CSE 237D: SoC Security
Milestones Report

Mahima Rathore | Richa Pallavi

Milestones & Deliverables:

Ownership: All the milestones are joint milestones. Both the team members will be working together on all of them. This was done to encourage better discussion, brainstorming and debugging for various design code/specifications.

Milestone 0 -

A complete plan for the project SoC Security after understanding the past developments within the project. Requires a detailed discussion with project leads Francesco & Armita.

Weekly Milestones:

- Understood the current progress of the SoC progress - meeting with project leads.
- Read documentation on firmware security with security policies, AXI protocol & development of fair interconnect for AXI.
- Had a detailed discussion with team leader (Francesco) to figure out where we best fit in keeping our past experience in mind.
- Prepared a quarter-long plan for this project and got it reviewed by the project lead.

Deliverable: Project Specification Document

Due on: 04/23/2020, Thursday

Progress: Completed

Milestone 1 -

Demonstrating working simulation & successful build of the current SoC - CEP Architecture.

Weekly Milestones:

- Spent time to understand the AXI protocol (AXI4 full & lite standard interface).
- Spent time understanding the CEP Architecture and module hierarchy within it.
- Faced some issues related to software and platform required for this project (specifically w.r.t. fabricant server & sudo permissions) - couldn't resolve them
  Workaround:
  - Migrated some files (immediately required for developing AXI Machines) to our local machines.
  - Used ModelSim on our systems for their compilation. But without any testbench for this design architecture correct compilation or simulation couldn't be done.
- Kept the build of the CEP architecture (design + RISC-V toolchain) on hold for now.

Deliverable:

- Due to unsuccessful build of CEP design on fabricant server, there are no results to display.
- But with the help of the workaround, further delay was avoided for future design development.

Due on: 05/05/2020, Tuesday

Progress: Completed with a workaround for build

Milestone 2 -

Development & verification of AXI lite Slave reading & writing registers internal to hardware accelerator.

Weekly Milestones:

PART A

- Spent time to understand the AXI Master skeleton code from Xillinx.
- Old: Integrate an AXI full master machine with the CEP hardware accelerator (md5).
- New: Use the code design from the AXI full master machine and implement it in md5 top instead of instantiating the module, to make the local variables available to this piece of code.
- Develop a testbench to be able to verify the md5 top design.

PART B

- Develop internal registers within the hardware accelerator to initiate the AXI master transactions & to provide the AXI master source & destination address.
- Implement register access mechanism by AXI4 lite slave interface.
- Through the testbench make the AXI4 lite slave interface able to write and read these registers inside md5 top.

Deliverable: Report presenting code-design & details of the developed module.

Due on: 05/18/2020, Monday

Progress: Completed

Milestone 3 -

Development & verification of AXI Master controlled by control registers of hardware accelerator & its interaction with Pancham.

Weekly Milestones:

PART A

- Use the written register values to control the AXI Master machine by modifying the AXI Master design code.
- Perform data manipulation and conversion into a wishbone interface so that AXI4 Master can write data (read from the testbench) into an input buffer for the Pancham.

- Pancham processes this input data and writes an output data into the output buffer, which the AXI Master writes into the testbench.

PART B

- Through the testbench emulate an AXI slave (which should function like a mock memory) to interact with the modified AXI Master module.
- Debug & complete verification of this interaction.

Deliverable: Report presenting code-design & details of the developed module.

Due on: 05/24/2020, Sunday

Progress: Ongoing

Milestone 4 -

Development & verification of AXI Interconnect & complete test-planning for it.

Weekly Milestones:

PART A

- Understand the AXI interconnect code from Xillinx to integrate with CEP.
- Develop and integrate AXI Interconnect with the hardware accelerator core.
- Make this interconnect configurable for 2 slave interfaces & 1 master interface.

PART B

- Testbench development to verify the interconnect handshake and register access feature.

Deliverable: Report presenting code-design & details of the developed module.

Due on: 05/28/2020, Thursday

Progress: Ongoing

Milestone 5 /Future Goal -

Development and verification of AXI Slaves & complete test-planning for it.

Weekly Milestones:

- Understand the AXI Slave module w.r.t. the interface it communicates within the CEP SoC and its proposed functionality.
- Development of AXI slave and handshake with AXI interconnect master interface.
- Integrate a slave configuration port and registers to the AXI Interconnect to set the memory boundaries.
- Develop testbench to verify this handshaking.

Deliverable: Report presenting code-design & details of the developed module.

Milestone 6/ Future Goal-
Integration of AXI Modules developed, with the CEP Architecture.

Weekly Milestones:

- Complete verification from Master (Hardware accelerator) → Interconnect → Slave(scratchpad) handshake.
    - Develop a verification environment for the AXI Modules with testbench, drivers and monitors.
- Complete the verification by running the test cases and fixing design bugs.
- **Future Goals -- If time permits, we can go ahead integrating our AXI machines with CEP SoC and verify it.**

Deliverable: Report presenting the verification summary.

Grading:

Assigned grades for milestones in mid quarter:

Grade	Milestones
B+	0, 1, 2(part A)
A-	0, 1, 2 (part B), 3 (part A)
A	0, 1, 2, 3 (part B)
A+	0, 1, 2, 3, 4(part A & B)

If we are able to complete the Milestone 4 in time, i.e. get the interconnect up and tested, we will develop and integrate the Slave Scratchpad memory to bring up the full flow
(HW accelerator <-> interconnect <-> Scratchpad Memory) and test it.