KATAN & KTANTAN
A Family of Small and Efficient Hardware-Oriented Block Ciphers

Christophe De Cannière[1], Orr Dunkelman[1,2], Miroslav Knežević[1]

[(1)]Katholieke Universiteit Leuven, ESAT/SCD-COSIC

[(2)]Département d'Informatique, École normale supérieure

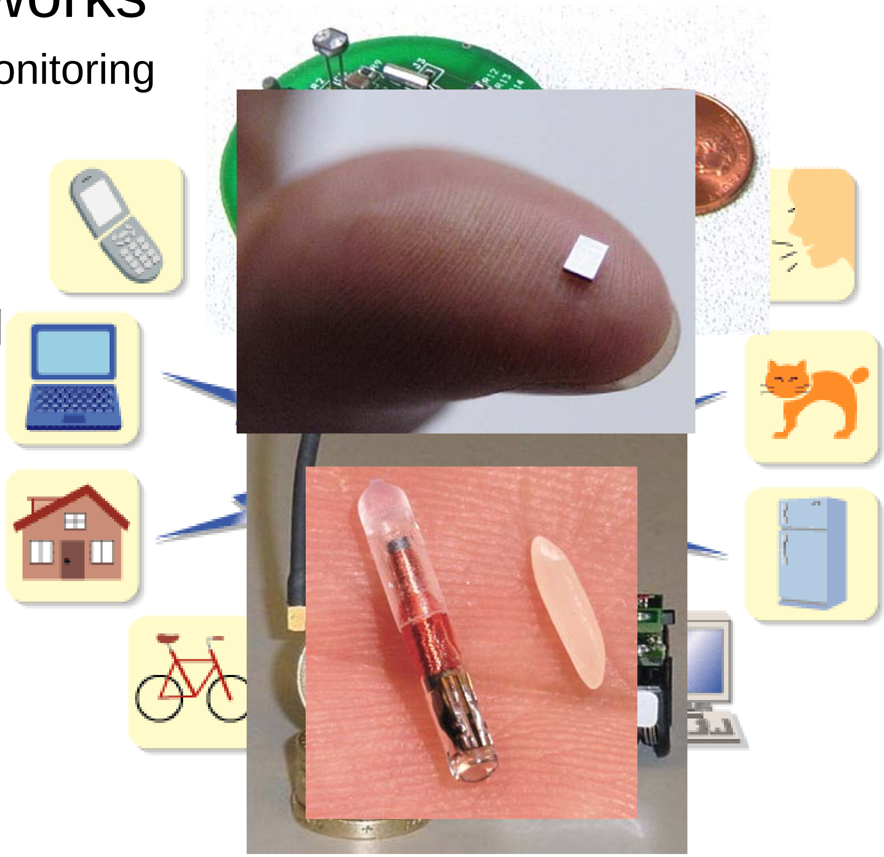
CHES 2009 September 8, 2009

Outline

- Motivation
  - Why do we fight for a single gate?
  - What are the options so far?
  - Design Goals
- Design Rationale
  - Memory Issues
  - Control part
  - Possible Speed-Ups
- Implementation Results
- Conclusion

Why do we fight for a single gate?

- Wireless Sensor Networks
  - Environmental and Health Monitoring
  - Wearable Computing
  - Military Surveillance, etc.
- Pervasive Computing
  - Healthcare
  - Ambient Intelligence
- Embedded Devices
- It's a challenge!



What are the options so far?

- Stream ciphers
  - To ensure security, the internal state must be twice the size of the key.
  - No good methodology on how to design these.
- Use the standardized block cipher: AES
  - The smallest implementation consumes 3.1 Kgates.
  - Recent attacks in the related-key model.
- Other block ciphers?
  - HIGHT, mCrypton, DESL, PRESENT,…
  - Can we do better/different?

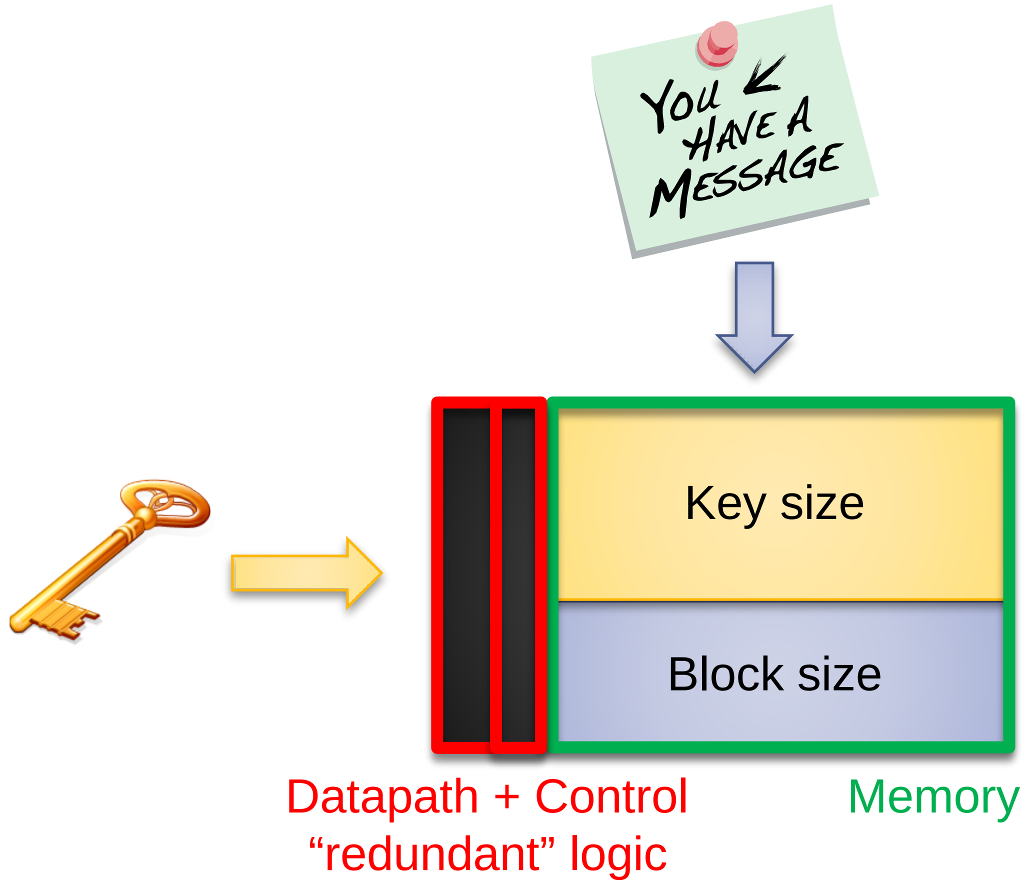
Design Goals

- Secure block cipher
  - Address Differential/Linear cryptanalysis, Related-Key/Slide attacks, Related-Key differentials, Algebraic attacks.
- Efficient block cipher
  - Small foot-print, Low power consumption, Reasonable performance (+ possible speed-ups).
- Application driven
  - Does an RFID tag always need to support a key agility?
  - Some low-end devices have one key throughout their life cycle.
  - Some of them encrypt very little data.
  - Why wasting precious gates if not really necessary?

