Data Acquisition

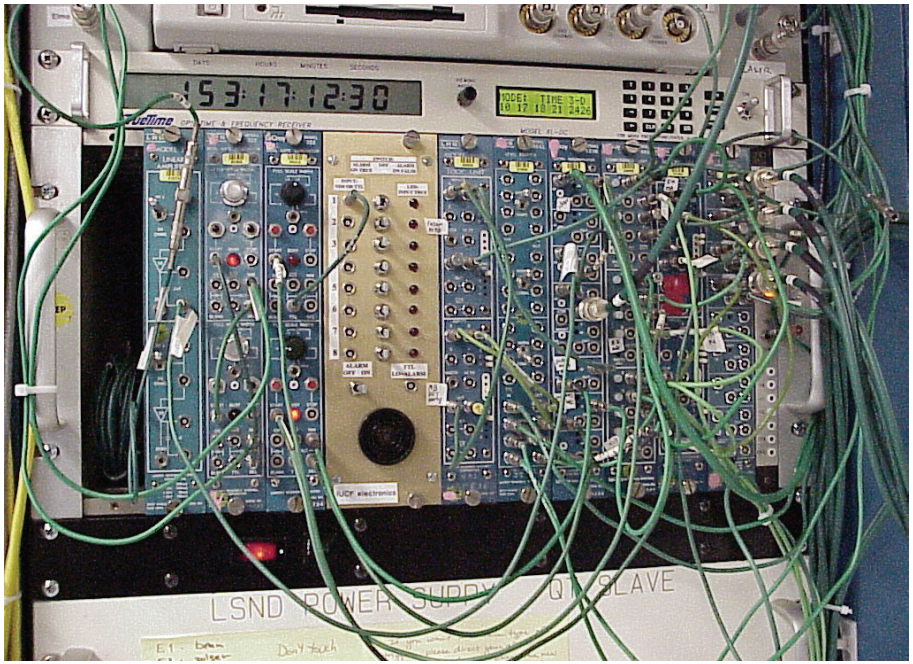
PSI Practical Course 2014

Niklaus Berger

Physics Institute, University of Heidelberg

Overview

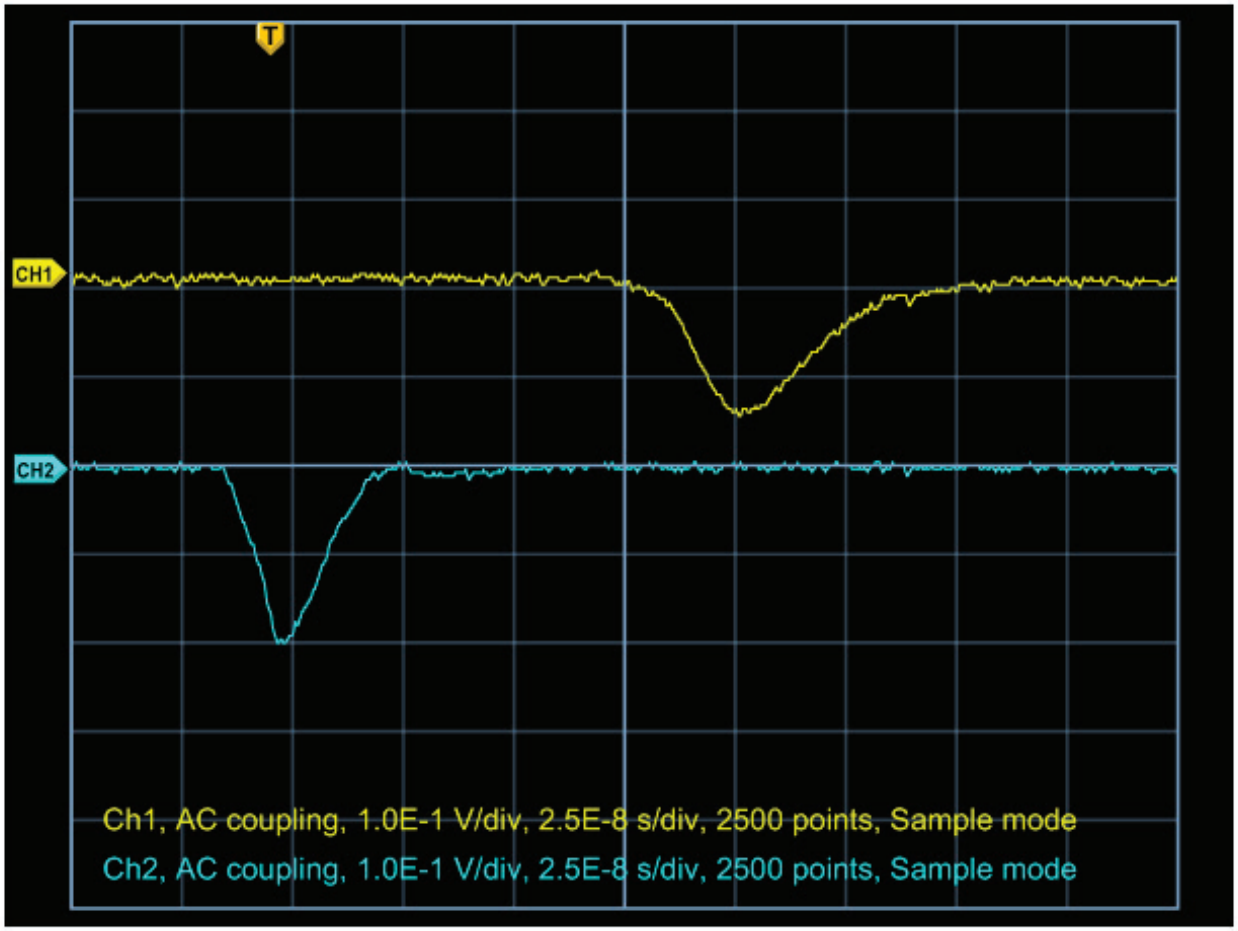
How to get data into a computer?



