86, diganta, mbh, leyla, schaum}@vt.edu

***Abstract*—Throughout the NIST SHA-3 competition, in relative order of importance, NIST considered the security, cost, and algorithm and implementation characteristics of a candidate [1]. Within the limited one-year security evaluation period for the five SHA-3 finalists, the cost and performance evaluation may put more weight in the selection of winner. This work contributes to the SHA-3 hardware evaluation by providing timely cost and performance results on the *first* SHA-3 ASIC in 0.13 μm IBM process using standard cell CMOS technology with measurements of all the five finalists using the latest Round 3 tweaks. This article describes the SHA-3 ASIC design from VLSI architecture implementation to the silicon realization.**

I. Introduction

A cryptographic hash function is a deterministic procedure which takes an arbitrary size message and returns a fixed length bit string named message digest. It has been used in many security applications, such as digital signatures and Message Authentication Codes (MACs). In response to the advances in the cryptanalysis of hash algorithms in recent years [2], National Institute of Standards and Technology (NIST) opened SHA-3 competition, which aims to select, in three phases, a successor for the mainstream SHA-2 hash algorithm. In December 2010, the SHA-3 competition entered into Phase III and five SHA-3 candidates were selected for further evaluation as SHA-3 finalists.

Although security is of the primary importance, the lack of systematic cryptanalysis of hash function makes it very hard to compare the security strength of different hash candidates. Within the limited one year period for the final round evaluation, the cost and performance of SHA-3 software and hardware implementations aspects are expected to put more weight in the selection of SHA-3 winner.

eBACS is a well known benchmarking environment, including a scripting environment and a performance database, for the evaluation of crypto-software [3]. Compared to such a benchmarking environment, benchmarking crypto-hardware is ad-hoc. There are several reasons why the same progress is not seen in the hardware design community. This is due, in part, to the large heterogeneity of the hardware design space, to the absence of standard metrics for cost and performance, and to the absence of standard interfaces [4].

To address the above issues in the domain of hardware ASIC evaluation, we designed a SHA-3 ASIC by following a fair and consistent SHA-3 hardware evaluation methodology [5], [6]. We started by defining a standard interface, and optimized the designs with a single metric, *Throughput-to-Area ratio*. Next, we developed an FPGA prototype that can provide a seamless transition into ASIC implementation. Finally, we designed an ASIC chip with all the five finalists using the latest Round 3 tweaks and SHA256 as a reference design.

The remainder of this paper is organized as follows. Section II gives an overview of the SHA-3 ASIC benchmark status. In Section III, the VLSI architecture of the SHA-3 ASIC will be described. The silicon implementation constraints, testing environment and measurements will be discussed in Section IV. Section V concludes the paper with some future work.

II. Related Work

The hardware evaluation of SHA-3 candidates has started shortly after the specifications of 51 algorithms submitted to the contest became available. More comprehensive efforts became feasible only after NIST's announcement of 14 candidates qualified to the second round of the competition in July 2009. Since then, several comprehensive studies in SHA-3 ASIC implementations have been reported [6]–[13]. Guo et al. [6] used a consistent and systematic approach to move the SHA-3 hardware benchmark process from the FPGA prototyping by Kobayashi et al. [14] to ASIC implementations based 130*nm* CMOS standard cell technology. Tillich et al. [8] presented the first ASIC post-synthesis results using 180*nm* CMOS standard cell technology with high throughput as the optimization goal and further provided post-layout results [7]. Henzen et al. [9] implemented several architectures in a 90*nm* CMOS standard cell technology, targeting high- and moderate-speed constraints separately, and presented a complete benchmark of post-layout results. Knezevic et al. [13] provided ASIC synthesis results in a 90*nm* CMOS standard cell technology as a comparison with their primary FPGA prototyping results. In December 2010, five candidates were selected for the last round of SHA-3 competition. These candidates then submitted the final specification of their algorithms in January 2011. The only comparison of the five candidates in ASIC implementations at this stage was provided by [5], [15] based on post-layout simulation.

The work described in this article presents implementation details and measurement results on the *first* SHA-3 test-chip, and as such it stands out among all the previous work summarized in Table I. Although Henzen et al. [12] reported the performance results of a compact BLAKE implementation

978-3-9810801-8-6/DATE12/©2012 EDAA

TABLE I
THE RELATED SHA-3 HARDWARE BENCHMARKING WORK IN ASICS.

