Hardware performance of eStream phase-III stream cipher candidates

T. Good and M. Benaissa

Department of Electrical & Electronic Engineering,
University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK
{t.good, m.benaissa} @ sheffield.ac.uk

Abstract. This paper presents hardware implementation and performance metrics for the candidate stream ciphers remaining in the phase-III hardware profile. The results are presented in tabular and graphical format together with summarising the utility of the candidates against two notional applications: one for 10Mbps wireless network and a second for 100kHz RFID. An attempt has been made to quantify the flexibility and scalability dimensions of performance.

Keywords. Stream Ciphers, Hardware, ASIC, Performance Evaluation.

1 Introduction

This paper follows on from previous papers [1][2] submitted to the previous SASC conferences and presents hardware implementation results for all the remaining hardware profile candidates. Design performance metrics are presented together with the relevance to two typical application areas.

Security analysis remains the overriding concern compared to hardware/software performance analyses, however, *performance results are key to focussing the security analysis effort on the low resource candidates.* A second aim is to provide an independent set of hardware results for the promising candidates to further the understanding of their relative merits.

Hardware performance is multi-dimensional and the importance of the various quantities such as area, throughput and power depends on the specific application. The eStream hardware testing framework [3] defines five dimensions: compactness, throughput, power consumption, scalability and simplicity. It was also stated that the Advanced Encryption Standard (AES) is to be used as the benchmark for comparison and candidates should be "smaller" and "faster" than the AES.

2 Measuring Hardware Performance

The majority of this section is repeated verbatim from our previous SASC paper [2] for ease of reference with additional definitions for flexibility and simplicity being appended.

For any digital design there is a small set of metrics which can be obtained from the design flow together with some simulations. It is this primary set of metrics which is used to calculate the other derived metrics which designers use as a convenient method for comparing different designs. The particular metrics chosen by any designer as the most appropriate depends very much on application. Typically, a number of metrics are considered to represent the "cost" of implementing a design. Two strong drivers of cost for ASIC design are energy consumption and device area. For high performance applications the throughput to area ratio and energy per processed bit are suitable metrics. Such metrics would be inappropriate in applications where the primitive operates for very short periods at a time. The definitions used in this paper are given below:

Process: The fabrication technology used. The name normally indicates the smallest feature size, library usage and gate construction (eg 0.13μm standard cell CMOS).

Interface: Designs are invariably part of a larger system and thus require connections (on or off chip) with other designs. All the designs in this paper use a synchronous interface with handshaking and on-chip communication is assumed. In this paper, the interfaces differ by their bus widths. Thus the bus width in bits for I/O is included in the results.

Area: Amount of silicon used for the core design (excluding power rings and I/O cells). This result is typically expressed in μm^2 for a specified process. However, the more usable process independent method of expressing the area is to calculate the Gate Equivalence (GE) of the total area by dividing by the lowest power two-input NAND gate's area.

Load/Initialisation Cycles: The definition used here was from RESET going inactive, through loading key and IV, until the validity of the first output bit is signalled. Many would quote just the key/IV mixing cycles however this would fail to account for the impact on interfacing decisions on the latency.

Bits per cycle (running): For the simplest stream ciphers is the number of bits of output keystream per clock cycle. However, many operate in a way that produces batches of output (eg a block cipher in output feedback mode) thus the definition has to include a second clause on sustainable output rate. Thus the better definition is number of bits of output for all subsequent batches/blocks of keystream divided by the number of cycles per batch/block.

Design frequency: This is the clock rate selected by the designer and applied as a constraint to the design tools. The tools will make decisions on driver strengths to meet this requirement. Thus the higher the constraint the more area will be consumed. For low resource design a modest rate must be selected.

Max. Clock frequency: Designs have many connections between inputs outputs and registers, each of these form a timing path (or arc). Simplistically, the slowest timing arc in the design is the critical path and sets upper bound on the clock frequency. The design may be clocked at a significantly lower rate.

Power consumption: Ideally a chip would be manufactured and measurements made for a large set of operations, however, this would be both time-consuming and costly. The alternative is to use specialist tools which operate using estimations of parasitic parameters (resistance and capacitance) from the physical layout of a design together with switching activity from a set of random test vectors. For CMOS there are two components to the power: the static power (roughly proportional to area) and a second dynamic component proportional to the switching activity (probability of a switching event occurring and frequency of operation). Both components also depend on supply voltage. The typical core voltage for the process should be used. At low frequency the static power is significant whilst at the other extreme may be neglected. Power results can be scaled with an acceptable margin of error to other frequencies if the static and dynamic components are treated separately.

The primary metrics may be used to wholly describe a design's performance, however, as can be seen there are many dimensions to performance so engineers often use derived metrics to provide a single dimension for comparison. There is no universal agreement on which metric is the best. The true requirement is to meet all the application driven design constraints. The commonly used derived metrics are given below:

Throughput: The rate at which new output is produced with respect to time, typically expressed in bits-per-second. This definition is further clarified to be the sustainable rate once initialisation is completed at a given operating clock frequency. It is thus simply bits-per-cycle multiplied by the clock-frequency. The maximum throughput will occur at the maximum clock frequency, however, remember that the design tools were given a slack timing constraint to favour area so this metric must be used with care when considering low resource design performance.