The KATAN/KTANTAN Block Ciphers

- Block ciphers based on Trivium (its 2 register version–Bivium).
- Block size: 32/48/64 bits.
- Key size: 80 bits.
- Share the same number of rounds – 254.
- KATAN and KTANTAN are the same up to the key schedule.
- In KTANTAN, the key is fixed and cannot be changed!

Block Cipher – HW perspective



Design Rationale – Memory Issues (1)

- The more compact the cipher is, a larger ratio of the area is dedicated for storing the intermediate values and key bits.
- Difference not only in basic gate technology, but also in the size of a single bit representation.

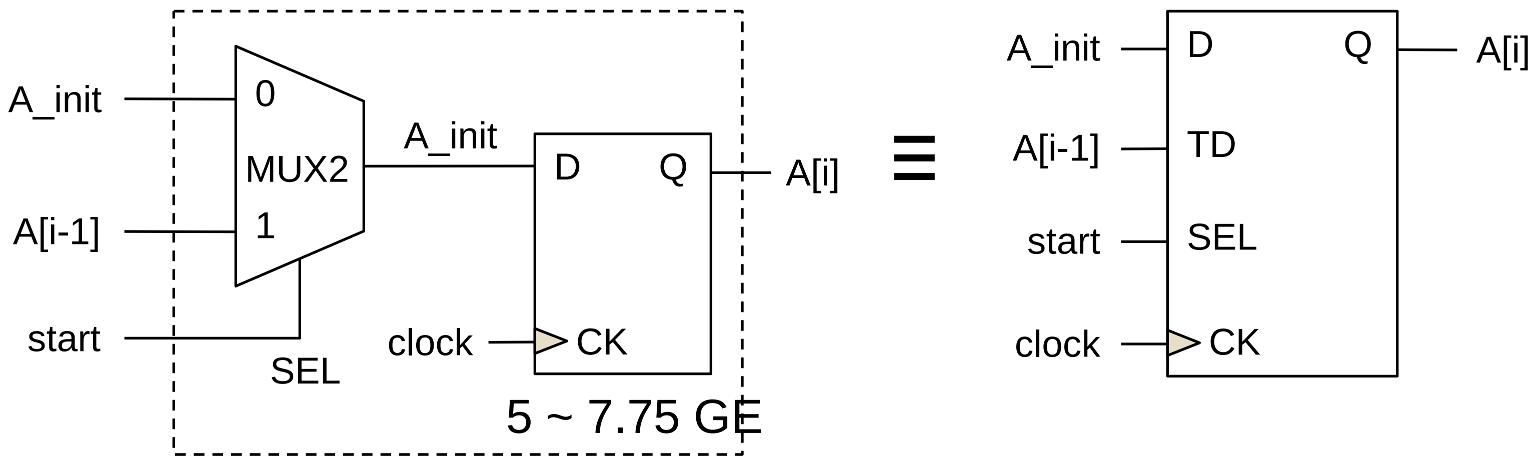
Cipher	Block [bits]	Key [bits]	Technology [μm]	Size [GE]	Memory [%]	Memory/bit [GE]
AES-128 [8]	128	128	0.35	3400	**60**	**7.97**
AES-128 [10]	128	128	0.13	3100	**48**	**5.8**
HIGHT [12]	64	128	0.25	3048	**49**	**~7**
mCrypton [15]	64	64	0.13	2420	**26**	**5**
DES [19]	64	56	0.18	2309	**63**	**12.19**
DESL [19]	64	56	0.18	1848	**79**	**12.19**
PRESENT-80 [4]	64	80	0.18	1570	**55**	**6**
PRESENT-80 [20]	64	80	0.35	1000	**≥80**	**≤ 6**

Design Rationale – Memory Issues (2)

- The gate count (GE) DOES depend on the library and tools that are used during the synthesis.
- Example:
  - PRESENT[20] contains 1,000 GE in 0.35 µm technology – 53,974 μm^2.
  - PRESENT[20] contains 1,169 GE in 0.25 µm technology – 32,987 μm^2.
  - PRESENT[20] contains 1,075 GE in 0.18 µm technology – 10,403 μm^2.
- Comparison is fair ONLY if the SAME library and the SAME tools are used.

Design Rationale – A Story of a Single Bit

- Assume we have a parallel load of the key and the plaintext.
- A single Flip-Flop has no relevance – MUXes need to be used.
- 2to1 MUX + FF = Scan FF: Beneficial both for area and power.



