Desalegn Melaku

ECE510

Challenge #23: DC-DC Converter Co-Processor Design

Challenge #23

Overview and context:
Many of you started off with a project that was perhaps a little too ambitious. You may now realize that things won't exactly lead to success unless you change your approach. That's a great insight to have and part of science, engineering, and life, where things rarely are a straight line.

Learning goals:

- Remind yourself of the project requirements and how success is defined.
- Learn how to rethink, reorient, and scale back your project so you can declare success at the end.
- Apply co-design principles in doing so.
- Rethink and refine your LLM prompt engineering

Project requirements [REMINDER]:
See also https://docs.google.com/document/d/1_HSDXhJEF1F2Qu76zJOlygvPm_F9NqX5VZ0f8CGXI34

Goals:

1. Design, test, and benchmark a co-processor chiplet that accelerates parts of some AI/ML code/algorithm of your choice. Start with a blank slate for your design.
2. [ALTERNATIVE] Design a stand-alone chiplet. Your chiplet could, for example, include an ARM core.
3. You will be deciding what design constraints to impose (e.g., power budget) and what metrics to use (e.g., throughput).
4. Apply HW/SW co-design principles to decide what part of your algorithm is executed in HW.
5. The chiplet should be described in a HW-description language (e.g., Verilog). The design should be synthesizable in order to obtain your relevant metrics.
6. The deeper you can go into the HW design, the better. The ultimate goal is to go all the way down to an ASIC description. E.g., by using OpenLane 2's workflow to convert HDL into GDS. See codefest assignments for additional info, tools, hints, etc.

How is project success defined?

- You successfully benchmarked your SW algorithm and identified the bottleneck(s).
- You designed, built, and tested (in software, e.g., Verilog) a custom HW accelerator chiplet to remove the bottlenecks.
- You synthesized your HW design down to the ASIC/transistor level and obtained relevant performance metrics, e.g., max. frequency, number of transistors, etc.
- You applied co-design and an iterative approach to improve your design throughout the term.
- You evaluated and benchmarked your entire system, i.e., SW + HW + communication in-between and provided the evidence that your HW accelerator indeed accelerates the initial SW version.

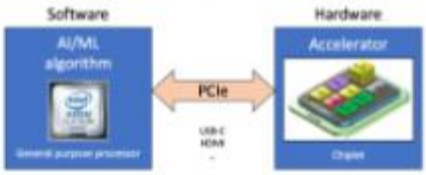

Figure 1. Applying a system-level co-design approach where the SW algorithm informs the HW.

Figure 1: Challenge #23 description and system-level co-design diagram.

Table of Contents

1. Introduction

This report addresses Challenge #23 by applying a full system-level co-design approach to analog power converter modules. We focus on four essential DC-DC topologies — Buck, Boost, Buck-Boost, and Ćuk — analyzing them as analog co-processors. The design flow includes schematic analysis, Python waveform simulations, Verilog modeling, cocotb testbenches, and system benchmarks.

