XBX Benchmarking Results January 2012

Christian Wenzel-Benner[1], Jens Gräf[2], John Pham[3], and Jens-Peter Kaps[3]

[1] ITK Engineering AG
Software Center 1, 35037 Marburg, Germany
Christian.Wenzel-Benner@itk-engineering.de,
WWW home page: http://www.itk-engineering.de
[2] LiNetCo GmbH
Hauptstrasse 17a, 35684 Dillenburg, Germany
jgraef@linetco.com,
WWW home page: http://www.linetco.com
[3] George Mason University
4400 University Drive
Fairfax, VA 22030-4444
ece@gmu.edu

1 Introduction

We benchmarked many implementations of all remaining SHA-3 candidate algorithms on several platforms. The benchmarking method used in this report is called XBX, short for eXternal Benchmarking eXtension, an extension of the SUPERCOP-eBASH framework [7] that allows benchmarking small devices. For details on how XBX works, please see [3]. The main sources of candidate implementations were SUPERCOP, the avr-crypto-lib [6] and its derivative arm-crypto-lib [5]. Implementations from sphlib [8] were included in the SUPERCOP package. Several embedded-friendly implementations which were added as of supercop-20120120 have been included separately as most benchmark runs were started before January 20th, 2012.

1.1 Embedded Devices

The microchip doing actual computations in a standard desktop or notebook computer is called processor or central processing unit (CPU). It is useless without additional supporting hardware like volatile random access memory (RAM) to store instructions and data, a permanent storage medium like a hard disk to load instructions into RAM when the power is turned on and a mainboard to connect the CPU to RAM and storage media.

The set of specific instructions a CPU can execute is called instruction set or instruction set architecture (ISA). A specific ISA can be implemented by several CPUs, e.g. the x86 ISA is used in CPUs manufactured by companies like Intel, AMD, VIA. ISAs are modified over time, a Pentium-II implements the x86 ISA of its time and a current Core-i5 implements an updated and extended version of that same ISA (among other things). The specific subset of an ISA implemented by certain CPUs also depends on the target market. Most CPU manufacturers have families of CPUs which implement larger or smaller subsets of that manufacturer's current ISA at different price points. The physical implementation of the CPUs is a microchip comprising silicon, internal metal connections, a housing package and external pins.

In order to put computers into industrial and consumer goods (”embed” them) a CPU, RAM, ROM (permanent storage) and all necessary connections are put onto a single chip. These chips are generally referred to as ”embedded devices”, the smaller ones are often called ”microcontrollers” and control functions in cars, refrigerator or medical equipment. Another important application for small microcontrollers are smart cards and security tokens. Larger embedded devices are often referred to as system-on-chips (SoCs) or digital media processors, depending on their intended field of application. Those are usually found in smartphones, DSL routers and home entertainment products. Some manufacturers have designed their own ISAs from which they derive CPUs to put into their microcontrollers. The Atmel AVR based microcontrollers are examples of this approach. Others choose to license ISAs or even entire CPU designs from a CPU design company. For example, chips based on ARM CPUs are widely used. Different subsets of the ARM ISA are available in old and new, small and large, cheap and expensive embedded devices.

In our work we benchmarked SHA-3 candidate implementations on real world embedded devices (platforms) available off-the-shelf.

2 Contributions

The benchmarking results on most platforms were collected and analyzed by Jens Gräf and Christian Wenzel-Benner. The MSP430 XBX platform was built at George Mason University (GMU) by Margaux McGivern, Fletcher Ta and Elio G Andia under supervision from Jens Peter Kaps. John Pham later finalized the setup and obtained the measurement data.

3 Structure of this Report

Three target platforms have been added to XBX since the 2nd SHA-3 candidate conference: MSP430 (proprietary ISA), LPC1114 (ARM Cortex-M0) and BeagleBoard-xM (ARM Cortex-A8). They are introduced in some detail in sections 6.2, 6.6 and 6.8. Platforms already introduced in previous reports are not described in detail. See [3] for a description of those. A short discussion on measurement error sources is provided in chapter 4, including an explanation for the deviation of performance results between XBX and SUPERCOP for the same CPU type. An overview of the raw results across all considered platforms is provided in **??**. The main result section lists detailed results and conclusions on a per-platform basis.

In chapter 7 we provide a summarized recommendation based on the individual platform winners according to the observations made in chapter 6.

4 Error Sources

This section lists additional error sources for platforms which deviate from the "classic" XBD design described in [3]. Classic XBDs do not have an operating system, they indicate start and end of timing measurements via direct I/O register access from the XBX software and run at or below the speed of the XBH and its timer (16Mhz).

Modified appliances such as DSL routers or NAS devices usually run an operating system so any I/O operation by XBX software has to be passed through the operation system application programming interface and device drivers. Many of these devices run at below 3.3V so voltage level shifting and in some cases optical signal pickup and amplification were applied. This may have introduced asymmetrical delay on the XBD-XBH timing signal path. Since it is usually not practical to change the CPU clock frequency of such appliances they were run at their stock clock speeds, typically beyond 100MHz. A 500ns asymmetry between the signals "start measurement" and "end measurement" would therefore result in an error of more than 50 cycles.

4.1 Speed

Asymmetrical circuit delay on targets with optical pickups can result in speed measurement results which are either too low or too high.

Spontaneous process switching under operating systems (usually embedded Linux) will put measurement results off in an unpredictable manner. This is the same effect present in standard SUPERCOP data and mitigated by increasing the sample size. The severity of the effect depends on background services like scheduled tasks and operating system activity.

Process switching under embedded Linux due to I/O calls (user / kernel space) is a related issue.

4.2 Size

As per SUPERCOP methodology fresh pseudo-random data is applied to the hash function input for each hash operation. Depending on the hash implementation parts of the input and/or state ends up on the stack. Pseudo-random data on the stack can put the stack measurement off by one or more bytes since it depends on recognition of a known bit pattern. The stack test pattern is byte based and changes between hash operations. Several operations are repeated and the maximum stack usage is reported.

5 XBX Results over all Platforms

This chapter presents the XBX results ”fastest”, ”smallest RAM” and ”smallest ROM” over all benchmarked platforms. Results are scaled on a per-platform basis to the best SHA-3 candidate. The plots are similar in structure to the ones presented in [1]. 256-bit and 512-bit output versions are not distinguished.

Results for Skein512256 are off in some cases due to a lack of optimized implementations. In these cases the results from the almost identical Skein512512 should be considered instead.