Area-Time product: The product of the time taken to produce each new output bit and the area of the design. The reciprocal metric is presented as the **throughput-to-area ratio (TPAR)**. Either representation is frequently used as a measure of design efficiency. However, once again, note that the metrics are at their best at the maximum clock frequency.

Energy-per-bit: This is calculated by dividing the total power consumption by the throughput. Care must be taken to ensure that the power and throughput figures used are for the same clock frequency. At first this measure may appear to be frequency independent, however, if modelled at a low frequency (eg 100kHz) the static power will have a significant impact thus larger area designs will be "less efficient". Conversely, at higher frequencies designs with large amounts of switching activity (including that from switching hazards to do path differences in the large fields of XOR gates present in most crypto-primitives) dominates the power.

Power-Area-Time product: This is the triple product formed from area-time product and the power consumption. As with energy per bit is maximised at the highest operating frequency due to the diminishing effect of the static power.

Power-Time product: Specifically, the product of power and latency (total time taken including initialisation and loading key and IV). This metric is particularly useful for measuring utility of a candidate in application such as RFID where both the power consumption and timeliness of response are important.

As has been frequently stated hardware performance analysis is multidimensional and application specific. Thus to resolve the impasse on which figures to quote the decision is made here to quote the following:

(1) The primary design results for designs prepared with a slack timing constraint of 10MHz clock.
(2) 'Best' metrics: Performance metrics for the designs operating at their maximum frequency given the 10MHz constraint.
(3) High-end wireless: Performance metrics for an output rate of 10Mbps, taken as a typical estimate for future wireless LAN (proposed standards range between 1-100Mbps).
(4) Low-end wireless: Performance metrics for a clock rate of 100kHz, as the low end of RFID/WSN tags which may be powered /clocked directly from the interrogating RF field.

The first three performance dimensions: compactness, throughput and power consumption may be readily compared quantitatively however the remaining two of flexibility and simplicity are much more subjective. There is little quantitative guidance in the testing framework so some definitions are offered here; admittedly the choice of metric is arbitrary but any "scale" is better than none.

Flexibility: It is assumed that a measure of the design space performance trade-offs is required. Herein defined as the (dimensionless) ratio of the throughput-to-area ratio for the maximum performance design variant ($TPAR_{max}$) divided by the corresponding ratio for a low-resource design operating at 100kHz ($TPAR_{100kHz}$).

Simplicity: It is assumed that the desired metric here is a measure of the design time (unfortunately the design work had to be fitted around existing work load thus this could not be reliably accounted for). There are a number of software-engineering metrics which are generally used to describe the complexity / simplicity of a source file. Metrics vary in sophistication and applicability to hardware design; one of the simplest, used here is the number of lines excluding blank lines and comments for all the design source (VHDL) files.

3 Results

The results of the authors previous design work presented at SASC07 [2] have been updated to reflect the "tweaks" made to the candidate ciphers and a number of new designs are presented to complete the phase-III hardware profile.

Candidates such as Grain, Trivium, Mickey and F-FCSR are essentially formed around shift registers together with a combinatorial feedback and output filter functions. All these designs have straight forward implementations. In the case of Grain and Trivium the location of the feedback taps allows feedback and output functions to be replicated allowing more than one bit to be processed per cycle. This is a very convenient feature for a hardware designer as it provides an easily accessible range of throughput, area and power figures to match a given application. For both Grain and Trivium a number of designs have been implemented for different amounts of parallelism. This is indicated after the ciphers name in the results table.

Moustique is a self-synchronising stream cipher which for some communications systems is an advantageous property. It has a small design space in that a number of the "stages" may be performed iteratively to save area. The key is contained in a static register thus could support the use of a one-time-programmable memory for key storage directly.

Pomaranch has 80-bit and 128-bit versions and consist of a 6 or 9 sections each being a 18-bit jump-controlled linear feedback shift register (two types) connected by non-linear function. This function includes the requirement to perform GF(2^9) inversion. The ciphers authors provide the necessary field constructions for a composite field equivalent in GF(2^3^3). However, the jump-control feedback between the sections frustrates attempts to roll the design into a single configurable stage supported by a suitable memory.

Decim is another linear feedback shift register based design with both 80 and 128 bit variants. A state-machine, "ABSG" is fed from a filter function to generate the keystream. The state machine although of relatively simple construction makes generation of parallel outputs troublesome for the hardware designer, the simplest option being to resort to a lookup table approach at relatively high area cost; this greatly limits the design space (practically to at most a four fold parallelism).

Edon80 by design was intended as an 80-element pipeline. However, the relatively neat software definition for the initial mixing and running phases belies relatively high hardware complexity for its implementation. The nature and direction of shifting between loading key, iv and padding, mixing phase and running phase changes resulting in a significant number of additional multiplexers and a need to duplicate the key register. In the implementation in this paper an additional 80 cycles at the end of the initialisation phase were expended to avoid requiring additional pipelining of control lines (saves 160 FF). Edon80(pipelined) is the largest design in the hardware profile so a more iterative and lower area version was also designed (Edon80x4). This comprises only four "e-transformers" rather than the more usual 80.