Digitization

From Signal to Number

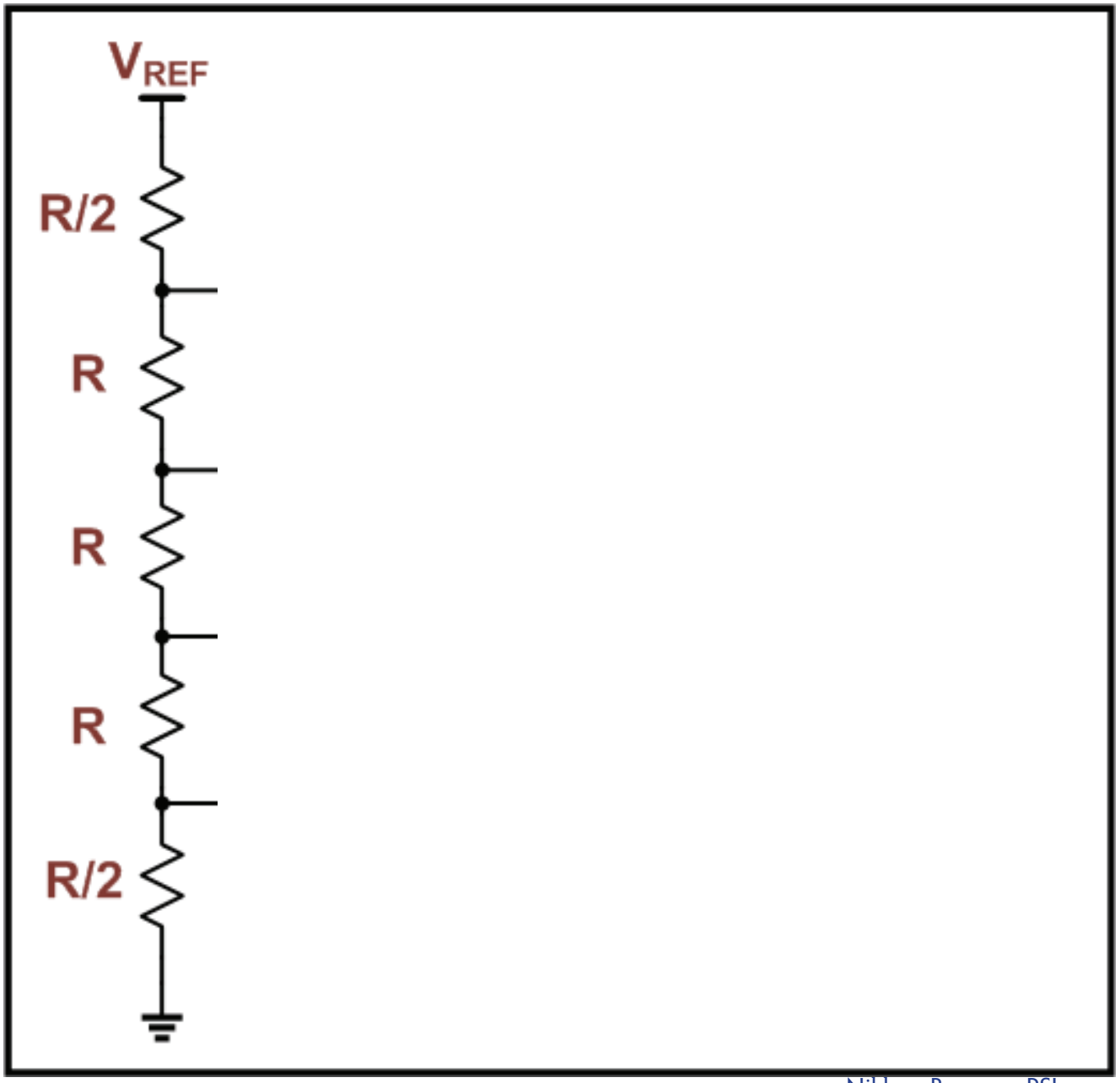
Detectors produce electrical pulses



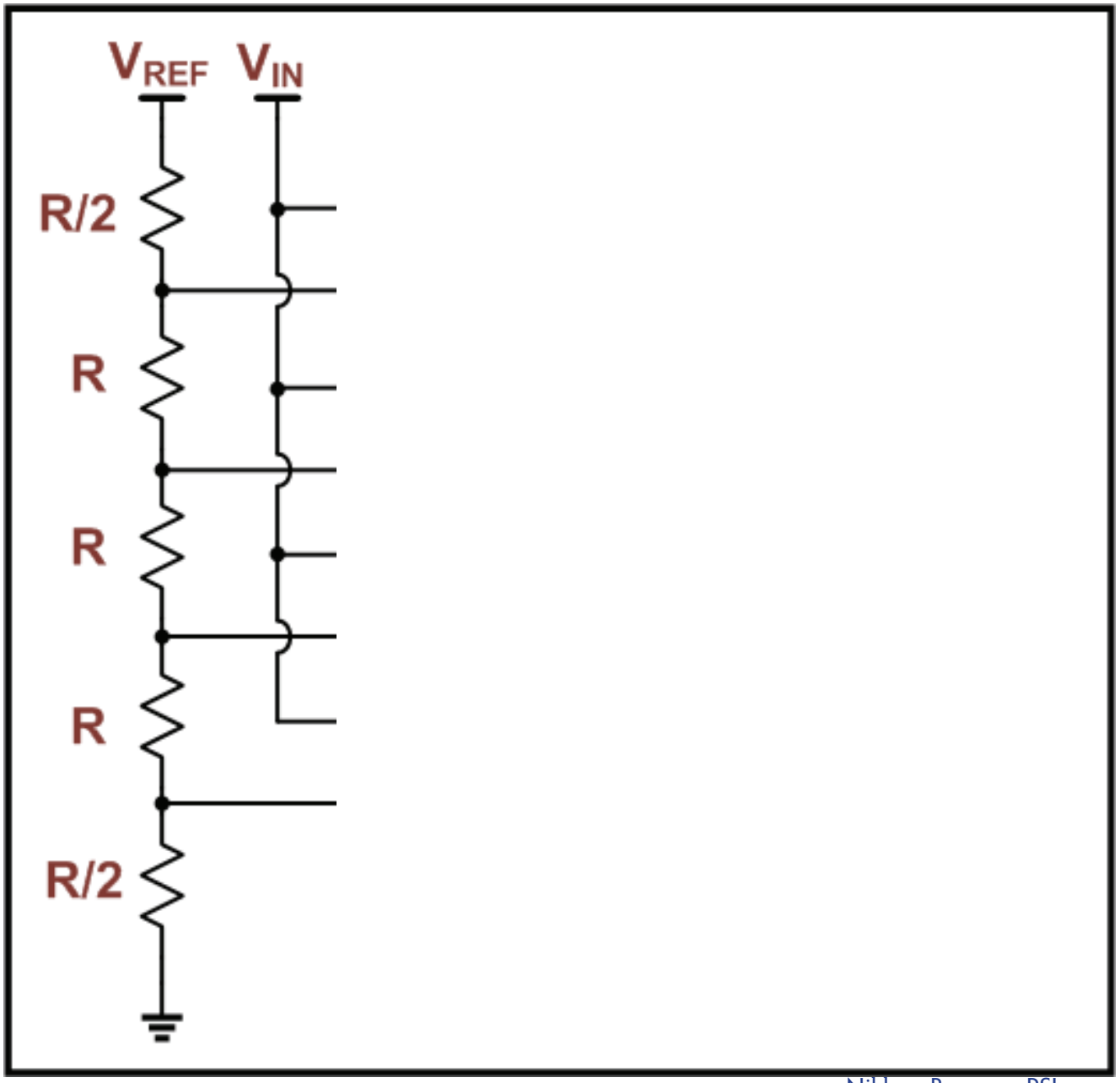
How big was the pulse?

