Efficient FIR Filter Implementation in FPGA

Mariusz Suchenek
AstroCent, Warsaw

Garching 12.02.2019

The aim of the work

The aim of the work is to reduce use of hardware resources of FPGA

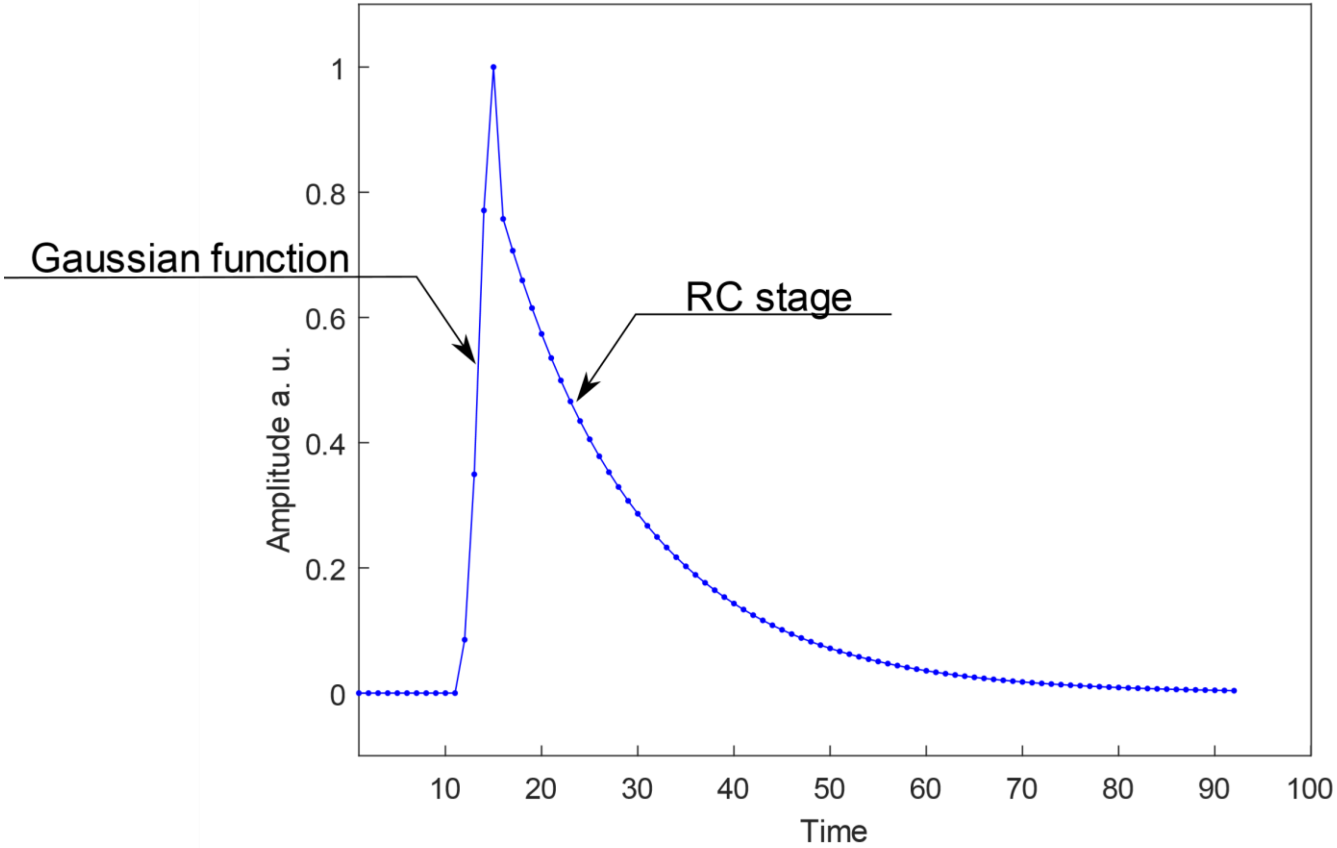
FPGA hardware resources (embedded in FPGA):

- Hardware multipliers (18x18), software multipliers
- Adding and subtracting logic elements
- RAM memory

Power consumption ?

