TFE4152 Term Project

October 7, 2025

1 Instructions

In previous years, students have been required to write a full report (in a style similar to what is expected from you when writing your project thesis and master thesis). This year we are testing out something new by giving you a more direct and limited template. Our hope is that this will allow you more time to focus on the doing-part of the project.

Anything written in blue is an instruction on what you should include in that part of the document.

1.1 Non-technical Requirements

The project must be completed in groups of two (2) people. You have to sign up to one of the groups on Blackboard together with another student. Please find a partner BEFORE you sign up to the group so that we don't have to spend a lot of time removing people from groups they no longer intend to be in. Feel free to use the "Looking for a project partner"-forum on Blackboard to find someone to work with (either respond to an existing post in the forum or create your own post). You find a link to the forum under *Forums* on the TFE4152 Blackboard page.

The project consists of both an analogue and digital part. We encourage you to contribute equally to both parts, as the knowledge you gain from doing this project will benefit you both in the exam and in your future career as a circuit designer.

Both students are expected to invest similar efforts into the project. In case of particular problems, due to for example illness or other unforeseen circumstances, please bring it to the attention of course coordinator Snorre as soon as possible and we will address the issue. We are not able to do anything about it after the project has been handed in.

Doing the project alone is by default not accepted. If you feel there are special circumstances that should allow you to complete the project alone, you must contact the course coordinator (`https://www.ntnu.edu/studies/courses/TFE4152#tab=omEmnet`).

The document you hand in must be in English. This is to ensure that those grading your work are able to understand it.

1.2 Deadlines

Note that the deadlines are absolute. Handing in too late is an immediate fail, so do not wait until the last minute. It will be possible to upload your report several times, **so please upload a version of your work the day before the deadline to ensure that you have handed in something that can be graded.** This goes for both the midterm delivery and the final delivery.

Both deadlines are set to 2pm so that you will be able to contact the course

coordinator and/or Orakeltjenesten if you are having technical issues when trying to hand in your work.

Please make sure to communicate properly with your project partner, so that one of you actually hands in the project and that the version you deliver is the correct one.

1.2.1 Midterm Delivery

To be handed in on Blackboard, deadline Friday 24th of October at 2pm (14:00).

Should include:

- A brief reflection on your progress so far, and an overview of the remaining work. Which circuits do you have to design/implement? Which simulations are you planning to run? Are there any parts of the work that you have already done that you plan to improve/change?
- An implementation of the analog circuit in AIMSpice.
- Initial circuit simulations in AIMSpice.
- A short description of each group member's contribution to the work.

The midterm delivery does *not* count towards the project grade.

1.2.2 Final Delivery

To be handed in on Inspera, deadline Friday 21st of November at 2pm (14:00). This work will be graded A-F, and count as 30% of your final grade in TFE4152.

The final project report must follow the structure outlined in this document.

2 Introduction

- Welcome to our headquarters! And thank you for being able to come here at such a short notice! We've had some big projects come in lately, and find ourselves in a bit of a pickle as our lead engineer is on a three month long vacation to New Zealand with her wife. Now we have no way of finishing our new chip before the tapeout deadline in November! Unless the two of you can help us out? I've heard such great things about you from your professors at NTNU.

You look at each other and nod in agreement.

- Wonderful! You'll be trusted with creating two crucial parts of our new chip. I'll email you the details.

From:	ingcognito@email.com
To:	icdesigners@tfe4152.ntnu.no
Date:	September 29th 2025
Subject:	Project description

Hei! As promised, here are more details on the project.

Firstly, we need a wide-swing current mirror. We use a 180 nm transistor technology, and create all our designs in AIMSpice. I've attached more information about specs (see section 3), please read and follow that carefully!

Secondly, our digital design team has been struggling to solve a problem for some time. Fortunately, our team lead sent us a solution idea right before boarding the plane! It involved something called a Tsetlin machine, which you can choose to read more about here `https://www.literal-labs.ai/logical-ai/` and here `https://tsetlinmachine.org/`. Your professor told me that Tsetlin machines were also covered in a guest lecture 27/8/2025. All the details she sent us regarding specifications etc. are included in the attached document (see section 4).

This is an ongoing project, so as I'm sure you understand there may be updates to the spec at a later date. If that is the case, you will be told through an announcement on your TFE4152 Blackboard page.

Thank you for helping us out!