5.1 Speed

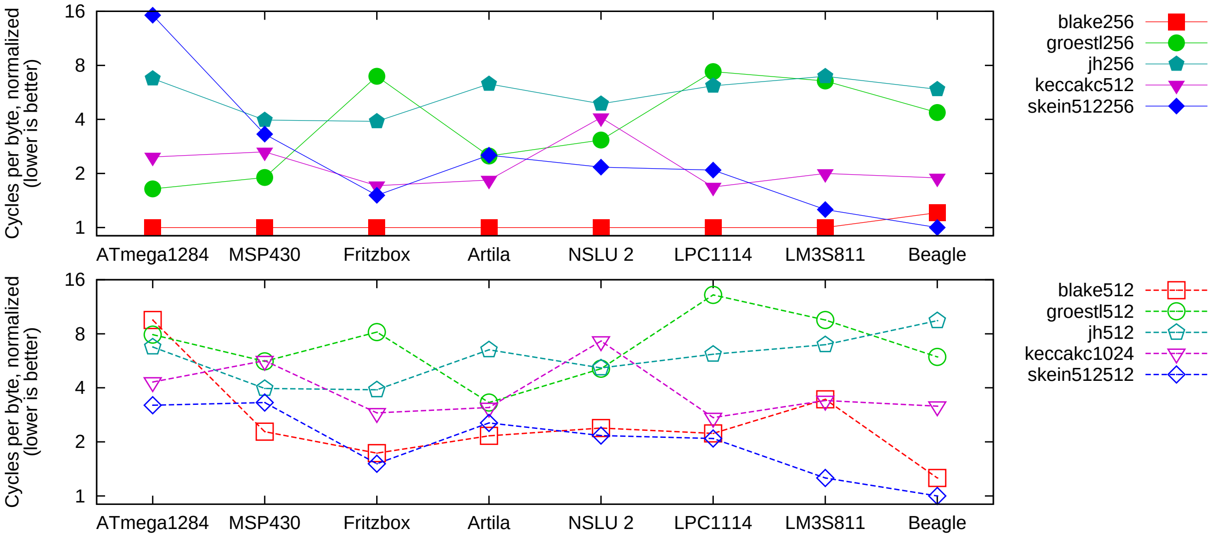

Fig. 1. Cycles per Byte, relative, 256-bit output size (top), 512-bit output size (bottom)

Figure 1 shows that BLAKE excels at 256-bit, Skein is very strong on modern, high speed ARM cores at both 256 and 512-bit. Skein and Keccak aren’t to widely separated otherwise. JH and Grøstl trail behind, but Grøstl does well on the 8 and 16-bit platforms at 256-bit.

5.2 RAM

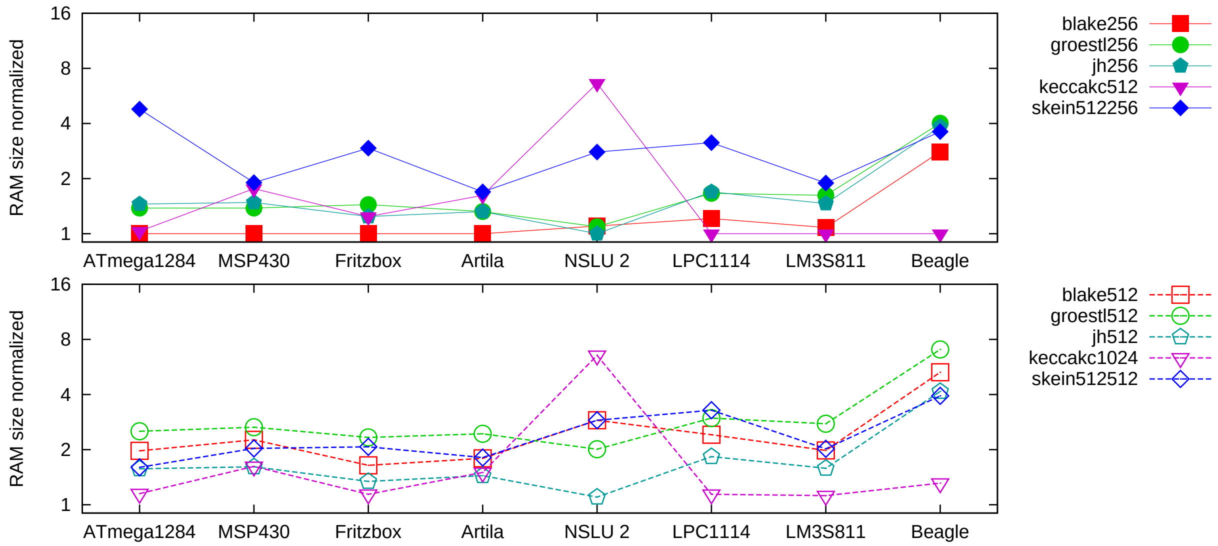

Fig. 2. Smallest RAM usage, relative, 256-bit output size (top), 512-bit output size (bottom)

Figure 2 shows that BLAKE is the smallest (RAM) at 256-bit by a narrow margin. Keccak has very low RAM consumption on many platforms at 256-bit and on almost all at 512-bit, it also beats BLAKE at 512-bit. JH shows very low RAM usage, too.

5.3 ROM

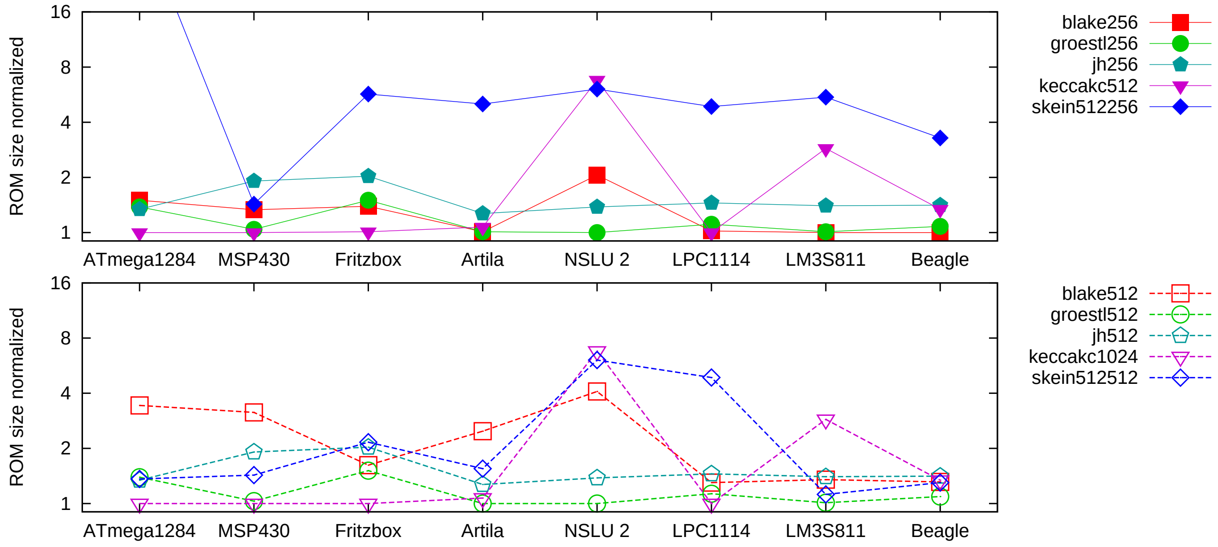

Fig. 3. Smallest ROM usage, relative, 256-bit output size (top), 512-bit output size (bottom)

Figure 3 shows that Grøstl is small across many platforms doesn't have size issues on any platform. Keccak excels on small platforms, is often very small but on some platforms rather large. JH has a consistent ROM consumption but not as low as Grøstl.

6 XBX Result Details per Platform

The following pages contain aggregated results of the benchmarking runs per platform.

