System-on-Chip (SoC) Integration: Challenges, Methodologies, and Hands-On Implementation

The design of electronic chips for any application is centered around the well-known trade-off triangle: higher speed, reduced area, and lower power consumption. A key optimization strategy in achieving these objectives is minimizing the need for multiple chips by integrating as many functions as possible into a single integrated circuit, commonly known as a System-On-Chip (SoC). A typical SoC can integrate various components, such as one or more central processing units (CPUs), graphics processing units (GPUs), embedded storage memories, interface peripherals, sensors, radio modems, phase-locked loops (PLLs), and a power management unit (PMU).
The integration of both digital and analog components onto a single chip adds significant complexity, as each component may have unique requirements and interfaces that must be carefully managed. This increases the need for specialized expertise, advanced tools, and often leads to longer development times and higher costs. Additionally, this complexity requires extensive verification throughout the digital design flow to ensure all components work together seamlessly, as a single point of failure can compromise the functionality of the entire system.

The primary objective of this project is to acquire hands-on experience in SoC integration, understand the associated challenges, apply common integration methodologies, and develop a system-level perspective of a fully functional SoC.
Although this project is primarily based on synchronous digital design and will involve a variety of block-level digital design tasks—such as developing a basic RISC-V processor, interface peripherals, digital controllers, and on-chip bus agents—the core activities and learning objectives will focus on SoC integration. Key aspects include multi-clock domain crossing techniques, clock/reset generation logic for multi-clock and multi-power domains, HW/SW interfacing through RISC-V system integration, low-power design methodologies, and behavioral modeling for interfaces with analog components.

Project Activities:

1. Reading and understanding design specifications (e.g., AMBA bus protocol, RISC-V architecture, and interface protocols).
2. Translating design specifications into synthesizable RTL code using HDL.
3. Verifying the design by running simulations with CAD tools and debugging design/integration issues.
4. Performing low-power optimization at the RTL level and writing UPF code to define power intent.
5. Implementing the design on an ASIC technology library or FPGA platform.

Pre-requisites:

- Solid understanding of digital circuit fundamentals.
- Experience with Verilog coding and familiarity with the digital design flow.
- Basic knowledge of computer architecture and the ability to write simple C code to run on a RISC-V embedded processor.