Department of Electronics
Faculty of Engineering, Carleton University,
Ottawa, Canada.

ELEC 3500 [0.5 credit]
Microprocessor Systems

Course Description:

Digital circuit design using SystemVerilog and logic synthesis, the electronic properties of logic gates, electrical interfacing between logic families, asynchronous to synchronous interfacing, clock distribution and timing, VLSI design options. Students implement substantial circuits with field-programmable gate arrays (FPGA).

Prerequisites: **ELEC 2507** and **ELEC 2607**.
Lectures three hours a week, laboratory three hours a week.

1. These learning outcomes were originally formulated by Prof. Atia, (2018) and Prof. MacEachern continued with these goals (2019-2021).

Specific Goals - Student Learning Objectives

To equip students with the theoretical knowledge and practical skills required to design, verify, and implement complex digital systems using modern, industry-relevant methodologies and tools. This course bridges the gap between fundamental digital logic concepts and the advanced techniques.

Upon successful completion of this course, students will be able to:

1. Design and Model with Modern Hardware Description Languages (HDLs)
    - **Analyze** a set of digital system requirements to select appropriate architectural strategies and data structures.
    - **Model** complex combinational and sequential logic circuits using synthesizable SystemVerilog constructs.
    - **Differentiate** between synthesizable and non-synthesizable language constructs, and explain the implications of their use in a design flow.
    - **Implement** finite state machines (FSMs) and data path components for practical applications.
2. Design and Implement on FPGA
    - **Demonstrate** an understanding of the constituent components of complex digital systems, including processors, memory subsystems, and custom peripherals.
    - **Integrate** custom hardware accelerators (developed using both HDL and HLS) with soft-core blocks on a single FPGA.
    - **Design and test** the interface required for communication between the custom hardware components and interfaces.
5. Employ Professional Engineering Tools and Practices
    - **Utilize** industry-standard Electronic Design Automation (EDA) tools for simulation, synthesis, implementation, and debugging.
    - **Critique** design and basic verification strategies based on their adherence to industry best practices.
    - **Prepare** clear and concise design documentation and reports to communicate technical decisions and outcomes effectively.

Recommended Resources:

- **Course Brightspace**
- **Digital System Design**
  1. Brock, J. LaMeres. *Introduction to Logic Circuits & Logic Design with Verilog*. Spinger, 3rd ed, 2024.
  2. Tu, Kaihui, Xifan Tang, Cunxi Yu, Lana Josipović, and Zhufei Chu. *FPGA EDA: Design Principles and Implementation*. Springer, 2024.
  3. Kilts, Steve. *Advanced FPGA design: architecture, implementation, and optimization*. John Wiley & Sons, 2007.
  4. Pedroni, Volnei A. *Finite state machines in hardware*. MIT Press, 2016.
  5. Meyer-Baese, Uwe. *Embedded microprocessor system design using FPGAs*. Springer International Publishing, 2021.

- **SystemVerilog**
  1. Taraate, Vaibbhav. *SystemVerilog for Hardware Description: RTL Design and Verification*. Springer Nature, 2020.
  2. Spear, Chris. *SystemVerilog for verification: a guide to learning the testbench language features*. Springer Science & Business Media, 3rd ed. 2012.
  3. Mehta, Ashok B. *Introduction to SystemVerilog*. Springer Nature, 2021.

Lecture Plan: A draft of the lecture plan is given below. I will make materials available to the class on Brightspace. The flow of the course will be as below, with some variation depending on interests and circumstances.

Module 1: Foundations of Digital Systems (Weeks 1-2)

This module establishes the physical and architectural groundwork for the course, connecting abstract design concepts to the reality of silicon and FPGAs.

