UNIVERSITÀ
DEGLI STUDI
DI TORINO

INFN

Istituto Nazionale di Fisica Nucleare

Advanced Electronics Laboratory / Part I

Dott. Luca Pacher

luca.pacher@cern.ch

University of Torino

A.Y. 2019/2020, Spring 2020

Course introduction and overview

Introduction

Part I - Dott. Luca Pacher

- introduction to FPGA programming using Xilinx Vivado and Verilog HDL
- simulation of analog circuits using LTspice (advanced topics)
- 40 hours (4 CFU)

Part II - Prof. Michela Greco

- introduction to micro-controllers programming using Arduino
- introduction to LabView programming for Data Acquisition (DAQ) systems
- 20 hours (2 CFU)

Exam

Also the exam is split into two parts :

- one grade for each part
- the final grade will be the **weighted mean** between the two

For the first part :

- **design and simulation** of a small digital system using Verilog HDL and Vivado (max. 2 students per project)
- **description of your work** in form of a **short report in English** (4-5 pages) using LaTex, *"paper style"* (as it happens in the real research life)
- **oral presentation of your project** with slides, *"conference style"* (as it happens in the real research life) followed by a few questions

For the second part, details from Prof. Greco

To make you feel like a "pro" ...

Developement and test of the front end electronics for a Hamamatsu S13360-3050 Silicon Photomultiplier

Cecilia Borca, Pierangelo Di Crescenzo, Alberto Occelli, *University of Turin - Department of Physics*

***Abstract*—The aim of this project is to design and build a proper readout circuit for the Hamamatsu S13360-3050 MPPC. This silicon photomultiplier will produce analog signals which will be sent to a preamplifier and a discrimination module, in order to be converted into a digital signal. In the following pages, it will be used an Artix 7 FPGA board for the analysis of the dark count rate (DCR).**

I. Introduction

THE silicon MultiPixel Photo Multiplier (MPPC) consists of an array of np junctions, also known as *Avalanche PhotoDiodes (APDs)*. When a photon hits a single junction (or pixel), a pair electron-hole is created and accelerated by the electric field provided by the inverse polarization of the APD. If the field is high enough (in particular, if the reverse voltage is higher than the breakdown voltage), electrons and holes will be accelerated and produce further pairs electron-hole. For that reason his process is called "avalanche effect" and causes a current that can be measured. As reported by the datasheet[2], the gain of an MPPC working under this condition (called *"Geiger mode"*) is around 10^6.

An MPPC can be schematized as follows (figure 1a): the np junction behaves like an inversely polarized diode. The

Since the avalanche process generates a current, the signal is simulated by a current generator, with his parasitic capacitance in parallel.

The generator signal is a single pulse whose fall follows the exponential law of the RC circuit. τ is given by the product of the equivalent capacitance and resistance of the pixels. The intensity of the current is:

$$I_1 = \frac{Q_{in}}{\tau}. \tag{1}$$

Q_{in}, in turn, is given by the charge of the single electron multiplied by the MPPC gain M. τ, as it can be seen on the oscilloscope, is around $100ns$.

V is the High Voltage provided to the SiPM. C_{filter} and R_{filter} build up a simple low-pass filter, as suggested by the typical application circuit drawn in the datasheet[2].

In series with the generator is represented the quenching resistor R_Q, along with the quenching capacitance C_Q. C_{dn} represents the capacitance of all the pixels that aren't active (and so it's simply calculated as the parallel of $N-1$ condensers with the value of C_d). Similar calculation is made for C_{qn} and R_{qn}. Finally, C_g represents the grid capacitance.

Final project

Feel free to implement whatever you want ! ...

... or simply google for "FPGA project", "Verilog project" or "VHDL project" etc.

In case you really run out of ideas :

- a sine-wave generator
- a FIFO in Verilog (we will use a soft IPs in lectures instead...)
- a master/slave system connected through UART protocol
- a master/slave system connected through SPI protocol
- a master/slave system connected through JTAG protocol as from IEEE Std. 1149.1-2001
- a clock-domain-crossing (CDC) FIFO
- etc.

