Yet Another SHA-3 Round 3 FPGA Results Paper

Brian Baldwin and William P. Marnane

Claude Shannon Institute for Discrete Mathematics, Coding and Cryptography.
Dept. of Electrical & Electronic Engineering, University College Cork, Cork, Ireland
{brianb,liam}@eleceng.ucc.ie

Abstract. The NIST run SHA-3 competition is nearing completion. Currently in its final round, the five remaining competitors are still being examined in hardware, software and for security metrics in order to select a final winner. While there have been many area and speed results reported, one such metric that does not appear to be covered in very great detail is that of power and energy measurements on FPGA. This work attempts to add some new results to this section, namely, measured area, power, energy and iteration time results thereby giving NIST further metrics on which to base their selection decision.

1 Introduction

The family of Secure Hash Algorithms (SHA) began in 1993 when the National Institute of Standards and Technology (NIST) published the Secure Hash Standard [1]. However, when insecurities were found following attacks against reduced versions of the SHA-2 family [2], the National Institute of Standards and Technology (NIST) started a competition for a new hash algorithm [3], namely SHA-3, similar to the former AES effort [4], with the intention of developing a more secure family of hash functions.

The contest initially received 64 submissions from designers worldwide. 51 of these designs progressing through to round one of the contest which began on November 1^{st} 2008. Approximately a year was given for each round of the competition, with round one being used to examine the security of the applicants. Round two candidates were announced on July 24^{th} of 2009 and the number of competing designs were reduced to 14 [5].

For round two, the NIST competition specifications [6], 6.C, *Round 2 Technical Evaluation* gave the criteria for hardware and software testing; *"Round 2 testing by NIST will be performed on the required message digest sizes"* and *"the calculation of the time required to compute message digests for various length messages"*.

On December 9^{th} 2010, round two was completed and the field was reduced further to 5 competing designs, BLAKE, Grøstl, JH, Keccak and Skein for round three. The competition is currently ongoing at time of writing and the successful candidate selected to represent SHA-3 will be selected some time in 2012.

The Third SHA-3 Candidate Conference took place on March 22-23, 2012 and while there were many good papers on FPGA implementations by Gaj et al. [7], Jungk [8], Latif et al. [9] and Kaps et al. [10] amongst others, only the paper by Jens Peter Kaps included power and energy results. Indeeed, to the best of the authors knowledge, the only other paper to include power results was the round two paper by Miroslav Kneževiĉ et al. [11], and both of these papers only present results for the 256-bit digest versions.

As such the authors felt that there was room for yet another SHA-3 round 3 FPGA paper; this one containing measured power, energy, computation time and area reasults. Also given as a comparison metric are area-time product and area-energy product results as a metric for overall comparison between the designs.

The rest of the paper is presented as follows; Section 2 presents the update *tweaks* and how they will affect the hardware implementations. Section 3 presents the interface used and describes the FSL bus operation and interaction with the communications. It also defines the hardware used and the design methodology. Section 4 presents our results, including area, processing time, power and energy for the round three designs and their round two equivalents. Finally we conclude in Section 5.

2 Round 3 Updates

At the end of each round, the remaining contestants are allowed make minor changes to their designs. These updates, known as *tweaks*, are used for the purposes of increasing the security or decreasing the area or timing on a given platform. The round three implementations presented in this work are updated versions of those given for round two [12], with as few as possible changes made to implement the round three *tweaks* as given in the submission documentation.

While the round 3 changes presented here are obviously short, the interested reader is invited to examine the full changes as given in the round three submission documentation [13]. Also, some minor deviation was found in the expected results as shown below. This is due to the design package routing of the implementations in the case of area results and also minor atmospheric changes and levels of precision in the case of timing and power results.

The round three changes, and their expected impact on the FPGA designs were as follows:

2.1 Blake Round 3

- The number of rounds for the 224 and 256-bit digest versions was changed from 10 to 14.
- The number of rounds for the 384 and 512-bit digest versions was changed from 14 to 16.
- The BLAKE naming convention was changed from BLAKE-28, BLAKE-32, BLAKE-48, and BLAKE- 64 to, respectively, BLAKE-224, BLAKE-256, BLAKE-384, and BLAKE-512.

These extra rounds should result in an increase in timing for the Blake designs, and a slight change to the area due to routing.

