Lab 7: Design Project – Discrete Analog-to-Digital Converters (ADCs)

1. Introduction & Project Goals

This lab is the culmination of the skills you've developed throughout the course. Your team will design, implement, and test several discrete Analog-to-Digital Converters (ADCs) on the Basys3 FPGA board.

This project challenges you to integrate concepts from previous labs—including the XADC, PWM and R2R circuits, and hierarchical design—into a single, complex system. The goal is to build a functional prototype, verify its performance, and professionally document your design and results. Success will require careful planning, iterative development, and effective teamwork.

---

2. Key Deadlines & Grading

This project constitutes **40% of your final grade** and is broken down into three main deliverables. **Note:** As per the course outline, you must achieve a minimum grade of 60% on this design project to be eligible to pass ENEL 453.

Deliverable	Weight	L01 Dates (Thurs)	L02 Dates (Fri)
MVP Demonstration & Design Record	10%	Nov 20	Nov 21
Final Demonstration & Design Record	20%	Nov 27 / Dec 4	Nov 28 / Dec 5
Final Report Submission	10%	Due Monday, Dec 1 (All Sections)	Due Monday, Dec 1 (All Sections)

---

3. Core Design Requirements (Tiered Grading)

The project is structured in tiers, allowing your team to target a grade level (70%, 90%, or 100%) by completing a corresponding set of features. You must complete all requirements from a lower tier to be eligible for the next.

Baseline System (Eligible for up to 70%)

Implement a system with two discrete Ramp ADCs and an XADC for comparison.

- **Menu & Display:**
    - Use slide switches to select which ADC value is shown on the 7-segment displays. You are encouraged to use some slide switches for menu selection and some slide switches to pass a known value to be displayed (useful for debugging).
    - Implement a selector for Hexadecimal or Decimal display format.
- **XADC Subsystem (Reference):**
    - Display the raw 12-bit XADC value.
    - Display the averaged and scaled XADC voltage value.
- **Discrete PWM Ramp ADC Subsystem:**
    - Display the raw 8-bit ADC value from your PWM-based ramp-compare ADC.
    - Display the averaged and scaled voltage value.
- **Discrete R2R Ladder Ramp ADC Subsystem:**
    - Display the raw 8-bit ADC value from your R2R-ladder-based ramp-compare ADC.
    - Display the averaged and scaled voltage value.
- Make reasonable assumptions to interpret these requirements.

Advanced System (Eligible for up to 90%)

Includes all **Baseline System** requirements, plus the following:

- **Successive Approximation Algorithm:**
    - For both the PWM and R2R Ladder subsystems, add a switch to select between the **Ramp ADC** algorithm and a **Successive Approximation ADC** algorithm.
    - The Successive Approximation algorithm **must** be implemented as a well-structured Finite State Machine (FSM) with clear state names.

Creative Extension (Eligible for up to 100%)

Available **only** after completing all **Advanced System** requirements. You may implement a combination of the following features to add up to 10% to your 90% score.

- **(2%) Maximize Clock Frequency:** Use the Vivado PLL IP to generate a system clock significantly higher than 100 MHz while still meeting timing (positive WNS). Document the procedure.
- **(5%) Implement an Additional ADC Algorithm:** Design and implement another ADC type, such as a Flash, Sub-Ranging, or Delta-Sigma ADC. For example, Flash takes many hardware resources, so a 3-bit Flash ADC is acceptable as a proof of concept.
- **(5%) Auto-Calibration:** Implement a function, triggered by a pushbutton, that uses the XADC as a reference to auto-calibrate your discrete ADCs.
- **(2%) Expand ADC Resolution:** Increase the raw conversion range of both discrete ADCs to 16 bits (before averaging).
- **(5%) Sample-and-Hold Circuit:** Implement an external sample-and-hold circuit at the discrete ADC inputs to freeze the analog value during conversion for improved accuracy.
- **(Variable %) Propose Your Own Extension:** You may propose a different creative feature to the instructor in-person. Its scope and point value must be pre-approved by the instructor via email.

