Covert Communications in Heterogeneous SoCs

Carlos Andres Lara-Nino, Lilian Bossuet

Université Jean Monnet Saint-Étienne

cnrs

Laboratoire Hubert Curien
UMR • CNRS • 5516 • Saint-Étienne

fetch 2024

Communications cachées dans des SoC hétérogènes

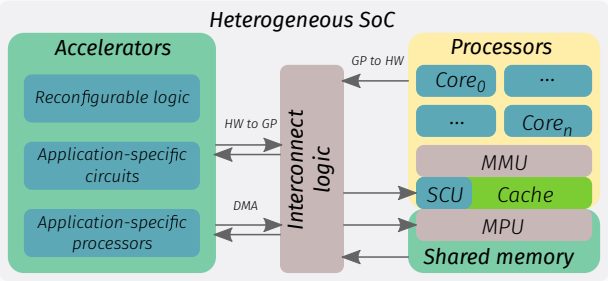
Carlos Andres LARA-NINO

École d'hiver Francophone sur les Technologies de Conception des Systèmes Embarqués Hétérogènes (FETCH)
Maillen, Belgique
Février 7-9, 2024

Université Jean Monnet Saint-Etienne, CNRS, Laboratoire Hubert Curien
UMR 5516, F-42023, SAINT-ETIENNE, France

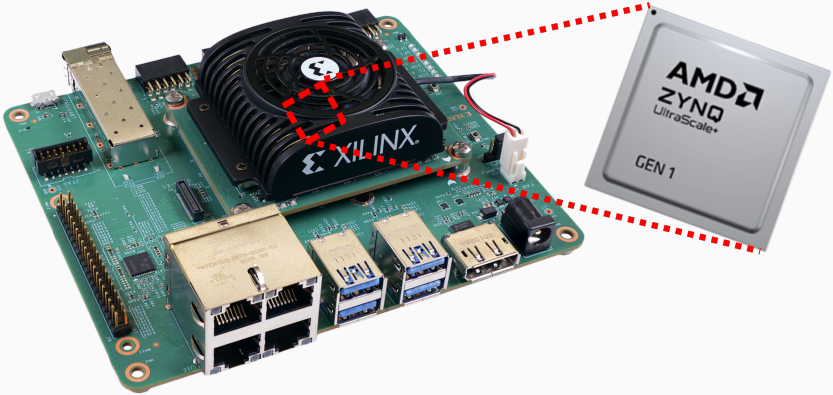
FRANCE 2030

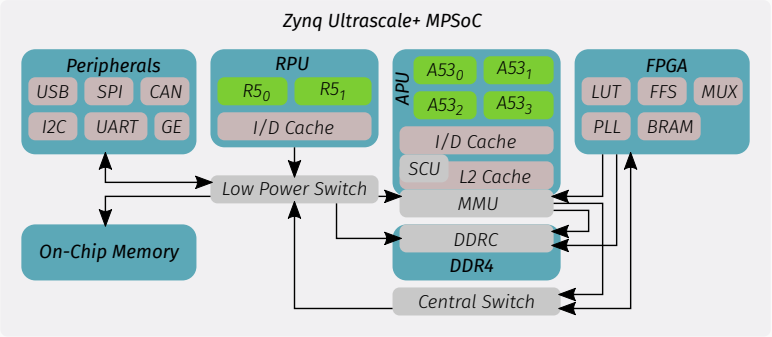
System(s)-on-a-Chip



A system-on-a-chip (SoC) is a device which integrates multiple processing elements in the same die. An heterogeneous SoC may feature different processors, hardware accelerators, and even an FPGA.

Case study: AMD-Xilinx MPSoC




Zynq Ultrascale+ MPSoC
Peripherals
USB
SPI
CAN
I2C
UART
GE
RPU
R5$_0$
R5$_1$
I/D Cache
APU
A53$_0$
A53$_1$
A53$_2$
A53$_3$
I/D Cache
SCU
L2 Cache
MMU
FPGA
LUT
FFS
MUX
PLL
BRAM
Low Power Switch
On-Chip Memory
DDRC
DDR4
Central Switch

Intrinsic channels

The shared resources of the SoC may be leveraged to covertly transfer information.

