FPGA security

Nele Mentens
nele.mentens@kuleuven.be

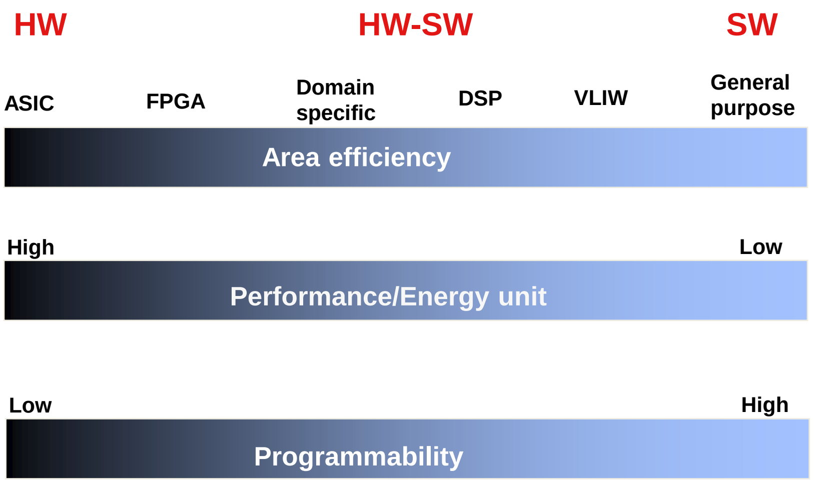
Design and security of cryptographic algorithms and devices
for real-world applications

June 1-6, 2014, Šibenik, Croatia

Outline

- Introduction
  - FPGA vs. ASIC
  - FPGA application
- FPGA technology
  - Architecture
  - Configuration
  - Design flow
  - Performance comparison
- Crypto on FPGA
  - Area and speed optimization
  - AES design examples
- Dynamic/partial reconfiguration

Introduction
FPGA vs. ASIC



HW

HW-SW

SW

ASIC

FPGA

Domain specific

DSP

VLIW

General purpose

Area efficiency

High

Low

Performance/Energy unit

Low

High

Programmability

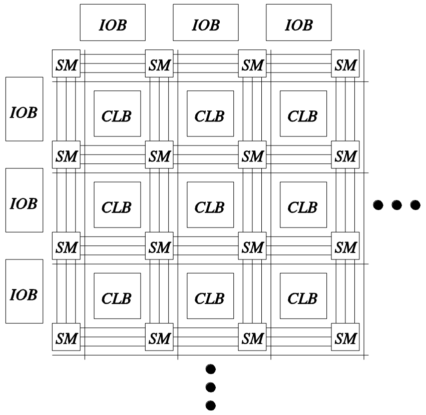
Introduction
FPGA vs. ASIC

- FPGA = Field-Programmable Gate Array
- ASIC = Application-Specific Integrated Circuit
- FPGA advantages over ASIC
  - faster time-to-market
  - smaller Non-Recurring Engineering (NRE) cost
  - programmable in the field
- ASIC advantages over FPGA
  - lower cost for high volumes
  - better performance

Introduction
FPGA application

- Prototype for ASIC design
- End product
  - Recently developed FPGAs are heterogeneous systems with dedicated building blocks.
  - FPGAs closely follow technology scaling because they are manufactured in high volumes.
- Application domains:
  - space
  - telecommunication
  - signal processing
  - …
- Many applications require data security on FPGA.

FPGA technology
Architecture

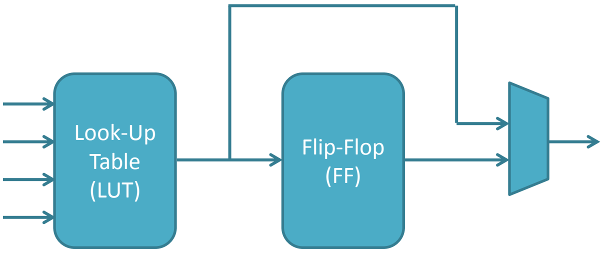


Basic FPGA architecture:

- CLB = Configurable Logic Block
  - CLBs consist of slices.
  - Slices consist of
    - Look-Up Tables (LUTs),
    - Multiplexers,
    - Flip-Flops (FFs),
    - Carry logic.
- SM = Switch Matrix
- IOB = Input/Output Block

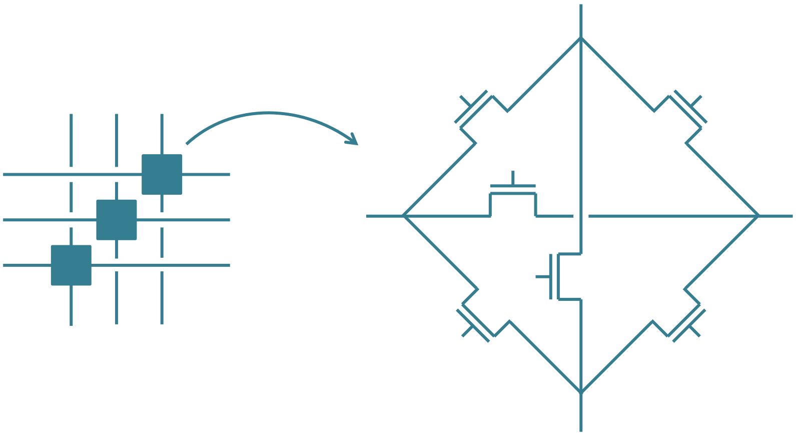
FPGA technology
Architecture

basic content of a slice (excluding carry-logic)



FPGA technology
Architecture

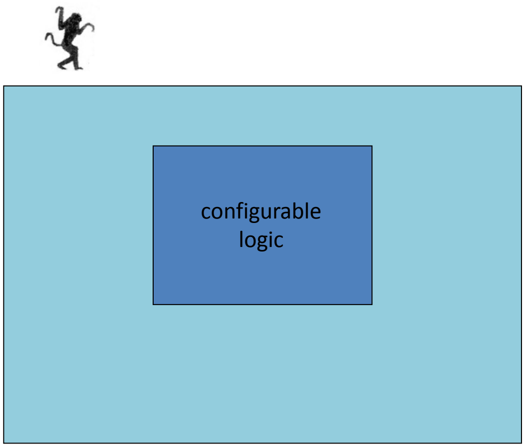
basic principle of a switch matrix



FPGA technology

Architecture

1991: XC4000



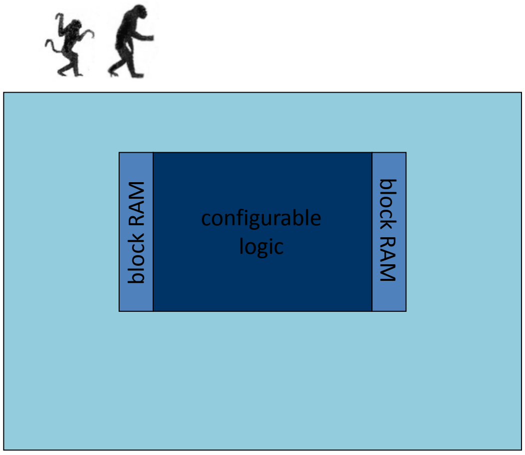
technology node:

0.25 µm

FPGA technology

Architecture

1991: XC4000
1998: Virtex

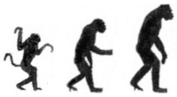


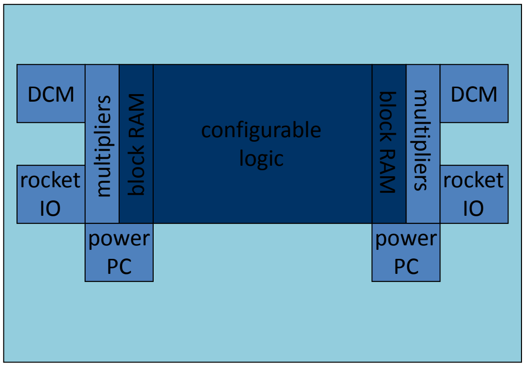
technology node:

0.22 µm

FPGA technology
Architecture

1991: XC4000
1998: Virtex
2002: Virtex-II Pro



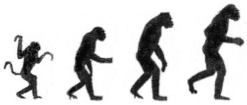


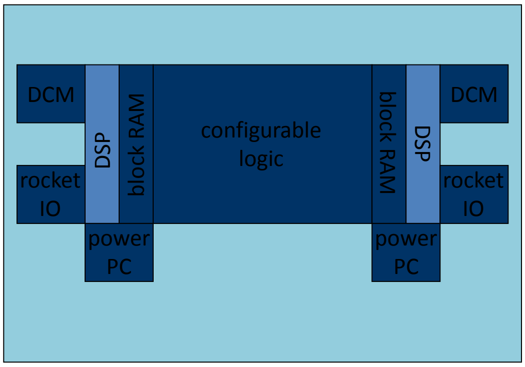
technology node:

0.13 µm

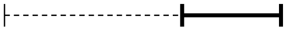
FPGA technology
Architecture

1991: XC4000
1998: Virtex
2002: Virtex-II Pro
2004: Virtex-4





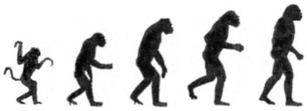
technology node:

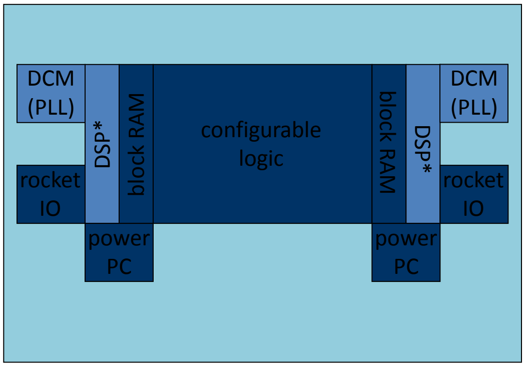


90 nm

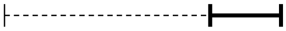
FPGA technology
Architecture

1991: XC4000
1998: Virtex
2002: Virtex-II Pro
2004: Virtex-4
2006: Virtex-5





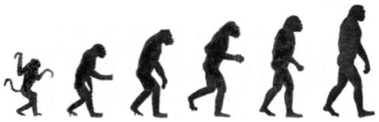
technology node:

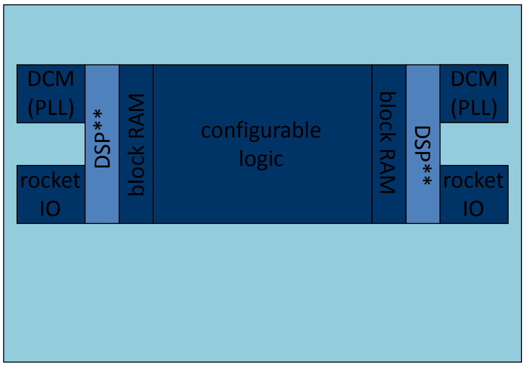


65 nm

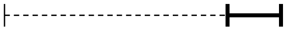
FPGA technology
Architecture

1991: XC4000
1998: Virtex
2002: Virtex-II Pro
2004: Virtex-4
2006: Virtex-5
2009: Virtex-6





technology node:



45 nm

FPGA technology
Architecture

1991: XC4000

1998: Virtex

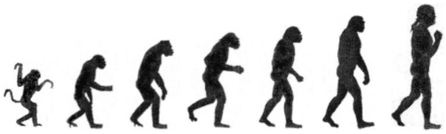
2002: Virtex-II Pro

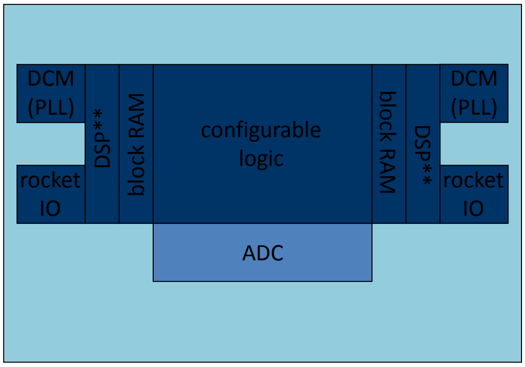
2004: Virtex-4

2006: Virtex-5

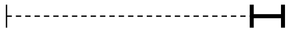
2009: Virtex-6

2010: Virtex-7





technology node:



28 nm

FPGA technology
Architecture

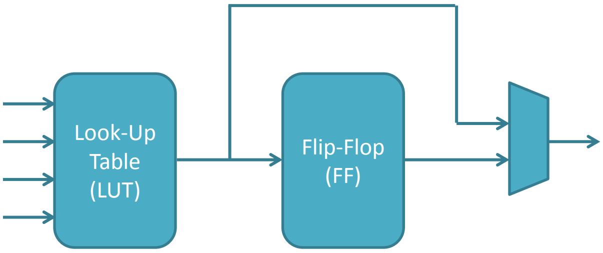
- Latest development of Xilinx FPGAs:
  - Zynq-7000 series
  - ARM + FPGA
  - Processor-centered architecture

FPGA technology
Configuration

- Configuration data: bitstream
- Configuration technology:
  - (anti-)fuse: one-time programmable
  - flash: non-volatile configuration memory
  - SRAM: volatile configuration memory
- SRAM (vs. flash) configuration memory
  - Higher density
  - Higher power consumption
  - On-board or on-chip non-volatile memory needed to store the bitstream during power-off
  - Higher configuration speed

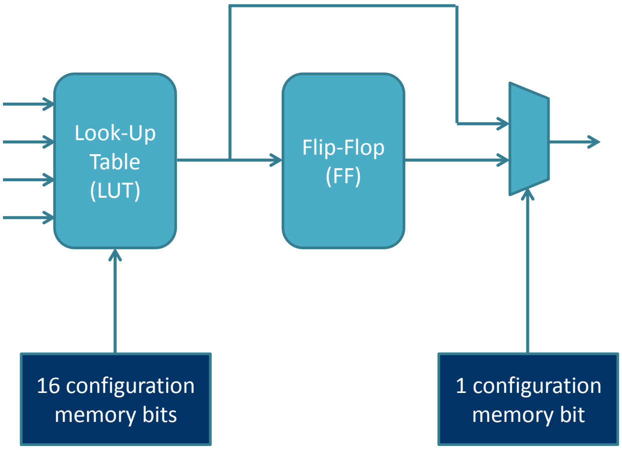
FPGA technology
Configuration

basic content of a slice (excluding carry logic)



FPGA technology
Configuration

basic content of a slice (excluding carry logic) + configuration



FPGA technology
Configuration

A	B	C	D	Z_0	Z_1	Z_2	Z_3	…	Z_{65280}	…	Z_{65535}
0	0	0	0	0	1	0	1		0		1
0	0	0	1	0	0	1	1		0		1
0	0	1	0	0	0	0	0		0		1
0	0	1	1	0	0	0	0		0		1
0	1	0	0	0	0	0	0		0		1
0	1	0	1	0	0	0	0		0		1
0	1	1	0	0	0	0	0		0		1
0	1	1	1	0	0	0	0		0		1
1	0	0	0	0	0	0	0		1		1
1	0	0	1	0	0	0	0		1		1
1	0	1	0	0	0	0	0		1		1
1	0	1	1	0	0	0	0		1		1
1	1	0	0	0	0	0	0		1		1
1	1	0	1	0	0	0	0		1		1
1	1	1	0	0	0	0	0		1		1
1	1	1	1	0	0	0	0		1		1

Why 16 configuration bits for a 4-to-1 LUT?