2.2 Grøstl Round 3

- New Shift Values: In the original Grøstl-224/256, the transformation ShiftBytes was used in both P_{512} andQ_{512}. In the round three version, different ShiftBytes values are used in Q_{512}. An equivalent update is applied to Grøstl-384/512 and Q_{1024}.
- New Round Constants: In the original Grøstl-224/256, the round constants $C_{[i]}$ of P_{512} and Q_{512} used in the transformation AddRoundConstant in round i were sparse with only a single byte value different from zero. In the tweaked Grøstl-224/256, additional round constants are added for P_{512} and additional round constants are added for Q_{512} along with an xor by FF. Similar tweaks are applied to Grøstl-384/512 round constants.

These updates should result in no increase in the timing but an increase in the area.

2.3 JH Round 3

- The round number of JH is changed from 35.5 rounds to 42 rounds.

This tweak should result in an increase in the timing due to the additional rounds but a decrease in the area as the additional circuitry required for the final half block is removed.

2.4 Keccak Round 3

- The padding rule has been shortened and simplified. The new padding rule is the pad10*1 rule.

This update only affects the padding, and as such the timing and area results should remain the same.

2.5 Skein Round 3

- The only change to the Skein hash function is in the key schedule parity constant where the Skein *tweak* constant is changed.

This change of the tweak should result in minimal changes to timing and area results.

3 Implementation

3.1 Interface

An interface was implemented which makes use of the FSL bus [14] and is designed for SOC type implementations. Similar to Kaps et al. [10], this results in the padding and counters being moved to software. Compared to the wrapper used for the round two designs [15], where the entire design was in hardware, these changes were necessary both due to the fixed size of the FSL bus FIFO and also as it is more feasible when using an FPGA based microprocessor (e.g. Microblaze or PowerPC) to move these calculations to software. Methods were initially tested whereby the first message block defined the message size and counter. However this led to both an unnecessary extra latency in the load time and extra logic increasing both the complexity and the area. As such, software was generated in C to generate the padding prior to the message being loaded to the hash function. Extra counter blocks of equivalent size to a single message block, where required (i.e. Blake, Skein), were also added to the message at this point. The connections between the FSL and the hash interface, unchanged from the round two communications [15], Table 1, are shown in Figure 1.

Table 1. Wrapper Interface

Signal	**IO**	**Description**
clk	in	Global clock
rst	in	Global reset, **Active HIGH**. Initialises the circuitry to begin hashing a new message
d_in	in	The input bus
dp_in	in	Data present on the **input bus**
ack_in	out	Data present on the **input bus** has been read
lb_in	in	Data present on the **input bus** is the last block of the message to be hashed
d_out	out	The output bus
dp_out	out	Data present on the **output bus**
ack_out	in	Data present on the **output bus** has been read
lb_out	out	Data present on the **output bus** is the last block of the hashed message

3.2 FSL Bus

The Fast Simplex Link (FSL) [14] is a uni-directional point-to-point communication channel bus used to perform fast communication between a microprocessor pipeline and user developed custom hardware accelerators (co-processors), in this case the hash functions. This provides a mechanism for unshared and non-arbitrated communication mechanism, thus allowing fast transfer of data words

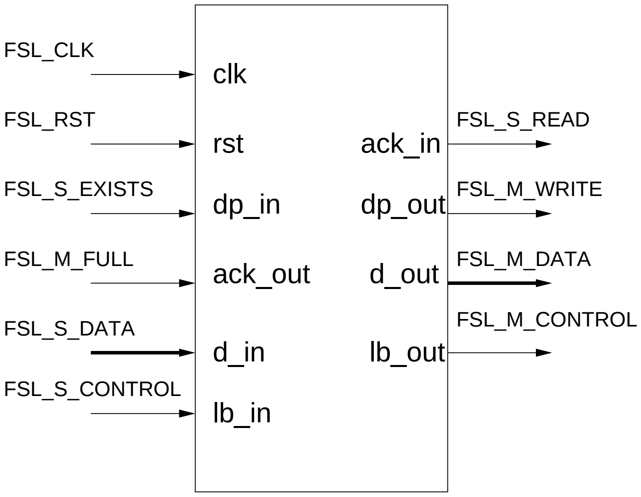

Fig. 1. Hash Wrapper using FSL Bus

between master and slave implementing the FSL interface. Table 2 gives the I/O signals used by the FSL.