Week	Lec #	Lecture Title	Description	Suggested Reading
1	1	Course Introduction & The Modern Design Landscape	An overview of the course's modern, system-level focus. We will introduce the key tools (Vivado, VS Code, Git) and methodologies for the semester. This lecture establishes the "why" behind moving from basic logic design to complex system integration and verification.	Instructor Notes
	2	From Transistors to Gates: CMOS Logic Fundamentals	A detailed review of the physical basis for digital logic, essential for students new to circuit-level implementation. We will cover how NMOS and PMOS transistors are combined to create CMOS inverters and basic logic gates. The focus will be on the Voltage Transfer Characteristic (VTC) and the concept of noise margins and circuits operational costs, such as delay and power consumption.	Brock: Ch. 3
2	3	Physical Basis of Timing and Metastability	Building on the previous lecture, we will explore the physical origins of key non-idealities. Topics include propagation delay and how it leads to the critical timing parameters of flip-flops. This lecture will introduce the concept of metastability as a direct consequence of violating these timing constraints.	Brock: Ch. 7
	4	FPGA Architecture and Hardware Cost	An early introduction to the target hardware. We will explore the fundamental building blocks of FPGAs: Configurable Logic Blocks (CLBs), Look-Up Tables (LUTs), Flip-Flops (FFs), Block RAM (BRAM), and DSP slices. Crucially, we will learn about synthesis tools and understand the hardware cost of our HDL code, connecting design choices to resource utilization.	Tu: Ch. 1, 2

Module 2: RTL Design and System Components (Weeks 3-6)

This module focuses on using SystemVerilog as a modern, powerful language for describing and creating synthesizable hardware tackling more complex components.

Week	Lec #	Lecture Title	Description	Suggested Reading
3	5	Modernizing RTL	An introduction to the powerful, synthesizable features of SystemVerilog that improve upon traditional Verilog. We will cover essential constructs like the logic data type, specialized always blocks (always_comb, always_ff), and enhanced port connection syntax, demonstrating how they lead to more robust and readable code.	Taraate: Ch. 1, 2, 3
	6	Interfaces and Abstraction	We will explore advanced SystemVerilog constructs for designing complex, reusable components. Key topics include using interfaces to bundle signals and simplify module connections, parameterization for creating flexible designs, and the use of structs and enums for better data organization.	Taraate: Ch. 4, 5
4	7	Combinational Logic Building Blocks	Applying SystemVerilog concepts to design fundamental combinational circuits. This lecture will cover the implementation of decoders, encoders, multiplexers, and comparators, focusing on efficient and synthesizable coding styles.	Taraate: Ch. 6
	8	Sequential Logic Building Blocks	This lecture focuses on creating core sequential components. We will cover the design of various registers (with synchronous load, reset, and	Taraate: Ch. 7

			enable), shift registers, and counters, emphasizing best practices for synthesizable sequential logic.	
5	9	**FSM Design (1)**	This lecture reviews the implementation of Moore and Mealy Finite State Machines (FSMs) using SystemVerilog. We will cover best practices for coding FSMs with separate blocks for next-state logic, state registers, and outputs to create robust and readable controllers.	Pedroni: Ch. 1-4
	10	**FSM Design (2)**	We will build on FSM fundamentals by introducing advanced techniques like FSM partitioning for managing complexity and reducing power consumption. We will also discuss different state encoding schemes (binary, one-hot, gray) and their impact on performance and area.	Pedroni: Ch. 5, 7, 8, 10, 11, 13
6	11	**Midterm Review**	A dedicated session to review all topics covered in Modules 1, 2, and 3. This lecture will focus on key concepts and problem-solving strategies in preparation for the midterm examination.	All previous readings
	12		Midterm Exam	

Week 7: Study Week (No Lectures)

Module 3: System Timing and Integration (Weeks 8-9)

This module addresses critical system-level challenges, including Static Timing Analysis (STA), Clock Domain Crossing (CDC), reset strategies, and power management.

