Backend Design of the Read-Out System for Micromachined Sensors

HIRSAP Workshop 2019 - Karlsruhe, Germany

PhD Student:
Luciano Ferreyro

Scientific Supervisors:
Dr. Oliver Sander
Dr. Eng. Agustín Lucero

UNSAM Director:
Dr. Eng. Manuel Platino

KIT Director:
Dr. Marc Weber

ITeDA - Instituto de Tecnologías en Detección y Astropartículas, Buenos Aires, Argentina
UNSAM - Universidad Nacional de San Martín, Buenos Aires, Argentina
KIT - Karlsruher Institut für Technologie
IPE - Institut für Prozessdatenverarbeitung und Elektronik, Karlsruhe, Germany

Schedule

Introduction

Introduction

This thesis proposes a real-time processing backend for the read-out electronics of cryogenic particle detectors, aiming to process a high number of frequency tones for a FDM multiplexing scheme (≈ 1000). It's focused in the design and development of a high performance processing approach, high scalabity and minimizing the used resources.

High performance and scalable processing scheme

- FPGA based architecture, allowing a high level of system scalability and huge processing capabilities, at reasonable costs. Custom Linux (Yocto), applications (C,assembler,Julia) matched with hardware architecture, etc..

HDL resources optimization strategy

- No inferation. Explicit declaration of BRAMs (Block RAMs), FIFOs (First In, First Out buffers), algorithms directly mapped on DSP Slices primitives. Constraints files for Place and Route step, etc..

Projects I'm involved

- QUBIC, for CMB observation (**ITeDA**),
- ECHo, neutrino mass determination (**IPE**).

Micromachined sensors for Cryogenic Particle Detectors

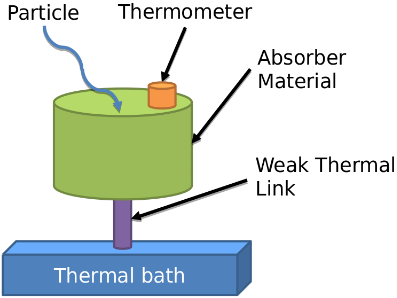


Figure: Basic detector scheme

Possible applications

- Photons,
- Mass Spectrometry,
- Heavy Ion Physics,
- Neutrinos,
- Dark Matter (WIMPs),
- CMB observations,
- α, β, γ spectroscopy, astronomy.

Thermometer Detector

- TES - Transition Edge Sensors,
- MKID – Microwave Kinetic Inductance Detector,
- NTD - Semiconductor Thermistors.
- MMC – Metallic Magenitc Calorimeters

Read–Out Electronics

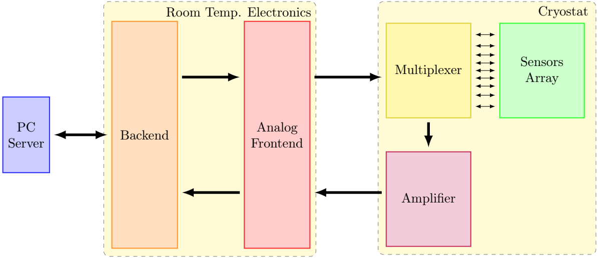


Figure: Cryogenic Particle Detectors' Read-Out System.