Table 2. FSL Bus Signals

Signal	**IO**	**Description**
FSL_Clk	in	Synchronous clock
FSL_Rst	in	System reset, should always come from FSL bus
FSL_S_Clk	in	Slave asynchronous clock
FSL_S_Read	in	Read signal, requiring next available input to be read
FSL_S_Data	out	Input data. Multiple of 32-bit
FSL_S_Control	out	Control Bit, indicating the input data are control word
FSL_S_Exists	out	Data Exist Bit, indicating data exist on the input FSL bus
FSL_M_Clk	in	Master asynchronous clock
FSL_M_Write	in	Write signal, enabling writing to output FSL bus
FSL_M_Data	in	Output data. Multiple of 32-bit
FSL_M_Control	in	Control Bit, indicating the output data are control word
FSL_M_Full	out	Full Bit, indicating output FSL bus is full

The input and output are read in 32-bit blocks and require two clock cycles, a read and an acknowledge, per block. FIFO buffers read and release the data and are set prior to runtime.

3.3 Sasebo GII

The hardware used to test the various algorithms was implemented on the Xilinx Virtex-5 (xc5vlx50-3ff324) FPGA and evaluated on the SASEBO-GII cryptographic evaluation board [16], similar to that of Kneževič et al. [11].

The Sasebo GII evaluation board comprises:

- Two Xilinx FPGAs:
  - A cryptographic FPGA : XC5VLX50 -FFG324 (Virtex-5 series).
  - A control FPGA : XC3S400A-4FTG256 (Spartan-3A series).
- An onboard 24Mhz clock. An external clock input is also supported.
- External power source supplying the on-board power regulators and the FPGAs.

3.4 Methodology

All implementation results were recorded using the Sasebo onboard 24Mhz clock. The external power supply was directly connected for the core voltage of the cryptographic FPGA via an external power meter, namely the Agilent N6705A. Results were taken from both this meter and a Lecroy Waverunner 104Xi oscilliscope, measured across the Sasebo's 1 ohm shunt resistor on the V_{core} line and were generated as follows for complete messages (tested against the KAT values) including all load times:

$$\begin{aligned}
resistor &= 1;\\
meanResistorVoltageDrop &= mean(trace);\\
calculatedCurrent &= meanResistorVoltageDrop/resistor;\\
resistorPower &= (calculatedCurrent^2) * resistor;\\
meanFpgaVoltage &= suppliedVoltage - meanResistorVoltageDrop;\\
meanFpgaPower &= (meanFpgaVoltage^2) * calculatedCurrent;\\
checkFpgaPower &= meanFpgaPower + resistorPower\\
energy &= sum((time(end) - time(1)) * meanFpgaPower)
\end{aligned} \tag{1}$$

4 Round Three Hash Results

It was described in Section 3.1 that the input and output are read in 32-bit blocks on the FSL bus and require two clock cycles[1], a read and an acknowledge, per block. Therefore each input message requires $(MES/32) * 2$ clock cycles to load, where MES is the input message size. Also communication issues occurred when loading of a new message and processing of the current message occurred

[1] In actuality, only the first block requires two clock cycles with subsequent blocks only requiring one. However, in order to avoid undue complication when reading in messages of different sizes, the two clocks per message block standard was used.

simultaneously. In order to work around this, the load stage and the processing stage were completely separated out in respect of the round two wrapper.

The updated round three designs were implemented on the SASEBO and Equation 1 was used to get the timing, power and energy measurements.

4.1 Round Three Area Results

The area results, presented in Table 3, give the Post-Place and-Route results of each hash block as a stand-alone entity, implemented using Xilinx ISE Project Navigator 12.3. Results were taken for both short (S) and long (L) messages, with a short message comprising a message up to 512-bit blocks (or 1088-bit in the case of Keccak), and a long message comprising a randomly selected 11624-bit message. In both cases, the messages tested required padding (generated in software), and in the latter case a number of rounds is required to process the hash. The *Occupied Slices* give the full measurement of area for the FPGA. the *Slice Reg* and *Slice LUT* provide secondary measurements

