Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow

Nele Mentens, Edoardo Charbon, Francesco Regazzoni
nele.mentens@kuleuven.be, edoardo.charbon@epfl.ch, regazzoni@alari.ch

Cryptarchi, 29 June 2018

Outline

- Configurable computing
  - What? Why? Why for crypto?
- Configurable computing for crypto
  - How?
- Target application platforms

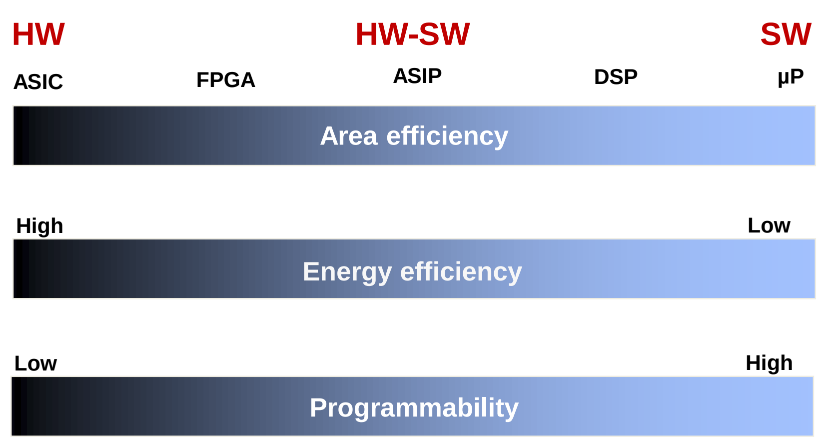
Outline

- **Configurable computing**
  - **What? Why? Why for crypto?**
- Configurable computing for crypto
  - How?
- Target application platforms

Configurable computing

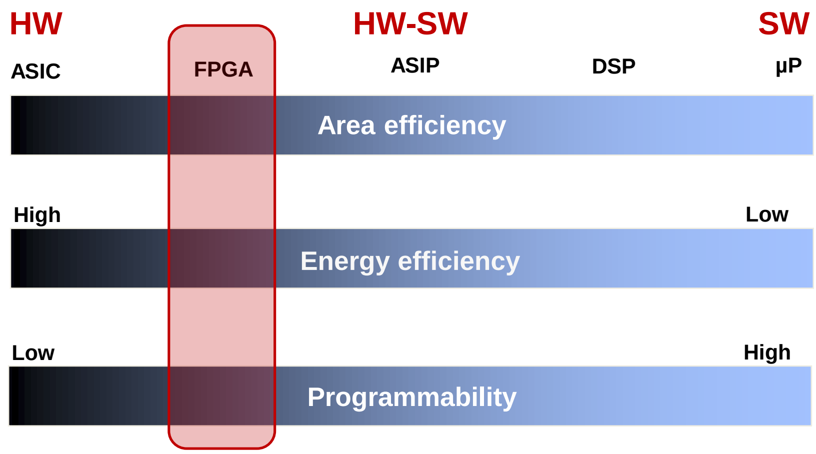
What?

Configurable computing platforms are hardware platforms to which changes in the datapath can be applied in addition to changes in the control flow



Configurable computing
What?

FPGAs are the most widely used configurable computing platforms



Configurable computing
Why?

- Configurable computing platforms have
  - the flexibility of software,
  - the performance of hardware,
  - the energy efficiency of hardware.

Configurable computing
Why for crypto?

- Minimize the overhead of crypto in terms of performance and area/energy efficiency
- Provide cryptographic agility, i.e. the ability of cryptographic implementations to be upgraded or updated depending on newly detected vulnerabilities or changing standards

Outline

- Configurable computing
  - What? Why? Why for crypto?
- **Configurable computing for crypto**
  - **How?**
- Target application platforms

Configurable computing for crypto
How?

