# Hardware results for selected 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 in the Phase II Hardware Focus. Quantitative consideration is also given to all candidate ciphers as to whether any should be added to the Hardware Focus set. In this treatment, only the submissions without licensing restrictions have been considered. The results are presented in tabular and graphical format together with some recommendations aimed at simplifying the implementation task for future engineers and a priority order for cryptanalysis, solely from a hardware perspective, is presented.

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

## 1 Introduction

In 2004, a project under the Information Societies Technology (IST) Programme of the European Commission "eCrypt" network of excellence called "eStream" was started tasked with seeking a strong stream cipher. Thirty-four candidate ciphers were submitted. From initial evaluations at SASC 2006 [1], which concluded phase-I, the e-Stream project formally archived a number of the candidates which had a low security outlook and created two focus sets, one for hardware and the second for software to further the analysis of the candidates.

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

Software performance analysis is effectively one-dimensional in that "speed" is the only metric considered. An analysis of the 256-bit candidates is given in [2]. However, hardware performance is multi-dimensional and the importance of the various quantities such as area, throughput and power depends on the specific application. A 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. At SASC 2006 hardware analyses of selected candidates were presented [4,5], subsequently many of the candidate ciphers have been amended to improve their security thus further review is necessary.

This paper presents a comprehensive analysis of all the remaining candidates from a hardware perspective, including implementation results for the most promising ones. Candidates submitted only to the software profile were also considered from a hardware perspective, with particular attention given to those already in the software focus. Design performance metrics are presented together with the relevance to two typical application areas.

## 2 Selection

The starting point is to review the current status of all the candidates. The first "cut" is to eliminate candidates which have already been archived at the end of phase-I and those where the posted cryptanalysis indicates a low security outlook. All reviews are subject to some kind of bias: <u>only candidates which are "free-for-all" have been considered and then only from a low resource hardware perspective</u>. The authors have no affiliations to any of the candidates and this review is carried out independent of eStream. As security is the overriding concern, the key question is: "should there be any changes to the hardware focus set?" in order to re-focus the cryptanalysis effort.

Table 1 shows all the submitted candidates together with their submitted eStream profile, free-for-all status, security outlook and if they are part of the software or hardware focus group. Profile 1 is software, profile 2 hardware and suffixed with 'A' where the primitive directly supports a Message Authentication Code (MAC).

The security outlook indicates where cryptanalysis (by others) has been carried out on the current version of a primitive the order of complexity is given and those without analysis marked “no ca”.

At the end of phase I, eight candidates (Frogbit, Mag, Mir-1, Mosquito, Sfinks, SSS, Trbdk3yaea and Yamb) were archived due to security concerns and not considered further in this analysis.

**Table 1.** Summary of selection of phase-II candidates: first cut

| Cipher | Profile | Free4all | Security Outlook | Focus | Key & IV bits | Cut | Notes |
|---|---|---|---|---|---|---|---|
| ABC v3 | 1 | yes | $2^{50.56}$ | no | 128<br>128 | ✗ | ca shows weak keys attack O($2^{50.56}$) [6] |
| Achterbahn | 2 | yes | $2^{56.32}$ | no | 80/128<br>8N | ✗ | ca shows O($2^{56.32}$) [7] |
| CryptMT/ Fubuki | 1 | no | no ca | no | - | ✗ | not free for all |
| Decim | 2 | no | no ca | no | - | ✗ | not free for all |
| Dicing | 1 | yes | no ca | no | 128/256<br>128/256 | ✓ | |
| Dragon | 1 | yes | no ca | SW | 128/256<br>128/256 | ✓ | |
| Edon80 | 2 | no | no ca | no | - | ✗ | not free for all |
| F-FCSR | 1&2 | yes | no ca | no | 80<br>32-80 | ✓ | |
| Grain | 2 | yes | no ca | HW | 80<br>63 | ✓ | ok |
| HC-256 | 1 | yes | no ca | SW | 256<br>256 | ✓ | Unchanged |
| Hermes8 | 1&2 | yes | low (ph.2) | no | 80<br>184 | ✗ | ca less one sec on PC [8] |
| LEX | 1&2 | yes | no ca | SW | 128<br>128 | ✓ | ok |
| MICKEY | 2 | yes | no ca | no | - | ✗ | Existing independent hardware results shows lacks scalability [9] |
| MICKEY-128 | 2 | yes | no ca | HW | 128<br>128 | ✓ | ok |
| NLS | 1A&<br>2A | yes | $2^{74}$ | no | 256<br>256 | ✗ | Distinguishing attack $2^{74}$ [10] |
| Phelix | 1A&<br>2A | yes | ph2 | HW/SW | 256<br>128 | ✓ | ok, Unchanged |
| Polar Bear | 1&2 | yes | no ca | no | 128<br><248 | ✗ | 5 round AES + RC4, one round of AES is still relatively large |
| Pomaranch v3 | 1&2 | yes | no ca | no | 128<br>144 | ✓ | |
| Py (“Roo”) | 1 | yes | $2^{72}$ | SW | 256<br>128 | ✗ | Key recovery to give effective key of $2^{72}$ [11] |
| Rabbit | 1&2 | no | ph2 | no | - | ✗ | not free for all |
| Salsa20 | 1 | yes | ph2 | SW | 256<br>64 | ✓ | ok, Unchanged |
| Sosemanuk | 1 | yes | no ca | SW | 128/256<br>128 | ✓ | ok |
| Trivium | 2 | yes | ph2 | HW | 80<br>80 | ✓ | ok, ca to date shows strength |
| TSC-4 | 2 | yes | no ca | no | - | ✗ | Devastating attack on previous version waiting for ca. |
| VEST | 2A | no | ph2 | no | - | ✗ | not free for all |
| WG (P2) | 2 | yes | no ca | no | 80/128<br>32/64 | ✓ | |
| ZK-Crypt | 2 | no | ph2 | no | - | ✗ | not free for all |

From the initial list of candidates, only 14 remain as part of this treatment. A second cut is made in terms area using the stated eStream requirement “candidates should be smaller than the AES”. The actual area is only known once the design is completed however an early estimate can be made. Consider the reasoning: if a candidate is to be small in size then the area will be dominated by the flip-flops needed to store the internal state. The D-type flip-flop with clock enable is the most likely gate per bit of storage, thus storage can be costed as 8 NAND Gate Equivalents (GE) per bit.