Attack goals

- Bypass isolation policies
- Change the nature of the channel[1]
- Enable more complex attacks[2]

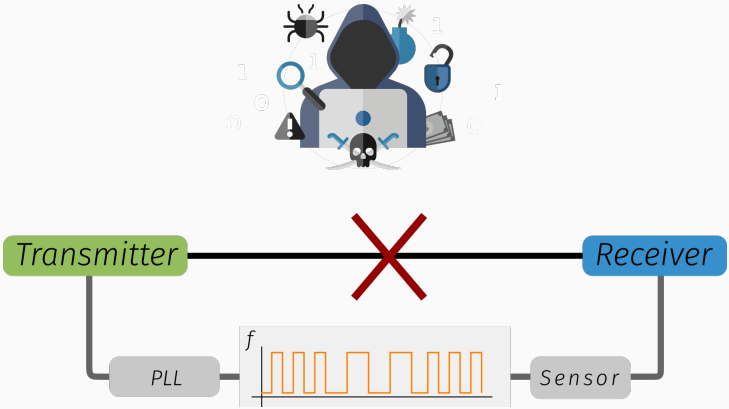
[1] Benhani, E.M. & Bossuet, L. (2018). DVFS as a security failure of TrustZone-enabled heterogeneous SoC. In *25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (pp. 489-492). IEEE.

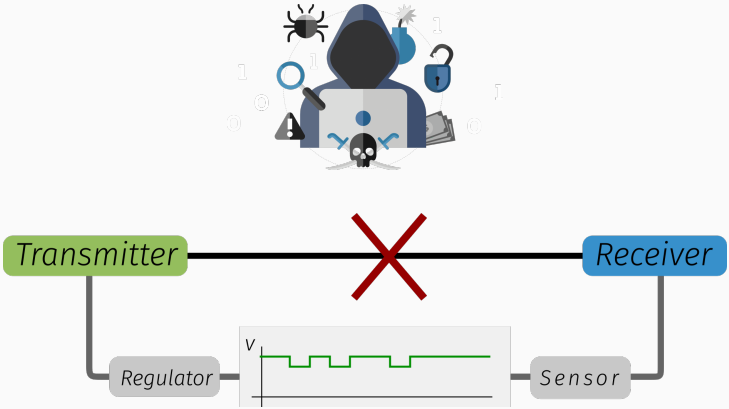
[2] Fellah-Touta, A., Bossuet, L., & Lara-Nino, C. A. (2023). Combined Internal Attacks on SoC-FPGAs. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 281-286). IEEE.

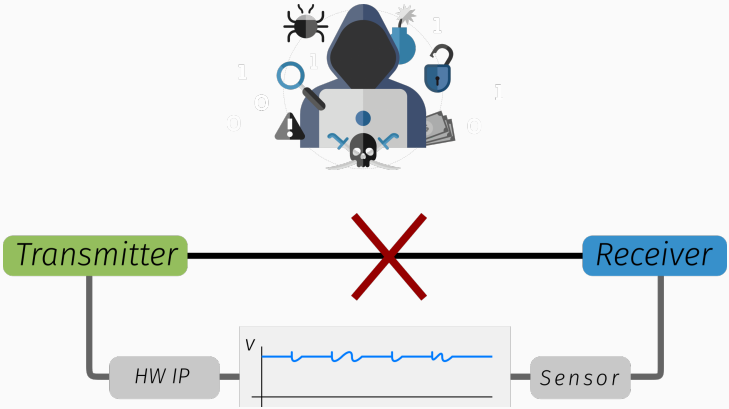
Require: k, an encryption key
Require: p, a plaintext to be encrypted
`covert_bit_set()` {Pull trigger}
AES(p, k)
`covert_bit_clear()` {Release trigger}

