Project Specification

AKER: Safe and Secure SoC Access Control

Chi Chow, Brandon Erickson, Hosein Yavarzadeh

Summary

This document contains the project specifications. It begins by providing an overview of the project. In the second section, the key milestones of the project are described. Following that, we discuss the MVP, group management, project development, and schedule.

Project Overview

Modern System on a Chip (SoC) architectures can consist of many different resources, including processors, hardware accelerators, and IO. To handle the modularity of these SoC architectures, on-chip networks are used to communicate effectively between resources. Of course, sharing resources safely and securely is also critical. This prompts the use of access control systems that define the permissions and abilities of an SoC controller. However, it is challenging to implement secure SoC access control systems. Hardware vulnerabilities are hard to patch, often requiring a firmware rewrite, or even remanufacturing the chip.

This problem motivates the existence of AKER – a framework for the development of safe and secure on-chip access control systems targeting the requirements of modern safety and security critical applications. AKER utilizes the ACW (Access Control Wrapper) which is a high-performance and easy-to-integrate hardware module that dynamically manages access to shared resources. ACWs are used to wrap controller IP cores and to act as local access controllers. ACWs are also managed by a Trust Entity (TE). All this structure makes sure that the only legal requests are transmitted to the AXI interconnect, and thus the shared resources will only receive the legal requests.

This Project is going to implement the AKER access control system on the ESP platform. ESP is an open source platform for heterogeneous system-on-chip design that combines a tile-based architecture and a flexible system level design methodology. There are three accelerator flows available with ESP: RTL, high-level synthesis (HLS), and machine learning. In all three design flows, the automated SoC integration flow of the ESP provides the necessary hardware and software interfaces for rapid full-system prototyping on FPGA. In this project we will implement the **simplified** AKER access control on top of the ESP platform.

Project Approach

For this project, the AKER access control system will be implemented onto ESP as the target platform. ESP is an open-source platform developed by Columbia University for heterogeneous System-on-Chip (SoC) design. It provides an automated SoC integration flow that facilitates RTL, HLS, and machine learning accelerator flows. The streamlined hardware and software interfaces ESP provides allows for rapid development and prototyping of SoCs on FPGA.

We will implement the AKER access control system onto ESP by dividing our project into four major milestones:

1. ESP Project Setup: An ESP architecture will be developed using RTL (SystemVerilog) for integration with the AKER access control system. Examples for designing an accelerator and SoC using RTL are available within the documentation of the ESP platform. Using these examples, we expect to create a functional ESP architecture within one to two weeks.

2. AKER Integration: This milestone includes the RTL development of the AKER access control module with the access control wrapper and trusted entity features, and the integration of the AKER module with the ESP architecture. Four weeks are allocated for the completion of this milestone, and the milestone is considered complete upon presentation of a functional ESP SoC architecture with AKER access control – our minimum viable product.

3. Testing and Validation on the Xilinx Zynq Ultrascale+ Platform: This milestone includes testing and analysis of the performance and security of the modified ESP architecture. Evaluation will be conducted using results from simulation and execution of performance benchmarks. Testing of the final implementation is expected to conclude in the following two weeks.

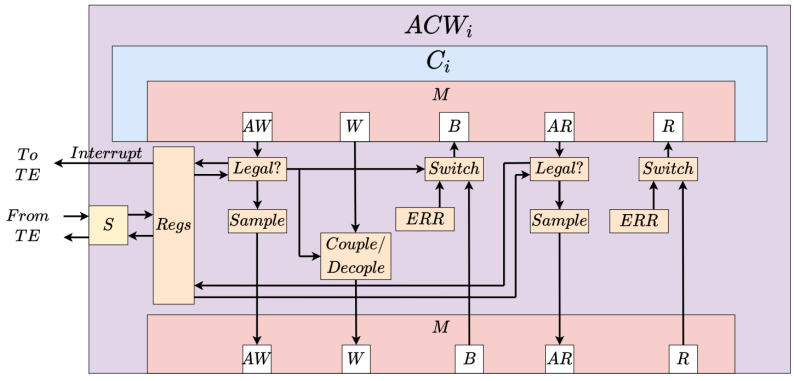
4. Cross-Comparison of Access Control Systems: If sufficient time remains after the completion of the previous milestones, a comparison of the performance and security of the AKER module will be compared against state-of-the-art SoC access control systems (e.g. INTC, XMPU) in the final presentation and report.