Some designs incorporate key dependant lookup tables (eg S-boxes) which, from a hardware perspective, must be considered as part of the internal state.

**Table 2.** Summary of selection of remaining phase-II candidates: second cut

| Cipher | Profile | Focus | Key &<br>IV bits | Internal state | Approx. GE<br>(mem only) | Cut | Notes |
|---|---|---|---|---|---|---|---|
| Dicing | 1 | no | 128/256<br>128/256 | 768<br>+2k | 22.5k | ✗ | Previous version showed weakness, not in either focus, relatively large GE for memory |
| Dragon | 1 | SW | 128/256<br>128/256 | 192<br>+16k | 131k | ✗ | 16k randomly generated Sbox. Area>>AES |
| F-FCSR-H | 1&2 | no | 80<br>80 | 160<br>+243 | 3.2k | ✓ | Initialisation requirement effectively doubles internal state + storage for “carry bits”. |
| Grain | 2 | HW | 80<br>63 | 160 | 1.3k | ✓ | |
| Grain-128 | 2 | HW | 128<br>128 | 256 | 2k | ✓ | |
| HC-256 | 1 | SW | 256<br>256 | 64k | 524k | ✗ | 64k of randomly generated Sboxes. Area>>AES |
| LEX | 1&2 | SW | 128<br>128 | 256 | 2k | ~ | Basically an AES with x2.5 throughput. Problems with reference C-code thus cannot test hardware design. |
| MICKEY-128 | 2 | HW | 128<br>128 | 256 | 2k | ✓ | |
| Phelix | 1A&<br>2A | HW/<br>SW | 256<br>128 | 352 | 2.8k | ✓ | |
| Pomaranch v3 | 1&2 | no | 80/128<br>32/162 | 192 | 1.5k | ~ | Previous version showed weakness, not in either focus, uses GF(2^9) inversion. Relatively large compared with Grain/Trivium. |
| Salsa20 | 1 | SW | 256<br>64 | 1024 | 8k | ✓ | Final addition double internal state |
| Sosemanuk | 1 | SW | 128/256<br>128 | 768 | 6.1k | ✓ | ~0.5k of sboxes<br>Includes “state” needed for init. & output fn |
| Trivium | 2 | HW | 80<br>80 | 288 | 2.3k | ✓ | |
| WG (P2) | 2 | no | 80/128<br>32/64 | 319 | 2.5k | ~ | Previous version showed weakness, not in either focus, output function needs GF(2^29) multiplications and inversion. Relatively large compared with Grain/Trivium. |

The result of the second cut (table 2) is not a simple binary decision. The first category is composed of those already recognised as most suitable for hardware (in the hardware focus set). These are Grain, Grain-128, Mickey-128, Phelix and Trivium. For each of these the full hardware design will be updated/carried out. The second category is those whose memory requirements alone make them “expensive” for hardware implementation. These are Dicing, Dragon and HC-256 which will not be considered any further by this treatment.

Two designs, Salsa20 and Sosemanuk are in the software focus profile and initial examination indicates that they merit further consideration in terms of low resource implementation. F-FCSR-H was not in either focus group however from a hardware perspective is worthy of a little attention. Thus hardware designs for each will be carried out.

Pomaranch and WG were previously identified as having security issues during phase-I, thus were not part of these authors previous analysis [5]. Modifications have been made to address the weaknesses, however, the designs share similar features as with Sfinks which make scalable implementation difficult and results in a relatively low bit rate when compared with candidates such as Trivium and Grain. These will not be considered further by this analysis, however, others may wish to.

Finally, it should be noted that Lex, based on the AES in OFB mode, thus is approximately the same area, however, for the same throughput can be clocked 2.5 times slower thus potentially consume proportionately less power. However, there is no apparent advantage for low end applications such as RFID, where the clock rate is fixed.

The result of the selection is that the carefully selected set of eight candidates will be taken forward in this paper for full hardware design, testing and performance analysis. It should be noted that two of the candidates were submitted only to the software profile, however, may have suitable hardware implementation. If so would merit changing to be in both the hardware and software profiles. The eight selected candidates are:

**Grain, Grain-128, Mickey-128, F-FCSR-H, Phelix, Trivium, Salsa20 and Sosemanuk**

## 3 Measuring Hardware Performance

The eStream hardware testing framework [3] sets out five aspects of performance for consideration: area compactness, performance in terms of throughput, power/energy consumption, flexibility and simplicity. The call is for moderate throughput (better than the AES) with lower resource utilisation (smaller than the AES).

The first three aspects (area, throughput and power) are relatively straight forward to quantify, however, the remaining dimensions (Flexibility/Scalability/Pipelining and Simplicity/Completeness/Clarity) are much more subjective and no comment is intended by this paper.