	14 Second Round Candidates				5 Third Round Finalists
	Tillich [7], [8]	Guo [6]	Henzen [9]	Knezevic [13]	Guo [5], [15]
Technology Choices	180*nm* CMOS	130*nm* CMOS	90*nm* CMOS	90*nm* CMOS	130*nm* CMOS
Hardware Interface	Assume infinite bandwidth interface	Defined standard 'handshake' interface	Assume infinite bandwidth interface	Defined standard 'handshake' interface	Defined standard 'handshake' interface
Chosen Metrics	Area, Throughput	Area, Throughput, Power, Energy	Area, Throughput, Energy	Area, Throughput, Power, Energy	Area, Throughput, Power, Energy
Design Flow	Post-layout/synthesis	Post-layout	Post-layout	Post-synthesis	Post-layout

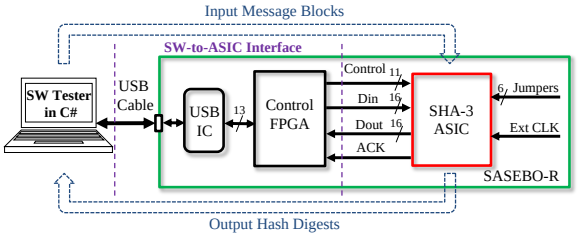

Fig. 1. The SASEBO-R platform for SHA-3 ASIC testing.

based on ASIC measurements, as the BLAKE hash designers they only focused on the BLAKE ASIC characterization.

III. VLSI ARCHITECTURE OF SHA-3 ASIC

This section covers the VLSI architecture design of the SHA-3 ASIC as shown in Fig. 2 under the physical constraints imposed by the selected SASEBO-R board [16]. As an open platform the SASEBO-R board was originally developed for side-channel analysis. Hence, a potential research area for SHA-3 ASIC is side-channel analysis of SHA-3 candidates. In our experiments, we used the SASEBO-R board for a more obvious application, namely the measurement of power dissipation of the SHA-3 candidates mapped to ASICs.

The experimental environment for SHA-3 ASIC contains a PC and a SASEBO-R board as shown in Fig. 1. A SASEBO-R board contains a control FPGA, which supports the interfacing activities with the PC and SHA-3 ASIC.

A. Clock Management

For signal integrity issues associated with the on-board wire transfers, the ASIC interface clock used to synchronize all the data and control signals from/to the control FPGA should only run at a relatively low frequency. The SHA-3 ASIC chip can operate at 250 *MHz* for some candidates, so an additional stable fast clock is required, which will be gated and shared by all the hash modules.

- *Clock Generation.* For high frequency testing purpose, an on-chip clock generation module is integrated. We used the custom-cell design approach to integrate a ring oscillator (RO) based voltage-controlled oscillator (VCO) into the chip. VCO is an electronic oscillator designed to be controlled in oscillation frequency by a DC voltage input(i.e. PBIAS port in SHA-3 chip). In addition, we also integrated three standard-cell ROs to provide fixed high frequency clocks.

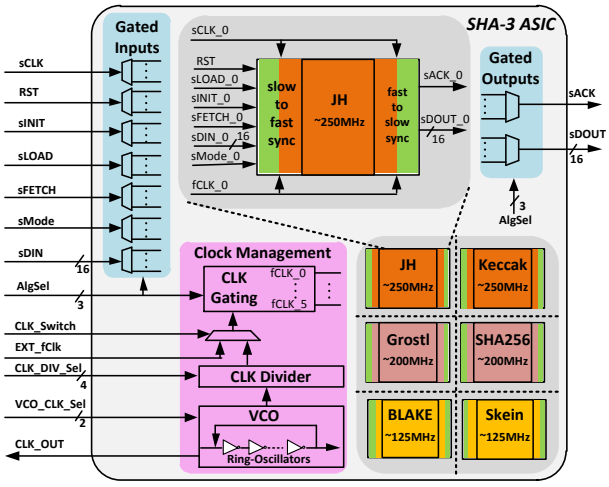

Fig. 2. The block diagram of SHA-3 ASIC.