Table 3. SHA-3 Round 2 & 3 Area PPR Results

		Round 2			Round 3		
Algorithm	Digest	Occupied Slices	Slice Reg	Slice LUT	Occupied Slices	Slice Reg	Slice LUT
Blake	224/256	**1584**	3872	3986	1568	3872	4172
Grøstl	224/256	4124	2594	11632	3137	2594	11484
JH	224/256	1752	2328	4917	**1452**	2328	4110
Keccak	224/256	1908	2712	5561	1908	2712	5561
Skein	224/256	2842	5191	8052	2718	5191	8052
Blake	384/512	2757	7458	7547	2799	7458	7796
Grøstl	384/512	6675	5156	21614	6673	5156	23140
JH	384/512	1851	2329	5211	**1426**	2329	3784
Keccak	384/512	**1551**	2200	4785	1551	2200	4785
Skein	384/512	3044	5448	8509	2998	5448	8509

The Round Two versions of the Round Three designs are also given for comparison purposes since different metrics are used compared to the results as given in [12]. Table 3 shows a number of things. The round two area results mostly show an increase from the values given in the previous work [12] (using the no-wrapper area results), due to the additional wrapper logic. Comparing the round two and round three values directly in Table 3 shows that while Blake and Keccak had the lowest area for 256 and 512 digest sizes respectively for round two, the tweak updates to JH result in it having the lowest area for all digest sizes of the round three designs. As stated in Section 2, there are in some cases slight deviations due to routing.

Table 4. FPGA Power and Timing Results for SHA-3 Round Two

	Power Meter			Oscilliscope					
Algorithm	Supplied	Supplied	Supplied	Mean Resistor	Resistor	Mean FPGA	Mean FPGA	Iteration	Energy
	Voltage	Current	Power	Voltage Drop	Power	Voltage	Power	Time	
224/256	V	mA	mW	mV	mW	mV	mW	μS	μJ
Blake S	1.15	229.5	263.7	226.5	51.3	923.5	193.2	3.75	0.724
Blake L	1.15	247.3	284.2	239.6	57.4	910.4	198.6	76.99	15.3
Grøstl S	1.15	300	322.8	336.3	113.1	813.7	222.7	**2.8**	**0.623**
Grøstl L	1.15	300	275.3	296	87.6	854	215.9	**33.97**	**7.333**
JH S	1.15	226.2	259.9	221.2	48.9	928.8	190.8	6.34	1.209
JH L	1.15	230.5	264.8	222.9	49.7	927.1	191.6	71.38	13.675
Keccak S	1.15	222	255.1	216.5	46.9	933.5	188.7	4.34	0.818
Keccak L	1.15	226	259.7	217.9	47.5	932.1	189.3	43.38	8.213
Skein S	1.15	290.7	334	302.1	91.3	847.9	217.2	3.64	0.79
Skein L	1.15	269.1	309.3	261.4	68.3	888.6	206.4	59.83	12.349
384/512									
Blake S	1.15	264.8	304.4	259.7	67.4	890.3	205.8	5.92	1.218
Blake L	1.15	284.8	327.3	272.5	74.3	877.5	209.8	66.4	13.938
Grøstl S	1.15	206.3	237.1	343.5	118	806.5	223.4	4.85	1.083
Grøstl L	1.15	300	275.3	296	87.6	854	215.9	**33.97**	**7.333**
JH S	1.15	224.9	258.3	215.4	46.4	934.6	188.2	6.67	1.255
JH L	1.15	230.5	264.8	222.9	49.7	927.1	191.6	71.38	13.675
Keccak S	1.15	222.5	255.6	215.2	46.3	934.8	188	**3.34**	**0.628**
Keccak L	1.15	232.4	267.1	221.7	49.2	928.3	191.1	55	10.508
Skein S	1.15	293.1	336.9	298.9	89.3	851	216.5	4.34	0.939
Skein L	1.15	271	311.5	260.2	67.7	889.8	206	60.26	12.414