Detector response

- Sampling frequency 100 Msample/s (10 ns sampling period)
- Rising edge, behaviour as a Gauss function, 4 samples
- Falling edge, behaviour like an RC stage, about 80 samples

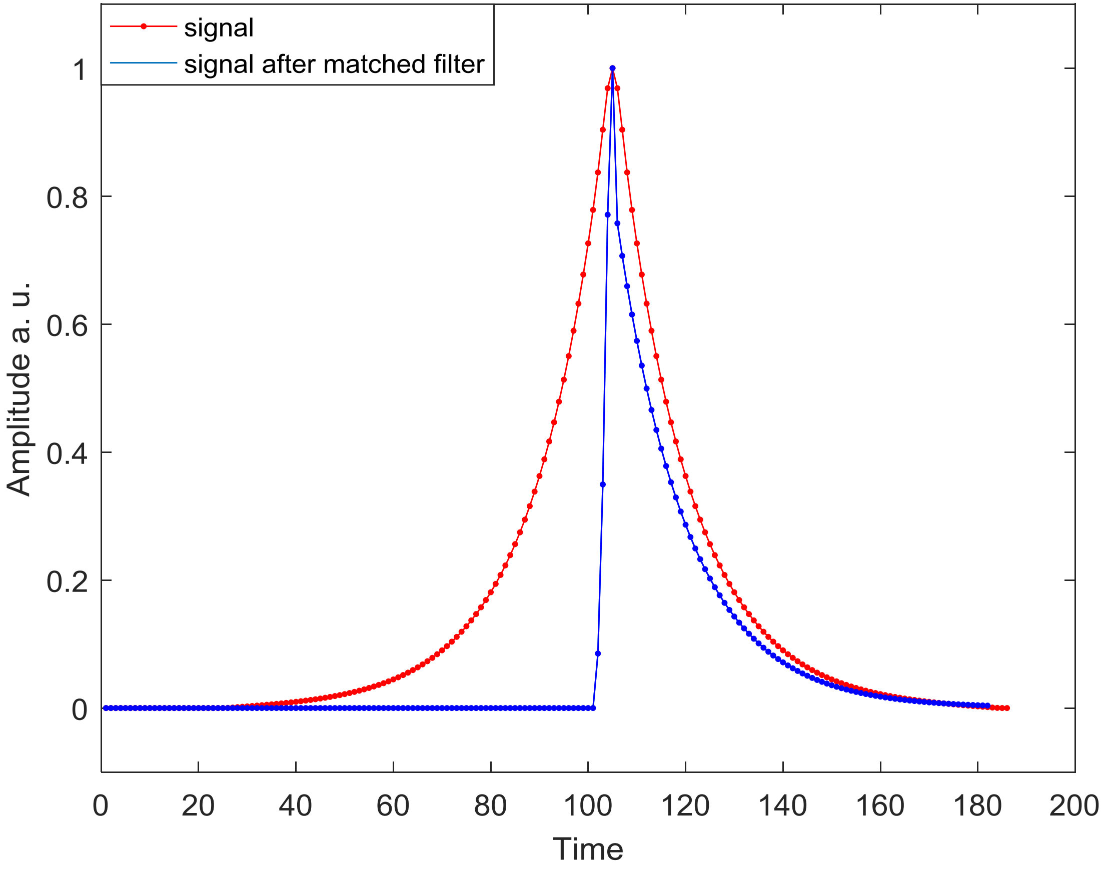


Matched filter

- Detector response about 800 ns (80 samples)
- 80 filter coefficients

Requirements for the filter:

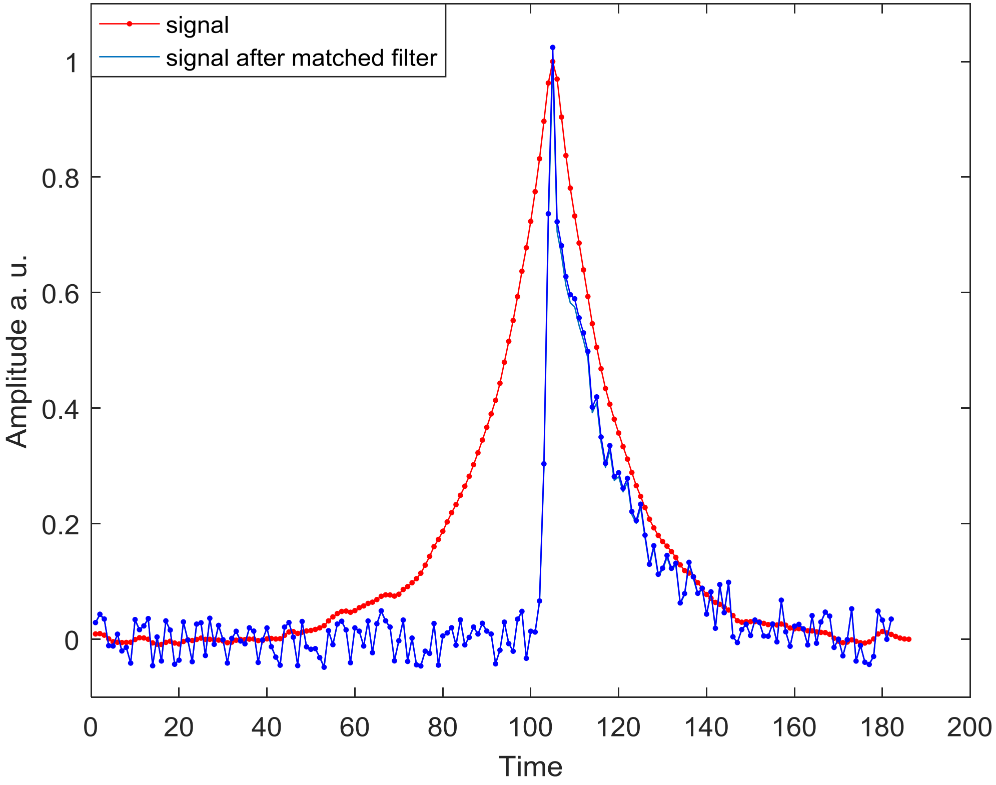
- 80 multiplications
- 80 additions / subtractions