The result chapters a labeled as [CPU type] (CPU width): [microchip name]. Devices which use the same CPU type show similar performance at low frequencies (up to 50MHz), above that the memory subsystem designed by the microchip manufacturer has an increasing impact on the results.

A throughput over area plot presents an overview of the available tradeoffs between speed and size on the platform. The data used was collected during the ”try” phase, so the only input length measured is 1536 bytes. Memory usage in terms of ROM, static RAM, and stack RAM is determined for every triple of compiler-options-implementation. Using the total RAM and ROM usage numbers a notion of occupied silicon area is calculated as $area = ROMusage + 4 * RAMusage$ based on the number of transistors needed to implements SRAM and Flash ROM in hardware [9].

For each SHA3-candidate the pareto frontier for throughput over area is constructed and plotted. Layout and purpose of the plot is very similar to [4], the logarithmic scaling and the color coding of the algorithms is similar to the SUPERCOP speed overview plots [7]. For each algorithm the 256-bit version is plotted using solid lines and solid symbols, the 512-bit version using dashed lines and outlined symbols. The tables contain the results for performance and memory usage. Fastest and smallest implementations with respect to both ROM and RAM usage are provided. The columns ”long” and ”1536” list the speed results in cycles per byte at the respective message lengths.

Due to limited space the compiler used, build options and implementation name are not listed. This information can be looked up on the XBX website [2] using the run number provided in the table. The size tables (”smallest RAM/ROM”) are ordered by the amount of RAM/ROM required by the smallest implementation on the platform. The “try” speed at 1536 byte input length is also included. Readers interested in the arguably more realistic input lengths of 64 or 512 bytes should look at the XBX website and sort the result tables there by the criteria of their choice.

6.1 AVR (8-bit): Atmel ATmega1284P

Hardware platform

8-bit RISC microcontroller manufactured by Atmel. Since the 1284P results are virtually the same results as the ATmega1281 results, we report only the former in this work. See `http://www.atmel.com/dyn/products/product_card.asp?part_id=4331` for details.

This is a classic XBD with a fully working GCC toolchain and no issues were observed. Timing results are considered very reliable.

Results

Since 8-bit CPUs are used in applications which are very cost sensitive and usually have no high-bandwidth connections we consider memory footprint as the most important aspect of a SHA-3 candidate on this kind of platform. Throughput differences are considered unimportant up to at least an order of magnitude. Applications on 8-bit CPUs will rely on 256-bit hashes whenever possible.

XBX Team's choice: Keccak, with BLAKE as runner-up, Grøstl and Skein are tied in 3rd position.

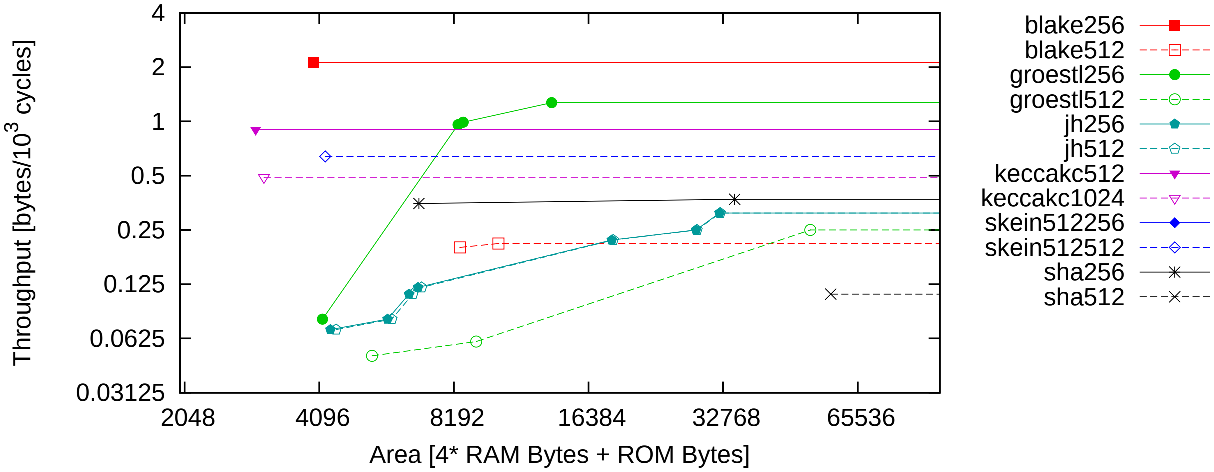

Fig. 4. Throughput over area on ATmega1284P

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	451	471	2764	303	302
groestl256	739	785	9506	1013	302
keccakc512	1115	1115	1848	276	301
skein512512	1444	1566	2508	429	302
keccakc1024	1945	2038	1848	308	302
jh256	3043	3177	22998	2293	302
jh512	3043	3177	22998	2325	302
groestl512	3573	4023	27492	5964	302
blake512	4304	4670	6734	904	302
skein512256	6870	7165	105662	1390	302
sha256	2578	2687	33676	708	302
sha512	8041	8728	48492	2141	302

Table 1. Fastest on atmega1284p_16mhz

Algorithm	RAM	ROM	cpb 1536	Run
blake256	267	3434	1617	302
keccakc512	276	1848	1115	302
keccakc1024	308	1848	2038	302
groestl256	368	3528	20019	302
jh256	388	4950	9191	302
jh512	420	4950	9191	302
skein512512	427	2524	1566	302
blake512	525	6350	5043	302
groestl512	672	3604	29495	302
skein512256	1279	96346	7210	302
sha256	359	77720	2768	302
sha512	2118	50810	8759	302

Table 2. Smallest (RAM) on atmega1284p_16mhz

Algorithm	ROM	RAM	cpb 1536	Run
keccakc512	1848	276	1115	302
keccakc1024	1848	308	2038	302
jh256	2470	467	14288	302
jh512	2470	499	14288	302
skein512512	2508	429	1566	302
groestl256	2550	710	16839	302
groestl512	2574	1014	24959	302
blake256	2764	303	471	302
blake512	6342	911	4913	302
skein512256	96346	1279	7210	302
sha256	3816	757	2880	302
sha512	48492	2141	8728	302

Table 3. Smallest (ROM) on atmega1284p_16mhz

6.2 MSP430 (16-bit): Texas Instruments MSP430FG4618

Hardware platform

16-bit microcontroller manufactured by Texas Instruments. Optimized for extremely low power applications, by now a few years old. This XBX target was originally built at GMU. See `http://focus.ti.com/paramsearch/docs/parametricsearch.tsp?familyId=912&sectionId=95&tabId=1528&family=mcu` for details.

Results

Since 16-bit CPUs are used in applications which are cost sensitive and usually have no high-bandwidth connections we consider memory footprint as the most important aspect of a SHA-3 candidate on this kind of platform. Throughput differences are considered unimportant up to at least an order of magnitude. Applications on 16-bit CPUs will rely on 256-bit hashes in most cases. Grøstl is the smallest candidate at 256-bit but BLAKE is only slightly larger yet roughly 16 times faster at that area consumption. Keccak is the next smallest, both at 256 and 512-bit and ranks in speed between Grøstl and BLAKE.
XBX Team's choice: Grøstl and BLAKE are tied winners, with Keccak in 3rd position.

