# Specific Countermeasures Against Physical Attacks in FPGAs

Jean-Luc DANGER Shivam BHASIN Guillaume DUC Tarik GRABA Sylvain GUILLEY Houssem MAGHREBI Olivier MEYNARD Maxime NASSAR Laurent SAUVAGE Nidhal SELMANE Youssef SOUISSI

Institut TELECOM / TELECOM-ParisTech/ CNRS – LTCI (UMR 5141)

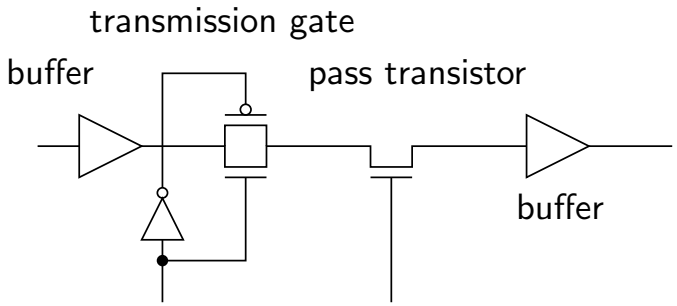
Thursday, December 7th, 2010,
Journée SocSip Securité

# Presentation Outline

# Presentation Outline

# FPGA specificity

- Price to pay for reconfigurability:
  - Size 35X $\Rightarrow$ 18X , Consumption 14X ASIC size (Kuon and all 2007)
- Many high-gain DFFs
- Many memories:
  - distributed: LUTs
  - embedded
- Many DSPs
- Many long lines and switches : Interconnect = 80% of the total area, and **unknown**



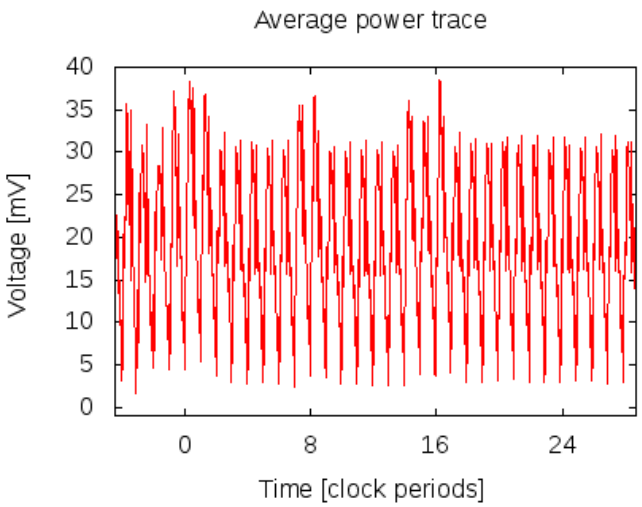
# Vulnerability against side-channel attacks

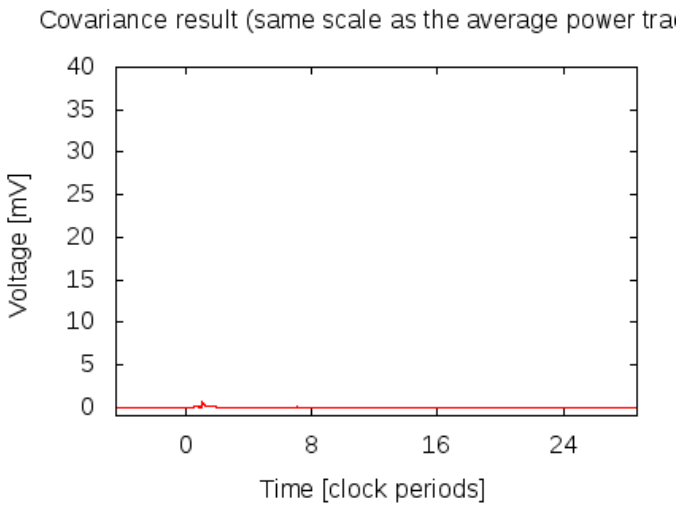
Comparison between ASIC and FPGA in terms of power leakage:

1. **SecMat v3[ASIC]:**
   - Shared power supply between all modules
2. **SecMat v3[FPGA]:**
   - SecMat v3[ASIC] VHDL code synthesized in an Altera Stratix EPS1S25
   - Global power supply
   - 10,157 logic elements and 286,720 RAM bits for the whole SoC
   - DES alone is 1,125 logic elements (LuT4)

The power traces acquired from those three circuits are available for download from http://www.dpacontest.org/.

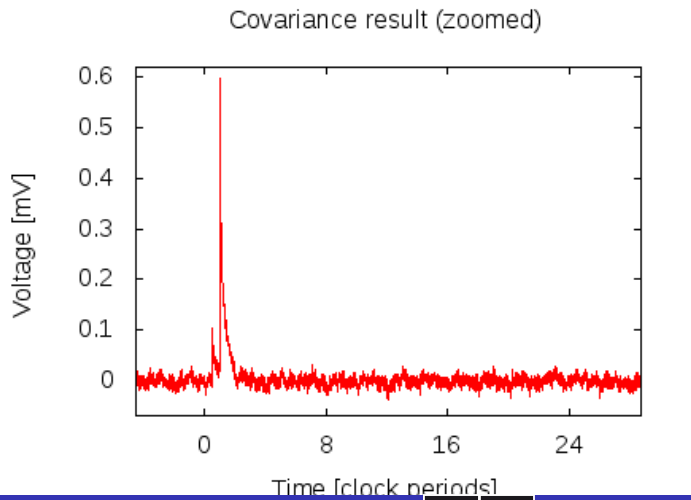
# SecMat v3[ASIC] – covariance with $|LR[0] \oplus LR[1]|$



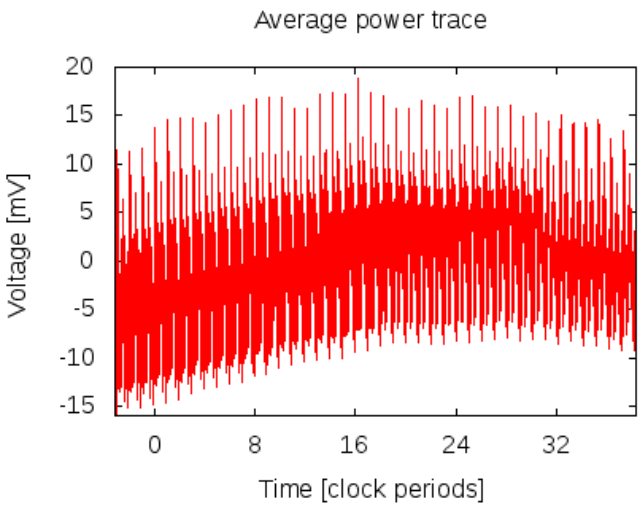



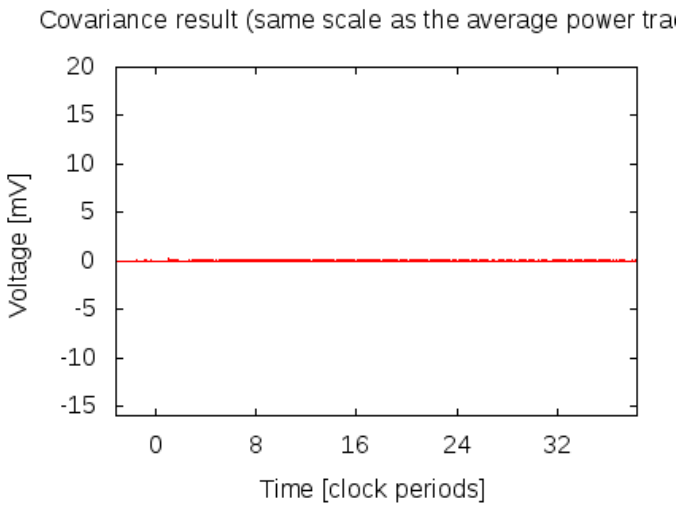

**SecMat v3[ASIC]**:

- Typical trace: 38 mV
- Typical DPA: 0.6 mV
- $\Rightarrow$ Side-channel leakage: 1.5 %