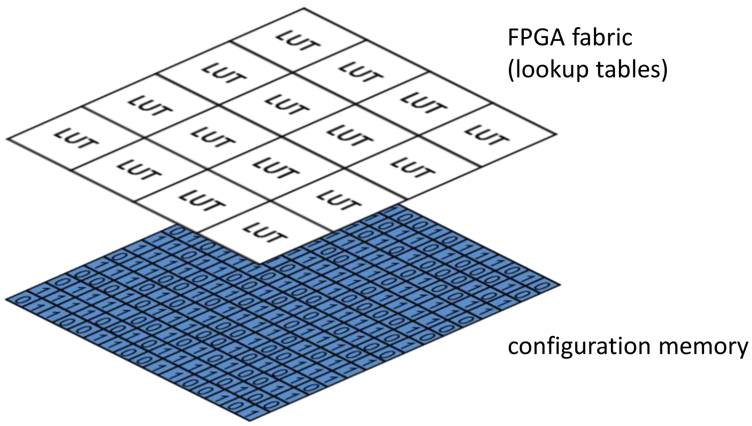
- Cryptographic algorithms
- Cryptographic algorithms on commercial FPGAs
- Crypto-oriented configurable hardware

Cryptographic algorithms

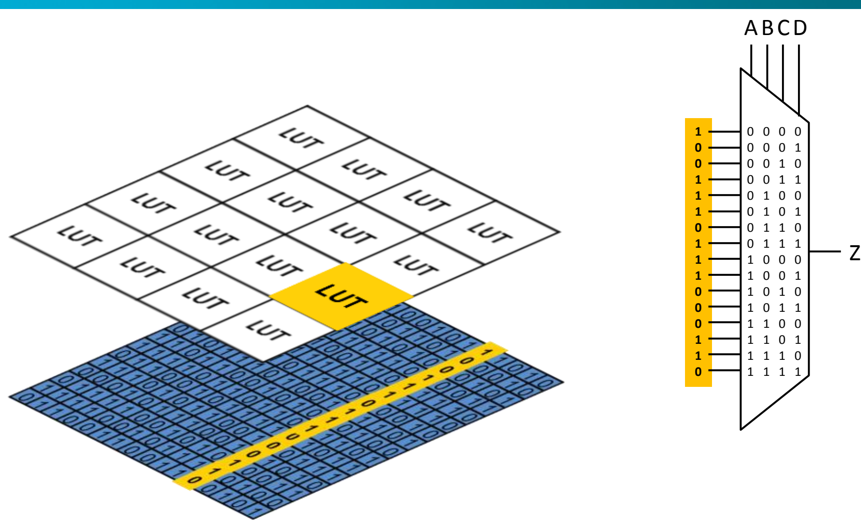
- Symmetric-key cryptography:
  - bit permutation
  - rotation
  - addition modulo 2^n (in ARX-based ciphers)
  - addition modulo 2, i.e. exclusive OR (XOR)
  - substitution box (S-box)
  - quadratic functions (for threshold implementations)

    $f(x,y,z,w) = a_0 \oplus a_1 x \oplus a_2 y \oplus a_3 z \oplus a_4 w \oplus a_{12} xy \oplus a_{13} xz \oplus a_{14} xw \oplus a_{23} yz \oplus a_{24} yw \oplus a_{34} zw$
- Public-key cryptography:
  - operations in prime fields
  - operations in binary extension fields

