# MICRO432
## Exercises and Laboratories

a.y. 2021/2022

Mariagrazia Graziano, Gianluca Piccinini, Edoardo Charbon

Giovanna Turvani, Yuri Ardesi, Giuliana Beretta, Fabrizio Mo, Marco Vacca

MICRO432 TAs

# Exercise set n. 13: Magnetic Nano Circuits

## 1.1 From micromagnetic simulations to high-level simulations

From the previous lab exercises you should have understood how much time is required to simulate also the smallest NML circuit with physical simulators. Through micromagnetic simulators it is possible to understand how devices exactly behaves from a physical point of view; they are based on complex equation and for this reason, they are quite expensive in term of computation time. To understand the logic behavior of circuits, high level models can be designed extracting information from low level simulations. Behavioral models allow to test the logic behavior of very complex architectures with reasonable timing costs. In this second part of the lab you will learn how to design NML circuits with MagCad and how to simulate them using CMOS switch-level simulators.

## 1.2 Design a simple circuit with MagCad and extract its VHDL description

In this first exercise you will learn how to use MagCad; with this software it is possible to design NML circuits through a graphical user interface. The circuit you are going to design is depicted in Fig. 1.1.

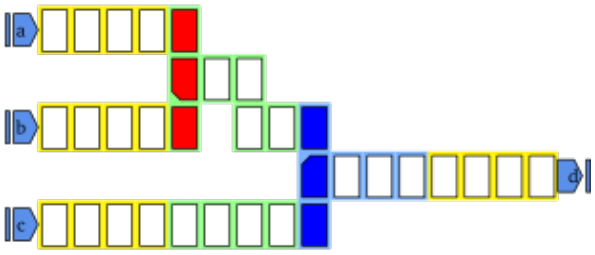


Figure 1.1: Sample Circuit

Run MagCad and, from the drawing settings windows, select all technological constraints as depicted in Fig. 1.2 and press Close.

Now you can see the main window of MagCad, click on any of the FCN items and then click on the empty tab to release your component. To change the clock zone, select a group of elements and press on "1", "2" or "3" from the clock zone menu (bottom-left of the main window). Once you have designed the circuit shown in Fig. 1.1, exactly as it is, you have to mark the inputs and outputs of your architecture. In order to do this, click on the **IN** icon (top-right) and enter the input name through the I/O menu (Fig. 1.3), then click on the main window where you want to place the input/output. The arrows let you choose the direction of the input/output signal.

Repeat this process for the other inputs and outputs (remember to select the proper settings and names in the Menu shown in Fig. 1.3). Once you have designed your architecture marking all inputs and outputs, you are ready to extract your component. Click on **File→Export Component**, insert the name of the component, choose a saving location and press **Save**. The name of your architecture MUST BE **ex1**. In the next window set up all the settings as depicted in Fig. 1.4 and click on **export**. In the next window leave all the parameters as they are and click on **OK**.

