IN3170, spring 2023, mandatory laboratory exercise 1: Lab equipment and circuit simulation to analyse gate delay (deadline 26-Feb-2024, 10:15!)

P. Häfliger & Tonny N. D. D.
Institute of Informatics
University of Oslo
e-mail: hafliger@ifi.uio.no

January 31, 2024

change log	
29-Jan-2024	First release
31-Jan-2024	Corrected description of initial waveform setting of GPIBexample.m from ramp to square.

Abstract

For the following labs we are going to use electronics lab equipment as well as the advanced integrated circuits simulation tool in the software package Cadence. This first lab task is mostly to get acquainted with those tools and to use them for the first time for some circuit analysis.

Furthermore we'll get acquainted with the setup in the teaching lab, i.e. the instruments and the GPIB interface in Matlab

This is the first in a series of three lab tasks that will be graded and will count 40% towards the final grade. The first lab, however, will only be 'pass' or 'fail'. The students are required to pass this lab assignment to be admitted to the exam. The deadline in the title is a hard deadline! Do not miss it! Plan to submit well ahead of the deadline! We will use devilry.ifi.uio.no for submission of you lab report.

1 Report and Group Assignments

1.1 Requirements for the Lab Report (read carefully!)

You are required to execute the tasks and answer all the questions posed below and to submit a report on your work. The report needs to be explaining clearly what you have done, how you have done it, what the results were and what

you conclude from them. A good basic rule for any scientific writing is that a document should provide sufficient detail for someone reading it to be able to replicate the results that are presented. Make sure to answer all questions! Supply the report with drawings of the circuits (including the values of the components and parameters you used where appropriate, e.g. bias voltages/currents, component sizes etc.) and measurement setups! Show your measurements in graphs! Use labels in the schematics that you draw, such as M_1, M_2 (M is often used for labelling CMOS transistors), $opamp_1$, I_1, V_1 etc. You should then use those labels in your text, since it is much easier to write: 'transistor M_1 in figure 1' than 'the transistor third from the top and second from the left in the righthand side circuit in figure 1'. **MANDATORY when assembling a circuit in the lab (so not yet for this lab)**: Include a photograph of your circuit into the report!

1.2 Graded Mandatory Group Assignments

Note that this is part of the course's exam and strict rules apply as described in the document https://www.uio.no/english/studies/examinations/compulsory-activities/mn-ifi-mandatory.html . The page explains the significance of mandatory assignments in a course and in particular group assignments. It also specifies your responsibility to not plagiarize anybody else's work and that you are required to conduct and understand your own experiments and obtain your own results, while you are still allowed and actually very much encouraged to exchange advice and experiences also between groups.

Each group must deliver a written lab report using the Devilry online submission system **before** the **hard** deadline indicated in the title. Note that you can submit multiple times and the last submission before the deadline will be graded, so it might be a good idea to plan to submit preliminary versions well before the deadline. The points given for this lab assignment will determine if the lab assignment is accepted or rejected. You will need to pass this lab assignment in order to be admitted to the exam. The next two lab assignments will be weighted as 20% of the total score of the course, i.e. your final grade.

Each task is labelled with how many points it will contribute towards the score.

2 Task

2.1 Introduction

First you'll acquaint yourself with the lab equipment that can be interfaced from the MATLAB command line and by running MATLAB scripts and functions, via a GPIB bus with a set of MATLAB-functions designed for that very purpose. Furthermore you will be using the schematic design tool as well as the simulation tool of the circuit design software Cadence. And last but not least we'll be analysing gate-delay of inverters both in the lab and in simulation, both for

a physical IC in the lab, and a simulated inverter in Cadence. You will also learn some soldering skills to be able to connect the IC under test properly on a printed circuit board (PCB).