Backend Scheme

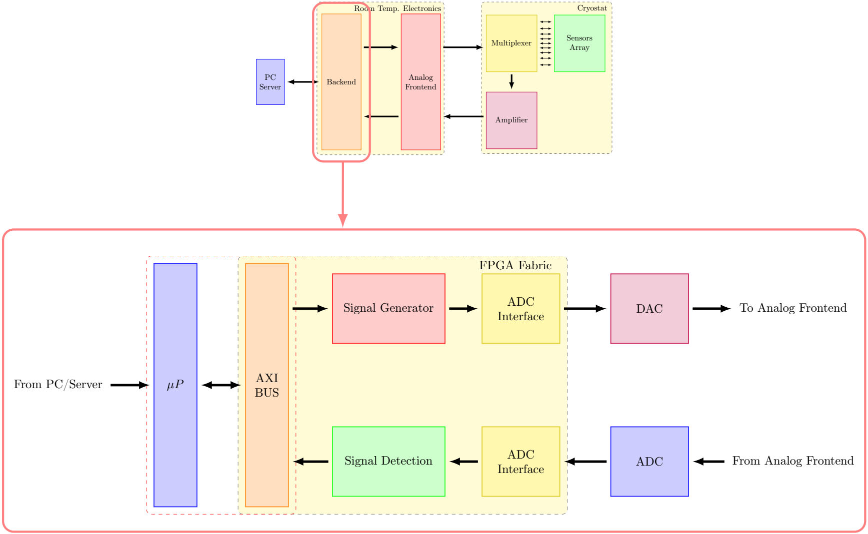


Figure: Complete proposed Backend.

Current State of Development

The Goertzel Algorithm

- It's a localized Discrete Fourier Transform.

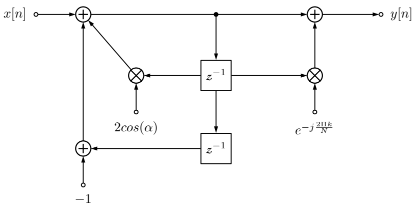


Figure: Goertzel Algorithm - Space-State of a 2nd order IIR system.

[1] "An Algorithm for the evaluation of Finite Trigonometric Series", G. Goertzel, The American Mathematical Monthly, Vol. 65, No. 1, January 1958, pp- 34-35

[2] "Goertzel algorithm generalized to non-integer multiples of fundamental frequency", P. Sysel and P. Rajmic, EURASIP, 2012, 2012:56

Transfer function:

$$H_k(z) = \frac{1 - e^{-j\frac{2\pi k}{N}} z^{-1}}{1 - 2cos(\alpha) z^{-1} - z^{-2}}$$

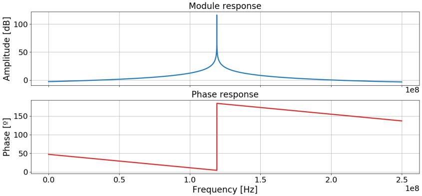


Figure: Transfer Function Bode for: $F_s = 500 Msps$, $N = 256$ and $F_{det} = 119 MHz$

Window Functions

- Spectral Analysis -> Window function.

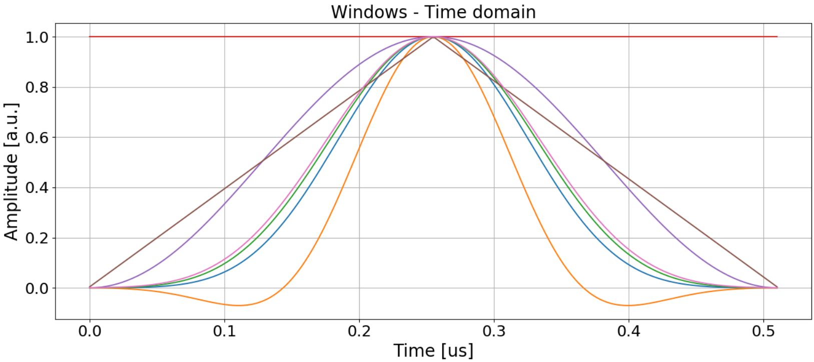


Figure: Some different windows types ($N = 256$).

[3] "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform", F. J. Harris, Proceedings of the IEEE, Vol. 66, No. 1, January 1978.

Window Functions

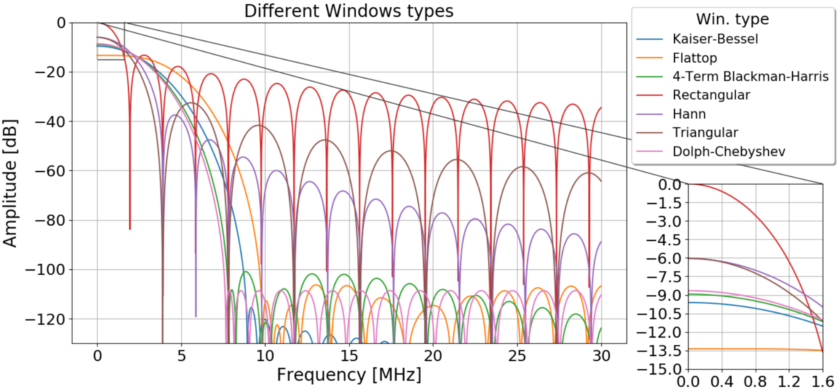


Figure: Frequency response of each window ($N = 256$).

