MUNI
FI

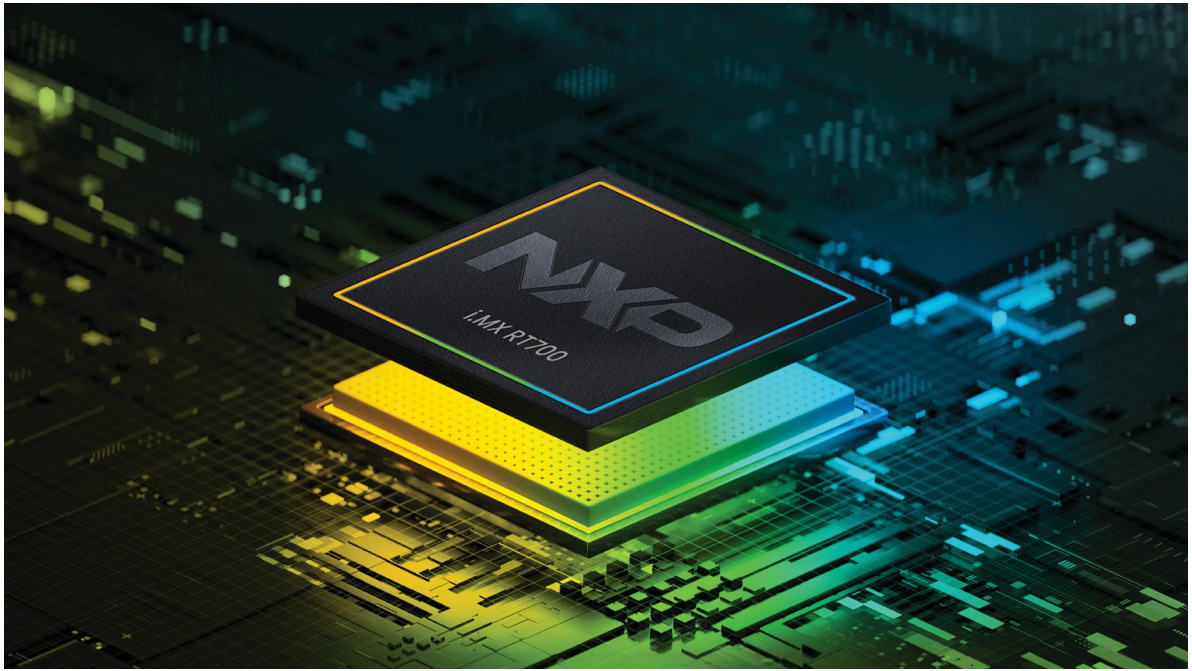
PA192: Communication Buses & Safe Lock Discussion

Jan Kral

jan.kral@fi.muni.cz

Invite

- Topic: **Security in modern NXP MCUs**
- When: 11.11.2025, 14:00 – 15:50
- Where: A415
- Who:
  Pavel Grasblum



Communication Busses

- Risks:
  - Why are there UART, SPI, I2C on PCBs?
  - What risks arise if an attacker locates these pins?
  - Could an attacker modify the traffic? Consequences?
- Countermeasures:
  - How can a designer hide these interfaces?
  - Which protections can be used?
  - Impact on the price of the final product?

Safe Lock – side channel attacks

- Principles:
  - Why does execution time reveal information?
  - What does higher consumption usually mean?
- Countermeasures:
  - How can a designer protect systems against side channel attacks?
  - Which protections can be used?
- Impact on the repair-ability and servicing?

Legal Obligations in Device Design

- CE Marking and Conformity
- Cyber Resilience Act
  - What is it and its implications?
  - What is SBOM?
- Data protection and Privacy

Project Topics

- Your own projects are encouraged
- Work in pairs is allowed (with reasonable difficulty)
- Describe the project assignment and your goals (in bullet points) in markdown format (project.md).
- Discuss the assignment.
- https://docs.google.com/document/d/1pu_B5ZH62m3DgipfpenqILvISzifUZ9iwRJsM8RdRfE/edit?usp=sharing