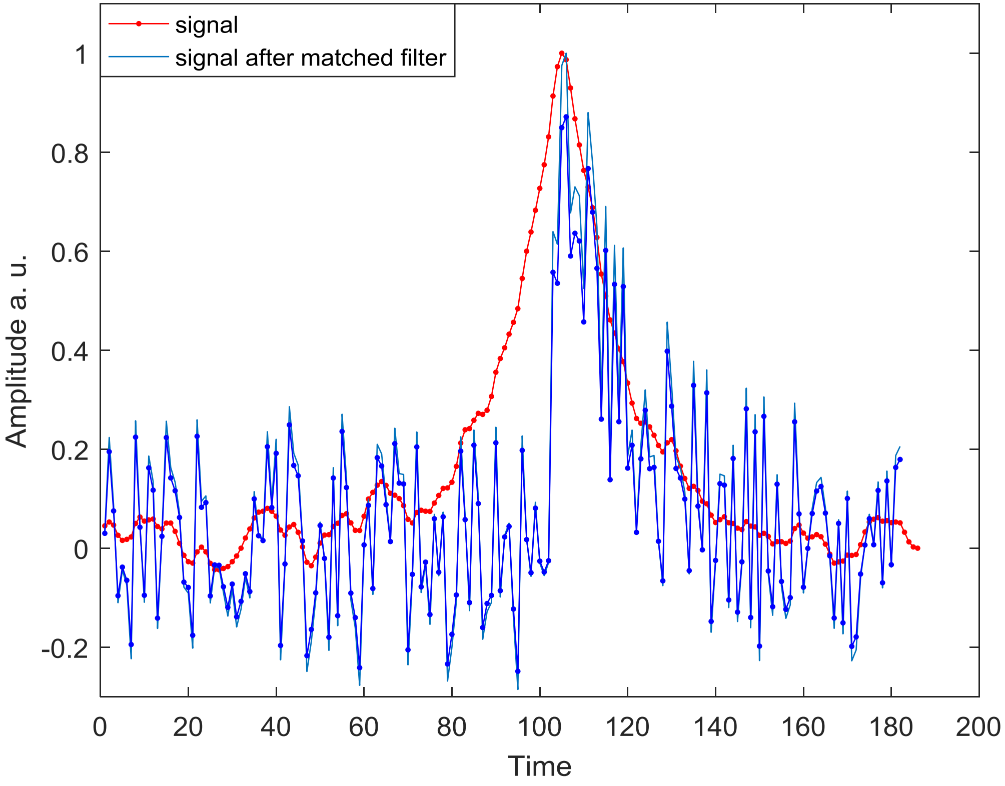
The signal from the detector with noise

Matched filter with a high and low signal to noise ratio

high

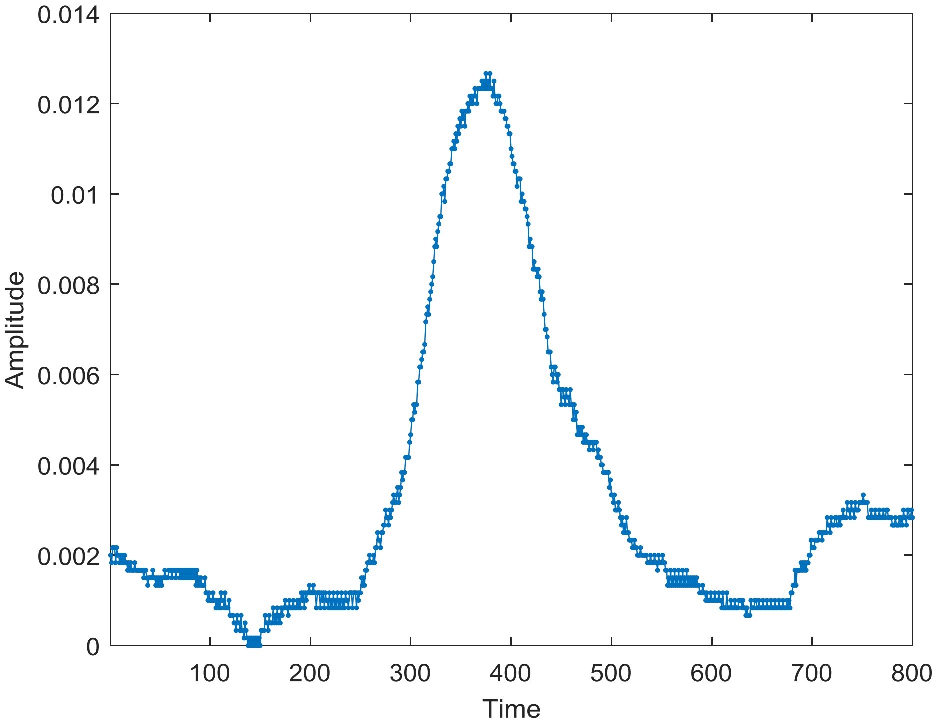


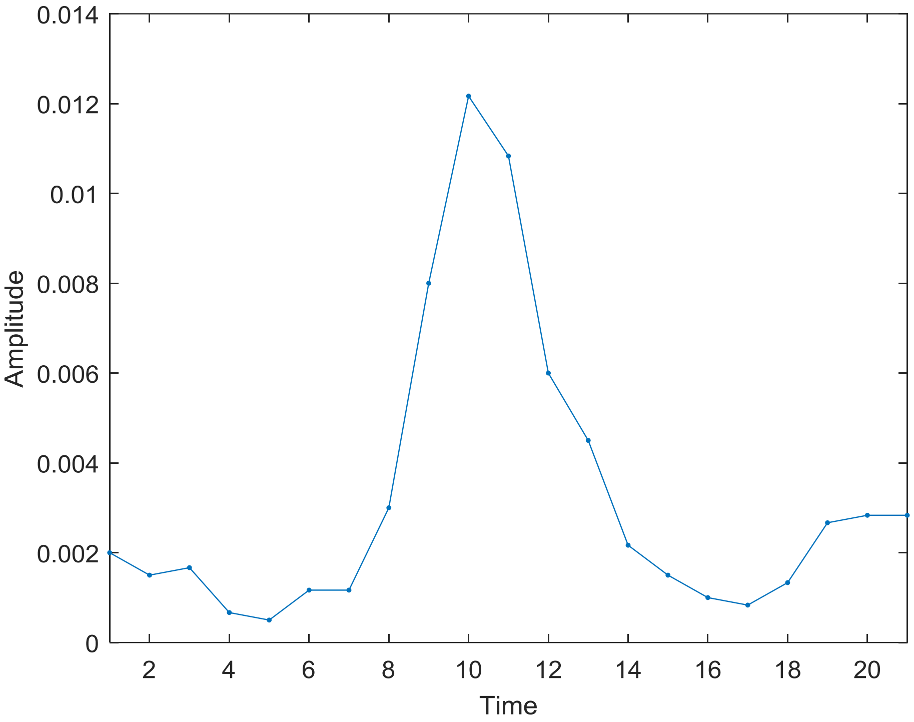
low



Signal from the detector

- 4 Gsamle/s 100 Msamle/s



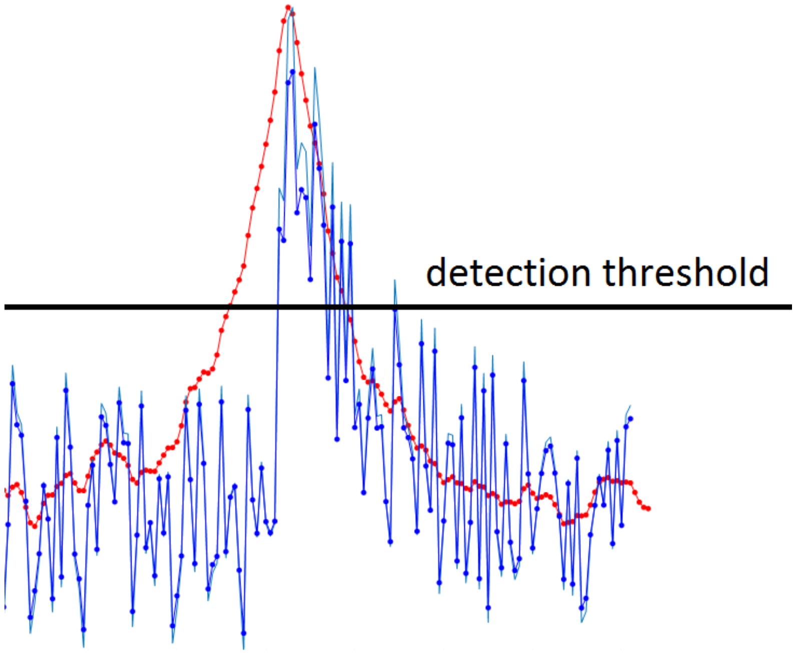


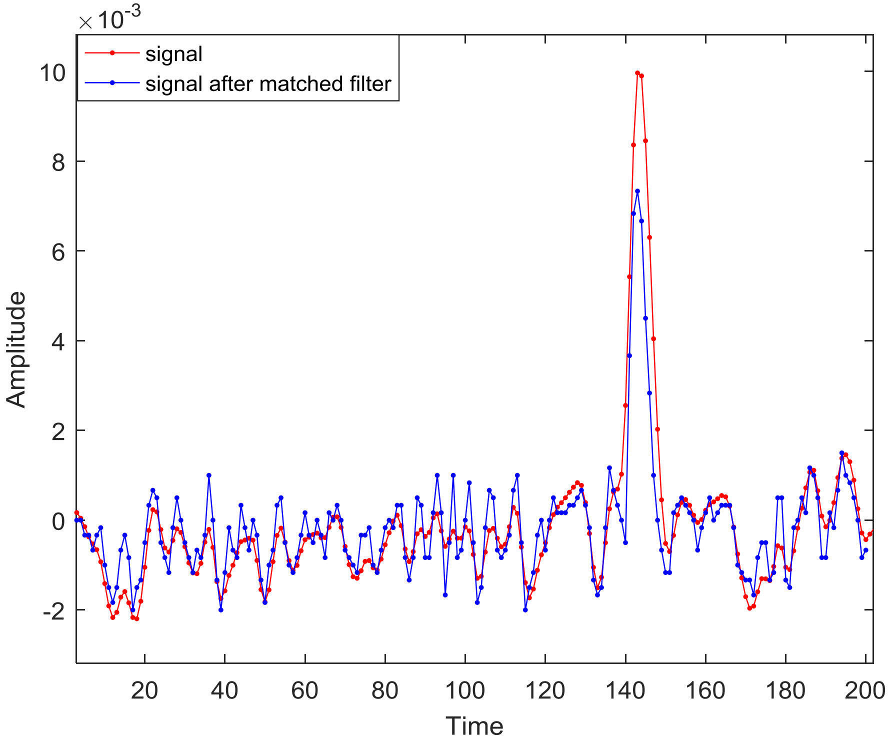
- 450 samples 11 samples

The number of filter coefficients: 450 (4 Gs/s) or 11 (100 Ms/s)

Matched filter