The baseline for comparison was defined in the testing framework for ASIC as two designs [12,13].

There are a number of issues, which were raised at SASC06 still to be addressed:

(1) Comparison of designs with and without Message Authentication Code (MAC) support

(2) Performance, in terms of throughput only, does not address latency issues which may be critical in some applications.

(3) Impact of application driven constraints on design results (designers want to do the minimum to meet their needs in terms of area, latency, throughput, power, interfacing and design time).

### 3.1 Performance

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 definitions used in this paper are given below:

**Process:** The fabrication technology used, name normally is a mixture of smallest feature size, gate construction and library usage (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 $\mu m^2$ 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 arc in the design is the critical path and sets upper bound on lock 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 designs 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**. 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.

# 4 Results

The results of the authors previous design work presented at SASC06 [5] have been updated to reflect the “tweaks” made to the candidate ciphers and a number of new designs are presented. All the results quoted are for the phase-II versions of the ciphers.

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.

Phelix remains unchanged since submission. Two designs were implemented, one using the complete round function and the second using a half round function. For the full-round design, approximately 50% of the area is consumed by the hardware required for the key expansion (including key and nonce storage). Simplification of the initialisation including a definition which permitted the nonce being loaded into a counter’s register would be beneficial from a hardware perspective.

The hardware design space for Salsa20 has been explored. Salsa20 performs a relatively simple hash operator on a 4x4 array of words alternating between operating on rows and columns. The simplicity of the hash combined with the alternating row-column processing frustrates pipelining. The designs are characterised by the number of hash functions included in the datapath. This has been explored from a single hash (1h) contained within a 32-bit processor style architecture up to the full Salsa20 double-round function, based around shift registers, containing 32 hashes. The design containing four hash operators was found to be the best. The final addition of the previous state is the one feature of Salsa20 (was intended as a software cipher) effectively doubles the internal state, thus considerably adds to the area for a hardware implementation.

Sosemanuk, was intended as a software cipher, however can be implemented in hardware with good performance for higher end applications such as wireless networks. The design has a number of distinct phases of processing, initial key mixing, key-IV mixing to create an internal state and finally iteration of the internal state using feedback and output functions to yield the keystream. The inability to directly operate on the internal state increases both the size and latency of the cipher however once operational the comparative short critical path and high bits per cycle rate yield an impressive performance. To aid others in improving the hardware design, a block diagram is included in Appendix A.

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 [5]. 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 3.** 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 | 2.22 | 109.45 |
| Grain80, x4 | 80 | 4 | 81 | 4 | 694.4 | 1678 | 3.24 | 126.59 |
| Grain80, x8 | 80 | 8 | 41 | 8 | 632.9 | 2191 | 4.63 | 150.66 |
| Grain80, x16 | 80 | 16 | 21 | 16 | 617.3 | 3239 | 7.40 | 200.53 |
| Trivium | 80 | 1 | 1333 | 1 | 358.4 | 2599 | 3.84 | 181.18 |
| Trivium, x4 | 80 | 4 | 336 | 4 | 413.2 | 2660 | 4.04 | 184.83 |
| Trivium, x8 | 80 | 8 | 170 | 8 | 359.7 | 2801 | 4.45 | 199.59 |
| Trivium, x16 | 80 | 16 | 87 | 16 | 408.2 | 3185 | 5.84 | 231.23 |
| Trivium, x32 | 80 | 32 | 45 | 32 | 350.9 | 3787 | 7.50 | 282.55 |
| Trivium, x64 | 80 | 64 | 24 | 64 | 348.4 | 4921 | 10.68 | 374.19 |
| F-FCSR-H | 80 | 8 | 225 | 8 | 392.2 | 4760 | 7.97 | 269.27 |
| Grain128 | 128 | 1 | 513 | 1 | 925.9 | 1857 | 2.70 | 167.73 |
| Grain128, x4 | 128 | 4 | 129 | 4 | 584.8 | 2129 | 3.81 | 183.37 |
| Grain128, x8 | 128 | 8 | 65 | 8 | 581.4 | 2489 | 4.90 | 205.12 |
| Grain128, x16 | 128 | 16 | 33 | 16 | 540.5 | 3189 | 6.88 | 254.64 |
| Grain128, x32 | 128 | 32 | 17 | 32 | 452.5 | 4617 | 11.44 | 344.74 |
| Mickey128 | 128 | 1 | 417 | 1 | 413.2 | 5039 | 8.14 | 310.73 |
| Phelix, ½ rnd | 256 | 32 | 51 | 16 | 88.1 | 13159 | 23.90 | 928.85 |
| Phelix, 1 rnd | 256 | 32 | 34 | 32 | 63.0 | 15032 | 27.60 | 1432.37 |
| Sosemaunk | 256 | 32 | 255 | 32 | 188.3 | 18819 | 33.55 | 812.47 |
| Salsa20, 1h | 256 | 32 | 533 | 0.994 | 121.9 | 12126 | 19.36 | 708.46 |
| Salsa20, 4h | 256 | 32 | 100 | 5.28 | 155.0 | 12914 | 22.34 | 883.94 |
| Salsa20, 16h | 256 | 32 | 40 | 13.84 | 54.4 | 16394 | 31.43 | 2368.98 |
| Salsa20, 32h | 256 | 32 | 30 | 18.96 | 35.2 | 18626 | 35.06 | 3375.28 |
| AES [12]* | 128 | 32 | 50 | 2.37 | 131.2* | 5398 | - | - |
| AES [13]* | 128 | 8 | 1016 | 0.124 | 80.0* | 3400 | - | - |
| Better is: | | | lower | lower | higher | lower | lower | lower |