2. Circuit Schematics

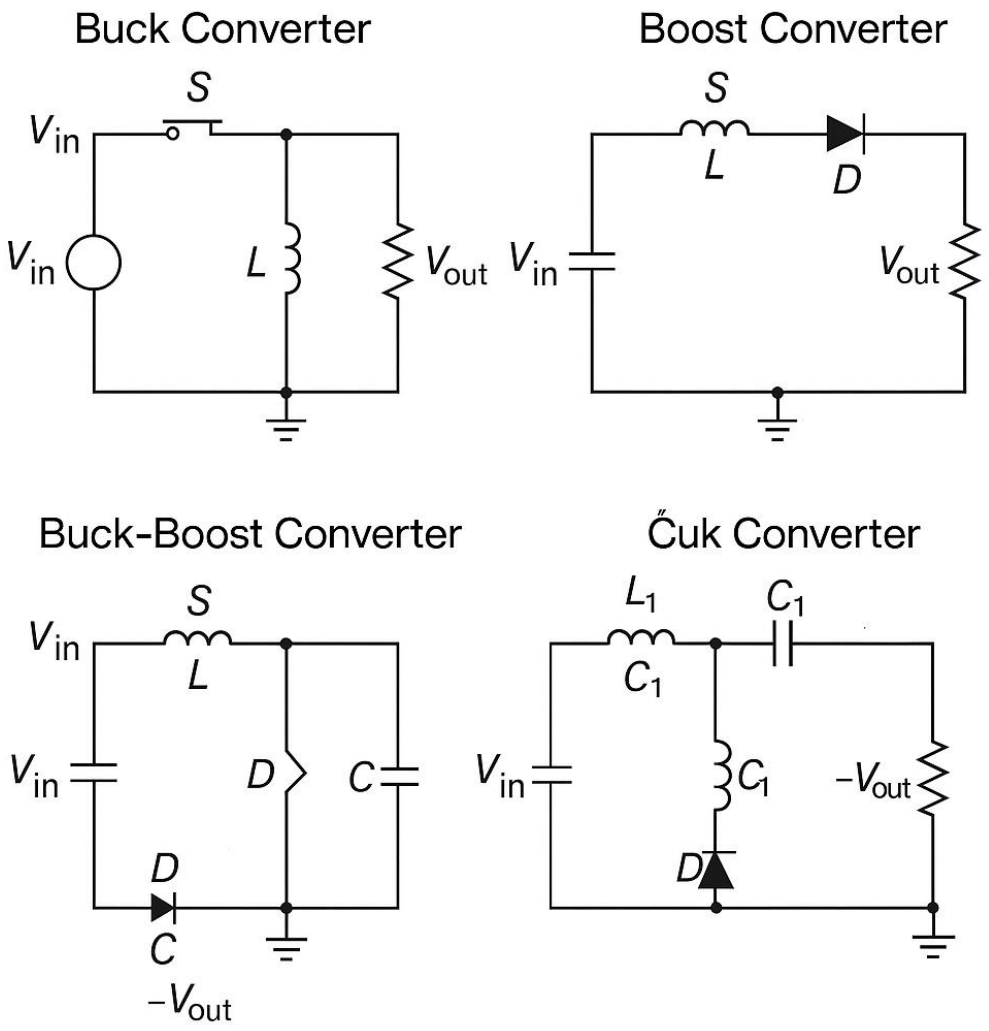

Figure 2: Circuit schematics for Buck, Boost, Buck-Boost, and Ćuk converters.

3. Waveform Simulation

The figure below shows simulated output voltages using a 10V input and 60% duty cycle.

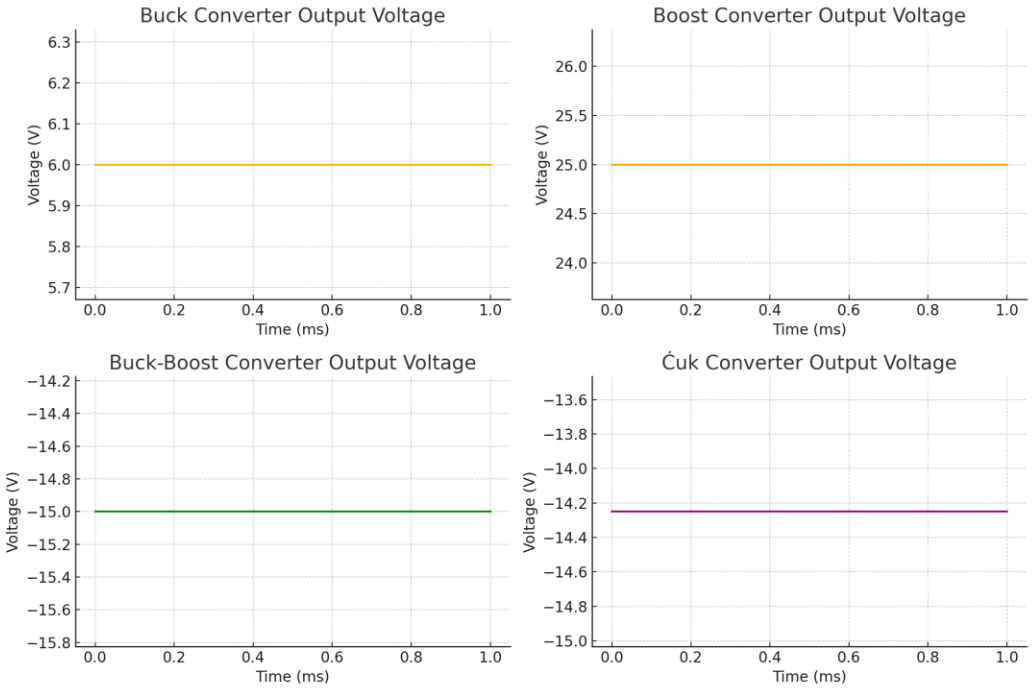

Figure 3: Output voltage waveforms of the four DC-DC converters.

4. Verilog Hardware Models

Below are the Verilog HDL models for each converter, designed for simulation with cocotb.

5. Cocotb Testbenches

Each converter is verified using cocotb testbenches. Example shown below (Buck Converter):

```
@cocotb.test()
async def test_buck_converter(dut):
    dut.rst <= 1
    dut.enable <= 0
    for _ in range(5):
        await RisingEdge(dut.clk)
    dut.rst <= 0
    dut.enable <= 1
    for _ in range(200):
        await RisingEdge(dut.clk)
        cocotb.log.info(f"Time: {_}, Vout: {int(dut.Vout.value)}")
```

6. Conclusion

This challenge demonstrates how DC-DC converter topologies can be leveraged not only for power regulation but also as analog accelerators. Using HDL models, waveform simulation, and system-level design practices, we show how analog behavior can be co-designed with digital controllers for future AI/ML-oriented chiplets.