2.2 Tools

- **Linux command line interface**

  Both the server rh7login.ifi.uio.no and the computers in the course lab will run the Linux operating system, since the SW we'll be using runs under Linux. In Linux you'll be working with a command line interface to start the relevant software (MATLAB and Cadence). The command line interface is called 'terminal' or 'xterm'. On rh7login.ifi.uio.no you'll open a 'Terminal by choosing 'Activities' from the top bar menu and then click on the 'Terminal' icon. On the lab machines there is a 'console' terminal open when you log on that you can use. Else you can also choose 'Applications' in the top bar menu and then 'System Tools' and then 'Terminal'.

  When you open a terminal, your 'home directory' is the active directory by default. The active directory is the default directory where a program that is started stores any files it creates and where it is looking for files it needs first. Your 'home directory' is a directory on a file server where you can store your files and that is accessible from any Linux machine on campus. It also contains a directory called 'pc' which is the directory that is accessible by any Windows PC on the campus, containing your typical 'Desktop' and 'Documents' and similar Windows folders. You can, for instance, create a sub-directory 'IN3170' to work in with your labs, and possibly more subdirectories for individual labs to keep things organized. To make a directory use the command 'mkdir <directory_name>' and to change the active directory 'cd <directory_name>' or 'cd ..' to go 'up' to a parent directory.

- **MATLAB**

  Please refer to the online html-document https://www.uio.no/studier/emner/matnat/ifi/IN3170/v22/resources/oppstsettcadencehjemmefra.html for instruction on how to get a copy of matlab installed on your home computer. It IS also installed on rh7login.ifi.uio.no and on the computers in the course lab OJD5419. Under Linux, you simply type 'matlab' in a terminal to start it.

  We will be using MATLAB to control lab-instruments. Furthermore it's the best tool for plotting all of your results as nice graphs, not just for the lab measurements but also for simulation results from Cadence, and it's a powerful tool for doing maths, especially working on vectors and matrices, so it's perfect to analyse your data too.

  One useful function is 'help'. 'help <command_name>' will display an explanation on how to use '<command_name>'. Another help function

that helps you find functions that you do not know the exact name of is 'lookfor'. Type 'help lookfor' to learn more.

- **GPIB instrument interface in MATLAB**

  On the course web-page under 'resources' you'll find a file containing a MATLAB script function 'GPIBexample.m'. Copy it to your working directory in your home directory. If you open MATLAB on your lab (!) Linux computer from the directory where you have stored GPIBexample.m, you can look at the function in an editor when you type 'edit GPIBexample.m' or by double clicking it in the left column where the content of your active directory is displayed. Have a good look at the code! You can use it as a template to make your own matlab functions that run the instruments and collect results. You can also run that function by typing 'GPIBexample', but first turn on the 4 GPIB instruments at your work bench: the scope HP54622, the multimeter HP34401, the waveform generator HP33120 and the voltage source HPE3631.

  When you run GPIBexample it tests whether the GPIB card and drivers are properly installed on your computer (they should be!) and will give an error message if they aren't. It also calls a command that adds the search path (specifying the directories where matlab should look for functions other than the active directory) to contain the path to the GPIB functions: therefore you should run this function at least once after having started MATLAB! Short the 6V output of voltage source to the input of the multimeter, and the output of the waveform generator to a scope probe on the scope plugged in into channel 1. The matlab function will program the waveform generator to produce a saw tooth waveform of 2kHz and 2V amplitude (4V peak-to-peak). If shorted to the scope, you should see that waveform there on the display. It will be read from there into matlab and plotted in figure 1 on your computer. Furthermore, the function will program the voltage source to put out voltages from 0V to 1V in 0.1V steps on the 6V output. Those voltages are then read via the multimeter back into matlab. Figure 2 should then plot the intended output of the voltage source vs the measured voltage from the multimeter. If everything works out well, this should result in a graph with a straight line from (0,0) to (1,1).

  To get a list of functions you can use to talk to one instrument type for example 'HP33120' and then hit the 'TAB' key. This gives you a list of all functions starting with that string to choose for autocomplete. For a short help text type for example 'help HP33120_SetDCOffset'.