2^{16} possible output functions:

$Z_0 = 0$

$Z_1 = A'.B'.C'.D'$

$Z_2 = A'.B'.C'.D$

$Z_3 = A'.B'.C'$

…

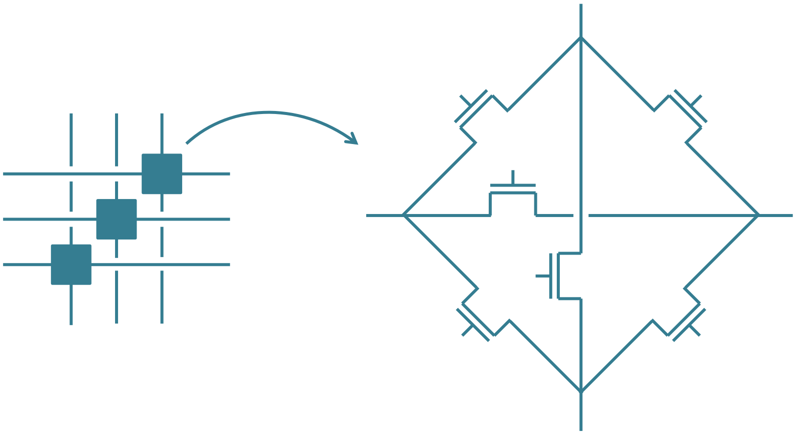
$Z_{65280} = A$

…

$Z_{65535} = 1$

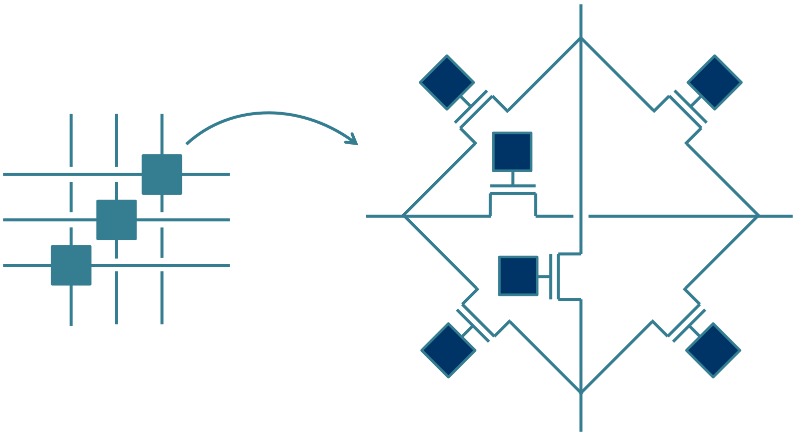
FPGA technology
Configuration

basic principle of a switch matrix



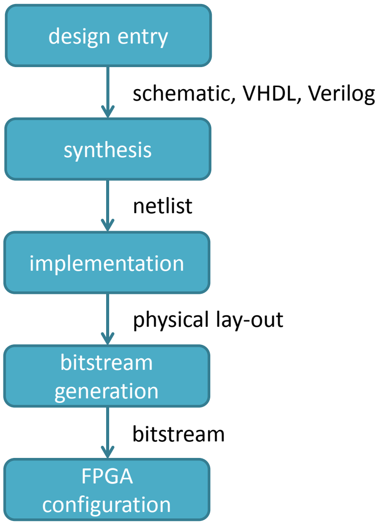
FPGA technology
Configuration

basic principle of a switch matrix + configuration



■ = 1 bit configuration memory

FPGA technology
Design flow


design entry

schematic, VHDL, Verilog

synthesis

netlist

implementation

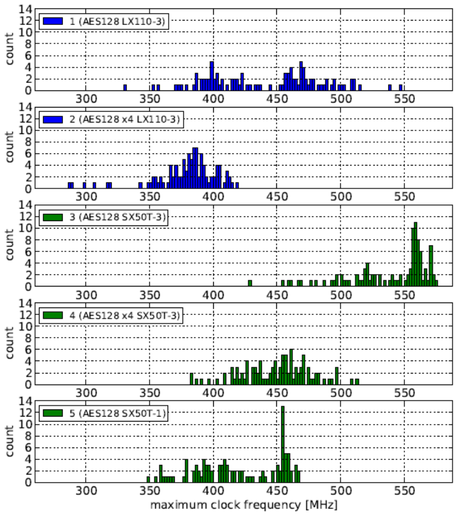
physical lay-out

bitstream generation

bitstream

FPGA configuration

FPGA technology
Performance comparison



- Be careful not to compare apples to oranges.
- Performance depends on:
  - the place & route seed,
  - the degree of occupation,
  - the speed grade of the device.
- Results from Saar Drimer’s Ph.D. dissertation