- Maximize SNR (signal level increases, noise decreases)
- Improve the shape of the pulse (the filter smooths the pulse) which improves signal parameters for detection. The noise on the pulse does not trigger the detection system once again
- Simulation   Signal from the detector (11 coefficients)





Considerations about the filter

The speed of embedded multipliers in FPGA Xilinx (hardware multipliers):

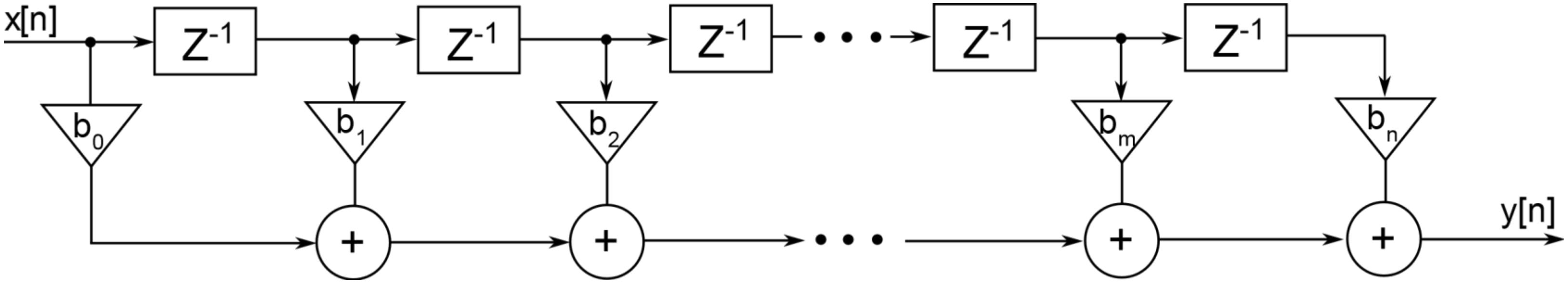
- Kintex UltraScale, 631 MHz