  Known issues:

  – Driver 'tnt4882' is not present on the machine. The GPIBexample-function gives an error message to that effect. Solution: reboot your machine with some of the lab devices (scope, waveform generator, multimeter etc.) switched on.

- Voltage source does not source the voltage it shows when controlled by GPIBexample. So the voltage is shown on the display, but is not measured by the multimeter. That's an error in the original version of GPIBexample: the voltage has not only be set but the output has to be turned on by HPE3631_operate. Alternatively you can also first manually turn the output on and only then run GPIBexample.
- Some waveform generators show double the amplitude and offset from what they are actually programmed to do. This is only apparent when you observe that output on the scope. So if you have such a waveform generator you simply need to half offset and amplitude in your programming.
- GPIB error message on the code line that tries to HP54622_Autoscale. This occures when the Scope displayed a 'low battery' error message and initiated a factory reset. After that the GPIB interface is no longer able to talk to the scope on its *default address*, since the factory reset sets that address to another number than what the MATLAB GPIB interface expects. You need to reset that number on the scope: hit the 'Utility' button, select I/O, then set the 'address' to 24.

- **Working with Cadence from home or a UiO-computer**

  In order to use Cadence and other SW that runs on a Linux server at IFI/UiO you will need the means of logging onto such a server and have a user interface to it presented on your home/UiO computer. I have made a HTML document that explains how to do that for Windows, Linux, and Mac PCs here: https://www.uio.no/studier/emner/matnat/ifi/IN3170/v22/materiale/oppstsettcadencehjemmefra.html

- **Cadence, "Schematics Editor" and "Analog Design Environment explorer" (ADE explorer) simulation tool**

  Cadence is an industrial tool for integrated circuit design. When using it for simulation, its models are much more detailed than the hand calculation models you learn about in the course. These models include a lot of secondary effects and have a lot more parameters and are able to predict with much higher accuracy how a circuit will really behave when coming back from production. Still you'll see that the hand calculation models are a useful approximation in many cases.

  Please also refer to the online html-document https://www.uio.no/studier/emner/matnat/ifi/IN3170/v22/resources/oppstsettcadencehjemmefra.html for an introduction on how to set up Cadence and how to use the "Schematics Editor" and "Analog Design Environment".

- **Soldering on a PCB** You will need the skill of soldering components and cables to a printed circuit board (PCB) for this and future labs.

There are pipes with air vents at each work place that suck in air: turn them on and suck in the fumes that occur during soldering. Small doses of those fumes are unproblematic but substantial continuous exposure over a long time to such fumes can be unhealthy.

When soldering any kind of conductor to a PCB pad you want to establish an electric contact between that conductor and the pad on the PCB surface. To make a proper contact you need to heat BOTH the pad AND the conductor, so try to touch BOTH with the soldering iron. The solder tin only need to touch one of those two and will melt automatically if that surface becomes hot enough. So I usually hold the soldering iron to the conductor and try to place the tip on the pad on the PCB, while I hold the solder to the conductor from the opposite side. Then the solder will liquefy automatically as the conductor becomes hot enough and the connection between those two is thus guaranteed a good connection. If the pad becomes hot enough as well you will notice the liquid solder 'flowing out' onto the pad. In contrast, if you do not heat the pad enough the solder may look like it touches the pad as a bobble 'tucking in' at the contact point instead of flowing out. It may still touch and may establish a finicky contact, but it is not solidly 'glued' to the pad, may be unreliable and can get loose.

You'll find some good video on this on Youtube, for example: https://www.youtube.com/watch?v=vAx89WhpZ3k

- **IC CD4007UBE with 6 transistors** There will be generic PCBs in the lab where you shall solder on an IC labelled 'CD4007UBE' and some resistors and where necessary some cables. The IC contains 6 transistors and some interconnections. Check the data sheet 'cd4007ub.pdf' under 'resources' on the course 2024 webpage. We will only need a single nFET from it for lab 1, so pay attention to the interconnections already on the IC: you cannot use any of the three nFETs as a single transistor.