Crypto on FPGA
Area and speed optimization

- Maximize the use of dedicated building blocks
  - Multipliers (in older FPGAs)
    - A*B
    - with or without registers
  - DSP slices (in more recently developed FPGAs)
    - version 1: A * B + C
    - version 2: (A + B) * C + D
    - many options for including or excluding pipeline registers
  - Block RAM
    - single-port or dual-port
  - Shift registers
    - a LUT can also be used as an addressable shift register

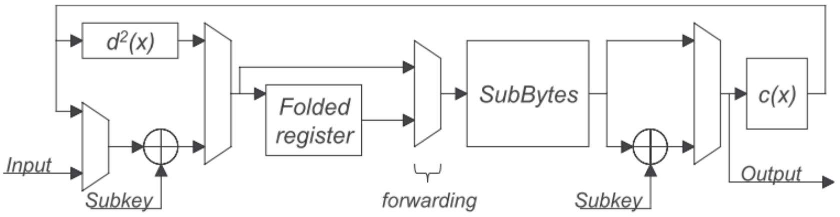
Crypto on FPGA
AES design examples

Two examples:

1. P. Chodowiec, and K. Gaj, “Very Compact FPGA Implementation of the AES Algorithm”, C.D. Walter et al. (Eds.): CHES 2003, LNCS 2779, pp. 319–333, 2003.
2. S. Drimer, T. Güneysu, and C. Paar, “DSPs, BRAMs and a pinch of logic: extended recipes for AES on FPGAs”, ACM Transactions on Reconfigurable Technology and Systems (TRETS), 3(1), 2010.

(pictures in the slides are copied from these publications)

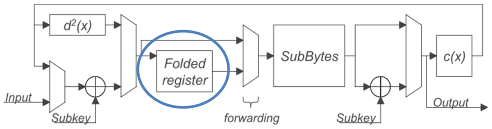
Crypto on FPGA
AES design example 1

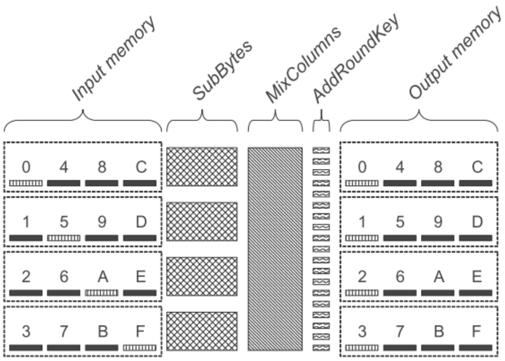


Encryption:	addroundkey shiftrows, subbytes, mixcolumns, addroundkey (execute 9 times) shiftrows, subbytes, addroundkey
Decryption:	addroundkey invshiftrows, invsubbytes, addroundkey, invmixcolumns (execute 9 times) invshiftrows, invsubbytes, addroundkey

