SIP WS 2022
Project 2

1 Organization

- Group size: 3
- Deadline 2a: 09/12 23:59
- Deadline 2b: 25/01 23:59
- git-Repositories: `git.teaching.iaik.tugraz.at` (will be sent out via E-mail)

1.1 Where to ask questions

- In our weekly meetings
- Discord (`https://discord.com/invite/k2Njmev`)
- E-Mail: `mailto:sip-team@iaik.tugraz.at`

2 Project 2

For this project, use the template repository for your board:

- `https://extgit.iaik.tugraz.at/sip/zybo_base_design`
- `https://extgit.iaik.tugraz.at/sip/zybo_z7_base_design`

For Project 2, you are tasked with developing a system which decrypts and displays a secret image onto the screen. This project will be split into two parts with separate deadlines. The goal of project 2a is to develop an `ASCON` cryptographic accelerator which supports the AXI protocol and accompanying device drivers. The goal of project 2b is to integrate the accelerator with the video pipeline (already included in the template repository) and display an encrypted image via HDMI. Below is the breakdown of the minimum requirements to complete both parts.

2.1 Project 2a Requirements

- Develop **your own AXI IP core** which can encrypt/decrypt with the `ASCON` cipher. The source code for the `ASCON` CryptoCore will be provided, however you must adapt it to use the AXI interface.
- **Create a VHDL or (System-)Verilog testbench** for your AXI IP core.
- Build and deploy Linux onto your board using Buildroot (recommended) or Yocto. Modify the device tree as needed and **write a driver** for the AXI IP mentioned above.
- **Provide a C example** showing how to use your AXI IP and driver from Linux.

2.2 Project 2b Requirements

- Your Linux host should **deploy a webserver** which allows the user to upload an encrypted image to be displayed.
- Your system must be able to **display the decrypted image** via HDMI.
- **Write a short (one paragraph) description** of your system, optionally include a diagram.

You are encouraged to expand and extend the above list with additional requirements to enhance the system. For instance, you may choose to design your system in a way which protects the memory holding the decrypted image from the Operating System, or decrypt a video stream.

3 Submission

1. Export your block design within Vivado: write_bd_tcl –force bd.tcl
2. Commit bd.tcl and your constraints file (base.xdc)
3. Commit your custom IP core
4. Try to recreate the project using the template
   - Zybo 7000: https://extgit.iaik.tugraz.at/sip2020/zybo_base_design/-/blob/master/HW/project.tcl
   - Zybo Z7: https://extgit.iaik.tugraz.at/sip2020/zybo_z7_base_design/-/blob/master/HW/project.tcl
   - Be aware of the todos in the file!
   - Adapt if necessary.
5. Commit all the relevant software (device driver, bare metal program, ...)
6. Add a readme including:
   - Where to find which software files
   - Any other relevant information
7. Tag your submission:

```
git tag Project2
git push --tags
```

4 Useful Links

- Zynq Book http://www.zynqbook.com/
- Zybo Reference Manual http://www.digilentinc.com/Data/Products/ZYBO/ZYBO_RM_B_V6.pdf
- Zybo Embedded Linux Hands-on Tutorial http://www.farnell.com/datasheets/1904568.pdf
- Zybo Embedded Linux Hands-on Tutorial #2 https://www.instructables.com/Embedded-Linux-Tuto

- Tutorial: Create a custom IP core https://reference.digilentinc.com/learn/programmable-logic/tutorials/zybo-creating-custom-ip-cores/start
- Linux GPIO Drivers on Zynq https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842398/Linux+GPIO+Driver
- AXI4 Stream Protocol Specification https://developer.arm.com/documentation/ihi0051/a
- LWC Hardware Protocol Specification used by Ascon https://cryptography.gmu.edu/athena/LWC/LWC_HW_API.pdf
- AXI DMA Product Guide https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf
- AXI DMA Example Code https://github.com/Xilinx/embeddedsw/tree/master/XilinxProcessorIP drivers/axidma/examples
- Xilinx Digital Video and VDMA tutorial https://forums.xilinx.com/t5/Video-and-Audio/Xilinx-Video-Series/td-p/849583