- Virtex UltraScale+, 800-866 MHz
- Kintex-7, 544 MHz
- Zynq UltraScale+, 820-860 MHz

Proposed filter:

- reduce filter length
- use the filter several times for the same data, filter coefficients divided into several sets

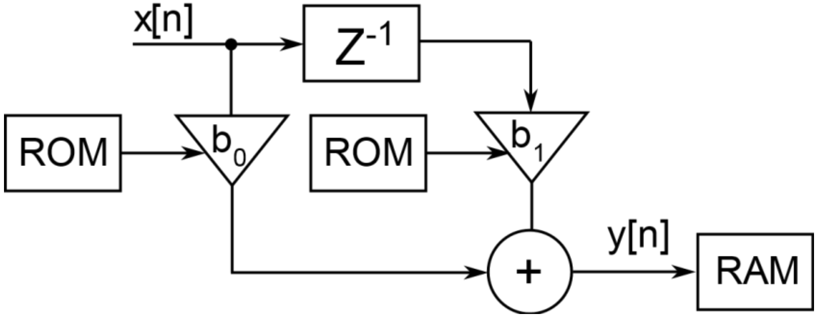
Considerations about the filter

- Direct implementation



- In one sampling cycle, we can make several multiplications, additions, subtractions - filter cycles. It is possible to shorten the filter around 630/100~ 6 or 800/100~8