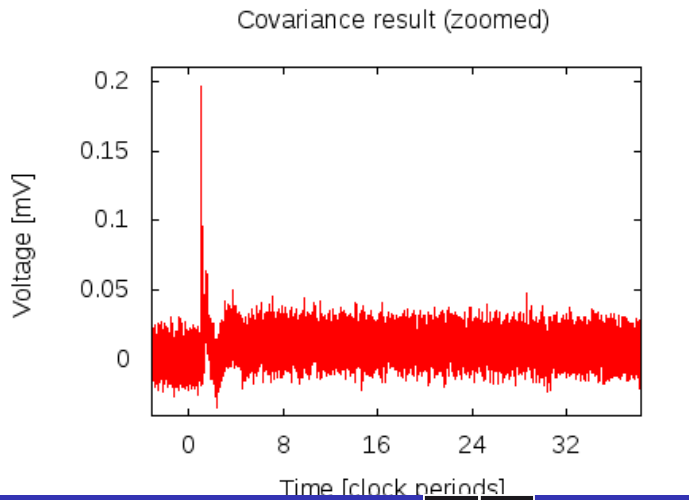
# SecMat v3[FPGA] – covariance with $|LR[0] \oplus LR[1]|$




**SecMat v3[FPGA]**:

- Typical trace: 19 mV
- Typical DPA: 0.19 mV
- $\Rightarrow$ Side-channel leakage: 1.0 %

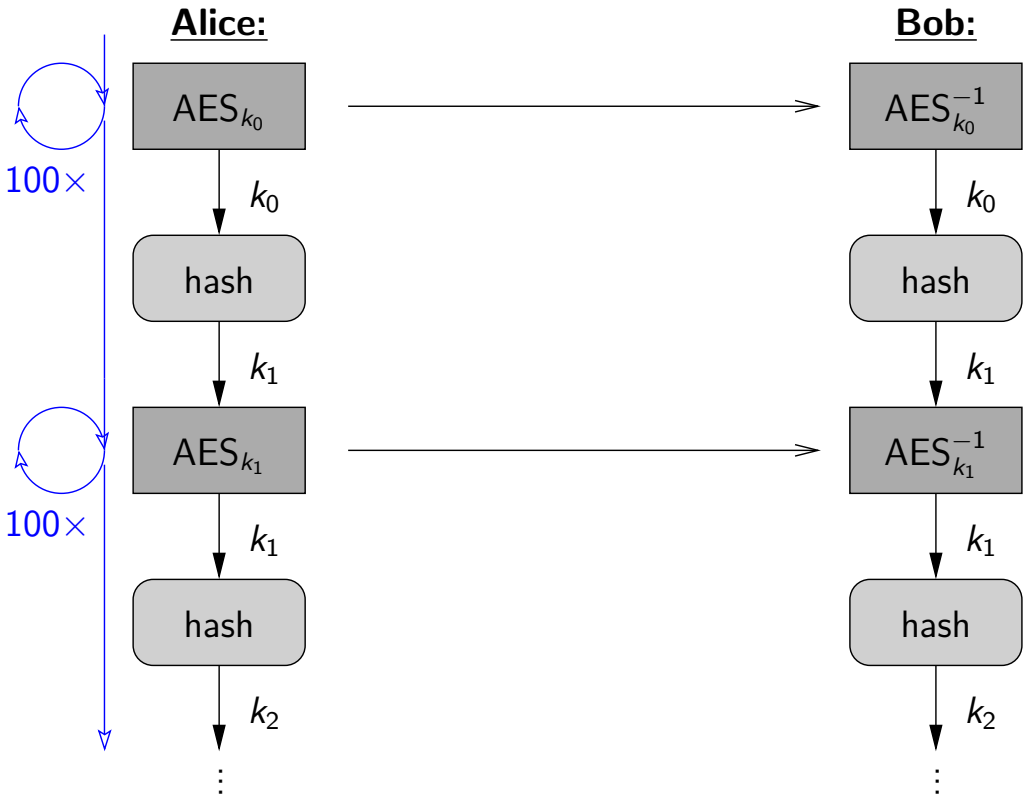


# Presentation Outline

# Targeted strategies

- Protocol-level:
  - Most wanted since provable
- Register-Transfer Level:
  - **Masking**, boolean or algorithmic.
  - Encrypted leakage
  - Glitch-full circuits
- Netlist or implementation level:
  - **Hiding**= DPL, Dual-rail with Precharge Logic
- Degenerated counter-measures
  - Noise generator, Dummy instructions, Varying clock, *etc.*

if $\approx 1$ bit is leaked per 100 encryptions...




The FPGAs designs can take advantage of Reconfigurability to change regularly the implementation.

# Masking

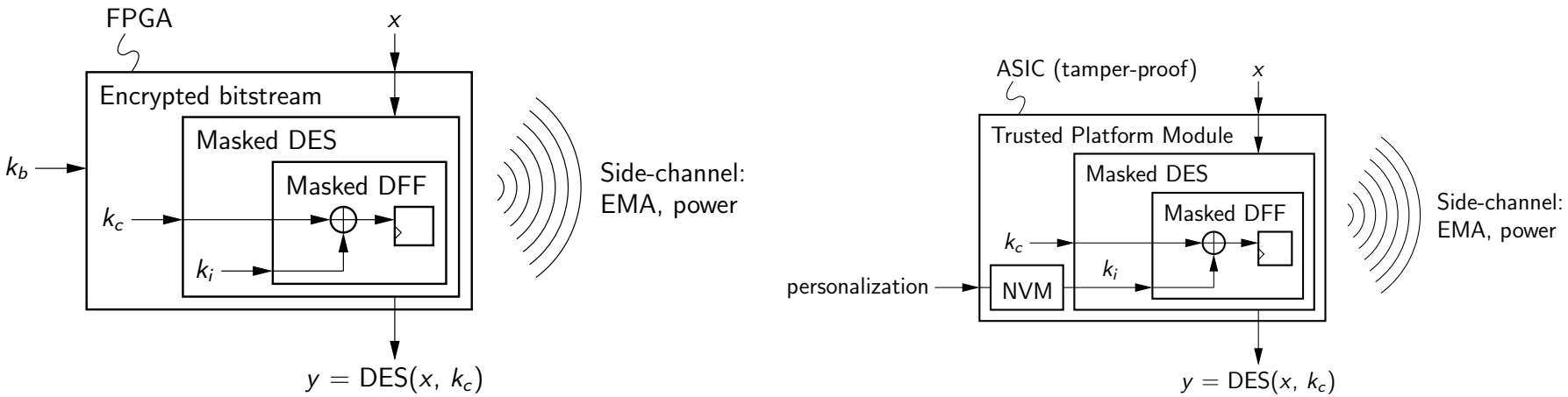
## Principle

- Every variable $s$, potentially sensible, is represented as a share $\{s_0, s_1, \cdots, s_{n-1}\}$
- To reconstruct $s$, all the $s_i$ are required.
- Example: $n = 2$, $s \doteq s_0 \oplus s_1$.

## Constraints and Drawbacks

- Leakage resistant since variables are never used plain.
- Attractive but works only fine for registers.
- Efforts done to protect also the combinational logic.
- Sensitive to Hi-orders attacks.
- Ineffective against Fault attacks.

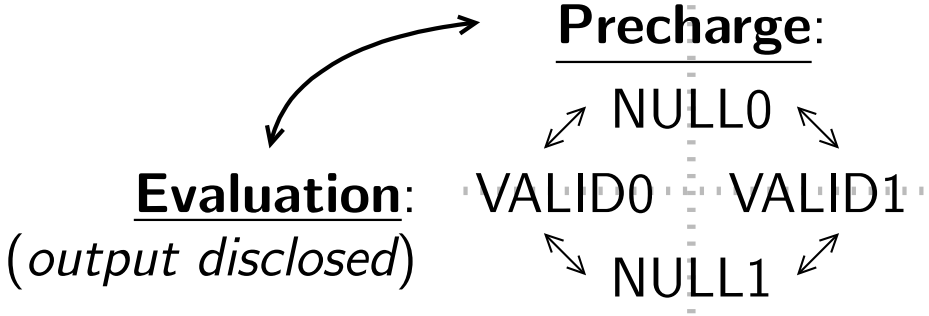
# Encrypted Leakage



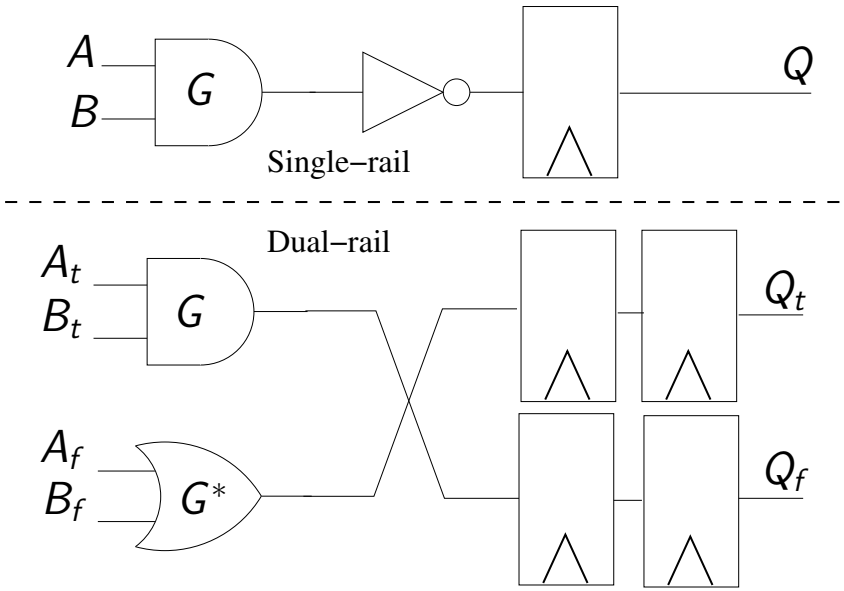
# Hiding by using DPL: Dual Rail with Precharge Logic

