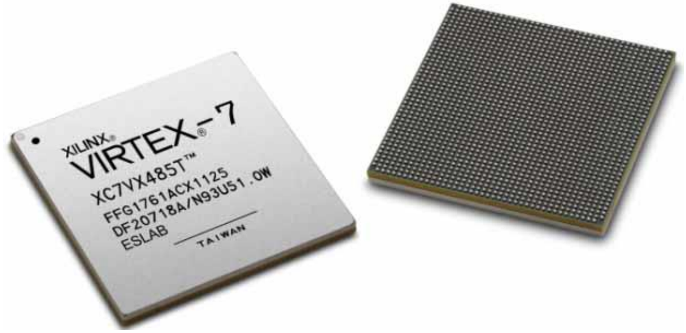
technology
from seed

Secure partial dynamic reconfiguration of FPGAs

Hirak Kashyap and Ricardo Chaves INESC-ID / IST



inesc id lisboa

Outline

- Introduction & Motivation
  - ❑ Computing, configurable devices, and FPGAs
  - ❑ FPGA dynamic re-configuration, benefits, and applications
  - ❑ Attacks against FPGAs
  - ❑ Built-in FPGA security features
  - ❑ State of the art in Secure dynamic reconfiguration
- Proposed solution: description and analysis
- Implementation
  - ❑ The 3-AES crypto kernel
  - ❑ The 1-AES crypto kernel
  - ❑ The configuration processor
- Evaluation
  - ❑ Resource requirements and performance
  - ❑ Comparison with related state of the art
- Conclusions

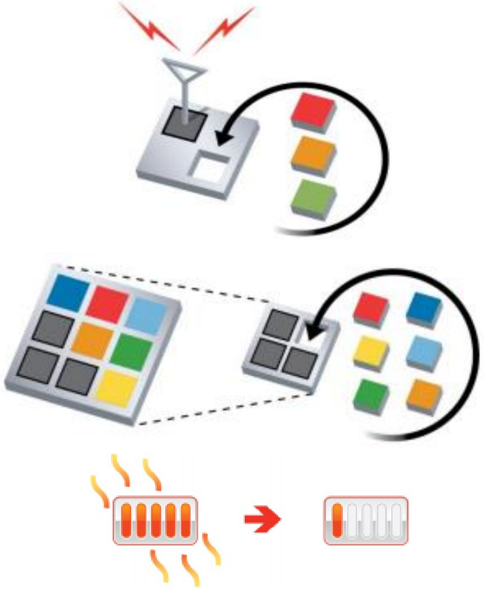
Introduction

Benifits of dynamic reconfiguration

- System Flexibility
  - Performs changing functionality

- Size and Cost Reduction
  - Time-multiplexing of hardware require a smaller FPGA

- Reduction in energy consumption
  - Shut down power-hungry tasks when not needed

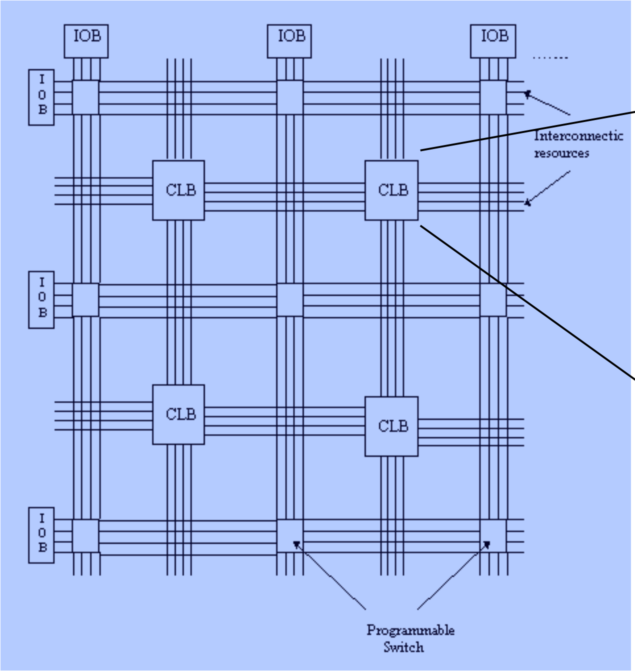


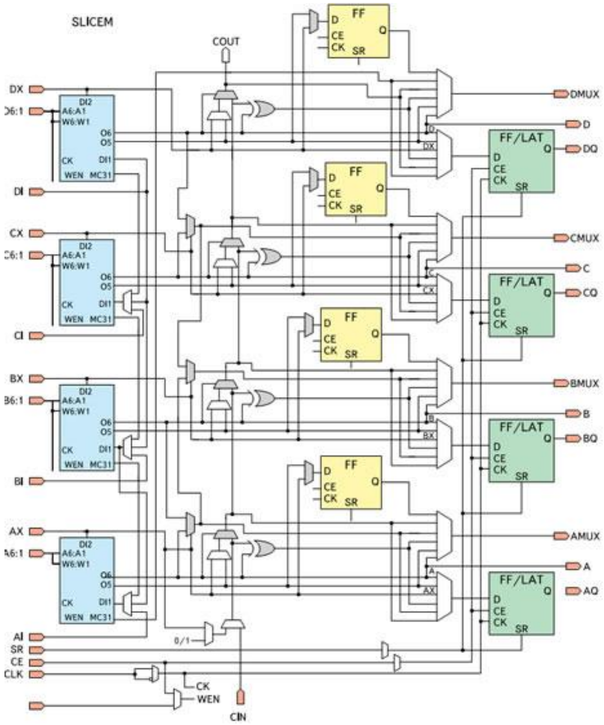
- Applications requiring dynamic reconfiguration:
  - Communication HUBs
  - Multipurpose satellites / micro-satellites
  - Robotic rovers and orbiters
  - Software defined radio etc.

Introduction

FPGAs - Organization

technology
from seed





Structure:

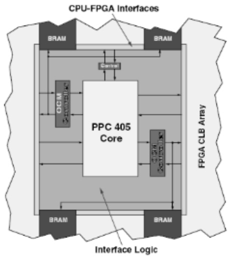
- Configurable Logic Blocks (CLBs)
- Interconnection network
- Programmable Switches
- I/O Interfaces
- Others

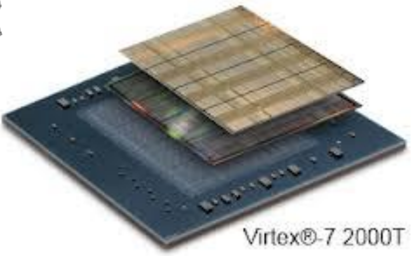
CLBs (slices):

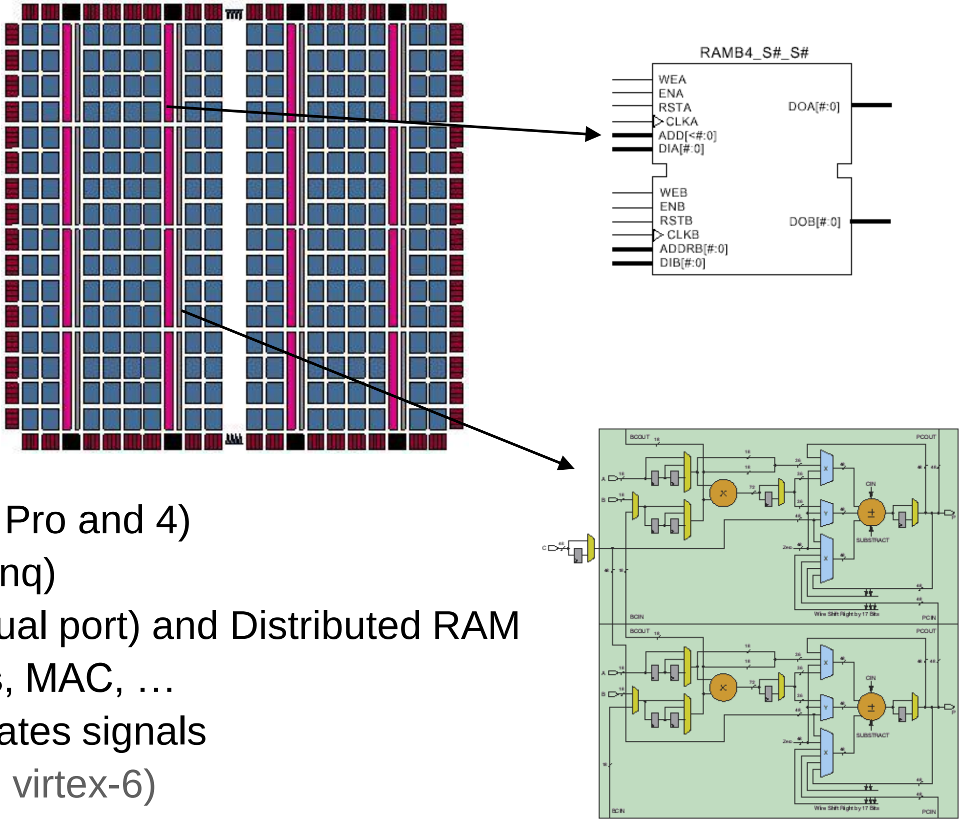
- Look Up Tables (LUT)
- Registers (FF)
- Fast carry chains
- Multiplexers
- Selection logic