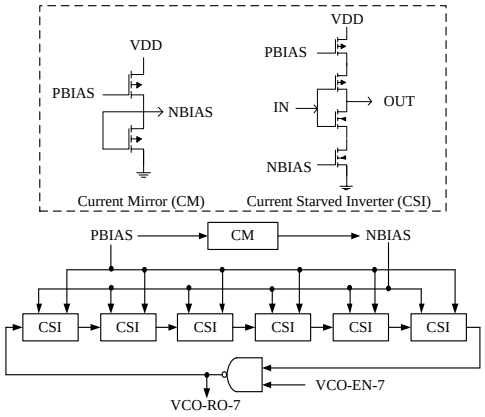

Fig. 3. The diagram of a 7-stage VCO-RO clock design.

The VCO takes four input ports PBIAS, VCO-EN-7,9,11, and three output frequencies, VCO-RO-7,9,11. The PBIAS voltage can be varied to produce a range of clock frequencies. The voltage can be varied from 0V to 0.8V. The 'EN' is used to turn on/off the clock outputs of VCO. VCO-RO-7,9,11 are the three frequencies produced by the block for any particular PBIAS voltage. VCO-RO-7 is the clock from a 7-stage RO.

We did extensive Spice simulations to determine the optimum stage length and appropriate device dimensions. We performed simulations at different process corners, and we se-

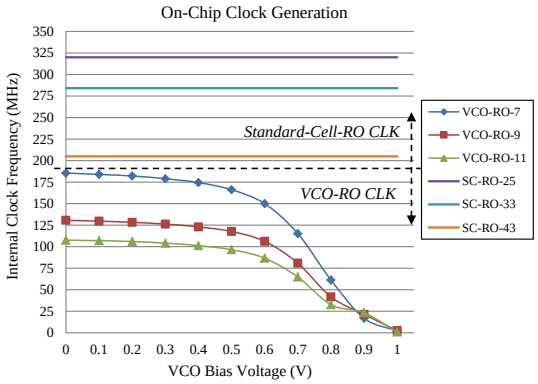

Fig. 4. The range of on-chip generated clock frequencies.

lected VCO configurations to meet the frequency requirements even in the worst process corners. Simulations have shown that the ROs with stage lengths of 7, 9, and 11 are required to generate the range of frequencies. As shown in Fig. 3, the Ring Oscillator (RO) of the VCO is realized using NAND2, Current Starved Inverter (CSI), and Current Mirror (CM) custom cells. We designed the custom cells using Cadence Virtuoso, and we performed the Design Rule Check and Layout vs. Schematic with Assura. We used Synopsys library compiler and Milkyway to generate libraries for synthesis and place and route. We used Synopsys Design Compiler (C-2009.06-SP3) and IC Compiler (C-2009.06-SP5) for synthesis and place & route, respectively. The RC parasitics have been extracted for the post-layout design and spice simulations have been performed for verification.

The standard cell RO based clock generation together with the VCO clocks and a standard cell implemented clock divider can support a wide range of clock frequencies to fill our need for performance testing. Fig. 4 shows averaged measurements of on-chip clock speed for a batch of 10 fabricated chips.

• *Clock Configurations.* The on-chip generated clocks are also MUXed with the external fast clock input and can be configured through dedicated ports. The external fast clock can be fed through a SMA connector from a signal generator or it can be provided through the control FPGA. Clock gating is implemented to guarantee that only one hash module is enabled at a time.

B. Chip Interface

• *Standard Hash Interface.* The chip interface shown in Fig. 2 adopted the standard hash interface by Chen et al. [17] and extended it to add mode selection and dual-clock support.

• *Clock Domain Crossing (CDC) Synchronizer.* There are two clock domains in our chip: the slow one is for the interfacing logic and the fast one is for hash modules. In order to avoid complex synchronizer designs based on asynchronous FIFOs or feedback synchronization and alleviate the burden of the backend process to deal with the two clock domains, we simplified the synchronizer design by making a reasonable assumption that the internal hash clock working frequency is always at least two times faster than the slow interface clock. As a result, the slow interface clock is treated as a plain control signal and the whole chip only has one single fast clock. To synchronize the slow interface signals to the fast hash core, a synchronizer with 2-stage flip-flops is used. The fast-to-slow synchronizer for LOAD/FETCH acknowledge signal is designed based on a 4-stage FSM. As shown in Fig. 5, the f2sACK signal high will last for PROCESS and WAIT states period in order to be captured by the rising edge of sCLK. A handshake signal, f2sACK$_{fb}$, is sent back to the control FSM of hash core to indicate a successful LOAD/FETCH. Within this approach we extended the standard hash interface [17] of each candidate to integrate this low-cost and simplified synchronizer, and the final reported layout area for each candidate will also include the overhead of this extended hash interface.

