ⁱ Cover page

Home Exam IN3170/IN44170 V2020

This is the front page for the exam for the lecture 'Microelectronics' IN3170/IN44170 for the spring semester 2020 with some general information. **It is important that you read this information carefully!**

Exam date and duration

- The exam is disclosed 27-May 14:30 and submission closes 4-June 14:30
- The submission needs to comprise a ZIP-file containing a Cadence library (see task 1 for instructions) as well as a Text file (Word or PDF format, see task 2 for instructions)
- Remember that the exam should be anonymous and no submission should contain any reference to your name.
- All aids are allowed. It's up to the candidates to consult relevant resources, assess quality and to compose responses based on your own knowledge and treatment of the material. Submitted material must reflect your own level of understanding and skill.
- You are required to read the IFI rules regarding cheating
- You are required to read the UiO's guide for uploading files.

Rules for cheating on exams

The general information on cheating is also published here.

The exam must be the result of the student's own efforts. It is okay to discuss theory and the assignment text with others. It is also okay to get hints on how to solve a task, but this should be used as a basis for your own solution and not copied unchanged from another student. Sharing code or (parts of) solutions to other types of assignments is not allowed. In particular for THIS course's exam you shall draw your own schematics in Cadence and not copy any Cadence library files from other students. If you include text, program code, illustrations or other things from the internet or elsewhere, you must clearly mark it and state where it comes from - in an independent answer to an exam assignment, this should rarely occur.

Questions during the exam

For questions during the exam use the courses MS Teams general chat channel for the Team 'IN3170/4170' such that any communication is visible to all candidates. Use it if anything is unclear with the exam in relation to exam tasks and instructions, or for technical issues with Cadence. For issues with Inspera you can send a mail to studieinfo@mn.uio.no. The last questions to receive a guaranteed answer on the chat channel need to be submitted 24h before the final submission deadline of the exam!

Illness at exams / postponed exams

See here for more information.

During the spring semester 2020, UiO allows self-notification in case of absence from exams. If you are ill or in quarantine during an exam, you can send us an online message without attaching documentation.

If you are ill or have another valid reason for not taking an exam, your absence will be approved. With an approved reason for absence, you do not use an examination attempt, and you can apply for a postponed examination or an extended submission deadline. In case of a postponed exam this may be decided by the lecturer to be another home examination or an oral exam instead.

Good luck!

1 Cadence Simulation

Cadence Simulation Task (Total Weight: 70%)

⬆

Upload ZIP-file containing your Cadence library

Følgende filtyper er tillatt: **.zip**   Maksimal filstørrelse er **2 GB**.

⌷ Velg fil for opplasting

Overview

For this task you shall download a ZIP-file containing a Cadence library and some MATLAB functions. You shall unzip the library in your Cadence working directory that you have established for lab 3 (TSMC 65nm technology, consult lab3 if for some reason you do not yet have a Cadence setup and working directory) and edit the Cadence library, adding a circuit schematic that implements an electronic circuit with a function specified in the text below. The MATLAB functions can be used to evaluate the performance of your circuit. Once you are satisfied with your circuit's performance you shall zip the entire library again and upload it here as your final result.

Download and install the Cadence library

Download the Zip-file "IN3170_exam.zip" by clicking on the name! Log onto rh6login.ifi.uio.no, go to your Cadence TSMC65nm working directory you have established for lab3 and unpack the ZIP-file there using "unzip IN3170_exam.zip". There should now be a directory called IN3170_trial_exam in your Cadence working directory. Start Cadence (refer to the description in lab3 if you do not remember how). Open the Library Manager (Tools -> Library Manager). To make the directory visible as a Cadence library in the library manager, you have to click 'Edit -> Library path ...'. A pop-up Window appears specifying the paths to all the visible Cadence libraries. You need to add the new library by writing its name 'IN3170_exam' in the first column, and the entire path to the directory in the second column, i.e. '<path of your Cadence working directory>/IN3170_exam'. Click on 'save' and exit the library path editor. Now the library should appear in the library manager. It contains two cells: 'dual_comp' and 'dual_comp_testbench'.

Creating your schematic

'dual_comp' has only a 'symbol' view. It is will be your task to **create a 'schematic' view for it using only standard transistors (!)**, i.e. cells 'nch' and 'pch' from library tsmcN65. You can test your cell by **simulating the cell 'dual_comp_testbench' for 128us (<u>Note: the duration is different from the trial exam!</u>),** i.e. in a 'trans' simulation in ADE L . Do not edit the cell 'dual_comp_testbench'!!! It creates appropriate test signals for your schematic 'dual_comp'. You circuit will be evaluated in the original testbench for your score. The testbench also provides bias voltages pBias and nBias. If you use them as gate to source voltages to transistors of the same dimension as the transistors in the testbench, they will provide 1uA bias currents. You may or may not need them in your circuit.

If you do have a schematics that is working reasonably well, it is recommended that you make a copy of that cell, for instance nameing it 'dual_comp_v1', before you continue working with the original cell 'dual_comp'. This is in case you do actually not improve the performance but make it

worse instead. Then you can easily go back to earlier versions. Before submission make doubly sure that your **final version is in the original cell named 'dual_comp'!!!**

The specifications for your circuit

Your circuit is an example of a mixed-signal circuit, in this instance with analog inputs and a dgital output. It shall compare three analog voltage input signals, 'in1',' in2' and 'ref'. The output 'bothH' shall be digital and be high only if both 'in1' and 'in2' are higher than 'ref', i.e. it implements this expression: 'bothH=(in1>ref)&(in2>ref)'.