* 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 4.** 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 | 725 | 7,772 | 10.73 | 9.26 | 108.00 | 72.0 |
| Grain80, x4 | 2,778 | 8,569 | 3.08 | 3.13 | 319.33 | 26.8 |
| Grain80, x8 | 5,063 | 9,247 | 1.83 | 2.24 | 445.78 | 20.7 |
| Grain80, x16 | 9,877 | 11,929 | 1.21 | 1.70 | 588.27 | 20.3 |
| Trivium | 358 | 6,360 | 17.74 | 37.58 | 26.61 | 239.1 |
| Trivium, x4 | 1,653 | 7,475 | 4.52 | 8.34 | 119.88 | 62.3 |
| Trivium, x8 | 2,878 | 7,024 | 2.44 | 5.05 | 198.16 | 35.4 |
| Trivium, x16 | 6,531 | 9,205 | 1.41 | 2.53 | 395.57 | 23.3 |
| Trivium, x32 | 11,228 | 9,658 | 0.86 | 1.75 | 571.88 | 16.9 |
| Trivium, x64 | 22,300 | 12,677 | 0.57 | 1.14 | 874.14 | 14.5 |
| F-FCSR-H | 3,137 | 10,255 | 3.27 | 7.87 | 127.13 | 80.7 |
| Grain128 | 926 | 15,283 | 16.51 | 10.39 | 96.20 | 158.9 |
| Grain128, x4 | 2,339 | 10,505 | 4.49 | 4.72 | 211.98 | 49.6 |
| Grain128, x8 | 4,651 | 11,646 | 2.50 | 2.77 | 360.52 | 32.3 |
| Grain128, x16 | 8,648 | 13,399 | 1.55 | 1.91 | 523.10 | 25.6 |
| Grain128, x32 | 14,480 | 15,093 | 1.04 | 1.65 | 604.92 | 24.9 |
| Mickey128 | 413 | 12,512 | 30.28 | 63.21 | 15.82 | 790.9 |
| Phelix, ½ rnd | 1,410 | 7,997 | 5.67 | 48.39 | 20.66 | 387.0 |
| Phelix, 1 rnd | 2,016 | 8,879 | 4.40 | 38.65 | 25.88 | 343.2 |
| Sosemaunk | 6,026 | 14,702 | 2.44 | 16.19 | 61.77 | 238.0 |
| Salsa20, 1h | 121 | 8,423 | 69.47 | 518.50 | 1.93 | 4367.3 |
| Salsa20, 4h | 818 | 13,380 | 16.35 | 81.80 | 12.22 | 1094.6 |
| Salsa20, 16h | 753 | 12,756 | 16.93 | 112.82 | 8.86 | 1439.2 |
| Salsa20, 32h | 668 | 11,801 | 17.67 | 144.56 | 6.92 | 1705.9 |
| AES [12]* | 311 | - | - | 90.12 | 11.10 | - |
| AES [13]* | 10 | - | - | 1776.33 | 0.56 | - |
| Better is: | higher | lower | lower | lower | higher | lower |

**Table 5.** 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 | 10.000 | 109.4 | 10.95 | 671 | 1.490 | 73.4 |
| Grain80, x4 | 2.500 | 34.1 | 3.41 | 870 | 1.150 | 29.6 |
| Grain80, x8 | 1.250 | 22.9 | 2.29 | 1136 | 0.880 | 26.0 |
| Grain80, x16 | 0.625 | 19.5 | 1.95 | 1679 | 0.596 | 32.7 |
| Trivium | 10.000 | 181.2 | 18.12 | 1347 | 0.742 | 244.1 |
| Trivium, x4 | 2.500 | 49.2 | 4.92 | 1379 | 0.725 | 67.9 |
| Trivium, x8 | 1.250 | 28.8 | 2.88 | 1452 | 0.689 | 41.9 |
| Trivium, x16 | 0.625 | 19.9 | 1.99 | 1651 | 0.606 | 32.9 |
| Trivium, x32 | 0.313 | 16.1 | 1.61 | 1963 | 0.509 | 31.6 |
| Trivium, x64 | 0.156 | 16.4 | 1.64 | 2551 | 0.392 | 41.7 |
| F-FCSR-H | 1.250 | 40.6 | 4.06 | 2468 | 0.405 | 100.3 |
| Grain128 | 10.000 | 167.7 | 16.77 | 962 | 1.039 | 161.4 |
| Grain128, x4 | 2.500 | 48.7 | 4.87 | 1104 | 0.906 | 53.7 |
| Grain128, x8 | 1.250 | 29.9 | 2.99 | 1290 | 0.775 | 38.6 |
| Grain128, x16 | 0.625 | 22.4 | 2.24 | 1653 | 0.605 | 37.0 |
| Grain128, x32 | 0.313 | 21.9 | 2.19 | 2394 | 0.418 | 52.3 |
| Mickey128 | 10.000 | 310.7 | 31.07 | 2612 | 0.383 | 811.6 |
| Phelix, ½ rnd | 0.625 | 80.5 | 8.05 | 6822 | 0.147 | 548.9 |
| Phelix, 1 rnd | 0.313 | 71.5 | 7.15 | 7793 | 0.128 | 557.1 |
| Sosemaunk | 0.313 | 57.9 | 5.79 | 9756 | 0.103 | 564.8 |
| Salsa20, 1h | 10.059 | 712.5 | 71.25 | 6286 | 0.159 | 4479.0 |
| Salsa20, 4h | 1.895 | 185.6 | 18.56 | 6694 | 0.149 | 1242.3 |
| Salsa20, 16h | 0.723 | 200.356 | 20.04 | 8499 | 0.118 | 1702.8 |
| Salsa20, 32h | 0.527 | 211.208 | 21.12 | 9656 | 0.104 | 2039.4 |
| AES [12]* | 4.219 | - | - | 2798 | 0.357 | - |
| AES [13]* | 80.625 | - | - | 1763 | 0.567 | - |
| Better is: | lower | lower | lower | lower | higher | lower |