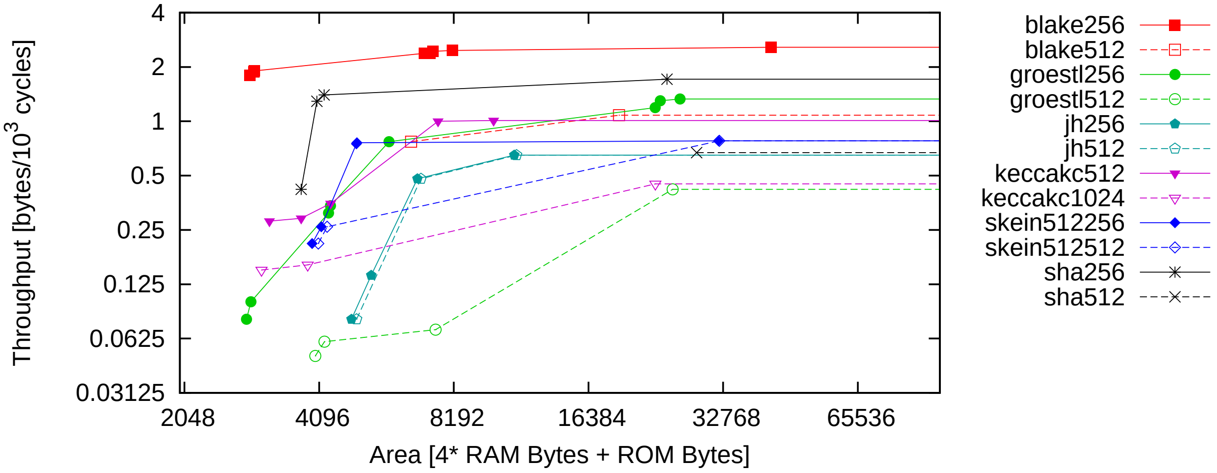

Fig. 5. Throughput over area on MSP430

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	373	389	31982	2488	318
groestl256	707	752	21482	1196	318
blake512	851	926	13746	1360	318
keccakc512	984	985	7642	604	318
skein512256	1236	1289	26944	1268	318
skein512512	1236	1290	26944	1300	317
jh256	1480	1542	9676	378	318
jh512	1480	1542	9676	410	316
groestl512	2095	2358	15544	2430	318
keccakc1024	2115	2217	18178	1234	318
sha256	560	584	21984	930	318
sha512	1379	1496	19848	2188	318

Table 4. Fastest on msp430fg4618_4mhz

Algorithm	RAM	ROM	cpb 1536	Run
blake256	240	1974	526	318
groestl256	332	2104	25483	318
jh256	354	12364	1908	318
jh512	386	12364	1908	318
keccakc1024	390	1916	8344	318
keccakc512	422	1916	4556	318
skein512256	456	2318	3838	318
skein512512	488	2318	3838	318
blake512	542	4408	1294	318
groestl512	636	2144	37691	318
sha256	470	1854	2406	318
sha512	2188	19848	1496	318

Table 5. Smallest (RAM) on msp430fg4618_4mhz

Algorithm	ROM	RAM	cpb 1536	Run
keccakc512	1404	442	3537	318
keccakc1024	1404	410	6481	318
groestl512	1448	642	18649	318
groestl256	1458	340	12331	318
blake256	1864	251	554	318
skein512256	2006	486	4780	318
skein512512	2006	518	4781	318
jh256	2688	538	12384	318
jh512	2688	570	12384	318
blake512	4408	542	1294	318
sha256	1854	758	2406	318
sha512	19848	2188	1496	318

Table 6. Smallest (ROM) on msp430fg4618_4mhz

6.3 MIPS (32-bit): Texas Instruments AR7

Hardware platform

The AR7 is a 32-bit, MIPS based system-on-chip (SoC) manufactured by Texas Instruments. It is not sold to end customers and specifications are not generally public. However, there is a Linux kernel available. See `http://www.linux-mips.org/wiki/AR7`. The XBX team got their hands on an AR7 by modifying a Fritzbox DSL router `http://www.avm.de/de/Produkte/FRITZBox/FRITZ_Box_Fon_WLAN/index.php`. This is a Linux based XBD with a fully working GCC toolchain. Timing results are obtained using 4 times as many samples as standard XBX but are still considered noisy with worst case relative interquartile ranges (WRIR) of up to 27%.

Results

The main application of AR7 chips is DSL routers with Linux operating system. High-bandwidth Ethernet connections with 100Mbps or more require maximum throughput from a SHA-3 candidate while the fully featured operating system already present makes memory footprint below the megabyte range a non-issue. Both 256-bit and 512-bit hashes are likely to be used.

XBX team's choice: BLAKE, with Skein as runner-up and Keccak in 3rd place.

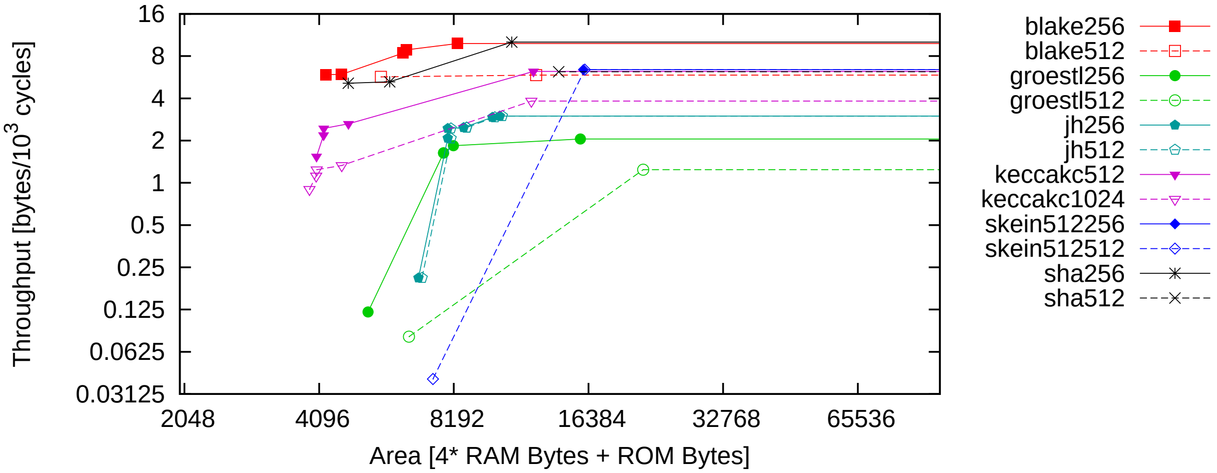

Fig. 6. Throughput over area on FRITZBox

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	77	101	6083	568	311
skein512256	116	155	11638	1080	309
skein512512	116	156	11638	1112	309
keccakc512	132	160	8663	920	309
blake512	133	170	8390	1032	309
keccakc1024	223	261	8664	888	311
jh256	300	333	8577	452	309
jh512	300	334	8577	484	311
groestl256	535	487	12698	755	311
groestl512	629	807	14922	1700	311
sha256	68	99	9568	368	309
sha512	124	161	7990	1520	309