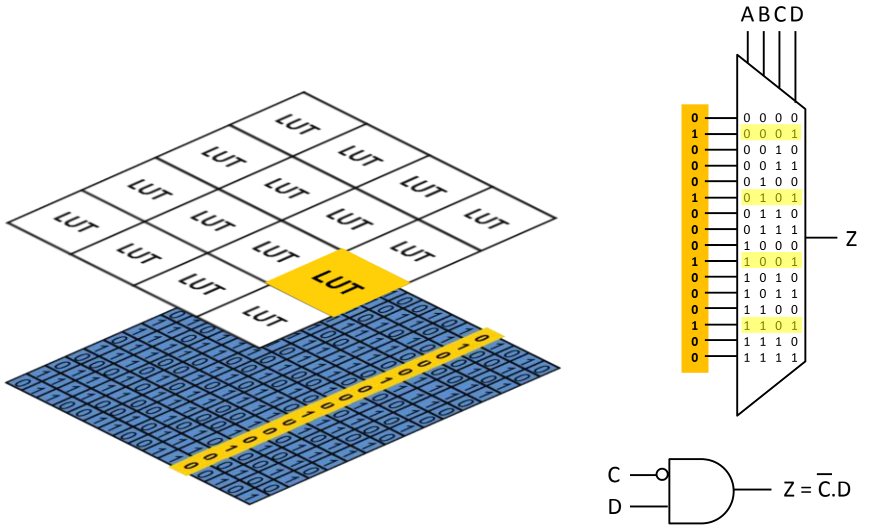
Commercial FPGAs



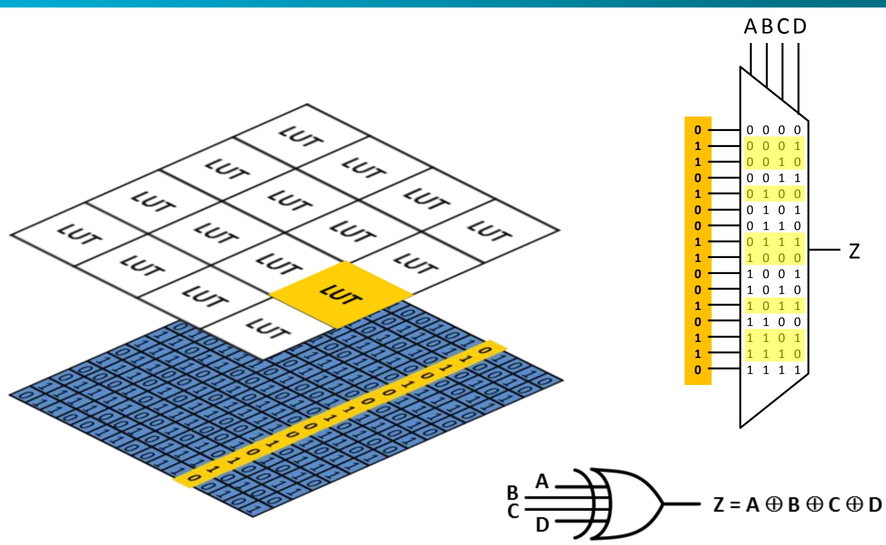
Commercial FPGAs



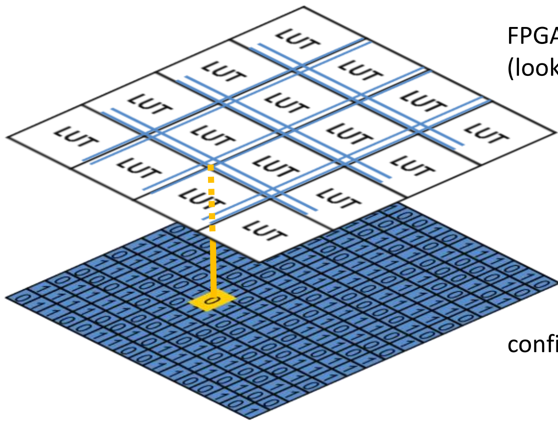
Commercial FPGAs



Commercial FPGAs



Commercial FPGAs

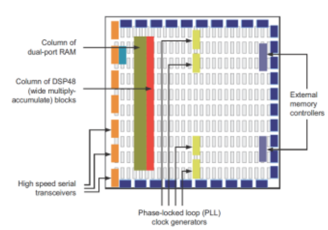


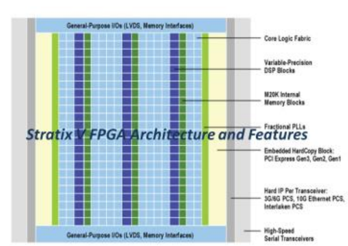
FPGA fabric
(lookup tables + routing)

configuration memory

Commercial FPGAs

- Besides lookup tables and routing elements, FPGAs also contain:
  - distributed and centralized storage (BRAM),
  - DSP blocks,
  - fast carry chains,
  - high-speed I/O,
  - clock management blocks.