2.3 Lab Tasks

task 1 (2p), set up lab equipment: In short: run an example function in matlab and verify the proper function and control of the lab equipment. Log on to the lab Linux PC, create a work directory and copy the file GPIBexample.m into it from the 2024 course web-page under 'resources'. Type 'edit GPIBexample.m' or double click it in the left hand side file browser to open an editor in MATLAB and look at the code! You shall edit or create such code in the following to control the lab devices, so you should find some useful examples in this code.

Short the Waveform generator output to a scope probe connected to channel 1. Then short the 6V output of the current source to the multimeter. Start MATLAB in that work directory containing GPIBexample.m (!) and run the function GPIBexample.

The first few commands in that function check the proper installation of the GPIB drivers and set the path in MATLAB to find the functions to control the lab equipment. This only needs to run once per lab session (per login) to set the MATLAB path. That part is not really needed in any future functions you might create just to control the instruments, so you can leave that out, but it's a good idea to run GPIBexample (or just the initial part) again if you enter the lab anew and sit down in a work place for the first time, log on and start working. (No need to hook up the voltage sources, multimeter and waveform generator, scope for that each time: this is just to run the initial test for the driver installation and path setup in MATLAB.)

Then the function executes some measurements and produces two plots: it sweeps a voltage from the voltage source and measures it with the multimeter and plots that I/O relationship, which should be the 'identity' function if everything works, and it programs the waveform generator to provide a 'square' waveform and records it on the oscilloscope, downloads the data from the oscilloscope and plots it in a separate figure. Include those two first plots into your report to illustrate that you got your lab setup to work as it should.

task 2 (2p), change the GPIB example: In short: make your own MATLAB function to generate a ramp of 5V peak-to-peak amplitude at 1kHz from the waveform generator and program the voltage source to provide a constant 5V supply voltage.

Edit the function GPIBexample. You may store it under a new file name. If you do also change the function header to have the same name as the file name. (The latter may in fact not be necessary, as MATLAB uses the file names to identify the function and not the function header, but it's good practice to keep the function header and file name consistent.)

Edit it such that the waveform generator produces a ramp (sawtooth) wave with 5V p2p amplitude and 1000Hz frequency and goes from 0V to 5V (not -2.5V to +2.5V!), and that the 6V output of the voltage source simply puts out a constant 5V. You do not need to verify the constant voltage with the multimeter, so you can drop the second figure from the code. Make sure that the scope triggers it's recording on the rising flank at about 2.5V. You may later also want the scope to zoom in on the x-axis, i.e. that the recording only covers a short time span after that trigger (see task 4), but for now set the recording length appropriately that you will see about two full cycles of the 1000Hz. You can do this either with the appropriate GPIB command (recommended to get a consistent configuration should you repeat the measurement!) or using the knobs on the scope. Put the plot of the ramp waveform from the scope into your report.

In short: solder together a PCB with the CD4007UBE IC and a resistor to form a common source amplifier and verify that it works.