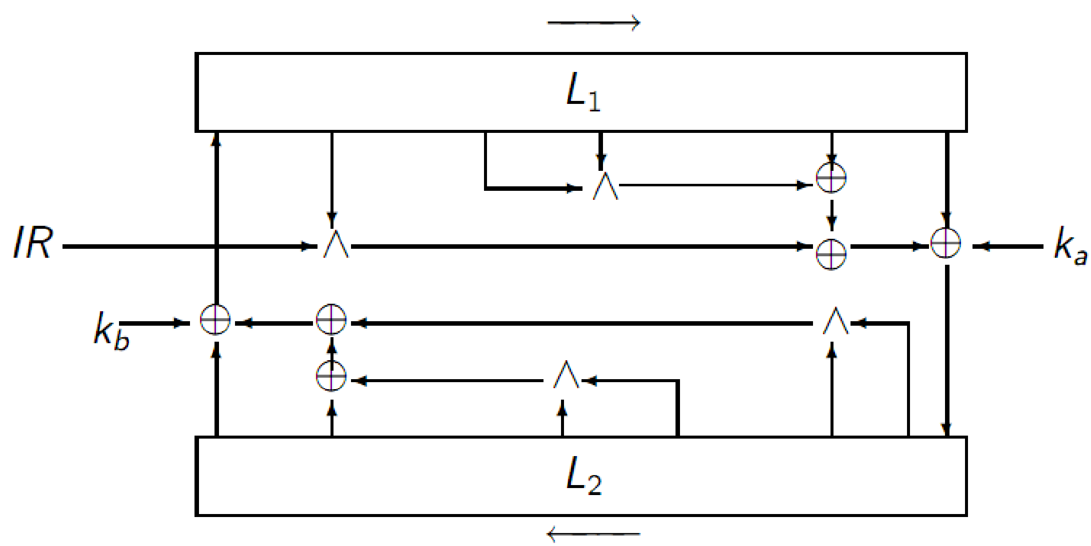
7.25 ~ 13.75 GE

6.25 ~ 11.75 GE

- (64 + 80 + 8) × 6.25 = 950 GE ☺

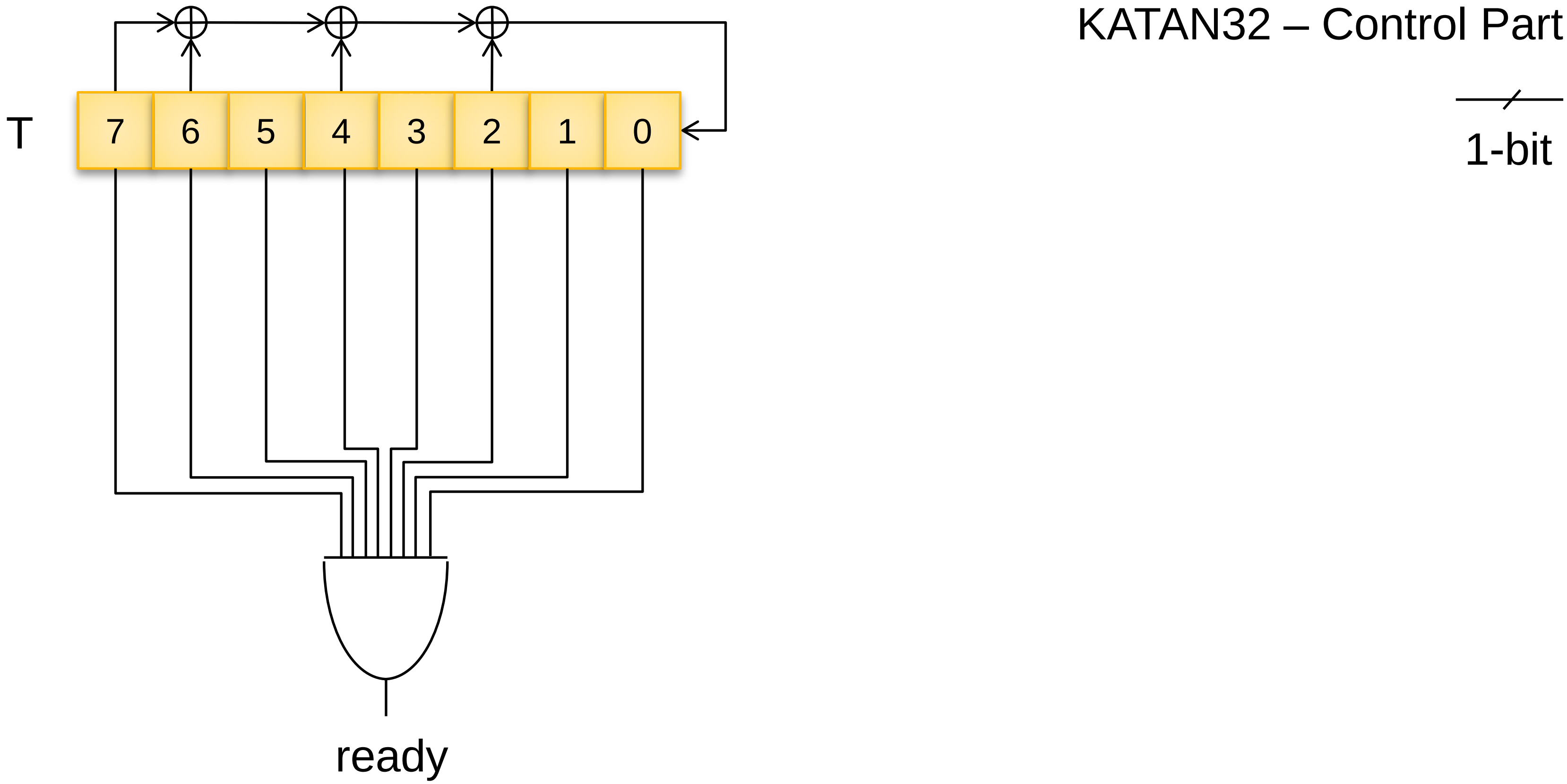
Design Rationale – Control Part

- How to control such a simple construction?

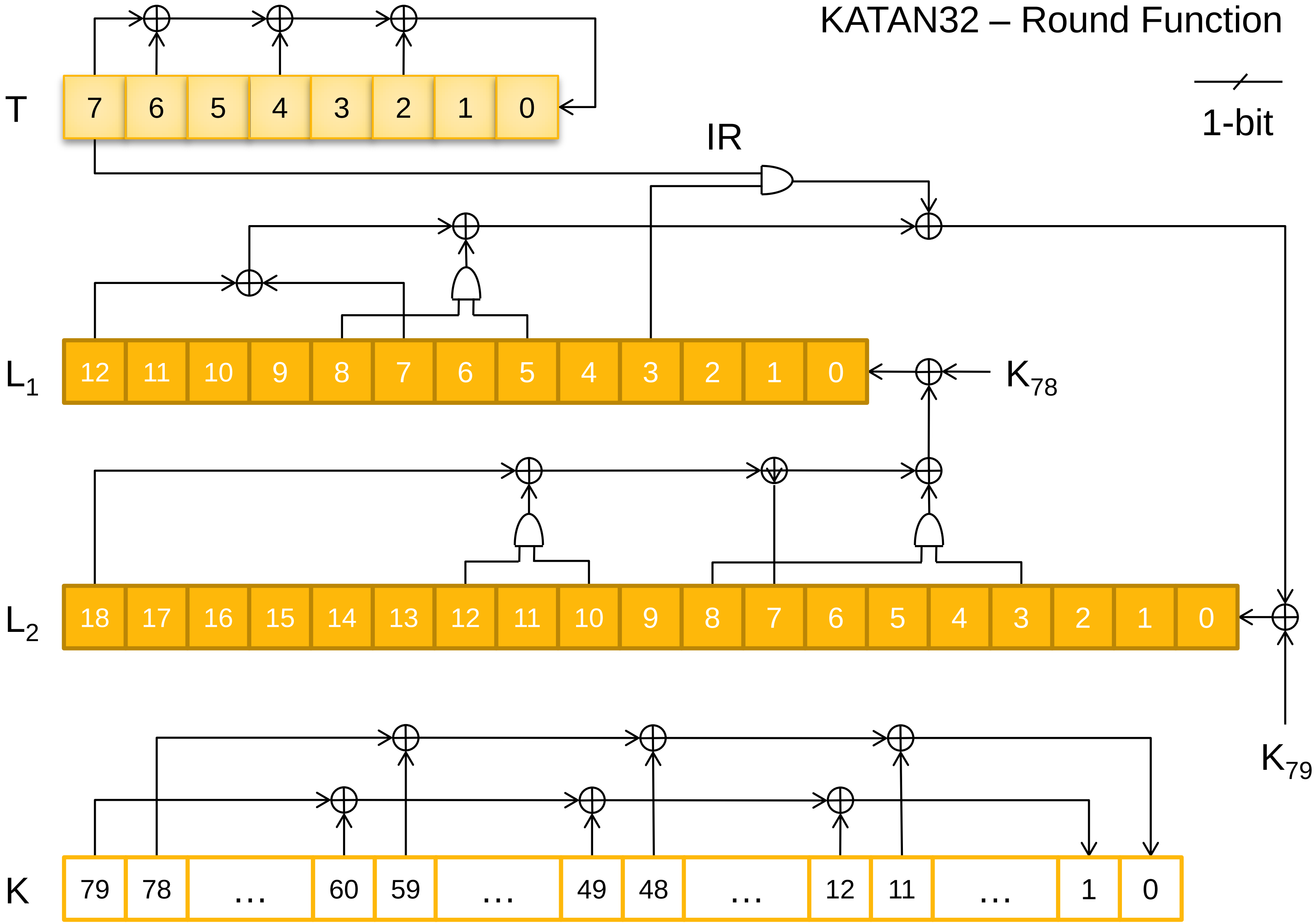


- *IR* stands for *Irregular update Rule*.
- We basically need a counter only. Can it be simpler than that?
- Let the LFSR that is in charge of *IR* play the role of a counter.