Crypto algorithms on commercial FPGAs

- Symmetric-key cryptography:
  - bit permutation
  - rotation
  - addition modulo 2^n (in ARX-based ciphers)
  - addition modulo 2, i.e. exclusive OR (XOR)
  - substitution box (S-box)
  - quadratic functions (for threshold implementations)

    $f(x,y,z,w) = a_0 \oplus a_1x \oplus a_2y \oplus a_3z \oplus a_4w \oplus a_{12}xy \oplus a_{13}xz \oplus a_{14}xw \oplus a_{23}yz \oplus a_{24}yw \oplus a_{34}zw$
- Public-key cryptography:
  - operations in prime fields
  - operations in binary extension fields

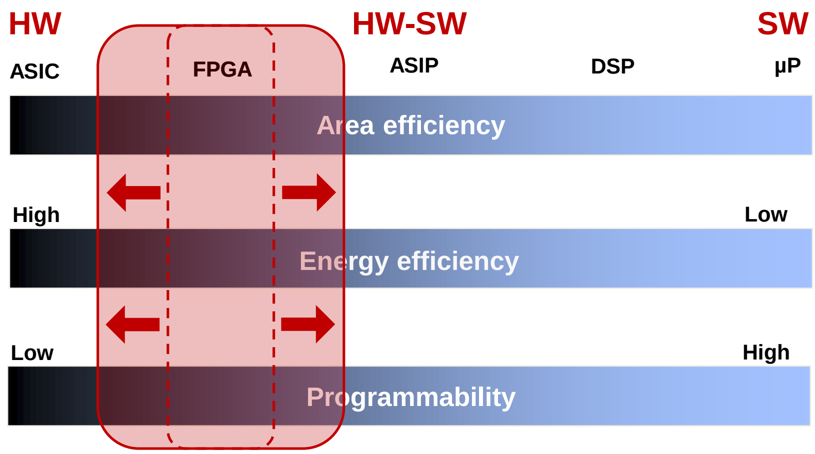
Crypto algorithms on commercial FPGAs

- Symmetric-key cryptography:
  - bit permutation $\rightarrow$ routing
  - rotation $\rightarrow$ routing
  - addition modulo 2^n (in ARX-based ciphers) $\rightarrow$ fast carry chains
  - addition modulo 2, i.e. exclusive OR (XOR) $\rightarrow$ LUTs
  - substitution box (S-box) $\rightarrow$ LUTs, BRAM
  - quadratic functions (for threshold implementations) $\rightarrow$ LUTs

    $f(x,y,z,w) = a_0 \oplus a_1x \oplus a_2y \oplus a_3z \oplus a_4w \oplus a_{12}xy \oplus a_{13}xz \oplus a_{14}xw \oplus a_{23}yz \oplus a_{24}yw \oplus a_{34}zw$
- Public-key cryptography:
  - operations in prime fields $\rightarrow$ DSP blocks, fast carry chains
  - operations in binary extension fields $\rightarrow$ LUTs

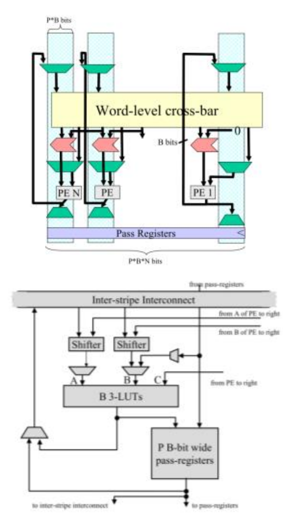
Crypto-oriented configurable hardware

Solutions consist of coarse-grained configurable HW architectures (combined with SW programmability) or fine-grained configurable HW architectures



PipeRench [1]

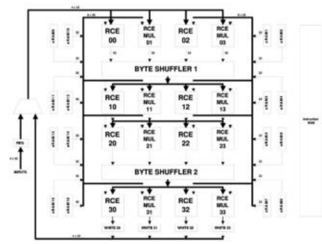
- Coarse-grained reconfigurable architecture
  - Parallel stripes of processing elements with pipelining registers in between
- 6 (unprotected) block ciphers evaluated
- 0.25 µm technology
- 2x-12x speedup over microprocessors
- No comparison to FPGAs
- Dedicated compiler with Dataflow Intermediate Language (DIL) as an input

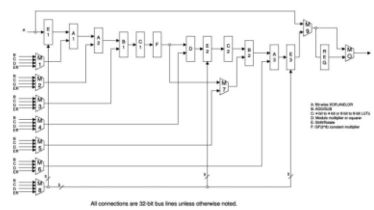


[1] R. Taylor and S. Goldstein, "A high-performance flexible architecture for cryptography", CHES 1999.