Online resources /1

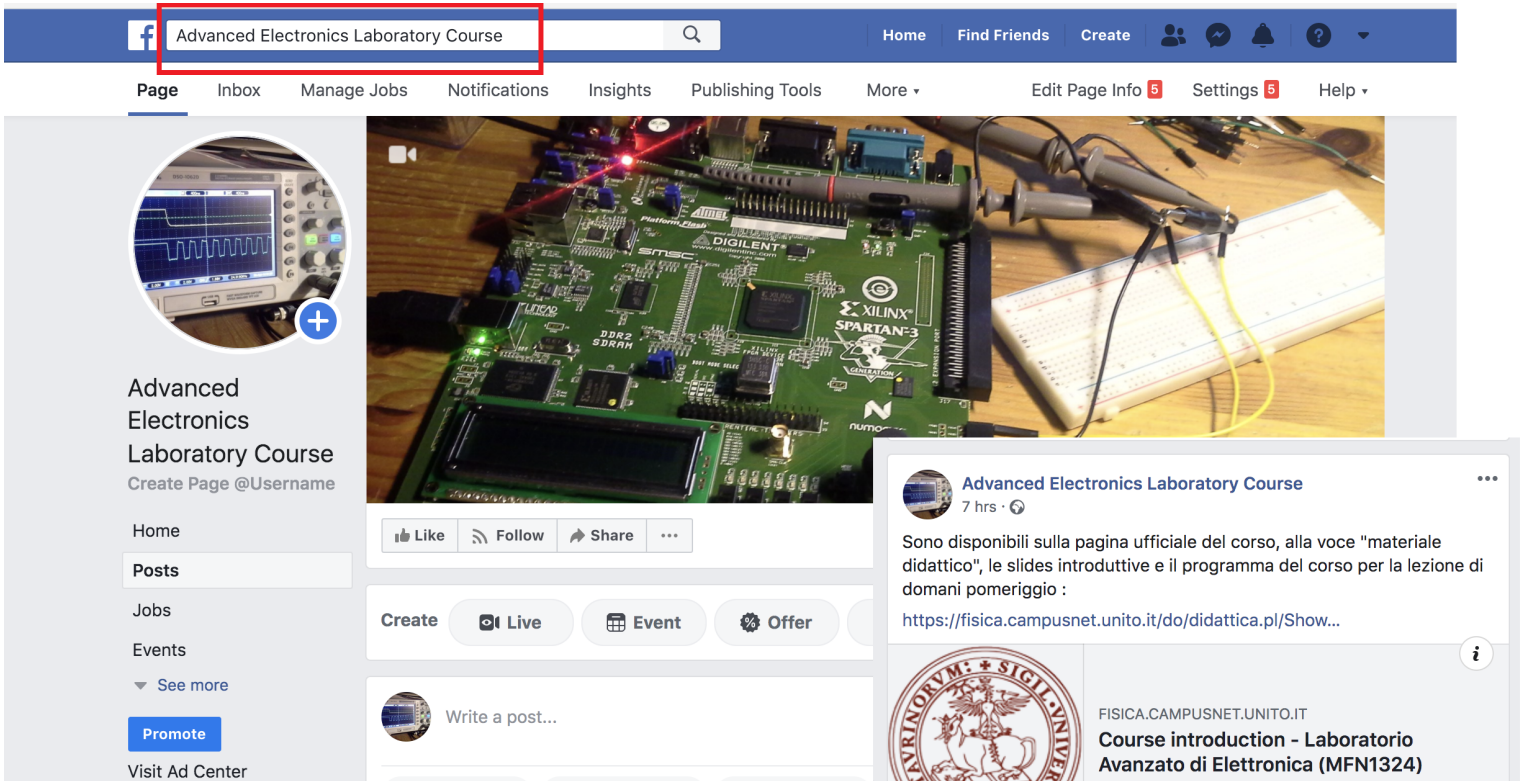
main course page on CampusNet :

- https://fisica.campusnet.unito.it/do/corsi.pl/Show?_id=70d4
- slides and additional course material (documentation, datasheets etc.)
- this is intended to be **THE MAIN** place to keep track of lectures and material !

Online resources /2

informal FaceBook page :