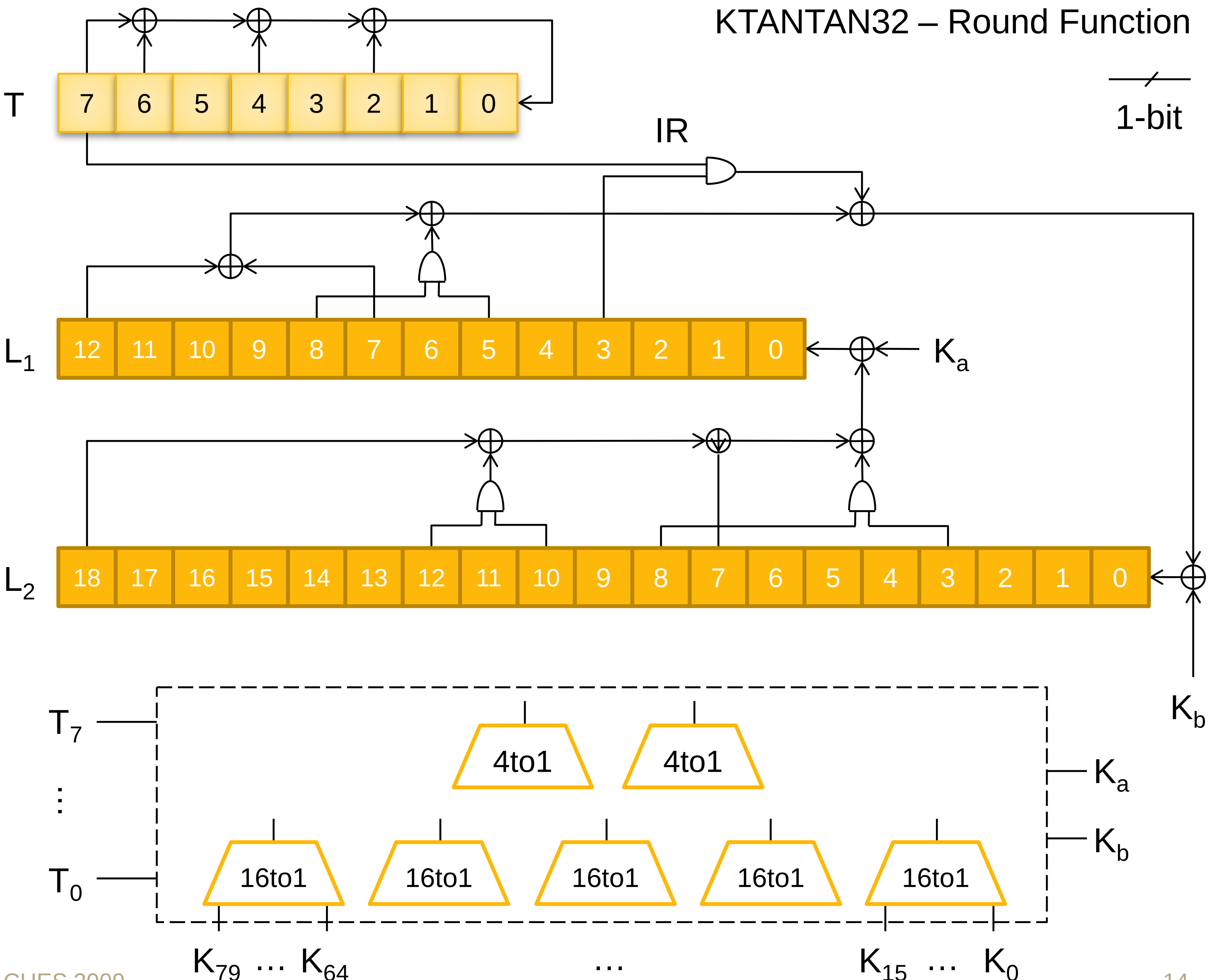
KATAN32 – Control Part



KATAN32 – Round Function



KTANTAN32 – Round Function



Implementation Results

- All designs are synthesized with Synopsys Design Vision version Y-2006.06, using UMC 0.13µm Low-Leakage CMOS library.

Cipher	Block [bits]	Key [bits]	Memory/bit [GE]	Throughput* [Kbps]	Size [GE]
KATAN32	32	80	6.18	12.5	**802**
KATAN48	48	80	6.19	18.8	**927**
KATAN64	64	80	6.15	25.1	**1054**
KTANTAN32	32	80	6.10	12.5	**462**
KTANTAN48	48	80	6.14	18.8	**588**
KTANTAN64	64	80	6.17	25.1	**688**

* A throughput is estimated for frequency of 100 kHz.

1027 GE

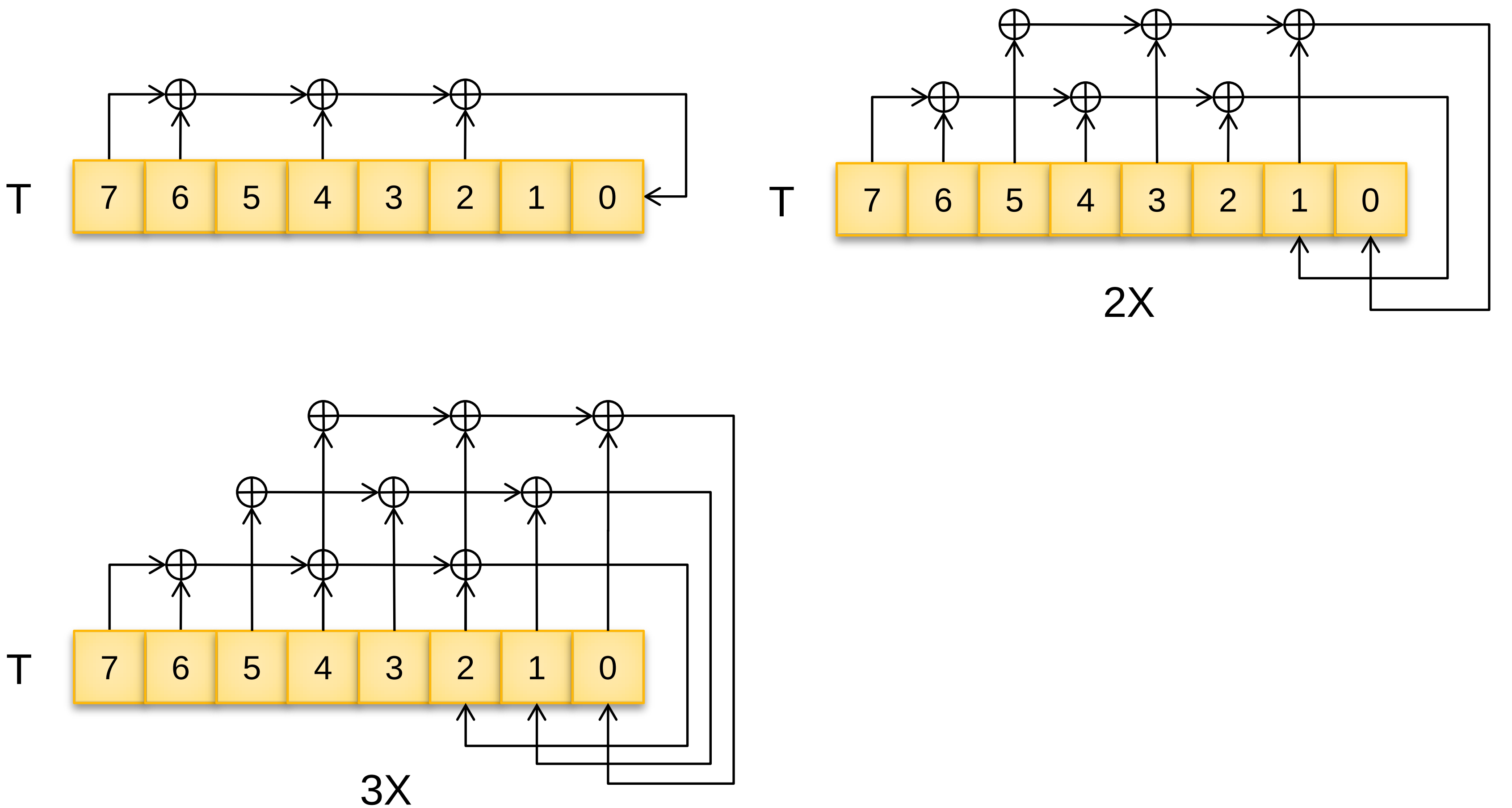
Design Rationale – Memory Issues (3)