[3] “On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform”, F. J. Harris, Proceedings of the IEEE, Vol. 66, No. 1, January 1978.

Goertzel Filter

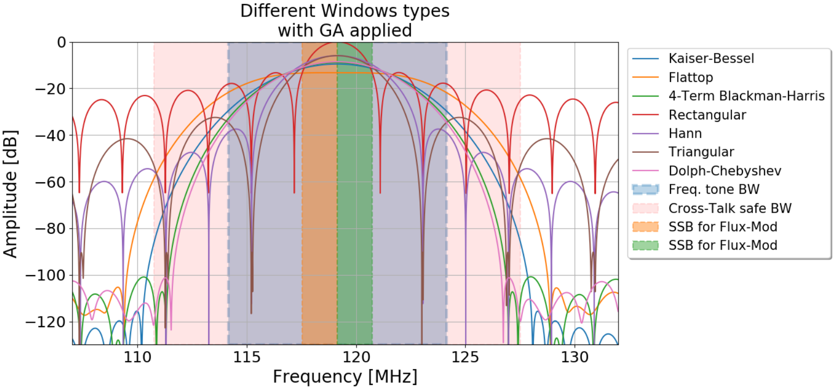


Figure: Goertzel Filter frequency response for each window: $F_s = 500 Msps$, $N = 256$ and $F_{det} = 119 MHz$ and $N = 256$.

Goertzel Filter - Full Characterization

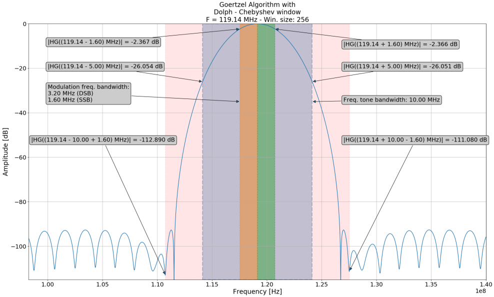


Figure: Goertzel Filter frequency response for a Dolph-Chebyshev window: $F_s = 500 Msps$, $N = 256$ and $F_{det} = 119 MHz$ and $N = 256$.

*: ECHo parameters.

Parameter	Value
Highest side lobe level [dB]	-92.66 / -92.65
Side Lobe Fall-off [dB/dec]	-42.79
Side Lobe Fall-off [dB/oct]	-12.88
Coherent Gain [dB]	8.59
ENBW [Bins]	5.42
ENBW [MHz]	10.58
3.0 dB BW [MHz]	3.6
6.0 dB BW [MHz]	5.05
60.0 dB BW [MHz]	13.72
Shape Factor	3.81
Scallop Loss [dB]	0.88
Processing Gain/Loss [dB]	7.34
Worst Case Process Loss [dB]	8.22
|H(f_det)| [dB]*	0
|H(f_det +/- 5 MHz)| [dB]*	-26.05
|H(f_det +/- 1.6 MHz)| [dB]*	-2.37
|H(f_det + (1.6 - 10) MHz)| [dB]*	-111.08
|H(f_det – (1.6 + 10) MHz)| [dB]*	-112.89

A proof of algorithm feasability in an FPGA implementation

1. Implemented a *baremetal* application running on the Zynq μP,
2. Stored simulated modulated signal in the board's RAM,
3. Read and sent to the *Goertzel_Filter_Unit_0* block,
4. Retrieved the processed data with a Python application running on a PC.