Table 5. FPGA Power and Timing Results for SHA-3 Round Three

	Power Meter			Oscilliscope					
Algorithm	Supplied	Supplied	Supplied	Mean Resistor	Resistor	Mean FPGA	Mean FPGA	Iteration	Energy
	Voltage	Current	Power	Voltage Drop	Power	Voltage	Power	Time	
224/256	V	mA	mW	mV	mW	mV	mW	μS	μJ
Blake S	1.15	232.4	267	225.6	50.9	924.3	192.8	4.42	0.852
Blake L	1.15	247.6	284.5	235.4	55.4	914.6	196.9	92.42	18.2
Grøstl S	1.15	300	325.6	324.9	105.6	825.1	221.2	**2.8**	**0.619**
Grøstl L	1.15	300	291.1	294.5	86.7	855.5	215.5	**33.9**	**7.3**
JH S	1.15	221	253.9	211.5	44.7	938.5	186.3	6.84	1.274
JH L	1.15	224.7	258.1	213.1	45.4	936.9	187	77.42	14.48
Keccak S	1.15	220.6	253.5	211	44.5	939	186	4.34	0.807
Keccak L	1.15	224.9	258.3	212.8	45.3	937.2	186.9	43.51	8.12
Skein S	1.15	289.6	332.3	296.6	88	853.4	216	4.01	0.866
Skein L	1.15	269.3	309.5	258.4	66.8	891.6	205.4	59.92	12.308
384/512									
Blake S	1.15	268.4	308.3	262.7	69	887.3	206.8	6.59	1.36
Blake L	1.15	289.6	332.9	276	76.2	874	210.8	70.75	14.915
Grøstl S	1.15	206.3	237	342.4	117.2	807.6	223.3	5.02	1.121
Grøstl L	1.15	206.1	236.8	348.4	121.4	801.6	223.9	**35**	**7.855**
JH S	1.15	220.5	253.2	209.6	43.9	940.4	185.4	7.17	1.32
JH L	1.15	222.1	255.2	209.8	44	940.2	185.5	77.75	14.419
Keccak S	1.15	221.7	254.5	215.5	46.5	934.5	188.2	**3.34**	**0.628**
Keccak L	1.15	233	267.6	223.1	49.8	926.9	191.7	55	10.541
Skein S	1.15	300	342.4	308.6	95.2	841.4	218.5	4.34	0.94
Skein L	1.15	269.2	309.3	257.7	66.4	892.3	205.2	60.26	12.36

4.2 Round Three Power, Energy and Processing Time Results

Table 4 and Table 5 give the time, power and energy results for the round two designs and the round three designs respectively. Columns 2, 3 and 4 give the supply data from the power meter, while subsequent columns give the results either obtained from the oscilloscope or calculated using equation 1. We draw attention to the final two columns, the former being the time taken to perform a full calculation of the algorithm and the latter being the energy expended to perform that calculation. In both cases, the lower the value, the better the result. It is shown that for both the round two and round three designs for a digest size of 256 that Grøstl has the highest mean FPGA power but the lowest energy usage due to it having the shortest processing time for both long and short messages. For the 512-bit designs, Keccak has the lowest energy and shortest processing time for short messages, while Grøstl again gives the best results for long messages. A general *rule of thumb* to get the timing or energy results per bit is to divide the result by the message size.

4.3 Area-Time & Area Energy Product

As a method of comparing the designs examined so far, some metrics were employed. The area-time product (ATP) was calculated to get a representation of any speed decrease against the increase in size. This gives a more accurate representation of the cost of each implementation in relation to the overall system. The area-energy product (AEP) was calculated to give a representation of the power increase against the increase in size. The power naturally increases with an increase in area, however the energy costs are reduced due to the calculation being performed faster. This therefore shows the best increase in area for a decrease in power. Graphing the Area-Time and Area-Energy product for the 224/256-bit implementations, as shown in Figures 2, 3, 4 and 5, show that while Blake-32 had the best overall results for short messages, Keccak gives a better figure for long.

For the 384/512-bit implementations, Keccak has the lowest and best Area-Time and Area-Energy product as shown in Figures 6, 7, 8 and 9 for both long and short messages.

4.4 Comparison of Energy Results

The results presented here can be most directly compared against the *round two* results of Kneževiĉ et al. [11], although while a different method was used to calculate the energy, similar hardware was used and the *round two* results presented here in this paper would appear to be similar in most cases. As stated in Section 4, a general rule of thumb to get the timing or energy results per bit is to divide the result by the message size. When the measured *round two* results presented here are converted to their *energy per bit* equivalents, the ranking of the designs from lowest to highest energy usage is in the same order, i.e. Grøstl, Keccak and Skein and differs only for JH and Blake for long 256 *round*