MUNI
FI

PA192: JTAG and SWD

Jan Kral

jan.kral@fi.muni.cz

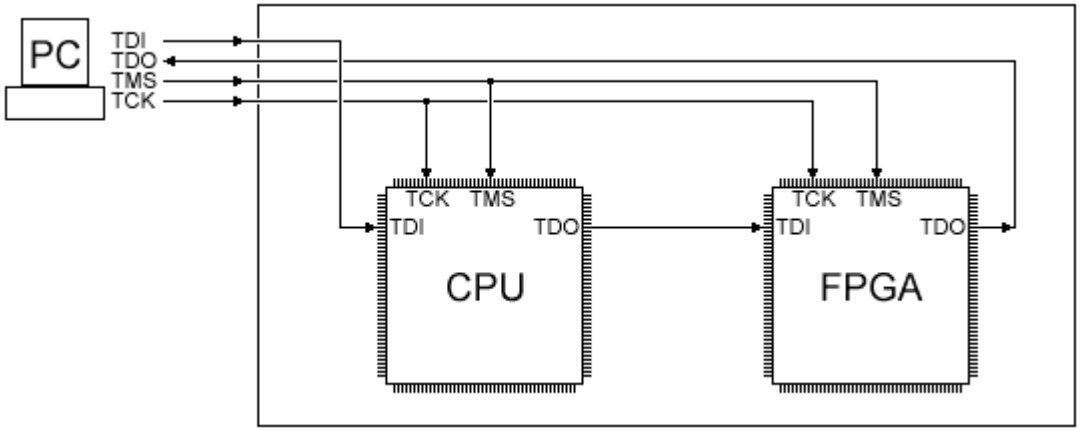
JTAG

- Joint Test Action Group
- Originally designed for board-level testing (boundary scan)
- Later for firmware programming and real-time debugging
  - MCUs, FPGAs
  - Halt or step CPU
  - Read/write memory and registers
  - Access peripherals directly
- If left unprotected, it is effectively a **hardware backdoor**.

SWD

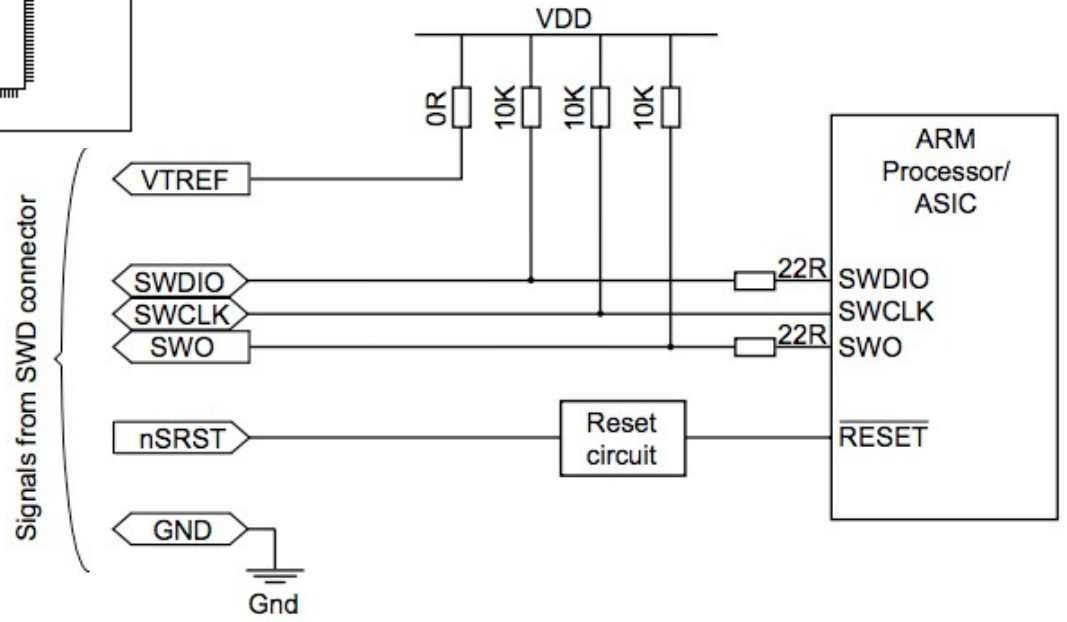
- Serial Wire Debug
- ARM ecosystem only
- Compared to JTAG:
  - Strictly for point-to-point communication
  - Packet based communication
  - More efficient for memory/register writes/reads
  - Support direct memory access
  - SWD is faster and simpler FW development
  - JTAG is more flexible for testing complex boards
- SWD adds optional Secure Debug Authentication