All the designs have been implemented using *the same design flow*. The natural bus-width for interfacing to each design was selected rather than forcing all designs to use the same bus-width in order to avoid skewing the results. Cadence tools were used together with ModelSim. The process selected was the same 0.13 CMOS and standard cell library as used in [1] and [2]. Best-case worst-case timing analysis was carried out for a desired clock rate of 10MHz. The designs were taken through to physical layout (including clock tree synthesis, placement and routing). The final core area was converted to gate-equivalents. The resulting parasitic values were extracted and the netlist back annotated and simulated with known test vectors to validate the design. To estimate the power consumption, random test vectors were applied to the back annotated netlist and simulated to collect switching activity for a set of 100 different 1 kilobit keystream generations. The power modelling was done using the foundry typical values for the process (1.2Vcore 25°C), the total power and static component are

quoted in the results to permit scaling. The results incorporate both initialisation and operational phases of the design under test.

For the notional future wireless network application, battery life, meeting throughput requirements and area are important to the designer. A good measure for comparing designs is to consider the trade off between the Energy per bit and Throughput/Area metrics.

RFID applications place limits on power, area and latency directly, excesses in any would make a candidate unsuitable for the application. RFID tags must be fundamentally low cost thus low area. A good metric for performance would be power-latency product versus area.

Table 1. Our design results for 0.13μm Standard Cell CMOS

Design	Key bits	Interface bits	Load/Ini cycles	Bits/Cycle (running)	Max. clock freq. MHz	Area NAND GE, gates	Leakage power, μW	Total Power @10MHz, μW
Grain80	80	1	321	1	724.6	1294	2.224	109.4
Grain80x4	80	4	81	4	694.4	1678	3.243	126.6
Grain80x8	80	8	41	8	632.9	2191	4.634	150.7
Grain80x16	80	16	21	16	617.3	3239	7.399	200.5
Trivium	80	1	1314	1	327.9	2580	3.823	175.1
Triviumx2	80	2	660	2	574.7	2627	3.954	182.8
Triviumx4	80	4	332	4	473.9	2705	4.149	184.6
Triviumx8	80	8	168	8	471.7	2952	5.071	203.4
Triviumx16	80	16	86	16	467.3	3166	5.339	214.4
Triviumx32	80	32	45	32	350.9	3787	7.501	282.5
Triviumx64	80	64	24	64	348.4	4921	10.677	374.2
F-FCSR-H	80	8	225	8	392.2	4760	7.973	269.3
F-FCSR-16	128	16	308	16	317.5	8072	13.731	470.1
Grain128	128	1	513	1	925.9	1857	2.698	167.7
Grain128x4	128	4	129	4	584.8	2129	3.806	183.4
Grain128x8	128	8	65	8	581.3	2489	4.898	205.1
Grain128x16	128	16	33	16	540.5	3189	6.882	254.6
Grain128x32	128	32	17	32	452.5	4617	11.442	344.7
Mickey128	128	1	417	1	413.2	5039	8.144	310.7
Mickey2(80)	80	1	261	1	454.5	3188	5.195	196.5
Pomaranch80	80	1	472	1	124.5	5357	10.547	569.3
Pomaranch128	128	1	594	1	104.9	8039	16.185	878.4
Moustique	96	1	202	1	476.2	9607	16.078	464.0
Decim80	80	1	1012	0.25	427.3	2603	3.894	157.7
Decim128	128	1	1617	0.25	309.6	3819	6.052	242.2
Edon80x4	80	8	1869	0.0473	207.9	4969	7.775	280.1
Edon80pl	80	1	392	1	243.3	13010	20.467	478.9
AES [4]*	128	32	50	2.37	131.2*	5398	-	-
AES [5]*	128	8	1016	0.124	80.0*	3400	-	-

* Results are for different CMOS processes (Satoh 0.11, Feldhofer 0.35). Power cannot be scaled reliably between different processes and libraries. The area can be scaled to 0.13um for comparison.