task 3 (4p), observe a common source amplifier

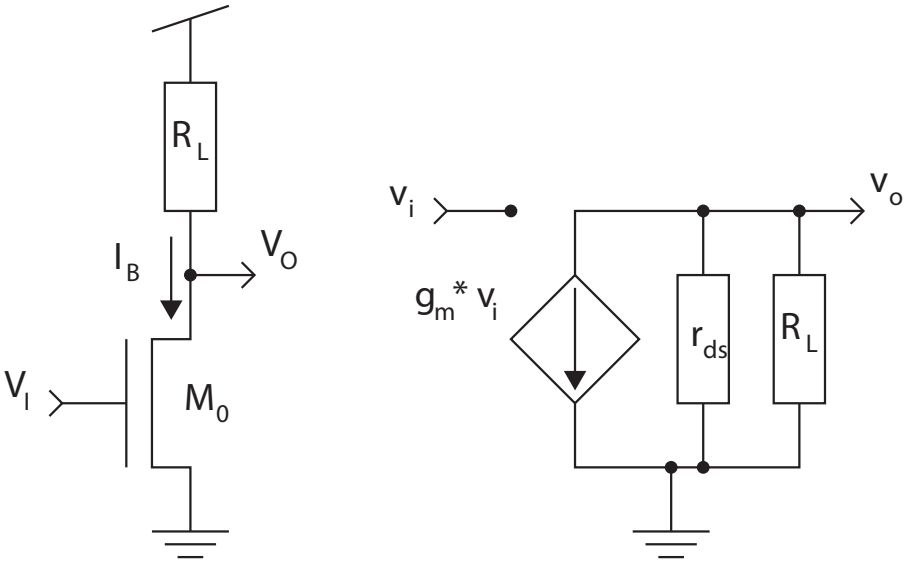


Figure 1: CS amplifier circuit and its small signal equivalent circuit.

Assemble a PCB with the 'complementary inverter' chip CD4007UBE using just one of the nFETs on it with its drain connected to a load resistor of 100kΩ (If 100kΩ will result in a inconvenient point of operation you are allowed to change that value, but report it!) to make a CS gain stage (left of fig. 1). You'll find a data sheet CD4007UB.pdf on the course 2024 web page under 'resources'. Make a plan of what to put where on the PCB before you start soldering!

Before you connect signals to the assembled PCB, it is good practice to have the voltage source's outputs turned off! Connect your circuit first and only then turn the voltage source's outputs on. Always turn the power supplies (and biases, if applicable) on first and then the input signals. Sometimes the order of powering up is important and might cause damage, if not done properly. In this first example it is not terribly critical though.

Connect the Gnd and 5V from the voltage source's 6V output to power the IC and CS amplifier appropriately. Connect the waveform generator output to the nFET gate input and the scope's channel 2 (!) to the CS amplifier output. Leave channel 1 connected to the waveform generator/CS input too, so that you can observe both input and output at the same time! Then turn the 5V power supply on and then activate the waveform generator output (still with the ramp setting from task 2). Observe the input and output and make a plot for your report.

task 4 (4p), measure CS amp point of operation and gain In short: You shall provide a slow ramp input to a CS gain stage, and find the point of operation and gain.

From the plot of task 3 find a rough point of operation for the input voltage V_I. Then program the waveform generator to have an offset that is V_I and an amplitude such that the output just about stays in the 'linear' range, i.e. appears about straight and going almost, but not quite from Gnd to Vdd, and starts to 'bend' towards Vdd and Gnd at the bottom and top. Adjust your estimation of V_I somewhat if necessary to have the plot nicely centred with the point of operation (V_I, V_O) nicely in the middle of the plot. Let us define $V_O := \frac{\text{Vdd}}{2} = 2.5\text{V}$. Put that plot into the report! Extract the gain A from the plot. If necessary, reduce the amplitude of the input waveform even more to get enough 'zoom in' to measure A with good accuracy. If you do, put that plot into the report too!

2.4 Simulation Tasks

In the following, to plot the curves and to mark important points in the graph, export them to MATLAB!

Task 5 (4p): In short: simulate an common source gain stage with a minimum size nFET and the same load resistor as in the previous physical experiment in the TSMC 65nm technology and again produce the same set of plots and measure the resulting point of operation and gain. Do the curves look similar. What are the similarities and differences to the physical experiment in the lab?

The normal transistor models to use in this technology's process development kit (PDK) are called 'nch' and 'pch' respectively and are in library 'tsmcN65'. You'll want to conduct a 'trans' simulation to observe temporal dynamics of your inverter. Use the cell 'vpulse' in library 'analogLib' to generate a ramp input waveform, and cell 'vdc' in the same library to provide the power supply of 1.2V (for this 65nm technology).