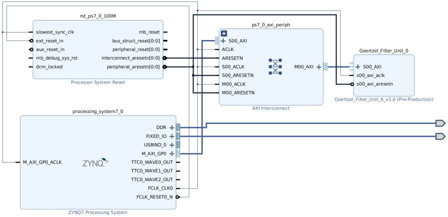


Figure: ZedBoard Vivado Block Diagrama implementation

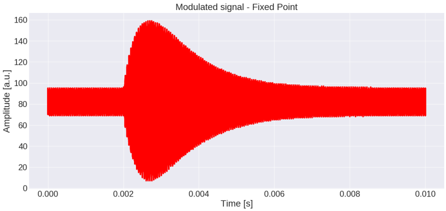


Figure: Modulated (in amplitude and phase) Bi-tonal signal

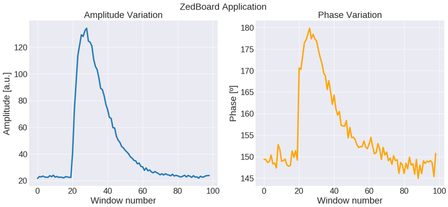


Figure: FPGA modulation signals reconstruction.

Zynq UltraScale+ (ZCU102) Implementation

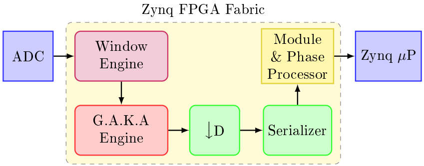


Figure: Detection stage block diagram

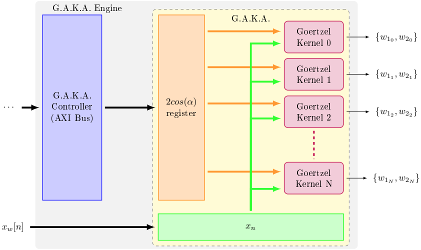


Figure: G.A.K.A.: Goertzel Algorithm Kernel Array

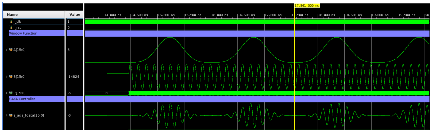


Figure: Window Function Engine Simulation

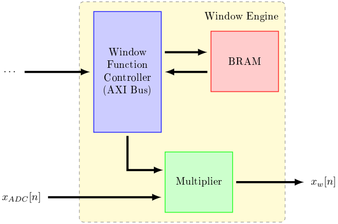


Figure: Window Function Engine

Resources Estimation

FPGA resource estimation

Module	DSP Slices	BRAMs
Win. Engine	1*$N_{ADC_{ch}}$	4
G.A.K.A.	2*N_f	0

Where N_f it's the number of required filters, and $N_{ADC_{ch}}$ it's the number of channels that the ADC interface with the FPGA.

FPGA resource estimation example for I-Q modulation scheme and 400 frequency tones

Module	DSP Slices	BRAMs
Win. Engine	1*2 = 2	4
G.A.K.A. (I Channel)	2*400 = 800	0
G.A.K.A. (Q Channel)	2*400 = 800	0
Total	1602	4

Using, for example, Zynq UltraScale+ (2520 DSP Slices) -> 918 DSP Slices left for further processing.

Future Work

Future Work:

- Design, develop and test the last module: Module & Phase Processor,
- Achieve multi-tonal signal detection with the Goertzel Filter approach,
- A cavity's S21 parameter measurement,
- Implement the approach in ECHo and QUBIC electronics,
- Implement the generation stage: store a pre-computed signal (previously optimized for low crest-factor, going from $C_F = 20$ to $C_F = 1.67$.) in a BRAM, and play it back with a DAC. Then, test the whole system working (technical-demonstrator),
- Develop formal applications for signal processing using the technical-demonstrator. Also, drivers and software integration,
- Start with a fully custom electronic design approach (no evaluation boards).