Table 2. Derived metrics for maximum clock frequency

Design	Max Throughput, Mbps	Estimated Power, uW	Energy/bit, pJ/bit	Area-Time product, um^2-us	Tput/Area, kbps/um^2	Power-Area-Time, nJ-um^2
Grain80	724.6	7772	10.72	9.26	107.99	72.0
Grain80x4	2777.7	8569	3.08	3.13	319.33	26.8
Grain80x8	5063.2	9247	1.82	2.24	445.78	20.7
Grain80x16	9876.5	11929	1.20	1.70	588.26	20.3
Trivium	327.9	5618	17.14	40.79	24.51	229.2
Triviumx2	1149.4	10283	8.95	11.85	84.40	121.8
Triviumx4	1895.7	8559	4.51	7.40	135.17	63.3
Triviumx8	3773.6	9360	2.48	4.06	246.62	38.0
Triviumx16	7476.6	9777	1.31	2.20	455.50	21.5
Triviumx32	11228.0	9658	0.86	1.74	571.88	16.9
Triviumx64	22299.6	12677	0.56	1.14	874.13	14.5
F-FCSR-H	3137.2	10255	3.26	7.86	127.13	80.7
F-FCSR-16	5079.3	14503	2.85	8.23	121.38	119.5
Grain128	925.9	15283	16.50	10.39	96.20	158.9
Grain128x4	2339.1	10505	4.49	4.71	211.97	49.6
Grain128x8	4651.1	11646	2.50	2.77	360.52	32.3
Grain128x16	8648.6	13399	1.54	1.91	523.09	25.6
Grain128x32	14479.6	15093	1.04	1.65	604.92	24.9
Mickey128	413.2	12512	30.27	63.21	15.82	790.9
Mickey2(80)	454.5	8701	19.14	36.35	27.50	316.3
Pomaranch80	124.5	6969	55.96	223.01	4.48	1554.3
Pomaranch128	104.9	9063	86.37	397.15	2.51	3599.6
Moustique	476.2	21347	44.83	104.59	9.56	2232.7
Decim80	106.8	6577	61.55	126.28	7.91	830.6
Decim128	77.3	7316	94.52	255.80	3.90	1871.6
Edon80x4	9.8	5670	576.13	2617.43	0.38	14840.8
Edon80pl	243.3	11174	45.92	277.18	3.60	3097.3
AES [4]*	311	-	-	90.12	11.10	-
AES [5]*	10	-	-	1776.33	0.56	-
Better is:	higher	lower	lower	lower	higher	lower

Table 3. Derived metrics for an output rate of 10 Mbps (estimated typical future wireless LAN)

Design	Clock Frequency, MHz	Estimated Power, uW	Energy/bit, pJ/bit	Area-Time, um2-us	Tput/Area, kbps/um2	Power-Area-Time nJ-um2
Grain80	10.00	109.45	10.94	671	1.490	73.4
Grain80x4	2.50	34.07	3.40	870	1.150	29.6
Grain80x8	1.25	22.88	2.28	1136	0.880	26.0
Grain80x16	0.63	19.47	1.94	1679	0.596	32.7
Trivium	10.00	175.06	17.51	1337	0.748	234.1
Triviumx2	5.00	93.38	9.34	1362	0.734	127.2
Triviumx4	2.50	49.27	4.93	1402	0.713	69.1
Triviumx8	1.25	29.86	2.99	1530	0.654	45.7
Triviumx16	0.63	18.41	1.84	1641	0.609	30.2
Triviumx32	0.31	16.09	1.61	1963	0.509	31.6
Triviumx64	0.16	16.35	1.63	2551	0.392	41.7
F-FCSR-H	1.25	40.63	4.06	2468	0.405	100.3
F-FCSR-16	0.63	42.25	4.22	4185	0.239	176.8
Grain128	10.00	167.72	16.77	962	1.039	161.4
Grain128x4	2.50	48.69	4.87	1104	0.906	53.7
Grain128x8	1.25	29.92	2.99	1290	0.775	38.6
Grain128x16	0.63	22.36	2.23	1653	0.605	37.0
Grain128x32	0.31	21.85	2.18	2394	0.418	52.3
Mickey128	10.00	310.72	31.07	2612	0.383	811.6
Mickey2(80)	10.00	196.49	19.65	1652	0.605	324.7
Pomaranch80	10.00	569.34	56.93	2777	0.360	1581.2
Pomaranch128	10.00	878.38	87.83	4167	0.240	3660.6
Moustique	10.00	464.02	46.40	4980	0.201	2311.0
Decim80	40.00	619.10	61.91	1349	0.741	835.3
Decim128	40.00	950.52	95.05	1980	0.505	1882.0
Edon80x4	211.25	5761.22	576.12	2576	0.388	14840.5
Edon80pl	10.00	478.88	47.88	6744	0.148	3229.7
AES [4]*	4.22	-	-	2798	0.357	-
AES [5]*	80.63	-	-	1763	0.567	-
Better is:	lower	lower	lower	lower	higher	lower

Table 4. Derived metrics operating at 100kHz clock (low-end RFID/WSN applications)