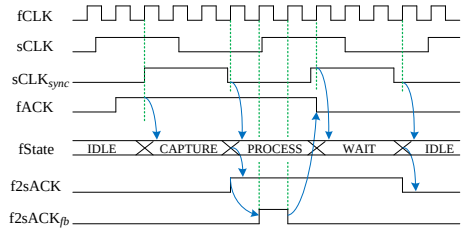

Fig. 5. The timing of slow-to-fast synchronizer design.

C. Power/Energy Approximation

In order to obtain accurate power profiles of each SHA-3 finalist, the ideal solution is to have separate power networks for each hash module. Since SASEBO-R test platform only supports a single power network for the chip, we use gated logics to force the inactive hash modules to enter into idle state by pulling down the clock and all the control and data signals. We estimated the static power dissipation of each module based on the area ratio and the standby power measured for the full chip. Note also that in 130nm, the static power dissipation is typically only a small portion of the complete power dissipation. In order to justify the power measurements, we have also compared them with the results from post-layout simulation and they closely match with each other as shown in Table III.

D. SHA-3 Finalists Implementations

Although we have fixed our optimization target, *Throughput-to-Area ratio*, to fully understand each SHA-3 candidate by its specification and reference C codes and optimize its hardware implementation to achieve the goal is far from trivial. We had looked into several public available reference implementations [18]–[20] and optimized them for our system architecture.

Table II summarizes the major implementation aspects of each SHA-3 finalist. The design decisions are made to achieve the primary optimization for *Throughput-to-Area ratio*. For details on the SHA-3 candidates please refer to the related specification documents on the NIST SHA-3 web site [21].

TABLE II
THE SUMMARY OF DESIGN SPECIFICATIONS OF SHA-3 FINALISTS

Algorithm	Implementation Descriptions
Blake-256	4 parallel **G** functions; 1-stage pipeline in permutation
Grøstl-256	Parallel **P** and **Q** with 128 GF-based AES SBoxes
JH-256	SBoxes **S0** and **S1** are implemented in LUT
Keccak-256	One clock cycle per round
Skein512-256	Unrolled 4 Threefish rounds

Implementation details for each hash module can be consulted in [5].

IV. SILICON IMPLEMENTATION OF SHA-3 ASIC

This section will first compare the synthesis and layout constraints of the SHA-3 ASIC and then discuss the ASIC measurement results.

A. Timing Constraints

The timing constraints are selected to optimize *Throughput-to-Area ratio*, using the methodology described in [10]. Although all the RTL designs are optimized for *Throughput-to-Area ratio*, depending on the different scenarios we may put different constraints during the synthesis and layout which may greatly affect the quality of the ASIC results. For synthesis, we evaluate four design points for every implementation.

MinArea: A minimum-area design will minimize the use of logic resources (GEs) at the expense of performance.

MaxSpeed: A maximum-speed design will minimize the computational delay of the design, at the expense of area.

TradeOff0: The first trade-off point is chosen to have a computational delay which is two-thirds between the MinArea and MaxSpeed design points.

TradeOff1: The second trade-off point is chosen to have a computational delay which is five-sixths between the MinArea and MaxSpeed design points.

The TradeOff points are chosen to investigate how the relationship (speed, area) evolves when a design gradually moves from the *MinArea* design point to the *MaxSpeed* design point. As shown in the Fig. 6(a), all the dash lines connect to the points with highest *Throughput-to-Area ratio*, which are always the *MaxSpeed* point except for Grøstl whose optimal point is *Tradeoff1*.