When was the pulse?

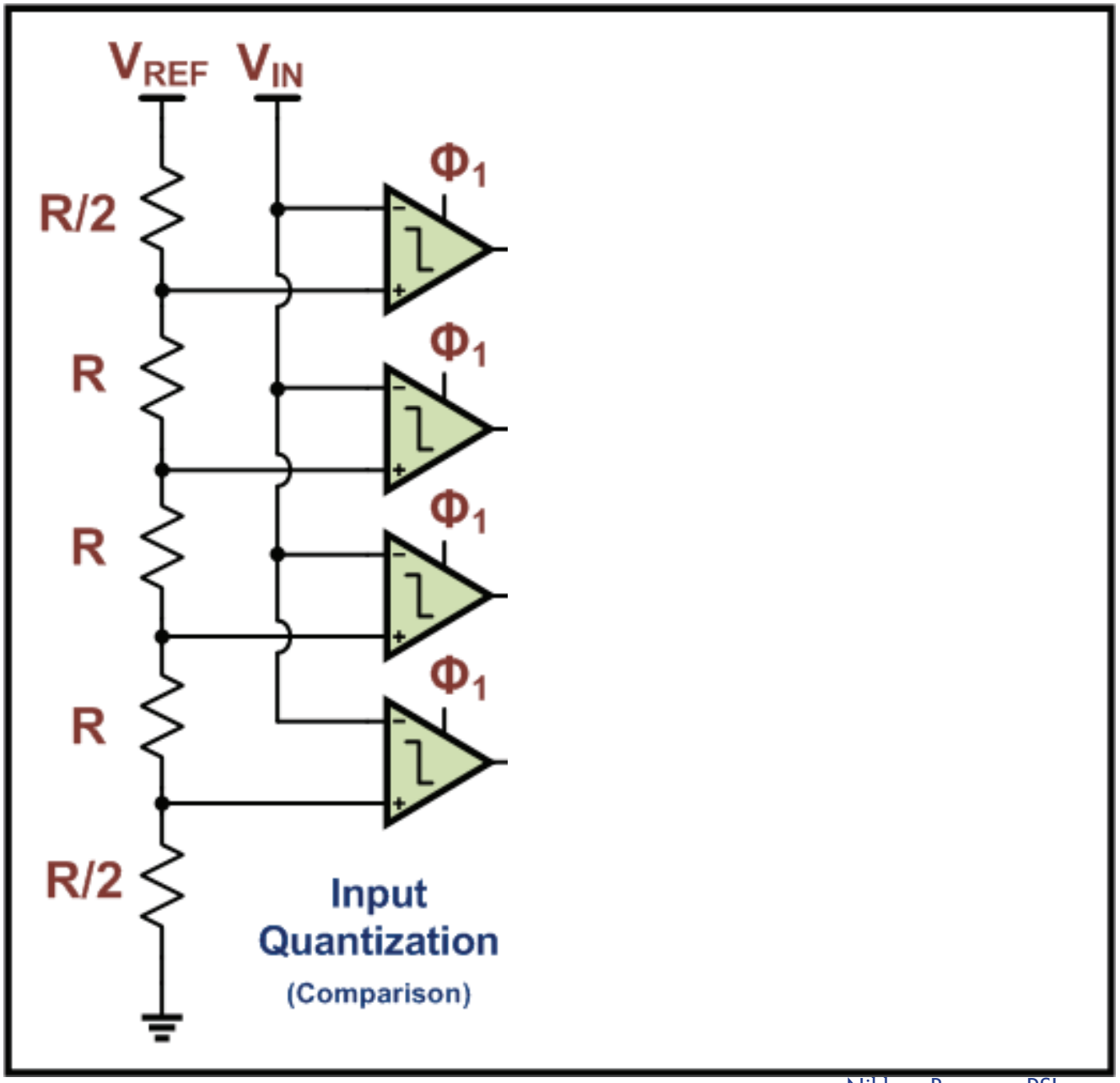
Analogue/Amplitude-Digital-Converter (ADC)



