Design and Benchmarking of an ASIC with Five SHA-3 Finalist Candidates

Meeta Srivastav, Xu Guo, 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

(meeta, xuguo, shuang86, diganta, mbh, leyla, schaum)@vt.edu

Preprint submitted to Microprocessors and Microsystems: Embedded Hardware Design, April 20, 2012

Design and Benchmarking of an ASIC with Five SHA-3 Finalist Candidates

Meeta Srivastav, Xu Guo, 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

(meeta, xuguo, shuang86, diganta, mbh, leyla, schaum)@vt.edu

Abstract

This contribution describes our efforts in the design of a $130nm$ CMOS ASIC that implements Skein, BLAKE, JH, Grøstl, and Keccak, the five candidates selected by NIST in the third round SHA-3 competition. The objective of the ASIC is to accurately measure the performance and power dissipation of each candidate when implemented as an ASIC. The design of this ASIC, and its optimization for benchmarking, creates unique problems, related to the integration of 5 heterogeneous architectures on a single chip. We implemented each algorithm in a separate clock region, and we integrated an on-chip clock generator with flexible testing modes. The chip is further designed to be compatible with SASEBO-R board, a power-analysis and side-channel analysis environment. We report the design flow and test results of the chip, including area, performance and shmoo plot. Furthermore, we compare our ASIC benchmark with an equivalent FPGA benchmark.

Keywords: Application Specific integrated circuit (ASIC), field programmable gate array (FPGA), hash algorithm (HASH), SHA-3 competition

1. Introduction

The SHA-3 competition organized by NIST aims to select, in three phases, a successor for the mainstream SHA-2 hash algorithms in use today. By the

completion of Phase II in December 2010, 5 out of the 14 second round candidates were identified for further evaluation as SHA-3 finalists. For each round in this competition NIST wants to evaluate algorithms [19] and find the best performing algorithm across a large set of computers/architectures. So, their benchmarking process studies how a single algorithm behaves across a broad range of architectures (ASIC being one of them). The winner will be announced by NIST in spring 2012.

NIST recommended benchmarking of both software and hardware platforms. Although the underlying goal of doing benchmarking for both software and hardware is the same, the methodology used is very different and unique. Our effort and contribution to this competition, is to develop an environment for doing un-biased and comprehensive evaluation of SHA-3 candidates on hardware platform. Hardware benchmarking is an important aspect as it evaluates the algorithm based on area, performance and power. There are two primary hardware benchmarking targets: FPGA and ASIC implementations. FPGA benchmarking is very similar to software benchmarking. Because an FPGA can be reprogrammed, each SHA-3 algorithm can be tested in isolation from the others. ASIC benchmarking, on the other hand, requires an expensive and labor intensive tape-out process. Therefore, we need to design all SHA-3 candidates in a single chip, and their low-level implementation (place-and-route) is a shared effort for all candidates at the same time. Since ASICs still cover a significant portion of the hardware design market, we cannot ignore ASIC benchmarking. At the same time, ASIC implementation generally has better performance, smaller die area, and lower power consumption than FPGA.

To evaluate each algorithm on ASIC, we have designed, functionally verified and successfully fabricated chip with SHA-3 finalists. The chip is compatible with SASEBO-R board, which is widely used among cryptographic research community. It provides early access to SHA-3 ASIC hardware. We have open-sourced RTL designs and are also providing other teams with copies of this chip. However, to achieve benchmarking in ASIC in a timely manner and to ensure fairness, we have faced several challenges at different phases, which includes design, implementation and testing. In this article, we will discuss this in further details and present our findings as measured on hardware.

To summarize, key contributions of this article are as follows.

- First, we propose a platform, methodology and evaluation criteria for

	14 Second Round Candidates				5 Third Round Candidates
	Tilinch [18][17]	Guo [7]	Henzen [12]	Knezevic [14]	Guo [10]
Technology Node	180nm CMOS	130nm CMOS	90nm CMOS	90nm CMOS	130nm 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	Power, Energy	Area, Throughput, Power, Energy
Design Result	Post-layout	Post-layout	Post-layout	Post-synthesis	Post-layout
Hardware Testing	No	No	No	No	No

Table 1: The related SHA-3 hardware benchmarking work in ASICs.

a comprehensive comparison between five finalists in FPGA and ASIC platforms.

- Second, we present design details and challenges faced in ensuring fairness while benchmarking in ASIC.

- Third, we present the measurement, trends and ranking of these candidates across both FPGA and ASIC platforms.

The rest of the article is structured as follows. In Section 2, we will discuss related work towards ASIC benchmarking. Our methodology towards prototyping is described in Section 3. In Section 4, we present evaluation metrics for these candidates. Section 5 describes the implementation details of ASIC chip. In Section 6, we will analyze the ASIC measurement results and conclude our work in Section 7.

2. 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 for the second round of the competition in July 2009. Since then, several comprehensive studies for FPGA [6, 13, 16] and ASIC implementations [18, 17, 12, 7, 8, 11, 14] have been reported. Guo et al. [7] used a consistent and systematic approach to move the SHA-3 hardware benchmark process from the FPGA prototyping by [15] to ASIC implementations using $130nm$ CMOS standard cell technology. Tillich et

al. [17] presented the first ASIC post-synthesis results using $180nm$ CMOS standard cell technology with high throughput as the optimization goal and further provided post-layout results [18]. Henzen et al. [12] implemented several architectures in a $90nm$ CMOS standard cell technology, targeting high- and moderate-speed constraints separately, and presenting a complete benchmark of post-layout results. Knezevic et al. [14] provided ASIC synthesis results in a $90nm$ 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 [10] based on post-layout simulation. Although Henzen et al. [11] reported the performance results of a compact BLAKE implementation based on ASIC measurements. However, as the BLAKE hash designers they only focused on the BLAKE ASIC characterization. In this article we present implementation details and results that are measured on hardware chip. This is likely the first SHA-3 test chip with five finalists, and as such stands out among all the previous work summarized in Table 1.

3. Methodology

In this section we describe the overall design environment that we have built for both FPGA and ASIC prototyping.