Week	Lec #	Lecture Title	Description	Suggested Reading
8	13	**Clocking Fundamentals**	A detailed look at on-chip clock generation, distribution, and management. We will cover the use of Mixed-Mode Clock Managers (MMCMs) and Phase-Locked Loops (PLLs) via the Xilinx Clocking Wizard IP. We will also discuss safe clock gating techniques for power reduction.	Kilts: Ch. 1, Lecture notes,
	14	**Static Timing Analysis (STA)**	An essential lecture on the dangers of passing signals between different clock domains. We will define metastability and its potential to cause system failure and introduce the fundamental two-flip-flop synchronizer for safely crossing single-bit control signals.	Kilts: Ch. 6
9	15	**Clock Domain Crossing (CDC)**	This lecture explores the critical role of reset signal circuits in advanced digital design. We will learn the differences between asynchronous and synchronous resets, the risks of metastability, and the necessity of proper synchronization. The session covers reset tree design, timing and noise immunity, and addresses challenges like reset domain crossing in complex SoC and FPGA systems.	Kilts: Ch. 9
	16	**Clock Domain Crossing (CDC)**	Metastability and Mean Time Between Failures (MTBF). Synchronization techniques: 2-Flip-Flop Synchronizer, Pulse Synchronizers, and Handshaking protocols.	Lecture Notes

Module 4: System Design Techniques (Weeks 10-13)

This module introduces advanced design methodologies that raise the level of abstraction and explores the interaction between hardware and software.

Week	Lec #	Lecture Title	Description	Suggested Reading
10	22	**Verification (1): Mindset and Testbench Architecture**	An exploration of why verification consumes the majority of effort in modern chip design.[1]This lecture focuses on building a verification mindset and introduces the architecture of a modern testbench using SystemVerilog's powerful features, moving beyond simple Verilog testbenches.	Spear: Ch. 1, 2

	18	Verification (2): Fundamentals	A foundational lecture on the building blocks of a modern testbench. We will cover SystemVerilog classes, objects, properties, and methods. Understanding OOP concepts in building complex verification components.	Spear: Ch. 3, 4
11	19	Embedded Processors on FPGAs	This lecture explores the use of soft-core processors (like RISC-V, MicroBlaze) and hard-core processors (like ARM) within an FPGA. We will discuss how they serve as powerful controllers and introduce key concepts like the AXI bus and JTAG for debugging.	Meyer-Baese Ch.1
	20	A Brief on High-Speed Data Transfer	This lecture addresses the challenges of high-speed data transfer, explaining the limitations of wide parallel buses. We will introduce high-speed serial communication as the solution, covering the concepts of SERDES (Serializer/Deserializer), Clock and Data Recovery (CDR), and the use of dedicated multi-gigabit transceiver IP blocks in modern FPGAs.	Lecture Notes
12	21	Design Metric Analysis	An introduction to power management in complex systems. We will discuss power domains, using power estimation tools like the Xilinx Power Estimator.	Tu, Ch. 4, 5, 6 / Xilinx Docs (UG440)
	17	System Memory and High-Speed Signaling Technologies	A system-level overview of different memory types (SRAM, DRAM, ROM, Flash) and their integration into digital designs. We will discuss memory interfaces, controllers, and how to model memories in SystemVerilog for synthesis into Block RAM (BRAM).	Lecture Notes
13	23	High-Level Synthesis	This lecture introduces HLS, a powerful technique for generating RTL hardware from high-level languages like C/C++. We will discuss the benefits, such as faster design exploration, and walk through the basic HLS flow.	Tu, Ch. 8
	24	Final Course Review & Exam Preparation	A comprehensive review of all major topics covered during the semester, with an open Q&A session to prepare students for the final examination.	All Materials

The course grade will be evaluated as follows:

Method of Evaluation	% of Final Grade	Due Dates*
Lab Assignments	25	lab days
Midterm Exam (Individual)	25	After week 6
Final Exam (Individual)	50	TBA

Notes:

- The exams are for evaluation purposes only and will not be returned to the student.
- Health and Safety: See https://carleton.ca/ehs/programs/working-lab/laboratory-health-and-safety/ for general guidelines.

Satisfactory term work

- In order to pass the course students must achieve *satisfactory performance* during the term.
- Satisfactory performance during the term requires completion of all lab experiments with a combined average grade of >30% on lab reports.
- The final exam must be completed with a minimum grade of 40% to pass the course.