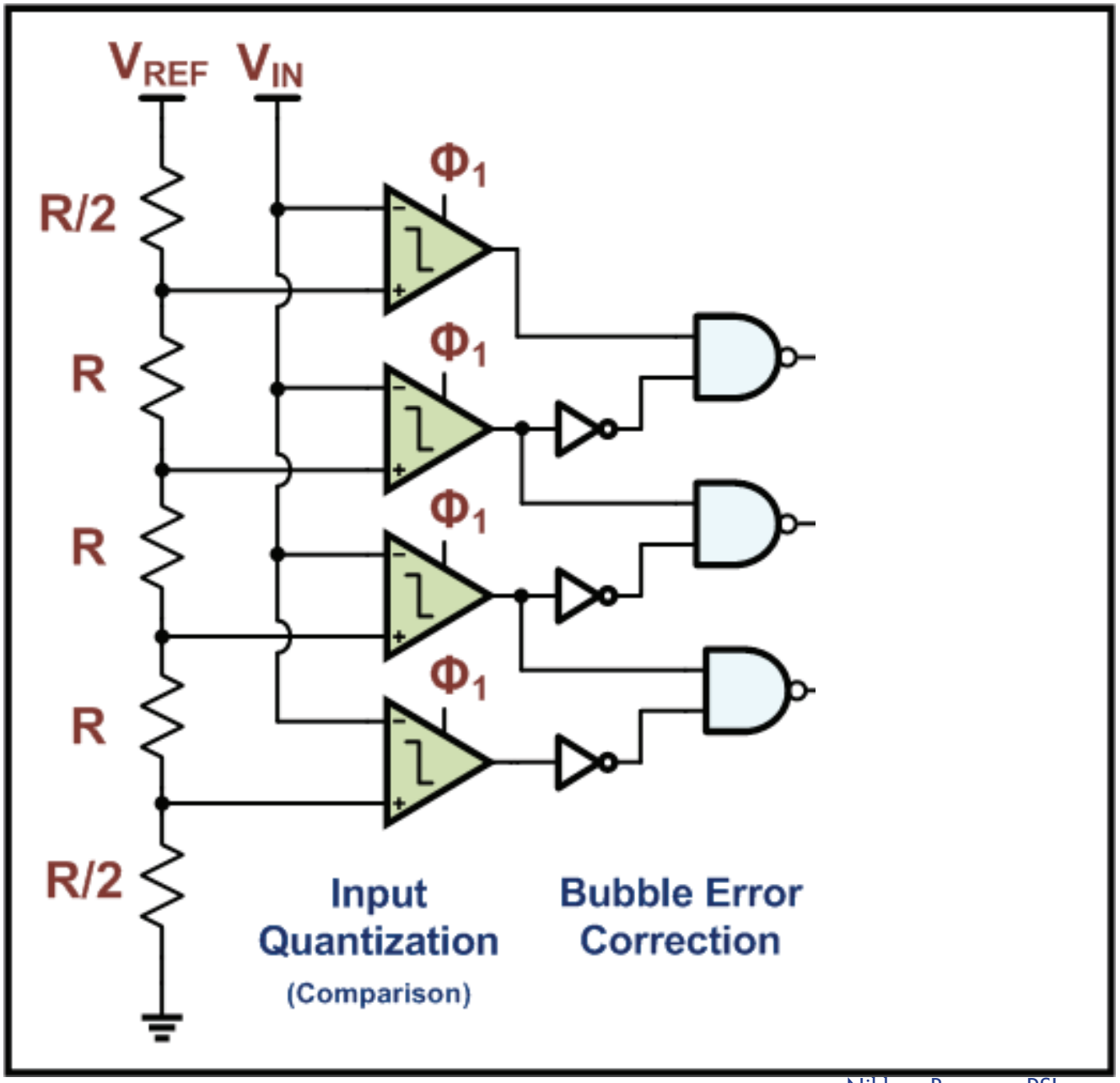
Analogue/Amplitude-Digital-Converter (ADC)



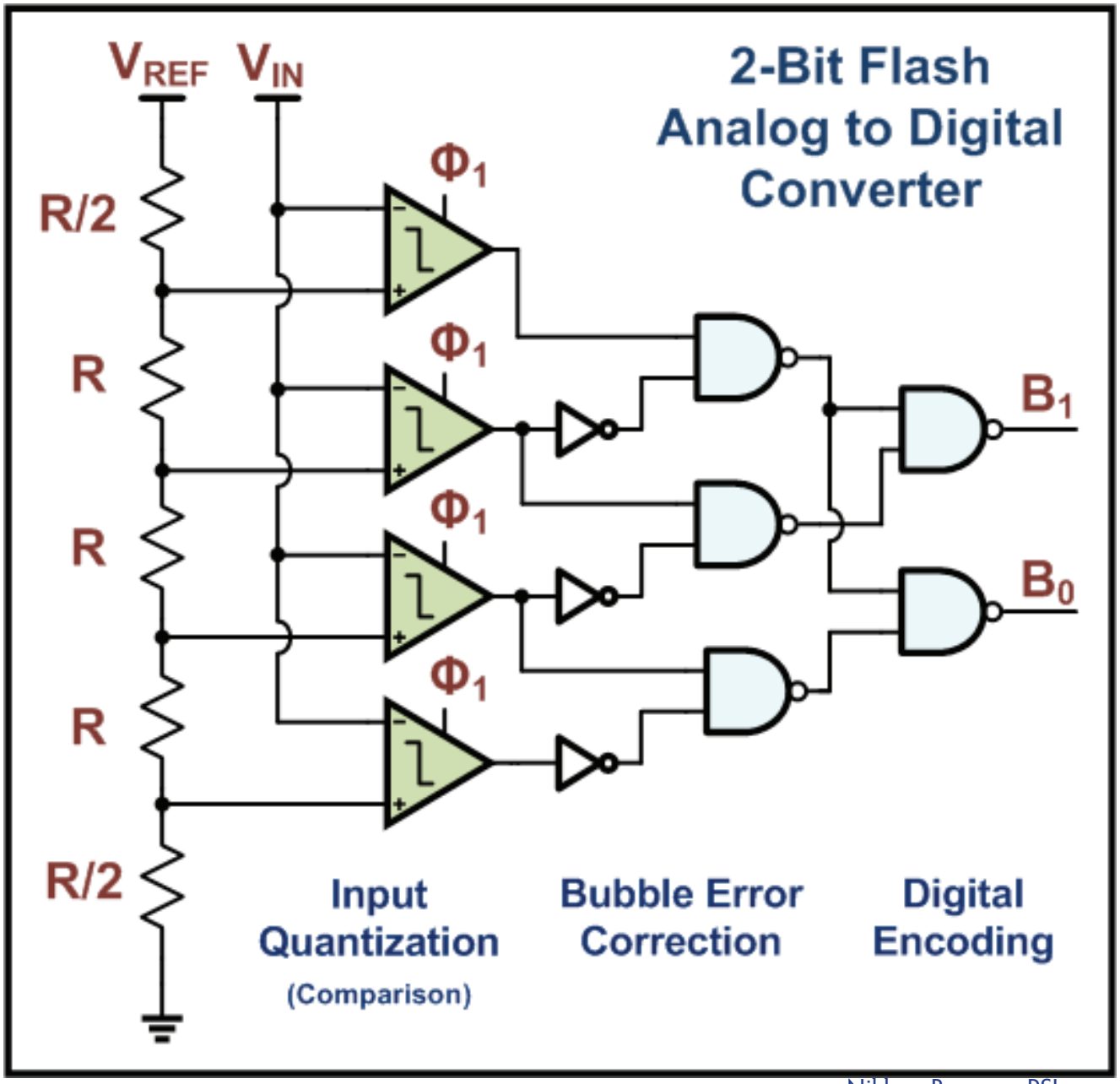
Analogue/Amplitude-Digital-Converter (ADC)