The Side-channel Attack Standard Evaluation Board (SASEBO) [2] is a board specifically designed to develop standard evaluation schemes to secure the cryptographic module against physical attacks. The experimental environment for FPGA prototyping was done using SASEBO-GII board as shown in Fig 1. A SASEBO-GII board contains two FPGAs: a control FPGA, which provides the interfacing activities with a PC, and a cryptographic FPGA, which contains the hashing candidate. We use the same environment for ASIC benchmarking. ASIC implementation is tested using SASEBO-R board. SASEBO-R board contains a socket, which is used to mount our SHA-3 chip and a control FPGA, which is used to provide interface logic. We test the functionality of each candidate by sending message blocks from the host PC to the SASEBO board and reading the digest generated by the SHA-3 ASIC once it is ready. The digest is then compared with

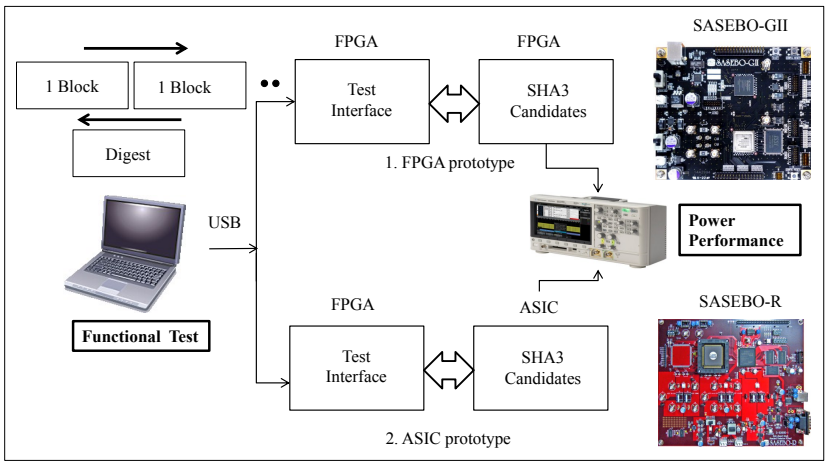

Figure 1: Environment for FPGA and ASIC prototyping.

a pre-computed digest by a software testbench running on PC. Power and performance analysis is also performed for both FPGA and ASIC platforms.

FPGA prototyping was done in an earlier phase of our project [7], and in this article we will not cover any details on FPGA prototyping. However, we want to emphasize the advantages of using this setup. First, we now have unique capability of analyzing the design and implementation characteristics for all candidates in both FPGA and ASIC platforms. Second, by using the same test interface we accelerate the process of building and testing of ASIC chip.

4. Evaluation Metrics

In this section, we discuss the various metrics based on which we evaluate the five candidates. Common metrics include area, maximum frequency, maximum throughput and power/energy consumption.

4.1. Area

We use the circuit area of each SHA-3 candidate including both, the interface and hash core after layout. The area is reported in kilo-gate-equivalents (kGE), where a gate equivalent corresponds to the area of a standard NAND2 gate in the standard-cell library. We divide the reported layout area with

unit in mm^2 by the area of an NAND2 gate for conversion from the absolute circuit area to kGE.

4.2. Throughput(Tp)

The time required to hash a message consists of four parts: the latency for loading one block of message, L_{in}, the hash core latency, L_{core}, the latency for finalization step, L_{final}, and the latency for outputting the message digest, L_{out}. For short message hashing, all these four latencies are important performance factors. The total latency is frequently used to characterize the short message hashing speed instead of throughput. In the case of hashing a long message, L_{final} and L_{out} can be neglected. Since L_{in} is dependent on the system I/O throughput which may vary in different contexts, here we report the throughput Tp of the hash core function as follows:

$$Tp = \frac{BlockSize * MaxFreq}{Latency} \tag{1}$$

4.3. Throughput-to-Area (T/A)

The Throughput-to-Area Ratio measures the hardware efficiency, where the Throughput is the above defined Tp and the Area is the layout circuit area expressed in terms of kGE. We use T/A as optimization target, which was first proposed by Gaj et al. [11], and later appeared in NIST status report on the Second Round of the SHA-3 Competition as a hardware evaluation criterion. One of the advantages of throughput over area is that this criterion will avoid high throughput designs get an unfair advantage.

4.4. Power/Energy

The power is measured with a fixed achievable clock frequency based on the average power during hashing of long messages, and the capture period is only for the core hashing operations (e.g. round function for each message block). The energy metric is expressed as energy per bit of the input message block compressed by the hash core function.

5. *Design of SHA-3 ASIC*

In this section, we present details of the overall design process of an ASIC chip. This process can be broadly categorized in three main phases, design, implementation and testing. In the design phase section, we will present RTL

Table 2: List of ASIC Library and EDA tools.

Technology	130nm
Cell library	'CMR8SF-RVT ARM'
Transistor Model	IBM 8 Metal
Synthesis	Design Compiler C2009.06 SP3
Timing measurement	Prime Time
Power measurement	Prime Time, VCS
Place and Route	ICC Compiler C-2009.06-SP5
DRC and LVS	Hercules and Virtuoso

details of SHA-3 candidates and chip design details at architectural level. In the implementation phase section, we will present design details at layout level. In the testing phase section, we will present details on testing strategy. For each of these phases, we will also emphasize on the challenges faced to ensure fairness in evaluation. Table 2 summarizes the list of tools and library used for ASIC implementation.

5.1. Design Phase: RTL of SHA-3 Candidates

There are three strategies used for SHA-3 hardware evaluation, namely fully-autonomous, external-memory and core-functionality [1]. In this work, we have designed all the 5 finalists with a fully-autonomous architecture. In this architecture, one transfers message data to a hash function over multiple clock cycles until a complete message block is provided. The hash module buffers a complete message block locally, before initiating the hash operation. Therefore, this architecture can work autonomously, and the resulting hash module is well suited for hardware IP for system-on-chip integration. For hardware architectures, we have looked into several publicly available reference implementations [9, 3, 4] and optimized them for our system architecture. In earlier work [10], we provided design details for each candidate's hardware architecture. This article is an extension of [10], and emphasizes chip implementation, as well as measurements of ASIC characteristics, such as performance and power dissipation.

5.2. Design Phase: Architecture Level Details

In this section, we will discuss architecture details of the chip. This includes, standard interface, synchronizer modules, clock management unit

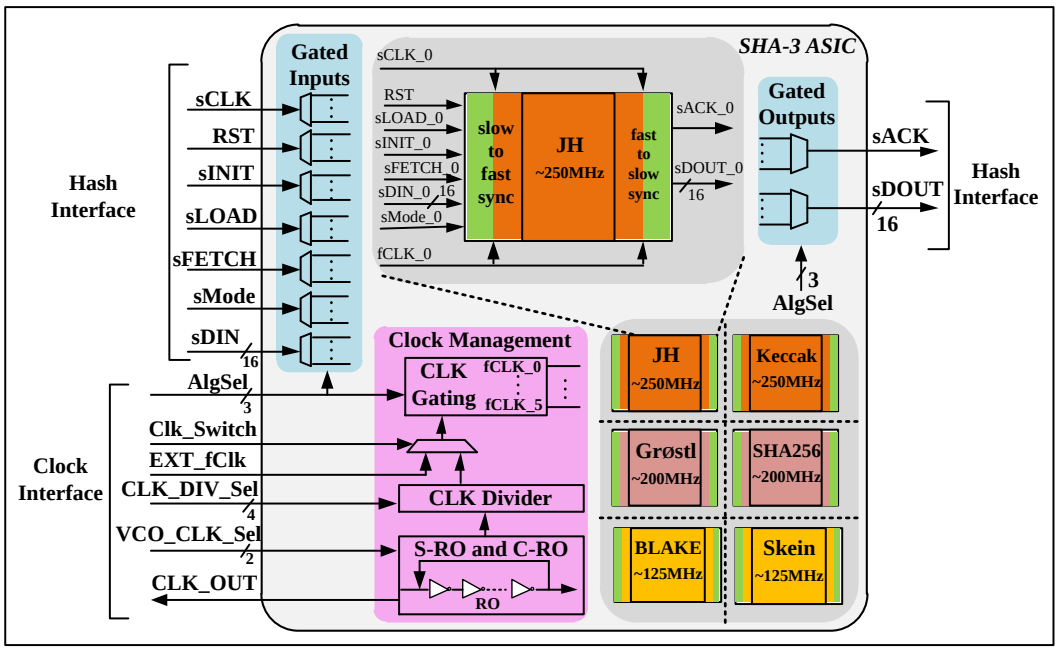

Figure 2: The block diagram of SHA-3 ASIC.

and synthesis approach.

