On The Impact of Target Technology in SHA-3 Hardware Benchmark Rankings

Xu Guo, Sinan Huang, Leyla Nazhandali and Patrick Schaumont

Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061, USA
{xuguo,shuang86,leyla,schaum}@vt.edu

Abstract. Both FPGAs and ASICs are widely used as the technology for comparing SHA-3 hardware benchmarking process. However, the impact of target technology in SHA-3 hardware benchmark rankings has hardly been considered. A cross-platform comparison between the FPGA and ASIC results of the 14 second round SHA-3 designs demonstrates the gap between two sets of benchmarking results. In this paper we describe a systematic approach to analyze a SHA-3 hardware benchmark process for both FPGAs and ASICs, and we present our latest results for FPGA and ASIC evaluation of the 14 second round SHA-3 candidates.

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 1 in July 2009, 14 out of the 51 hash candidate submissions were identified for further consideration as SHA-3 candidates. These 14 candidates will be further analyzed with respect to security, cost and performance, covering both algorithm and implementation characteristics [1]. For the second phase of the competition, NIST is looking for additional cryptanalytic results, as well as for performance evaluation data on hardware platforms.

Two major classes of hardware devices, Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs), were extensively studied during Round 2 SHA-3 hardware evaluation [2–12]. It is widely accepted that FPGAs and ASICs implementing the same design show different characteristics [13]. A hardware benchmarking process, therefore, starts by fixing the target technology, either ASICs or FPGAs, and then report the results based on selected metrics that are appropriate for the target technology. Several SHA-3 hardware rankings have been obtained in this manner. In this paper we intend to address the question if the choice of target technology can affect the resulting ranking between FPGA and ASIC designs built *based on the same HDL source code.* We motivate our work by the need of the SHA-3 hardware benchmarking process. Different ASIC and FPGA rankings have been provided and implied the superiority of certain algorithms.

In general, compared to ASICs, FPGAs offer many advantages including reduced nonrecurring engineering and shorter time to market. These advantages come at the cost of an increase in silicon area, a decrease in performance, and an increase in power consumption when designs are implemented on FPGAs. These inefficiencies in FPGA-based implementations are widely known and accepted, although there have been few attempts to quantify them. One exception is Kuon, who describes the gap between ASIC and FPGA in terms of area, performance, and power consumption [13]. Kuon compares a 90-nm CMOS FPGA and 90-nm CMOS standard-cell ASIC in terms of logic density, circuit speed, and power consumption for core logic. He finds that, for a representative set of benchmarks, the area gap between FPGA and ASIC is 35 times. He points out that the area gap may decrease when "hard" blocks in the FPGA fabric (multipliers, memories, and so on) would be used. The ratio of critical-path delay, from FPGA to ASIC, is roughly three to four times. The dynamic power consumption ratio is approximately 14 times and, with hard blocks, this gap generally becomes smaller.

In this work we report on a methodology to provide a consistent comparison between SHA-3 FPGA and ASIC designs with three major steps. First, we select the technology node for both FPGAs and ASICS as the starting point for our cross-platform evaluation. Second, we propose several metrics to approach a comparison between FPGA and ASIC results. Third, present an analysis of such results for 14 candidates implemented in ASIC and FPGA.

2 Related Work

The hardware evaluation of SHA-3 candidates has started shortly after the specifications and reference software implementations of 51 algorithms submitted to the contest became available. The majority of initial comparisons were limited to less than five candidates [2, 12]. 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, in both FPGA and ASIC categories, several comprehensive studies have been reported [3–11]. Matsuo et al. [8, 9] focused on the use of FPGA-based SASEBO-GII board from AIST, Japan. All the results are based on the prototyping results and real measurements on a Xilinx Virtex-5 FPGA on board. Gaj et al. [3, 4] conducted a much more comprehensive FPGA evaluation based ATHENA, which can generate multiple sets of results for several representative FPGA families from two major vendors. Baldwin et al. compared hardware implementations of different message digest sizes, including hardware padding, on a Xilinx Virtex-5 FPGA. Guo et al. [10] used a consistent and systematic approach to move the SHA-3 hardware benchmark process from the FPGA prototyping by [8, 9] to ASIC implementations based 130nm CMOS standard cell technology. Tillich et al. [6] 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 [5]. Henzen et al. [7] implemented several architectures in a 90nm CMOS standard cell technology, targeting high- and moderate-speed constraints separately, and presented a complete benchmark of post-layout results.