Introduction

FPGAs – Extra features







- Processor cores:
  - PowerPC (Xilinx VIRTEX II Pro and 4)
  - Dual-core ARM 9 (Xilinx Zynq)
- Embedded memory blocks: BRAM (dual port) and Distributed RAM
- DSP blocks: Multiplications, Additions, MAC, …
- Digital Clock Manager – Multiple clk rates signals
- MultiBoot up to 4 configurations (from virtex-6)
- ADC (virtex-7)
- Dedicated High throughput IOs
- Dynamic Partial Reconfiguration

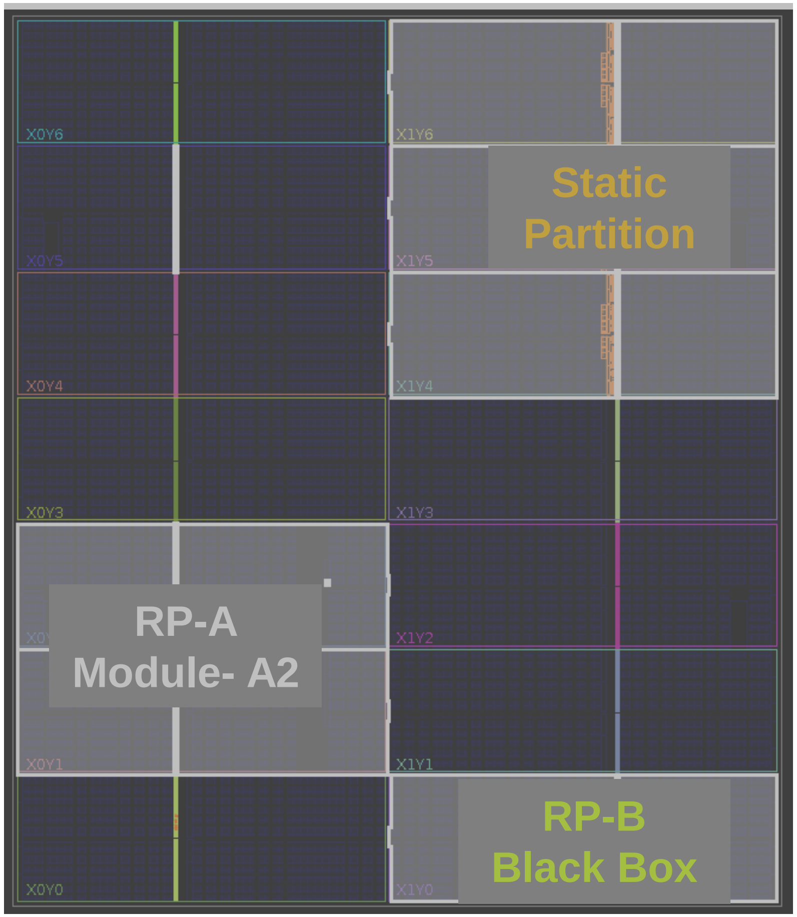
Introduction

FPGAs - Configuration

- Configuration: full or partial

- Full configuration:
  - Configures the LUTs, BRAMs, and interconnections of the whole device
  - Volatile FPGAs have to be configured every time after power-on

- Partial configuration:
  - Configures a specific part of the device
  - Partial configuration types:
    - Shutdown
    - Dynamic
  - Dynamic configuration is highly useful if hardware reconfigures frequently

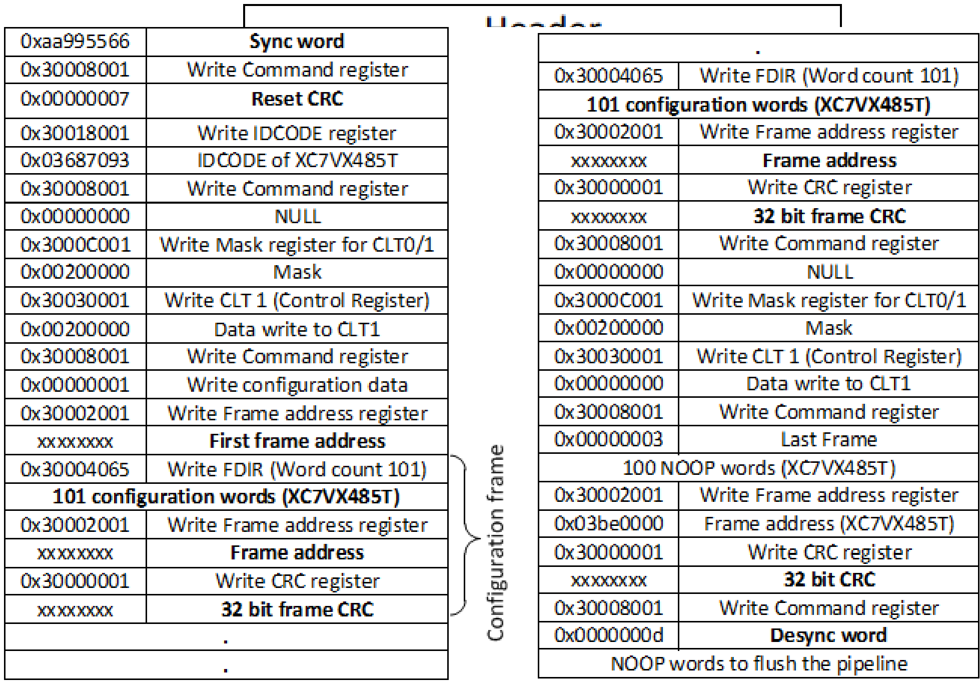


Introduction

FPGAs – Configuration bitstream

- FPGA is configured using a configuration file called Bitstream
  - Header (Ignored by the FPGA)
  - Sync word
  - General configuration info
  - Frame configuration data
    - Write location registers
    - Configuration data:
      - LUTs, BRAMs, routing, ...
  - Final CRC
  - Stop/Descync word

- Bitstream types:
  1. Full configuration
  2. Partial configuration
  - more or less the same as for full configuration



Header
Configuration frame

0xaa995566	**Sync word**
0x30008001	Write Command register
0x00000007	**Reset CRC**
0x30018001	Write IDCODE register
0x03687093	IDCODE of XC7VX485T
0x30008001	Write Command register
0x00000000	NULL
0x3000C001	Write Mask register for CLT0/1
0x00200000	Mask
0x30030001	Write CLT 1 (Control Register)
0x00200000	Data write to CLT1
0x30008001	Write Command register
0x00000001	Write configuration data
0x30002001	Write Frame address register
xxxxxxxx	**First frame address**
0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	**Frame address**
0x30000001	Write CRC register
xxxxxxxx	**32 bit frame CRC**
.	
.	