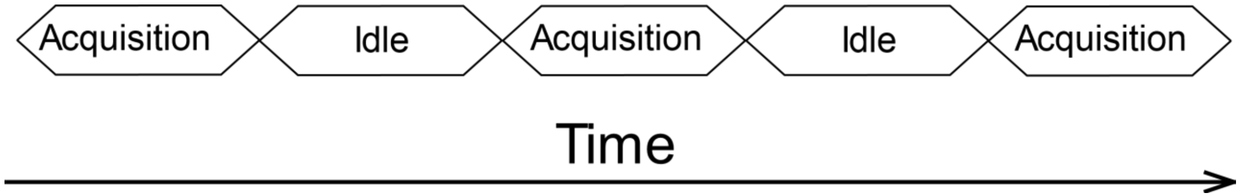
number of coefficients, reduce from 80 to: 80/6 ~ 13 or 80/8 = 10



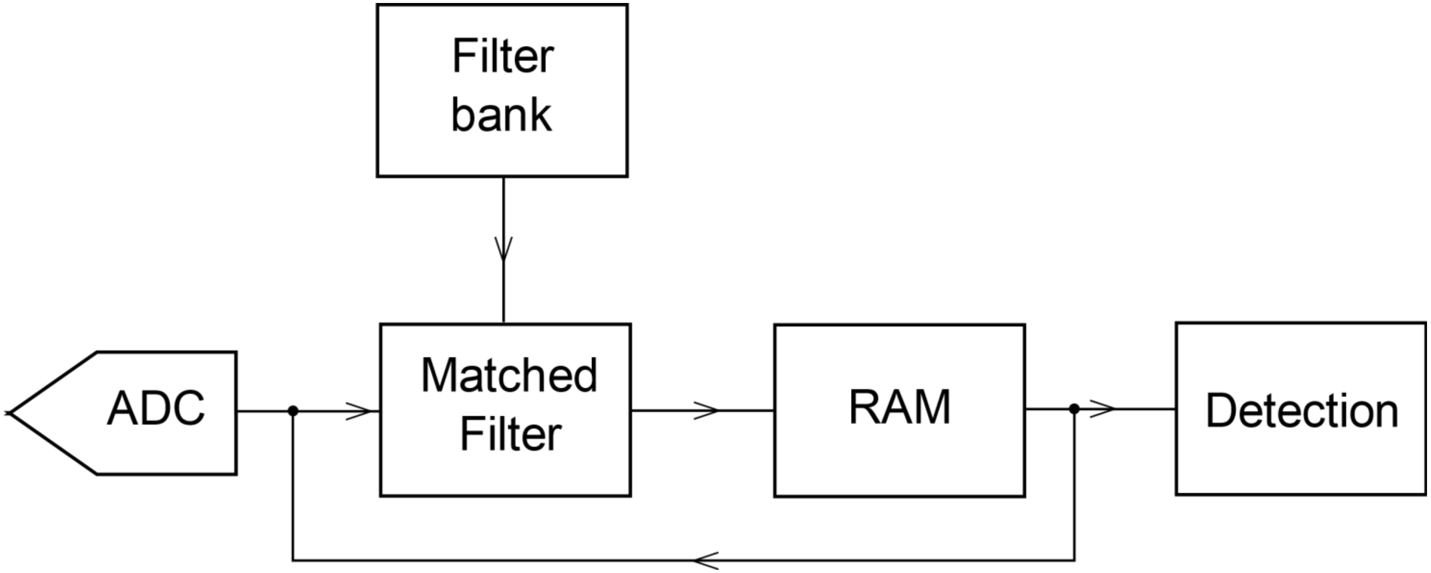
- Data acquisition in acquisition mode and fast processing in the idle mode