Table 1. Compare the related SHA-3 hardware benchmarking work in both FPGAs and ASICs

		FPGA	
	Matsuo [8, 9]	Gaj [3, 4]	Baldwin [11]
Own Source Code?	Yes	Yes	Yes
Technology Choices	Xilinx 65nm Virtex-5	Multiple FPGAs Xilinx & Altera	Xilinx 65nm Virtex-5
Hardware Interface	Defined standard 'handshake' interface	Defined standard 'FIFO'interface	Defined standard interface w/ HW padding
Chosen Metrics	Area, Throughput, Power, Energy	Area, Throughput, Throughput-to-area ratio	Area, Throughput, Throughput-to-area ratio
Design Flow	FPGA prototyping with measurements	Post-place & route simulation	Post-place & route simulation
		ASIC	
	Guo [10]	Tillich [5, 6]	Henzen [7]
Own Source Code?	Same as [9, 8]	Yes	Yes
Technology Choices	130nm CMOS Standard Cell	180nm CMOS Standard Cell	90nm CMOS Standard Cell
Hardware Interface	Same as [9, 8]	Assume infinite bandwidth interface	Assume infinite bandwidth interface
Chosen Metrics	Same as [9, 8]	Area, Throughput,	Area, Throughput, Energy
Design Flow	Post-layout simulation	Post-layout/synthesis simulation	Post-layout simulation

Table 1 compares these benchmarking efforts, and demonstrates that a comparison between FPGA and ASIC is hard because of several reasons. First, most groups do not share the same source codes. Second, the ASIC benchmarks do not use a common hardware interface. Third, the reported metrics do not allow a cross-platform (ASIC-FPGA) comparison. Although the joint work done by Matsuo et al. [8, 9] and Guo et al. [10] satisfy the first two conditions, still we believe that the chosen metrics are not well-suited for a cross-platform comparison between FPGA and ASIC benchmarks. All of the above issues motivate our work, namely an investigation of the (dis)similarity between FPGA and ASIC benchmarks for SHA-3 hardware candidates with 256 bits digest.

3 Methodology

In this section, we describe our efforts in comparing the FPGA and ASIC performance evaluations. We describe the overall design flow that combines FPGA prototyping with ASIC design, and next elaborate the efforts to automate and standardize the ASIC implementation process.

3.1 Standard Interface

So far, several research groups have proposed standard hardware interfaces with well supported design flows, including the interfaces defined by [3, 7, 14, 11]. A more detailed discussion on hash interface issues can be found at [9]. The key issue for a fair comparison is to use a common interface for all candidates. Therefore, we selected the interface proposal of Chen et al. [14] (with a data I/O width of 16-bits), but observe that other proposals may be equally valid choices.

3.2 Technology Node Selection for FPGAs and ASICs

It's not the intention of this article to pitch ASIC against FPGA. Instead, we want to evaluate how the performance numbers found on these two different technologies would be different assuming that someoone starts from the same RTL source code. This consideration affects how the target technologies for comparison are selected.

We have done a survey of hash hardware implementation papers published in CHES proceedings, Cryptology ePrint Archive and SHA-3 zoo in the past five years from 2005 (shown in Figure. 1). For around 90 reported hash implementations in FPGAs, around 56% of them are using 65nm FPGAs and 34% with 90nm FPGAs. For 61 ASIC implementations, 48% designs choose 180nm and 33% for 130nm. Thus, the most popular ASIC technology is several generations behind FPGAs, from 180nm to 65nm. Excluding high-end hardware components such as microprocessors, similar trends exist when looking at industry designed hardware. In our comparisons, we opted for the 65nm technology node for FPGA and the 130nm technology mode for ASIC.

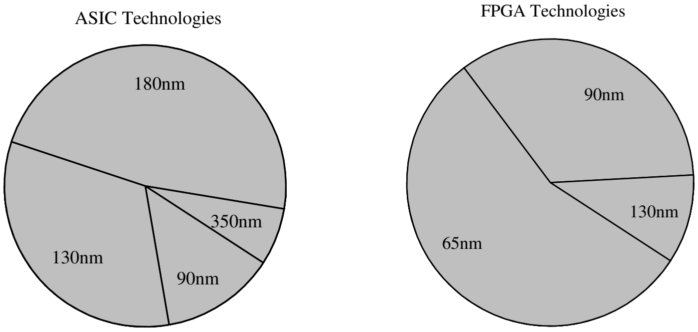

Fig. 1. Technology nodes used for ASIC and FPGA hash implementations in the last 5 years.