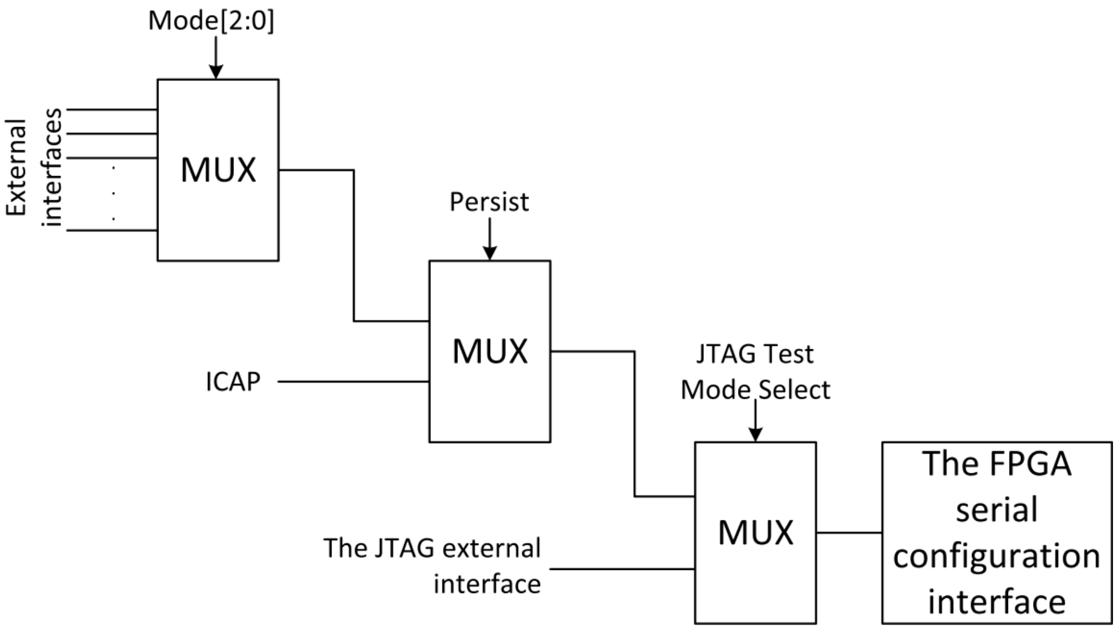
.	
0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	**Frame address**
0x30000001	Write CRC register
xxxxxxxx	**32 bit frame CRC**
0x30008001	Write Command register
0x00000000	NULL
0x3000C001	Write Mask register for CLT0/1
0x00200000	Mask
0x30030001	Write CLT 1 (Control Register)
0x00000000	Data write to CLT1
0x30008001	Write Command register
0x00000003	Last Frame
100 NOOP words (XC7VX485T)	
0x30002001	Write Frame address register
0x03be0000	Frame address (XC7VX485T)
0x30000001	Write CRC register
xxxxxxxx	**32 bit CRC**
0x30008001	Write Command register
0x0000000d	**Desync word**
NOOP words to flush the pipeline	

Configuration bitstream format

Introduction

FPGAs – Configuration interfaces

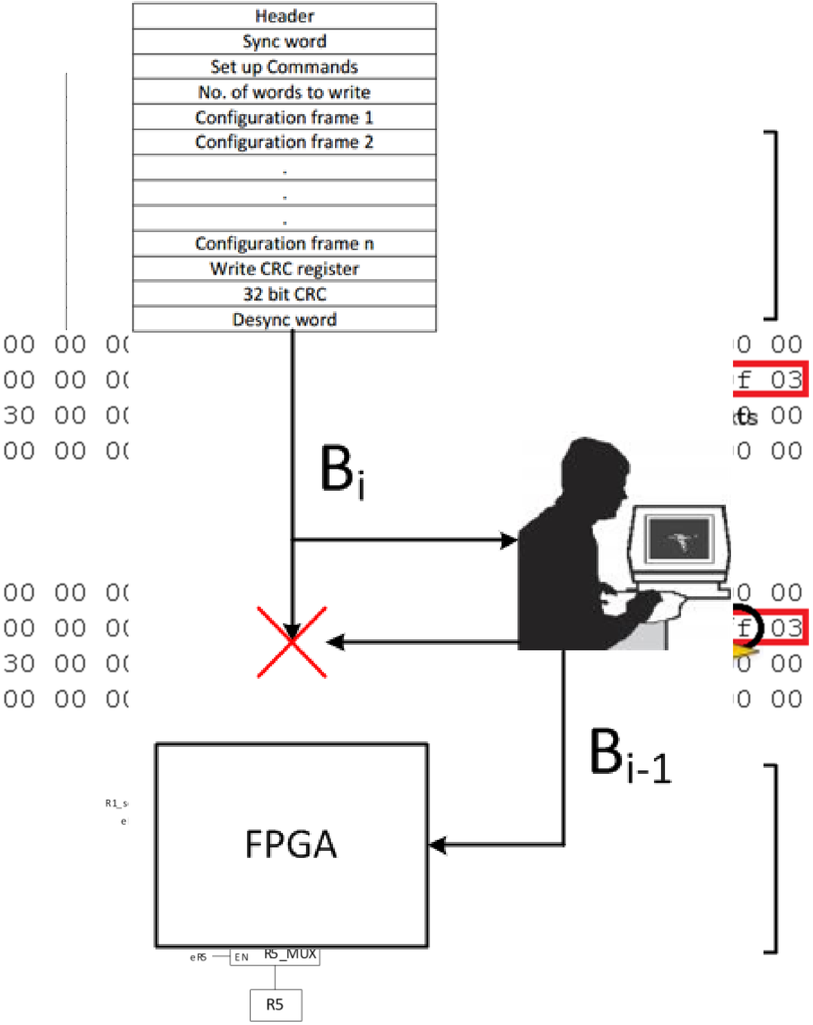
- Configuration interfaces:
  - External configuration ports (JTAG, SelectMAP, Serial, BPI, and SPI) or
  - Internal configuration access port - ICAP or ICAPE2 (Xilinx 7 series)
- They internally use the same FPGA serial configuration interface. Only one can be used at a given moment.
  - To use ICAP after initial configuration: *BitGen –g Persist:No*



State of the art

Attacks against FPGAs

- **Cloning of SRAM FPGAs**
  - Eavesdropping and read-back
- **Reverse engineering of the bitstreams**
  - Design can be reconstructed from raw bitstream
- **Bitstream tampering**
  - Manipulate a particular field of the bitstream, MITM attacks.
- **Side channel attacks**
  - Power analysis, timing behavior attacks etc.
- **Replay attacks**
  - Prevent reconfiguration with an updated bitstream (System Downgrade)



State of the art

FPGAs – Xilinx built-in security mechanisms

- Bitstream Encryption
  - Software-based bitstream encryption and on-chip bitstream decryption
  - **Same key** for **all bitstreams** and cannot be reprogrammed without resetting the device
  - Key is **stored internally** in either *Battery Backed RAM (BBRAM)* or *eFUSE* (OTP)
  - AES-CBC (256-bit)
- Bitstream Authentication
  - Available **when using bitstream encryption**
  - The authentication key is not stored inside the FPGA
  - The key and the MAC are sent **as a part of the encrypted bitstream**
  - SHA-2 (256-bit)
- Improved partial bitstream integrity
  - Error in the address portion of the partial configuration bitstream can overwrite the static portion
  - **Frame-wise CRC** verification was introduced in the **Xilinx 7 series** FPGAs