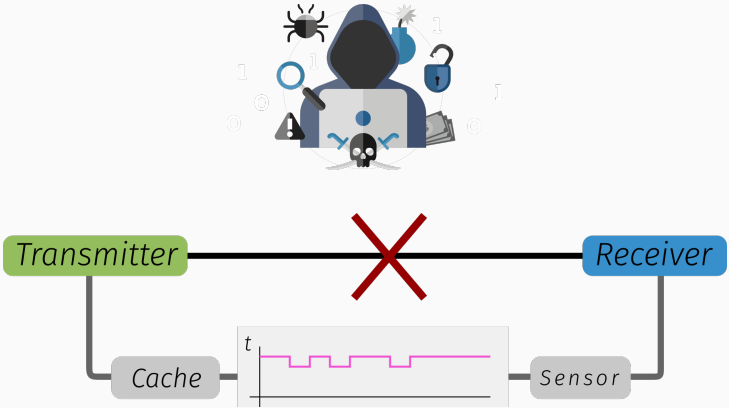
Require: k, an encryption key
Require: p, a plaintext to be encrypted
Require: f, f', two core frequencies
cpu_freq $\leftarrow f'$ {Pull trigger}
AES(p, k)
cpu_freq $\leftarrow f$ {Release trigger}

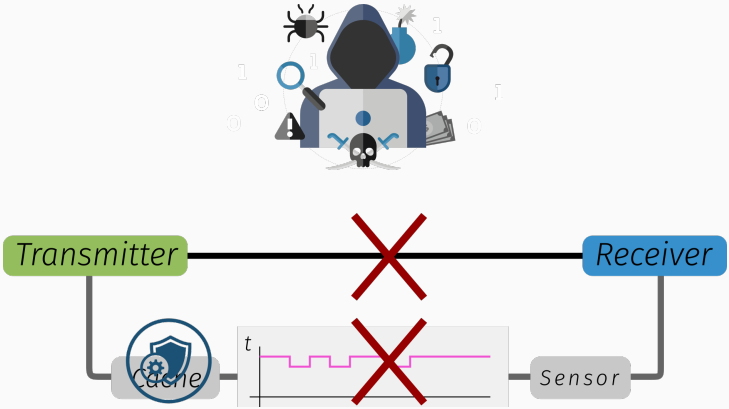

Transmitter
Receiver


Transmitter
Receiver
PLL
f
Sensor


Transmitter
Receiver
Regulator
V
Sensor


Transmitter
Receiver
HW IP
V
Sensor


Transmitter
Receiver
Cache
t
Sensor


Transmitter
Receiver
Cache
t
Sensor

Challenges

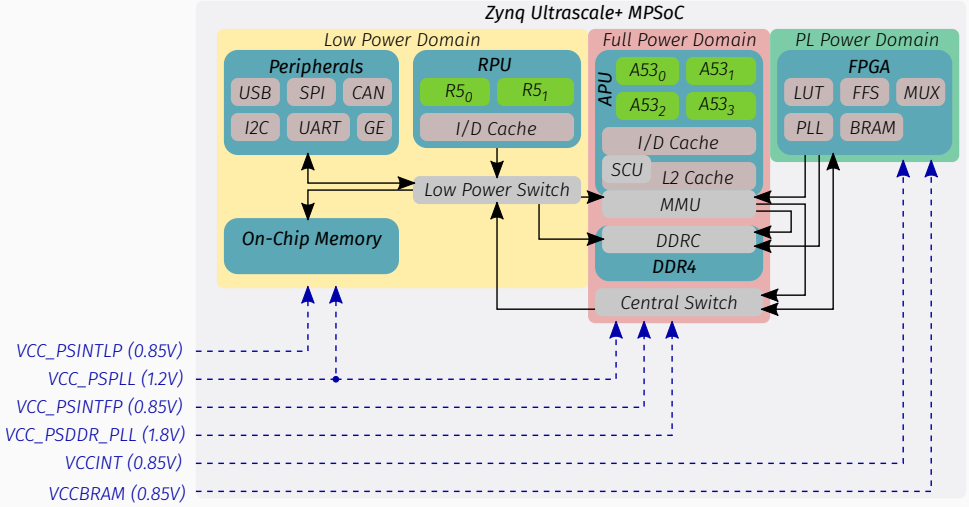
- Many intrinsic channels available
- Implementation strategies change from one platform to another
- Mitigating the attack, once discovered, is complicated