Table 7. Fastest on fritzbox-7170

Algorithm	RAM	ROM	cpb 1536	Run
blake256	340	3302	179	311
keccakc1024	388	27933	6393	311
keccakc512	420	27932	3491	311
jh256	423	27533	7584	311
jh512	456	27533	7585	311
groestl256	488	4983	20402	311
blake512	556	5870	535	311
skein512512	704	5050	24165	311
groestl512	792	5015	30028	311
skein512256	996	19818	308	311
sha256	368	9568	99	311
sha512	1520	7990	161	311

Table 8. Smallest (RAM) on fritzbox-7170

Algorithm	ROM	RAM	cpb 1536	Run
keccakc1024	2043	464	1115	311
keccakc512	2058	496	648	311
blake256	2846	348	169	311
groestl256	3063	552	8550	311
groestl512	3079	856	12743	311
blake512	3310	580	175	311
jh256	4157	668	4822	311
jh512	4157	700	4822	311
skein512512	4418	736	24236	311
skein512256	11638	1080	155	311
sha256	2374	880	190	311
sha512	7984	1520	161	311

Table 9. Smallest (ROM) on fritzbox-7170

6.4 ARM 920T (32-bit): Atmel AT91RM9200

Hardware platform
Artila M501, a single PCB computer. Used, among other things, for automation and remote data logging applications. See `http://www.artila.com/p_sbc.html#m_501` for details. This is a Linux based XBD with a fully working GCC 3.3 toolchain, GCC 4.x is not yet supported. Timing results are obtained using standard XBX sample sizes and are considered reliable with worst case relative interquartile ranges (WRIR) of below 4% for any SHA-3 candidate.

Results
Memory and throughput requirements for a SHA-3 candidate on this platform are the same as for the AR7. Under these conditions only throughput matters for the five finalists. BLAKE dominates the throughput category in both 256-bit and 512-bit output sizes. Keccak comes in second fastest at 256-bit, Grøstl-256 runs at roughly the throughput of Skein, with the latter offering both 256-bit and 512-bit output size at the same throughput in addition to better throughput for short messages.
XBX team's choice: BLAKE, with Keccak and Skein tied as runner-up.

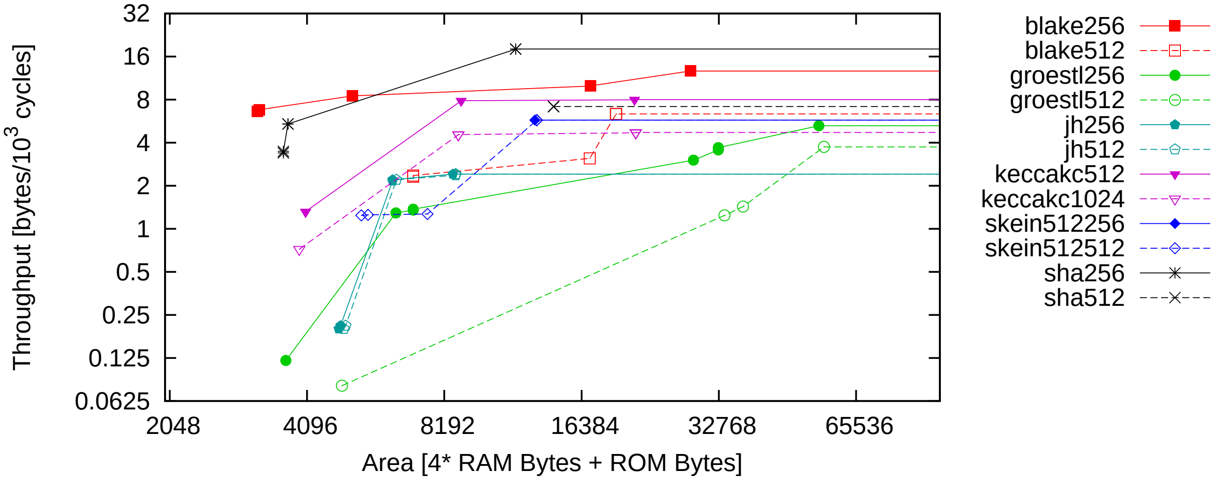

Fig. 7. Throughput over area on AT91RM9200

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	62	78	25488	716	313
keccakc512	114	124	17540	968	313
blake512	134	157	15188	1076	313
groestl256	155	190	49612	1172	313
skein512256	157	173	10244	676	312
skein512512	158	173	10244	708	313
keccakc1024	193	212	17540	1000	313
groestl512	205	267	49612	1524	313
jh256	391	414	7112	360	313
jh512	404	414	7112	392	313
sha256	47	55	10100	412	312
sha512	122	139	9900	1080	313

Table 10. Fastest on artila_m501

Algorithm	RAM	ROM	cpb 1536	Run
blake256	272	3952	604	313
groestl256	360	3532	24149	313
jh256	360	7112	414	313
jh512	392	7112	414	313
keccakc1024	408	21524	2793	313
keccakc512	440	21524	1541	313
skein512256	460	46820	606	313
blake512	488	5052	423	313
skein512512	491	46820	597	312
groestl512	663	3572	35546	312
sha256	308	38504	524	313
sha512	1080	9900	139	313

Table 11. Smallest (RAM) on artila_m501

Algorithm	ROM	RAM	cpb 1536	Run
groestl512	2036	712	12541	313
blake256	2052	284	150	313
groestl256	2064	404	8503	313
keccakc1024	2172	664	1719	313
keccakc512	2184	632	943	313
jh256	2584	556	5002	313
jh512	2584	588	5003	313
skein512512	3148	560	807	313
blake512	5052	488	423	313
skein512256	10244	676	173	313
sha256	1614	792	287	313
sha512	9900	1080	139	313

Table 12. Smallest (ROM) on artila_m501

6.5 ARMv5TE (32-bit): Intel XScale IXP420

Hardware platform

32-bit, ARMv5TE based Intel chip. We used a NAS server (NSLU2) and changed the firmware to `http://www.nslu2-linux.org/wiki/Main/HomePage`. This is a Linux based XBD with a fully working GCC toolchain, however, since it is big endian quite a few C implementations do not work correctly. The ranking of the candidates may be distorted by this effect and the chip's market penetration is limited. Timing results are obtained using standard XBX sample sizes and are considered reliable with worst case relative interquartile ranges (WRIR) of below 7% for any SHA-3 candidate.

Results

As with other Linux based platforms memory footprint as required by the SHA-3 finalists is unimportant. Only throughput matters.
XBX team's choice: BLAKE, with Skein as runner-up, Grøstl as 3rd.

