# ASIC Implementations of the Block Cipher SEA for Constrained Applications

François Macé, François-Xavier Standaert, Jean-Jacques Quisquater

Université Catholique de Louvain
DICE - Microelectronics Laboratory
UCL Crypto Group

RFIDSEC 2007

# SEA - Design Principles

- Feistel structure
- Parametric block cipher
- Limited instruction set
- Sbox computation $\longrightarrow$ recursivity + bitslice

$\longrightarrow$ *Targets resource constrained systems*
$\longrightarrow$ *Initially designed for software implementation*[1]

[1] On Atmel ATiny : $SEA_{96,8}$ : 1 byte or RAM, 32 Regs, 386 bytes for Code size, 17745 Clock Cycles

# SEA - Functional Details

- Important Parameters :
  - $n$ : plaintext size, key size
  - $b$ : word size
  - $n_b$ : number of words per Feistel branch
  - $n_r$ : number of block cipher rounds
  - Constraint : $n = x * 6 * b$, $x \epsilon \mathbb{N}$
- Limited Instruction Set (Bitwise XOR, mod $2^b$ Addition, 3-Bit Substitution box bitwise applied, Word Rotation R, Bit Rotation r)

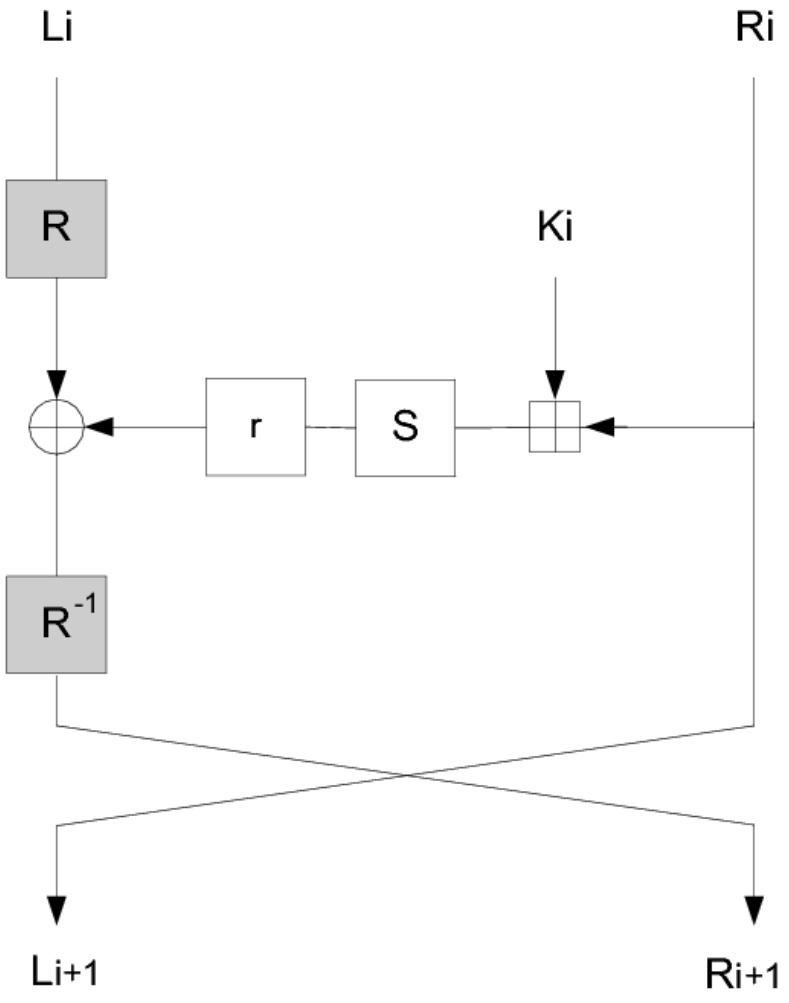
# SEA - Round

- Encryption :

$$[L_{i+1}, R_{i+1}] = F_E(L_i, R_i, K_i) \Leftrightarrow$$
$$R_{i+1} = R(L_i) \oplus r(S(R_i \boxplus K_i))$$
$$L_{i+1} = R_i$$