Best regards,
Ing. Cognito

3 Analog circuit design

Cascode current mirrors offer higher output impedance than simple current mirrors, and are therefore preferred in circuits where high output impedance is crucial. Unfortunately, this desired property of the conventional cascode current mirrors comes at the cost of a limited output voltage swing.

We need you to design a current mirror that offers higher output impedance, while still allowing a wide output swing. Your task is therefore to design a *wide-swing current mirror*.

The design has to meet the following specifications:

- Transistor technology: 180 nm (`https://analogicdesign.com/students/netlists-models/`)
- Design for low power consumption.
- Support an input current I_{in} ranging from 40 uA to 50 uA.
- This is a current mirror, so the output current I_{out} should follow the input current I_{in}. A small dc offset is allowable (but not desirable). The most important thing is that the relationship is linear (I_{in} increased by x => I_{out} increased by x).

You should aim to meet the circuit specifications for all the PVT-variations (see subsection 3.3).

To ensure that everything works as intended, your design must be properly tested. Use AIMSpice to implement and simulate your circuit.

3.1 Current Mirror Design

Include a schematic of your analog circuit, and describe how it works.

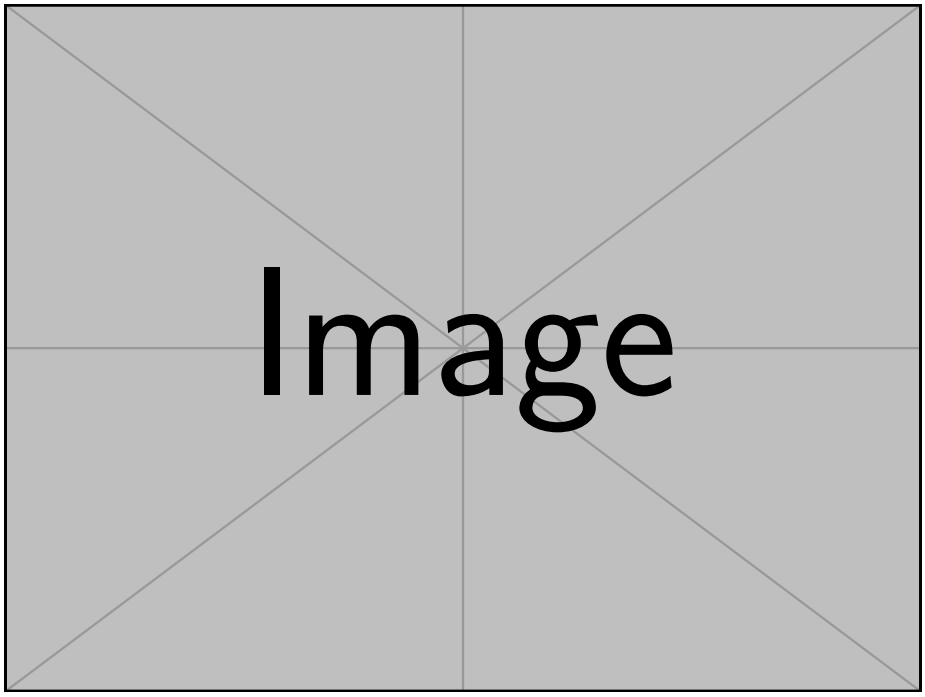

Figure 1: Schematic of the analog circuit.

Explain the design choices you've made. Feel free to include calculations.

What is the output resistance and minimum output voltage for your circuit? Show hand calculations.

3.2 Implementation

Create your analog circuit in AIMSpice, using the 180 nm technology.

Include your AIMSpice code here. Please format the code so that it is easily readable, e.g. using the LaTeX-package `listings`.

3.3 Circuit Simulation Results

Simulate power consumption and I_{out}, for at least three different values of I_{in} (including the upper and lower limit values).

Repeat the simulations for all combinations of the following PVT-variations:

- Process corners TT, SS, and FF.
- Supply voltage variation: -10%, 0, +10%.
- Operating temperature: 0°C, 27° C, and 50° C.

Explain how you set up the simulations. Include any code used to simulate the circuit.

Present the results from your simulation. Explain what the results show.

3.4 Discussion

Discuss the results from the AIMSpice simulations.

Did your circuit work as expected? If not, do you have any suggestions for how this can be corrected?

How was the circuit's performance affected by the PVT variations? Use theory to explain your results.

You were asked to design for low power consumption. Discuss your results, and whether there are other design choices you think should be made in future.