- KATAN32 has only 7.5% of “redundant” logic.*

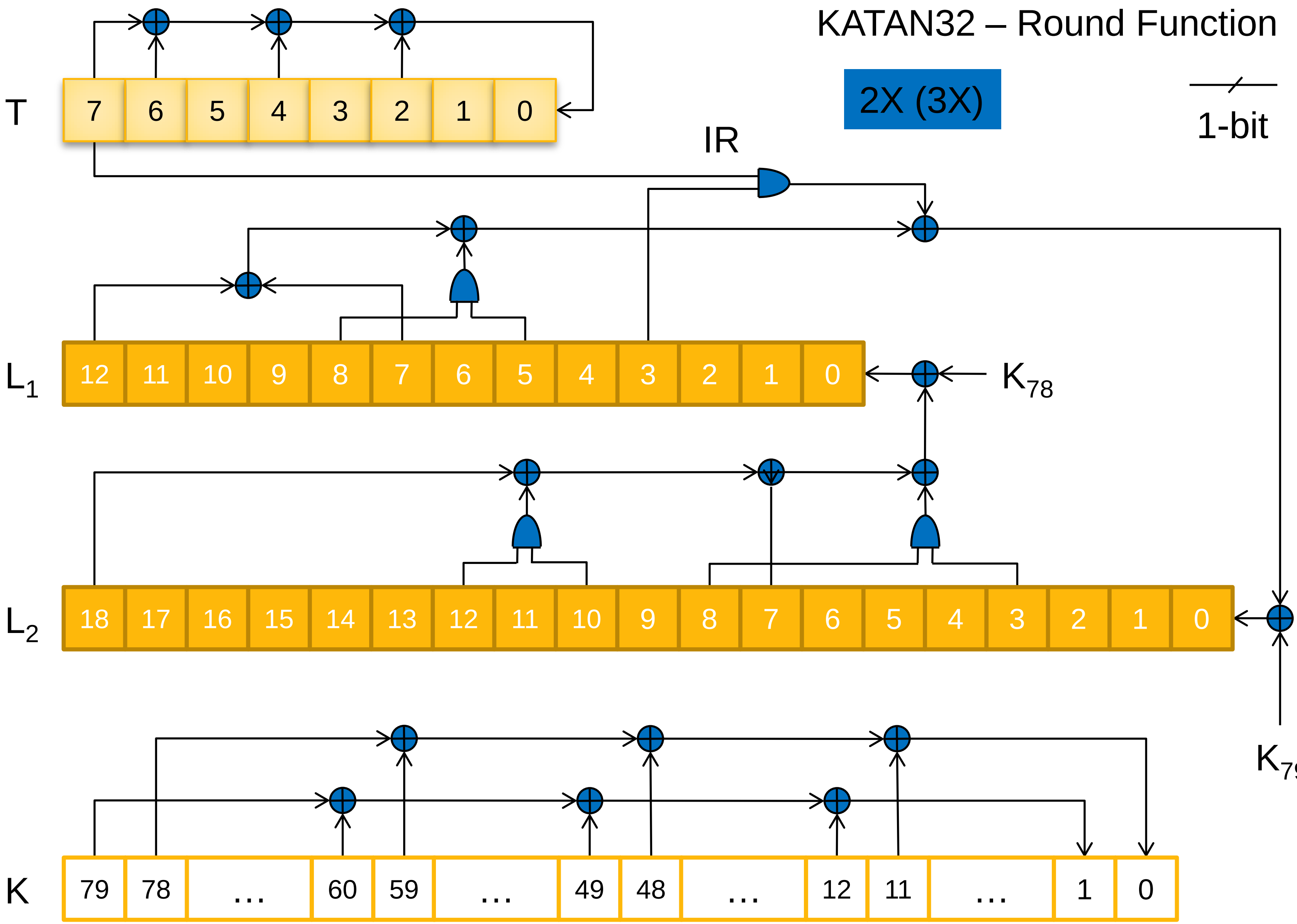
Cipher	Block [bits]	Key [bits]	Size [GE]	Memory /bit [GE]	Memory [GE]	[%]
KATAN32	32	80	802	6.18	742	**92.5**
KATAN48	48	80	927	6.19	842	**90.8**
KATAN64	64	80	1054	6.15	935	**88.7**
KTANTAN32	32	80	462	6.10	244	**52.8**
KTANTAN48	48	80	588	6.14	344	**58.5**
KTANTAN64	64	80	688	6.17	444	**64.5**

* not including controlling LFSR

Possible Speed-Ups



KATAN32 – Round Function



How fast can KATAN/KTANTAN run?

- Optimized for speed, using UMC 0.13μm High-Speed CMOS library, KATAN64 runs up to 1.88 Gbps.

Cipher	Size [GE]	Frequency [GHz]	Throughput [Mbps]
KATAN32	975	2.86	**1071.4**
KATAN48	1201	2.86	**1611.4**
KATAN64	1399	2.50	$\boxed{\textbf{1882.5}}$
KTANTAN32	1328	1.25	**468.7**
KTANTAN48	1677	1.23	**696.3**
KTANTAN64	1589	1.19	**896.4**

Power Consumption

- Synthesis results only!
- Estimated with Synopsys Design Vision version Y-2006.06, using UMC 0.13µm Low-Leakage CMOS library.

Cipher	Size [GE]	Frequency [kHz]	Power [nW]
KATAN32	802	100	**381**
KATAN48	927	100	**439**
KATAN64	1054	100	**555**
KTANTAN32	462	100	**146**
KTANTAN48	588	100	**234**
KTANTAN64	688	100	**292**

- Too optimistic?