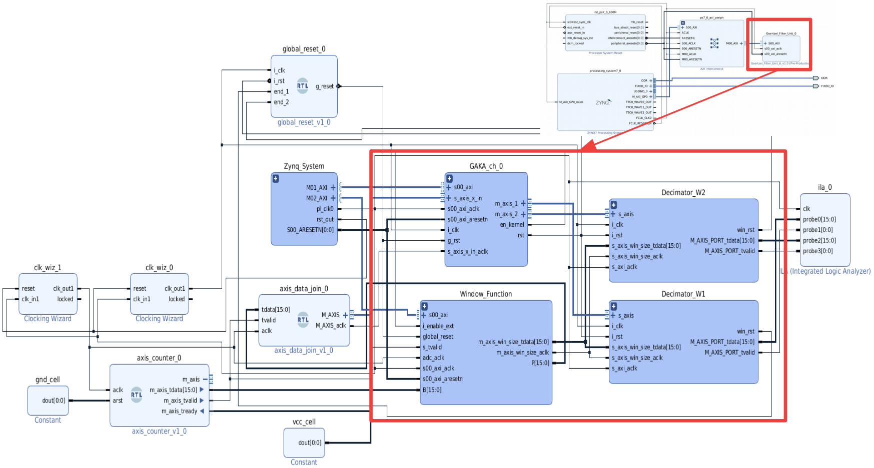
Selected electronics for prototype:

- Processor and FPGA: Zynq UltraScale+ MPSoC Evaluation Board (ZCU102),
- ADC: Texas Instruments ADS54J69 Evaluation Board (2 x 16bits, 500 Msps),
- DAC: Texas Instruments DAC39J84 Evaluation Board (4 x 16bits, 2.8 Gsps).

In the meantime, FOOD..I mean, cultural exchange :)

Vielen Dank!

Back-Up


global_reset_0
i_clk
i_rst
end_1
end_2
RTL
g_reset
global_reset_v1_0
Zynq_System
M01_AXI
M02_AXI
pl_clk0
rst_out
S00_ARESETN[0:0]
GAKA_ch_0
s00_axi
s_axis_x_in
s00_axi_aclk
s00_axi_aresetn
i_clk
g_rst
s_axis_x_in_aclk
m_axis_1
m_axis_2
en_kernel
rst
Decimator_W2
s_axis
i_clk
i_rst
s_axis_win_size_tdata[15:0]
s_axis_win_size_aclk
s_axi_aclk
win_rst
M_AXIS_PORT_tdata[15:0]
M_AXIS_PORT_tvalid
ila_0
clk
probe0[15:0]
probe1[0:0]
probe2[15:0]
probe3[0:0]
ILA (Integrated Logic Analyzer)
clk_wiz_1
reset
clk_in1
clk_out1
locked
Clocking Wizard
clk_wiz_0
reset
clk_in1
clk_out1
locked
Clocking Wizard
axis_data_join_0
tdata[15:0]
tvalid
aclk
RTL
M_AXIS
M_AXIS_aclk
axis_data_join_v1_0
Window_Function
s00_axi
i_enable_ext
global_reset
s_tvalid
adc_aclk
s00_axi_aclk
s00_axi_aresetn
B[15:0]
m_axis_win_size_tdata[15:0]
m_axis_win_size_aclk
P[15:0]
Decimator_W1
s_axis
i_clk
i_rst
s_axis_win_size_tdata[15:0]
s_axis_win_size_aclk
s_axi_aclk
win_rst
M_AXIS_PORT_tdata[15:0]
M_AXIS_PORT_tvalid
axis_counter_0
aclk
arst
RTL
m_axis
m_axis_tdata[15:0]
m_axis_tvalid
m_axis_tready
axis_counter_v1_0
gnd_cell
dout[0:0]
Constant
vcc_cell
dout[0:0]
Constant
ZYNQ
DDR
FIXED_IO

Back-Up

The approach can be implemented on different devices:

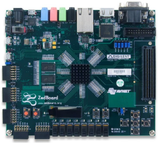


Figure: ZedBoard, Zynq 7020, 363 €

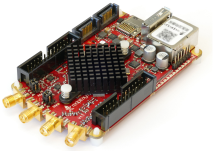


Figure: Red Pitaya, Zynq 7010, 279 €

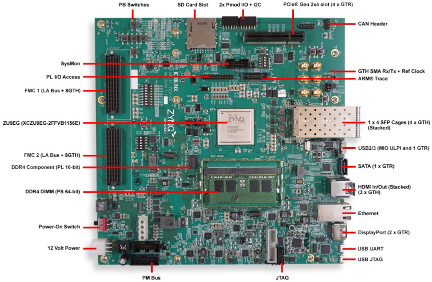


Figure: Zynq Ultrascale+ MPSoC, 2531 €

Back-Up

Filter test (in Python) using ECHo technical demonstrator raw data

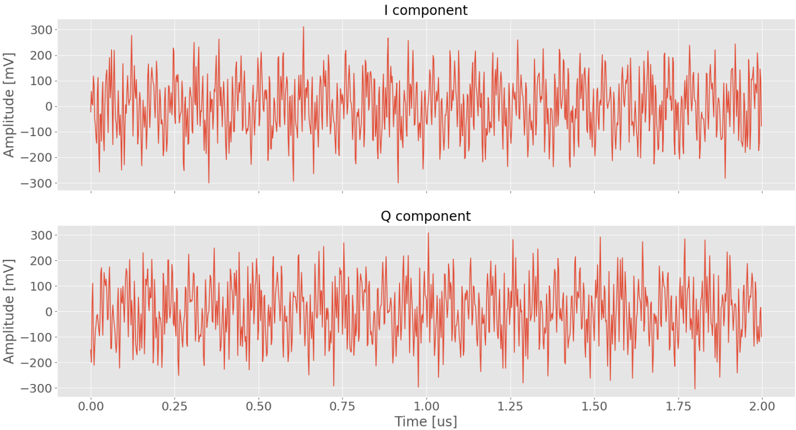


Figure: I and Q signals (raw data).

There where 5 different components: 119 MHz, -80 MHz (pilot), -15.6 MHz, -147 MHz, and 150.6 MHz.

Back-Up

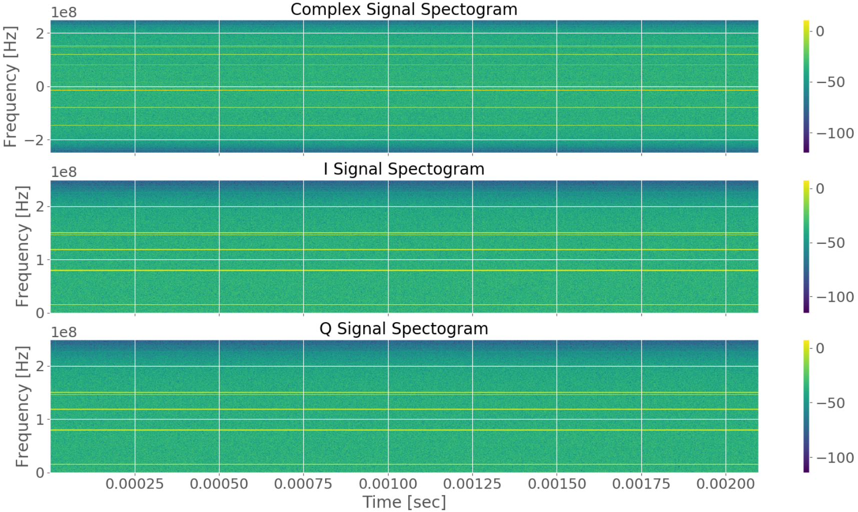


Figure: Periodogram (spectrogram) of the above I and Q signals. C = I + jQ

Back-Up

Pulse reconstruction - 1/2

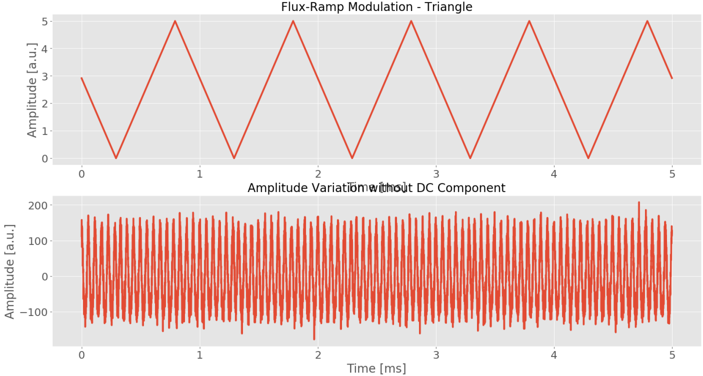


Figure: Filter applied to the raw data.