Design	Throughput, Mbps	Estimated Power, uW	Energy/Bit, pJ/bit	Area-Time, um2-us	Tput/Area, kbps/um2	Power-Area-Time, nJ-um2	Latency, us	Power-Area-Latency, uJ-um2	Power-Latency, nJ
Grain80	0.100	3.29	32.96	67,098	0.0149	221	3,210	70.99	10.58
Grain80x4	0.400	4.47	11.19	21,747	0.0460	97	810	31.54	3.62
Grain80x8	0.800	6.09	7.61	14,198	0.0704	86	410	28.38	2.49
Grain80x16	1.600	9.33	5.83	10,493	0.0953	97	210	32.89	1.95
Trivium	0.100	5.54	55.36	133,747	0.0075	740	13,140	972.87	72.74
Triviumx2	0.200	5.74	28.71	68,092	0.0147	391	6,600	516.14	37.90
Triviumx4	0.400	5.95	14.89	35,061	0.0285	209	3,320	277.22	19.77
Triviumx8	0.800	7.05	8.82	19,127	0.0523	135	1,680	181.35	11.85
Triviumx16	1.600	7.43	4.64	10,259	0.0975	76	860	104.88	6.39
Triviumx32	3.200	10.25	3.20	6,135	0.1630	62	450	90.56	4.61
Triviumx64	6.400	14.31	2.23	3,986	0.2509	57	240	87.62	3.43
F-FCSR-H	0.800	10.58	13.23	30,847	0.0324	326	2,250	587.78	23.81
F-FCSR-16	1.600	18.29	11.43	26,153	0.0382	478	3,080	2357.93	56.34
Grain128	0.100	4.34	43.48	96,250	0.0104	418	5,130	214.70	22.30
Grain128x4	0.400	5.60	14.00	27,588	0.0362	154	1,290	79.74	7.22
Grain128x8	0.800	6.90	8.62	16,127	0.0620	111	650	57.86	4.48
Grain128x16	1.600	9.36	5.85	10,333	0.0968	96	330	51.06	3.08
Grain128x32	3.200	14.77	4.61	7,480	0.1337	110	170	60.12	2.51
Mickey128	0.100	11.17	111.69	261,204	0.0038	2,917	4,170	1216.64	46.57
Mickey2(80)	0.100	7.10	71.08	165,249	0.0061	1,174	2,610	306.58	18.55
Pomaranch80	0.100	16.13	161.35	277,724	0.0036	4,481	4,720	2115.12	76.15
Pomaranch128	0.100	24.80	248.07	416,742	0.0024	10,338	5,940	6140.88	147.35
Moustique	0.100	20.56	205.58	498,044	0.0020	10,239	2020	2068.22	41.53
Decim80	0.025	5.43	217.28	539,689	0.0019	2,931	10,120	741.69	54.97
Decim128	0.025	8.41	336.54	791,977	0.0013	6,663	16,170	2693.63	136.04
Edon80x4	0.005	10.49	2217.91	5,441,651	0.0002	57,132	18,690	5054.66	196.22
Edon80pl	0.100	25.05	250.51	674,421	0.0015	16895	3,920	6622.82	98.20
AES [4]*	0.237	-	-	118,054	0.0085	-	500	-	-
AES [5]*	0.001	-	-	1,421,064	0.0007	-	10,160	-	-
Better is:	higher	lower *	lower	lower	higher	lower	lower *	lower	lower ***

Table 5. Flexibility and simplicity

Design	Flexibility ($TPAR_{max}$ ÷ $TPAR_{100k}$)	Source VHDL (bytes)	Simplicity § comment lines	empty lines	VHDL code lines
Grain80 (x1 to x16)	39,472	5,415	31	10	158
Grain128 (x1 to x32)	58,224	4,703	21	29	138
Trivium (x1 to x64)	116,913	5.916	45	26	159
F-FCSR-H	3,922	4,923	22	33	152
F-FCSR-16	3,175	5,668	20	38	177
Mickey128	4,132	6,399	41	34	127
Mickey2(80)	4,545	5,645	20	37	149
Pomaranch80	1,245	23,378	71	156	578
Pomaranch128	1,049	23,378	71	156	578
Moustique	4,762	16,960	44	77	496
Decim80 #	4,274	16,210	79	103	421
Decim128 #	3,096	16,560	95	117	396
Edon80 (x4 to x80pl)	19,632	20,704	95	149	618

Decim with x4 versions are possible but not implemented by these authors the estimated, however, "best-case" flexibility result will be less than 4 times the stated value.

§ Figures quoted for designer's first validated draft.