5.2.1. Overall Architecture

Our ASIC chip includes five finalist SHA-3 candidates, a reference SHA-256, a standard control interface and a clock management unit as shown in Fig. 2. Each candidate has synchronizer unit interface to synchronize data with slow IO pins.

5.2.2. Hash Interface

For testing each candidate on single platform and ensuring fair comparison we have built uniform standard handshake interface. The chip interface is adopted from the standard hash interface proposal by Chen et al. [5]. We have extended it to add mode selection and dual clock support. The standard interface is shown in Fig. 2.

5.2.3. Clock Domain Crossing (CDC) Synchronizer

Each candidate is capable of running at very high speed (e.g. 250 MHz for JH and Keccak), while the interface on the chip runs at much lower speed.

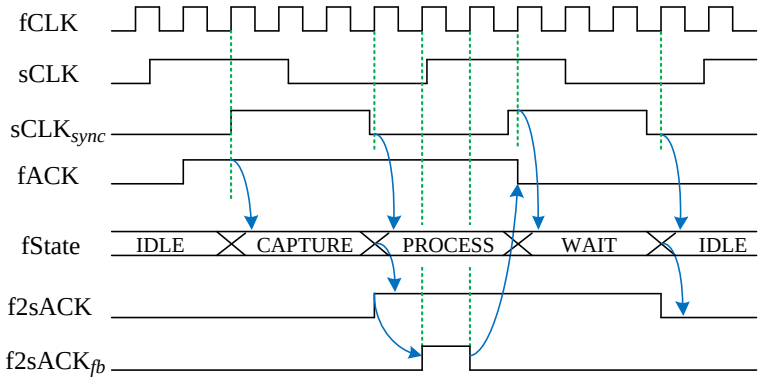

Figure 3: The timing of slow-to-fast synchronizer design.

This is due to the limitation of on-board connectivity as well as limitation of the fastest inputs ports in the IO library of $130nm$ technology. To interface data between these two domains running at different speed, we have designed CDC synchronizer around each candidate. There are two clock domains in our chip: the slow one is for the interfacing logic and the fast one is for hash modules as shown in Fig. 2. In order to avoid complex synchronizer designs based on asynchronous FIFOs or feedback synchronization and also alleviate the burden of the backend process to deal with the two clock domains, we have simplified the synchronizer design. This simplification is achieved 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. 3, 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 [5] 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.

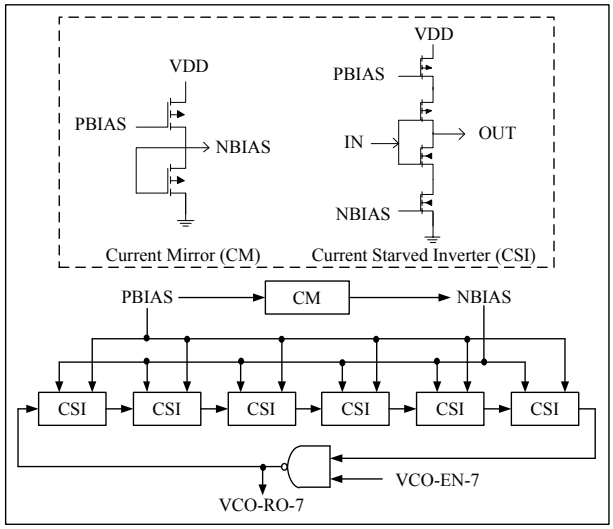

Figure 4: The architecture of 7-stage C-RO clock design.

5.2.4. Clock Management

There were three challenges in designing the clock distribution network for this chip. First, to accurately measure the power dissipation of each candidate we need to ensure that at any given time only one candidate core is active, which implies the clock to all other inactive cores should be turned off. We have used a clock gating technique to achieve this. Second, due to limitation of $130nm$ technology node and the library, IOs cannot route any off-chip signal with greater than 100 MHz frequencies into the chip. To allow testing of all candidates at a higher frequency, we should have a mechanism to generate high on-chip stable clock frequencies. For this purpose, we have designed and implemented an on-chip RO (Ring Oscillator). Third, to evaluate the performance for each candidate at different throughputs, we should have a range of such frequencies available on-chip. For this reason, we have designed a programmable on-chip clock divider. Later part of this Section summarizes architecture and design of individual modules.

(a) Clock Gating: During testing, each candidate is tested seaparately. Separate power domains and clock gating technique will yield an accurate measurement in leakage (static) and dynamic power, respectively. However, as SASEBO-R test platform only supports a single power network for the

chip and leakage at $130nm$ technology is negligible, we went ahead with single power domain with clock-gating technique for this chip. Later, we apply approximation to eliminate leakage component contributed by other inactive cores. We estimate 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 compared them with the results from post-layout simulation and they closely match with each other as shown in Table 4.

(b) Clock Generator Module

We used two different approaches for generating the on-chip clock, namely standard-cell based RO (S-RO) and custom-cell based RO (C-RO). S-RO and C-RO along with programmable clock divider module can support a wide range of clock frequencies for our needs.

Standard cell based RO: We have designed S-RO using inverters from standard cell library. S-RO is gated using a NAND gate thereby saving dynamic power when not in use. After carefully studying the desired range of frequencies and drive strength capabilities of inverters in the library, S-RO was implemented using 25, 33, 43 stage inverters. These are fixed frequency clocks for a given operating voltage "VDD" for the chip. If the supply voltage for the chip is increased, the frequency generated out of S-RO will increase and vice-versa.