**Table 6.** 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 | 3.3 | 33.0 | 67,098 | 0.0149 | 221.2 | 3,210 | 71.0 | 10.58 |
| Grain80, x4 | 0.400 | 4.5 | 11.2 | 21,747 | 0.0460 | 97.3 | 810 | 31.5 | 3.63 |
| Grain80, x8 | 0.800 | 6.1 | 7.6 | 14,198 | 0.0704 | 86.5 | 410 | 28.4 | 2.59 |
| Grain80, x16 | 1.600 | 9.3 | 5.8 | 10,493 | 0.0953 | 97.9 | 210 | 32.9 | 1.96 |
| Trivium | 0.100 | 5.6 | 56.1 | 134,715 | 0.0074 | 755.7 | 13,330 | 1007.3 | 74.77 |
| Trivium, x4 | 0.400 | 5.9 | 14.6 | 34,469 | 0.0290 | 201.7 | 3,360 | 271.1 | 19.67 |
| Trivium, x8 | 0.800 | 6.4 | 8.0 | 18,153 | 0.0551 | 116.2 | 1,700 | 158.0 | 10.88 |
| Trivium, x16 | 1.600 | 8.1 | 5.1 | 10,318 | 0.0969 | 83.6 | 870 | 116.3 | 7.05 |
| Trivium, x32 | 3.200 | 10.3 | 3.2 | 6,135 | 0.1630 | 62.9 | 450 | 90.6 | 4.61 |
| Trivium, x64 | 6.400 | 14.3 | 2.2 | 3,986 | 0.2509 | 57.0 | 240 | 87.6 | 3.43 |
| F-FCSR-H | 0.800 | 10.6 | 13.2 | 30,847 | 0.0324 | 326.5 | 2,250 | 587.8 | 23.82 |
| Grain128 | 0.100 | 4.3 | 43.5 | 96,250 | 0.0104 | 418.5 | 5,130 | 214.7 | 22.31 |
| Grain128, x4 | 0.400 | 5.6 | 14.0 | 27,588 | 0.0362 | 154.5 | 1,290 | 79.7 | 7.23 |
| Grain128, x8 | 0.800 | 6.9 | 8.6 | 16,127 | 0.0620 | 111.3 | 650 | 57.9 | 4.48 |
| Grain128, x16 | 1.600 | 9.3 | 5.8 | 10,333 | 0.0968 | 96.7 | 330 | 51.1 | 3.09 |
| Grain128, x32 | 3.200 | 14.8 | 4.6 | 7,480 | 0.1337 | 110.5 | 170 | 60.1 | 2.51 |
| Mickey128 | 0.100 | 11.2 | 111.7 | 261,204 | 0.0038 | 2,917.6 | 4,170 | 1216.6 | 46.58 |
| Phelix, ½ rnd | 1.600 | 32.9 | 20.6 | 42,635 | 0.0235 | 1,404.8 | 510 | 1146.3 | 16.80 |
| Phelix, 1 rnd | 3.200 | 41.6 | 13.0 | 24,352 | 0.0411 | 1,014.1 | 340 | 1103.4 | 14.16 |
| Sosemaunk | 3.200 | 41.3 | 12.9 | 30,487 | 0.0328 | 1,260.3 | 2,550 | 10284.0 | 105.41 |
| Salsa20, 1h | 0.099 | 26.3 | 264.0 | 632,312 | 0.0016 | 16,598.8 | 5,330 | 8795.6 | 139.92 |
| Salsa20, 4h | 0.528 | 31.0 | 58.7 | 126,828 | 0.0079 | 3,926.7 | 1,000 | 2072.7 | 30.96 |
| Salsa20, 16h | 1.384 | 54.8 | 39.6 | 61,416 | 0.0163 | 3,366.0 | 400 | 1863.1 | 21.92 |
| Salsa20, 32h | 1.896 | 68.5 | 36.1 | 50,920 | 0.0196 | 3,486.3 | 300 | 1983.3 | 20.54 |
| AES [12]* | 0.237 | - | - | 118,054 | 0.0085 | - | 500 | - | - |
| AES [13]* | 0.001 | - | - | 1,421,064 | 0.0007 | - | 10,160 | - | - |
| Better is: | higher | lower * | lower | lower | higher | lower | lower * | lower | lower *** |

