Implementation and Evaluation of WDDL in FPGAs

Jean-Luc DANGER Sylvain GUILLEY Ph. HOOGVORST
Laurent SAUVAGE Tarik GRABA Yves MATHIEU

Institut TELECOM / TELECOM ParisTECH
CNRS – LTCI (UMR 5141)

CryptArchi June 2008.

Presentation Outline

Historically, attacks target ASICs

- **SPA** [14],
- **DPA** [14, 10, 12],
- **IPA** [8],
- **CPA** [3, 4, 11],
- **EMA** [9, 1, 15] and
- **Template attacks** [6, 17, 2].

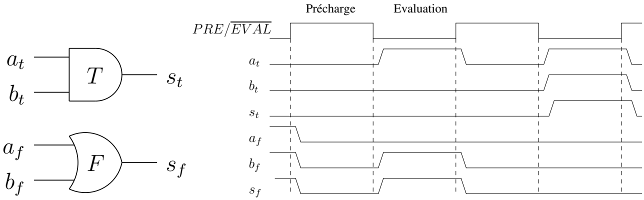
More recently, attacks on FPGAs have been reported

- **2003**: **SPA** on Xilinx Virtex 800 [13],
- **2004**: **CPA** on the same board [18],
- **2005**: **EMA** on an Altera Cyclone [5],
- **2006**: **CPA** improvements (filtering, averaging) in [19].

- **Logical WDDL** in FPGAs by Kris Tiri [21, 23, 22]:
  - [21]: suffers a large area overhead: {`INV`, `AND`, `OR`} (3 gates).
  - [23]: a clustering method allows to use all `AND-OR` combinations (166 gates in LuT4 FPGAs).
  - [22]: automation with ASIC synthesizers
- **Physical WDDL** in FPGAs by Pengyuan Yu & Patrick Schaumont [25, 26], based on copy-and-paste:
  - [25]:Separated Dynamic Dual-Rail Logic (SDDL) fails because of glitches
  - [26]: Double WDDL (DWDDL) at least quadruples the area. Moreover, [16] shows that an integrated antenna of about 40 μm extension can measure EM emanations selectively.
- Secured designs in FPGAs based on **masked logic** are reported by François-Xavier Standaert in [20].
- An excellent overview of **security issues** in FPGAs [7].
- The protection of the bitstream and of the applications in a **RTR context** [24].

DPL : Dual Rail Precharge Logic for Power balancing

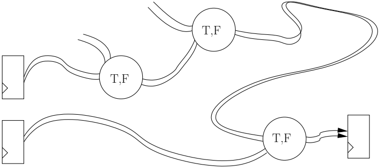
- Dual Rail : 1 variable $x \Rightarrow$ 1 couple of signals (x_t, x_f)
- Precharge Logic : at least two phases : precharge/evaluation
- Example : WDDL



- Power Consumption $= \alpha * (number \quad of \quad transitions) = 3$ when Precharge $\Rightarrow$ Evaluation and Evaluation $\Rightarrow$ Precharge

WDDL principle

- The Precharge state '0' propagates along the logic paths.



- constraints for FPGAs:
  - Two dual LUT4 T and F such that $T(0,0,0,0) = F(0,0,0,0) = 0$
  - $\Rightarrow T(1,1,1,1) = F(1,1,1,1) = 1$
  - $\Rightarrow$ a subset of LUT instances has to be considered
  - Necessity to use ASIC synthesizers

Dualization in LUT4

	OR4		AND4		MUX2 $a \cdot \overline{d} + b \cdot d$		MUX2N $a \cdot d + b \cdot \overline{d}$	
$d\ c\ b\ a$	FFFE		8000		CCAA		AACC	
0 0 0 0	0		0		0		0	
0 0 0 1	1	E	0	0	1	A	0	C
0 0 1 0	1		0		0		1	
0 0 1 1	1		0		1		1	
0 1 0 0	1		0		0		0	
0 1 0 1	1	F	0	0	1	A	0	C
0 1 1 0	1		0		0		1	
0 1 1 1	1		0		1		1	
1 0 0 0	1		0		0		0	
1 0 0 1	1	F	0	0	0	C	1	A
1 0 1 0	1		0		1		0	
1 0 1 1	1		0		1		1	
1 1 0 0	1		0		0		0	
1 1 0 1	1	F	0	8	0	C	1	A
1 1 1 0	1		0		1		0	
1 1 1 1	1		1		1		1	