4. Mandatory Design Standards

Your project must adhere to the following professional design standards. **Failure to meet these standards will result in significant mark deductions, potentially leading to a failing grade for the project.**

- **Synchronous Design (Absolute Requirement):**
    - The entire design must use a **single clock source** (e.g., `posedge clk`).
    - A single, consistent reset strategy must be used (e.g., synchronous, active-high).
    - The design must be free of inferred latches.
    - The final implementation must meet timing with a **positive Worst Negative Slack (WNS)**.
- **Modular & Hierarchical Design:**
    - The top-level module must be clean, containing only module instantiations and wiring. No RTL logic (e.g., always blocks, assign statements) is permitted at the top level.
    - The design must be logically partitioned into smaller, reusable modules with clear functions (e.g., a dedicated `seven_segment_subsystem`). Avoid monolithic modules that perform many unrelated tasks.
- **Code Formatting & Style:**
    - Use **SystemVerilog** syntax (`logic`, `always_ff`, `always_comb`). Do not mix Verilog syntax (e.g. `wire`, `reg`, `always@)`.
    - Use clear and descriptive names for all signals, modules, and parameters.
    - Format code with consistent indentation and whitespace for readability.
    - Include a header comment for each module describing its function, inputs, and outputs.
- **Full Functionality on Basys3 (Absolute Requirement):**
    - Every feature claimed in your report and documentation **must** be fully functional and demonstrable on the Basys3 board with your external circuits.

5. Deliverables & Demonstrations

A. MVP Demonstration (10%)

The Minimum Viable Product (MVP) shows that your core system architecture is functional.

- **Required Functionality:**
    1. A fully working XADC subsystem (from Lab 5).
    2. A functional Ramp Compare ADC using *either* a PWM or R2R ladder.
    3. A user-selectable display showing raw ADC values and scaled voltages for both the XADC and your one discrete ADC (four total display options).
- **Before Your Demo:** Submit a single **Design Record (PDF)** to D2L containing:
    - A screenshot of your top-level RTL schematic.
    - Your Vivado Timing Summary and maximum frequency calculation.
    - A copy-and-paste of all your SystemVerilog (.sv) and constraints (.xdc) files.
    - A clear statement of which design requirements you have met.

B. Final Demonstration (20%)

This is the demonstration of your completed project.

- **Before Your Demo:** Submit an updated **Design Record (PDF)** to D2L containing the same items required for the MVP, but reflecting your final design.
- **During Your Demo:**
    - State which tier (70%, 90%, 100%) you have completed.
    - Demonstrate every feature required for that tier.
    - Be prepared to explain any part of your hardware design, software implementation, or external circuitry. All team members must be present and may be questioned individually.

C. Final Report (10%)

Your report is a professional document detailing your project. It must be submitted as a **single PDF** to D2L.

