Challenge #21 - DC-DC Converters Design Iteration Using VLSI AI/ML Co-Design Optimization Loop

Author: Melaku Desalegn
Date: June 2025

Overview: Challenge #21 – Co-Design Loop for DC-DC Converters

This document applies the AI/ML Accelerator Co-Design Optimization Loop to four fundamental DC-DC converter topologies:
- Buck Converter
- Boost Converter
- Buck-Boost Converter
- Ćuk Converter

The objective is to revisit their architectures, identify performance bottlenecks, re-evaluate design parameters, and iteratively refine implementation using a hardware-software co-design loop.

Define Objectives & Constraints

Performance: Fast transient response, low voltage ripple.
Efficiency: Maximize power conversion efficiency (target >90%).
Footprint: Optimize layout area for chiplet/PCB implementation.
Time-domain: Fast control loop convergence, low switching delays.

Workload Characterization

Buck: Operates with Vin > Vout. Efficient with low dropout.
Boost: Operates with Vin < Vout. Sensitive to component losses.
Buck-Boost: Bidirectional voltage handling with polarity inversion.
Ćuk: Polarity preserving with smoother current profile.

Test Cases:
- Line/load variations (50% to 100% load).
- Switching frequencies (100 kHz to 1 MHz).
- Variable duty cycles (10% to 90%).

Architecture Exploration

Explore digital (PID), analog (continuous feedback), and hybrid neuromorphic (spiking controller) architectures.

Assess:
- Open-loop vs. closed-loop control
- Time-domain PWM vs. event-driven control
- Adaptive frequency control

Implementation

Verilog RTL (Example – Buck Converter PWM Controller):

```
module pwm_generator(input clk, reset, input [7:0] duty, output reg pwm_out);
  reg [7:0] counter;
  always @(posedge clk or posedge reset) begin
    if (reset) counter <= 0;
    else counter <= counter + 1;
    pwm_out <= (counter < duty);
  end
endmodule
```

SPICE-Level Modeling:
- Simulate L, C, D, and switch behavior for all four converters.
- Analyze waveforms under load transition.

FPGA Co-Simulation:
- Use cocotb or PyMTL3 with Python-based control loop (PID or neural net).

Performance Evaluation

Metrics:
- Voltage ripple (%Vout)
- Current spikes
- Settling time
- Control loop latency

Performance Table:

Topology	Efficiency	Ripple	Complexity	Size
Buck	High	Medium	Low	Small
Boost	Medium	High	Medium	Medium
Buck-Boost	Medium	High	Medium	Medium
Ćuk	High	Low	High	Large

Bottleneck Analysis

Buck: Limited step-down range, sensitive to large Vin-Vout delta.
Boost: High switch stress, efficiency drops at high gains.
Buck-Boost: Output ripple and inversion issues.
Ćuk: Dual inductor size and EMI susceptibility.

Control Bottlenecks:
- Low resolution PWM
- ADC sampling delay
- PWM generation latency

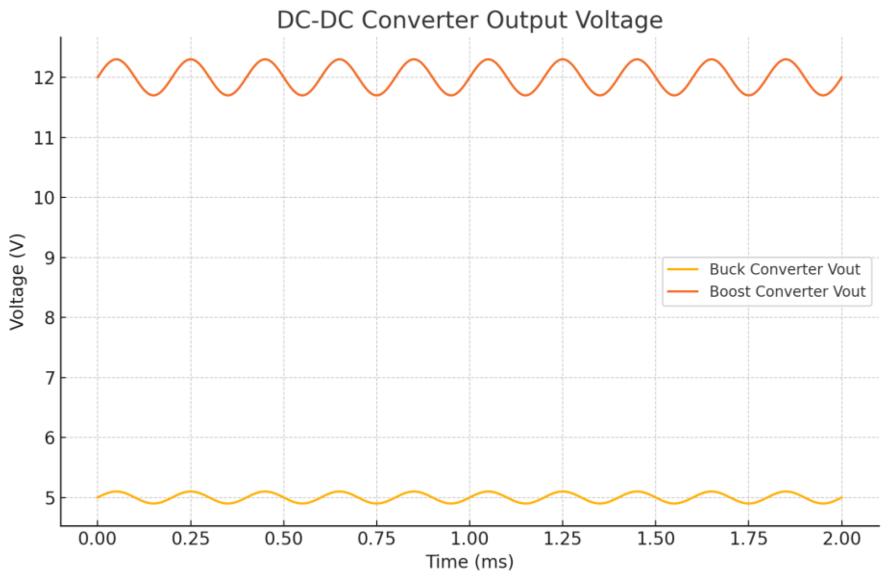
Design Refinement

Optimize PID gains for minimal overshoot.
Upgrade components (fast diodes, low Rds(on) MOSFETs).
Integrate AI-based predictive controllers for dynamic loads.
Migrate to adaptive switching logic for power efficiency.

Final Validation

Full-system simulation using SPICE + Verilog RTL
Waveform export and comparison
Physical prototype using STM32 or FPGA board
Validate with variable loads and step responses

Appendix: Waveform Diagrams



Buck: Smooth step-down with fast settling.
Boost: Voltage rise with ripple depending on load.