Instructor information

- Name: Arash Ahmadi
- Office: Mackenzie Building 5146
- Office Hours: by email appointment
- Office Phone Number: 613-520-2600 ext. 4451
- Email: aahmadi@doe.carleton.ca

Teaching Assistant (TA) information

	Name	**Email**	**Office Hours (by appointment)**
1			**TBA**
2			**TBA**
3			**TBA**

Class and lab information

- Class Location: **Check the university scheduling/calendar**
- Class Time: **Check the university scheduling/calendar**
- Lab Time: refer to your schedule
- Pre-requisites, from the current Carleton University Undergraduate Calendars (https://calendar.carleton.ca/undergrad/courses/ELEC/).

General Regulations

- *Copyright on Course Materials*: The materials created for this course (including the course outline and any slides, notes, program source code, labs, projects, assignments, quizzes, exams and solutions) are intended for personal use and may not be reproduced or redistributed or posted on any website without prior written permission from the author(s).

- *Attendance*: Students are expected to attend all lectures and lab periods. The University requires students to have a conflict-free timetable. For more information, see the current Undergraduate Calendar, Academic Regulations of the University, Section 1.2, Course Selection and Registration and Section 1.5, Deregistration.

- *Health and Safety*: Every student should have a copy of our Health and Safety Manual. A PDF copy of this manual is available online: http://sce.carleton.ca/courses/health-and-safety.pdf.

- *Deferred Term Work*: Students who claim illness, injury or other extraordinary circumstances beyond their control as a reason for missed term work are held responsible for immediately informing the instructor concerned and for making alternate arrangements with the instructor and in all cases, this must occur no later than three (3.0) working days after the term work was due. The alternate arrangement must be made before the last day of classes in the term as published in the academic schedule. For more information, see the current Undergraduate Calendar, Academic Regulations of the University, Section 2.6, Deferred Term Work. Since students are required to have a stable and reliable internet connection, a poor internet connection will not be considered a sufficient reason to defer an online exam.

- *Appeal of Grades*: The processes for dealing with questions or concerns regarding grades assigned during the term and final grades is described in the Undergraduate Calendar, Academic Regulations of the University, Section 2.7, Informal Appeal of Grade and Section 2.8, Formal Appeal of Grade.

- *Academic Integrity*: Students should be aware of their obligations with regards to academic integrity. Please review the information about academic integrity at: https://carleton.ca/registrar/academic-integrity/. This site also contains a link to the complete Academic Integrity Policy that was approved by the University's Senate.

- *Plagiarism*: Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated.

- *Academic Accommodation*: You may need special arrangements to meet your academic obligations during the term. You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://www.carleton.ca/equity/

Intellectual Property

Lectures and course materials prepared by the instructor are considered to be an instructor's intellectual property covered by the Copyright Act, RSC 1985, c C-42. These materials are made available to you for your own study purposes and cannot be shared outside of the class or "published" in any way. Lectures, whether in person or online, cannot be recorded without the instructor's permission. Posting course materials or any recordings you may make to other websites without the express permission of the instructor may constitute copyright infringement.

Classroom Conduct

- The classroom environment is premised on commitment to the following:
  - Professionalism
  - Respect
  - Honesty
  - Privacy

- In the context of online/in-person learning, students are expected to:
  - Represent themselves honestly in all communications, applications, assignments, tests, examinations, and other correspondence.
  - Respect the need of others to work in an environment that is conducive to learning in an online setting.
  - Be courteous and polite in all electronic exchanges with instructor and fellow classmates.
  - Be active and engaged participants in the learning process.
  - Respect the personal information and privacy of others.
  - Respect all copyright laws.

- While participating in this online/in-person course, students are encouraged to engage in appropriate behaviors. Inappropriate behaviors may include:
  - Using email or login account information that is not your own.
  - Engaging in any behavior that may be disruptive to other learners in the online learning environment.
  - Writing, using, sending, downloading, or displaying any information that is hostile, insulting to others, derogatory, obscene, harassing, threatening or otherwise offensive.
  - Reproducing course content or reposting course materials without explicit permission.