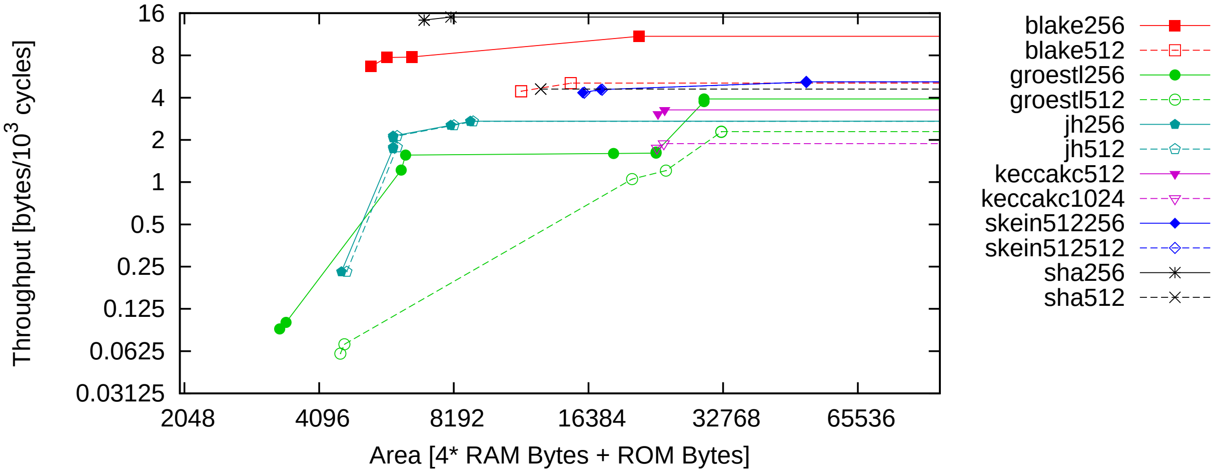

Fig. 8. Throughput over area on IXP420

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	70	91	13160	2028	308
skein512256	152	192	28764	5352	307
skein512512	152	193	28764	5384	307
blake512	167	197	10392	1140	307
groestl256	215	255	27968	432	308
keccakc512	286	306	14640	2408	307
jh256	342	368	6904	504	307
groestl512	358	437	27968	1124	307
jh512	361	368	6904	536	307
keccakc1024	510	532	14640	2376	307
sha256	53	66	6108	492	307
sha512	210	216	6668	1536	307

Table 13. Fastest on nslu2-openwrt

Algorithm	RAM	ROM	cpb 1536	Run
jh256	328	4680	566	308
groestl256	356	3408	25195	308
jh512	360	4680	566	308
blake256	360	6456	276	308
groestl512	660	3416	37218	308
skein512256	915	15684	397	307
skein512512	948	15684	398	308
blake512	948	12452	392	308
keccakc1024	2144	17096	1109	308
keccakc512	2176	17096	616	308
sha256	292	5864	70	308
sha512	1536	6668	217	308

Table 14. Smallest (RAM) on nslu2-openwrt

Algorithm	ROM	RAM	cpb 1536	Run
groestl512	1800	692	16089	308
groestl256	1808	384	10856	308
jh256	2480	528	4433	308
jh512	2480	560	4432	308
blake256	3716	408	149	308
blake512	7368	1056	225	308
skein512256	10932	1244	230	308
skein512512	10932	1276	230	308
keccakc512	12216	2804	326	308
keccakc1024	12216	2772	572	308
sha256	5864	292	70	308
sha512	6668	1536	217	308

Table 15. Smallest (ROM) on nslu2-openwrt

6.6 ARM Cortex-M0 (32-bit): NXP LPC1114

Hardware platform

32-bit ARM Cortex-M0 based microcontroller. Modern CPU design from the current ARM lineup, targeted for extremely low cost 32-bit microcontroller applications up to around 50MHz. Executes thumb instructions only (as opposed to the M3's thumb2) which gives good code density but weak performance per clock compared to larger ARM Cortex designs. See `http://ics.nxp.com/products/lpc1000/all/~LPC1114/` for details. This is a classic XBD with a fully working GCC toolchain but all generated binaries so far fail at optimization levels 2 and 3 for reasons as yet undetermined. Best case speed of many candidates may improve once this issue is fixed. Timing results are considered very reliable.

Results

This the uncompromising low cost ARM Cortex CPU. Memory footprint is the most important factor. However, as opposed to the 8-bit smart card situation, we expect 512-bit hash sizes to be required. BLAKE is the smallest candidate available at 256-bit hash size, Keccak is the smallest at 512-bit. Grøstl is the 3rd smallest at 256-bit but at a large throughput penalty versus BLAKE and Keccak. At 512-bit hash size both BLAKE and Keccak are smaller than Grøstl.
XBX team's choice: BLAKE, with Keccak as runner-up, Grøstl and JH are tied on 3rd position.

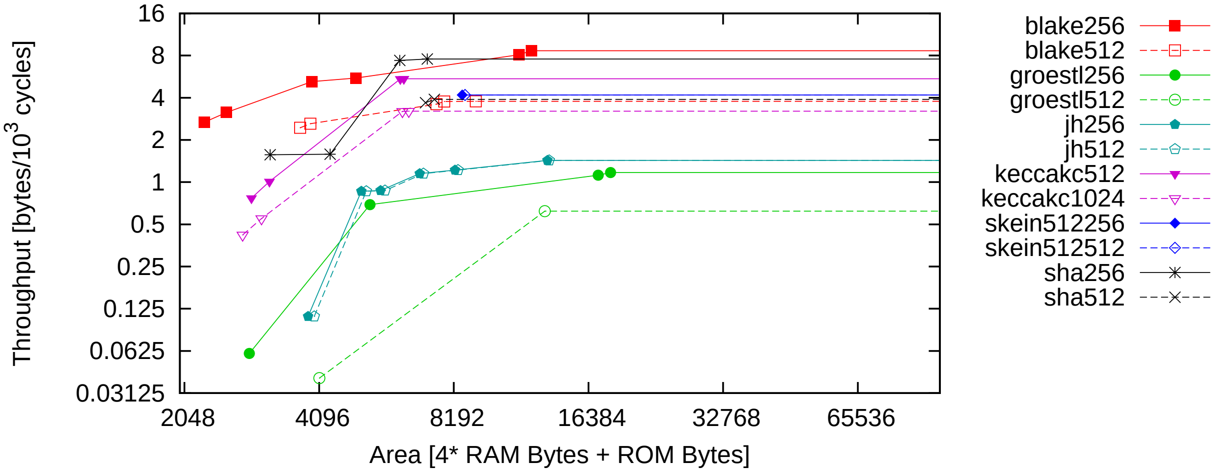

Fig. 9. Throughput over area on LPC1114

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	109	115	9124	772	319
keccakc512	183	183	5312	256	319
skein512256	228	238	5516	760	319
skein512512	228	238	5516	792	319
blake512	243	265	5876	824	315
keccakc1024	298	311	5248	312	319
jh256	671	701	10624	660	319
jh512	671	701	10624	692	319
groestl256	805	856	15184	792	319
groestl512	1437	1614	9784	824	315
sha256	126	132	5964	296	319
sha512	237	257	4116	824	319

Table 16. Fastest on lpc1114-evb

Algorithm	RAM	ROM	cpb 1536	Run
keccakc512	232	5296	183	319
keccakc1024	264	5232	312	319
blake256	280	1152	372	319
groestl256	384	2324	44896	319
jh256	392	5308	869	319
jh512	424	5308	869	319
blake512	560	1476	409	319
groestl512	688	2352	66166	319
skein512256	728	8920	400	319
skein512512	760	8920	400	319
sha256	252	5196	135	319
sha512	760	9744	937	319

Table 17. Smallest (RAM) on lpc1114-evb

Algorithm	ROM	RAM	cpb 1536	Run
keccakc512	1132	440	1295	319
keccakc1024	1132	408	2372	319
blake256	1152	280	372	319
groestl256	1260	400	17497	319
groestl512	1284	704	26109	319
blake512	1476	560	409	319
jh256	1644	556	9293	319
jh512	1644	588	9293	319
skein512256	5516	760	238	319
skein512512	5516	792	238	319
sha256	1216	780	634	319
sha512	3916	792	270	319

Table 18. Smallest (ROM) on lpc1114-evb

6.7 ARM Cortex-M3 (32-bit): Texas Instruments LM3S811

Hardware platform

Modern CPU design from the current ARM lineup, targeted for microcontroller applications up to 120MHz. This is a very popular design, powerful yet inexpensive and licensed by many semiconductor manufacturers for a large variety of chips. See `http://www.ti.com/product/lm3s811` for details. This is a classic XBD with a fully working GCC toolchain and no issues were observed. Timing results are considered very reliable.