JTAG vs SWD



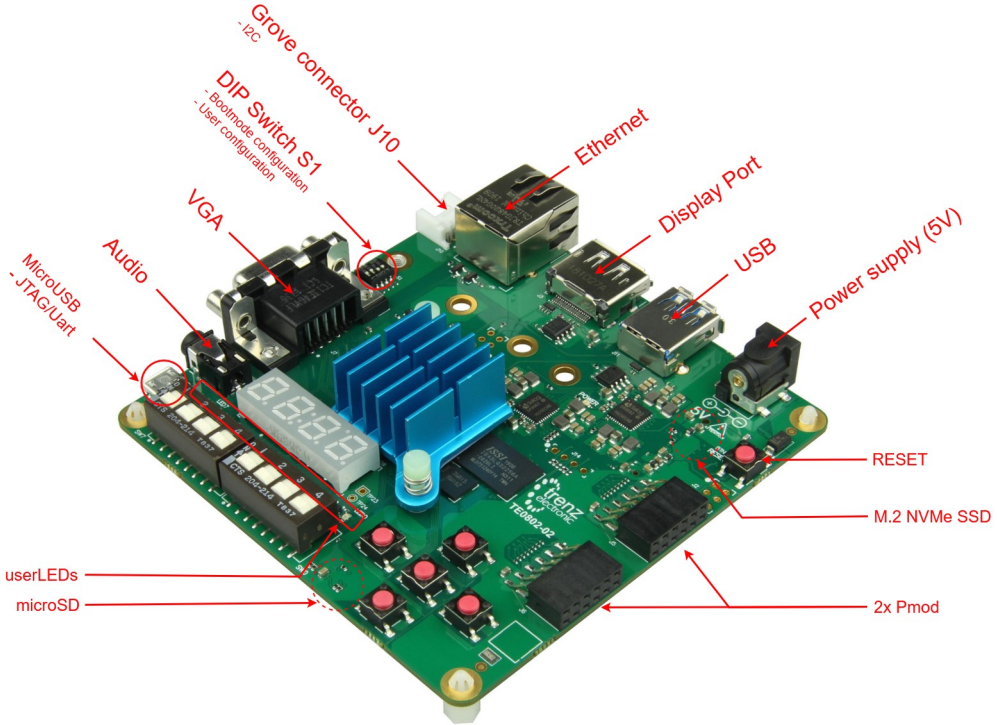
JTAG

SWD



JTAG and SoC Platform

- System on Chip (SoC) – Zynq Ultrascale+
- JTAG and UART are available on MicroUSB connector



Task 1

- Run JlinkExe
  - Download: https://www.segger.com/downloads/jlink/
- Connect to you board
  - cmd: connect
  - Select your device/core (find target IC label)
  - SWD is the target interface
  - Use modern tools to understand the text output of the connect phase.
- Reference: https://www.thyrasec.com/blog/hackers-guide-to-j-link-debuggers/

Task 2

- Examine available commands
  - cmd: ?
- Halt CPU
  - Examine and understand the text output
- Resume CPU
- Halt CPU
  - Examine and understand the text output
  - What are the differences to the previous output?

Task 3

- Determine the Flash and RAM memory base address and size.
  - Look at the device datasheet and reference manual.
- Read the whole flash memory into flash.bin file.
- Read the whole RAM memory into ram.bin file.
- Scan both files for strings
  - ASCII strings
    - `strings -a -n 4 -t x firmware.bin | less`
  - UTF-16LE strings
    - `strings -a -e l -n 4 -t x firmware.bin | less`
- Which strings are interesting for you in Flash and RAM?

Submission

- Download w07_results.txt
- Fill in all the results that you have found
- Remember to fill in:
  - Your student IDs (uco)
- Submit your file into:

  https://is.muni.cz/auth/el/fi/podzim2025/PA192/ode/182489950/w07_swd/