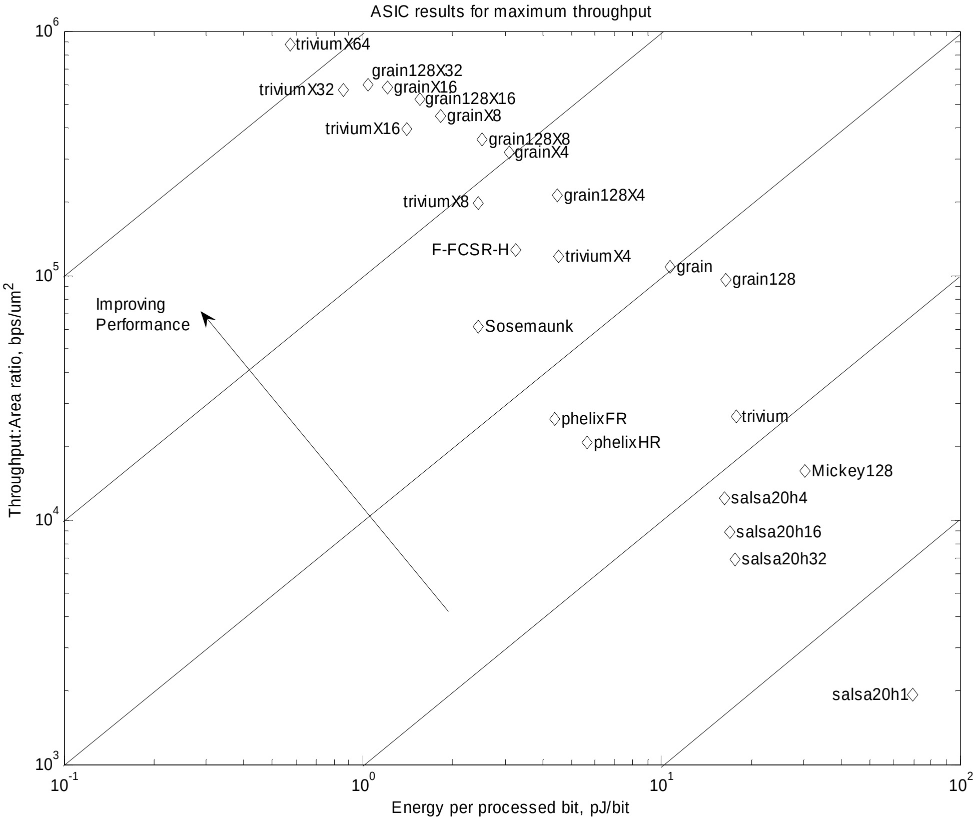


**Fig. 1.** 0.13um Standard Cell CMOS design performance metrics at maximum throughput

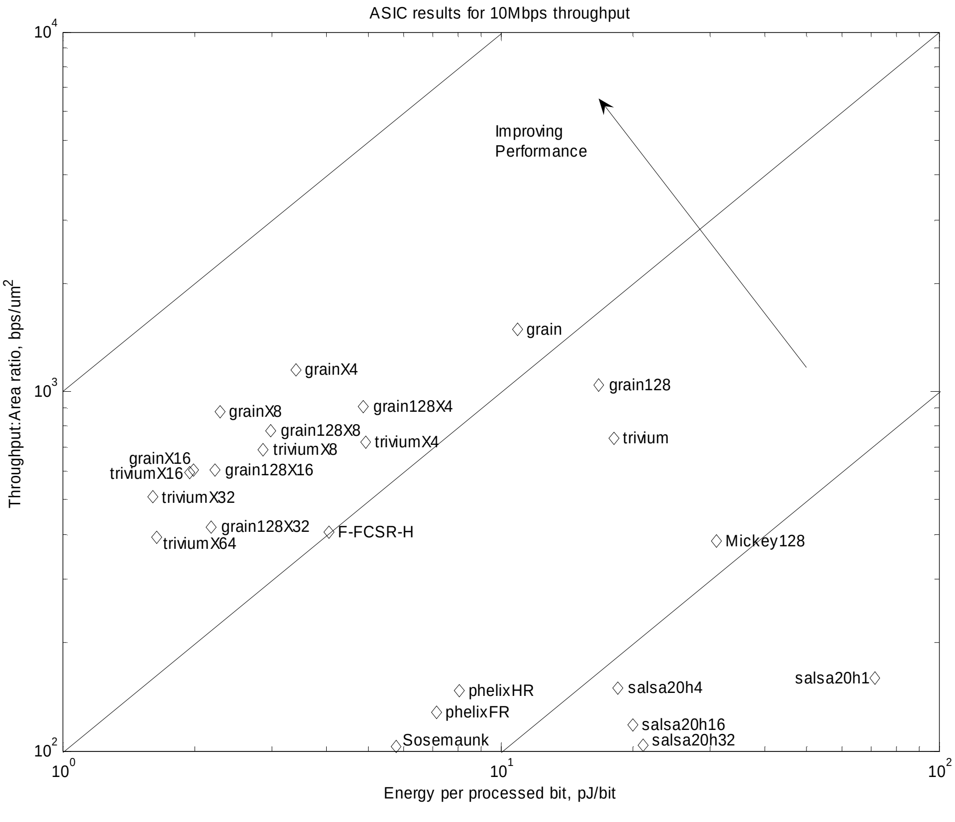


**Fig. 2.** Performance metrics for notional Wireless-LAN at 10Mbps throughput

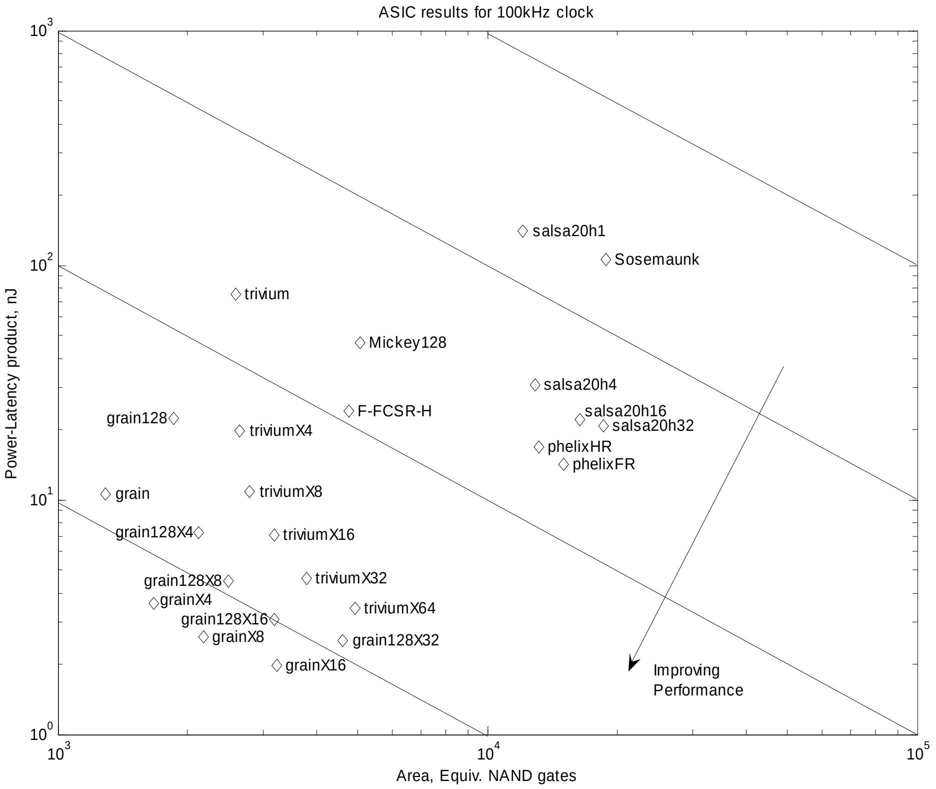


**Fig. 3.** Performance for low-end RFID/WSN application at 100kHz clock

## 6 Recommendations

In carrying out the designs to support this analysis a number of issues were encountered which caused additional engineering effort to be expended. These are documented here as the "authors experience" for consideration by others in future revisions of documentation. These have been formulated with a view to reducing the effort needed by other engineers when implementing and testing the various ciphers.

(1) Strong revision control in which each source file contains CVS tokens for identifier and revision, for example "$ID: $". The revision status of at least one candidate was found to be in question!

(2) Clearly state values for all parameters used in formulation of the each specifically proposed cipher. More than one paper is written mathematically to give a family of ciphers and much digging is needed to locate the required parameters and constants needed for implementation.

(3) Define the precise meaning and bit/nibble ordering for the binary/hexadecimal representation of quoted values and test vectors. It should be noted that for a few of the candidates the conversion between binary and hexadecimal representation of the test vectors is not so obvious.

(4) Test vectors need to be stated for all candidates. Specific to assisting the hardware engineer, the all zeros key and IV case is very helpful together with cases where each bit in turn is individually set.

(5) The reference C-code should include an optional output of the internal state of the cipher at each intermediate major operation (approximates to clock cycles).