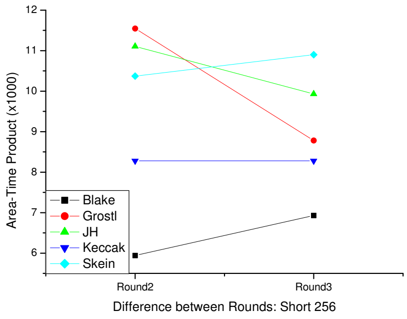

Fig. 2. Area-Time Product Round 2 & 3: Short-256

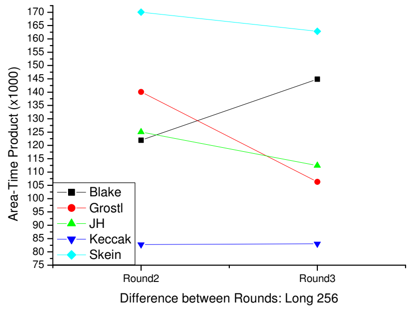

Fig. 3. Area-Time Product Round 2 & 3: Long-256

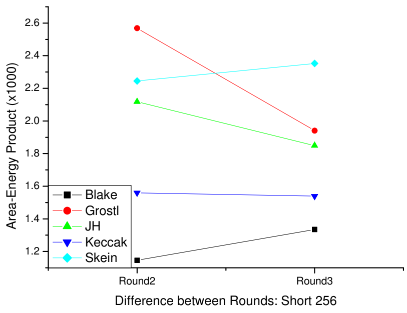

Fig. 4. Area-Energy Product Round 2& 3: Short-256

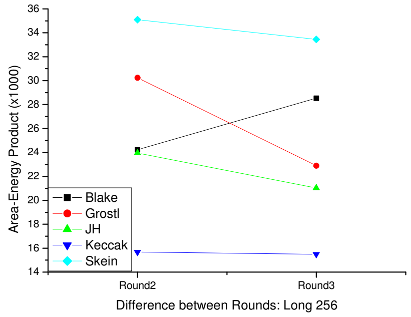

Fig. 5. Area-Energy Product Round 2& 3: Long-256

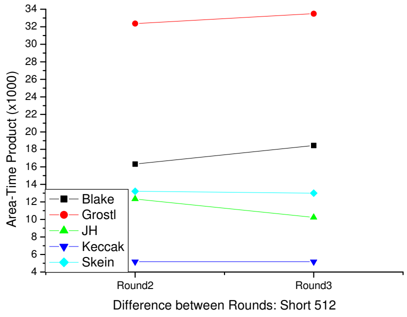

Fig. 6. Area-Time Product Round 2 & 3: Short-512

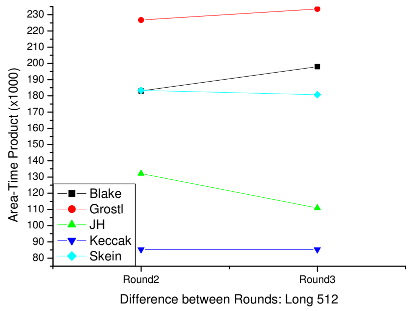

Fig. 7. Area-Time Product Round 2 & 3: Long-512

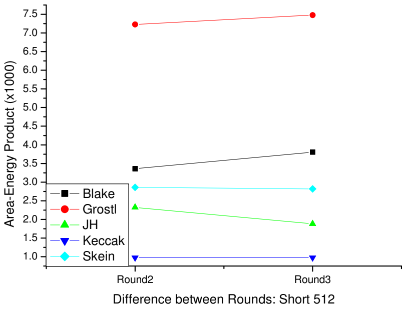

Fig. 8. Area-Energy Product Round 2 & 3: Short-512

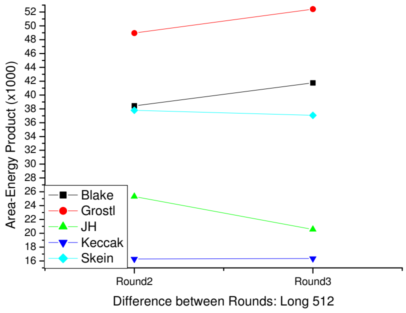

Fig. 9. Area-Energy Product Round 2 & 3: Long-512

two messages. However, the energy per bit results for the hash designs recorded here were found in all cases to be slightly higher than those in [11], although this difference decreases for the longer messages. As such the compared ranking differs slightly more for short messages.