Can we go more compact?

- Yes – applies to KATAN48, KATAN64, KTANTAN48 and KTANTAN64.
- Use clock gating – The speed drops down 2-3 times.
- The trick is to “clock” controlling LFSR every two (three) clock cycles.
- The improvement is rather insignificant:
  - 27 GE for KATAN64, 11 GE for KATAN48.
  - 4 GE for KTANTAN64, 17 GE for KTANTAN48.

Can we go even more compact?

- Probably! The speed drops down significantly.
- Serialize the inputs:
  - But, we still need a fully autonomous cipher.
  - Additional logic (counter and FSM) are needed in order to control the serialized inputs. Or try to reuse an LFSR for counting again…
- Combine it with clock gating.
- Worth trying if the compact design is an ultimate goal!

Conclusion

- KATAN & KTANTAN – Efficient, hardware oriented block ciphers based on Trivium.
- Key size: 80 bits; Block size: 32/48/64 bits; Key agility is optional.
- KTANTAN32 consumes only 462 GE (1848 μm^2).
- KATAN32 has only 7.5% of “redundant” logic.
- KATAN64 has a throughput of 1.88 Gbps.

Thank you!

Trade-Offs

Cipher	Block (bits)	Key (bits)	Size (GE)	Gates per Memory Bit	Throughput* (Kb/s)	Logic Process
KATAN32	32	80	802	6.25	12.5	0.13 μm
KATAN32	32	80	846	6.25	25	0.13 μm
KATAN32	32	80	898	6.25	37.5	0.13 μm
KATAN48†	48	80	916	6.25	9.4	0.13 μm
KATAN48	48	80	927	6.25	18.8	0.13 μm
KATAN48	48	80	1002	6.25	37.6	0.13 μm
KATAN48	48	80	1080	6.25	56.4	0.13 μm
KATAN64†	64	80	1027	6.25	8.4	0.13 μm
KATAN64	64	80	1054	6.25	25.1	0.13 μm
KATAN64	64	80	1189	6.25	50.2	0.13 μm
KATAN64	64	80	1269	6.25	75.3	0.13 μm
KTANTAN32	32	80	462	6.25	12.5	0.13 μm
KTANTAN32	32	80	673	6.25	25	0.13 μm
KTANTAN32	32	80	890	6.25	37.5	0.13 μm
KTANTAN48†	48	80	571	6.25	9.4	0.13 μm
KTANTAN48	48	80	588	6.25	18.8	0.13 μm
KTANTAN48	48	80	827	6.25	37.6	0.13 μm
KTANTAN48	48	80	1070	6.25	56.4	0.13 μm
KTANTAN64†	64	80	684	6.25	8.4	0.13 μm
KTANTAN64	64	80	688	6.25	25.1	0.13 μm
KTANTAN64	64	80	927	6.25	50.2	0.13 μm
KTANTAN64	64	80	1168	6.25	75.3	0.13 μm

* — A throughput is estimated for frequency of 100 KHz.
† — Using clock gating.

Non-Linear Functions

$$f_a(L_1) = L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x_4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a$$

$$f_b(L_2) = L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6]) \oplus k_b$$

Cipher	$\lvert L_1 \rvert$	$\lvert L_2 \rvert$	x_1	x_2	x_3	x_4	x_5
KATAN32/KTANTAN32	13	19	12	7	8	5	3
KATAN48/KTANTAN48	19	29	18	12	15	7	6
KATAN64/KTANTAN64	25	39	24	15	20	11	9

Cipher	y_1	y_2	y_3	y_4	y_5	y_6
KATAN32/KTANTAN32	18	7	12	10	8	3
KATAN48/KTANTAN48	28	19	21	13	15	6
KATAN64/KTANTAN64	38	25	33	21	14	9

Key Schedule – KTANTAN

- Main problem — related-key and slide attacks.
- Solution A — two round functions, prevents slide attacks.
- Solution B — divide the key into 5 words of 16 bits, pick bits in a nonlinear manner.
- Specifically, let $K = w_4 || w_3 || w_2 || w_1 || w_0$, $T = T_7 \ldots T_0$ be the round-counter LFSR, set:

$$a_i = MUX16to1(w_i, T_7 T_6 T_5 T_4)$$

$$k_a = \overline{T_3} \cdot \overline{T_2} \cdot (a_0) \oplus (T_3 \vee T_2) \cdot MUX4to1(a_4 a_3 a_2 a_1, T_1 T_0),$$

$$k_b = \overline{T_3} \cdot T_2 \cdot (a_4) \oplus (T_3 \vee \overline{T_2}) \cdot MUX4to1(a_3 a_2 a_1 a_0, \overline{T_1 T_0})$$

Key Schedule – KATAN

$$x^{80} + x^{61} + x^{50} + x^{13} + 1$$

In other words, let the key be K, then the subkey of round i is $k_a || k_b = k_{2 \cdot i} || k_{2 \cdot i+1}$ where

$$k_i = \begin{cases} K_i & \text{for } i = 0 \ldots 79 \\ k_{i-80} \oplus k_{i-61} \oplus k_{i-50} \oplus k_{i-13} & \text{Otherwise} \end{cases}$$

Security Targets

- Differential cryptanalysis — no differential characteristics with probability 2^{-n} for 127 rounds.
- Linear cryptanalysis — no approximation with bias $2^{-n/2}$ for 127 rounds.
- No related-key/slide attacks.
- No related-key differentials (probability at most 2^{-n} for the entire cipher).
- No algebraic-based attacks.

Security – Differential Cryptanalysis

- Computer-aided search for the various round combinations and all block sizes.
- KATAN32: Best 42-round char. has prob. at most 2^{-11}.
- KATAN48: Best 43-round char. has prob. at most 2^{-18}.
- KATAN64: Best 37-round char. has prob. at most 2^{-20}.
- This also proves that all the differential-based attacks fail (boomerang, rectangle).

Security – Linear Cryptanalysis

- Computer-aided search for the various round combinations and all block sizes.
- KATAN32: Best 42-round approx. has prob. at most 2^{-6}.
- KATAN48: Best 43-round char. has prob. at most 2^{-10}.
- KATAN64: Best 37-round char. has prob. at most 2^{-11}.
- This also proves that differential-linear attacks fail.

Security – Slide/Related-Key Attacks

- Usually these are prevented using constants.
- In the case of KATAN/KTANTAN — solved by the irregular function use.
- In KATAN — the key "changes" (no slide).
- In KTANTAN — order of subkey bits not linear.

Security – Related Key Differentials (1)

- No good methodology for that.
- In KATAN32 — each key bit difference must enter (at least) two linear operations and two non-linear ones.
- Hence, an active bit induces probability of 2^{-2}, and cancels four other bits (or probability of 2^{-4} and 6).
- So if there are 76 key bits active — there are at least 16 quintuples, each with probability 2^{-2}.
- The key expansion is linear, so check minimal hamming weight in the code.
- Current result: lower bound: 72, upper bound: 84.