Custom cell based RO: We used the custom-cell design approach to integrate a RO based voltage-controlled oscillator (VCO) into the chip. The architecture of 7-stage VCO is shown in Fig. 4. As shown, RO of the VCO is realized using a NAND cell as well as Current Starved Inverter (CSI) cells, and Current Mirror (CM) custom cells. The VCO takes four input ports PBIAS, VCOEN- 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. VCORO- 7, 9, 11 are the three frequencies produced by the block for any particular PBIAS voltage. PBIAS signal is used to control the bias current in PMOS. Therefore, the frequency generated will be highest at a bias voltage of 0V and it will decrease with any further increase in bias voltage. FREQ-7 is the clock from a 7-stage RO. For desired frequency requirements, extensive Spice simulations have been performed to determine the optimum stage length and appropriate device dimensions. The simulations were performed at different process corners. After analyzing the

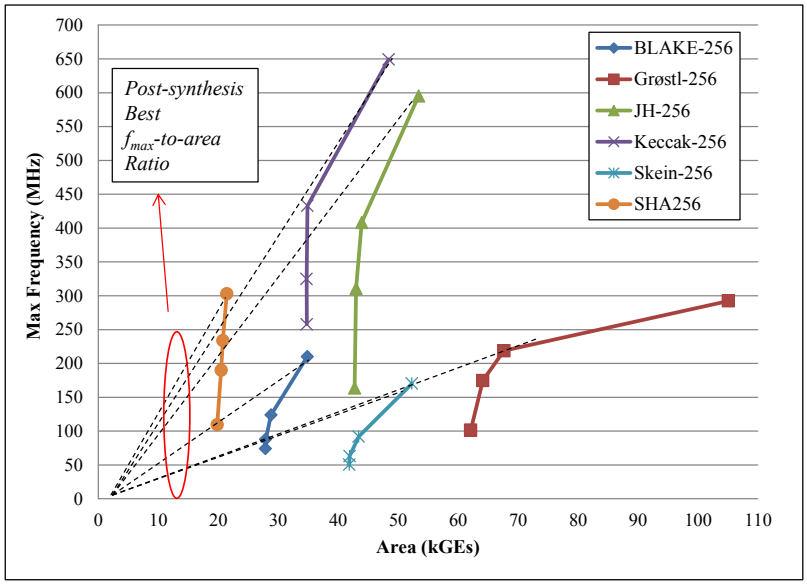

Figure 5: Area-Frequency (A/F) exploration at synthesis stage.

simulation results, we implemented ROs with stage lengths of 7, 9, and 11.

(c) Programmable Clock Divider: We have designed a programmable clock divider using standard cells. There are four pins dedicated to program the divisor. With this, it is capable of dividing the input clock by the following factors; 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192 and 384. The integration of this module along with clock generator module offers flexibility for generating a wide range of stable clock frequency on chip.

5.2.5. Synthesis of SHA-3 Cores: A Bottom-Up Approach

The decisions made during the synthesis process will dominate the performance, the area and the power characteristics of each candidate's implementation in ASIC. Hence, choosing the appropriate constraint at this point is very important. Analysis on various optimization targets were done in our previous work for maximum speed, minimum area and trade-off points [10]. However, as described in Section 4, for this study all candidates were optimized to achieve the best T/A ratio. In the process, extensive Area/Frequency (A/F) explorations were done for each candidate as shown in Fig. 5. We chose the maximum $f_{max}/Area$ point as indicated by dotted lines

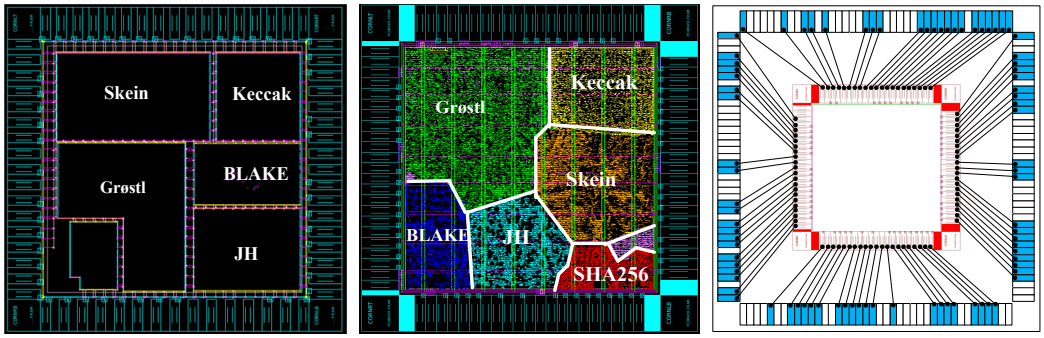

Figure 6: (a) HA Floorplan (b) FA floorpan of SHA-3 chip (c) IO-bonding diagram with 160-pin package.

in the graph. These points are then considered as desired target speed for each candidate and applied in the form of timing constraint, during synthesis. We have followed a bottom-up approach in which each candidate is synthesized separately. Later, the individual synthesized netlists are integrated in a top-level. This approach also aids in doing extensive post-synthesis verification for each candidate separately.

5.3. Implementation Phase: Layout Level Details

In this Section, we will present design details of floorplanning, IO-pin placement and power distribution network.

5.3.1. Floor-Planning

We started with two different exploration methods for optimum placement of all candidates namely hierarchical approach (HA) and flat approach (FA).

(a) Hierarchical Approach (HA): This approach is bottom-up, in which each candidate is placed and routed independently. Each core is then instantiated as a macro at the top-level, and the resulting top-level is interconnected, optimized for timing and routed. The advantage of the HA is that each candidate can be modified independently. For any change in any module, only that module can be re-worked and re-instantiated at the top-level. Design runtime for the entire place and route process is less. Commercial EDA tools offer macro floor-planning to be only in the rectilinear form as shown in Fig. 6(a). Since each candidate is optimized individually, the optimization

of interconnections and logic that falls outside each macro is sub-optimal. Therefore, the area utilization of the overall chip is sub-optimal.

(b) Flat Approach (FA): In this approach, the design is flattened physically after synthesis. We maintain logical hierarchy for analyzing post-layout performance and power for each candidate. In this approach, cross-boundary optimization can also be achieved. Therefore, the optimization and placement of all cells are now optimal. The hierarchy boundaries will no longer be in any rectilinear form as shown in Fig. 6(b).

(c) Analysis of HA and FA: The final goal of each approach is to route the design without any DRC (Design Rule Check) and LVS (Layout *vs.* Schematic) errors. To do a fair comparison between both approaches, we have applied same timing constraints and used same synthesized netlist. To evaluate their efficiency, we compare total area utilized by standard-cell placement. The measured core area using FA is $2.74mm^2$ and using HA is $3.82mm^2$. This difference is due to the fact that FA approach will incorporate both, within as well as cross-boundary optimization, and hence the total area will be less than HA. In this particular chip it is 28% less.

The hierarchical approach requires a designer to select a proper aspect ratio and shape for each module. This, however, is something that is subject to optimization as well. There can be many arguments on floor-planning considering our ultimate goal is fair evaluation of SHA- 3 candidates. Now each candidate can either have different shapes (for example rectangle, square, L-shape) or same shapes with different aspect ratios. Identifying an unbiased shape and aspect ratio for all candidates who would justify this approach is a research problem by itself. This is because every algorithm may have its own floorplan shape dependency and one may act in favor of another. In our case we did not solve this research problem but went with the smallest chip obtained using FA. Total chip area is $2.238\times2.238mm^2$, out of which inner core area is $1.656\times1.656mm^2$ and $2.88mm^2$ space is left for IO pads on all four sides.