Filter

- Data acquisition in acquisition mode and fast processing in the idle mode

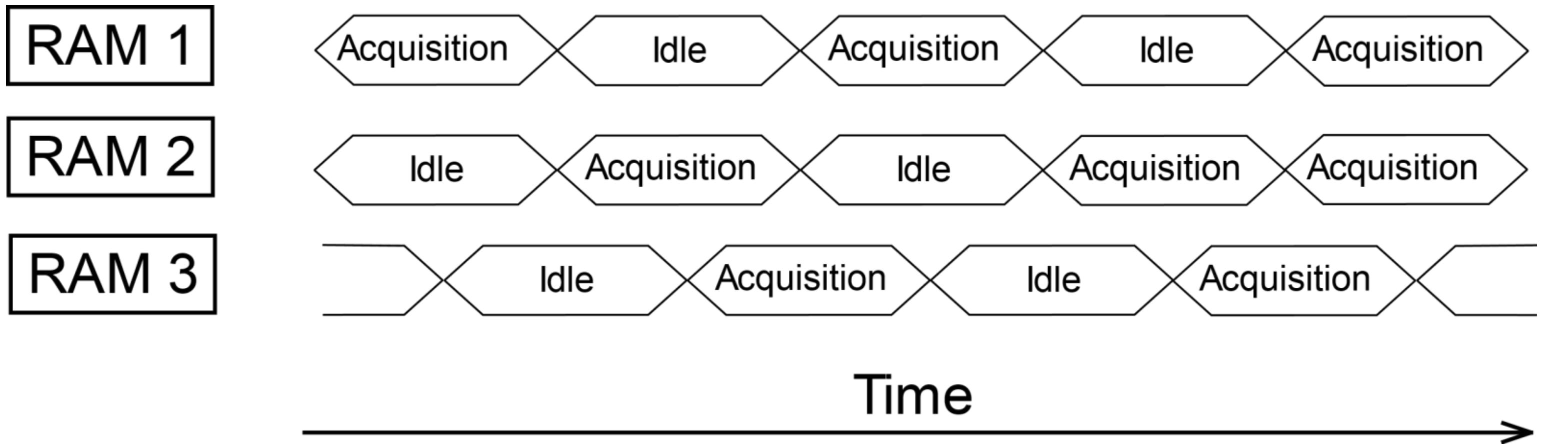


- System architecture with filter

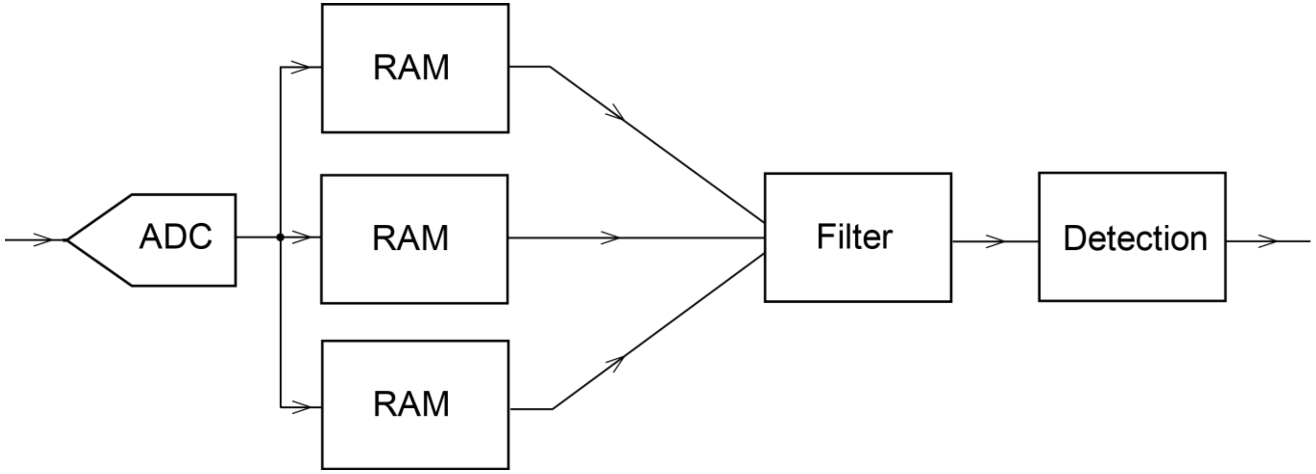


Filter

- Full time coverage, application of 1x filter + 3x RAM



- System architecture



Conclusions

- Sampling rate 100 Msamples/s, 80 filter coefficients
- Decrease filter order at least 6 to 8 times in exchang of increase filter speed
- Higher filter order (more coefficients, higher sampling rate), better matched filter

- Disadvantage: power consumption is proportional to the operating frequency

Thank you for your attention