We also evaluated the impact of technology scaling on FPGA and ASIC, i.e. we estimated the impact of more advanced technology nodes on our results. For FPGAs, the scaling factors are generally hard to quantify because different FPGA families may have drastically different architectures. In [3], researchers have already demonstrated the influence of different technology nodes on the FPGA results for SHA-3 Round 2 candidates. For example, when moving from a 90nm Xilinx Spartan3E to a 65nm Xilinx Virtex-5, the basic logic element changes from 4-LUT to 6-LUT. In addition, the presence of hardened IP blocks, such as embedded memory (Block RAM), clocking management blocks and DSP functions, can lead to differences between two FPGAs within even the same technology node. Therefore, our comparisons of the 14 SHA3 designs in FPGA are specifically made for a Xilinx 65nm Virtex-5 FPGA. For other FPGA technologies, we recommend the use of an automated framework such as ATHENA [3].

For ASICs, an almost linear scaling factor can be expected. In [10], we used Cubehash, one of the SHA-3 candidates, as a case study to evaluate the impact of different technology nodes (90nm vs. 130nm standard cell ASICs), different ASIC synthesis constraints and compare the post-synthesis results with post-layout results.

3.3 Comparison of FPGA and ASIC CAD flows

In the FPGA CAD flow, all the 14 SHA-3 designs were implemented on Xilinx Virtex-5 (XC5VLX30-3FF324) using the Xilinx ISE 11.1 software for all stages of the CAD flow. The synthesis was performed using ISE XST with default settings to perform speed optimization with normal effort. We changed the HDL options by disabling the tool to infer DSP blocks (which contain multiplier-accumulator circuits), ROMs and RAMS automatically from the RTL. These heterogeneous resources are specific to the Virtex device, and they complicate the analysis. Therefore, we restricted the synthesis tool from using these complex hard macro's. Placement and routing was performed using the standard effort level, and no timing constraints were placed on the design. After generating the post-place & route simulation model, we verified the functionality of each design and collect stimuli traces for power estimation with Xilinx XPower.

While the FPGA CAD flow is straightforward, the CAD flow for ASIC standard-cell implementations is significantly more complicated with more flexibility. We used the Synopsys Design Compiler (C-2009.06-SP3) to map the RTL codes to 130nm (FSC0G_D_SC_TP_2006Q1v2.0) technology. We use the typical case condition characterization of the standard cell libraries.

Although all the RTL designs are optimized for high throughput, depending on the different application scenarios we may put different constraints during the synthesis and layout which may then greatly affect the quality of the ASIC results. We evaluate four design points for every implementation.

MinArea: A minimum-area design will minimize the use of logic resources (gates) 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.

The Synopsys IC Compiler (C-2009.06-SP5) is used for the back-end process. For all the designs we start with 85% utilization of the core area. The *utilization* is the ration of the active chip area (gates) to the total chip area (gates, wires, and empty space). The 130nm technology uses 8 metal layers. In general, more metal layers allow for a denser interconnect, and hence a more optimal use of die area. Overall, we reused the recommended scripted flow from Synopsys Reference Methodology [15]. The area and timing results are obtained from post-layout steps. Power results are obtained from Prime Time (C-2009.06-SP3) after passing post-layout simulation.

3.4 Comparison of area, delay, and power between FPGAs and ASICs

After implementing each design in the ASIC and FPGA flow, the area, delay, and power of each implementation were compared. For ASIC area, we only consider the final core area of the layout without I/O pad cells in Gate Equivalent (GE); the FPGA area is directly retrieved from the post-place & route report in Slices. The critical path delay of both FPGA and ASIC are derived from static-timing analysis assuming worst case operating conditions.

The power metric for FPGA and ASIC includes the static and dynamic portions of the estimated power consumption. We made the following adjustment to make the metric comparable between ASIC and FPGA. The static power of the FPGA is scaled by the fraction of the core FPGA area used by the design. With this, we attempt to compensate for the portion of the FPGA that is not used by a design. Furthermore, a 65nm FPGA technology will have a significantly higher leakage than a 130nm ASIC technology.

We note once more that it's not our intention to pitch ASIC against FPGA, but instead of investigating how the selection of either ASIC or else FPGA may affect the ranking of SHA-3 candidates.

4 FPGA and ASIC Implementation Results for 14 Second-Round Candidates

In this section, we will discuss how to select meaningful metrics to produce comparative results for both FPGAs and ASICs.

Table 2. Proposed metrics for SHA-3 hardware benchmarking

	Description	Note
Metric 1	Maximum Throughput	Useful for both customized & fixed IP cases; Show the performance limits of designs by stretching technology.
Metric 2	Achievable Throughput per Area	Useful for both customized & fixed IP cases; Proportional to (f_{max} /area) which shows the price to pay for stretching technology.
Metric 3	Power and Area under Fixed Throughput	Useful for only fixed IP case; Compare designs considering technology influences but without stretching technology.