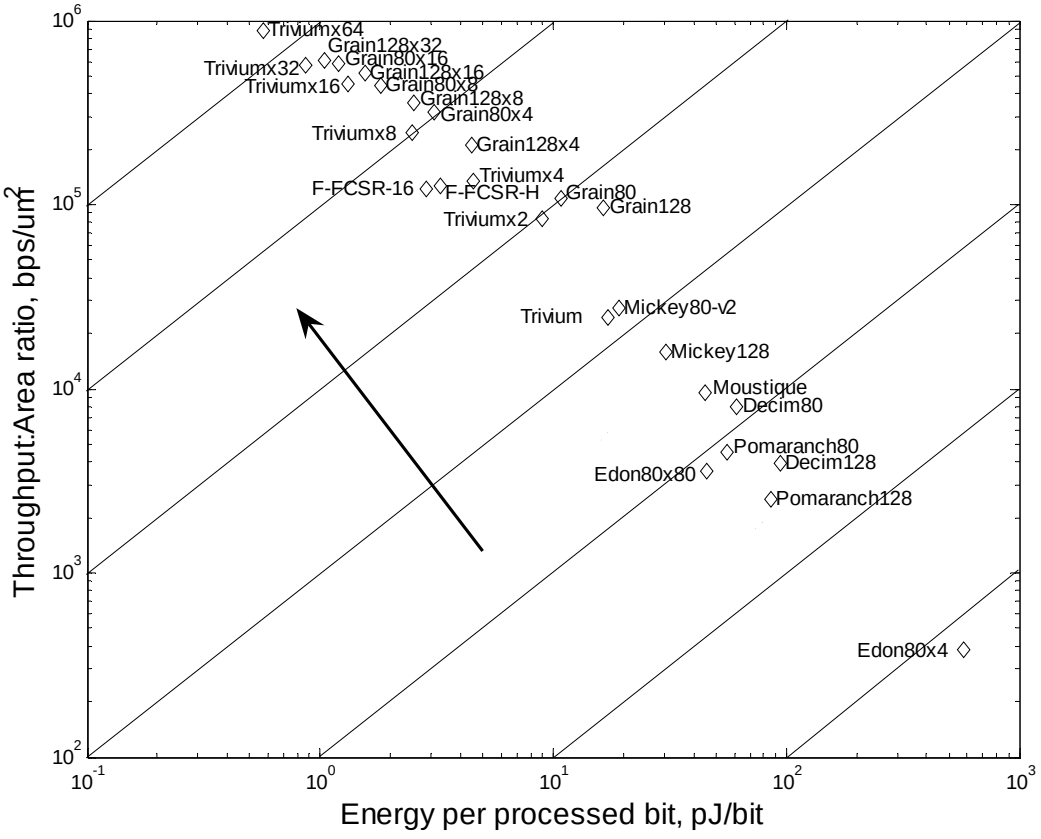

Fig. 1. 0.13um Standard Cell CMOS design performance metrics at maximum throughput, arrow shows improving performance

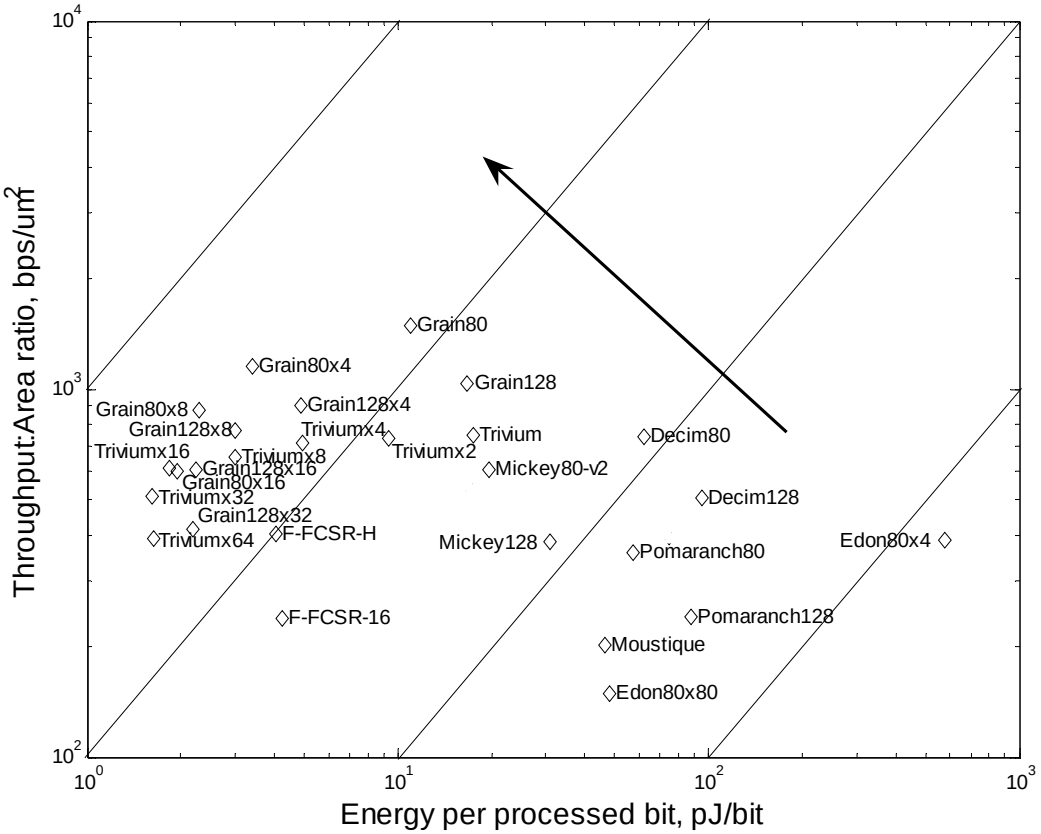

Fig. 2. Performance metrics for notional Wireless-LAN at 10Mbps throughput, arrow shows improving performance.