## $a \leftrightarrow (a_f, a_t)$ DPL representation:

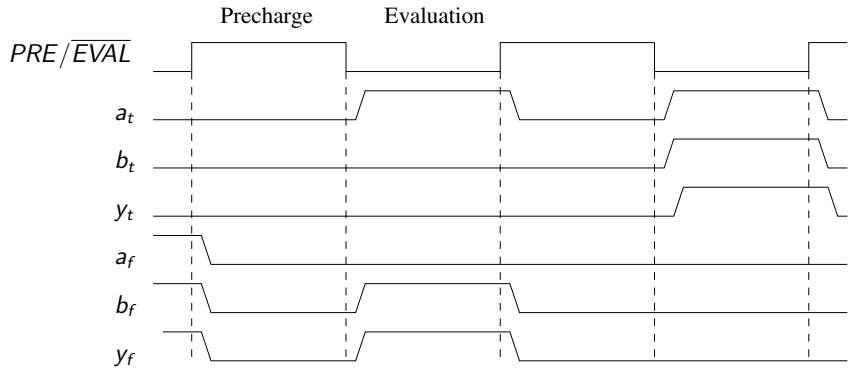
- $a$ is **VALID** if $\boxed{a_f \oplus a_t = 1}$. VALID $\doteq \{\text{VALID0}, \text{VALID1}\}$ or VALID $\doteq \{(1,0),(0,1)\}$.
- $a$ is **NULL** if $\boxed{a_f \oplus a_t = 0}$. NULL $\doteq \{\text{NULL0}, \text{NULL1}\}$ or NULL $\doteq \{(0,0),(1,1)\}$.



# A common DPL: WDDL=Waveform Dynamic Differential Logic




A digital circuit and its WDDL equivalent

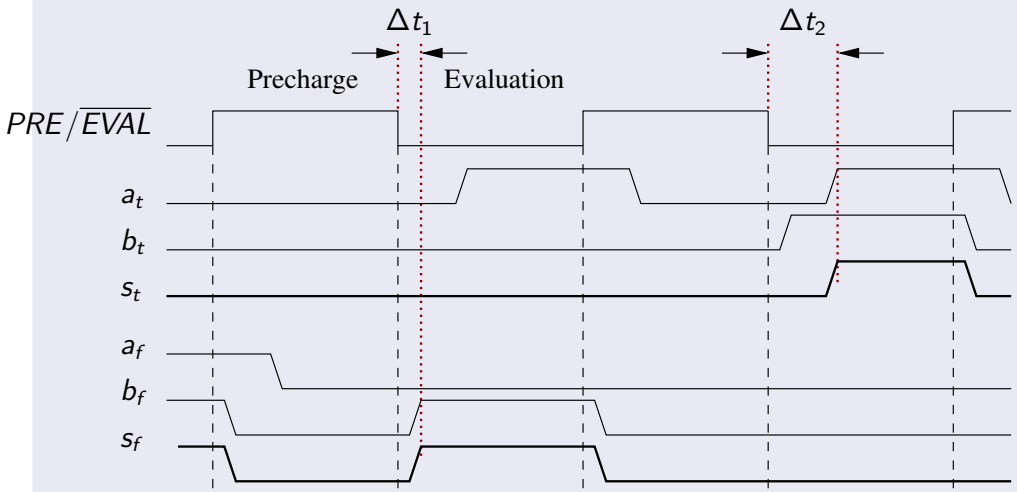


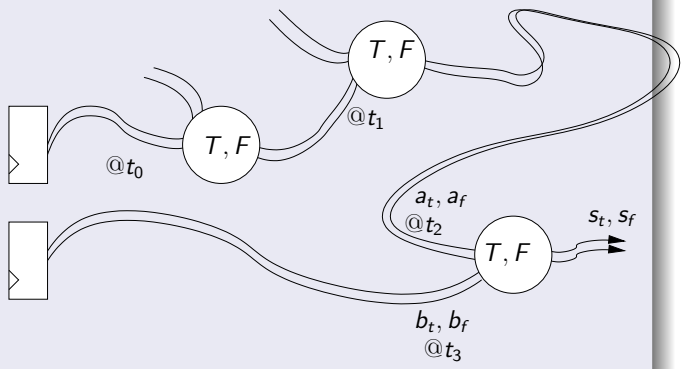

Timing Diagram of a WDDL AND gate

Only positive gates could be used for netlist synthesis.

# Important constraints in DPL : No glitches +

## No Early Evaluation




Cause of Early Evaluation

## No Technological Biais

- OR consumption = AND consumption
- routing T = routing F

# Presentation Outline

# Security constraints 1/2

## Logic without glitches and early propagation

⇒ **Synchronization**

The rules to be "synchronized":

- **Rule 1:** Evaluation starts after all the input signals are valid.
- **Rule 2:** Precharge starts:
  1. Either after all the inputs becomes NULL[1] but the outputs need to be memorized or
  2. Or before the first input becomes NULL (which does not need any memorization).

[1] NULL is the value in precharge phase

# Security constraint 2/2

## Logic with a minimum of technological biais

- Special care at placing and routing (but the FPGA vendors give few informations)
- Use of the same logic structure for True and False (e.g. MDPL with majority gates)
- Statistical balancing

## Logic resistant to fault attacks

- Detection capability or
- Resilience

# Cost and Speed constraints

## Logic with a minimum cost

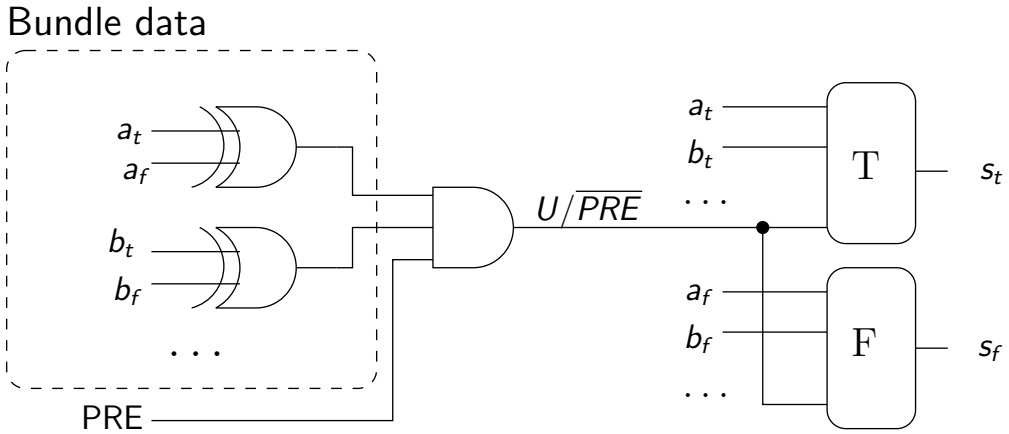
- A few more than X2
- Use of RAMs and DSP in FPGAs

## Fast speed

- speed divided by 2. Possible to be better?