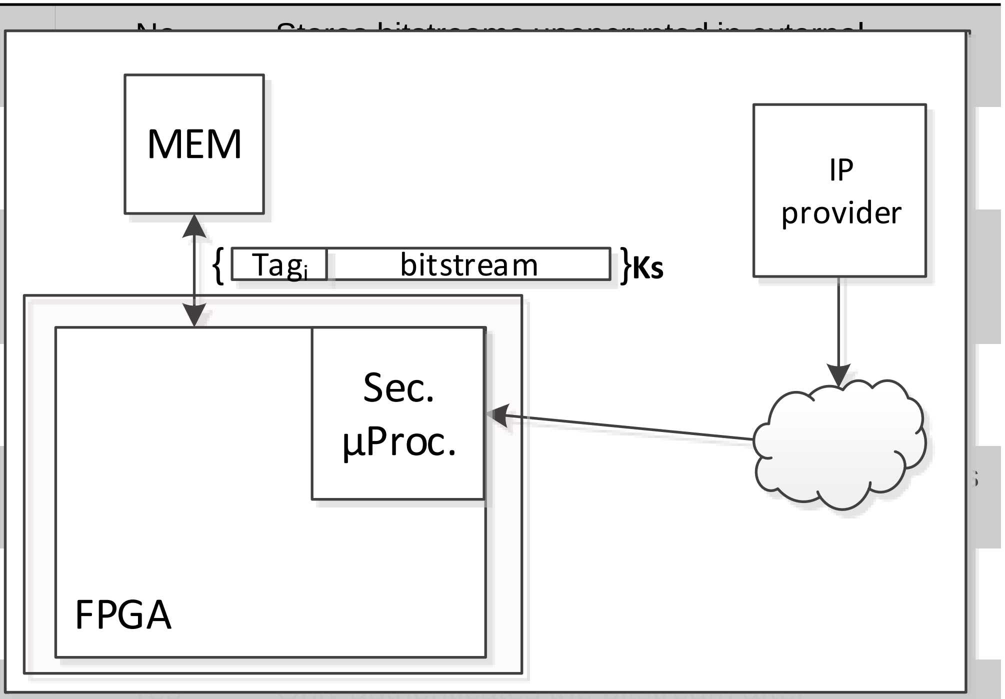
State of the art

FPGAs – Vulnerabilities in Xilinx built-in security

- Single key for all the bitstreams
  - Simplifies replay attacks and cryptanalysis
- The authentication tag and the key are part of the bitstream
  - Unable to detect the replay attacks using unintended bitstream
- The authentication is verified only at the end of the bitstream
  - The static partitions may be overwritten by tampered bitstreams
- Bitstream has a strict and well known format
  - Particular fields of the bitstream can be attacked
- CRC field can be attacked
  - CRC can be deterministically calculated for the tampered frame. The tampered frame will pass the CRC by ICAP before configuration.
- Unprotected AES implementation
  - Susceptible to DPA attacks allowing to retrieve the internal key

The state of the art

Ref.	Objective	System Downgrade	Secure external storage	Remarks
Braeken (2011)	Secure remote reconfiguration	Possible	No	Stores bitstreams unencrypted in external
Vliegen (2013)	Secure remote reconfiguration	Not possible		
Vliegen (2014)	Secure remote reconfiguration	Possible		
Hori (2012)	Prevent side channel attacks	Possible		
Hori (2013)	AES-GCM on Bit Stream Block	Possible		
Kepa (2008)	Integrity of DPR System	Possible		
Devic (2012)	Prevent Replay attacks	Not Possible	Yes	Only authenticates the bitstream after reconfiguration



Additional work exists more focused on the IP retrieval:

- Guneysu2007, Drimmer2009, Braeken2011, Vliegen2014

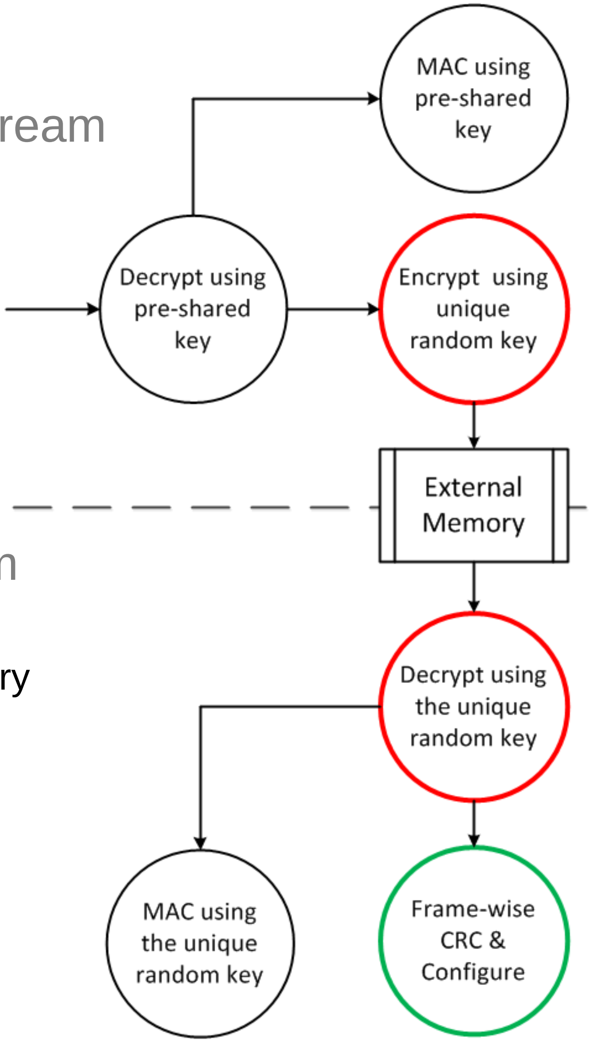
Outline

- Introduction & Motivation
  - ❑ Benefits and applications
  - ❑ FPGAs and reconfiguration
  - ❑ Attacks against configuration bitstreams
  - ❑ Technology support for bitstream security
  - ❑ State of the art in Secure dynamic reconfiguration
- **Proposed solution: description and analysis**
- Implementation
  - ❑ The cryptographic kernel
- Evaluation
  - ❑ Resource requirements and performance
  - ❑ Comparison with related state of the art
- Conclusions

The proposed solution

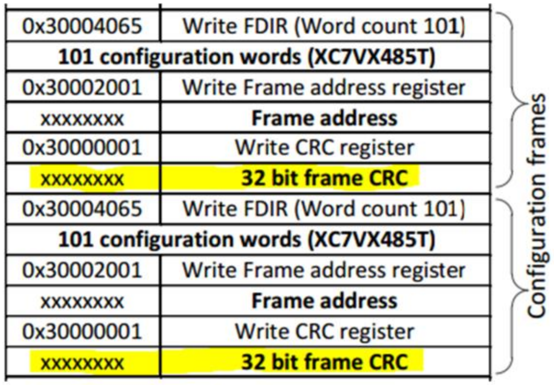
Divided into two phases:

- Phase 1 – Reception, validation, and storage of bitstream
  - Validation of the received (remote) bitstream
  - Re-encryption and storage in the external memory
    - Using an unique random key, stored internally
  - Storage of the new key and MAC inside the device
    - On an internal BRAM
  - Can be at any time without particular time constrains
- Phase 2 – Reconfiguration using the stored bitstream
  - Retrieved the internal key for a given bitstrem
  - Loads the decrypts the bitstream from the external memory
  - Simultaneously validates the bitstream
    - MAC calculation
    - Encrypted CRC send it to the configuration port (ICAP)
  - Performed on-the-fly
    - During operation/ reconfiguration
    - Using a 32 bits port

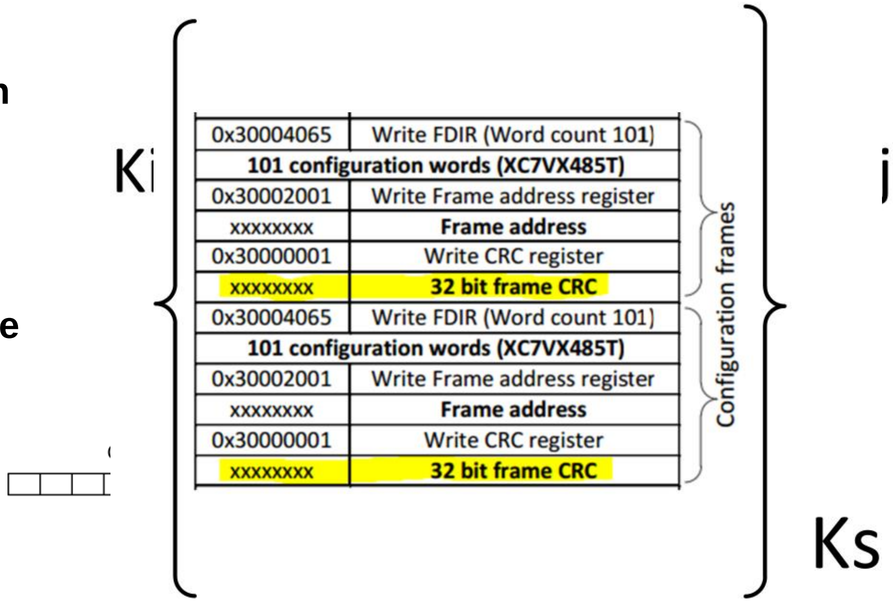

MAC using pre-shared key
Decrypt using pre-shared key
Encrypt using unique random key
External Memory
Decrypt using the unique random key
MAC using the unique random key
Frame-wise CRC & Configure

The proposed solution: security features

- **Authenticated encryption**
  - Confidentiality and authentication of the received bitstream
    - Using a shared key (K_S) with the IP source
- **Separate internal keys for external storage**
  - Prevents replay attack using a out-of-date bitstream
    - Unique random keys
- **Partial on-the-fly bitstream integrity validation**
  - Allows for on-the-fly bitstream authentication
    - By encrypting the bitstream and its CRC value
  - Final MAC verification using the unique key
- **Cipher block chaining (CBC) vs. counter mode**
  - Counter mode works like a stream cipher, CRC may be manipulated