After place&route, the routing delay will lower down the maximum frequency, and together with the added routing area the *Throughput-to-Area ratio* will be reduced. As shown in the equation below, the *weight* as the degradation factor will be always larger than one for post-layout.

$$[f_{max}/area]_{layout} = \frac{[f_{max}/area]_{synthesis}}{weight} \quad (1)$$

Ideally, we may relax the timing constraints of each hash module and obtain a uniform *weight* for all the candidates before and after layout. However, in practice by relaxing the timing constraint the degree of area decrease is not uniform for different candidates. This can also be observed

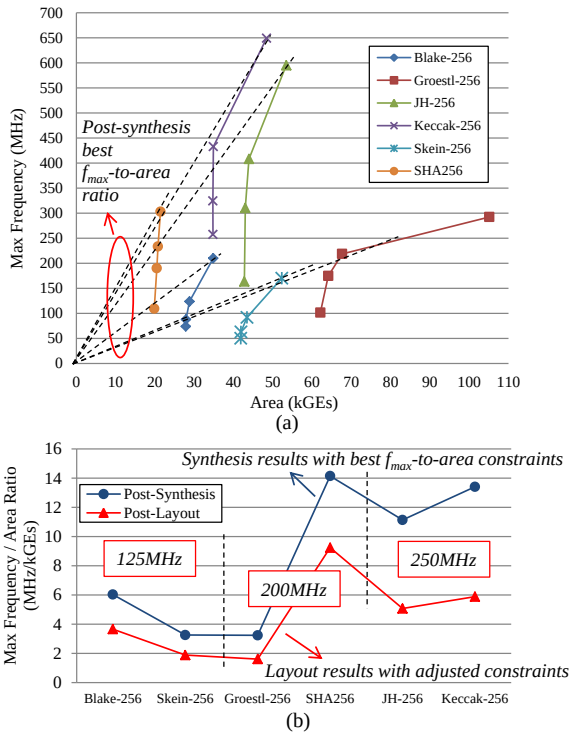

Fig. 6. (a) Synthesis exploration of each SHA-3 candidate; (b) Weight factors to determine the final ASIC layout constraints.

in synthesis results in Fig. 6(a). After we tried several rounds, we categorized the five SHA-3 candidates plus the SHA256 based on their achievable frequencies after layout into three groups: 250 *MHz* (JH and Keccak), 200 *MHz* (Grøstl and SHA256), and 125 *MHz* (BLAKE and Skein). As shown in Fig. 6(b), comparing the individual synthesis results with the final layout results of all candidates on the same die, we put larger *weight* to the high speed designs, JH and Keccak, with an average *weight* of 1.9 for all the designs. For high frequency designs after synthesis, more weight has been added for layout constraints to avoid explosion of the overall chip area.

The resulting layout, shown in Fig. 7, uses the IBM MOSIS $130nm$ CMR8SF-RVT standard cell library. The chip core area is 1.656 mm × 1.656 mm and overall chip die size is 5 mm^2 including pad cells. Out of seven metal layers, five metal layers are used for signal and clock routings and the top two layers are used for power and ground. The overall chip core area utilization ratio after layout is 73%. The chip is packaged with 160-pin QFP to be compatible with the SASEBO-R board.

B. ASIC Measurement Results and Analysis

From the ASIC measurement results shown in Table III, all the five SHA-3 candidates meet the target maximum frequency, and have a larger area but higher throughput than the reference SHA256; the maximum throughput of Grøstl and Keccak can almost reach 10 *Gbps*; Keccak is the best in hardware efficiency and energy efficiency; JH is the most power efficient SHA-3, closely followed by Keccak and BLAKE,

TABLE III
ASIC CHARACTERIZATION OF THE SHA-3 ASIC CHIP IN IBM MOSIS 130nm CMOS TECHNOLOGY WITH CMR8SF-RVT STANDARD CELL LIBRARY

	Block Size [*bits*]	Core Lat. [*cycles*]	Area[a] [*kGEs*]	Max Freq. [*MHz*]	Tp [*Gbps*]	Tp/Area [*kbps/GE*]	Power[b] [*mW*]	Energy[b] [*mJ/Gbits*]	Power[c] [*mW*]	Energy[c] [*mJ/Gbits*]
BLAKE-256	512	30	34.15	125	2.13	62.47	21.33	25.00	19.77	23.17
Grøstl-256	512	11	124.34	200	9.31	74.87	78.42	33.70	139.29	59.85
JH-256	512	42	49.29	250	3.05	61.83	12.57	20.63	13.01	21.35
Keccak-256	1024	24	42.49	250	10.67	251.05	19.12	8.96	19.78	9.27
Skein512-256	512	21	66.36	125	3.05	45.93	31.74	26.04	51.09	41.91
SHA256	512	68	21.67	200	1.51	69.54	5.18	13.76	5.05	13.42