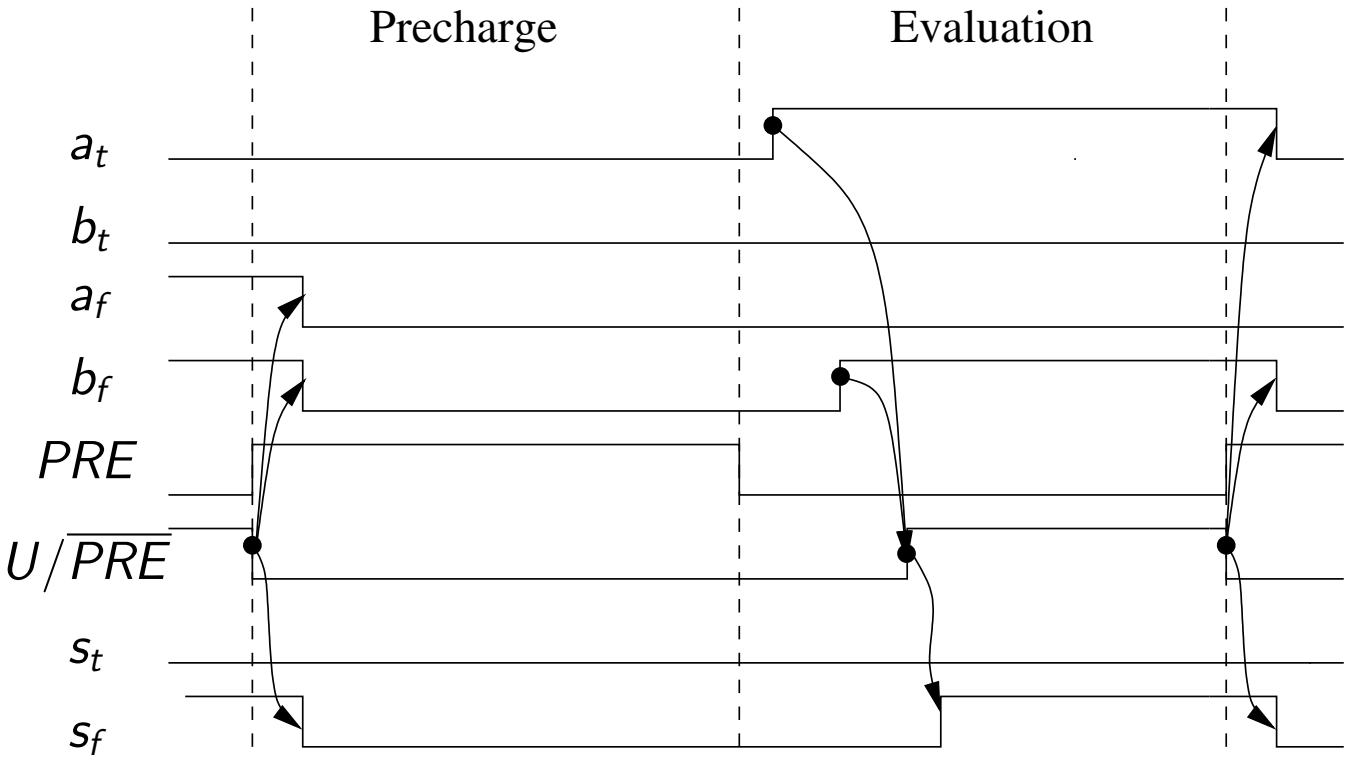
# Case study of BCDL: Balance Cell Differential Logic

## The BCDL gate: Synchronization with Global Precharge




- No need of memorization as a **global precharge** *PRE* is faster than any inputs.
- $U/\overline{PRE}$ falls to 0 $\Rightarrow$ precharge is forced immediately.
- $U/\overline{PRE}$ rises to 1 $\Rightarrow$ evaluation begins after "unanimity to 1".
- Tables T and F can be fully separated $\Rightarrow$ huge complexity gain.

# Exemple of a 2-input OR gate

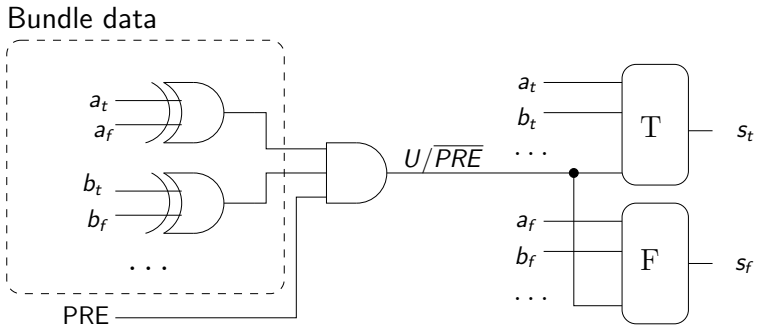


# Robustness against FA

## In-Built Robustness against Fault Attacks

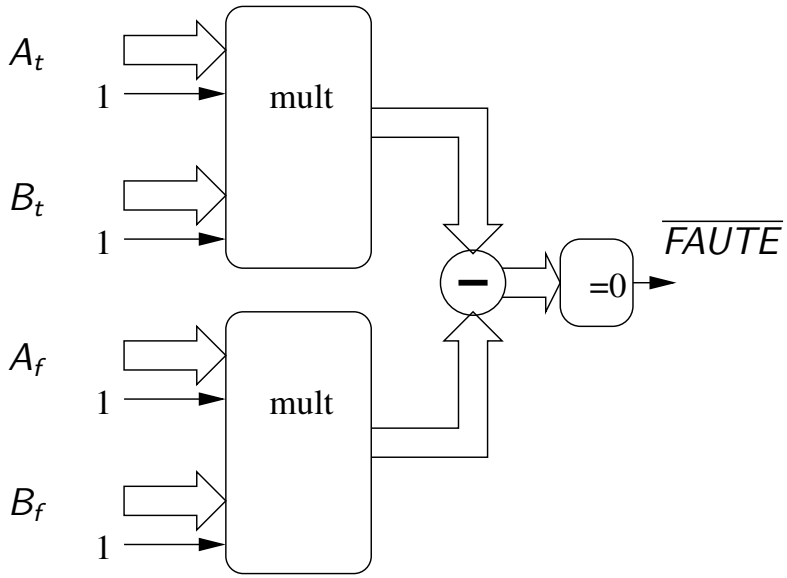
- Automatically detects symmetric faults: {VALID0, VALID1} $\overset{\downarrow \text{ or } \uparrow}{\longrightarrow}$ {NULL0, NULL1}($1 \rightarrow 0$ or $0 \rightarrow 1$).
- "Error state" is propagated throughout the design $\Rightarrow$ **Fault resilience**.

| *PRECHARGE* | Fault detection |
|---|---|
| 1 | state $\neq$ {NULL0, NULL1} |
| 0 | state$\neq$ {VALID0, VALID1} |



# Fault Detection with DSP blocks

- based on $AxB = (-A)x(-B) \Rightarrow (2A+1)x(2B+1) = (2\overline{A}+1)x(2\overline{B}+1)$
- Allows to detect and locate either during precharge or evaluation



# Area

## T and F easy to implement

- Not limited to positive functions
- **separable**
  - 1 additionnal input ($U/\overline{PRE}$) + duplication($T$ and $F$)
  - Area of tables $= 2.2^{n+1} < 2^{2n}$ if $n > 2$
  - $\Rightarrow$ S-Box area = only **4 times** the size of an unprotected one.