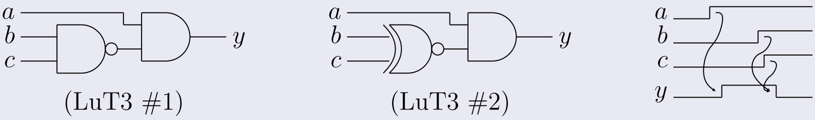
Example on Altera VQM (Verilog Quartus Mapping)

```
stratix_lcell \y~20_I (
.dataa(d),
.datab(b),
.datac(c),
.datad(a),
.combout(\y~20 ));
defparam \y~20_I .operation_mode = "normal";
defparam \y~20_I .synch_mode = "off";
defparam \y~20_I .register_cascade_mode = "off";
defparam \y~20_I .sum_lutc_input = "datac";
defparam \y~20_I .lut_mask = "EFFF";
defparam \y~20_I .output_mode = "comb_only";
```

```
lut_mask = "EFFF" = OR4    /* Direct */
lut_mask = "8000" = AND4   /* Dual   */
```

Positive Logic

Glitch can occur in a non-positive gate.



Solution: positive Logic

No local inversion if only OR and AND are used internally to the LUT $\Rightarrow \quad x \cdot y = x \quad f(x) \geq f(y)$

ASIC Synthesizers and specific Library

- FPGA synthesizers cannot be constrained to use only a subset of instances
- ASIC synthesizers = `bgx_shell` or `rc` from Cadence with custom lib

Number of cells in the ASIC LIBERTY cells of FPGA.

n	Complete			Compacted		
	plain	**WDDL**	**positive**	**plain**	**WDDL**	**positive**
2	16	4	4	4	2	2
3	256	64	18	14	11	5
4	65536	16384	166	222	212	16

bgx_shell synthesis

Table 1: Various substitution boxes area in LuT{2,3,4} with Cadence `bgx_shell` synthesis in compacted libraries.

Compacted library	DES S1	S2	S3	S4	S5	S6	S7	S8	AES S	S^{-1}
LuT2 plain	219	240	207	216	202	248	204	226	1248	1230
LuT2 WDDL	304	312	290	300	286	332	288	300	1716	1708
LuT2 positive	304	312	290	300	286	332	288	300	1716	1708
LuT3 plain	128	139	119	129	121	153	125	134	692	715
LuT3 WDDL	174	188	160	174	170	202	170	176	1012	1026
LuT3 positive	186	194	174	178	172	204	184	178	1024	1028
LuT4 plain	93	103	85	89	77	114	91	93	527	540
LuT4 WDDL	118	132	114	116	116	148	116	126	742	742
LuT4 positive	138	144	134	134	136	150	134	134	758	748

rc synthesis

Table 2: Various substitution boxes area in LuT{2,3,4} with Cadence `rc` synthesis in compacted libraries.

Compacted library	DES S1	S2	S3	S4	S5	S6	S7	S8	AES S	S^{-1}
LuT2 plain	235	215	229	238	236	236	234	232	1207	1232
LuT2 WDDL	312	294	300	310	310	314	314	316	1630	1668
LuT2 positive	312	294	300	310	310	314	314	316	1630	1668
LuT3 plain	168	157	153	158	164	159	168	171	834	837
LuT3 WDDL	208	186	192	196	196	200	204	204	1026	1044
LuT3 positive	210	194	196	200	196	204	206	204	1028	1054
LuT4 plain	126	121	125	136	122	124	131	125	636	641
LuT4 WDDL	158	144	152	152	156	150	156	148	788	798
LuT4 positive	160	146	152	154	156	152	156	1286	792	802

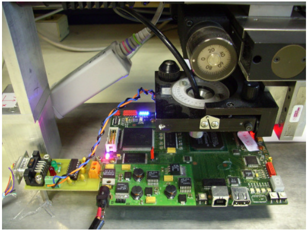
Area and speed results

Table 3: Performances of the regular and the two dual-rail DES modules.