Analogue/Amplitude-Digital-Converter (ADC)

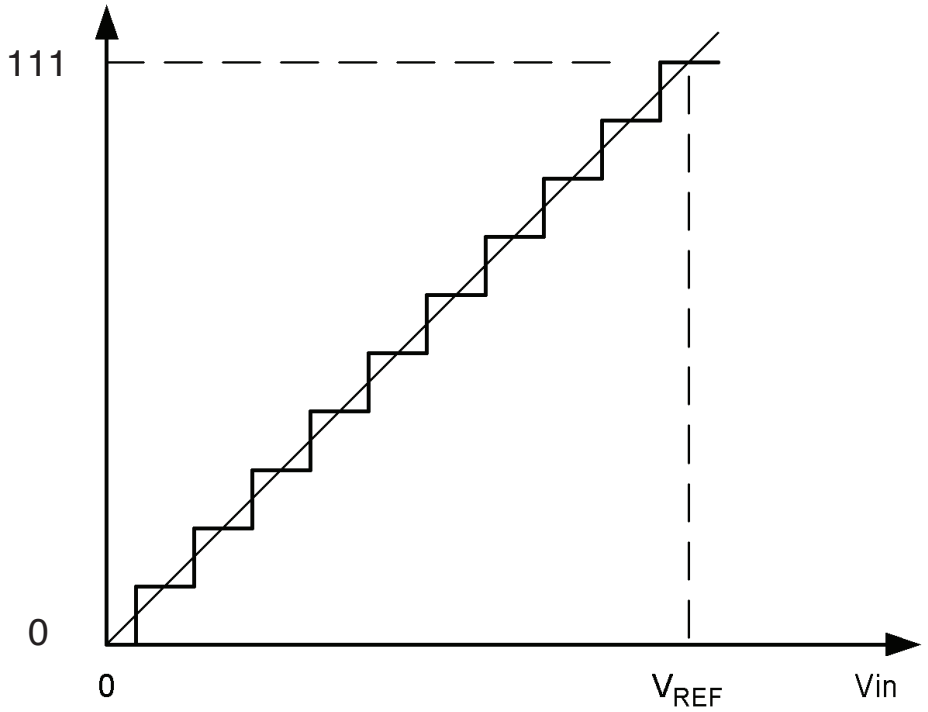


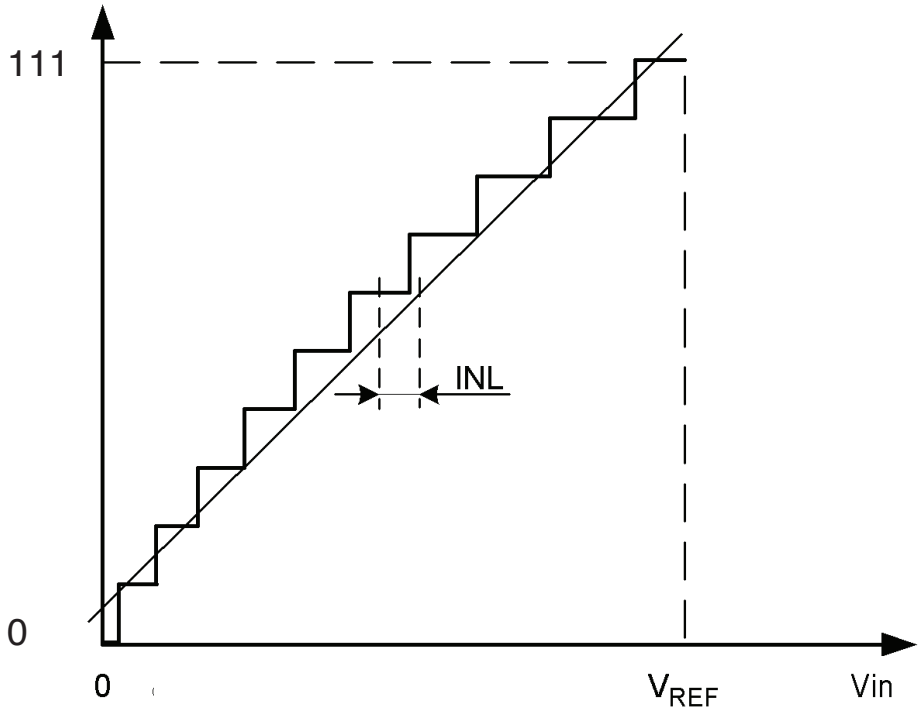
Analogue/Amplitude-Digital-Converter (ADC)