The testbench sweeps 'ref' from 0V to 1.2V and back again during the 128us simulation time. In that time 'in1' sweeps up and down 40 times and 'in2' 800 times. So your circuit will compare a lot of the possible input combinations where all three inputs span the entire range between the supply rails.

A typical application for such a circuit may be in a sensor node that is in sleep mode (i.e. a low power mode) but still observing sensor input to look for a 'wake up' condition. When this condition is met it will 'wake up' and for instance turn on more sensors, do a full ADC of the sensor values and perform some digital analysis and send a wireless transmission etc.

Score 1 (Weight: 50%): Comparison Error

For some times in the simulation your output will be wrong. Difficulties you may encounter are for instance that your comparisons have a small offset error, that the comparison does not work for the entire input range of between [Gnd,Vdd], that your output may not switch instantaneously and might have a small delay, that your output signal is oftentimes not close to Vdd or Gnd, i.e. not nicely digital etc. The matlab function described below computes the percentage of simulation time during which your output is wrong, so the lower its 'Err' result the better. Results to strive for are towards 0.1 (i.e. 10% error) or even better.

Score 2 (Weight: 20%): Power Consumption

Often mixed signal circuits like this one are intended for wireless sensor nodes that run on battery or even on power harvesting. So it's often important that the power consumption is kept low. So the second figure of merit (FOM) for this task is the power consumption, as computed by the result 'pow' by the matlab function. It includes the bias generating circuit that consumes 2uA@1.2V so it will be a number that is at least 2.4uW and likely a couple of times larger than that. Results to strive for are less than 20 Micro-Watts. However, this is less important here than the circuit working correctly (weighted only 20%!). So rather concentrate more on Score 1.

Score 3 (Weight: 30%):

See the PDF deliverable, i.e. the next task in Inspera.

Computing error and power consumption with MATLAB functions *.m in 'IN3170_exam'

There is a matlab function 'evaluateLocalFilesDualComp.m' in the directory 'IN3170_trial_exam' that can compute the difference between the ideal output signal and the real output signal from your simulation, as well as the average power consumption of your circuit during the simulation. Note that the other *.m files in that directory are needed by 'evaluateLocalFilesDualComp.m', and need to be in the same directory where you execute it. You can use it as follows:

- for the 128us transient simulation select (at least) the voltage signals 'bothH', 'in1', 'in2', and 'ref', as well as the current output at the plus-terminal of the vdc connected to Vdd 'V2/PLUS' to be stored and plotted.

- once you complete a 128us transient simulation you can type (copy-paste) these commands into the main Cadence/Virtuoso window:

```
ocnPrint(v("/bothH" ?result 'tran) ?numberNotation 'none ?output "bothH.ocp" )
ocnPrint(v("/ref" ?result 'tran) ?numberNotation 'none ?output "ref.ocp" )
ocnPrint(v("/in1" ?result 'tran) ?numberNotation 'none ?output "in1.ocp" )
ocnPrint(v("/in2" ?result 'tran) ?numberNotation 'none ?output "in2.ocp" )
```

ocnPrint(i("/V2/PLUS" ?result 'tran) ?numberNotation 'none ?output "Isupply.ocp")

This will create 5 *.ocp files containing the simulation results in your Cadence working directory.

- Run MATLAB in a directory where you place those files together with the MATLAB functions '*.m' from the library directory IN3170_exam. (So for example you can copy the *.ocp files into the IN3170_exam directory and run MATLAB there, or you can copy both the *.ocp files from your Cadence working directory and all *.m files from IN3170_exam into another directory where you run MATLAB).

- Run the matlab function '[Err, Pow] = evaluateLocalFilesDualComp( )'. This will read the signal files '*.ocp' and compute two figures of merit. It's important that the signal files have the expected names for this to work! As mentioned above, 'Err' is the fraction of the simulation time during which the output is wrong as compared to the ideal output, and 'Pow' is the average power consumption of the circuit as derived from the output of the vdc that supplies Vdd.

Packing and uploading your final result

Once you are done upload your Cadence library. You need to ZIP that directory before uploading. On your cadence host machine rh6login.ifi.uio.no in your Cadence working directory you can execute the command "zip -rv IN3170_exam.zip IN3170_exam/" to do this. make sure that this works correctly!!! You can check this by copying your zip-file into another directory, unzip it there and verify that it has the same content as your working IN3170_exam directory.

Maks poeng: 0.7

2 Explain Your Circuit

Explain your Circuit and its Performance (Weight: 30%)

⬆

Upload explanation (PDF or Word format)

Følgende filtyper er tillatt: **.pdf,.txt,.doc,.docx**   Maksimal filstørrelse er **2 GB**.

📂 Velg fil for opplasting

In the previous Cadence simulation task you submit a circuit that compares three analog inputs and provides a digital result. Explain in a PDF or Word document how your circuit works. The explanation should at least contain these following elements (but can contain others!). Be concise, i.e. short (but not too short) and to the point!

1) Explain the operation principle of your circuit.

2) Explain where and when your circuit fails and why it's output is not always correct.

3) Are there different regions of operation where its performance is better and where it is worse? How do these regions of operation come about?

4) What have you done to optimize its performance? Are there certain parameters that you have tuned? Are there any extra elements that you have added to enhance performance?

5) Did you have to make trade-offs between the error score and the power consumption?

Answer with about 1-2 pages of text. Feel free to add figures for explanations. The figures do not count towards the text limit.

Maks poeng: 0.3