Results

Due to the large variety of Cortex-M3 based chips we present two choices for this platform. One is purely memory footprint based as for the Cortex-M0 and acknowledges the fact that Cortex-M3 based microcontrollers with 8KiB of ROM and 2KiB of RAM running at 20MHz are commercially available. The other is purely throughput based and considers the priorities of applications using Cortex-M3 based chips with 1MiB of ROM, 128KiB of RAM and 100Mbps Ethernet.

For low memory footprint the choice is BLAKE, then Grøstl and Skein at 256-bit hash size For 512-bit hashes the order changes to Skein, then BLAKE and Grøstl. XBX team's low cost choice: BLAKE, with Skein and Grøstl tied as runner-up.

In high-throughput applications BLAKE is the fastest at 256-bit hash size but only the third fastest at 512-bit hash size, although it is almost tied with the second fastest, Keccak, at this output size. Skein delivers high throughput at both sizes, narrowly trailing the faster BLAKE at 256-bit while maintaining a comfortable lead on the next fastest 512-bit hash, Keccak. XBX team's performance choice: BLAKE and Skein are tied winners, with Keccak in 3rd place.

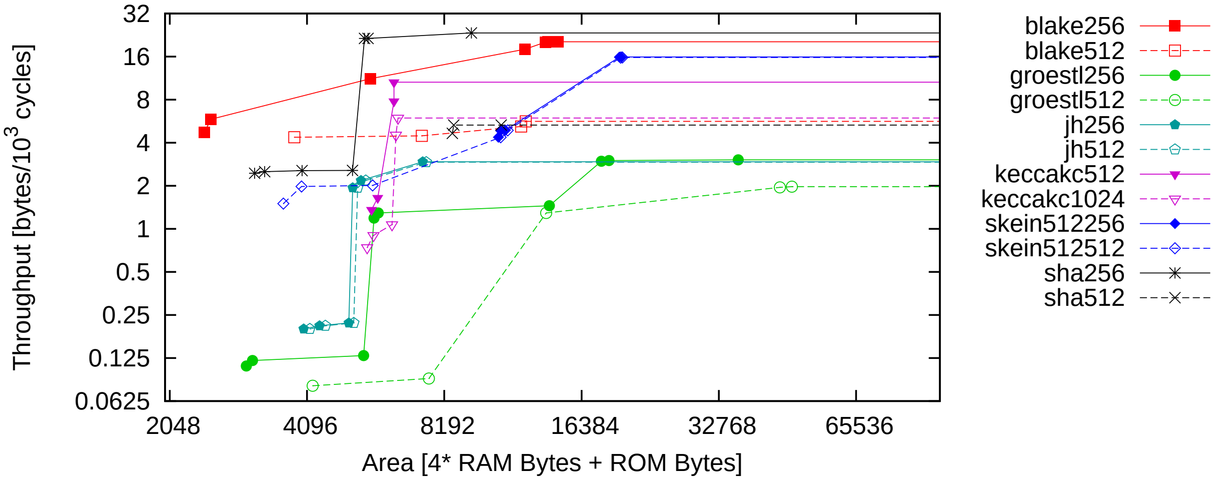

Fig. 10. Throughput over area on LM3S811

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
blake256	47	49	12496	508	320
skein512256	59	62	17924	524	320
skein512512	59	63	17924	556	320
keccakc512	94	94	5316	260	320
keccakc1024	160	167	5316	292	320
blake512	162	177	8768	916	304
groestl256	308	328	32596	884	320
jh256	327	340	5588	440	320
jh512	327	340	5588	472	320
groestl512	449	507	43700	916	304
sha256	41	42	7636	464	320
sha512	173	187	7252	916	320

Table 19. Fastest on lm3s811-evb

Algorithm	RAM	ROM	cpb 1536	Run
keccakc512	260	5316	94	320
blake256	280	1320	210	320
keccakc1024	292	5316	167	320
jh256	380	3640	514	320
jh512	412	3640	514	320
groestl256	420	1336	8871	320
skein512256	492	17860	62	320
blake512	516	1776	228	320
skein512512	524	17860	63	320
groestl512	720	1336	13134	320
sha256	300	4276	46	320
sha512	916	4864	215	320

Table 20. Smallest (RAM) on lm3s811-evb

Algorithm	ROM	RAM	cpb 1536	Run
blake256	1320	280	210	320
groestl256	1336	420	8871	320
groestl512	1336	720	13134	320
skein512512	1476	540	664	320
blake512	1776	516	228	320
jh256	1852	544	5020	320
jh512	1852	576	5020	320
keccakc512	3804	468	735	320
keccakc1024	3804	436	1345	320
skein512256	7228	884	228	320
sha256	1208	772	406	320
sha512	4864	916	215	320

Table 21. Smallest (ROM) on lm3s811-evb

6.8 ARM Cortex-A8 (32-bit + SIMD): Texas Instruments DM3730

Hardware platform

ARM Cortex-A8 based ”Digital Media Processor”. Modern CPU design from the current ARM lineup, targeted for applications up to beyond 1GHz. Superscalar and comes with SIMD (NEON vector unit). The DM3730 also comprises a TI TMS320C64x+ DSP core, which we are currently not benchmarking. See `http://www.ti.com/product/dm3730` for details. We use a BeagleBoard-xM `http://beagleboard.org/hardware-xM` as XBD. This is a Linux based XBD with a fully working GCC toolchain. Timing results are obtained using 16 times as many samples as standard XBX but are still considered noisy with worst case relative interquartile ranges (WRIR) of up to 16%. SUPERCOP numbers are available for two different Cortex-A8 chips at `http://bench.cr.yp.to/results-sha3.html`, however there are no SUPERCOP results for the BeagleBoard-xM. Due to different clock rates, memories, caching strategies and operating systems the numbers are not directly comparable even for assembly implementations.

Results

As with all Linux based platforms only throughput matters due to large memories and fast interfaces. The fastest candidates on the Cortex-A8 all use the NEON vector unit. Skein is the fastest candidate for both 256-bit and 512-bit output. BLAKE trails it by about 25% for both hash sizes. Keccak trails BLAKE by 50% for 256-bit and 150% for 512-bit output size.
XBX team’s choice: Skein, with BLAKE as runner-up, Keccak is 3rd.