To conduct a meaningful comparison, we believe an application scenario must be chosen. Two cases can be considered. The first one is the "customized IP" case, which means the designer will use application-specific information to constrain the FPGA and ASIC CAD flow to achieve the best possible hardware area and performance results of a given IP in a given application. The second one is the "fixed IP" case. In this case, system designers will just reuse a 'pre-made' IP and adapt them to their requirements only by adjusting the clock frequency. In this paper, we will consider the latter case. This leads to the three metrics summarized in Table 2.

For each chosen metric we provide the *relative ranking* of 14 Second-Round Candidates. Each column in the graph of ranking is normalized with respect to the lowest number of that column. The model of rankings exhibits the relative distances among consecutively ranked candidates since some of the designs have very close results which can all be considered as equally good ones. In this way, we can categorize all the candidates into several small groups.

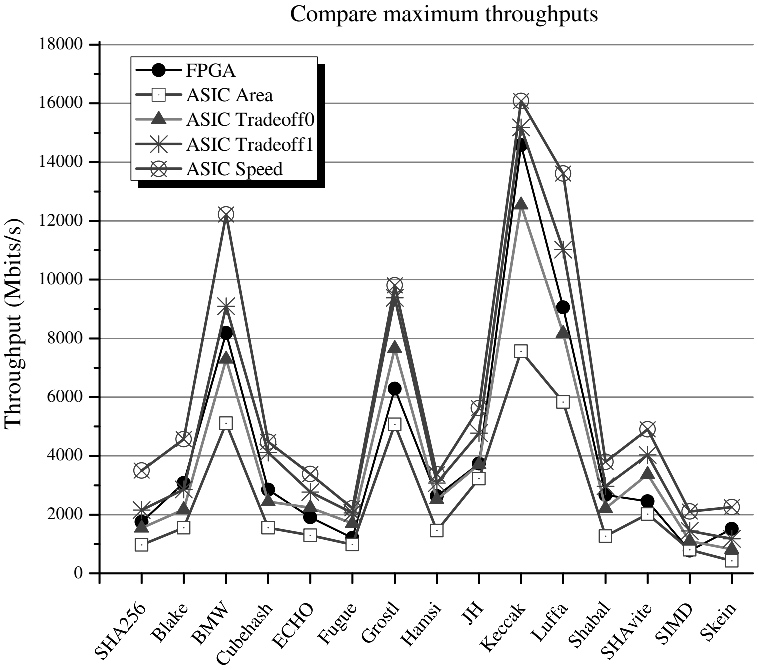

Fig. 2. Compare the maximum throughput between ASICs and FPGAs

4.1 Metric 1: Maximum Throughput

The first metric compares the maximum throughput of different implementations when affected by different technologies and constraints. Since all the 14 Round 2 SHA-3 candidates are designed with high speed optimization in mind, this metric shows the potential of each candidate (see Figure 2).

From Figure 3, we can observe that the rankings of the algorithms under maximum throughput metric are quite uniform between FPGA and ASIC. Only small variations are found because of the impacts of different ASIC backend process constraints to different algorithms with very similar area. For both FPGA and ASIC, Keccak is the best one in terms of maximum throughput, and there are four candidates, Keccak, Luffa, BMW, Grøstl, standing out. In Figure 3 we can also observe how the user's defined backend process constraints will affect the rankings once we fix the ASIC technology .

4.2 Metric 2: Achievable Throughput per Area

In metric 2, we compare the relative achievable throughput per area between ASICs and FPGAs.

From Figure 4, it can be seen that for most of the 14 SHA-3 Round 2 candidates, ASIC Tradeoff1 case has the highest achievable throughput per unit of area and therefore provides an efficient trade-off point between area and throughput.