0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	Frame address
0x30000001	Write CRC register
xxxxxxxx	32 bit frame CRC
0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	Frame address
0x30000001	Write CRC register
xxxxxxxx	32 bit frame CRC



0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	Frame address
0x30000001	Write CRC register
xxxxxxxx	32 bit frame CRC
0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	Frame address
0x30000001	Write CRC register
xxxxxxxx	32 bit frame CRC

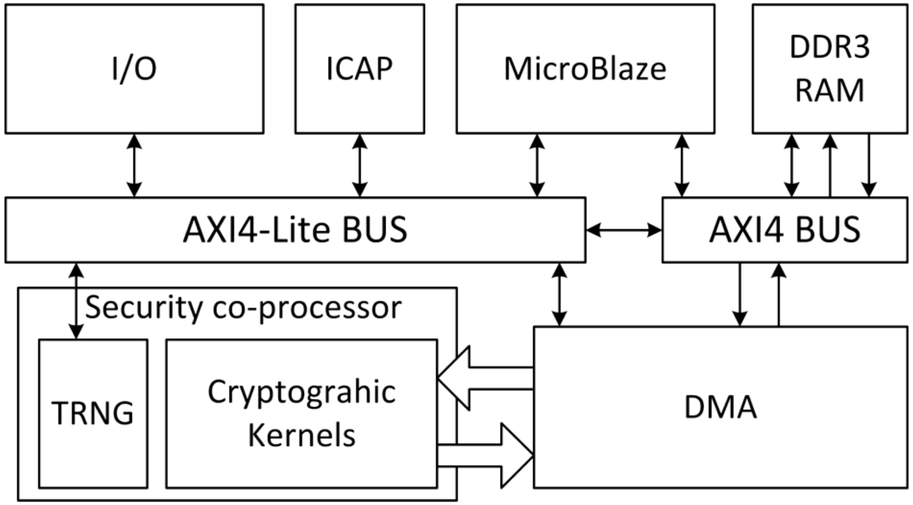
Outline

- Introduction & Motivation
  - ❑ Benefits and applications
  - ❑ FPGAs and reconfiguration
  - ❑ Attacks against configuration bitstreams
  - ❑ Technology support for bitstream security
  - ❑ State of the art in Secure dynamic reconfiguration
- Proposed solution: description and analysis
- Implementation
  - ❑ The cryptographic kernel
- Evaluation
  - ❑ Resource requirements and performance
  - ❑ Comparison with related state of the art
- Conclusions

Implementation

The configuration processor: System Architecture

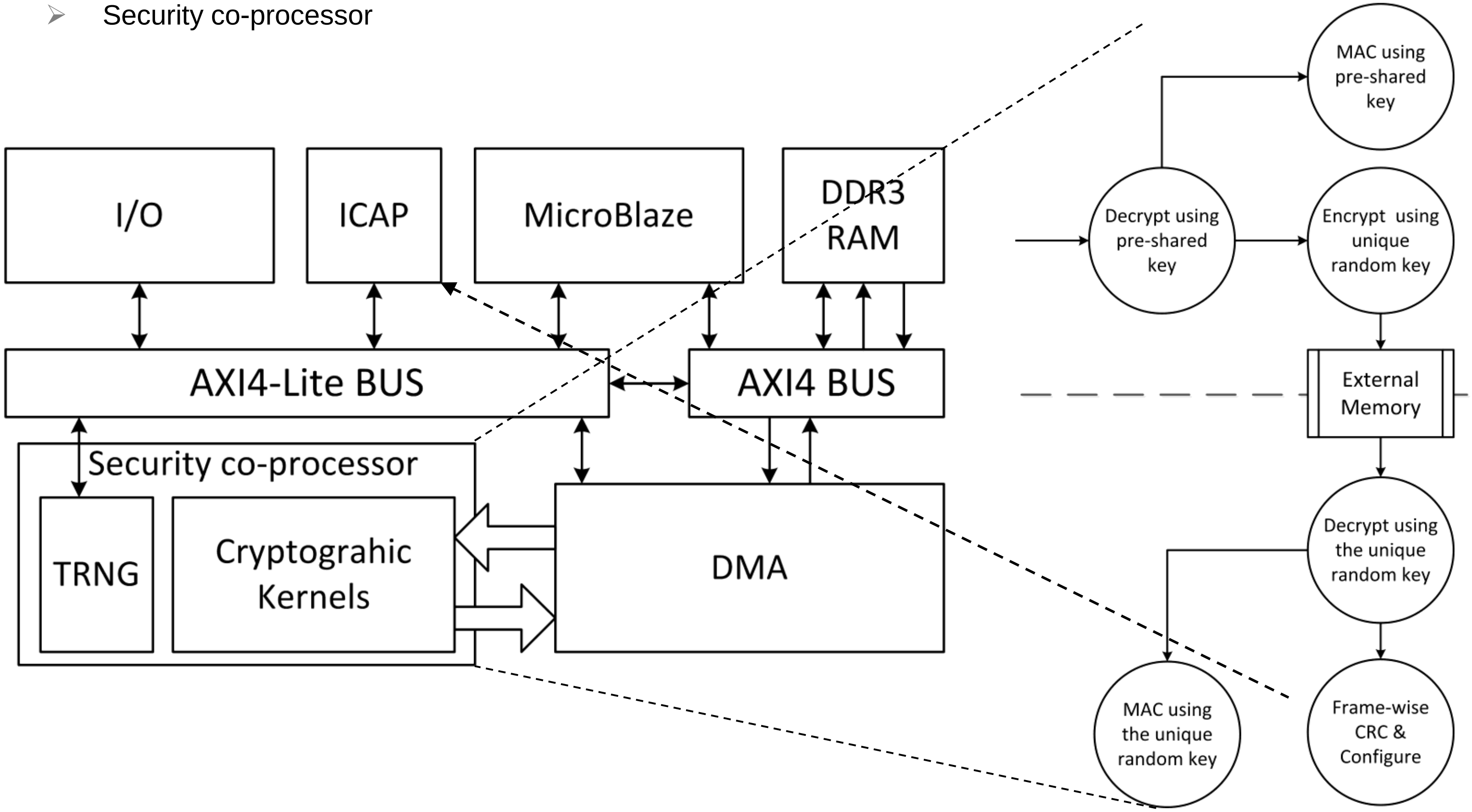
- A prototype was implemented on a Xilinx Virtex-7
  - VC707 FPGA board with a Virtex 7 XC7VX485T
    - 75k Slices, 1k BRAMs
- Core components:
  - MicroBlaze processor
  - AXI DMA core
  - Buses:
    - AXI4, AXI4-Lite, and AXI4 Stream
  - AXI DDR3 RAM controller
  - I/Os
    - serial com or ethernet port
  - ICAPE2 interface
- Security Co-processor
  - TRNG [1]
    - used for Key generation
      - connected to the MicroBlaze processor via the AXI4-Lite bus
  - Cryptographic Kernel



[1] Wold, K. and Tan, C. H. (2008). *Analysis and enhancement of random number generator in FPGA based on oscillator rings.* International Conference on Reconfigurable Computing and FPGAs, 0:385–390