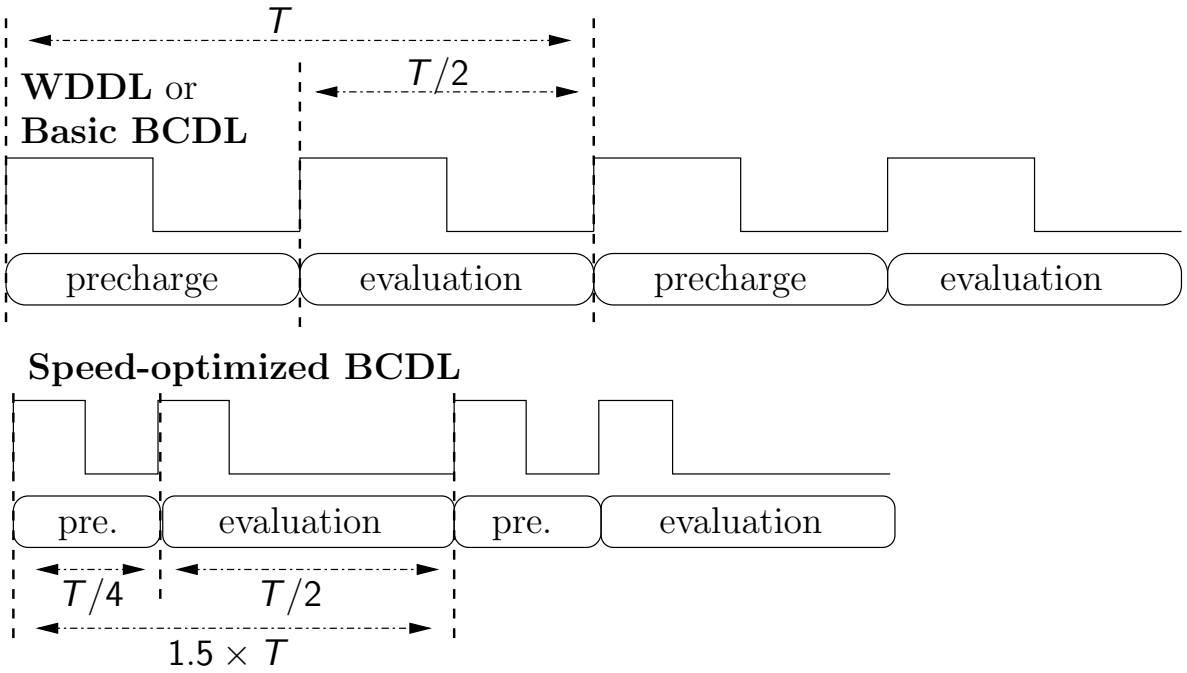
## Total Area

$=$ DFF($*4$) + [SYNC(*a few gates*) + T + F ] $* n$.

## Special case: MUX driven by single rail signal

- No needs of synchronization.

# Speed optimization




## Faster than other DPLs

- Evaluation time $>$ precharge time $\Rightarrow$ performances $\nearrow$
- Speed / $\sim 1.25 \leftrightarrow 1.75$

# results in FPGA Stratix for an AES implementation

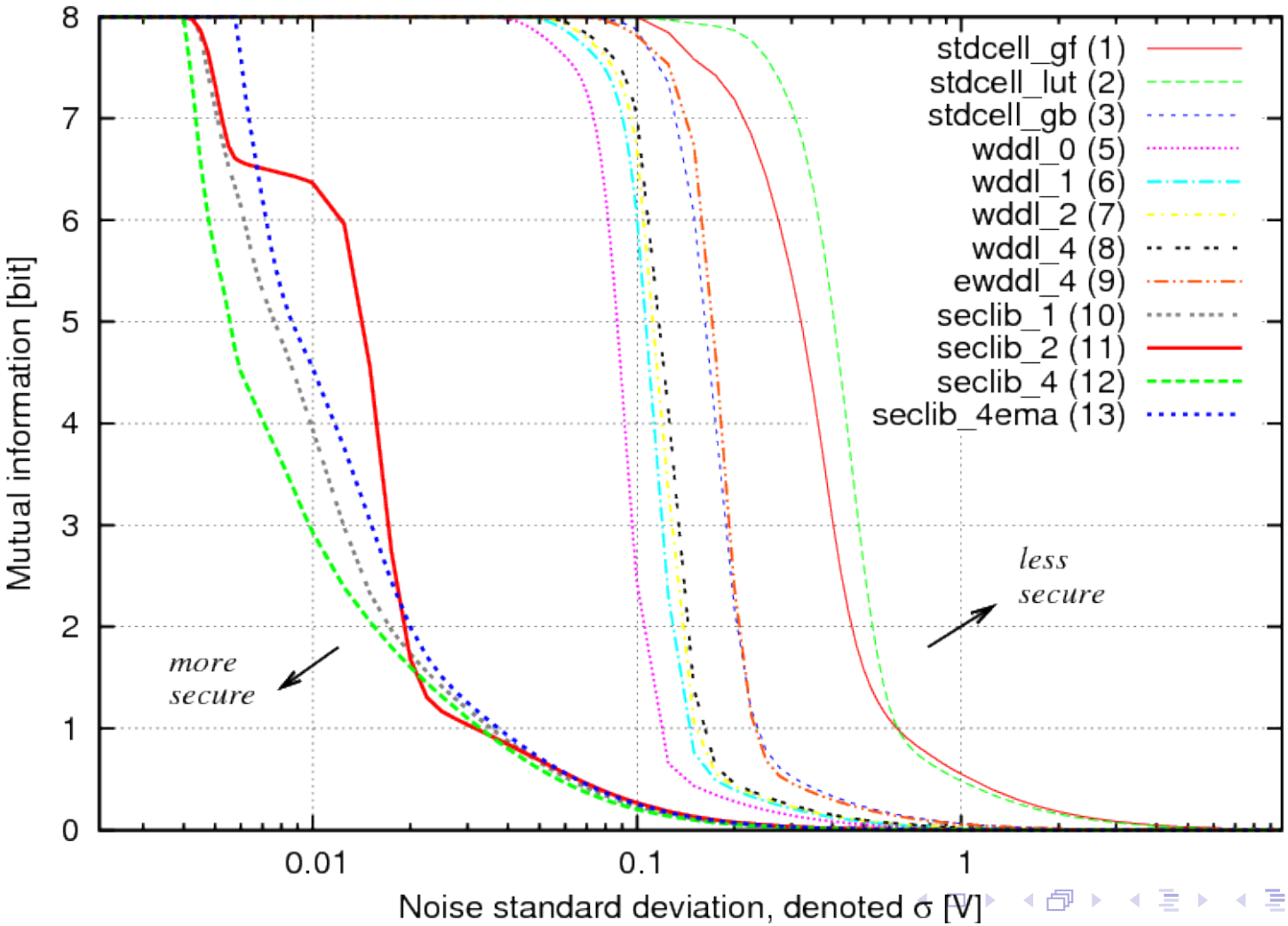
## Complexity and speed

| | ALM | Reg | RAM | Max. freq. | Max. throughput |
|---|---|---|---|---|---|
| **no protection** | 1078 | 256 | 40 Kb | 71.88 MHz | 287.52 Mbps |
| **WDDL** | 4885 | 1024 | — | 37.07 MHz | 74.14 Mbps |
| **BCDL** | 1841 | 1024 | 160 Kb | 50.64 MHz | 151.92 Mbps |

## CPA results

- Attack processed on 150000 power consumption traces.
- No subkey found for BCDL.

# MIA results for different subbytes implementations



# Comparison with other DPLs in FPGAs

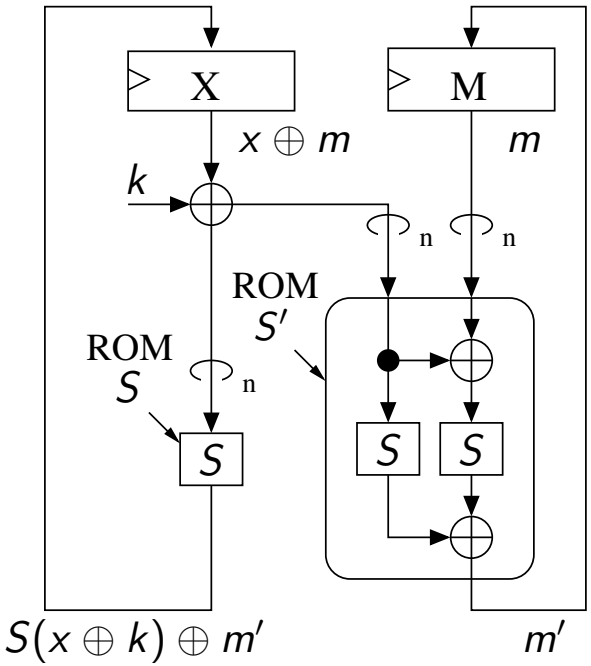
- **WDDL** : Propagation of the NULL state with positive functions
- **RCDDL** : WDDL with factored logic, which amplifies the early evaluation
- **MDPL** : T gate =F gate = Majority, random Mask to balance the True and False networks
- **STTL** : A third wire is added to synchronize with the last stable signal.
- **DRSL** : As MDPL with a synchronization before evaluation
- **IWDDL** : Isolated WDDL with separated T and F networks by means of superpipelining
- **BCDL** : The logic presented here
- **MBCDL** : BCDL with mask

# Comparison with other DPLs

| Logic | Compl. | Speed | Robust. SCA | | Robust. FA | | Design Constr. |
|---|---|---|---|---|---|---|---|
| | | | EE | T. B. | Fault | Det. | |
| **WDDL** | * | $< 1/2$ | | | asym | comb | Positive gates |
| **MDPL** | * | $< 1/2$ | | ✓ | asym | comb | MAJ gate + RNG |
| **STTL** | * | $< 1/4$ | ✓ | | sym | seq | 50% more wiring |
| **DRSL** | * | $< 1/2$ | partly | ✓ | sym | comb | + RNG |
| **IWDDL** | | $< 1/2 \cdot n$ | ✓ | | asym | comb | superpipeline |
| **BCDL** | ** | $> 1/2$ | ✓ | | sym | comb | |
| **MBCDL** | * | $> 1/2$ | ✓ | ✓ | sym | comb | + RNG |

# Presentation Outline

# ROM Hardware masking




*Masked DES implemented with ROMs.*

"Zero Offset" From Waddle *et al.*, Peeters *et al.*.