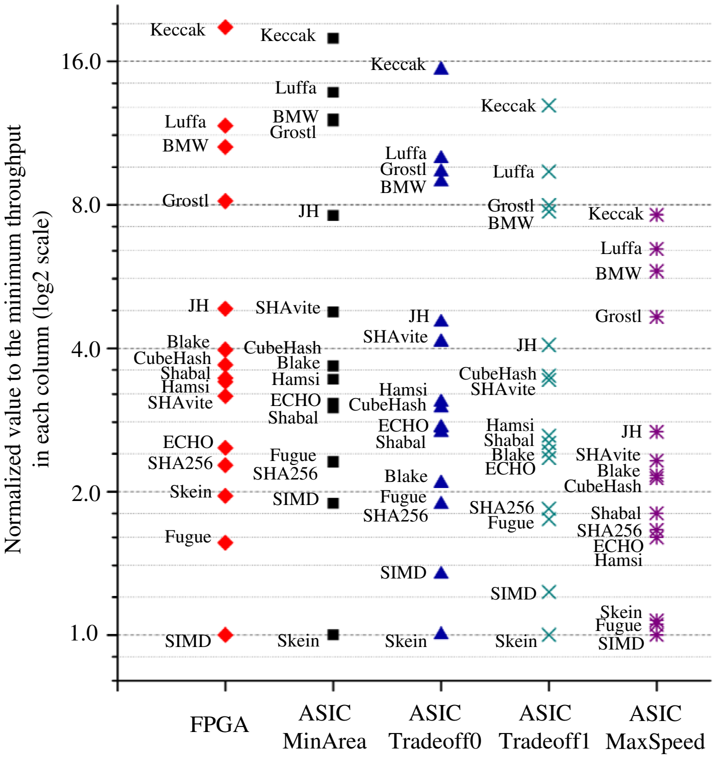

Fig. 3. The ranking of relative maximum throughput in FPGAs and ASICs

From Figure 5, we can observe that the rankings of the algorithms under achievable throughput per area metric have some differences between FPGA and ASIC. One of the major causes for these dissimilarities is the way to calculate the FPGA and ASIC area. Due to the fundamental architectural differences between FPGA and ASIC, it is inaccurate to transfer the basic element, *Slice*, for Xilinx FPGA as the area unit into Equivalent Gate (EG) counts in ASIC. Besides, the critical paths resulted from the existed interconnect networks inside the FPGA can be also an influential variant compared with those in customized ASIC layout. We think these two causes may roughly explain the big difference in rankings for Cubehash between FPGA and ASIC. A more detailed analysis to understand these dissimilarities is still important, and is part of our ongoing work.

This metric helps us to pick the most efficient ASIC implementation as the 'fixed IP' that we will use for point-to-point comparison between ASIC and FPGA. From Section 3, recall that each SHA-3 design has four different ASIC implementations (MaxSpeed, MinArea, TradeOff0, TradeOff1), while there is only one single FPGA implementation. Therefore, the question becomes which ASIC implementation should be finally chosen to compare the FPGA and ASIC

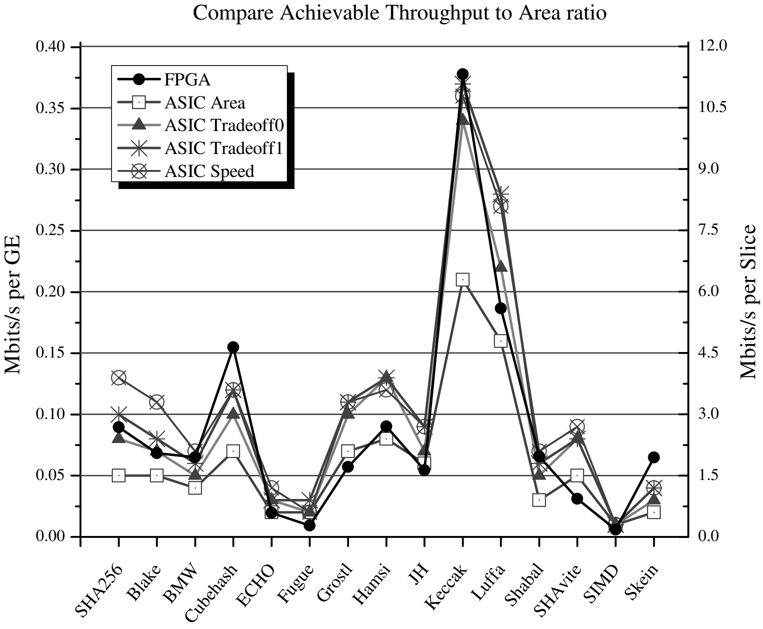

Fig. 4. Compare the achievable throughput per area between ASICs and FPGAs

results. The four ASIC implementations include 2 boundary points, at minimum area and maximum speed. These are extreme cases that are usually avoided in practical design. Instead, we opt to use the so called 'sweet spots' in the ASIC area-delay curve where there is an optimal trade-off between throughput and area. This is especially desirable in a 'fixed IP' scenario when the constraints of the final application are not known beforehand. Note that by choosing default settings of Xilinx ISE tools the FPGA results obtained can also be considered as a good trade-off between are and speed.

4.3 Metric 3: Power and Area under Fixed Throughput

By using the analysis results for metric 2, we can now do a point-to-point comparison between FPGAs and ASICs for all the SHA-3 designs.

The third metric is motivated by the application scenario we mentioned earlier. We assume that the system designers are now considering the system integration of two sets of SHA-3 hardware IPs implemented in ASICs and FPGAs, respectively. Since all those IPs have the same interface and since the system required throughput is fixed, the next step is to figure out whether the selected IP can satisfy a given area and power budget. Therefore we first fix the throughput of each design at 0.2 Gbps. Next, we compare the area and power of the candidates.