5.3.2. IO Placement: SASEBO-R Compatible

There are 84 IO pins on the chip. Out of the 84, 58 are logic connections to the chip and rest are power/ground pins. The size of our chip allows us to have over 86 pins around its periphery. This implies that our chip will not be pin limited. The ordering of functional and power pads conforms to SASEBO-R board. We have packaged our chip with 160 pin packages to facilitate testing. The bonding diagram of chip with the package is shown in

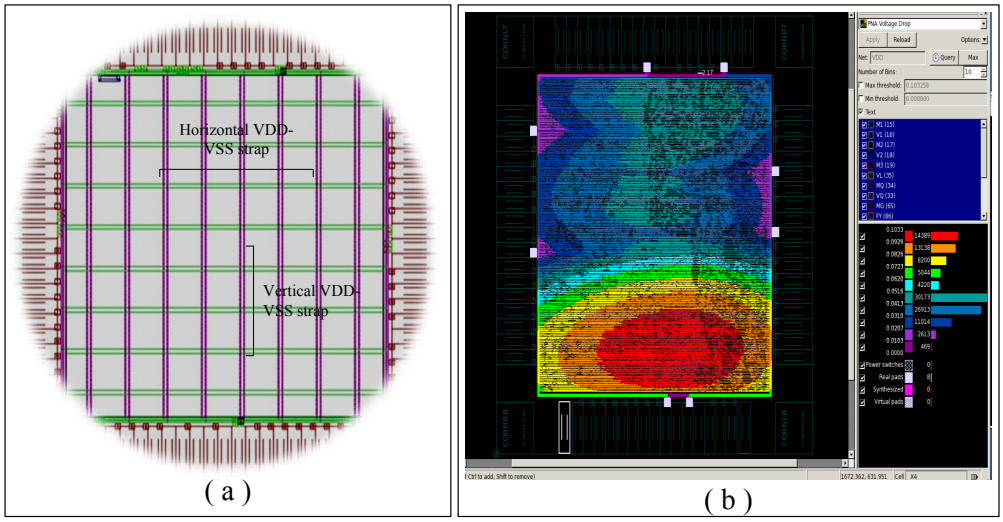

Figure 7: (a) Power mesh structure of chip. (b) IR drop analysis on chip.

Fig. 6(c).

5.3.3. Uniform Power Distribution

As the SASEBO-R board supports only one power domain, this chip was built with one domain. Considering this constraint each candidate now shares the same power network. Structure of power mesh was carefully designed to ensure fair and uniform power distribution. Power mesh structure (a) and IR drop analysis report (b) is shown in Fig. 7. IR drop analysis was carried out to ensure stability and robustness of power network in the presence of all candidates on single platform together. The chip uses a fixed number of power IO pads. Therefore, in order to improve the IR-drop degradation (to less than 10%), we optimized the thickness as well as the spacing of power straps across the chip.

5.4. Testing Phase: Test Setup and Strategy

Test setup is as shown in Fig. 8. It consists of SASEBO-R board, software tester, oscilloscope to measure power, external clock generator and power supply. An ASIC chip is inserted in the socket provided on board. The control FPGA is used to provide stimulus. The software tester is used to

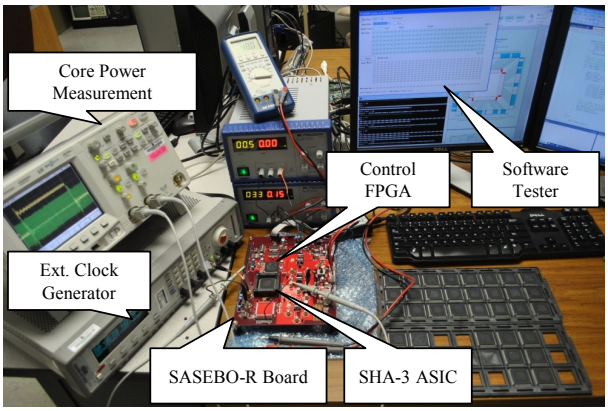

Figure 8: Test setup of SHA-3 chip with SASEBO-R.

measure results obtained from chip. Using this setup, we carry out performance measurement and power measurement.

5.4.1. Functional Testing and Performance Measurements

We perform the functional testing of the candidates in two different modes; normal hashing and high speed hashing mode. In the normal hashing mode, 16 bit of input data is transmitted using IO data pins on the chip for each transfer, until it buffers a whole block of data to start hashing. This I/O bottleneck makes it very difficult for full speed testing since there is a relatively long message loading period between two consecutive hashing. Therefore, we have put in place another mode of operation: high speed testing. Under this hash mode, the input buffer of each candidate only stores 16 bit of input data per block, and replicates this input data to fill an entire block. Each candidate is functionally verified by testing in both these modes. The nominal voltage for $130nm$ library is 1.2V. We use a voltage scaling technique to do further test at the lower limit of 0.8V and upper limit of 1.4V. At every voltage points, we test the functionality and also measure the performance and power characteristics of each candidate.

5.4.2. Power Measurements

Circuit to measure the power consumption for each candidate is shown in Fig. 9(a). Using oscilloscope, power traces were measured for all candidates over different messages. One such power trace is shown in Fig. 9(b). The

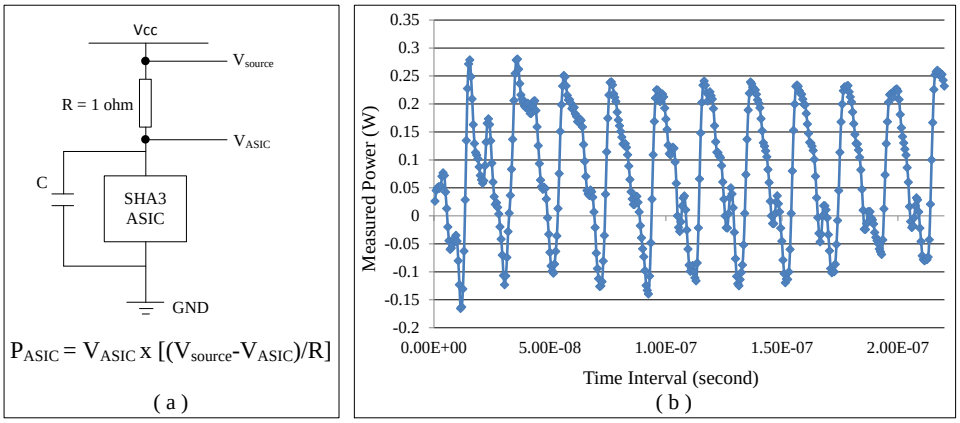

Figure 9: (a) Circuit to measure power. (b) Power trace of Grøstl.

power consumption of each candidate core can be measured in different ways depending on the design environment where it will be deployed. We present three such cases. For example, consider from a system architect's point of view, one would want to measure power for all architectures under a constant clock frequency. Such an environment assumes a predefined system architecture, with a predefined clock, so one wants to know if the SHA3 block after integration will break the system power budget or not. The second case is from an algorithm designer's point of view. The fact that each algorithm has different architecture signifies that the clock period may not necessarily scale with its throughput. In this case, one would want to measure the power and compare the candidates, when they are working under the same throughput. The third case is from an IP designer's point of view, in which one would want to demonstrate how good each candidate is by measuring power consumption under Max Throughput (or Max Frequency) condition for a given technology. In the result Section, we will present power measurement results under each of these cases.