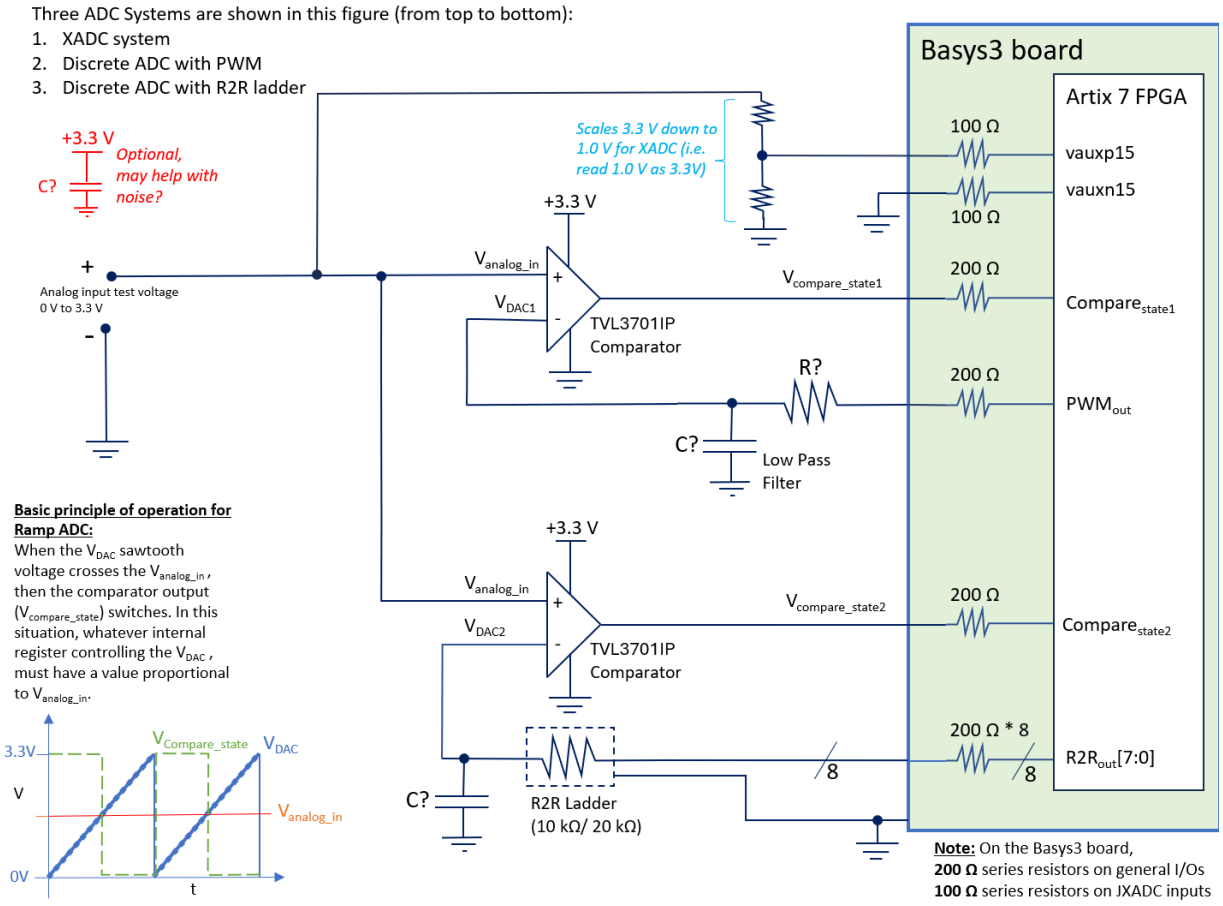
- **Report Structure:**
    1. **Title Page**
    2. **Table of Contents**
    3. **Introduction:** Project overview and objectives.
    4. **System Architecture:** High-level description of the design. Include a top-level RTL schematic, a schematic of your external circuitry, and photos of your prototype.
    5. **Detailed Design:** In-depth explanation of key modules. Include code snippets, FSM state diagrams, and calculations where appropriate.
    6. **Implementation Results:** Report and explain your FPGA resource usage and final timing (WNS) with screenshots from Vivado and calculation of your maximum clock frequency.
    7. **Testing & Verification:** Document how you tested your prototype against the design requirements.
    8. **AI Use Discussion:** Fully disclose which AI tools you used and for what purpose. Critically analyze their contributions and limitations. If you did not use AI, state so.
    9. **Lessons Learned:** Reflect on what went well, challenges you faced, and what you would do differently.
    10. **Conclusion:** Summarize the project and your results.
    11. **Appendix 1: Full Code:** Include all RTL (.sv) and constraints (.xdc) files.
    12. **Appendix 2: Team Contributions:** Each member must write a section detailing their specific contributions to the design, coding, testing, and report writing.

6. Guidance & Best Practices

- **Iterative Development:** Do not try to build everything at once. Start with the XADC subsystem, then implement one discrete ADC, verify it, and then add the next. A step-by-step approach is crucial for managing complexity.
- **Plan Your Design:** Draw block diagrams of your system *before* you start coding. Define the modules and the signals connecting them to visualize signal flows. This will save you significant time during implementation.
- **Avoid Paralysis by Analysis:** While planning is important, avoid overanalyzing a problem to the point that it delays or prevents progress. Aim for a balance between planning and execution. Once you have a reasonable understanding of the problem, begin implementing and be prepared to make adjustments as you go.
- **Avoid Premature Optimization:** Focus on building a working system first. Do not try to refine a block that is already functioning correctly unless it's necessary—for example, if you fail to meet timing constraints. To keep your design simpler, avoid sharing resources like the averager module; provide each subsystem with its own instead.
- **Test As You Go:** Use testbenches to verify individual modules before integrating them into the larger system. Debugging small, isolated parts is much easier than debugging the entire system.
- **AI Tool Usage:** You are permitted to use AI tools to support your learning and development, but be cautious as responses could be incorrect. You are responsible for validating any information or code they provide. You must fully understand and be able to explain everything you submit. Your use of AI must be fully disclosed and discussed in your final report.