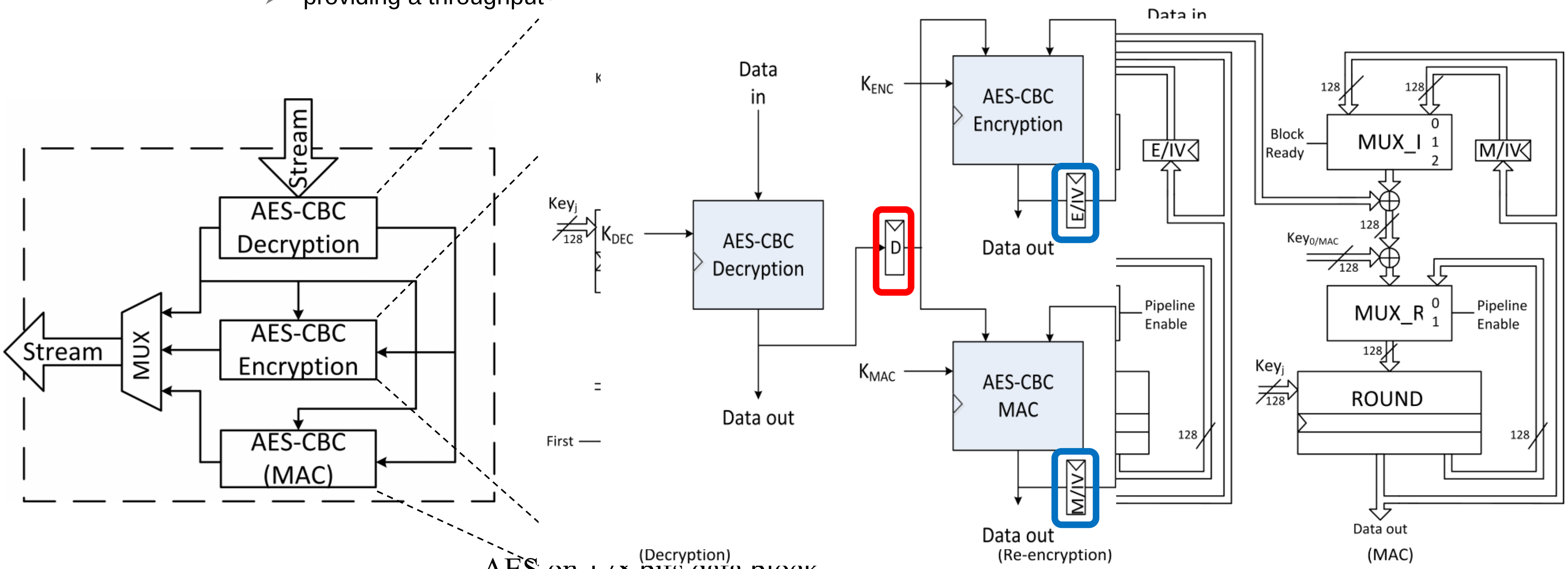
Implementation: System Architecture

- A prototype was implemented on a Xilinx Virtex-7 VC707 FPGA board
- The cryptographic operations are implemented in a single modular component
  - Security co-processor


I/O
ICAP
MicroBlaze
DDR3 RAM
AXI4-Lite BUS
AXI4 BUS
Security co-processor
TRNG
Cryptograhic Kernels
DMA
MAC using pre-shared key
Decrypt using pre-shared key
Encrypt using unique random key
External Memory
Decrypt using the unique random key
MAC using the unique random key
Frame-wise CRC & Configure

Implementation: The cryptographic kernel

- The cryptographic kernel architecture:
  - Uses three AES-CBC cores for the decryption, re-encryption, and CBC-MAC operations.
  - A folded architecture of Advanced Encryption Standard (AES) algorithm is used
    - requires 10 cycles per block.
    - providing a throughput of 128 bits each 10 cycles



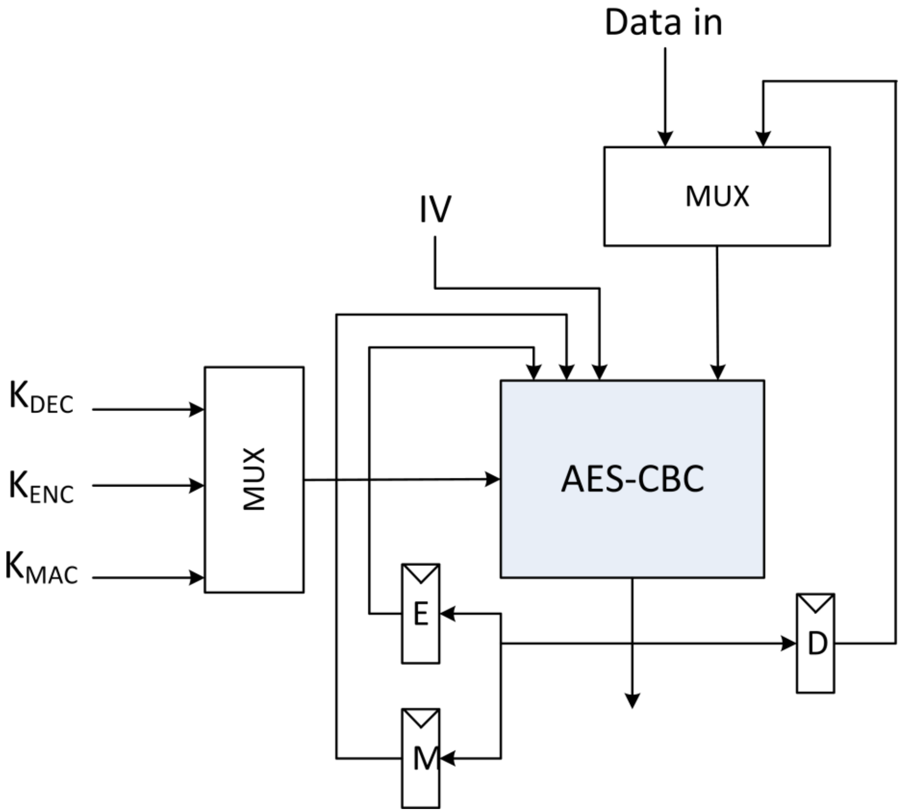
AES on 128 bits data block

Implementation

The cryptographic kernel: 1-AES crypto kernel

1-AES crypto kernel architecture

- A single AES-CBC core
  - performs decryption, re-encryption, and MAC
- Lower resource requirements
- Registers **D**, **E**, and **M** serve the same purpose as in the previous architecture



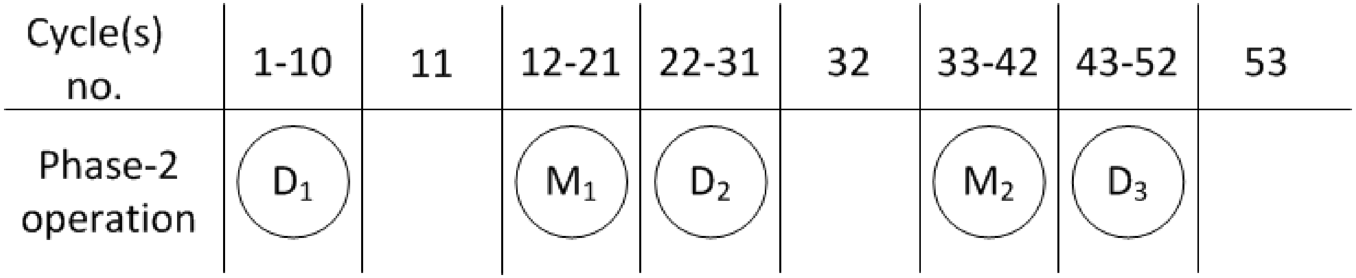
1-AES crypto kernel

Implementation

Scheduling – Option 1

- The encryption latency is of 11 cycles
  - and has a throughput of 1 block per 10 cycles
  - The re-encryption and MAC operations depend on the decrypted block (register D)

Phase 2 scheduling



Cycle(s) no.	1-10	11	12-21	22-31	32	33-42	43-52	53
Phase-2 operation	D_1		M_1	D_2		M_2	D_3	

- The kernel is idle for one cycle after each decryption:
  - Phase 2 throughput:
    - 128bits/21 cycles