For more information on the milestones above, see the section **"Project Milestones and Schedule."**

Minimum Viable Product (MVP)

The minimum goal of this project is to implement the basic features of AKER proposed in the "AKER: A Design and Verification Framework for Safe and Secure SoC Access Control" research paper. The minimum viable product should include these components: implementation of the Access Control Wrapper and Trusted Entity, runnable tests on the Xilinx Zynq Ultrascale+ platform, and identification of common weakness enumerations (CWEs) that are mitigated through the implementation of AKER. The details of each component are as follows:

1. Access Control Wrapper: the ACW is a configurable access control module designed to monitor an AXI-compliant controller. In an AXI-compliant SoC architecture (without AKER), a controller exports a manager (M) interface that connects to a subordinate (S) interface in the interconnect. With the addition of AKER, there is a layer of access control between the M interface of the controller and the S interface of the interconnect. The access control wrapper encapsulates the functionality of a controller, and exports its own M interface that connects to the S interface of an interconnect. The wrapper allows us to define permissions to access the interconnect. This is the fundamental concept behind AKER, and it is necessary to implement the access control system as stated in the paper.



2. Trusted Entity: ACWs are controlled by the trusted entity (TE). The TE can configure ACWs to allow or deny requests from any controller to the interconnect. After a configuration is defined, the TE can allow legal requests, and can detect and block illegal requests. Correctly implementing the TE allows us to define access policies on the ACWs dynamically.

3. Testing on the Xilinx Zynq Ultrascale+ platform: development of the ACWs and TE is closely related to testing against a real life SoC architecture. To show that the implementation is correct, we will run our implementation on the Xilinx Zynq Ultrascale+ platform. Tests will involve performing legal and illegal accesses, to verify the ability to ensure safe access to on-chip resources.

4. Identification of mitigated CWEs: during the implementation of AKER, there should be a benchmark to evaluate how good our solution is. AKER identifies 30 security weaknesses from the MITRE common weakness enumeration database. By the end of the project, there needs to be an analysis of how effective our implementation is in solving these weaknesses.

Group Management

Outside of communication from our mentor, Francesco Restuccia, our group will make decisions on project planning and organization by consensus. Our group will use a Slack channel as its primary medium of communication, and virtual meetings will be conducted weekly (or semi-weekly as needed) outside of normal class time. During these meetings, we will assign short-term milestones for each member to complete within the following two weeks. We will also conduct a retrospective on previous milestones that may not have been resolved successfully. Through these meetings, we will address scheduling conflicts and other challenges to the development of our project.

Project Development

During this project's development, the team will be divided so that the required modules may be implemented concurrently. These required modules include the ESP SoC architecture, its Network-on-Chip (NoC), and the AKER access control module. The responsibilities of each member of the team are defined by their respective assignments in the **Project Milestones and Schedule**.

The entire project will be developed in SystemVerilog using the Xilinx Vivado software. Xilinx offers its software and licenses at no cost to students for educational purposes; therefore, the software required for the development of this project is at no cost to the team. This software also includes tools for

timing analysis and simulation, which will be required for the testing and evaluation of our product's performance and security.

Future documentation for this project will be written in LaTeX via Overleaf. Our documentation will use templates to adhere to IEEE formatting standards. With the premium version of Overleaf, the entire team can easily collaborate on the writing and formatting of future technical documents.

Project Milestones

1. ESP Project Setup (Brandon)
    - Definition: Implementation and evaluation of an accelerator using ESP. A specific accelerator (e.g. adder) will be selected from the documentation and github repository of the ESP platform.
    - Deliverables: 1-2 pages report including the setup process (requirements, installation procedure, etc.), accelerator specification, implementation results, and evaluation (functionality correctness).
2. ESP Project Modifications (Brandon)
    - Definition: Modify the ESP platform so that specific architectures can be added on top of it.
    - Deliverables: Modified Codes of the ESP (a new branch in github repository) + 1-2 pages report including the modification process (e.g. modified files) and evaluation study. The evaluation study ensures that the modified version of ESP is bug-free and the corresponding codes are compiled and installed correctly.
3. AKER Project Simplification (Hosein and Chi)
    - Definition: Simplifying the access control system such that it can be implemented in a limited time. In this milestone we will describe the detailed architectures that are required to be implemented on top of the ESP.
    - Deliverables: 3-4 pages report including the architectures (e.g. ACWs, TE, and etc.), detailed structures (interconnections) and algorithms used in simplified version of the

AKER. Report will include the architectural issues to be addressed during the implementation. It will also include:

i. The high-level structure of the architecture (should be a standard tile-based architecture leveraging a Network-on-Chip (NoC) for data exchange)

ii. Internals of the NoC router (i.e., any technicalities related to the implemented routing algorithm, possible routing restrictions, and possible implemented optimizations).

iii. Details and technicalities related to the protocol used by routers for data exchanged among them.

4. Implementation of AKER on ESP (Hosein, Chi, and Brandon)

    - Definition: In this milestone, we will implement the architectures, interconnections, and all required structures of the AKER on the ESP platform. All implementable structures come from the third milestone report.

    - Deliverables: Modified Codes of the ESP (a new branch in github repository branch) + 3-4 pages report including the modification process and documentation about the architectures and algorithms employed in the implementation phase.