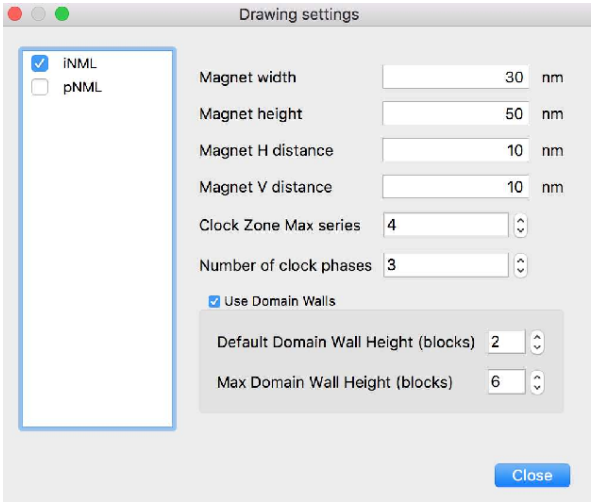


Figure 1.2: NML Settings

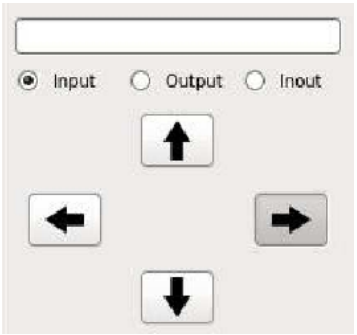


Figure 1.3: I/O Menu

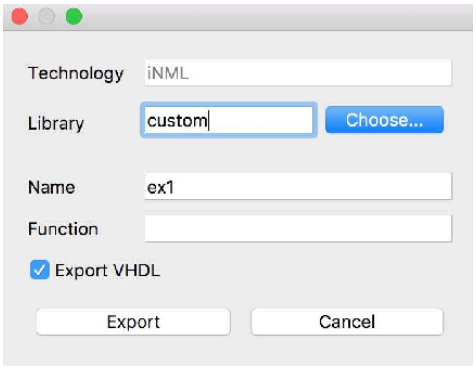


Figure 1.4: MagCad Export Menu

Now, you VHDL file has been extracted and it is ready to be simulated with Modelsim.

**MAGN_FOCUS1_Q1:** What is the logic function performed by the circuit you have just designed?

**MAGN_FOCUS1_Q2:** Sketch pen& paper the expected behavior and waveforms with the hypothesis of a three phases system and in the case of initial inputs a=1, b=0, c=1, that after 3 clock cycles change to a=1, b=1, c=0.

### 1.2.1 OPTIONAL: Simulate your NML circuit with Modelsim

Now that your architecture has been described in VHDL format, you have to test it with the simulator *Modelsim*. Create a new folder for the simulations and copy all the files generated by MagCad. You will notice that also a testbench templates has been created, modify it according to your needs.

Remember that if you have changed the name of your custom architecture you have to change the component's name also within the testbench. Within the generated testbench you can identify the declaration of the component:

```
component  iNML_custom_ex1 is
port(
        d: out std_logic;
        d_param: out param_data := (others => 0.0);
        A: in std_logic;
        A_param: in param_data := (others => 0.0);
        B: in std_logic;
        B_param: in param_data := (others => 0.0);
        C: in std_logic;
        C_param: in param_data := (others => 0.0);
        CLK: in std_logic_vector(0 to 2));
end component;
```

And the related *Port Map*:

```
DUT: iNML_custom_ex1
        port map
        (

        d => d,
        d_param => open,
        a => a,
        a_param => open,
        b => b,
        b_param => open,
        c => c,
        c_param => open,
        CLK => clk_i
        );
```

In this case the component name is **iNML_custom_ex1**, you can substitute this field with the name of your architecture. Instead, **DUT** indicates the name of the **iNML_custom_ex1** instance. The library_inml.vhd contains all technological information, open the file and have a look at the listed constants.

Now we want to use the ex1 file within a so called "test bench": **ex1_TB.vhd**. The test bench contains an *instance* named **DUT** of the **COMPONENT iNML_custom_ex1** which is first declared (keyword **component**) and then instanced (keyword **Port Map**). Remember that the assignment

**You must complete the last part of the testbench inserting the input stimuli for the signals. You can use for example the following statement**

```
a <= '0', '1' after 10 ns, '0' after 40 ns,'1' after 70 ns;
```

that creates a signal waveform for the signal named **a**, add additional values if you want to test more input values (use a time separation of 30 ns between one time step and the next).

Now let's proceed with the simulation. We use Modelsim by Mentor Graphics in these lab exercises. To launch the simulator left click with your mouse on the start menu icon and then type **modelsim**. Click on the version of Modelsim found by the system to launch the simulator.