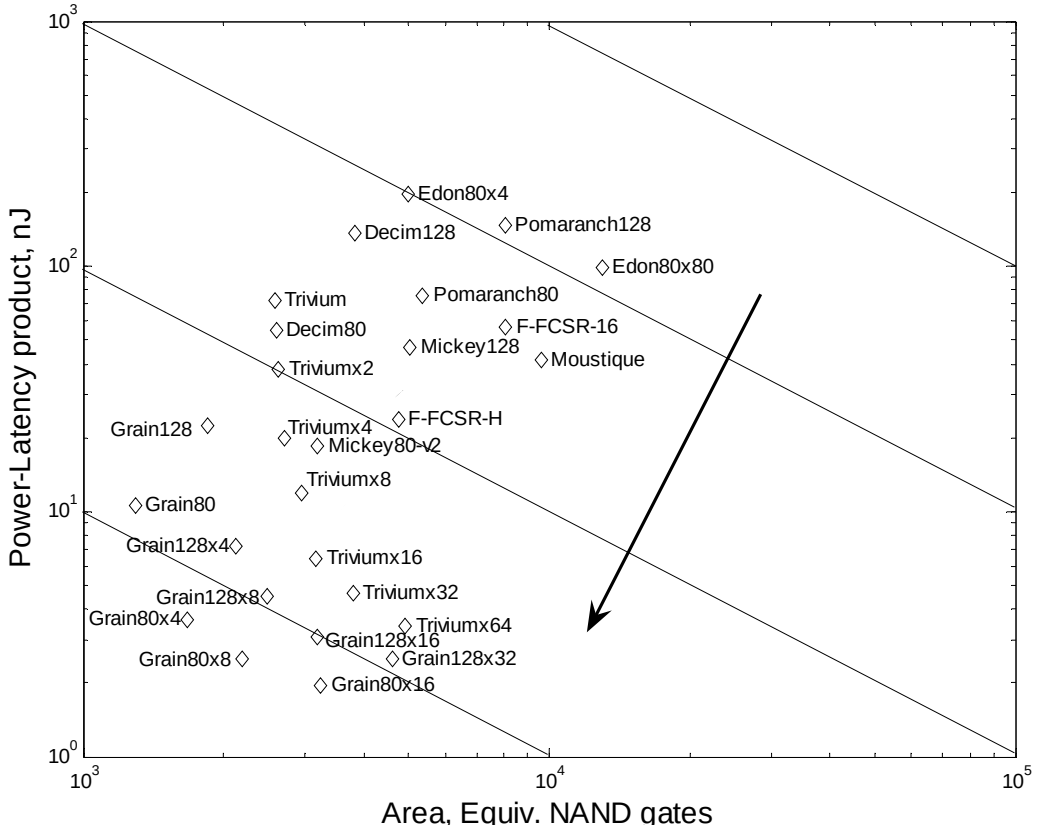

Fig. 3. Performance for low-end RFID/WSN application at 100kHz clock, arrow shows improving performance

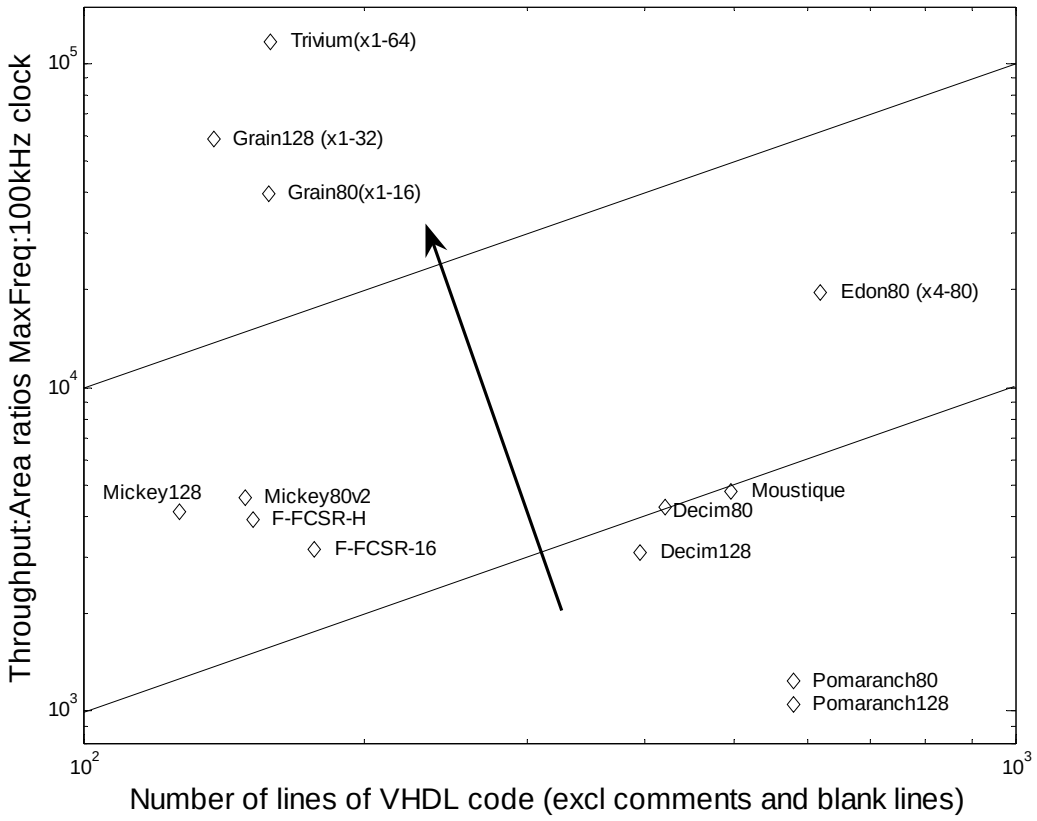

Fig. 4. Flexibility as Throughput : Area for MaxFreq:100kHz clock versus Simplicity as lines of VHDL