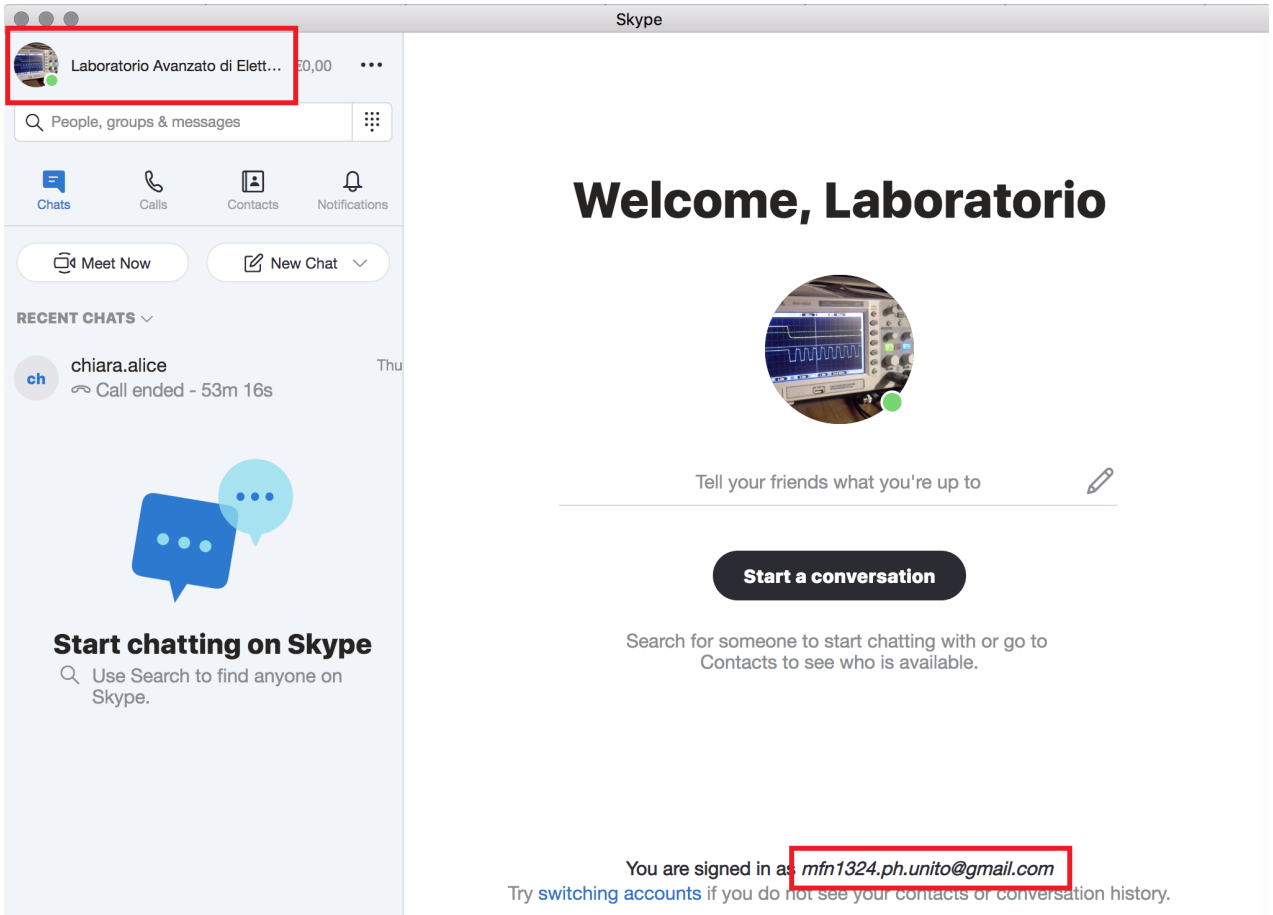
- basically useless, just created as an additional opportunity to remain connected given the current situation
- **not intended** to be a primary source of information !
- that is... don't create a FB account if you don't have it just to access this page
- feel free to post issues or ask questions in the page... don't be shy !



Online resources /3

informal Skype "lab account" :

- as before, just created as an additional opportunity to remain connected
- search for "Laboratorio Avanzato di Elettronica" or *mfn1324.ph.unito@gmail.com*
- mainly intended to **help me solving your problems** (e.g. tools installation)