The *round three* results from Kaps et al. [10] cannot be so directly compared against the *round three* results presented in this work, as both a different chip (Spartan-3) and design methodology (constrained implementations) were used.

5 Conclusions

In this paper we have presented measured area, power, energy and iteration time results of the round three (and indeed round two versions of the) SHA-3 hash functions. It was shown that JH has the lowest area of our implementations of the designs. It was also shown that Grøstl has the lowest iteration time and energy usage (while also having the highest mean power) for both the round two and round three implementations for both long and short messages for 256. Groestl also gave lowest and fastest energy and processing time for long messages for 512 while Keccak gave best for short 512 messages. Blake showed the best results for area-time and area-energy product for short 256 messages, while Keccak outperformed in all the other area-time and area-energy categories.

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 06/MI/006.

Special thanks to Neil Hanley for suggestions and advice on implementing the updated Keccak padding.

References

1. NIST. *Secure Hash Standard (FIPS–180-4)*. National Institute of Standards and Technology, April 1995.
2. Kazumaro Aoki, Jian Guo, Krystian Matusiewicz, Yu Sasaki, and Lei Wang. Preimages for Step-Reduced SHA-2. In *Advances in Cryptology ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 578–597. Springer Berlin / Heidelberg, 2009.
3. NIST. Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. *Federal Register*, 72(212):66212–66220, November 2007.
4. NIST. *Announcing the advanced encryption standard (AES)*. National Institute of Standards and Technology, `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`, 2001. FIPS PUB 197.
5. NIST. Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition. *NIST Interagency Report*, 7620, September 2009.
6. NIST. National Institute of Standards and Technology. [docket no.: 070911510751201] Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA3) Family. Federal Register, November 2007.
7. Kris Gaj, Ekawat Homsirikamol, Marcin Rogawski, Rabia Shahid, and Malik Umar Sharif. Comprehensive evaluation of high-speed and medium-speed implementations of five sha-3 finalists using xilinx and altera fpgas. In *The Third SHA-3 Candidate Conference*, 2012.
8. Bernhard Jungk. Evaluation of compact fpga implementations for all sha-3 finalists. In *The Third SHA-3 Candidate Conference*, 2012.
9. Kashif Latif, M Muzaffar Rao, Arshad Aziz, and Athar Mahboob. Efficient hardware implementations and hardware performance evaluation of sha-3 finalists. In *The Third SHA-3 Candidate Conference*, 2012.
10. Jens-Peter Kaps, Panasayya Yalla, Kishore Kumar Surapathi, Bilal Habib, Susheel Vadlamudi, and Smriti Gurung. Lightweight implementations of sha-3 finalists on fpgas. In *The Third SHA-3 Candidate Conference*, 2012.
11. Miroslav Kneževiȉc, Kazuyuki Kobayashiy, Jun Ikegamiy, Shinichiro Matsuoz, Akashi Satoh, Ünal Kocobaş, Junfeng Fan, Toshiro Katashita, Takeshi Sugawarax, Kazuo Sakiyamay, Ingrid Verbauwhede, Kazuo Ohtay, Naofumi Hommax, and Takafumi Aokix. Fair and Consistent Hardware Evaluation of Fourteen Round Two SHA-3 Candidates. In *IEEE Transactions on Very Large Scale Integration Systems*, volume PP, pages 1–13, 2011.
12. Brian Baldwin, Andrew Byrne, Liang Lu, Mark Hamilton, Neil Hanley, Maire O'Neill, and William P. Marnane. FPGA Implementations of the Round Two SHA-3 Candidates. *International Conference on Field Programmable Logic and Applications*, pages 400–407, 2010.
13. IAIK. SHA-3 zoo. `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.
14. Xilinx. *Fast Simplex Link (FSL) Bus*, ds449 (v2.11b) edition, April 2010.
15. Brian Baldwin, Andrew Byrne, Liang Lu, Mark Hamilton, Neil Hanley, Maire O'Neill, and William P. Marnane. A Hardware Wrapper for the SHA-3 Hash Algorithms. In *Signals and Systems Conference (ISSC 2010), IET Irish*, 2010.
16. *National Institute of Advanced Industrial Science and Technology (AIST), Research Center for Information Security (RCIS), Sidechannel Attack Standard Evaluation Board (SASEBO).*