- Decryption :

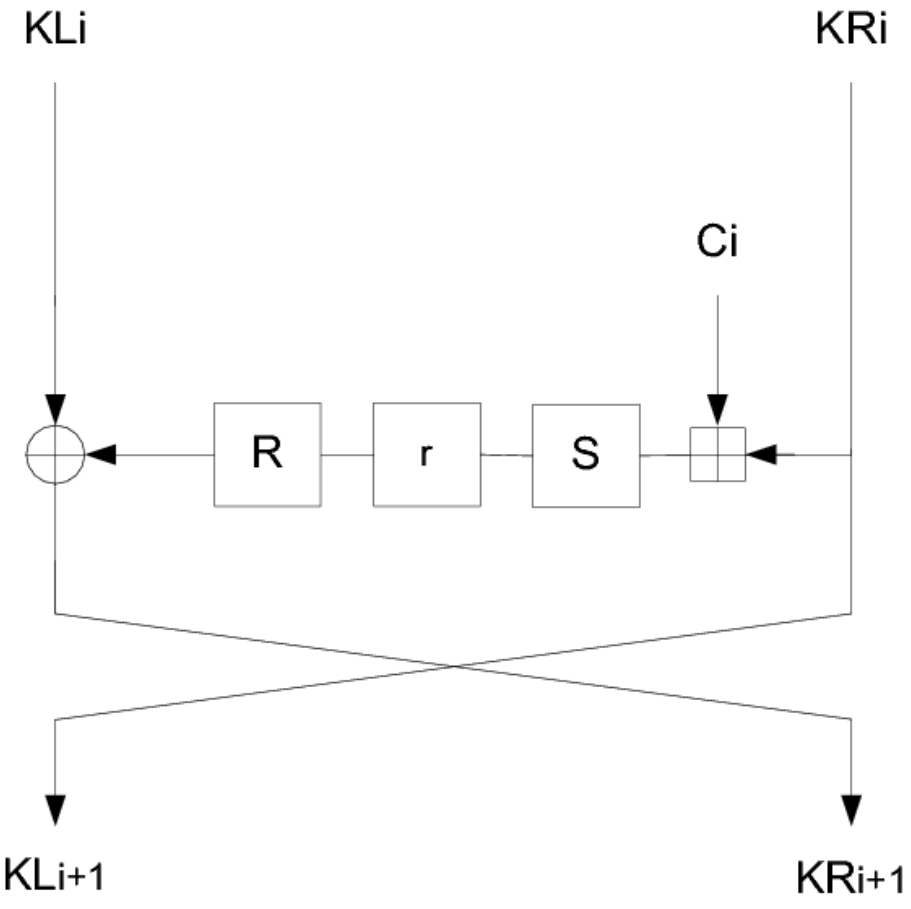
$$[L_{i+1}, R_{i+1}] = F_D(L_i, R_i, K_i) \Leftrightarrow$$
$$R_{i+1} = R^{-1}(L_i \oplus r(S(R_i \boxplus K_i)))$$
$$L_{i+1} = R_i$$



# SEA - KeySchedule

- Key Schedule :