a: the Gate Equivalent count is calculated by dividing the post-layout die area by the area of a NAND2XLTF (5.76 μm^2).
b: numbers are based on chip measurements of SHA-3 ASIC with slow chip interface clock at 1.5*MHz* and fast hash core clock at 50*MHz*.
c: numbers are based on post-layout simulation of SHA-3 ASIC with slow chip interface clock at 1.5*MHz* and fast hash core clock at 50*MHz*.
Note:
1: All five SHA-3 candidates are implemented with NIST SHA-3 Round 3 Specifications by January, 2011.
2: Each design's static power is estimated by multiplying the whole chip static power, 1.92 *mW*, with the area ratio of each design.

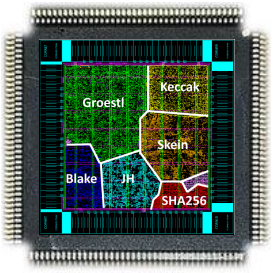

Fig. 7. The SHA-3 ASIC layout and 160-pin QFP package.

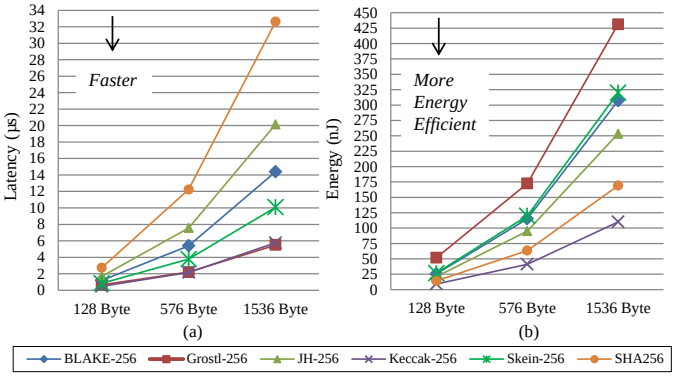

Fig. 8. (a) The latency curve and for different packet sizes assuming ideal interface; (b) The energy curve for different packet sizes assuming ideal interface (the energy numbers are estimated based on average power measurements at slow and fast clock of 1.5 *MHz* and 50 *MHz*, respectively).

but still less efficient than SHA256. The power consumption results are measured at fixed slow interface clock frequency of 1.5*MHz* and fast hash core clock at 50*MHz*. Both of the clocks are provided by the control FPGA through on-board wire connections to the SHA-3 ASIC, so relatively slow interface clock frequencies are chosen for stability. The latency and energy efficiency of SHA-3 finalists should be evaluated with different message lengths due to the different overheads in both of the hash initialization and finalization steps. We examined all the SHA-3 finalists with very short message and message lengths around the most common used Internet packet sizes (i.e. 576 and 1500 bytes) [22]. We compared the SHA-3 finalists without considering the interface overhead, which can better examine the characteristics of SHA-3 candidates themselves. As shown in Fig. 8, Keccak and Grøstl are the fastest for all the three cases; Keccak is also the most energy efficient finalist. Fig. 8 also shows that the rankings of candidates almost do not change based on message length, although their differences grow at longer message lengths. The overhead of finalization step in Grøstl makes it slightly slower than Keccak when hashing short messages, but for the case of hashing long messages Grøstl becomes faster than Keccak. Note that we have listed the power/energy results for both of the ASIC measurements and post-layout simulation results for comparisons. Grøstl and Skein show relatively larger differences in these two cases; however, there might be sources of inaccuracy in both power models and measurement tools which are difficult to justify. We consider these power variations are less important issues since the order of power/energy efficiency for different SHA-3 finalists does not change as shown in Table III.

V. CONCLUSIONS AND FUTURE WORK

This article reported the *first* ASIC measurement results for hardware evaluation of SHA-3 finalists based on the Round 3 specifications published online in January, 2011. Moreover, as our SHA-3 ASIC has been designed to be compatible with SASEBO-R board, which is an open platform for side-channel attack analysis and widely distributed among the cryptographic hardware community. We plan to distribute SHA-3 chips to other research groups and establish a public side-channel evaluation process on the SHA-3 ASIC implementations. As for power measurement setup in power analysis attacks, sample voltage drop (proportional to the power consumption) traces identifying the interested hash core operating period are shown in Fig. 9. The software tester running on a host PC, the control FPGA hardware on SASEBO-R, and the SHA-3 ASIC are all publicly available [19].