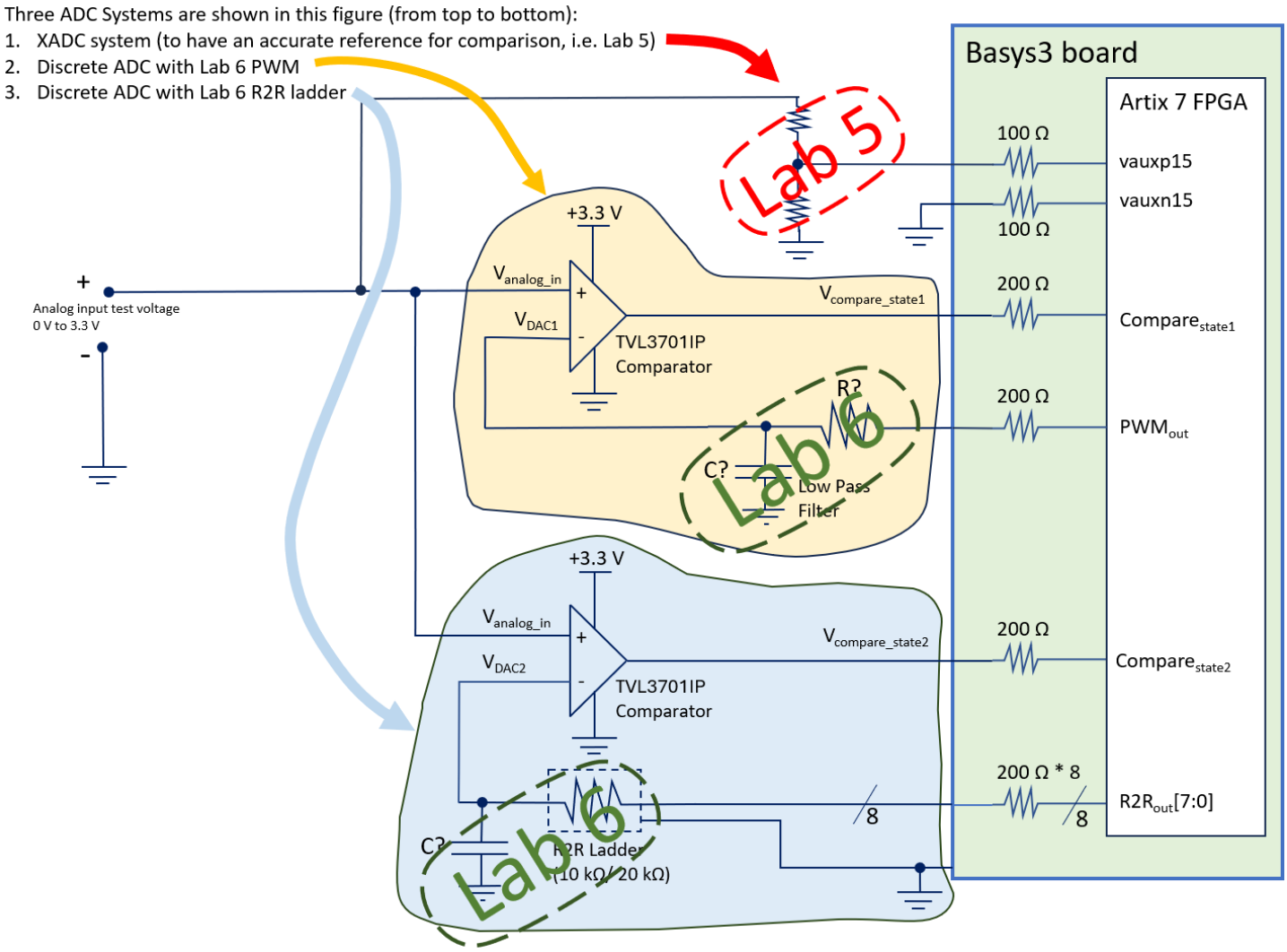
Below is a diagram representing the XADC and Discrete ADC systems, to help guide your development. The basic principle for the Discrete Ramp ADC is as follows.

- A ramp (i.e. sawtooth) waveform is generated by either PWM or R2R ladder.
- The ramp waveform (V_{DAC}) is fed into a comparator (the TVL 3701, see the datasheet). The analog input voltage (V_{analog_in}) is fed into the other comparator input.
- According to the diagram below, if the analog input voltage is greater than the ramp waveform voltage, then the comparator will produce a high value (3.3 V) on its output ($V_{compare\_state}$). If the analog input voltage is lower than the ramp waveform voltage, then the comparator will produce a low (0 V).
- Because the ramp waveform is always varying across the full range of 0 V to 3.3 V, then it will eventually cross the analog input voltage, and the comparator's $V_{compare\_state}$ will switch from high-to-low. Detecting the high-to-low transition will inform the system that the register value generating the V_{DAC} voltage will be proportional to V_{analog_in} . This register value will be the Duty Cycle for the PWM output, and the Duty Cycle equivalent for the R2L ladder output. Therefore, upon the $V_{compare\_state}$ low-to-high transition, the Duty Cycle will be captured as the raw ADC value, which will then undergo further processing (averaging and scaling).