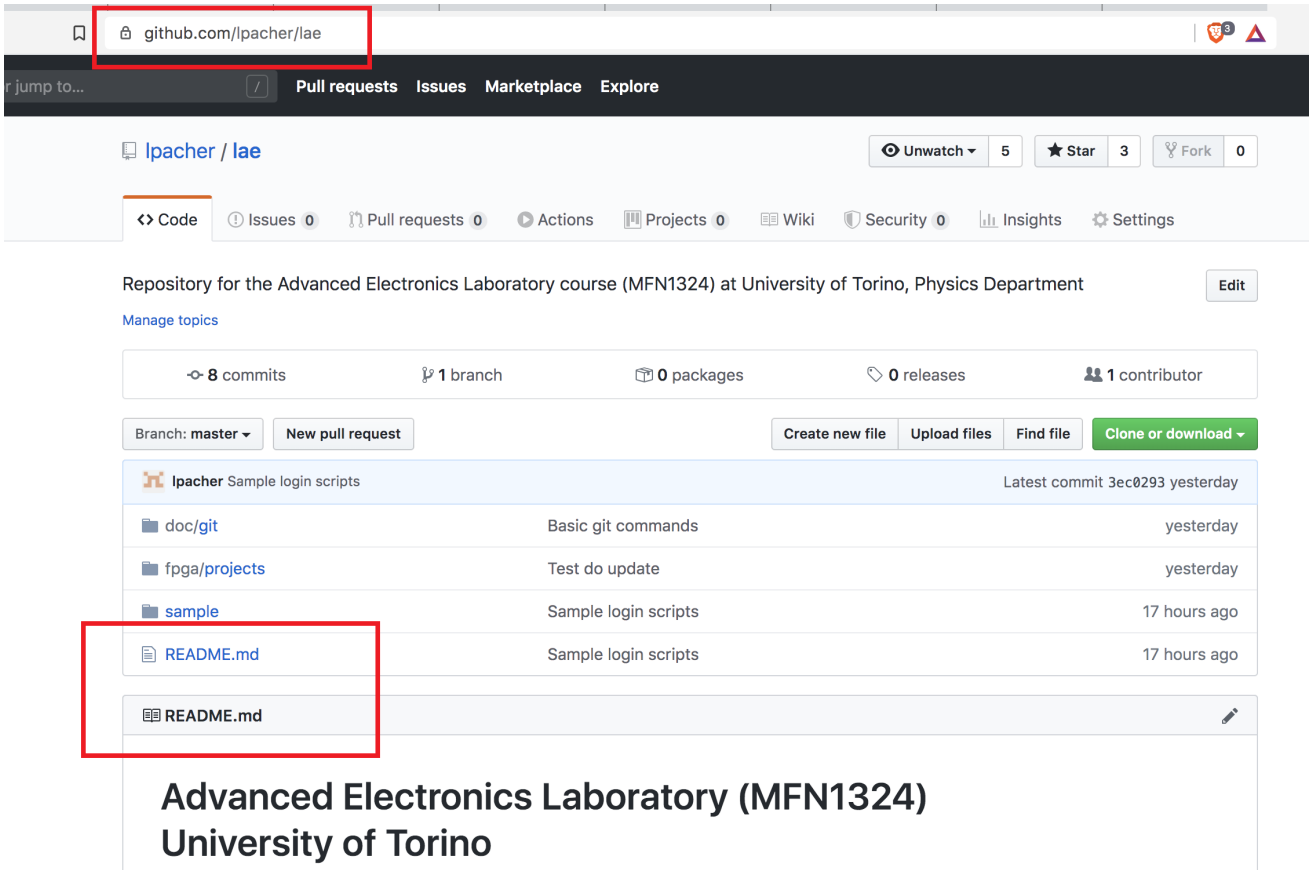


Online resources /4

GitHub repository :

- https://github.com/lpacher/lae
- setup scripts, RTL sources, Tcl scripts, Makefiles, XDCs (more details later on ...)

Register and create a GitHub account !



GitHub repository

In this course we will **write a lot of code** !

- Verilog RTL and testbech sources
- simulation and implementation scripts in Tcl, GNU Makefiles
- timing and physical constraints (XDC)
- XML configuration files for IPs (XCI) etc.

Final "working solutions" for all examples proposed during lectures will be uploaded to our GitHub repository after each lecture. To **get all updates** just use `git` from the command line :

```
cd /path/to/lae
git pull origin master
```

Detailed information about Git installation and configuration can be found in the main `README` file https://github.com/lpacher/lae/blob/master/README.md

I would also recommend to use Git for your final project !

Video-lectures

- course delivered in form of video-lectures using **Webex UniTO services**
- 2 hours per lecture, h 14-16
- links to Webex sessions in the main course page on CampusNet
- I will also send a reminder with the link in the morning to the mail list of students registered to the course
- I will also post the link in the FB page
- I will also put the link in the main README file on the GitHub repository

All lectures will be recorded !

Course program

- Verilog HDL fundamentals, HDL design flow
- Logic values, resolved vs. unresolved logic values, 3-state logic, buses and endianess
- review of boolean algebra
- introduction to Xilinx Vivado simulation and implementation flows
- design and simulation of combinational circuits with Verilog examples (multiplexers, decoders, encoders etc.)
- FPGA architectures overview and basic building blocks (fabric, BEL, LUT, CLB, CLA, slices, IOBs, hard-macros)
- introduction to Xilinx Design Constraints (XDCs)
- sequential circuits, latches and FlipFlops
- counters, registers, PWM, shift-registers, FSM, FIFOs, RAM/ROM
- advanced Xilinx Design Constraints (XDCs): timing fundamentals
- synchronous design good and bad practices, example Vivado IP flows (clock wizard, FIFO compiler)
- gate-level simulations with back-annotaed delays (SDF)
- advanced topics in analog circuits simulation using LTspice (e.g. OPAMPs stability)

Let's start !