Summer School, Šibenik, Croatia – June 1-6, 2014

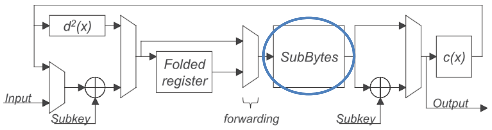
Crypto on FPGA
AES design example 1

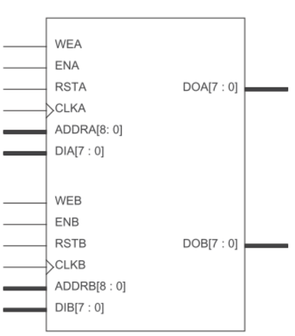


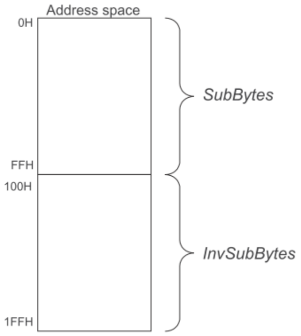


Summer School, Šibenik, Croatia – June 1-6, 2014

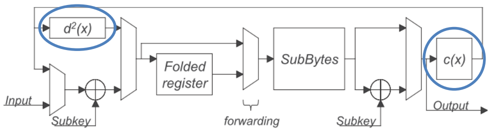
Crypto on FPGA
AES design example 1

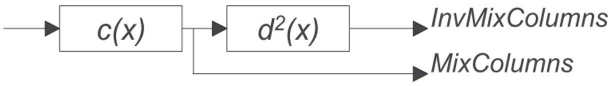




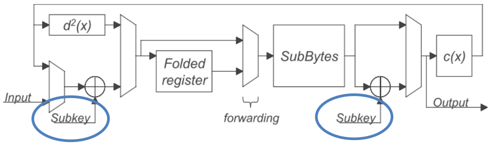


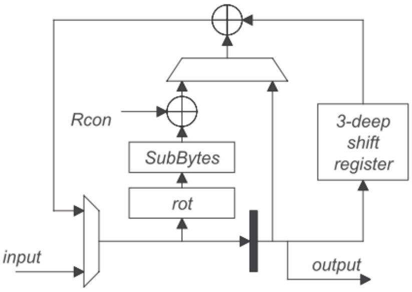
Crypto on FPGA
AES design example 1



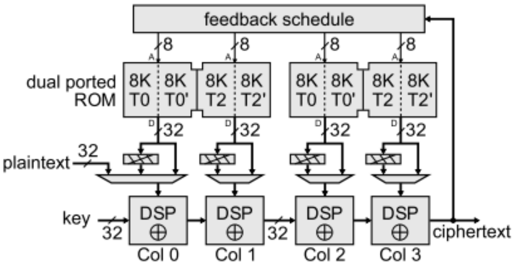


Crypto on FPGA
AES design example 1





Crypto on FPGA
AES design example 2



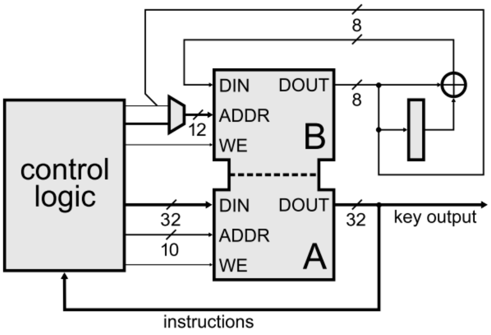
round functions

$$T_0[x] = \begin{bmatrix} S[x] \times 02 \\ S[x] \\ S[x] \\ S[x] \times 03 \end{bmatrix} \quad T_1[x] = \begin{bmatrix} S[x] \times 03 \\ S[x] \times 02 \\ S[x] \\ S[x] \end{bmatrix}$$

$$T_2[x] = \begin{bmatrix} S[x] \\ S[x] \times 03 \\ S[x] \times 02 \\ S[x] \end{bmatrix} \quad T_3[x] = \begin{bmatrix} S[x] \\ S[x] \\ S[x] \times 03 \\ S[x] \times 02 \end{bmatrix}$$

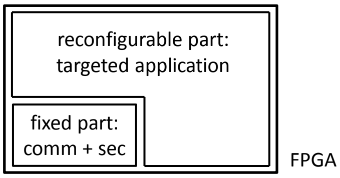
Crypto on FPGA
AES design example 2

key schedule



Dynamic/partial configuration

- possible in SRAM-based FPGAs,
- facilitates:
  - secure remote configuration,



  - IP core licensing,
  - implementation attack resistance.