5. Final Evaluation and Debugging ((Hosein, Chi, and Brandon)

    - Definition: Testing on the Xilinx Zynq Ultrascale+ platform

    - Deliverables: 3-4 pages report about the correctness, performance, and security evaluation of the implementation.

6. Final Project Video (Hosein, Chi, and Brandon)
    - Definition: Developing a professional looking video describing the project
    - Deliverables: A video (~5 minutes) including team members introduction, project goals, and the final results.
7. Final Report (Hosein, Chi, and Brandon)
    - Definition: Technical report summarizing the project goals and progress over the quarter
    - Deliverable: 8-10 pages report including abstract, introduction, background, implementation details, evaluation, conclusion, and references.

Schedule

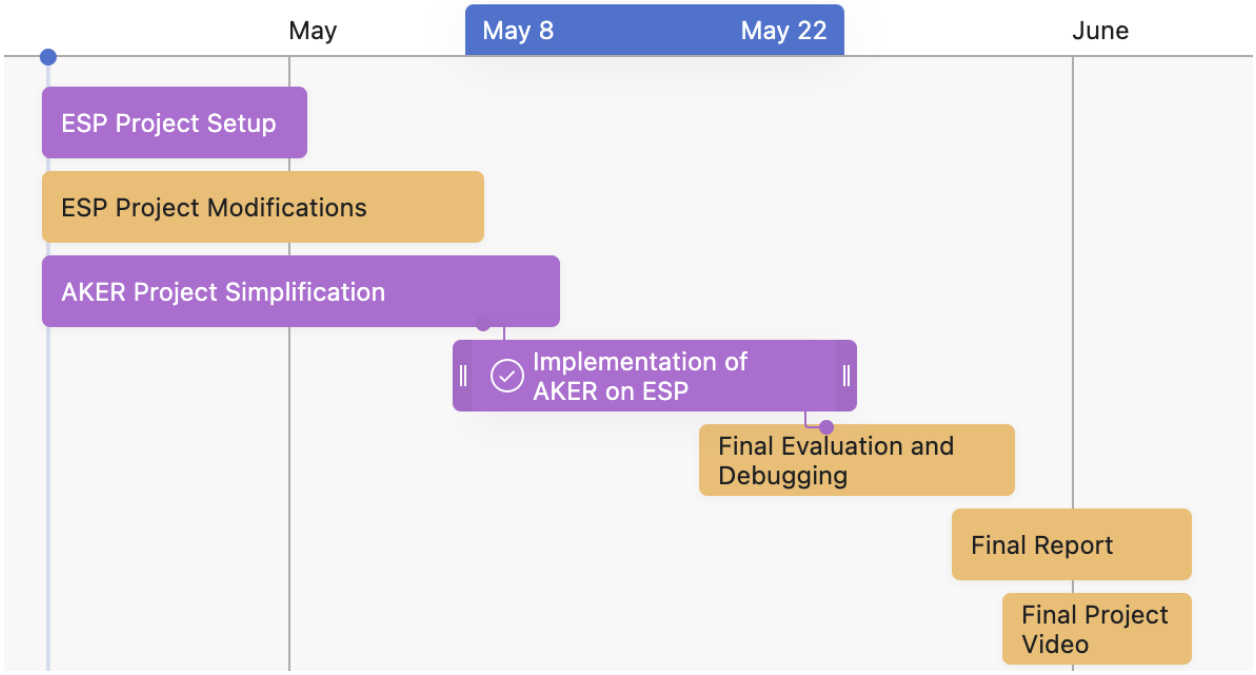
We have approximately 8 weeks to complete the MVP of our project. The following schedule corresponds each week to our planned progress in the project.

<table>
<tr><th>Week</th><th>Goals</th><th>Deliverables</th></tr>
<tr><td>3</td><td rowspan="2">Research ESP, NoC, AKER protocols
Project setup with ESP, documentation</td><td rowspan="2">Design Specification (due 4/12)</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td rowspan="4">RTL (SystemVerilog) Implementation of AKER on the ESP platform</td><td rowspan="2"></td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="2">Milestone Update Report (due 5/17)</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="2">Testing and validation</td><td></td></tr>
<tr><td>10</td><td>Final Video and Report (week of 6/5)</td></tr>
</table>

These are the deadlines for our milestones:

Milestone	Deadline
ESP Project Setup	5/1/2022
ESP Project Modifications	5/8/2022
AKER Project Simplification	5/11/2022
Implementation of AKER on ESP	5/22/2022
Final Evaluation and Debugging	5/29/2022
Final Project Video	6/7/2022
Final Report	6/7/2022

Gantt Chart:


May
May 8
May 22
June
ESP Project Setup
ESP Project Modifications
AKER Project Simplification
Implementation of AKER on ESP
Final Evaluation and Debugging
Final Report
Final Project Video