## 7 Conclusions

This treatment only considered the "free-for-all" candidates. 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 various ciphers can be ranked according to their key size and the most relevant hardware performance metrics for the application area in a suggested priority order for further cryptanalysis effort.

| Application | KeySize | Cryptanalysis priority from hardware perspective | |
|---|---|---|---|
| | | 1st (Priority) | 2nd |
| WLAN | 80 | Grain & Trivium | F-FCSR-H |
| RFID/WSN | 80 | Grain & Trivium | |
| WLAN | 128 | Grain128 | Mickey128 |
| RFID/WSN | 128 | Grain128 | |
| WLAN | 256 | Sosemanuk & Phelix | Salsa20 |

With regard to Sosemanuk, the utility as a hardware cipher is clear thus in our opinion requires adding to the hardware focus profile. The argument is not so clear cut for Salsa20 or Mickey128 given our results, however, other designers may do better.

## 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] The eStream web site, http://www.ecrypt.eu.org/stream/

[2] D. J. Bernstein, “Comparison of 256-bit stream ciphers at the beginning of 2006”, SASC06, 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] F. K. Gürkaynak, P. Luethi, “Recommendations for Hardware Evaluation of Cryptographic Algorithms”, available at: www.ecrypt.eu.org/stream

[5] 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

[6] H. Zhang, L. L. X. Wang, “Fast Correlation Attack on Stream Cipher ABC v3”, available at: www.ecrypt.eu.org/stream

[7] M. N. Plasencia, “Cryptanalysis of Achterbahn-128/80” available at: www.ecrypt.eu.org/stream

[8] S. Babbage, C. Cid, N. Pramstaller and H. Raddum “Cryptanalysis of Hermes8F”, available at: www.ecrypt.eu.org/stream

[9] F. K. Gürkaynak, P. Luethi, N. Bernold, R. Blattmann, V. Goode, M. Marghitola, H. Kaeslin, N. Felber and W. Fichtner, “Hardware Evaluation of eSTREAM Candidates: Achterbahn, Grain, MICKEY, MOSQUITO, SFINKS, Trivium, VEST, ZK-Crypt”, SASC06, available at: www.ecrypt.eu.org/stream

[10] J. Y. Cho and J. Pieprzyk, “Multiple Modular Additions and Crossword Puzzle Attack on NLSv2” , available at: www.ecrypt.eu.org/stream