Security – Related Key Differentials (2)

- In KATAN48 — each key bit difference must enter (at least) four linear operations and four non-linear ones.
- Hence, an active bit induces probability of 2^{-4}, and cancels four other bits (or probability of 2^{-8} and 6).
- Need 61 active bits in the expanded key. We have them.
- For KATAN64 — need 56.
- Conclusion: no related-key differential in KATAN family.
- KTANTAN family: still checking computer simulations.

What does KATAN/KTANTAN mean?

Katan – קטן – Small

Ktantan – קטנטן – Tiny

References (1)

1. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. Journal of Cryptology 7(4), 229–246 (1994)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
3. Biryukov, A., Wagner, D.: Slide Attacks. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 245–259. Springer, Heidelberg (1999)
4. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
5. Courtois, N.T., Bard, G.V., Wagner, D.: Algebraic and Slide Attacks on KeeLoq. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 97–115. Springer, Heidelberg (2008)
6. De Canniére, C., Preneel, B.: Trivium Specifications, eSTREAM submission, http://www.ecrypt.eu.org/stream/triviump3.html
7. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials, IACR ePrint report 2008/385, accepted to EUROCRYPT 2009 (2009)
8. Feldhofer, M., Wolfkerstorfer, J., Rijmen, V.: AES implementation on a grain of sand. In: IEE Proceedings of Information Security, vol. 152(1), pp. 13–20. IEE (2005)
9. Good, T., Benaissa, M.: Hardware results for selected stream cipher candidates. In: Preproceedings of SASC 2007, pp. 191–204 (2007)

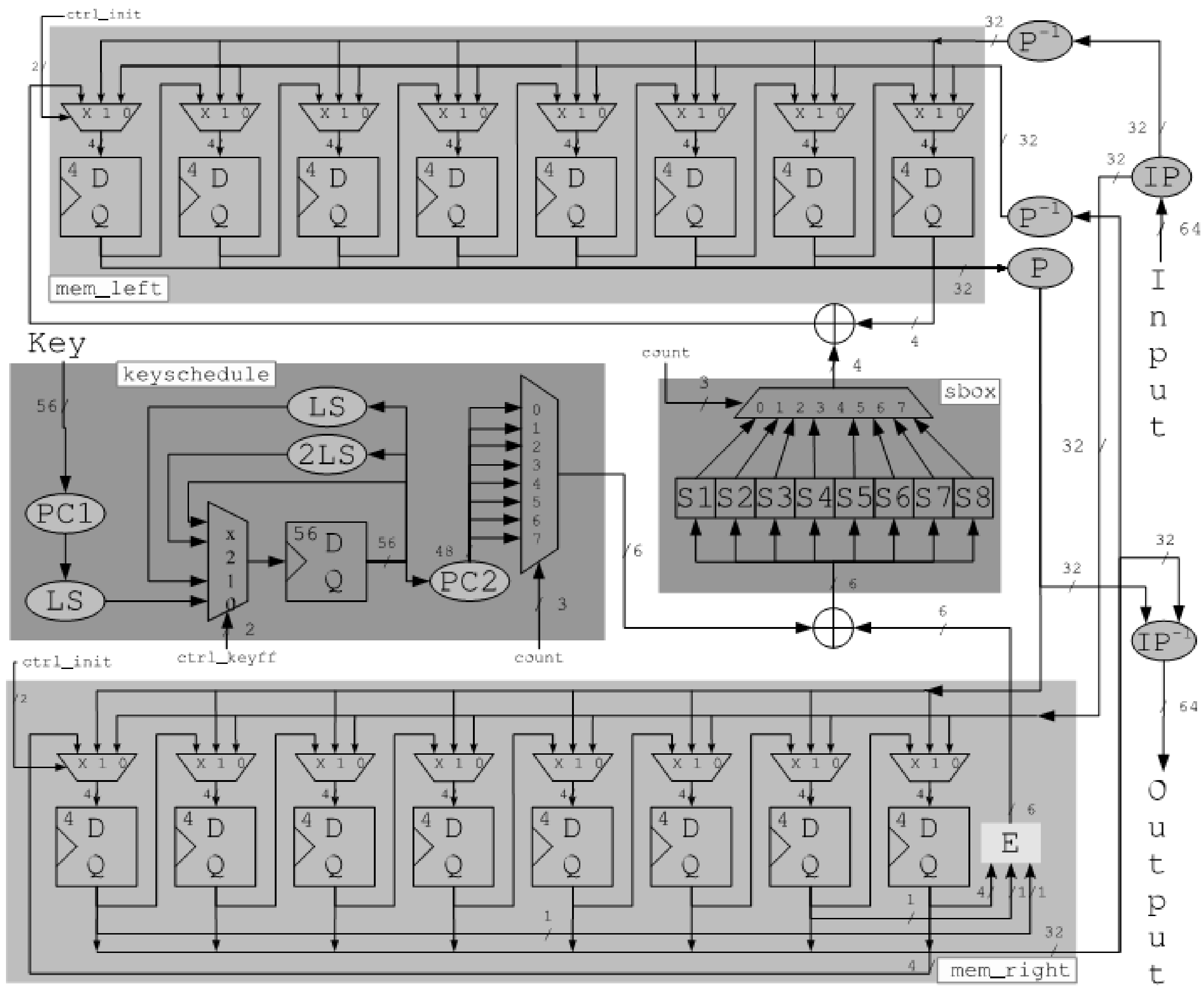
References (2)

10. Hämäläinen, P., Alho, T., Hännikäinen, M., Hämäläinen, T.D.: Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core. In: Ninth Euromicro Conference on Digital System Design: Architectures. IEEE Computer Society, Los Alamitos (2006)
11. Hell, M., Johansson, T., Meier, W.: Grain — A Stream Cipher for Constrained Environments, eSTREAM submission, http://www.ecrypt.eu.org/stream/p3ciphers/grain/Grain_p3.pdf
12. Hong, D., Sung, J., Hong, S.H., Lim, J.-I., Lee, S.-J., Koo, B.-S., Lee, C.-H., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J.-S., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
13. Indesteege, S., Keller, N., Dunkelman, O., Biham, E., Preneel, B.: A Practical Attack on KeeLoq. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 1–18. Springer, Heidelberg (2008)
14. Langford, S.K., Hellman, M.E.: Differential-Linear Cryptanalysis. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer, Heidelberg (1994)
15. Lim, C.H., Korkishko, T.: mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
16. Mentens, N., Genoe, J., Preneel, B., Verbauwhede, I.: A low-cost implementation of Trivium. In: Preproceedings of SASC 2008, pp. 197–204 (2008)
17. Microchip Technology Inc. KeeLoq® Authentication Products, http://www.microchip.com/keeloq/
18. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)