- Activity:

$$A = HW[(x \oplus m) \oplus (S(x \oplus k) \oplus m')] + HW[m \oplus m']$$

- The register data Hamming distance is:

$$\Delta(x) = x \oplus S(x \oplus k)$$

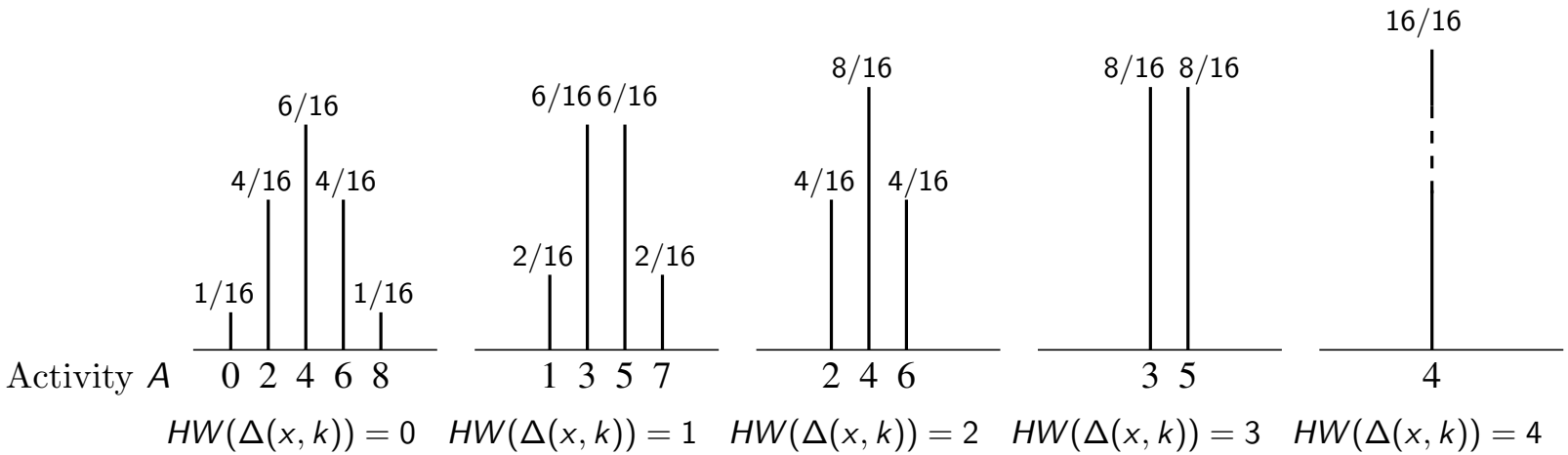
- The register mask Hamming distance is:

$$\Delta(m) = m \oplus m'$$

- Then:

$$A = HW[\Delta(x) \oplus \Delta(m)] + HW[\Delta(m)]$$

# Problem # 1: HO-attacks




*Power distributions of the five possible values of* $HW(\Delta(x,k))$.

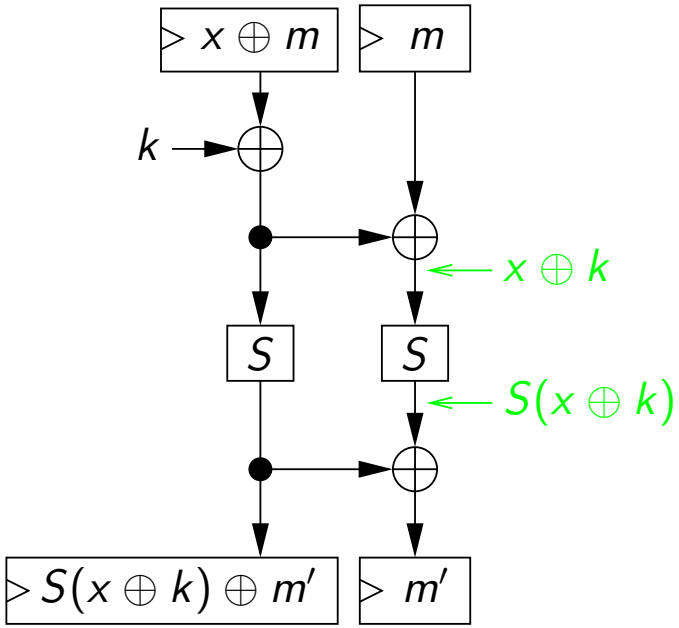
## Theoretic MIA attack evaluation

*Table: Theoretical conditional entropy of the ROM masked DES.*

| Theoretical entropies | The correct key | Any wrong key |
|---|---|---|
| $H(O\|HW(\Delta(x,k)))$ | **1.3992 bit** | **2.5442 bit** |

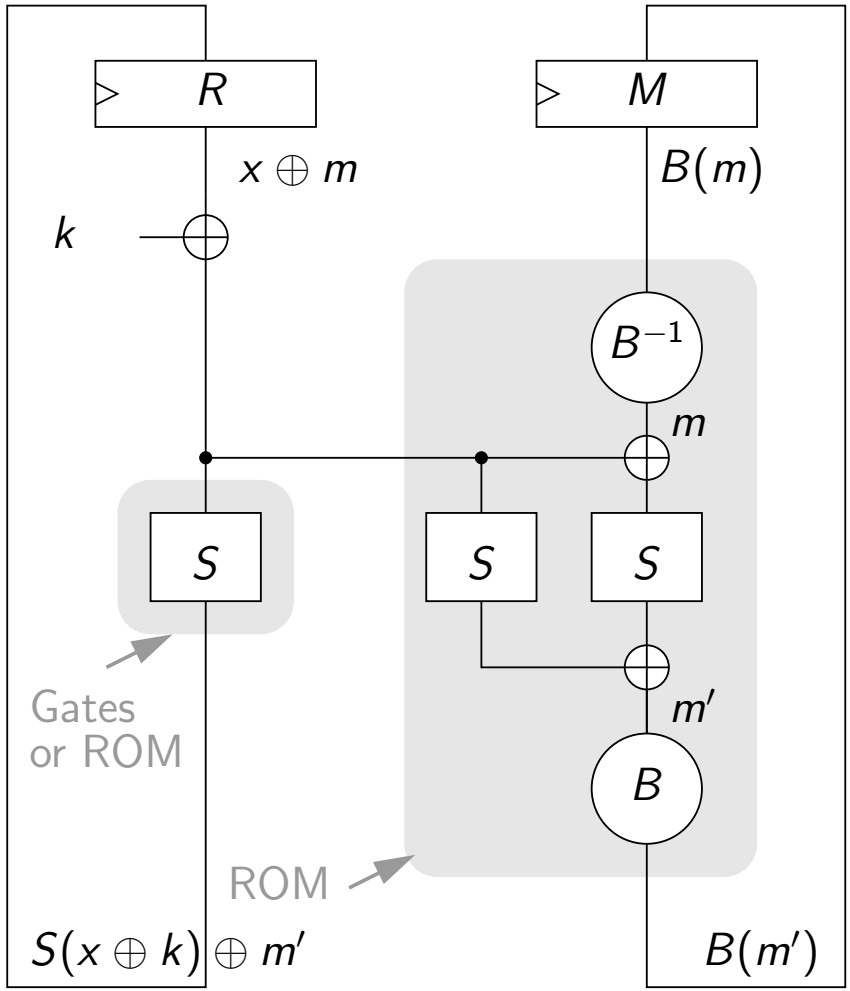
# Problem # 2: ROM too complex for FPGAs

- Need of $2^{2n}$ memory
- Use of external Mask recomposition with USM: Universal S-Box Masking