ADC-Characteristics

- Dynamic Range
- Resolution
- Linearity





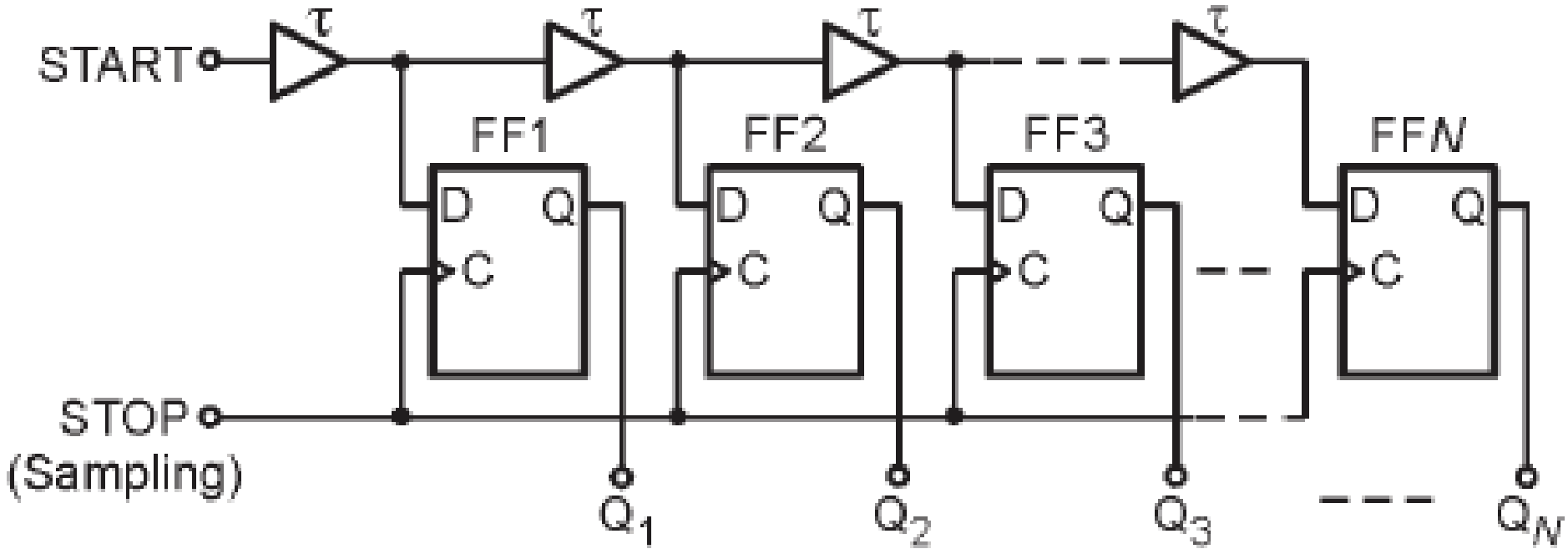
ADC-Characteristics

- Dynamic Range (few V)
- Resolution (12 bit)
- Linearity (0.5 LSB)

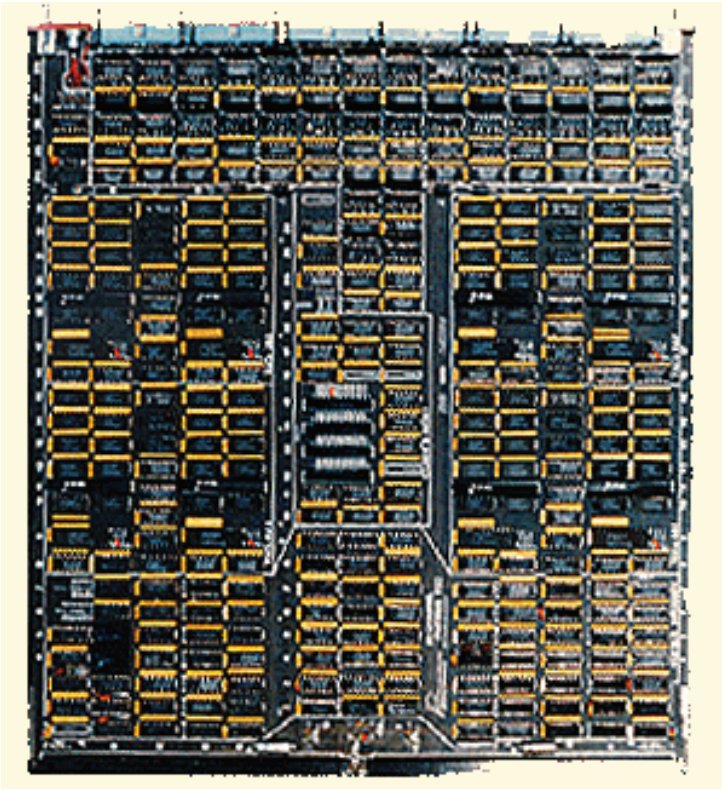
- Sampling rate (500 MS/s)

- Power consumption (3 W)
- Price (300 EUR)

Time-Digital-Converter (TDC)



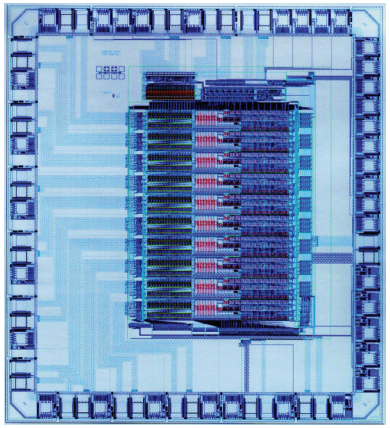
TDC Characteristics



- Resolution (20 ps)
- Dynamic Range
- Linearity

- Dead time

- Power consumption
- Price



-> Digital Data

but way too many...

12 bit ADC at 500 MS/s: 750 Mbyte/s per channel

Data rates

8 bit ADC running at 100 MHz

100 MByte/s per channel

Fills a 1 TB disk in about 3 hours

But: A USB 2.0 link only transmits 35 MByte/s

and most of the data will not be interesting....

Trigger

We need to find out on-line when something interesting happened

generate a trigger signal

only then we store the data

But: generating the trigger signal takes time - data will be gone...

Buffering

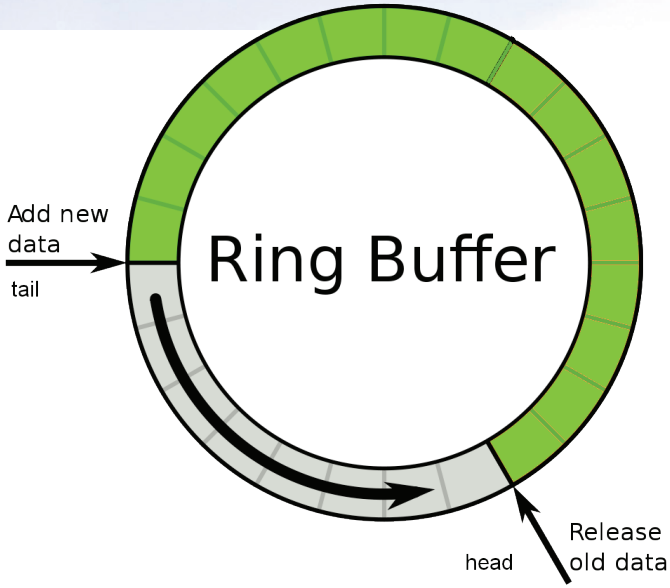
- Make sure the interesting data are still around when the trigger arrives