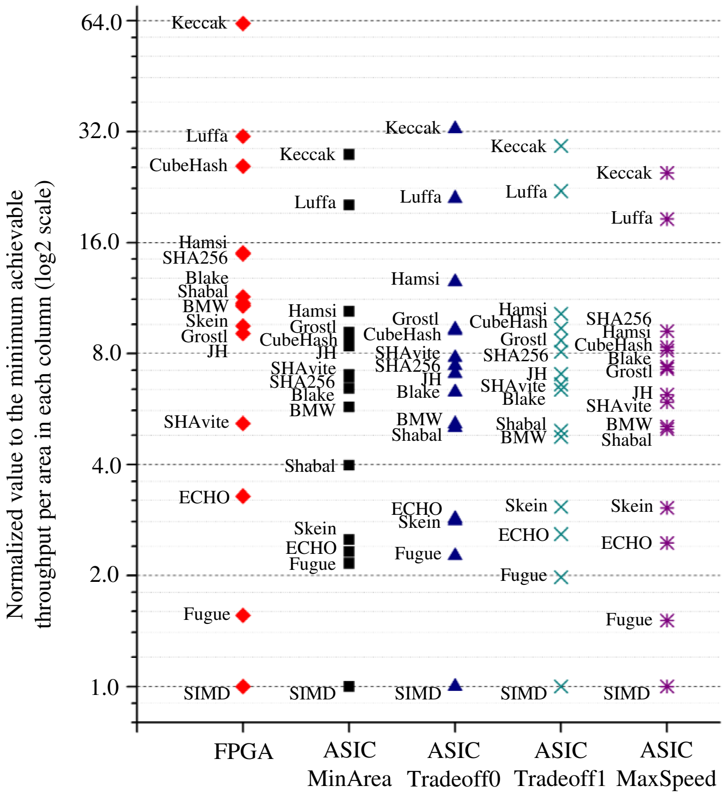

Fig. 5. The ranking of relative achievable throughput per area in FPGAs and ASICs

It can be observed from Figure 6 that the rankings of the algorithms are quite different between FPGA and ASIC, especially in terms of power. This means that characteristics of different candidates scale differently when moved from FPGA to ASIC. In order to study this more closely, we provide a point-to-point comparison between FPGA and ASIC implementation of each candidate. Figure 7 provides this comparison for area and achievable throughput, while Figure 8 shows the same for dynamic power and total power. Each of these metrics are discussed below.

Area. By default the unit of area in ASIC is Gate Equivalent (GE) and Slice for Xilinx FPGA. Internal data from FPGA vendors are needed if one tries to convert the slice to GE [13], but for simplicity here we only give estimation by using the ASIC-GE to FPGA-Slice ratio to denote the area gap between FPGAs and ASICs. The variation of this area ratio can be found in Figure 7, and range of this area ratio is from 18.11 to 56.10 with an average ratio of 29.51.

Throughput. The comparison results of achievable throughput ratio are shown in Figure 7. The ratio is ranged from 0.77 to 2.00, and the average is 1.35, which means when you shift from 65nm FPGA to 130nm ASIC, the maximum