COBRA [2]

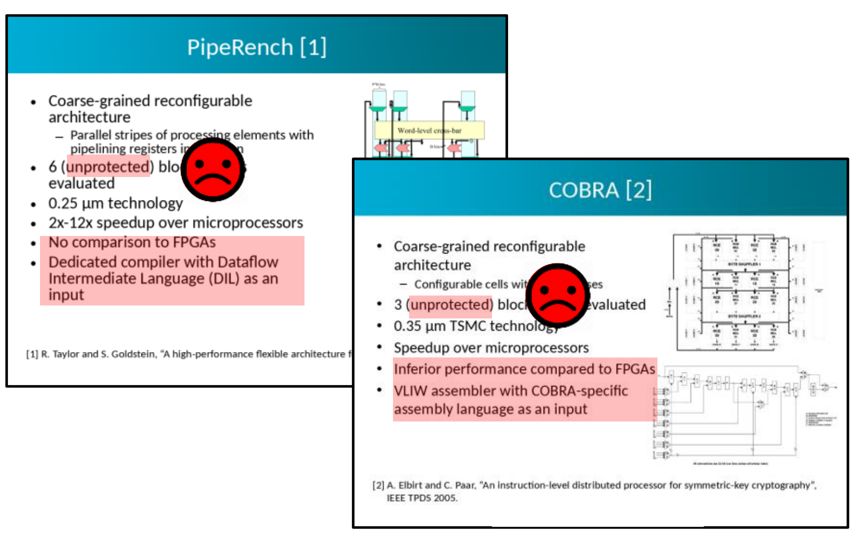
- Coarse-grained reconfigurable architecture
  - Configurable cells with 32-bit buses
- 3 (unprotected) block ciphers evaluated
- 0.35 µm TSMC technology
- Speedup over microprocessors
- Inferior performance compared to FPGAs
- VLIW assembler with COBRA-specific assembly language as an input





[2] A. Elbirt and C. Paar, "An instruction-level distributed processor for symmetric-key cryptography", IEEE TPDS 2005.

What's the problem?

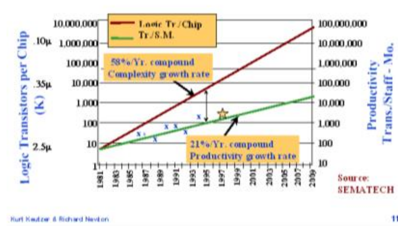


How can we do better?

1. Be lazy, why work if we can (re-)use the efforts of others
   – No new design tools → rely on the decades of experience of EDA companies

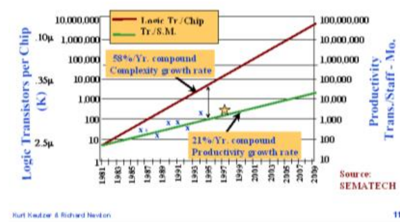
How can we do better?

1. Be lazy, why work if we can (re-)use the efforts of others
   – No new design tools → rely on the decades of experience of EDA companies
2. Maximize productivity
   – No new design languages/input → use VHDL, Verilog, HLS



How can we do better?

1. Be lazy, why work if we can (re-)use the efforts of others
   – No new design tools → rely on the decades of experience of EDA companies
2. Maximize productivity
   – No new design languages/input → use VHDL, Verilog, HLS
3. Optimize efficiency
   – Area (= cost), performance



Cryptographic algorithms

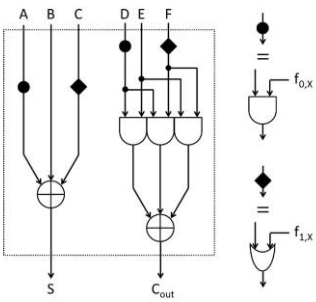
- Symmetric-key cryptography:
  – bit permutation
  – rotation
  – addition modulo 2^n (in ARX-based ciphers)

    $S = A \oplus B \oplus C_{in}$ $\quad$ $C_{out} = AB + (A+B)C_{in} = AB \oplus AC_{in} \oplus BC_{in}$
  – addition modulo 2, i.e. exclusive OR (XOR)
  – substitution box (S-box)
  – quadratic functions (for threshold implementations)

    $f(x,y,z,w) = a_0 \oplus a_1x \oplus a_2y \oplus a_3z \oplus a_4w \oplus a_{12}xy \oplus a_{13}xz \oplus a_{14}xw \oplus a_{23}yz \oplus a_{24}yw \oplus a_{34}zw$
- Public-key cryptography:
  – operations in prime fields
  – operations in binary extension fields

cFA-based cell architecture [3]