Potential approaches

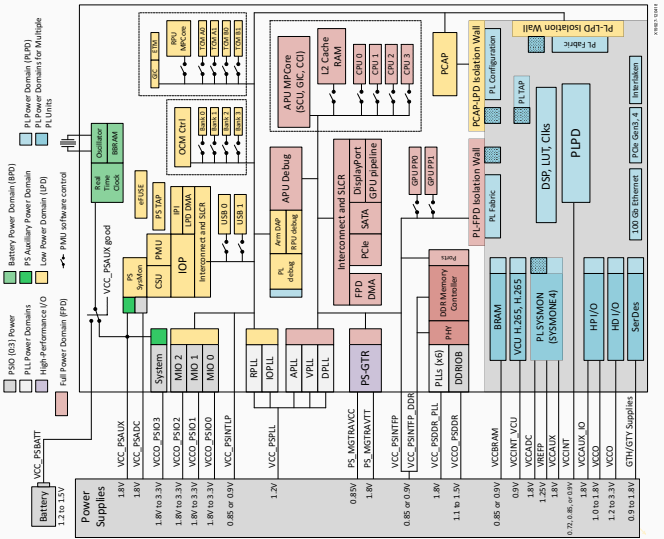
- Conventional study of architectural vulnerabilities
- Implementation of stricter isolation rules
- Use of pre-silicon tooling for the study of intrinsic channels

Case studies

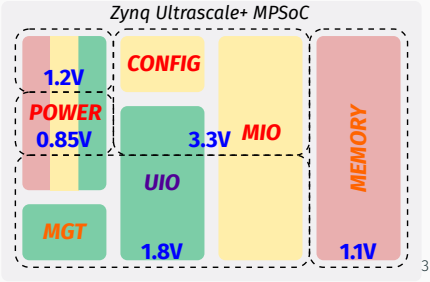
Case study: PDN intrinsic channels



Case study: PDN intrinsic channels



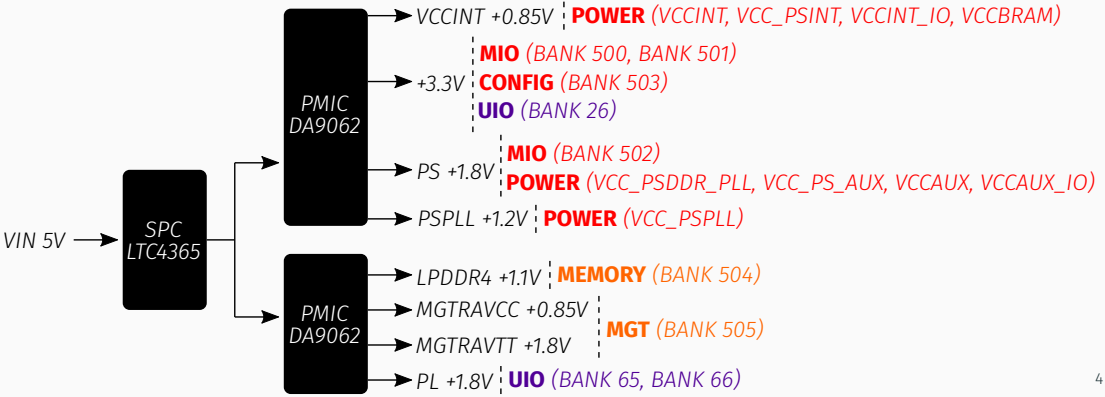
Case study: PDN intrinsic channels



[3]

[3] https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Development_Boards/TE0802/REV02/Documents