- Two approaches:

  - "Analog delay" - send signal through long cable, generate trigger in the meantime

  - Digital buffer - store digitized values in memory, discard if no trigger

- In both cases: careful timing alignment needed - get the right data out



FPGAs

Field programmable gate arrays

- Programmable logic
- Can implement buffers
- Select triggered data
- Interface to the rest of the world via ...
- Programming in VHDL or Verilog

We have a LogicBox

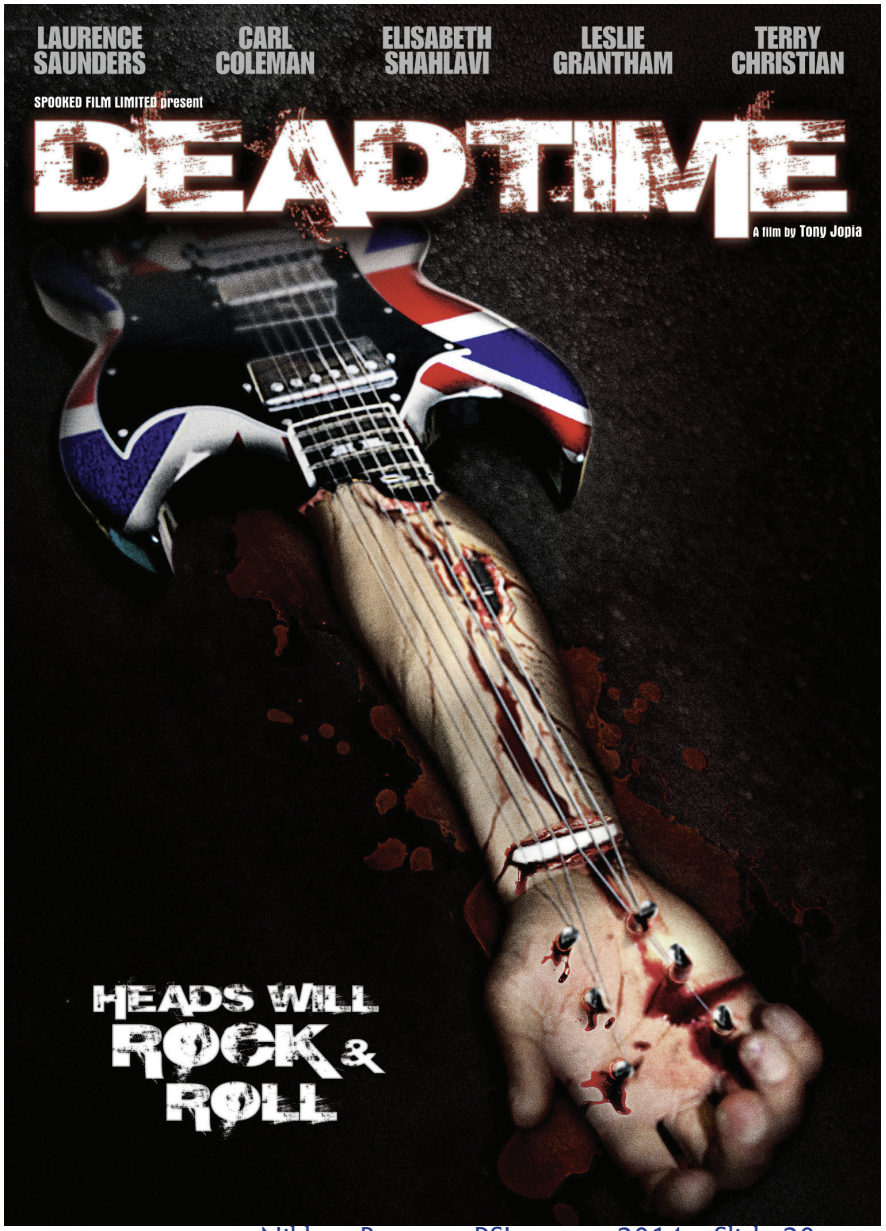
- ADCs
- FPGA
- USB link

Deadtime

The DAQ can cause several kinds of deadtime:

- No data taking during reading (e.g. ring buffer stopped)

- Buffers full on FPGA or on the PC

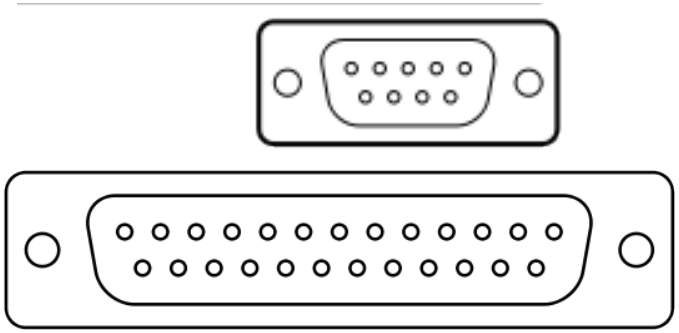


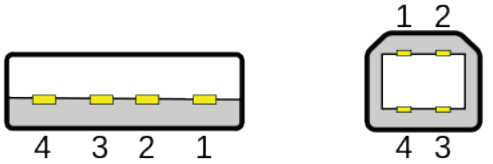
Getting stuff into the PC

Use some connection standard - look at the back of a PC for choices

- Serial port (RS 232) - up to 100 Kbit/s
- Parallel port - up to 2000 Kbit/s
- USB - 1.5/12/480/5,000 Mbit/s
- Ethernet - up to 10 Gbit/s (copper)
  - up to 100 Gbit/s (optical)

Also take into account protocols, ease of use etc...