- Fine-grained reconfigurable architecture
- Matrix of configurable Full Adder (cFA) cells



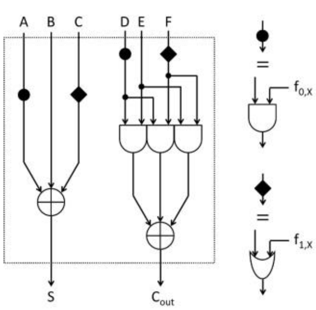
$f_{0,A}$	$f_{1,C}$	S
0	0	$0 \oplus B \oplus C = B \oplus C$
0	1	$0 \oplus B \oplus 1 = \overline{B}$
1	0	$A \oplus B \oplus C$
1	1	$A \oplus B \oplus 1 = \overline{A \oplus B}$

$f_{0,D}$	$f_{1,F}$	C_{out}
0	0	$0 \oplus 0 \oplus EF = EF$
0	1	$0 \oplus 0 \oplus E = E$
1	0	$DE \oplus DF \oplus EF = DE + (D+E)F$
1	1	$DE \oplus D \oplus E = D + E$

[3] N. Mentens, E. Charbon, and F. Regazzoni, “Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow”, accepted at FCCM 2018.

cFA-based cell architecture [3]

- Fine-grained reconfigurable architecture
- Matrix of configurable Full Adder (cFA) cells
- One cFA (with 6 inputs and 2 outputs) can be programmed to 8 functions
  - 8 functions are available in most standard cell libraries → **re-use ASIC synthesis tools**



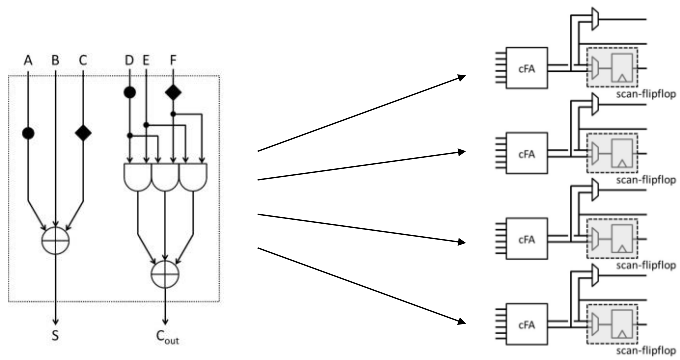
$f_{0,A}$	$f_{1,C}$	S
0	0	$0 \oplus B \oplus C = B \oplus C$
0	1	$0 \oplus B \oplus 1 = \overline{B}$
1	0	$A \oplus B \oplus C$
1	1	$A \oplus B \oplus 1 = \overline{A \oplus B}$

$f_{0,D}$	$f_{1,F}$	C_{out}
0	0	$0 \oplus 0 \oplus EF = EF$
0	1	$0 \oplus 0 \oplus E = E$
1	0	$DE \oplus DF \oplus EF = DE + (D+E)F$
1	1	$DE \oplus D \oplus E = D + E$

[3] N. Mentens, E. Charbon, and F. Regazzoni, “Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow”, accepted at FCCM 2018.

cFA-based cell architecture [3]

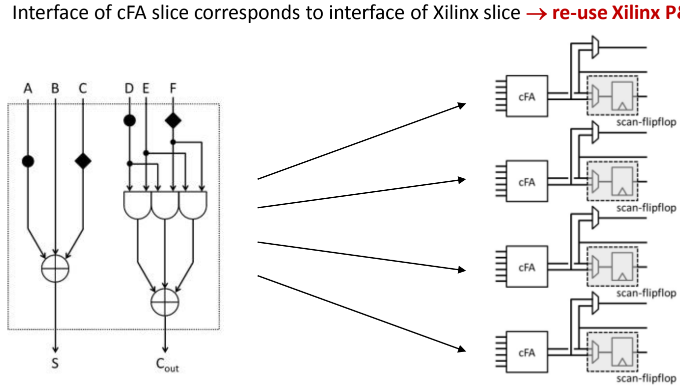
- 4 cFA cells and 4 flipflops are packed into one cFA slice



[3] N. Mentens, E. Charbon, and F. Regazzoni, "Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow", accepted at FCCM 2018.