6. Results

In this Section, we will summarize the overall result measured on chip and present further analysis for evaluation of candidates. We first present the range of frequencies available on-chip, followed by measurements of parameters necessary for ASIC benchmarking.

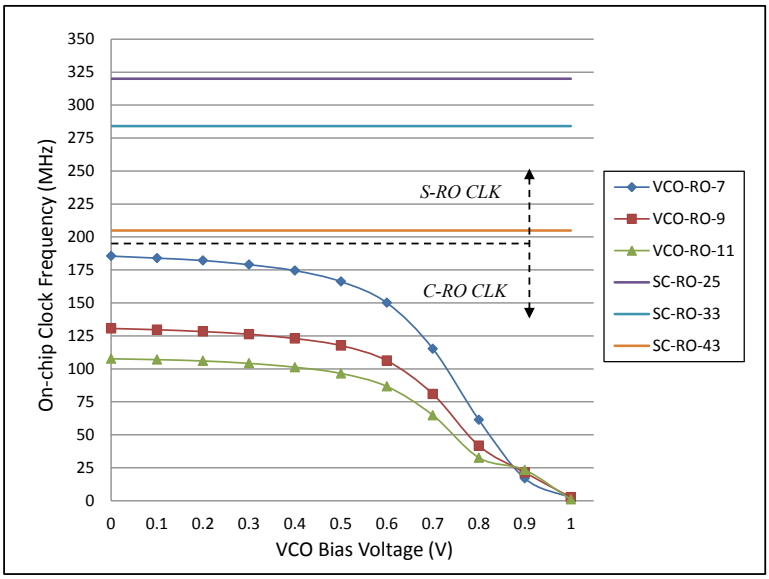

Figure 10: Measured frequencies with on-chip S-RO and C-RO.

6.1. Range of Frequencies Available on Chip

In this Section, we present different range of on-chip clock frequencies measured using S-RO and C-RO on this chip. The measurement includes average frequencies for a batch of 10 fabricated chips. C-RO can generate range of frequency from 20 MHz to 180 MHz over the range of BIAS voltage from 0V to 0.9V as shown in Fig. 10. The maximum frequency available using C-RO is, 180MHz from a 7-stage RO at BIAS voltage of 0V. Three S-RO operating at a nominal voltage of 1.2V, can generate fixed frequency of range 210 MHz, 280 MHz and 320 MHz, respectively. The variation in the clock frequencies were in the range of 3 MHz to 5.8 MHz. With the capabilities of ROs and clock divider module, we can now generate a range of on-chip frequencies from 20 MHz to 320 MHz to test different candidates.

6.2. Benchmarking on ASIC Prototype

In this Section, we present results of the ASIC benchmarking process. We first present the results of functional verification under different modes and under different supply voltages. Later, we will analyze rankings of candidates based on the evaluation metrics as discussed in Section 4.

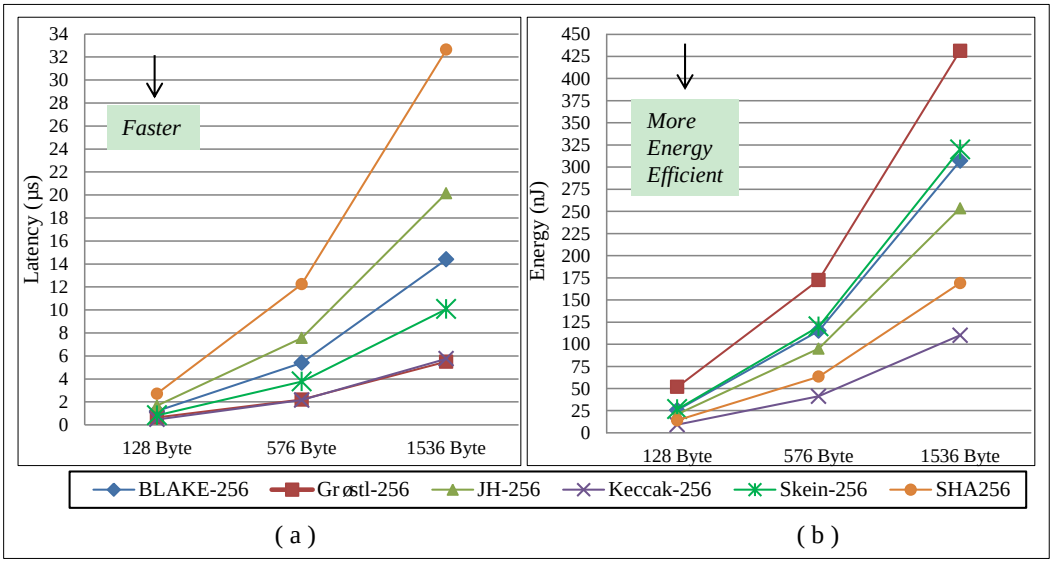

Figure 11: (a) Latency and (b) Energy curve for different message length packet size assuming ideal interface.

6.2.1. Functional Verification under Different Modes

All candidates on this chip have undergone rigorous functional verification. All candidates were functionally verified for both normal hashing and high speed mode. All candidates were also functionally verified using different message lengths as shown in Fig. 11. From the graph, we observe that the latency Fig. 11(a) and energy Fig. 11(b) of all candidates increases with increase in the message length. However the difference in the performance among candidates also increases accordingly.

6.2.2. Functional Verification under Different Supply Voltages

To understand the functional limits of each candidate at different operating voltages, we present the characteristic frequency *vs.* supply voltage shmoo plot in Fig. 12. Core supply voltage *VDD* is varied from 0.8V to 1.4V. At nominal voltage, the measured frequency for all candidates meets targeted frequency. At higher supply voltage of 1.4V, maximum frequency measured from 25-stage S-RO is 250MHz. We were unable to verify the functionality beyond 350 MHz with available range of on-chip frequencies. Hence, at an operating voltage of 1.4V there exists possibility that JH and Keccak can run faster than 350Mhz. From the shmoo plot, we can rank all candidates in

Candidates	BlockSize [bits]	Lcore [cycles]	Area [mm^2]	Area [kGE]	Max Freq. [MHz]	Tp [Gbps]	T/A [kbps/kGE]	Power [mW]	Energy [mJ/Gbits]
BLAKE-256	512	30	0.20	34	171	2.13	62.47	21.33	27.25
Grøstl-256	512	11	0.72	124	204	9.31	74.87	78.42	34.52
JH-256	512	42	0.28	49	284	3.05	61.83	12.57	23.78
Keccak-256	1024	24	0.24	42	284	10.67	251.05	19.12	9.86
Skein512-256	512	21	0.38	66	155	3.05	45.93	31.74	27.61

Table 3: Performance and power characteristics of all candidates as measured on ASIC chip.