4 Require even lower power?

For the primary set of results presented in this paper a typical general purpose standard cell library was used on a 0.13um process with standard process options for all the designs. The previous section provides a set of readily comparable results between all of the phase 3 hardware candidates. This section is only included to demonstrate the advantage to any hardware design by moving to a specialist low power library, selecting low-leakage process options and moving to a more advanced design flow significant power savings can be achieved at the expense of considerable additional design effort. At relatively low clock rates relative to the critical path the core voltage may be reduced accepting longer propagation delays thus further reducing the power consumption. As an example, the table below shows the results for Grain and Trivium.

Design	Grain80x8	Grain128x16	Triviumx8
Interface bits	8	16	8
Core voltage, V	0.8	0.8	0.8
Area, NAND GE	2796	4057	3244
Clock for 10Mbps, MHz	1.25	0.625	1.25
Power (10Mbps), μW	10.710	8.761	15.108
Energy/Bit (10 Mbps), pJ/bit	1.071	0.876	1.511
Power-Area-Time, nJ-um2	11.5	13.6	18.8
Power (100 kHz clk), μW	0.857	1.403	1.209
Power-Latency (100kHz clk), pJ	352	463	2056

At 100kHz Grain80x8 shows approximately a factor of 7 improvement in power-latency product (for the same VHDL source) by changing the library, process options and flow. These results have been included as a reminder that comparison in absolute units between different designs must be made using the same technology, libraries and process options and to demonstrate the low resource nature of stream ciphers using an advanced flow and process options for those who wish to make absolute comparisons with other designs.

5 Conclusions

This treatment has considered the entire set of phase-III candidates in the hardware profile. Using the two sample application of a notional future wireless network (WLAN) and low-end of radio frequency identification tags / wireless sensor network nodes (RFID/WSN). The table below provides the *first documented attempt summarising quantifiable results* for all the performance dimensions specified in [3] for each of the candidate ciphers. The authors overall view relative to the AES is summarised by the left hand column. It is left to others to form their own opinions on the applicability of each cipher to their specific design constraints.

	Power-Area-Time Max. clock	Power-Area-Time WLAN	Power-Area-Time RFID/WSN	Flexibility (design space)	Simplicity (code lines)
🏆	Trivium (x64)	Grain80 (x8)	Grain80 (x8)	Trivium	Mickey128
☺	Grain80 (x16) Grain128 (x32) F-FCSR-H F-FCSR-16	Trivium (x8–x32) F-FCSR-H	Grain128 (x16) Trivium (x8–x32)	Grain128 Grain80	Grain128 Mickey80v2 Grain80 Trivium F-FCSR-H F-FCSR-16
😐	Mickey80v2 Mickey128 Moustique *	F-FCSR-16 Mickey80v2	F-FCSR-H Mickey80v2 Decim80	Edon80 Decim80 Decim128 Moustique *	Decim128 Decim80 Moustique *
☹	Decim80 Edon80 Pomaranch80 Decim128 Pomaranch128	Mickey128 Decim80 Pomaranch80 Decim128 Pomaranch128 Moustique * Edon80	Mickey128 Pomaranch80 F-FCSR-16 Moustique * Decim128 Edon80 Pomaranch128	F-FCSR-H F-FCSR-16 Mickey80v2 Mickey128 Pomaranch80 Pomaranch128	Pomaranch80 Pomaranch128 Edon80

* Moustique is the only self synchronising stream cipher so should be considered of significant merit irrespective of other performance metrics.

Acknowledgements

The authors wish to thank the developers of the candidate ciphers for all their commitment and effort in continuing to refine their submission and further for their assistance in understanding and resolving minor discrepancies between the descriptions and reference designs.

References

[1] T. Good, W. Chelton and M. Benaissa, "Review of stream cipher candidates from a low resource hardware perspective", SASC06, available at: www.ecrypt.eu.org/stream

[2] T. Good and M. Benaissa, "Hardware results for selected stream cipher candidates", SASC07, available at: www.ecrypt.eu.org/stream

[3] L. Batina, S. Kumar, J. Lano, K. Lemke, N. Mentens, C. Paar, B. Preneel, K. Sakiyama and I. Verbauwhede, "Testing Framework for eSTREAM Profile II Candidates", SASC06, available at: www.ecrypt.eu.org/stream

[4] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization.", AsiaCrypt 2001, LNCS vol. 2249, pp 230-254, Springer 2001.

[5] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES Implementation on a Grain of Sand", IEE Proceedings on Information Security, 152:13-20, October 2005.