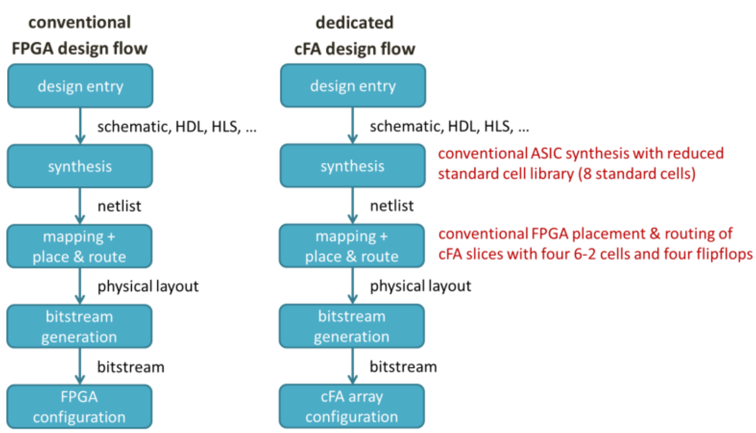
cFA-based cell architecture [3]

- 4 cFA cells and 4 flipflops are packed into one cFA slice
- Interface: 3x6 inputs, 3x3 outputs
  - Interface of cFA slice corresponds to interface of Xilinx slice → **re-use Xilinx P&R tools**



[3] N. Mentens, E. Charbon, and F. Regazzoni, "Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow", accepted at FCCM 2018.

cFA-based automated design flow [3]



[3] N. Mentens, E. Charbon, and F. Regazzoni, "Rethinking Secure FPGAs: Towards a Cryptography-friendly Configurable Cell Architecture and its Automated Design Flow", accepted at FCCM 2018.

Results after synthesis and mapping

cipher	Xilinx					cFA array			
	SLICEL	SLICEM	area	critical path	conf	cFA	area	critical path	conf
AES-128	404	0	179,053	4.95	105,040	624	27,886	4.97	14,976
PRESENT-80-D3	70	0	31,024	1.65	18,200	190	8,491	0.95	4,560
PRESENT-80-D2	74	0	32,797	1.65	19,240	139	6,212	1.64	3,336
SPECK-128/128	130	0	57,616	5.99	33,800	294	13,139	9.91	7,056
NOEKEON	168	0	74,458	3.30	43,680	288	12,871	2.41	6,912
KTANTAN-64	80	0	35,456	2.2	20,800	119	5,318	1.95	2,856
AES-128-TI	2,076	120	1,092,679	2.2	582,640	3,058	136,656	1.54	73,392
PRESENT-80-TI	350	0	155,120	1.65	91,000	638	28,511	1.01	15,312
SPECK-128/128-TI	436	0	193,235	1.10	113,360	1556	69,535	1.33	37,344
NOEKEON-TI	846	0	374,947	2.75	219,960	952	42,543	2.12	22,848

- TI = threshold implementation
- SLICEL / SLICEM / cFA: number of Xilinx slices / cFa slices
- area (µm²) and critical path (ns): based on re-implemented Xilinx slices / cFA slices in NanGate 45nm technology
- conf: number of configuration bits

Results after synthesis and mapping

3x – 9x area decrease

cipher	Xilinx					cFA array			
	SLICEL	SLICEM	area	critical path	conf	cFA	area	critical path	conf
AES-128	404	0	179,053	4.95	105,040	624	27,886	4.97	14,976
PRESENT-80-D3	70	0	31,024	1.65	18,200	190	8,491	0.95	4,560
PRESENT-80-D2	74	0	32,797	1.65	19,240	139	6,212	1.64	3,336
SPECK-128/128	130	0	57,616	5.99	33,800	294	13,139	9.91	7,056
NOEKEON	168	0	74,458	3.30	43,680	288	12,871	2.41	6,912
KTANTAN-64	80	0	35,456	2.2	20,800	119	5,318	1.95	2,856
AES-128-TI	2,076	120	1,092,679	2.2	582,640	3,058	136,656	1.54	73,392
PRESENT-80-TI	350	0	155,120	1.65	91,000	638	28,511	1.01	15,312
SPECK-128/128-TI	436	0	193,235	1.10	113,360	1556	69,535	1.33	37,344
NOEKEON-TI	846	0	374,947	2.75	219,960	952	42,543	2.12	22,848