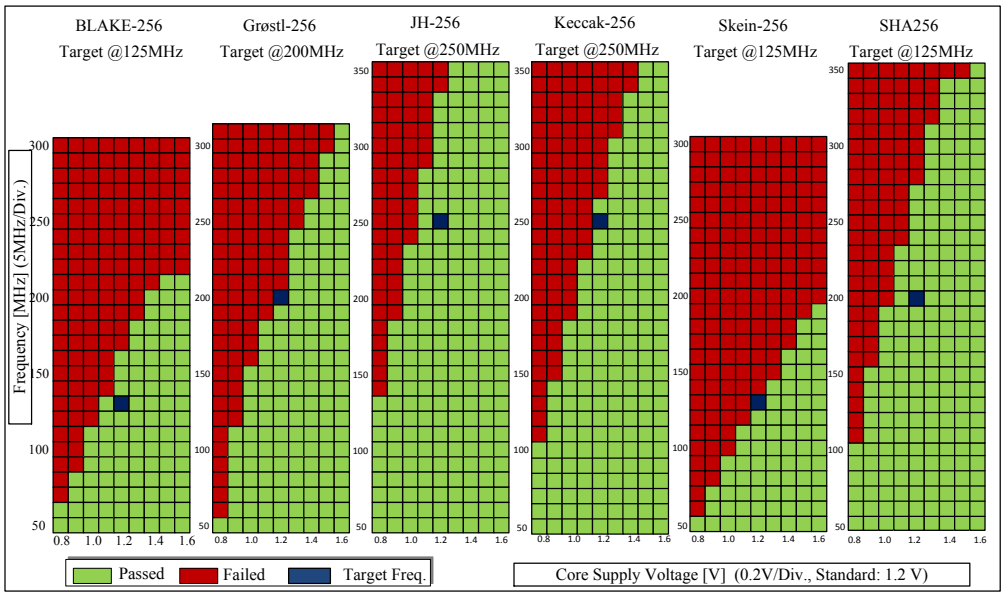

Figure 12: Shmoo plot for frequency-voltage for all candidates

terms of frequency of operation. JH and Keccak can run at maximum speed of 350 MHz, followed by Grøstl, BLAKE and Skein.

6.2.3. Evaluation of Candidates Based on Performance

In this section we will evaluate candidates based on different evaluation metrics that are measured. We have listed all these parameters in Table 3. Considering active area, we observe BLAKE implementation has lowest area

Table 4: Post-layout *vs.* Measured results on the chip

	T/A $[Kbps/GE]$		Power $[mW]$	
Candidates	Post-layout	Measured	Post-layout	Measured
BLAKE	88.75	62.47	19.77	21.33
Grøstl	75.08	74.87	139.29	78.42
JH	62.54	61.83	13.01	12.57
Keccak	254.05	251.05	19.78	19.12
Skein	46.21	45.93	51.09	31.74

of 34 kGE. In terms of performance, Keccak has highest throughput of 10.6Gbps. As discussed in Section 4, to do fair evaluation of all algorithms, we rank these candidates in terms of T/A. Within this, we observe Keccak ranks first with a highest T/A of 251.05 Kbps/GE, followed by BLAKE, Grøstl, JH and Skein. In order to justify the accuracy of these measurements, we also present post-layout results of T/A and power, in Table 4. The proximity of measured result with the estimated result from post-layout simulation eliminates any possibility of error in our measurement technique. Post-layout results also reflect the pessimism in the cell library to incorporate worst-case corner behavior.

6.2.4. Evaluation of Candidates Based on Power and Energy

In this section, we will evaluate candidates based on power consumption for three different cases: Same-Frequency (Same-Freq), Same-Throughput (Same-Tp) and Maximum-Throughput (Max-Tp) as discussed in the testing strategy of Section 5. These results for all candidates are summarized in Table 5. The energy consumption for all the candidates 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. To evaluate the candidates we study their normalized power graph shown in Fig. 13. For the first case, all candidates are running at a fixed frequency of 50MHz. We rank all candidates based on their power consumption. In this case, JH is the most power efficient. However, there are fundamental differences in each algorithm and its architecture, which is why at same operating frequencies the amount of

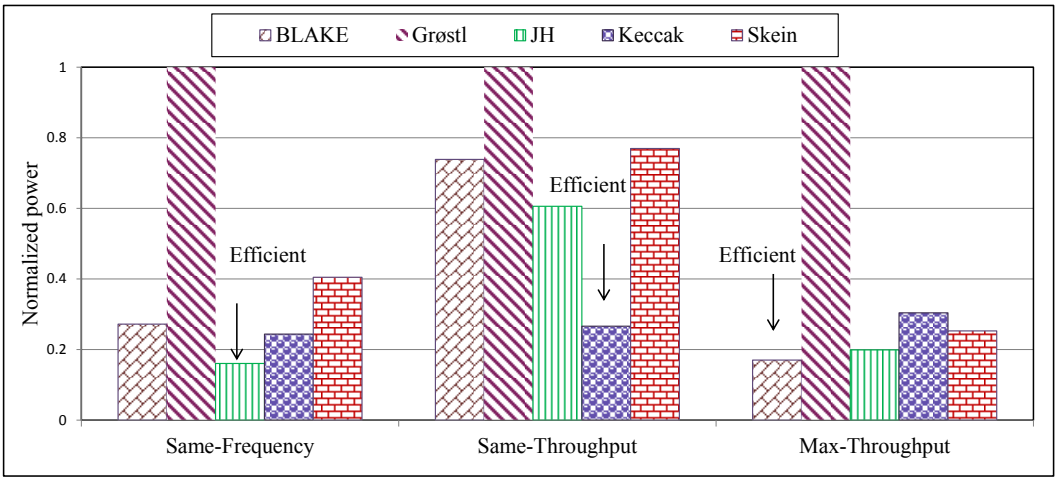

Figure 13: Ranking based on power measurements.

work (throughput) of each candidate is different. Hence, this case cannot be used for true fair comparison. For the second case we rank all candidates when they work to produce digest at constant throughput of 2Gbps. Each algorithm differs in terms of core latency and block-size as shown in Table 3. To hash at a throughput of 2Gbps, each candidate will now be operating at different clock frequency. This case represents true un-biased way of comparison, mainly because we are measuring power consumption under the same workload for each algorithm. In this case, Keccak is the most power efficient candidate. For the third case, we rank candidates when they are running at their respective maximum throughput. This case determines maximum power consumption of each candidate. Each candidate at a given technology ($130nm$) and at any given time will never exceed their measured value. In this case, BLAKE is the most power efficient candidate. We also compute the energy consumption for different candidates under discussed cases: Same-Frequency, Same-Throughput and Maximum-Throughput shown in Table 3. From the table, Keccak is the most energy efficient of all the candidates in all three cases.