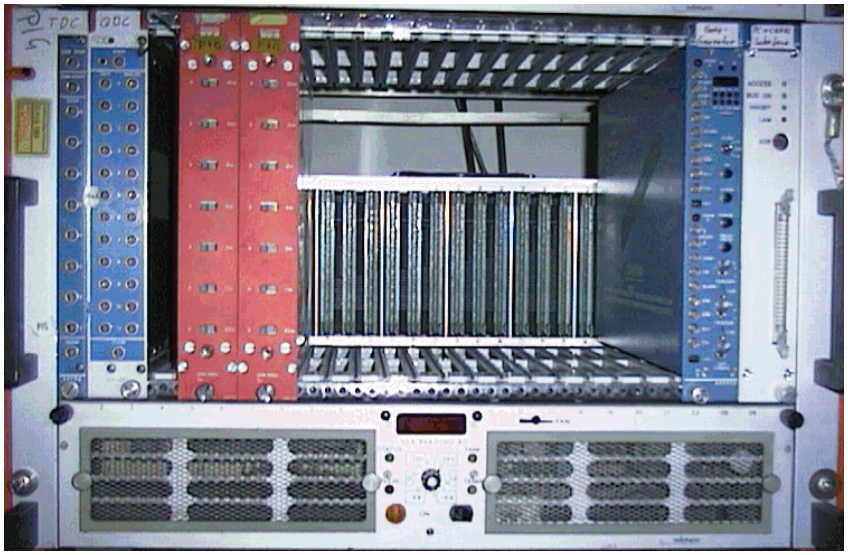




Aside: Crate based systems

Bring the PC to the electronics

- CAMAC (Computer Automated Measurement And Control)
  - around 1 Mbyte/s



- VME (VESA Module Europe)
  - up to 320 MByte/s

Push vs. pull

Two ways to move data

- Electronics write to PC memory (push)
  - make sure there is space...

  Typical for ethernet based systems

- PC reads electronics memory (pull)
  - make sure we pull fast enough

  Typical for bus based systems

Events and Runs

An Event contains all data registered for one trigger

- Header data like event number, time, event size...
- Raw data like ADC traces
- Later in the analysis also reconstructed energies, momenta

A Run is a set of events with the same conditions

- Detector positions, HV, beam settings, DAQ settings etc.
- Run header plus event data

On the PC we need...

A program that

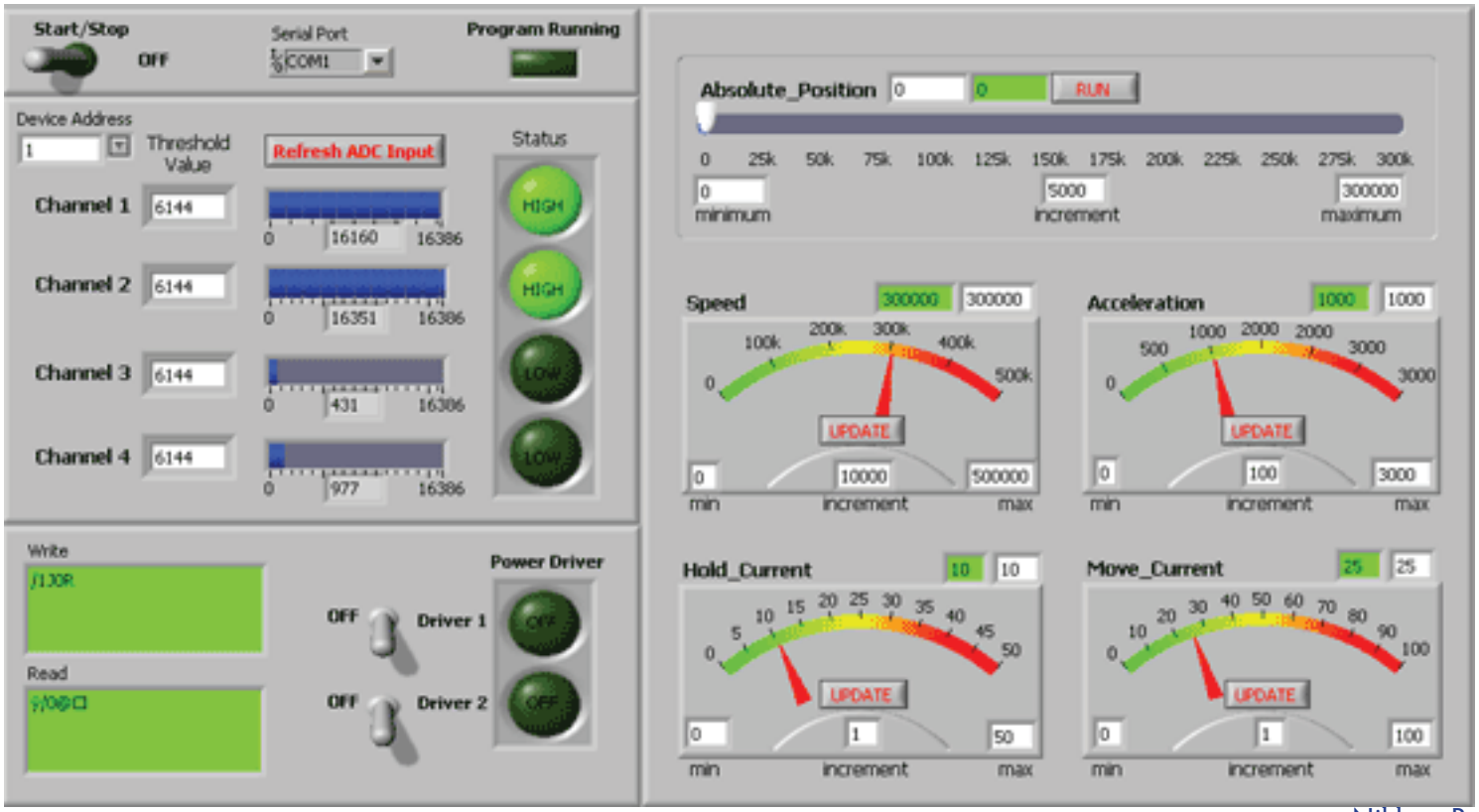
- Pulls data via USB
- Formats these data
- Saves them to a file
- Allows us to start and stop data taking
- Allows us to check that things work properly

Order is not arbitrary...

- Things have to run in parallel
- With the right priority
- A hard programming problem

LabView

- Data acquisition and display
- Easy multithreading
- Graphical programming
- Let's get hands-on



Tasks

Write the following sub VIs

- Generate a new run number that is the last run number plus one, even if the program was stopped in the meantime (Tip: Use a file)

- Take an ADC trace (an array of 16 bit unsigned integers of length N) and find the maximum, minimum, integral, pulse length above a threshold, pedestal, integral with pedestal subtraction...

- Take one to 14 ADC traces of length N plus some header data and write them to a file. Read back the file.

- Prepare a user interface for starting and stopping runs, choosing number of channels, number of samples and number of presamples. Disable settings changes if the run is ongoing