- TI = threshold implementation
- SLICEL / SLICEM / cFA: number of Xilinx slices / cFa slices
- area (µm²) and critical path (ns): based on re-implemented Xilinx slices / cFA slices in NanGate 45nm technology
- conf: number of configuration bits

Results after synthesis and mapping

similar speed

cipher	Xilinx					cFA array			
	SLICEL	SLICEM	area	critical path	conf	cFA	area	critical path	conf
AES-128	404	0	179,053	4.95	105,040	624	27,886	4.97	14,976
PRESENT-80-D3	70	0	31,024	1.65	18,200	190	8,491	0.95	4,560
PRESENT-80-D2	74	0	32,797	1.65	19,240	139	6,212	1.64	3,336
SPECK-128/128	130	0	57,616	5.99	33,800	294	13,139	9.91	7,056
NOEKEON	168	0	74,458	3.30	43,680	288	12,871	2.41	6,912
KTANTAN-64	80	0	35,456	2.2	20,800	119	5,318	1.95	2,856
AES-128-TI	2,076	120	1,092,679	2.2	582,640	3,058	136,656	1.54	73,392
PRESENT-80-TI	350	0	155,120	1.65	91,000	638	28,511	1.01	15,312
SPECK-128/128-TI	436	0	193,235	1.10	113,360	1556	69,535	1.33	37,344
NOEKEON-TI	846	0	374,947	2.75	219,960	952	42,543	2.12	22,848

- TI = threshold implementation
- SLICEL / SLICEM / cFA: number of Xilinx slices / cFa slices
- area (µm²) and critical path (ns): based on re-implemented Xilinx slices / cFA slices in NanGate 45nm technology
- conf: number of configuration bits

Results after synthesis and mapping

3x – 9x configuration memory decrease

cipher	Xilinx					cFA array			
	SLICEL	SLICEM	area	critical path	conf	cFA	area	critical path	conf
AES-128	404	0	179,053	4.95	105,040	624	27,886	4.97	14,976
PRESENT-80-D3	70	0	31,024	1.65	18,200	190	8,491	0.95	4,560
PRESENT-80-D2	74	0	32,797	1.65	19,240	139	6,212	1.64	3,336
SPECK-128/128	130	0	57,616	5.99	33,800	294	13,139	9.91	7,056
NOEKEON	168	0	74,458	3.30	43,680	288	12,871	2.41	6,912
KTANTAN-64	80	0	35,456	2.2	20,800	119	5,318	1.95	2,856
AES-128-TI	2,076	120	1,092,679	2.2	582,640	3,058	136,656	1.54	73,392
PRESENT-80-TI	350	0	155,120	1.65	91,000	638	28,511	1.01	15,312
SPECK-128/128-TI	436	0	193,235	1.10	113,360	1556	69,535	1.33	37,344
NOEKEON-TI	846	0	374,947	2.75	219,960	952	42,543	2.12	22,848

- TI = threshold implementation
- SLICEL / SLICEM / cFA: number of Xilinx slices / cFa slices
- area (µm²) and critical path (ns): based on re-implemented Xilinx slices / cFA slices in NanGate 45nm technology
- conf: number of configuration bits

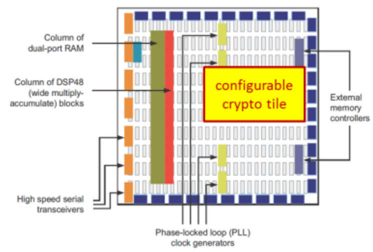
Outline

- Configurable computing
  - What? Why? Why for crypto?
- Configurable computing for crypto
  - How?
- **Target application platforms**

Target application platforms

- Embedded FPGAs (eFPGAs)
  - Configurable logic integrated in ASICs
  - Growing market (Achronix, Flex Logix, Menta)
  - Size matters!

- Dedicated configurable crypto tiles in existing FPGAs
  - Following the trend of adding dedicated features such as BRAM, DSP slices, microprocessors, fast carry chains,...
  - Speed matters!



Questions?

- Thank you for your attention!