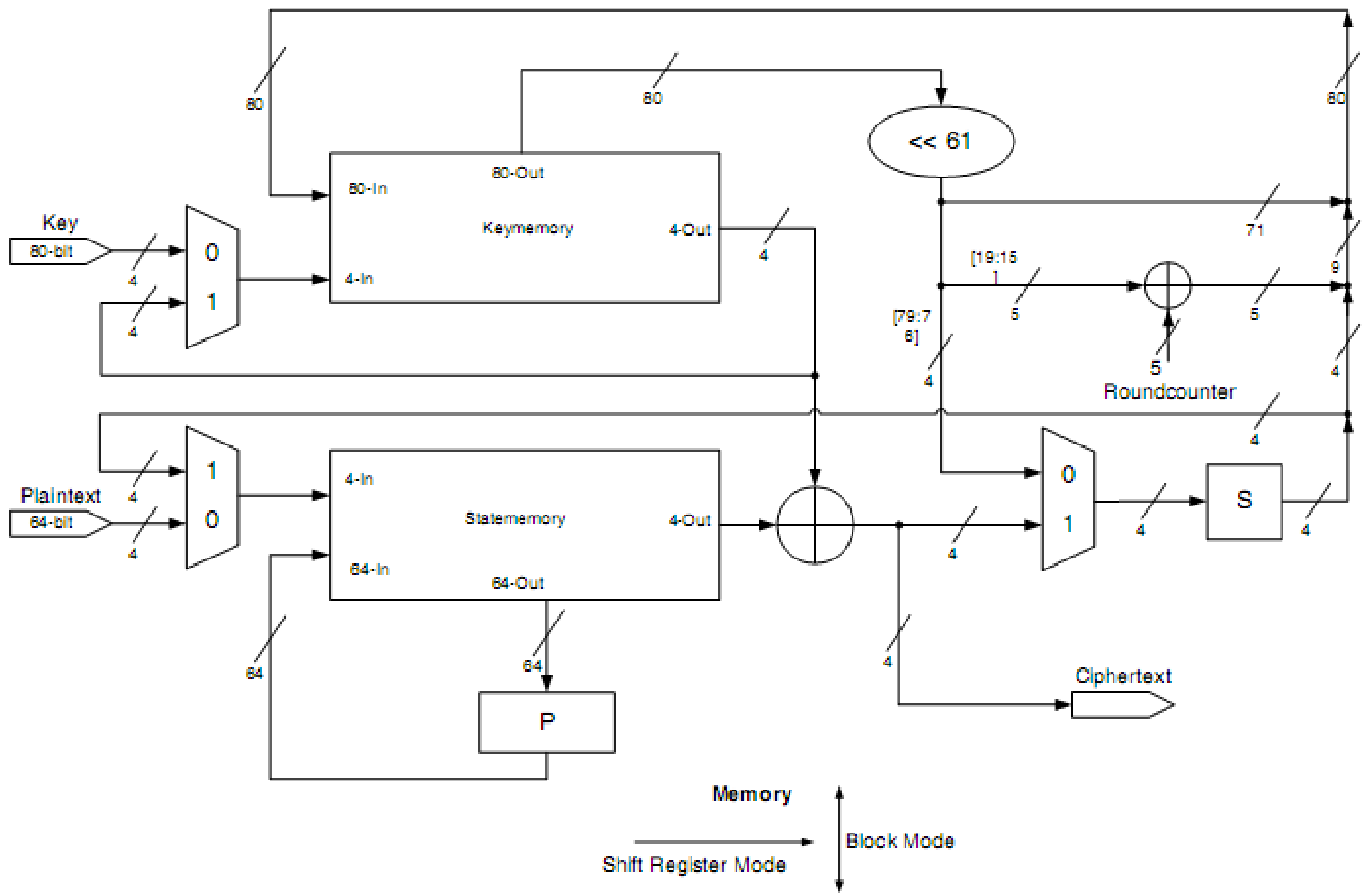
References (3)

19. Poschmann, A., Leander, G., Schramm, K., Paar, C.: New Light-Weight DES Variants Suited for RFID Applications. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
20. Rolfes, C., Poschmann, A., Leander, G., Paar, C.: Ultra-lightweight implementations for smart devices – security for 1000 gate equivalents. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 89–103. Springer, Heidelberg (2008)

DESL[19]

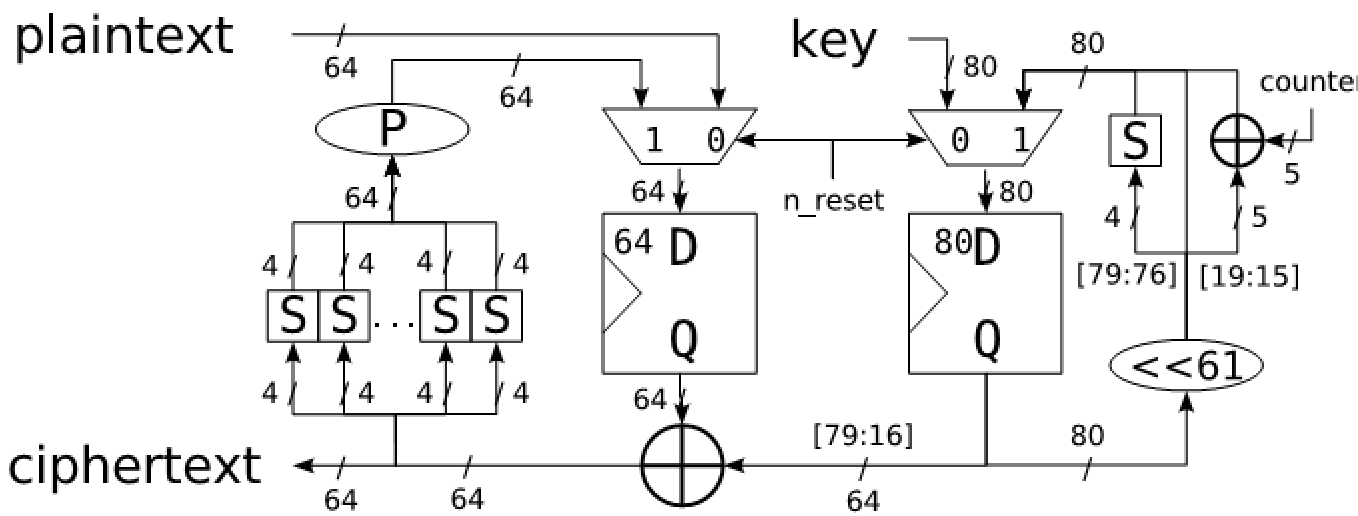

ctrl_init
mem_left
Key
keyschedule
PC1
LS
2LS
PC2
count
ctrl_keyff
sbox
S1 S2 S3 S4 S5 S6 S7 S8
P
IP
Input
mem_right
E
Output

PRESENT[20]



80
80
80
<< 61
80-Out
80-In
Keymemory
4-Out
4-In
Key
80-bit
4
4
0
1
4
71
[19:15]
5
5
9
[79:76]
4
5
Roundcounter
4
4
Plaintext
64-bit
4
4
1
0
4-In
Statememory
4-Out
64-In
64-Out
0
1
4
4
S
4
64
64
P
4
Ciphertext
Memory
Block Mode
Shift Register Mode

PRESENT[4]



plaintext
64
64
P
64
4
4
4
4
S
S
S
S
4
4
4
4
ciphertext
64
64
1 0
64
64 D
Q
64
n_reset
key
80
0 1
80
80D
Q
[79:16]
64
80
S
4
[79:76]
counter
5
5
[19:15]
<<61
80