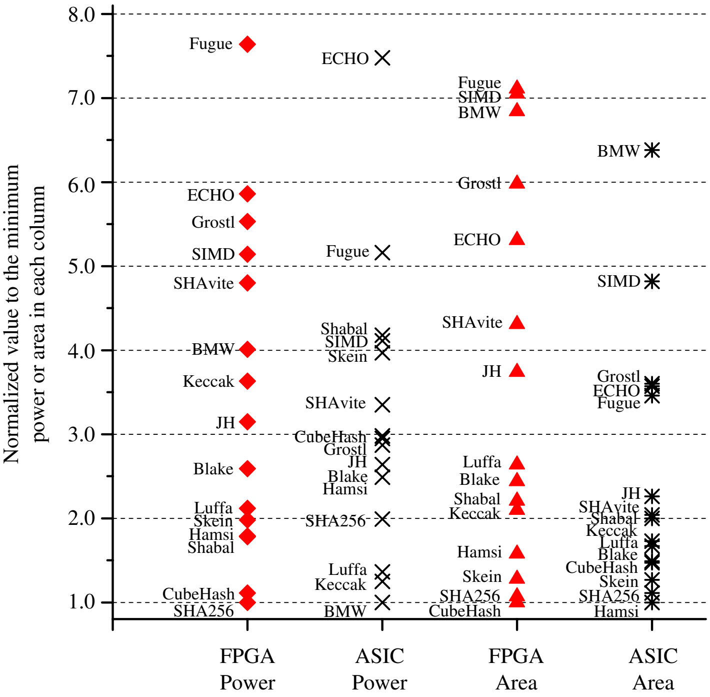

Fig. 6. The ranking of relative power and area with fixed throughput at 0.2Gbps

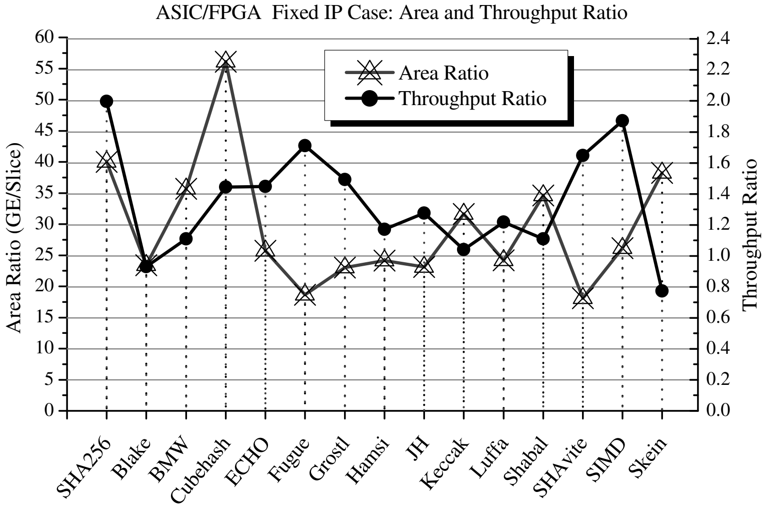

Fig. 7. Compare the ASIC/FPGA area and achievable throughput ratio

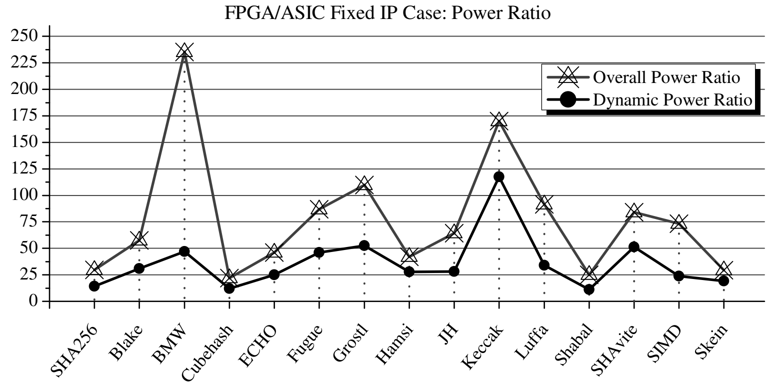

Fig. 8. Compare the FPGA/ASIC overall power ratio with dynamic power ratio

throughput in average will increase by 35%. This gap is much smaller compared to the previously reported numbers by Kuon [13] because they use the same technology node for ASIC and FPGA. In our case, FPGA is able to close the gap because is uses a more efficient technology.

Power. Even after scaling the FPGA static power consumption proportional to the FPGA area, we still find that the static power in FPGA can be as high as 90% for the BMW design with a minimum 31% for the Keccak design. In ASICs, the static power contributes less than 1% for all designs. The overall power ratio ranges from 21.92 to 235.20 with an average of 77.75, while the dynamic power ratio ranges from 11.23 to 117.53 with an average 36.05. So, even for dynamic power the FPGAs still consume 36 times of ASIC power in average from our SHA-3 benchmarking circuits.

5 Conclusions

In this paper, we studied the difference between FPGA and ASIC ranking for 14 SHA-3 Round 2 candidates. Three metrics are carefully selected to deliver meaningful comparison results for SHA-3 FPGA and ASIC implementations.

This paper shows that ASIC and FPGA designers may come to different conclusions when it comes to making a statement on the most efficient SHA-3 candidate in hardware. However, each of ASIC and FPGA SHA-3 designs offer a similar design space (tradeoffs of around 7 times between most and least efficient ones in both area and power metrics as shown in Figure 6).

This paper also lends some insights on how to look at SHA-3 hardware benchmarking results in different platforms. In cases where the platform is already fixed (ASICs or FPGAs), one should exclusively rely on FPGA-specific or ASIC-specific benchmarks, depending on the chosen platform. Conclusions on ASIC implementations based on FPGA results, or vice versa, will almost certainly be inaccurate. In some other cases, where you are looking to understand the SHA-3 candidates and where you do not yet have chosen a platform, it will be equally interesting to compare both the ASIC and FPGA SHA-3 results, because they point out different aspects of SHA-3 hardware implementations.

Future work may include a more detailed analysis of the inconsistent FPGA-to-ASIC gaps for different SHA-3 candidates found in this work, which requires detailed characterizations of each SHA-3 hardware implementations and insights of the ASIC and FPGA architectural differences.