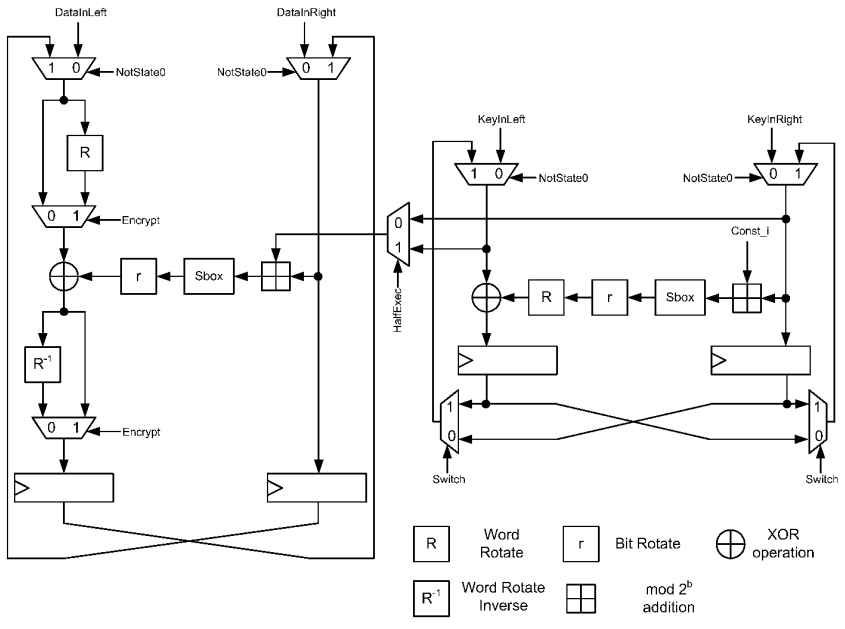
$$[KL_{i+1}, KR_{i+1}] = F_K(KL_i, KR_i, C_i) \Leftrightarrow$$
$$KR_{i+1} = KL_i \oplus R(r(S(KR_i \boxplus C_i)))$$
$$KL_{i+1} = KR_i$$



# Design Principles and Architecture

→ Direct Mapping of the Feistel Structure

- One Round per clock cycle
- On the fly computation of round keys
- Parametric description using *Generic VHDL* encoding
- $\frac{n}{2}$-bit operands
- Resource consuming blocks :
  - Sbox
  - mod $2^b$ adders



# Generic VHDL Coding

- *mod* $2^b$ *adders :*
  - Round Function : $n_b$ b bit adders without carry propagation between them
  - Key Schedule : `Const_i` $\epsilon\{0, ..., \frac{n_r}{2}\} \Rightarrow \lceil \frac{log_2(\frac{n_r}{2})}{b} \rceil$ b bits adders are necessary
- Sbox, R, $R^{-1}$ and r can easily be generically written for any set of n, b, $n_b$ parameters ;
- $n_r$ can be externally set or automatically computed from $n_r = [3\frac{n}{4} + 2(\frac{n}{2b} + \frac{b}{2})](+1)$

# Implementation Results and Comparison with other Block Ciphers

| Algo. | n | b | $n_r$ | Clock Freq. [MHz] | Throughput [Mbps] | Area [$\mu m^2$] | Gate Equ. @ Synt. | Gate Equ. @ P& R | Power [$\mu$W] |
|---|---|---|---|---|---|---|---|---|---|
| SEA | 96 | 8 | 93 | 250 | 258 | 22362 | 3758 | 4313 | 5102.64 |
| SEA | 108 | 6 | 111 | 250 | 243 | 23668 | 4003 | 4565 | 5844.02 |
| SEA | **126** | **7** | **117** | **250** | **269** | **28241** | **4770** | **5447** | **7216.96** |
| SEA | 132 | 11 | 121 | 250 | 273 | 29638 | 5071 | 5715 | 7894.62 |
| SEA | 144 | 4 | 149 | 250 | 242 | 32894 | 5764 | 6345 | 8029.56 |
| SEA | 144 | 6 | 139 | 250 | 259 | 32137 | 5525 | 6199 | 7789.28 |
| SEA | 144 | 8 | 135 | 250 | 267 | 31523 | 5427 | 6079 | 8201.22 |
| SEA | 144 | 12 | 133 | 250 | 271 | 31622 | 5550 | 6100 | 8183.44 |
| AES-Satoh | 128 | - | 10 | 224 | 2609.11 | 130 000 | - | 21337 | - |
| AES-Hodjat | 128 | - | 10 | 295 | 3840 | 790 000 | - | 73200 | 86 000 |
| ICEBERG | 64 | - | 16 | 250 | 1000 | 45679 | 7732 | 8811 | 9577.11 |

- Trade throughput for Area $\longrightarrow$ Consequence on power consumption
- Different Optimization goals : SEA $\longrightarrow$ SW code size, ICEBERG $\longrightarrow$ max thrpt/area ratio.