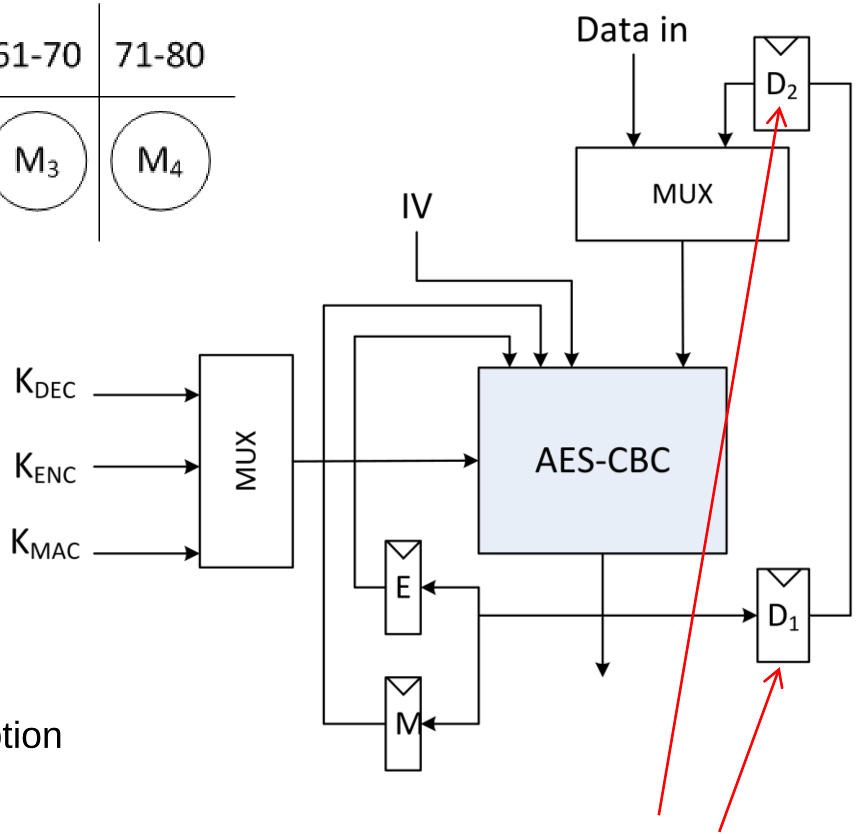
Implementation

Improved Scheduling – Option 2

Phase 2 rescheduling

Cycles no.	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
Phase-2 operation	D_1	D_2	M_1	D_3	M_2	D_4	M_3	M_4

- Throughput
  - Phase 1: 128 bits/ **30** cycles
  - Phase 2: 128 bits/ **20** cycles
- Resulting structure:
  - Requires one more register to held the decryption result
  - Critical path remains the same

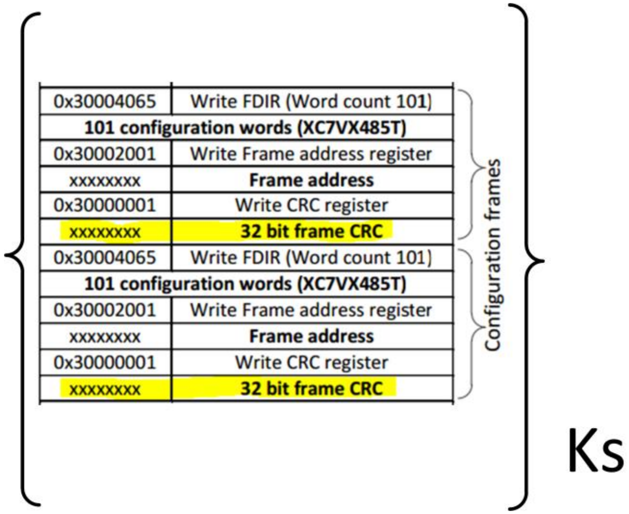


D1 and D2 registers store two consecutive decryption results

Implementation

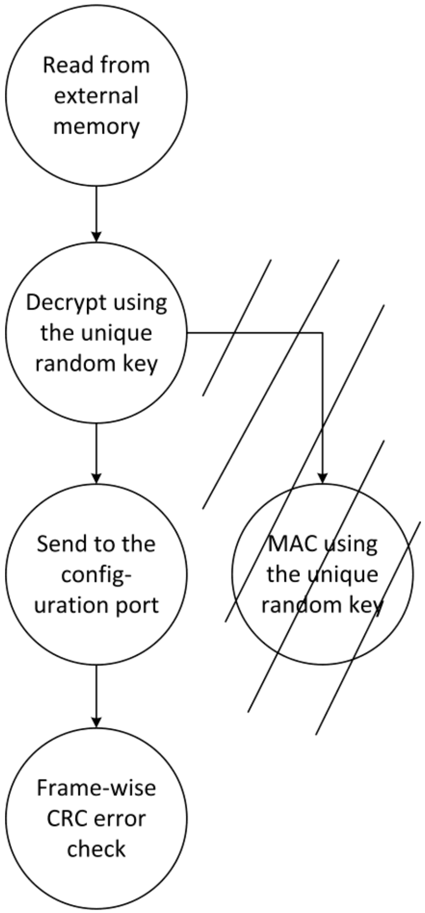
Improving phase 2 performance on VIRTEX 7

- Remove the MAC verification in phase 2
  - The authenticity is provided by the frame-wise CRC of VIRTEX 7
    - combined with the block cipher encryption



0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	**Frame address**
0x30000001	Write CRC register
xxxxxxxx	**32 bit frame CRC**
0x30004065	Write FDIR (Word count 101)
101 configuration words (XC7VX485T)	
0x30002001	Write Frame address register
xxxxxxxx	**Frame address**
0x30000001	Write CRC register
xxxxxxxx	**32 bit frame CRC**

  - Achieves identical throughput as the 3-AES crypto kernel solution (in phase 2)
    - 128 bits per 10 cycles



Read from external memory
Decrypt using the unique random key
Send to the config-uration port
MAC using the unique random key
Frame-wise CRC error check

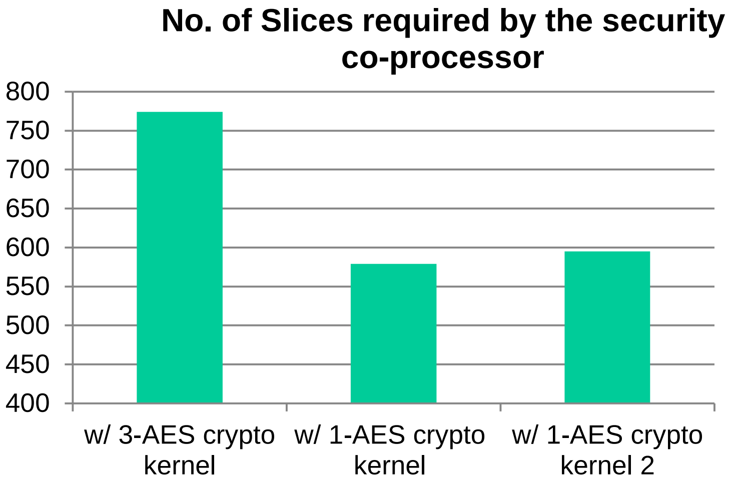
Phase 2 - Reconfiguration

Outline

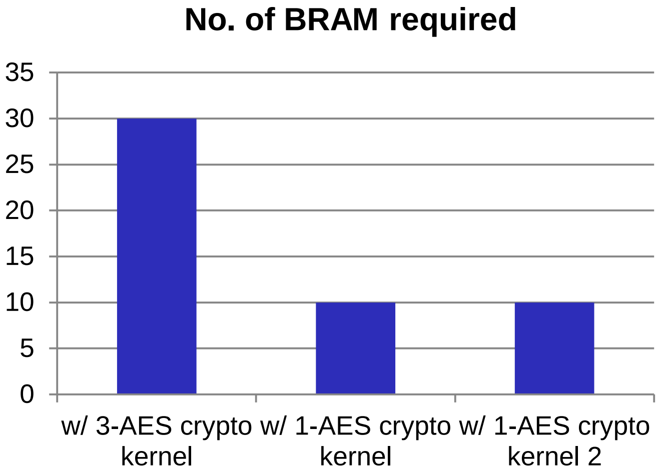
- Introduction & Motivation
  - ❑ Benefits and applications
  - ❑ FPGAs and reconfiguration
  - ❑ Attacks against configuration bitstreams
  - ❑ Technology support for bitstream security
  - ❑ State of the art in Secure dynamic reconfiguration
- Proposed solution: description and analysis
- Implementation
  - ❑ The cryptographic kernel
- **Evaluation**
  - ❑ Resource requirements and performance
  - ❑ Comparison with related state of the art
- Conclusions

Experimental Evaluation

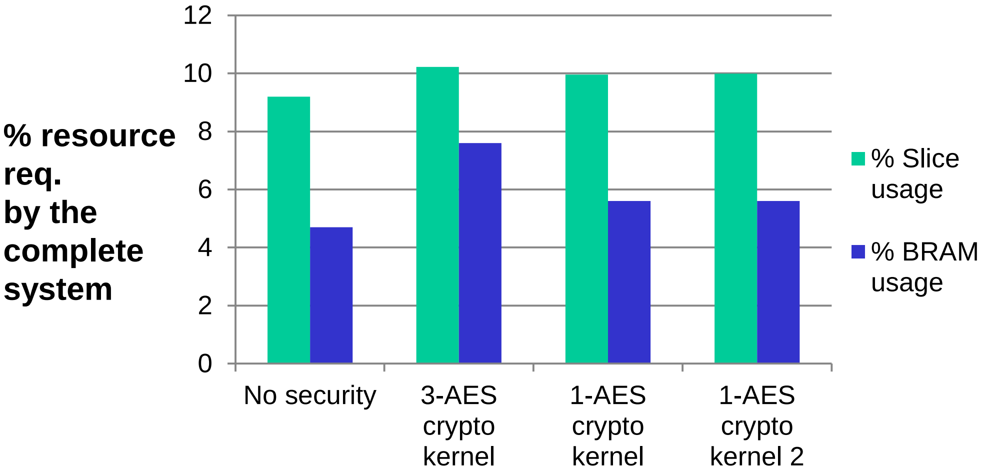
- Experimental results were obtained on a Xilinx Virtex-7 FPGA device XC7VX485T.