Implementation	Single-ended DES	WDDL DES	WDDL+ DES
Area	1,248 LEs	4,736 LEs	6,038 LEs
Max. Frequency	74.95 MHz	68.65 MHz	55.85 MHz
DES-ECB speed	300 Mbit/s	137 Mbit/s	111 Mbit/s
3DES-OCB speed	99 Mbit/s	45 Mbit/s	37 Mbit/s

Attack

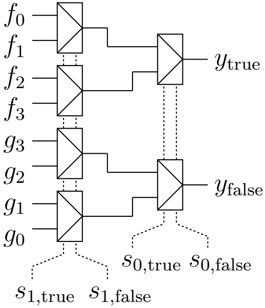
- XY table + 5GHz acquisition platform + DPA,CPA,EMA attack



- Regular DES : DPA Attack OK on allSboxes : maximum of 11103 traces
- WDDL : CPA attack OK for only 3 Sboxes : 193028 traces
- WDDL+ : DPA attack Only one Sbox attacked with 125743 traces (failure)

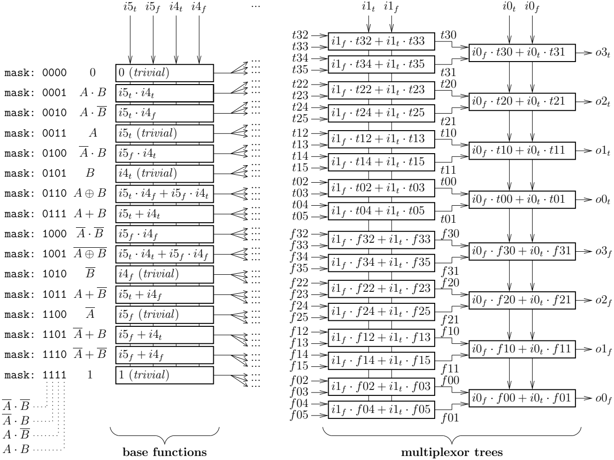
Optimization principle

- Each LUT4 is a positive MUX : $s_t = x_0 \cdot sel_f + x_1 \cdot sel_t$
- A Mux tree can be build for each Sbox



- optimizations are done by using symmetries and trivial functions to remove positive MUXes

Compact WDDL+ DES Sboxes: Architecture

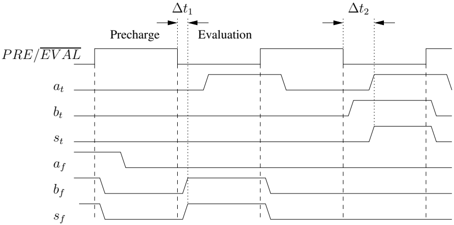


Optimization result

DES S-Box	Number of LUT4	
	heuristic	**bgx_shell**
# 1	102	138
# 2	98	144
# 3	98	134
# 4	64	134
# 5	106	134
# 6	98	136
# 7	96	150
# 8	86	134

Conclusions and perspectives

- WDDL can be attacked in FPGAS
- WDDL+ with positive functions greatly improves the countermeasures
- WDDL+ can be compacted in FPGAs
- Attacks can be improved by taking davantage of second order effects:



- Early evaluation

Dissemination

Related publications

- "*Evaluation of Power-Constant Dual-Rail Logic as a Protection of Cryptographic Applications in FPGA*", **SSIRI**, july 2008, Yokohama, Japan.
- "*Area Optimization of Cryptographic Co-Processors Implemented in Dual-Rail with Precharge Positive Logic*", **FPL**, september 2008, Heidelberg, Germany.
- "*Place-and-Route Impact on the Security of DPL Designs in FPGAs*", **HOST**, june 2008, Anaheim, CA, USA.

Valorisation

- 3 softwares deposited at the APP
- Spin-off in creation: http://www.Trust-IC.com/