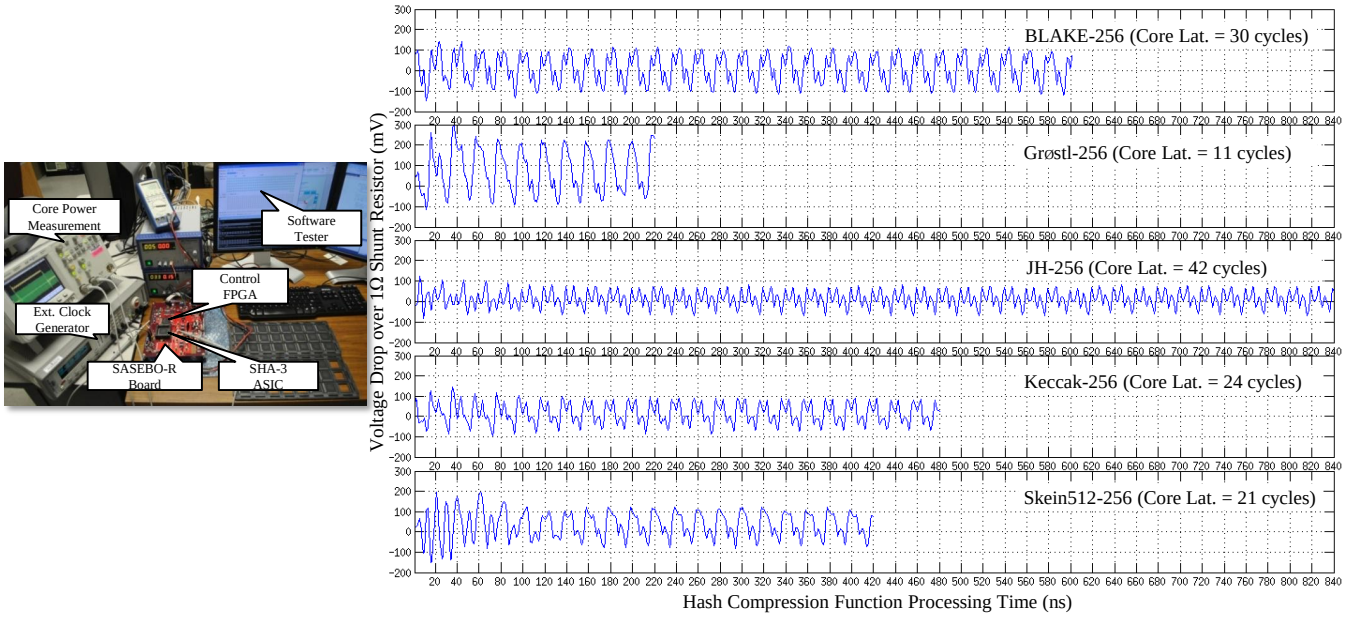

Fig. 9. The SHA-3 ASIC testing environment and sample voltage drop (power) traces measured at slow and fast clock of 1.5 *MHz* and 50 *MHz*.

ACKNOWLEDGMENT

The effort reported in this paper was supported by a NIST Measurement, Science and Engineering Grant ("Environment for Fair and Comprehensive Performance Evaluation of Cryptographic Hardware and Software"). The authors would like to thank AIST [16], Japan for their hardware support.

REFERENCES

[1] M. S. Turan, R. Perlner, L. E. Bassham, W. Burr, D. Chang, S. jen Chang, M. J. Dworkin, J. M. Kelsey, S. Paul, and R. Peralta, "Status report on the second round of the sha-3 cryptograhic hash algorithm competition," NIST Interagency Report 7764, February 2011, http://csrc.nist.gov/publications/nistir/ir7764/nistir-7764.pdf.

[2] X. Wang, Y. L. Yin, and H. Yu, "Collision search attacks on sha-1," 2005, http://www.c4i.org/erehwon/shanote.pdf.

[3] D. Bernstein and T. Lange, "eBACS: ECRYPT Benchmarking of Cryptographic Systems," May 2011, http://bench.cr.yp.to.

[4] F. K. Gurkaynak, "50 ways to report the performance of your circuits," 7th International Workshop on the state of the art in cryptology and new challenges ahead, May 2011.