Case study: PDN intrinsic channels



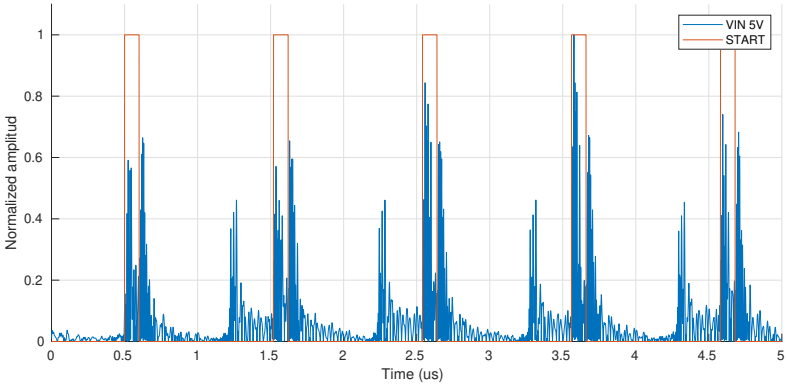
[4]

[4] https://shop.trenz-electronic.de/Download/?path=Trenz_Electronic/Development_Boards/TE0802/REV02/Documents

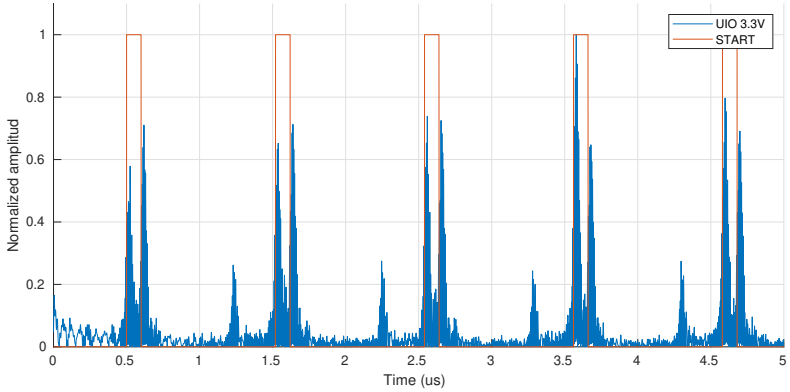
Considerations

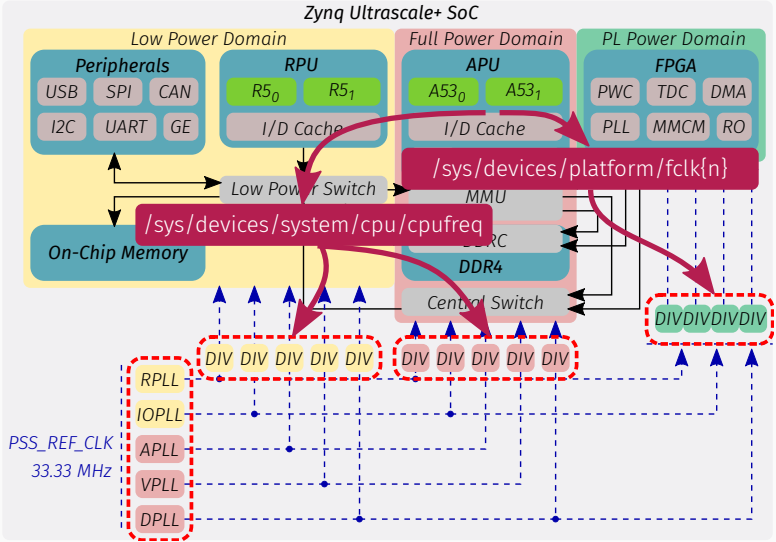
- Transmitter: Hardware accelerator in the FPGA
- Receiver: External
- Shared resource: V_PL