# Design Principles for Reduced Datapath

- Fixed value of the parameter $n_b = 6$
- Purpose :
  - Reduce the area consumption
  - Reduce the power consumption
  - Support both encryption and decryption
  - Achieve a good tradeoff between area, power and throughput
  - Operations on b-bit operands

# Transformed Algorithm - Round Function

**Input :** $R_i, L_i, RK_i \in \mathbf{Z}_{2^b}^{n_b}$

**Output :** $R_{i+1}, L_{i+1}$

| | E/D | Encryption | Decryption |
|---|---|---|---|
| 1 : | $A \leftarrow R_{i,0} + RK_{i,0}$ ; | | |
| 2 : | $B \leftarrow R_{i,1} + RK_{i,1}$ ; | | |
| 3 : | $C \leftarrow R_{i,2} + RK_{i,2}$ ; | | |
| 4 : | $(D, E, F) \leftarrow r(S(A, B, C))$ ; | $A \leftarrow R_{i,3} + RK_{i,3}$ ; | $C \leftarrow R_{i,5} + RK_{i,5}$ ; |
| | | $G \leftarrow L_{i,5}$ ; | $G \leftarrow R_{i,5}$ ; |
| 5 : | $B \leftarrow R_{i,4} + RK_{i,4}$ ; | $G \leftarrow L_{i,0}$ ; | $R_{i+1,5} \leftarrow L_{i,0} \oplus D$ ; |
| | $L_{i+1,0} \leftarrow R_{i,0}$ ; | $R_{i+1,0} \leftarrow D \oplus G$ ; | |
| 6 : | $L_{i+1,1} \leftarrow R_{i,1}$ ; | $R_{i+1,1} \leftarrow E \oplus G$ ; $G \leftarrow L_{i,1}$ ; | $A \leftarrow R_{i,3} + RK_{i,3}$ ; |
| | | $C \leftarrow R_{i,5} + RK_{i,5}$ ; | $R_{i+1,0} \leftarrow L_{i,1} \oplus E$ ; |
| 7 : | $(D, E, F) \leftarrow r(S(A, B, C))$ ; | $R_{i+1,2} \leftarrow F \oplus G$ ; | $R_{i+1,1} \leftarrow L_{i,2} \oplus F$ ; |
| | $L_{i+1,2} \leftarrow R_{i,2}$ ; | $G \leftarrow L_{i,2}$ ; | |
| 8 : | $L_{i+1,3} \leftarrow R_{i,3}$ | $R_{i+1,3} \leftarrow D \oplus G$ ; $G \leftarrow L_{i,3}$ ; | $R_{i+1,2} \leftarrow L_{i,3} \oplus D$ ; |
| 9 : | $L_{i+1,4} \leftarrow R_{i,4}$ ; | $G \leftarrow L_{i,4}$ ; $R_{i+1,4} \leftarrow E \oplus G$ ; | $R_{i+1,3} \leftarrow E \oplus L_{i,4}$ ; |
| 10 : | | $R_{i+1,5} \leftarrow F \oplus G$ ; | $R_{i+1,4} \leftarrow L_{i,5} \oplus F$ ; |
| | | $L_{i+1,5} \leftarrow R_{i,5}$ ; | $L_{i+1,5} \leftarrow G$ ; |