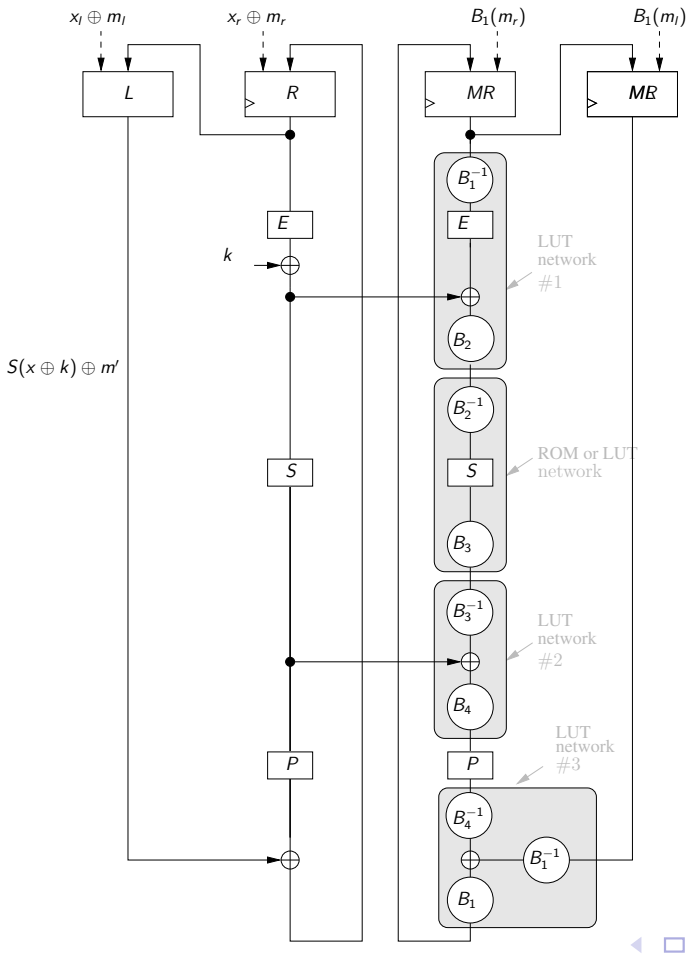



But attackable on the combinatorial logic!

# Solution #1: Squeezed leakage by encoding tables



# Solution #2: Squeezed leakage by encoding tables with USM

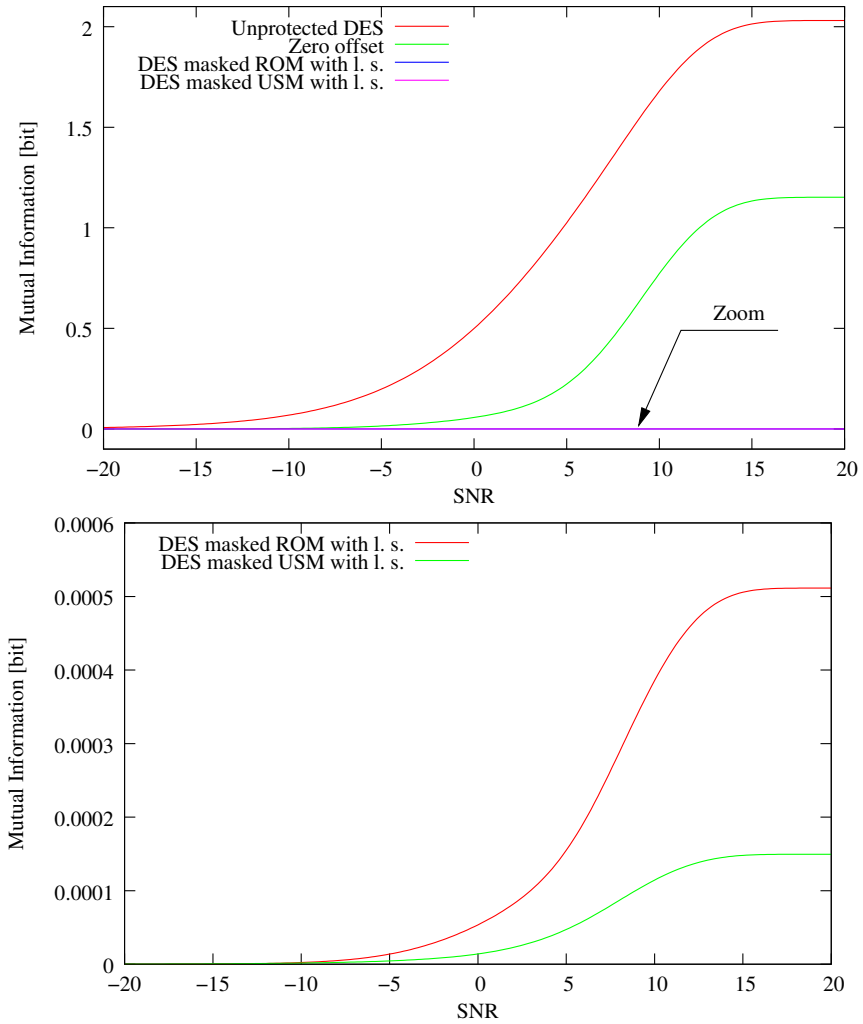


# Implementation results with leakage squeezing

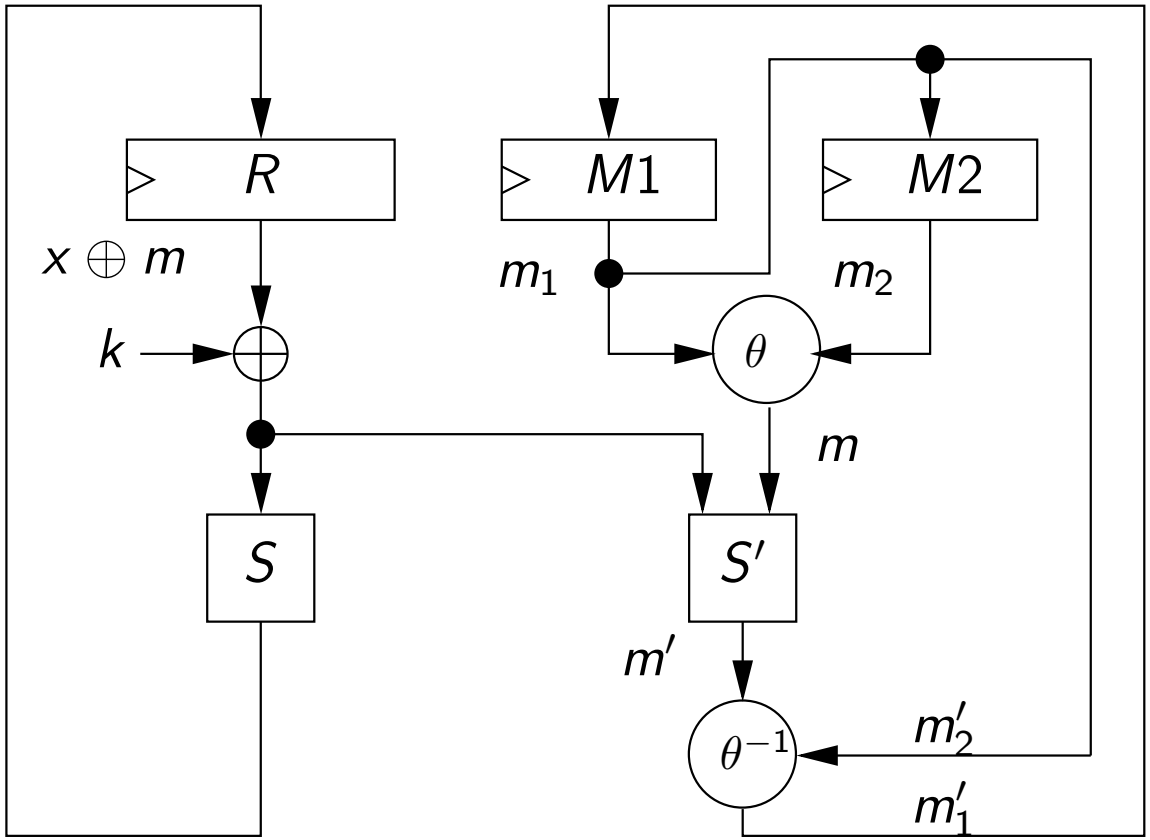
Table 1: Complexity and speed results. "l. s." denotes the "leakage squeezing" countermeasure.