6.3. Analyzing Impact of Different Hardware Platform

To analyze the impact on candidates performance (T/A) due to different hardware platforms, we implement same RTL in Virtex-6 FPGA. We present this analysis for both platforms in Fig. 14. The implementations are in dif-

Table 5: Power measurements

Candidates	Power $[mW]$			Energy $[mJ/Gbps]$		
	Same Freq[a]	Same Tp[b]	Max Tp[c]	Same Freq	Same Tp	Max Tp
BLAKE	21.33	49.83	53.15	27.25	24.92	56.26
Grøstl	78.42	67.45	312.33	34.52	33.73	37.87
JH	12.57	40.85	62.15	23.78	20.43	26.17
Keccak	19.12	17.93	94.96	9.86	8.96	13.73
Skein	31.74	51.92	78.99	27.61	25.96	34.73

[a] Same-Freq case is measured at 50MHz frequency.
[b] Same-Tp case is measured at a throughput of 2 Gbps.
[c] Max-Tp case is measured at a maximum frequency measured for each candidate.

ferent technology, $130nm$ in ASIC and $40nm$ in FPGA. Our intention here is not to compare the hardware platforms but to study the trend w.r.t. relative candidates behavior when subjected to different underlying technology (GEs $vs.$ LUTs). Performance of Keccak and Grøstl closely match in both platforms. This signifies Keccak and Grøstl implementations are indifferent to the underlying technology. From the graph, we also observe that the BLAKE and JH implementations are highly sensitive to the underlying platform and technology. Skein is comparatively less sensitive. Since few algorithms are sensitive to the underlying technology, relative rankings of candidates are bound to change in the two platforms for the same architecture. This underlines the importance of doing hardware benchmarking both in FPGA and ASIC.

7. Conclusions & Future Work

In this article we summarize our efforts in ASIC benchmarking of five SHA-3 finalists. We present their rankings based on actual measurements performed on chip based on performance and power consumption. We analyze their trends under different hardware platforms. From our knowledge this is the first ASIC implementation of SHA-3 finalists. For future work

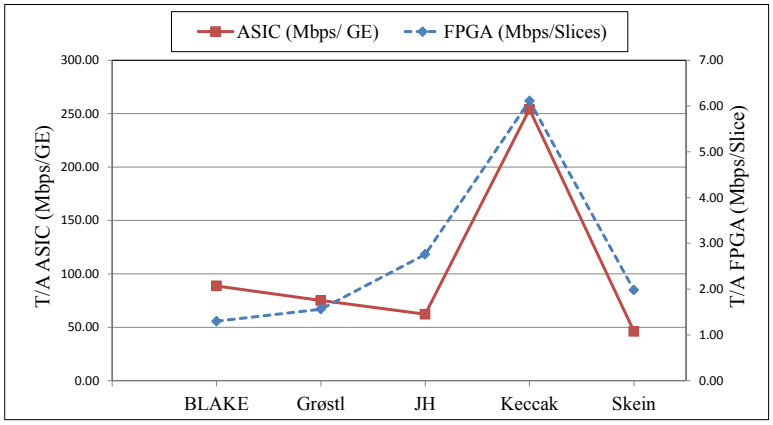

Figure 14: Analyzing T/A trends in FPGA and ASIC platforms.

we also plan to evaluate the side channel analysis (SCA) attack resistance of each SHA-3 candidate based on our SHA-3 ASIC.

Acknowledgment

We would like to thank NIST for funding this research. We would also like to thank National Institute of Advanced Industrial Science and Technology (AIST) [2], of Japan for their hardware platform support and thank Dr. David Blaauw from VLSI Design/Aumotation Lab at Univ. of Michigan for his useful suggestions for the tape-out.

References

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

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

[3] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak sponge function family – Updated VHDL package, May 2011. `http://keccak.noekeon.org/VHDL_3.0.html`.

[4] 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`.

[5] Zhimin Chen, Sergey Morozov, and Patrick Schaumont. A Hardware Interface for Hashing Algorithms. Cryptology ePrint Archive, Report 2008/529, 2008. http://eprint.iacr.org/2008/529.

[6] Kris Gaj, Ekawat Homsirikamol, and Marcin Rogawski. Fair and comprehensive methodology for comparing hardware performance of fourteen round two sha-3 candidates using fpgas. In *CHES*, pages 264–278, 2010.

[7] Xu Guo, Sinan Huang, Leyla Nazhandali, and Patrick Schaumont. Fair and Comprehensive Performance Evaluation of 14 Second Round SHA-3 ASIC Implementations. In *The Second SHA-3 Candidate Conference*, August 2010.

[8] Xu Guo, Sinan Huang, Leyla Nazhandali, and Patrick 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.

[9] Xu Guo, Sinan Huang, Meeta Srivastav, Leyla Nazhandali, and Patrick 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/.

[10] Xu Guo, Meeta Srivastav, Sinan Huang, Dinesh Ganta, Michael Henry, Leyla Nazhandali, and Patrick Schaumont. Pre-silicon characterization of nist sha-3 final round candidates. 14th Euromicro Conference on Digital System Design, August 2011.

[11] 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*, PP(99):1–9, 2010.

[12] Luca Henzen, Pietro Gendotti, Patrice Guillet, Enrico Pargaetzi, Martin Zoller, and Frank Gürkaynak. Developing a Hardware Evaluation Method for SHA-3 Candidates. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, volume 6225 of *LNCS*, pages 248–263. Springer Berlin / Heidelberg, 2010.

[13] Stéphanie Kerckhof, François Durvaux, Nicolas Veyrat-Charvillon, Francesco Regazzoni, Guerric Meurice de Dormale, and François-Xavier Standaert. Compact fpga implementations of the five sha-3 finalists. In *CARDIS*, pages 217–233, 2011.

[14] 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*, PP(99):1 –13, 2011.

[15] K. Kobayashi, J. Ikegami, M. Knezevic, X. Guo, S. Matsuo, Sinan Huang, L. Nazhandali, U. Kocabas, Junfeng 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*, pages 60 –63, june 2010.

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

[17] Stefan Tillich, Martin Feldhofer, Mario Kirschbaum, Thomas Plos, Jörn-Marc Schmidt, and Alexander Szekely. 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`.

[18] Stefan Tillich, Martin Feldhofer, Mario Kirschbaum, Thomas Plos, Jörn-Marc Schmidt, and Alexander Szekely. Uniform Evaluation of Hardware Implementations of the Round-Two SHA-3 Candidates. In *The Second SHA-3 Candidate Conference*, August 2010.

[19] Meltem Sönmez Turan, Ray Perlner, Lawrence E. Bassham, William Burr, Donghoon Chang, Shu jen Chang, Morris J. Dworkin, John M. Kelsey, Souradyuti Paul, and Rene 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`.