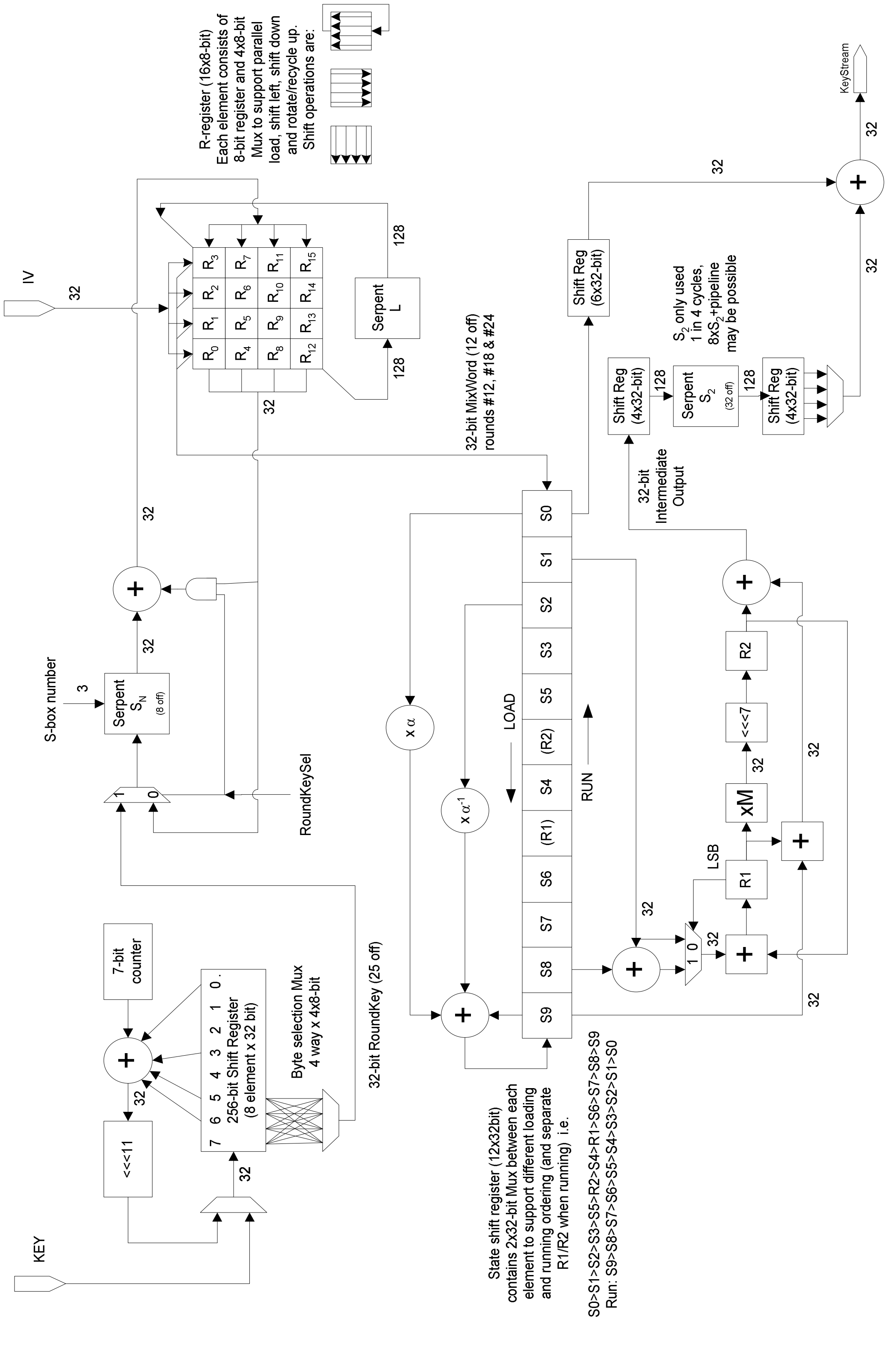
[11] H. Wu and B. Preneel, “Key Recovery Attack on Py and Pypy with Chosen IVs”, available at: www.ecrypt.eu.org/stream

[12] 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.

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

# Appendix A

Block diagram of Sosemanuk implementation datapath


R-register (16x8-bit)
Each element consists of 8-bit register and 4x8-bit Mux to support parallel load, shift left, shift down and rotate/recycle up.
Shift operations are:
KeyStream
32
32
32
128
128
IV
32
R$_0$ R$_1$ R$_2$ R$_3$
R$_4$ R$_5$ R$_6$ R$_7$
R$_8$ R$_9$ R$_{10}$ R$_{11}$
R$_{12}$ R$_{13}$ R$_{14}$ R$_{15}$
Serpent L
32
Shift Reg (6x32-bit)
S$_2$ only used 1 in 4 cycles, 8xS$_2$+pipeline may be possible
Shift Reg (4x32-bit)
128
Serpent S$_2$ (32 off)
128
Shift Reg (4x32-bit)
32-bit MixWord (12 off) rounds #12, #18 & #24
32-bit Intermediate Output
32
S-box number
3
Serpent S$_N$ (8 off)
32
+
1 0
RoundKeySel
S0 S1 S2 S3 S5 (R2) S4 (R1) S6 S7 S8 S9
LOAD
RUN
x α
x α$^{-1}$
+
+
1 0
32
+
R1
LSB
+
xM
32
<<<7
R2
+
32
32
32
7-bit counter
+
32
<<<11
7 6 5 4 3 2 1 0
256-bit Shift Register (8 element x 32 bit)
Byte selection Mux 4 way x 4x8-bit
32-bit RoundKey (25 off)
32
KEY
State shift register (12x32bit) contains 2x32-bit Mux between each element to support different loading and running ordering (and separate R1/R2 when running) i.e.
S0>S1>S2>S3>S5>R2>S4>R1>S6>S7>S8>S9
Run: S9>S8>S7>S6>S5>S4>S3>S2>S1>S0