To help you put your past lab projects in context with Lab 7, please refer to the figure below.

- The XADC system you built in Lab 5, will be incorporated in the design, to provide an accurate reference value for the voltage readings of the analog test voltage.
- The PWM and R2R ladder circuits you built in Lab 6 to generate the sawtooth waveforms, will be incorporated into the Discrete ADCs that you will design and build for Lab 7.
- Your previous lab projects where you built registers, multiplexers, 7-segment display drivers, and other circuit elements, all give you many concepts to apply to this project. In addition, you have explored hierarchical design, verification in simulation, debugging, synchronous design, and especially, incremental and iterative development – all of which you will employ to complete the Lab 7 project successfully.



7. Deliverables & Rubrics

This section outlines the specific project deliverables, their deadlines, and the rubrics that will be used for grading.

A. Deliverables & Deadlines

Mark these dates in your calendar. It is your team's responsibility to sign up for demonstration times and submit all materials by the deadlines.

- **MVP Demonstration**
    - **L01:** During lecture time on **Thursday, November 20**.
    - **L02:** During lecture time on **Friday, November 21**.
    - Your MVP Design Record must be submitted to the D2L Dropbox *before* your demonstration begins.
- **Final Report Submission**
    - **Due Date:** The report is due in the D2L Dropbox by **11:59 pm on Monday, December 1** for all sections.
    - **Late Policy:** A 24-hour grace period is provided. After that, a penalty of 2 points per day will be applied.
- **Final Demonstration & Q&A**
    - **L01:** During lecture times on **Thursday, November 27 and December 4**.
    - **L02:** During lecture times on **Friday, November 28 and December 5**.
    - **Missed Presentations:** If you miss your scheduled time, you may sign up for a makeup slot but will incur a 2-mark deduction.

B. Grading Rubrics

Important: As per the course outline, you must receive a grade of **at least 60% on this Design Project** to be eligible to pass the course. Fractional points will not be awarded for any component.

MVP Demonstration & Q&A (10 points)

Points	Criteria
10	**All four MVP requirements were successfully demonstrated** (XADC and one Ramp Compare ADC, showing both raw values and voltages on the 7-segment display). The team explained the design clearly, answered questions well, and was prepared and efficient.
6	The **Ramp Compare ADC (raw value and voltage) was successfully demonstrated**. The team explained the design well and answered questions proficiently.
0 - 5	The Lab 5 (XADC) functionality was demonstrated, and the team showed **significant progress** toward the Ramp Compare ADC, but technical issues prevented full functionality. The team must clearly explain the bugs and their plan to fix them.

Final Report (10 points)

Points	Criteria
9 – 10	**Excellent Quality.** The report is fully complete, well-written, and well-organized, with clear visuals and thorough proofreading.
7 – 8	**Good Quality.** The report is complete, but may have some weak areas, such as insufficient explanations, unclear visuals, or minor organizational issues.
5 – 6	**Weak Quality.** The report shows a lack of care, with poor coverage of the project, sloppy writing, poor visuals, or significant organizational problems.
0 - 5	**Poor Quality.** The report has significant deficiencies in content and presentation.

Final Demonstration & Q&A (20 points)

Points	Criteria
20	**100% Level:** All claimed requirements were successfully demonstrated. The team provided excellent explanations and answers, and the presentation was organized and efficient.
18	**90% Level:** All claimed requirements were successfully demonstrated. The team provided excellent explanations and answers, and the presentation was organized and efficient.
14	**70% Level:** All claimed requirements were successfully demonstrated. The team provided excellent explanations and answers, and the presentation was organized and efficient.
0 - 13	**Marginal Performance:** The demonstration showed limited functionality, and the team provided weak answers or explanations. An inability to explain the design or navigate the Vivado project will result in a score of zero and may lead to a report for academic misconduct.

Note on Demonstrations: The TA may ask any team member to answer any question about the project. All members are expected to have a comprehensive understanding of the entire design. Be prepared to demonstrate the full Vivado design flow if asked, including creating a project, synthesizing the design, and simulating it. Any deficiencies may result in point deductions from those listed in the Grading Rubrics.