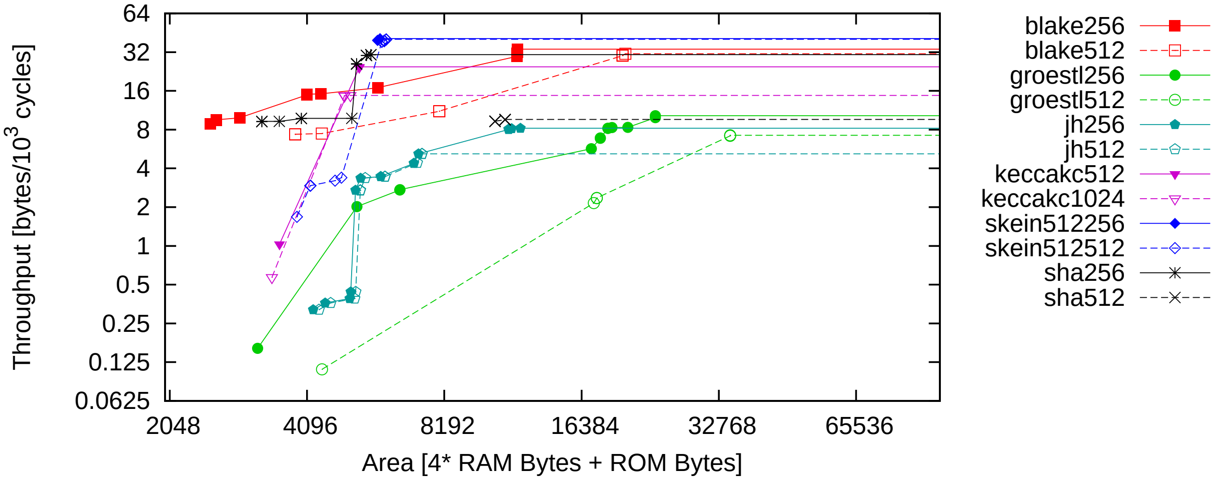

Fig. 11. Throughput over area on BeagleBoard-xM

Algorithm	cpb long	cpb 1536	ROM	RAM	Run
skein512256	19	24	4304	404	306
skein512512	19	24	4352	440	305
blake256	23	29	7790	1013	306
blake512	24	32	12020	2104	306
keccakc512	36	40	4916	104	306
keccakc1024	60	67	4540	140	306
groestl256	83	97	19128	1160	306
jh256	112	121	9812	556	305
groestl512	113	138	28640	1512	305
jh512	180	192	5528	448	306
sha256	27	32	4360	324	305
sha512	91	104	4977	1540	306

Table 22. Fastest on beagleboard_xm

Algorithm	RAM	ROM	cpb 1536	Run
keccakc512	100	5100	41	306
keccakc1024	131	4588	68	305
blake256	280	1472	104	306
skein512256	361	5613	27	306
jh256	384	3832	297	306
skein512512	393	5613	27	306
groestl256	401	3208	18874	306
jh512	416	3832	296	306
blake512	529	4101	387	306
groestl512	705	3289	27924	306
sha256	300	4060	38	306
sha512	1516	4520	107	306

Table 23. Smallest (RAM) on beagleboard_xm

Algorithm	ROM	RAM	cpb 1536	Run
blake256	1296	304	112	306
groestl256	1400	448	6147	306
groestl512	1412	752	9064	306
skein512512	1688	551	594	306
blake512	1700	540	135	306
keccakc512	1736	457	966	306
keccakc1024	1736	424	1758	306
jh256	1824	600	3170	306
jh512	1824	632	3170	306
skein512256	4248	400	25	306
sha256	1158	816	151	306
sha512	4520	1516	107	306

Table 24. Smallest (ROM) on beagleboard_xm

7 Conclusion

BLAKE is our first choice on 7 out of 9 analyzed platforms with the Cortex-M3 counting as two analyzed platforms as it is quite versatile and we made two choices based on low cost and high throughput scenarios. On the remaining two platforms BLAKE takes second place. BLAKE appears to be the most suitable SHA-3 candidate for embedded platforms because it is scalable and balanced, fast and small.

Skein is strong on high-throughput platforms and therefore should be the first choice on fast modern ARM Cortex designs, especially when the NEON SIMD extension is available. It comes in second on 4 other platforms and claims one 3rd place. Overall we consider Skein to be the second best choice for SHA-3 on embedded platforms. It is very fast on fast CPUs and faster than BLAKE at 512-bit on more than half the platforms. It is also relatively small for 512-bit output.

With 1 first place, 2 second places and 4 third places Keccak comes in third in the overall picture. It is the smallest candidate on our 8-bit platform and among the smallest on the other size sensitive platforms. Concerning speed it can claim rank three, sometimes even two on most platforms.

Table 7 list the rankings achived by the candidates according to chapter 6 together with a simple score by awarding three points for every first place, two points for every second place and one point for every time the candidate was ranked third.

SHA-3 Candidate	Ranked 1st	Ranked 2nd	Ranked 3rd	Score
BLAKE	7	2	0	25
Keccak	1	2	4	11
Grøstl	1	1	3	8
JH	0	0	1	1
Skein	2	4	1	15

Table 25. Overall SHA-3 candidate ranking

While all five finalists do work on all analyzed platforms Grøstl and JH can not threaten the three in the top group when it comes to embedded platforms. Keccak excels at the low end, Skein at the high end and BLAKE works very well across the board.

8 Appendix

References

1. `http://eprint.iacr.org/2010/536.pdf` Xu Guo, Sinan Huang, Leyla Nazhandali and Patrick Schaumont. ”On The Impact of Target Technology in SHA-3 Hardware Benchmark Rankings”
2. `http://xbx.das-labor.org/trac/wiki` Christian Wenzel-Benner and Jens Gräf. ”XBX: eXternal Benchmarking eXtension Web Page”
3. Christian Wenzel-Benner and Jens Gräf. ”XBX: eXternal Benchmarking eXtension for the SUPERCOP Crypto Benchmarking Framework” in: S. Mangard and F.-X. Standaert (Eds.): CHES 2010, LNCS 6225, pp. 294-305, 2010.
4. Ekawat Homisirikamol, Marcin Rogawski and Kris Gaj. ”Throughput vs. Area Trade-offs in High-Speed Architectures of Five Round 3 SHA-3 Candidates Implemented Using Xilinx and Altera FPGAs” in: Lecture Notes in Computer Science, 2011, Volume 6917, Cryptographic Hardware and Embedded Systems - CHES 2011, pp. 491-506
5. Daniel Otte. ARM-Crypto-Lib http://www.das-labor.org/wiki/ARM-Crypto-Lib/en, accessed 25 November 2011.
6. Daniel Otte. AVR-Crypto-Lib http://www.das-labor.org/wiki/AVR-Crypto-Lib/en, accessed 25 November 2011.
7. Daniel J. Bernstein and Tanja Lange (editors). eBACS: ECRYPT Benchmarking of Cryptographic Systems. http://bench.cr.yp.to, accessed 25 November 2011, supercop-20111120.tar.bz2
8. Projet RNRT SAPHIR: sphlib http://www.saphir2.com/sphlib/, accessed 25 November 2011.
9. Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic: *Digital Integrated Circuits*, Prentice Hall, 2002.