[5] X. Guo, M. Srivastav, S. Huang, D. Ganta, M. B. Henry, L. Nazhandali, and P. Schaumont, "Silicon Implementation of SHA-3 Finalists: BLAKE, Grøstl, JH, Keccak and Skein," in *ECRYPT II Hash Workshop 2011*, May 2011.

[6] X. Guo, S. Huang, L. Nazhandali, and P. Schaumont, "Fair and Comprehensive Performance Evaluation of 14 Second Round SHA-3 ASIC Implementations," in *The Second SHA-3 Candidate Conference*, 2010.

[7] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, and A. Szekely, "Uniform Evaluation of Hardware Implementations of the Round-Two SHA-3 Candidates," in *The Second SHA-3 Candidate Conference*, August 2010.

[8] ——, "High-Speed Hardware Implementations of BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, and Skein," Cryptology ePrint Archive, Report 2009/510, 2009, http://eprint.iacr.org/2009/510.

[9] L. Henzen, P. Gendotti, P. Guillet, E. Pargaetzi, M. Zoller, and F. Gürkaynak, "Developing a Hardware Evaluation Method for SHA-3 Candidates," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, ser. LNCS, 2010, vol. 6225, pp. 248–263.

[10] X. Guo, S. Huang, L. Nazhandali, and P. Schaumont, "On The Impact of Target Technology in SHA-3 Hardware Benchmark Rankings," Cryptology ePrint Archive, Report 2010/536, 2010, http://eprint.iacr.org/2010/536.

[11] A. Namin and M. Hasan, "Hardware implementation of the compression function for selected SHA-3 candidates," CACR 2009-28, July 2009.

[12] L. Henzen, J.-P. Aumasson, W. Meier, and R. C.-W. Phan, "VLSI Characterization of the Cryptographic Hash Function BLAKE," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 19, no. 10, pp. 1746 –1754, 2011.

[13] M. Knezevic, K. Kobayashi, J. Ikegami, S. Matsuo, A. Satoh, U. Kocabas, J. Fan, T. Katashita, T. Sugawara, K. Sakiyama, I. Verbauwhede, K. Ohta, N. Homma, and T. Aoki, "Fair and consistent hardware evaluation of fourteen round two sha-3 candidates," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –13, 2011.

[14] K. Kobayashi, J. Ikegami, M. Knezevic, X. Guo, S. Matsuo, S. Huang, L. Nazhandali, U. Kocabas, J. Fan, A. Satoh, I. Verbauwhede, K. Sakiyama, and K. Ohta, "Prototyping platform for performance evaluation of SHA-3 candidates," in *Hardware-Oriented Security and Trust (HOST), IEEE International Symposium on*, 2010, pp. 60 –63.

[15] X. Guo, M. Srivistav, S. Huang, D. Ganta, M. B. Henry, L. Nazhandali, and P. Schaumont, "Pre-silicon Characterization of NIST SHA-3 Final Round Candidates," in *14th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2011)*, 2011.

[16] AIST-RCIS, "Side-channel attack standard evaluation board," May 2011, http://staff.aist.go.jp/akashi.satoh/SASEBO/en/index.html.

[17] Z. Chen, S. Morozov, and P. Schaumont, "A Hardware Interface for Hashing Algorithms," Cryptology ePrint Archive, Report 2008/529, 2008, http://eprint.iacr.org/2008/529.

[18] AIST-RCIS, "SHA-3 hardware project," May 2011, http://www.rcis.aist.go.jp/special/SASEBO/SHA3-en.html.

[19] X. Guo, M. Srivastav, S. Huang, D. Ganta, M. B. Henry, L. Nazhandali, and P. Schaumont, "Performance Evaluation of Cryptographic Hardware and Software – Performance Evaluation of SHA-3 Candidates in ASIC and FPGA," May 2011, http://rijndael.ece.vt.edu/sha3/.

[20] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "The Keccak sponge function family – Updated VHDL package," May 2011, http://keccak.noekeon.org/VHDL_3.0.html.

[21] NIST, "Third (final) round candidates," December 2011, http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/submissions_rnd3.html.

[22] CAIDA, "Packet size distribution comparison between internet links in 1998 and 2008," July 2011, http://www.caida.org/research/traffic-analysis/pkt_size_distribution/graphs.xml.