The operations needed for the simulations are:

- Go to **File** → **New** → **Project**
- Choose a name for the project easy to identify (like *ex1*, leave all the other parameters as default, then select **Ok**.
- In the window that appears select **Add Existing File**.
- Select **Browse** and then select the VHDL files created by MagCad and select **Open** and then **Ok**.
- All VHDL files should appear in the Modelsim Project Windows. Right click on **definition_inml.vhd**, select **Compile** → **Compile Selected**. The result of the compilation is written in the directory **work**.
- Now compile the other files in a hierarchical sequence from bottom to top (**definition_inml.vhd** → **library_inml.vhd** → **ex1.vhd** → **ex1_TB.vhd**). You can use either the menu, as in the step before, or type the equivalent command manually in the command window: for example `vcom ex1.vhd`.
- When it is finished open the simulation window (Simulate→Start Simulation). Choose a simulator *resolution* of 100 ps and *add* the design **work/tb_inml_custom_ex1** (which is the configuration name in the test bench). Then click **load**. Disable the optimization option. The design is now ready for simulation.
- If you want to see the waveforms type the following command

  ```
  add wave *
  ```

  in the command window. A new waveform window opens. You can also play with the menus to decide the waveforms you would like to display. Obviously there's nothing to plot until you don't run the simulation!
- To do so type

  ```
  run 2.5 ns
  ```

  which means that you start the simulation from 0 to 2.5 ns. The step for the simulation is 100 ps as you chose before. The time unit defaults is nanoseconds. But you can choose another value. Try for instance to type again

  ```
  run 1000000 ps
  ```

**MAGN_FOCUS1_Q3:** OPTIONAL Look at the waveform window. Is the result coherent with what you expected by the function implemented by the circuit?

**MAGN_FOCUS1_Q4:** OPTIONAL Change the inputs to the same you tested manually before and verify he behavior, and if different w.r.t. what you sketched discuss the reason.

**MAGN_FOCUS1_Q5:** OPTIONAL Ananlyzing the files structure, identify what MagCAD hase

operated and which is the organization of the files and the model used of each subcomponent included in the **ex1** architecture

## 1.3 Design your custom Half Adder

In this exercise you will have to design a custom Half Adder with the NML technology. In Fig. 1.5 is shown the logic function you have to design.

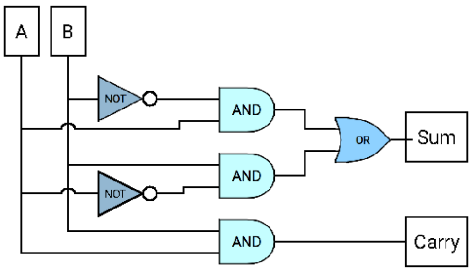


Figure 1.5: Half Adder

To properly design your architecture remember to follow these simple rules:

- Each logic gate (AND, OR, INV ) must be placed at the beginning of each clock zone
- Each clock zone must a width equal to 4 nanomagnets (no more than 4 nanomagnets can be put in cascade horizontally).
- No more than three nanomagnets can be put in vertical.
- In this structure, where there are no feedbacks, the clock zones order must be respected (1, 2, 3).
- SAVE YOUR DESIGN AS *ex2*, this will simplify the simulation phase.
- Extract the VHDL files following the same steps explained for the previous exercise

**MAGN_FOCUS2_Q6:** What is the NML circuit that you obtain? Report the NML schematics of the Half Adder circuit.

**MAGN_FOCUS2_Q7:** Which are the subelements that you expect to find in the extracted equivalent model? Explore the generated architecture and analyze the structure: is it what you expected?

### 1.3.1 Test the half adder

Once you have designed you circuit extract it and simulate it with the provided testbench. To test your architecture, create another simulation folder and copy inside it the files from the directory. As done before, modify the generated testbench file and test your half adder with modelsim.

**MAGN_FOCUS2_Q8:** OPTIONAL Plot some meaningful waveforms and analyze them. Is the logic behavior correct? Can you identify the important output changes?