# Transformed Algorithm - Key Schedule

**Input :** $KR_i$, $KL_i \in \mathbf{Z}_{2^b}^{n_b}$, $Const_i \in \mathbf{Z}_{2^b}$

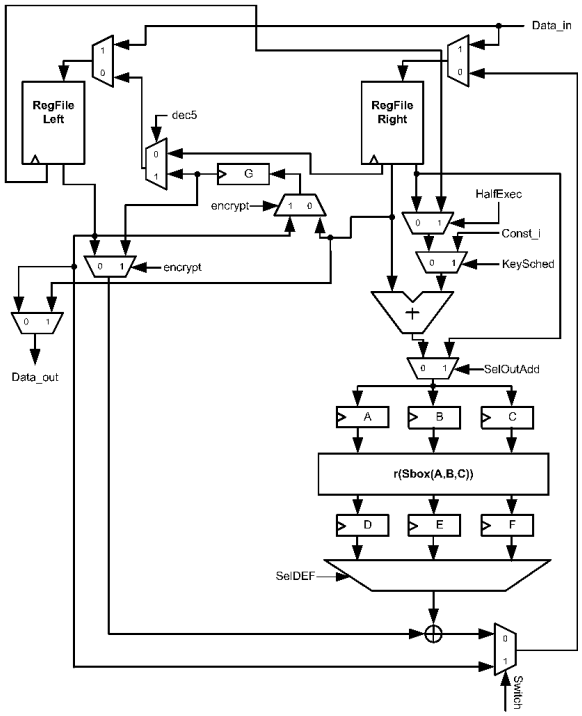
**Output :** $kR_{i+1}$, $kL_{i+1}$

| | E/D | Encryption | Decryption |
|---|---|---|---|
| 1 : | $Ak \leftarrow KR_{i,0} + Const_i$ ; | | |
| 2 : | $Bk \leftarrow KR_{i,1}$ ; | | |
| 3 : | $Ck \leftarrow KR_{i,2}$ ; | | |
| 4 : | $(Dk, Ek, Fk, ) \leftarrow r(S(1k, Bk, Ek))$ ; | $Ak \leftarrow KR_{i,3}$ ; | $Ck \leftarrow KR_{i,5}$ ; |
| 5 : | $Bk \leftarrow KR_{i,4}$ ; $KR_{i+1,1} \leftarrow KL_{i,1} \oplus Dk$ ; $KL_{i+1,1} \leftarrow KR_{i,1}$ | | |
| 6 : | $KR_{i+1,2} \leftarrow KL_{i,2} \oplus Ek$ ; $KL_{i+1,2} \leftarrow KR_{i,2}$ ; | $Ck \leftarrow KR_{i,5}$ ; | $Ak \leftarrow KR_{i,3}$ ; |
| 7 : | $(Dk, Ek, Fk, ) \leftarrow r(S(Ak, Bk, Ek))$ ; $KL_{i+1,3} \leftarrow KR_{i,3}$ ; $KR_{i+1,3} \leftarrow KL_{i,3} \oplus Fk$ ; | | |
| 8 : | $KR_{i+1,0} \leftarrow KL_{i,0} \oplus Fk$ ; $KL_{i+1,0} \leftarrow KR_{i,0}$ ; | | |
| 9 : | $KR_{i+1,4} \leftarrow KL_{i,4} \oplus Dk$ ; $KL_{i+1,4} \leftarrow KR_{i,4}$ ; | | |
| 10 : | $KR_{i+1,5} \leftarrow KL_{i,5} \oplus Ek$ ; $KL_{i+1,5} \leftarrow KR_{i,5}$ ; | | |

# Implementation Structure

- Shared resources between Round and Keychedule
- I/O functionality
- Concomitant execution of :
  - $k_1$ and $r_8$
  - $k_2$ and $r_9$
  - $k_3$ and $r_{10}$
  - $r_1$ and $k_9$
  - $r_2$ and $k_{10}$

TOTAL : $33 + 15 * n_r$ cycles.



# Results and Comparison

| b | n | nr | ♯ Cycles | Throughput [Mbps] | Area [$\mu m^2$] | Gate Equ. @ Synt. | Gate Equ. @ P& R | Leak. Power [$\mu W$] | Power 80 MHz [$\mu$W] | Power 100kHz [$\mu W$] |
|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 96 | 93 | 1428 | 5.38 | 23186 | 3925 | 4472 | 17.453 | 1376 | 19.238 |
| 9 | 108 | 99 | 1518 | 5.69 | 25294 | 4281 | 4879 | 18.693 | 1546 | 20.527 |
| 10 | 120 | 113 | 1600 | 6 | 27606 | 4673 | 5325 | 19.911 | 1598 | 21.923 |
| 11 | 132 | 121 | 1712 | 6.17 | 29742 | 5035 | 5737 | 20.287 | 1664 | 23.101 |
| 12 | 144 | 133 | 1880 | 6.13 | 31342 | 5406 | 6046 | 22.351 | 1886 | 24.682 |