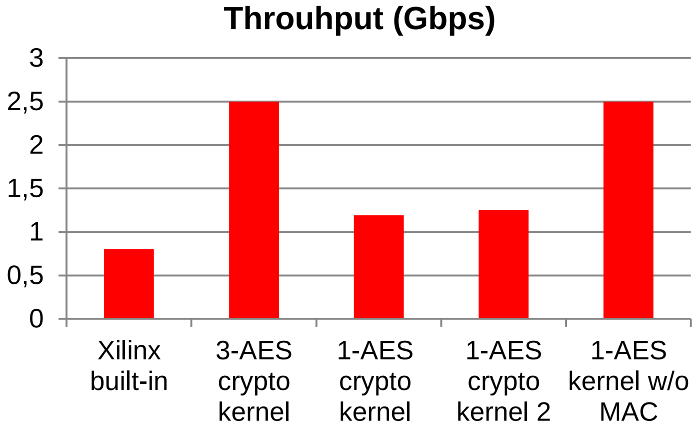


No. of Slices required by the security co-processor



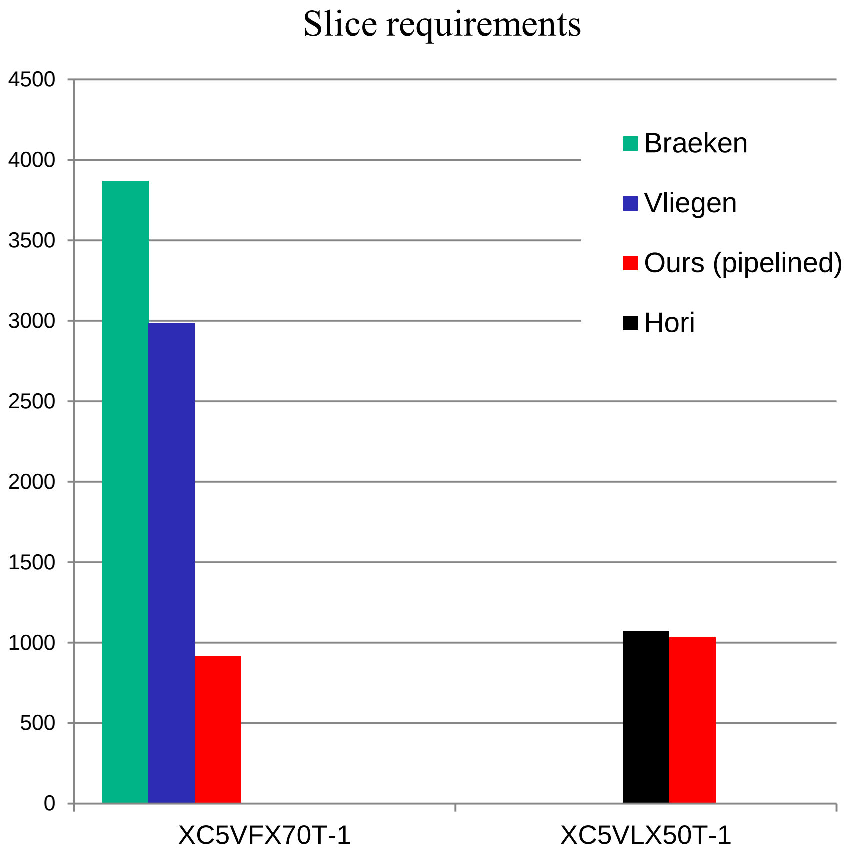
No. of BRAM required

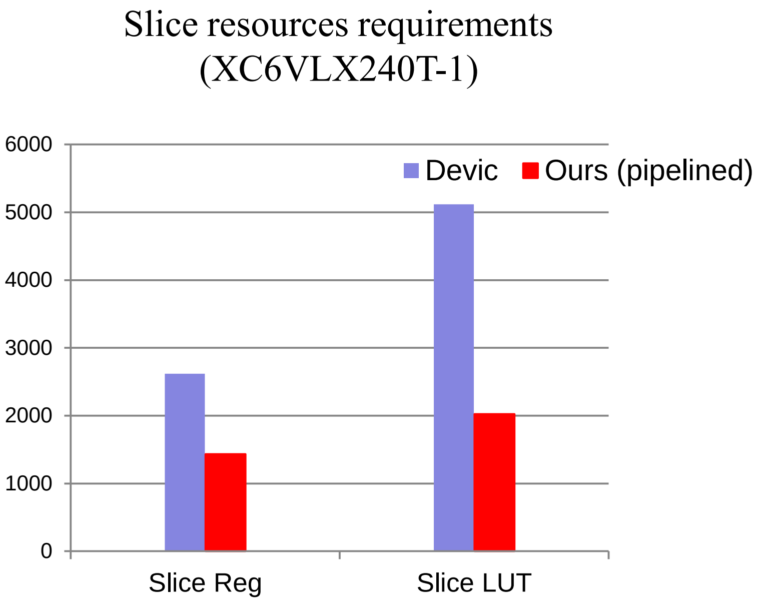




Throuhput (Gbps)

Comparison with the state of the art





Provided security mechanisms

Requirements	Built-in	Vliegen	Hori	Devic	Ours
Confidentiality	✓	✓	✓	✓	✓
Authentication	✓	✓	✓	✓	✓
Freshness	x	✓	x	✓	✓
Secure ext.Storage	✓	x	✓	✓	✓
Verify before configuration	x	✓	✓	x	✓
On-the-fly	x	x	✓	x	✓
Max. throughput (Mbps)	800	NA	913	23	2508

Outline

- Introduction & Motivation
  - ❑ Benefits and applications
  - ❑ FPGAs and reconfiguration
  - ❑ Attacks against configuration bitstreams
  - ❑ Technology support for bitstream security
  - ❑ State of the art in Secure dynamic reconfiguration
- Proposed solution: description and analysis
- Implementation
  - ❑ The cryptographic kernel
- Evaluation
  - ❑ Resource requirements and performance
  - ❑ Comparison with related state of the art
- **Conclusions**

Conclusions

The proposed systems allows to improve the security features:

- Prevents system downgrade and assures bitstream freshness
- On-the-fly detection of tampering attacks, before configuration
  - Prevents overwrite of the static region
- Secure usage of (large capacity) external memory

Performance and resource improvements:

- **1%** overall area increase, considering the core system of the target FPGA
  - Requires only 1% additional slices and 3% additional BRAMs
- Fast: **2.5Gbps**
  - only limited by the AES core
- 3 times faster than the built-in security mechanism of the Xilinx FPGAs
- Requires **45% less Slice** resources and **93 times faster**
  - Regarding the most secure state of the art (Devic et al.)

Conclusions

Future work

- Future improvements:
  - By using faster encryption/decryption cores,
    - to achieve a configuration throughput of 3.2 Gbps
      - limit of the configuration port.
  - SCA and Fault attack protection
  - The initialization problem
    - how to assure the correct initialization of the device
  - Evaluate differente technologies

Thank you!

Questions?

Email:

Ricardo.Chaves@inesc-id.pt

Project sources and implementation details can be accessed at:

http://sips.inesc-id.pt/~rjfc/cores/SecDR/

[1] Hirak Kashyap and Ricardo Chaves, “Secure partial dynamic reconfiguration with unsecured external memory", *24th International Conference on Field Programmable Logic and Applications (FPL 2014)*, September 2014.