Notice: all the Verilog/VHDL source codes for 14 SHA-3 algorithms with the SHA256 reference design can be found at SASEBO-SHA3 project website (http://www.rcis.aist.go.jp/special/SASEBO/SHA3-en.html).

Table 3. FPGA and ASIC results with fixed throughput at 0.2 Gbps

	Block	Core	Work	ASIC			FPGA		
	Size	Latency	Freq.	Area	Max Freq.	Power	Area	Max Freq.	Power
		(cycles)	(MHz)	(EGs)	(MHz)	(mW)	(Slices)	(MHz)	(mW)
SHA256	512	68	26.6	26167	465.1	2.20	654	232.9	65.10
Blake	512	22	8.6	35062	122.7	2.93	1498	131.9	168.63
BMW	512	2	0.8	149858	35.5	1.11	4200	32.0	261.08
Cubehash	256	16	12.5	34443	257.0	3.31	614	178.1	72.55
ECHO	1536	99	12.99	83747	178.3	8.30	3258	123.1	381.78
Fugue	32	2	12.5	81343	128.5	5.73	4366	75.1	497.54
Grøstl	512	10	3.9	84607	183.2	3.28	3672	122.8	359.97
Hamsi	32	4	25.0	23484	384.6	2.77	973	328.4	116.24
JH	512	36	14.1	53055	335.6	3.18	2294	263.0	204.96
Keccak	1024	24	4.7	40712	355.9	1.39	1288	341.8	236.18
Luffa	256	9	7.0	39152	387.6	1.51	1620	318.3	138.08
Shabal	512	47	18.4	47051	272.5	4.64	1360	245.5	116.13
SHAvite	512	38	14.8	47887	299.4	3.72	2644	181.7	312.54
SIMD	512	46	18.0	113202	129.9	4.56	4331	69.4	334.32
Skein	256	21	16.4	29931	96.3	4.41	783	124.5	129.15

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").

References

1. E. Barker, et al. *Report on the Development of the Advanced Encryption Standard (AES).* Available at: http://csrc.nist.gov/archive/aes/round2/r2report.pdf, Aug., 2010.
2. The SHA-3 Zoo - The ECRYPT Hash Function Website. Available at: http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo , Aug., 2010.
3. K. Gaj, E. Homsirikamol, and M. Rogawski. *Fair and comprehensive methodology for comparing hardware performance of fourteen round two SHA-3 candidates using FPGA.* Proceedings of CHES2010, LNCS, Springer, 2010.
4. K. Gaj, E. Homsirikamol, and M. Rogawski. *Comprehensive Comparison of Hardware Performance of Fourteen Round 2 SHA-3 Candidates with 512-bit Outputs Using Field Programmable Gate Arrays.* NIST 2nd SHA-3 Candidate Conference, 2010.
5. 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.* NIST 2nd SHA-3 Candidate Conference, 2010.
6. S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, and A. 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.
7. L. Henzen, et al. *Developing a Hardware Evaluation Method for SHA-3 Candidates.* Proceedings of CHES2010, LNCS, Springer, 2010.
8. Matsuo et al. *How Can We Conduct "Fair and Consistent" Hardware Evaluation for SHA-3 Candidate?.* NIST 2nd SHA-3 Candidate Conference, 2010.
9. K. Kobayashi, et al. *A Prototyping Platform for Performance Evaluation of SHA-3 Candidates.* Proceedings of HOST2010, 2010.
10. X. Guo, S. Huang, L. Nazhandali, and P. Schaumont. *Fair and Comprehensive Performance Evaluation of 14 Second Round SHA-3 ASIC Implementations*, NIST 2nd SHA-3 Candidate Conference, 2010.
11. B. Baldwin, et al. *FPGA Implementations of the Round Two SHA-3 Candidates*, NIST 2nd SHA-3 Candidate Conference, 2010.
12. B. Baldwin, et al. *FPGA Implementations of SHA-3 Candidates: CubeHash, Grøstl, LANE, Shabal and Spectral Hash.* IACR ePrint Archive, Report 2009/342, 2009.
13. I. Kuon, and J. Rose. *Measuring the Gap Between FPGAs and ASICs.* IEEE Tran. Computer-Aided Design of Integrated Circuits and Systems, vol.26, no.2, pp.203-215, 2007.
14. Z. Chen, S. Morozov, and P. Schaumont. *A Hardware Interface for Hashing Algorithms.* IACR ePrint archive, 2008/529, 2008.
15. Reference Methodology Retrieval System from Synopsys SolvNet. Available at: https://solvnet.synopsys.com/rmgen/ , Aug., 2010.