| Implementation | ALMs | Block memory [bit] | M4Ks | Throughput [Mbit/s] |
|---|---|---|---|---|
| **Unprotected DES (reference)** | 276 | 0 | 0 | 929.4 |
| **DES masked USM** | 447 | 0 | 0 | 689.1 |
| **DES masked ROM** | 366 | 131072 | 32 | 398.4 |
| **DES masked ROM with l. s.** | 408 | 131072 | 32 | 320.8 |
| **DES masked USM with l. s.** | 488 | 0 | 0 | 582.8 |

# MIA results with leakage squeezing



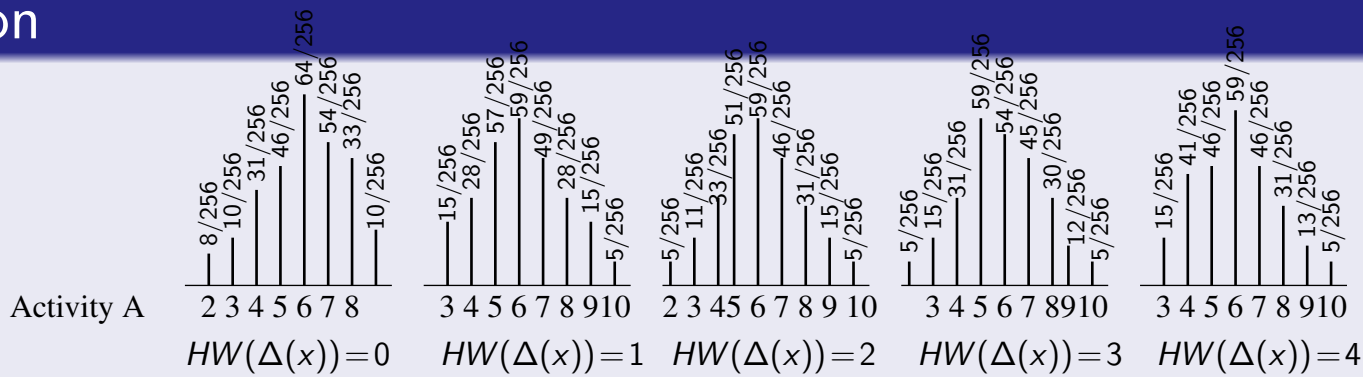
# Squeezed leakage by mask decomposition



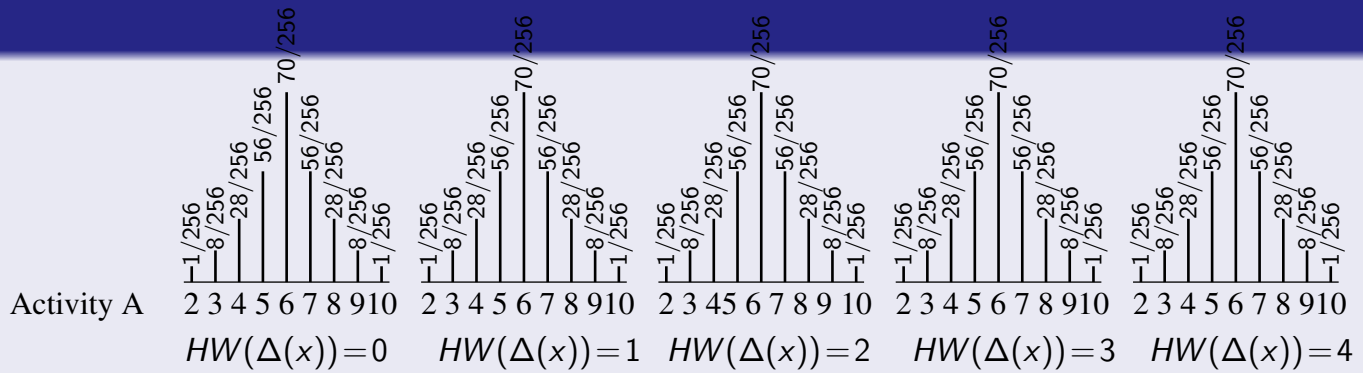

$S(x \oplus k) \oplus m'$

# Distributions obtained for different Θ

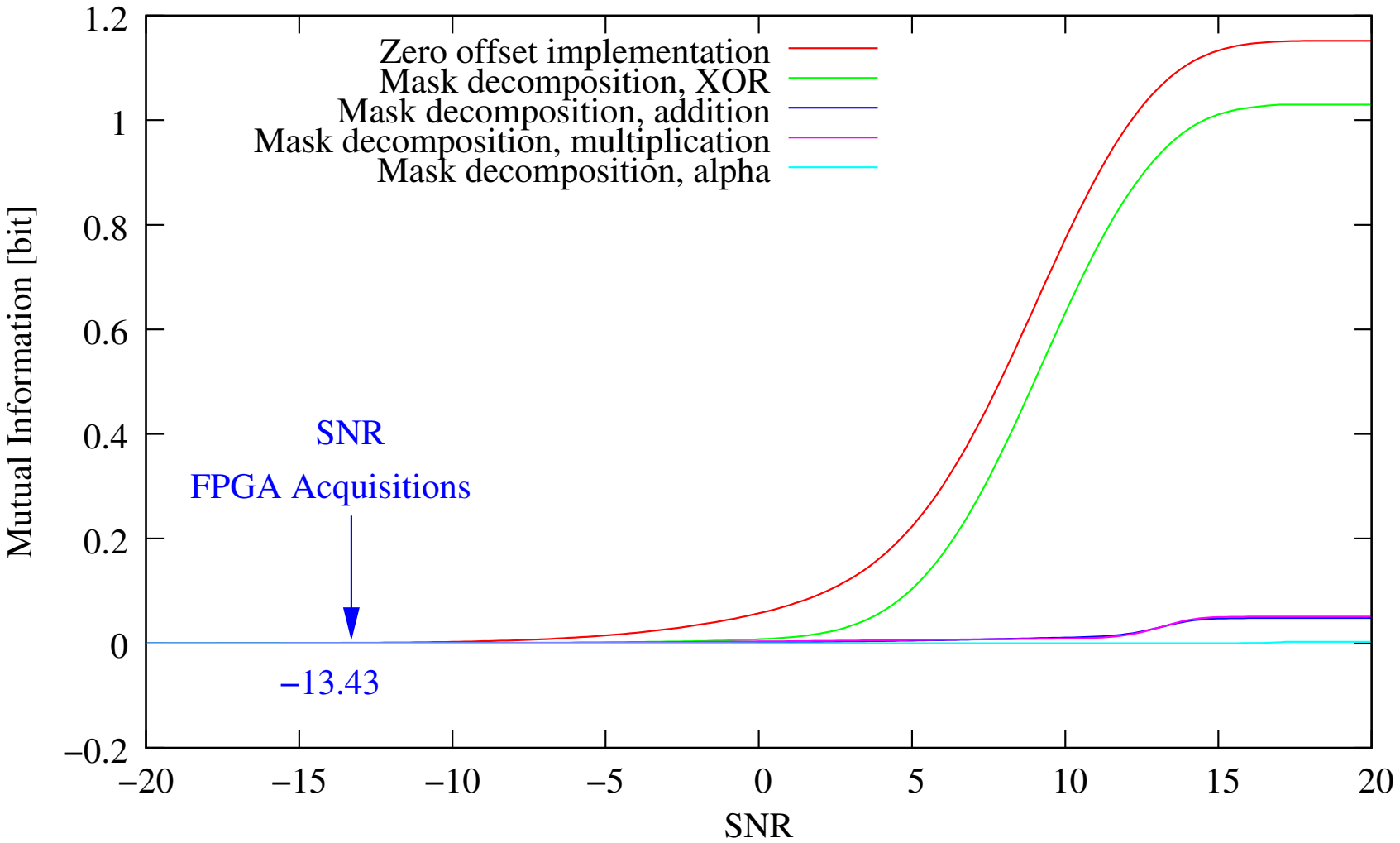
## addition




## alpha



# MIA by Squeezed leakage by mask decomposition



# Presentation Outline

- The FPGAs need efficient countermeasures to be protected against physical attacks.
- Three levels:
  - Protocol:
    - Reconfiguration can be done in FPGAs
  - RTL : Masking by taking davantages of RAMs but care has to be taken against HO-DPA. Exemples:
    - Leakage squeezing
    - Mask decomposition
  - Netlist : By using DPL. Examples:
    - STTL: no EE, need of 3rd wire, care of P/R
    - BCDL: no EE, low complexity, care of P/R
    - MBCDL: BCDL + easy P/R

Thanks for your attention.
Any question?