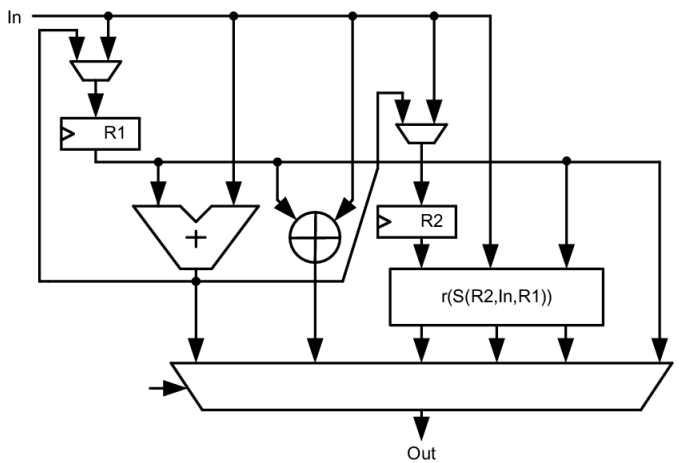
| AES | Width [bit] | Equ. Gate | Process [$\mu m$] | Freq [MHz] | Latency [♯ cycles] | Thrpt [Mbps] | Power 80 MHz [$\mu W$] | Enc/Dec |
|---|---|---|---|---|---|---|---|---|
| Satoh et al. | 32 | 5400 | 0.11 | 131 | 54 | 311 | - | yes |
| Feldhoffer et al. | 8 | 3600 | 0.35 | - | 1016 | - | - | no |
| Pramstaller et al. | 32 | 8500 | 0.6 | 50 | 92 | 70 | - | yes |
| Hämäläinen et al. | 8 | 3200 | 0.13 | 130 | 160 | 104 | 2400 | no |
| Hämäläinen et al. | 8 | 3100 | 0.13 | 152 | 160 | 121 | 2960 | no |

- % Loop Arch. $\rightarrow$ low area gain due to the I/O interface but improved power consumption
- AES $\rightarrow$ better area and/or thrgpt but higher power (cfr Techno).

# Proposal

$\longrightarrow$ SEA designed for small-code SW implementations
$\Rightarrow$ Minimal dedicated datapath with low throughput

- Dual ported 32 words RAM (data + working regs)
- $\sim$ 50 cycles/round with $n_b = 6$




- Close to SW approach ($\uparrow$ memory access, $\downarrow$ power consumption) [2]

[2] For $SEA_{96,8}$, reduction to 25% of number of cycles required on ATiny

# Results and Comparison

| b | Equ. Gate @ Synthesis | Leakage [$\mu W$] | Total Power 100kHZ [$\mu W$] | Total Power 80MHz [$\mu W$] |
|---|---|---|---|---|
| 8 | 449 | 2.865 | 3.218 | 293.5 |
| 9 | 507 | 3.083 | 3.421 | 308.8 |
| 10 | 563 | 3.246 | 3.636 | 328.6 |
| **11** | **620** | **3.499** | **3.878** | **346.1** |
| 12 | 677 | 3.704 | 4.128 | 357.6 |

For AES [Feldhoffer-2005] $\rightarrow$ datapath $\pm$950 gates (28% of 3400 gates)

# Conclusions

- Parametric description achievable in VHDL for different architectures
- Loop Architecture :
  - High scalability properties
  - Good Area/Throuput tradeoff
- Reduced datapath with I/O :
  - Reduced scalability
  - More realistic
  - Reduced power
- Direction for a minimum datapath
- Possibility to trade security with energy (reduce $n_r$)
- Further improvements with modification of the Sbox, ...