4 Digital Design - Creating a State Machine

"The Tsetlin Machine is not just another machine learning algorithm. It draws on principles laid down by Mikhail Tsetlin, a visionary Soviet mathematician, who in the 1960s explored a radically different path to artificial intelligence. Rather than mimicking biological neurons, Tsetlin's approach was rooted in learning automata and game theory. He recognised that logic—expressed through what we now call Tsetlin automata—could classify data more efficiently, forging a new direction in AI." (`https://www.literal-labs.ai/tsetlin-machines/`)

Design a two-action Tsetlin automaton. Since the Tsetlin automaton can be seen computationally as a finite state machine (FSM), it will be referred to as such in the rest of this document.

Here are the relevant specifications for your FSM:

- It should have 6 states: $\Phi = \{\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$.
- Transition from one state to another is triggered by the input signal β, which can be either $\beta_{Penalty} = 0$ or $\beta_{Reward} = 1$.
- The expected transition for a state ϕ_u, given an input β_x:
$$\phi_{next} = \begin{cases} \phi_{u-1}, \text{ if } 0 \leq u \leq 2 \text{ and } x = \text{Reward}, \\ \phi_{u+1}, \text{ if } 0 \leq u \leq 2 \text{ and } x = \text{Penalty}, \\ \phi_{u+1}, \text{ if } 3 \leq u \leq 5 \text{ and } x = \text{Reward}, \\ \phi_{u-1}, \text{ if } 3 \leq u \leq 5 \text{ and } x = \text{Penalty}, \\ \phi_u \text{ otherwise.} \end{cases}$$
- The output signal α will be either $\alpha_0 = 0$ or $\alpha_1 = 1$.
- Implement your finite state machine as a Moore Machine,
where $\alpha = \begin{cases} \alpha_0, \text{if } 0 \leq u \leq 2. \\ \alpha_1, \text{if } 3 \leq u \leq 5. \end{cases}$
- Design the FSM with low power consumption in mind. Both static and dynamic power consumption should be considered.

The FSM should be implemented in Verilog using ActiveHDL. You must write the verilog code on structural (logic gate) level, where all gates and their interconnects are identifiable. This is done to ensure that you actually simulate the circuit you have designed.

You do *not* have to write the testbenches in logic gate level Verilog. Here you are free to use high level constructions in Verilog, or simply the graphical user interface in ActiveHDL.

4.1 Construction of the State Machine

Construct the state machine described above by following the 7 step method (covered in a lecture on FSM). Describe what you do in each of the seven steps, and include relevant figures and tables. Justify the design choices you make.

4.2 Implementing the State Machine

The state machine should be implemented in Verilog. **Remember to write logic level code, i.e. code that directly describes the logic diagram / schematic of your state machine (which you created in subsection 4.1).**

Implement the register and the combinatorial block as two separate blocks, so that you can test these by themselves before you combine them to create the FSM.

Include your Verilog code here. Please format the code so that it is easily readable, e.g. using the LaTeX-package `listings`. If you have code in several files, make sure this is easily understandable from the way you format your included code.

4.3 Verification Plan for the State Machine

You must create a plan for how to test your state machine to verify that it works as expected.

Simulate the combinatorial part and the register separately, before simulating the whole state machine as one.

Describe your testbenches. How is stimuli applied? How do you check that the behaviour is as expected?

Include relevant code/pictures here, that show how your testbench is implemented.

Write your test cases in Table 1. Why have you chosen these test cases? Explain your thought process.

Table 1: Verification plan for the state machine.

Testcase	Circuit	Description	Stimuli applied	Expected output(s)
T0	Combinatorial	Lorem Ipsum	a=0 at t=0, a=1 at t=1	y=42 at t=3

4.4 Results from the State Machine Tests

Follow your test plan. Plot your results in Table 2. NB! Remember to use the same name for each testcase as in Table 1.

Table 2: Results from state machine testing.

Testcase	Expected output(s)	Observed output(s)	Pass/Fail
T0	y=42 at t=3	y=42 at t=5	Fail

Comment on your results. Are you certain that the circuit will function as intended based on these tests and results?

If any of your tests fail, discuss why that might be and propose a plan for solving the problem(s).

4.5 Discussion

You were asked to design the FSM with low power consumption in mind. Discuss the design choices you made to account for this. Do you have suggestions for any further improvements that could help lower the power consumption?