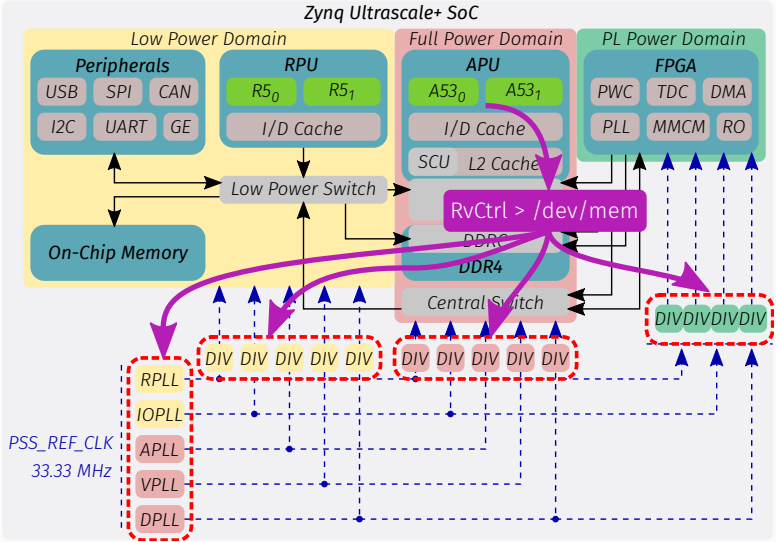
PDN intrinsic channel



PDN intrinsic channel



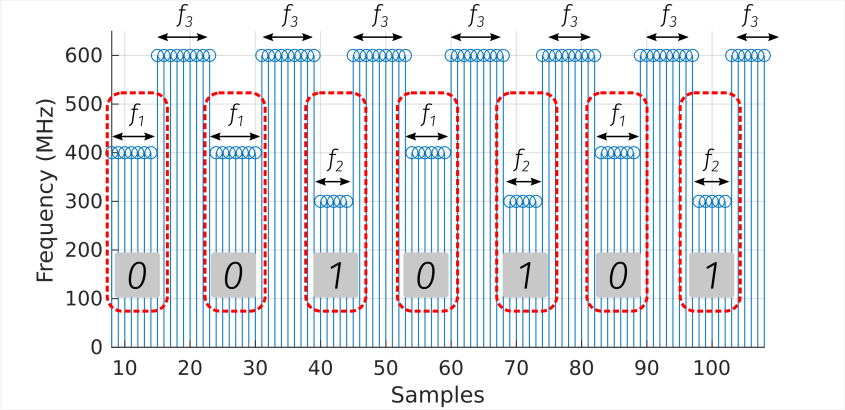

Zynq Ultrascale+ SoC
Low Power Domain
Full Power Domain
PL Power Domain
Peripherals
USB
SPI
CAN
I2C
UART
GE
RPU
R5$_0$
R5$_1$
I/D Cache
APU
A53$_0$
A53$_1$
I/D Cache
FPGA
PWC
TDC
DMA
PLL
MMCM
RO
/sys/devices/platform/fclk{n}
Low Power Switch
MMU
/sys/devices/system/cpu/cpufreq
On-Chip Memory
DDR4
Central Switch
DIV DIV DIV DIV DIV
DIV DIV DIV DIV DIV
DIV DIV DIV DIV
RPLL
IOPLL
APLL
VPLL
DPLL
PSS_REF_CLK
33.33 MHz


Zynq Ultrascale+ SoC
Low Power Domain
Full Power Domain
PL Power Domain
Peripherals
USB
SPI
CAN
I2C
UART
GE
RPU
R5$_0$
R5$_1$
I/D Cache
APU
A53$_0$
A53$_1$
I/D Cache
SCU
L2 Cache
FPGA
PWC
TDC
DMA
PLL
MMCM
RO
Low Power Switch
RvCtrl > /dev/mem
On-Chip Memory
DDRC
DDR4
Central Switch
DIV DIV DIV DIV DIV
DIV DIV DIV DIV DIV
DIVDIVDIVDIV
RPLL
IOPLL
APLL
VPLL
DPLL
PSS_REF_CLK
33.33 MHz

Considerations

- Transmitter: User application coded in C
- Receiver: User application coded in C
- Shared resource: `cpufreq` driver

DVFS intrinsic channel



Final remarks

- Intrinsic channels have niche applications but are difficult to mitigate
- Their transmission rate is low, but the volume of information required is not high
- We need to rethink the design of SoC platforms to introduce more effective isolation strategies

This work has been supported by the French government in the framework of the *France 2030* initiative under project ARSENE (ANR-22-PECY-0004).

Thanks !