# MAGCAD/ToPoliNano installing instructions

ToPoliNano is a framework and a CAD tool, invented and developed at the Politecnico di Torino. The framework is composed of two separate programs: ToPoliNano and MagCad. Details on these two programs can be found on the official website:

https://topolinano.polito.it

The procedure the download and install MagCAD/ToPoliNano is the following:

- Connect to the official website https://topolinano.polito.it/register/
- Register an account by inserting your personal information and a valid e-mail address;
- Once pressed the "Register" button, an e-mail will be sent to your inbox folder. Activate your account by pressing the link in the e-mail;
- If you didn't receive any e-mail, check your spam folder.
- Go to section Downloads of the website and obtain your free copy of MagCad and ToPoliNano for your OS. The installing should be immediate.

In case of troubles, the documentation can be found here: https://topolinano.polito.it/documentation/

In case you are a Linux Use and are not using a Fedora distribution, but Ubuntu or Debian or Mint distribution you can use the tool Alien to transform the .rpm in a .deb package and then you can install directly the package using *dpkg -i xxx.deb*

## MAGCAD test

In order to verify whether the installing procedure has been correct please execute the following steps:

- Open MagCAD
- The window "Drawing settings" should show up, press Close;
- Ignore the possible warning "no technology selected" by pressing Close;
- Chose File ¿ Open, navigate to the directory where you saved the given files and select the file "inml test.qll". The iNML circuit should appear in the main Window, Figure 1.6;
- Select File ¿ Export Component. Do not change anything, just press Export;
- If everything is ok, you should see the message "Component has been correctly exported!", by pressing Close, a second message should show up: "Vhdl generation correctly performed". Press Close;
- Eventually, some *.vhdl files should have been created in your user folder, typi- cally similar to: C:/Users/USERNAME/MagCADFiles/VHDL/inml test/.

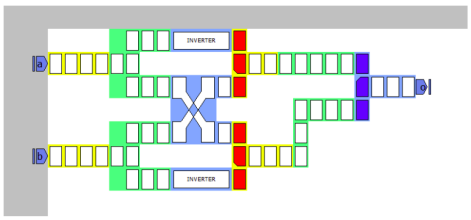


Figure 1.6: MagCAD circuit example

# OPTIONAL: Modelsim installing

ModelSim is an environment developed by Mentor Graphics, for simulation of hardware description languages such as VHDL and Verilog. Details on this program can be found on the Intel official website:

https://www.intel.it/content/www/it/it/software/programmable/quartus-prime/model-sim.html

The procedure the download and install ModelSim is the following:

- Connect to the Intel Download Center for FPGAs website

  https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
- Select **Version 20.1 Lite Edition**
- In the new page check the version and the operating system, then click on the tab **Individual Files** and click on **ModelSim-Intel FPGA Edition**
- If you are not registered, fill the form with your personal data to create an account
- Download the ModelSim program
- Launch the executable file and follow the installation process

## OPTIONAL: Modelsim test

In order to verify whether the installing procedure has been correct please execute the following steps:

- Open ModelSim;
- A window named “IMPORTANT Information” might appear, close it by pressing Close;
- Create a new Project with File > New > Project...
- Fill the appeared form by inserting the name “tb test” and the Location you prefer. A new window will appear, select “Add existing file”. Navigate to the folder where you saved the given files and select the file “tb test.vhd”, then press OK.
- The file will appear in the ModelSim window, right-click on it and do Compile > Compile selected.
- Now we have compiled the file, we have to simulate it. Choose from the nav Simulate > Start Simulation.
- The window Start Simulation will appear. Expand the work directory and select the tb test.vhd file. Then press OK
- In the bottom part of the ModelSim interface, there is a console (Transcript). Type “add wave *” and then “run 50 ns”
- Figure 1.7 reports what you should see after a few seconds.

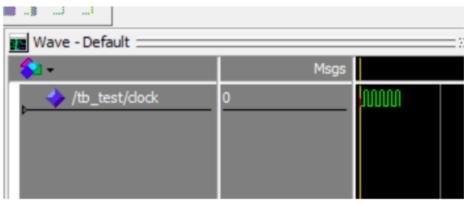


Figure 1.7: Modelsim output example