- SQUIDs are stimulated by a triangle signal -> Flux-ramp modulation method.
- Triangle signal: 1 kHz, 5 V. SQUID: 565 mV -> 1 flux period.
- Expected recovered period: $F_{flux} = 2 * (5V/565mV) * 1kHz = 17.06kHz$
- Measured: $F_{flux_m} = 17.98kHz$.

Back-Up

Low crest factor I/Q modulated multitonal signal. Up to N different frequency tones, spaced 10 MHz from each other:

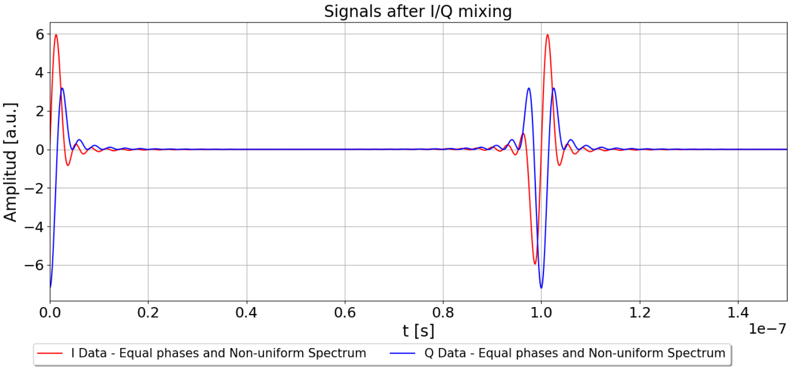


Figure: Time domain, 32 probe signals, non-optimzed (equal phases).

Back-Up

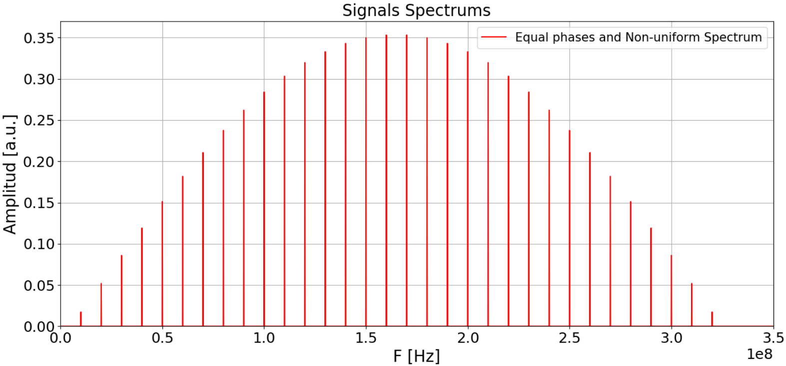


Figure: Frequency domain, 32 probe signals, non-optimzed (equal phases).

Back-Up

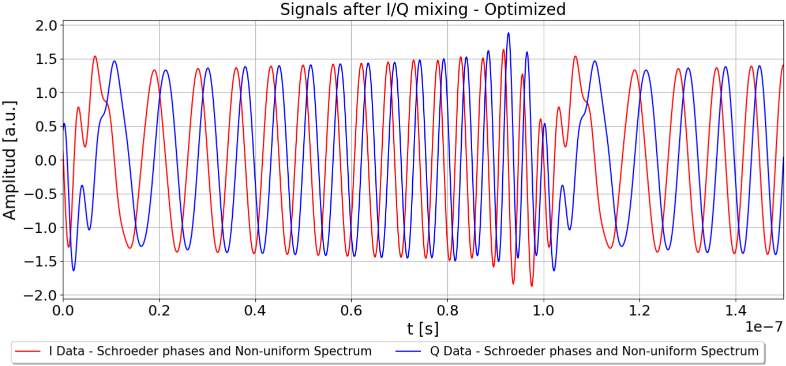


Figure: Time domain, 32 probe signals, optimzed but non-uniform spectrum.

[4] “Synthesis of Low-Peak-Factor Signals and Binary Sequences with Low Autocorrelation”, M. R. Schroeder, IEEE Transactions on Information Theory, January 1970.

Back-Up

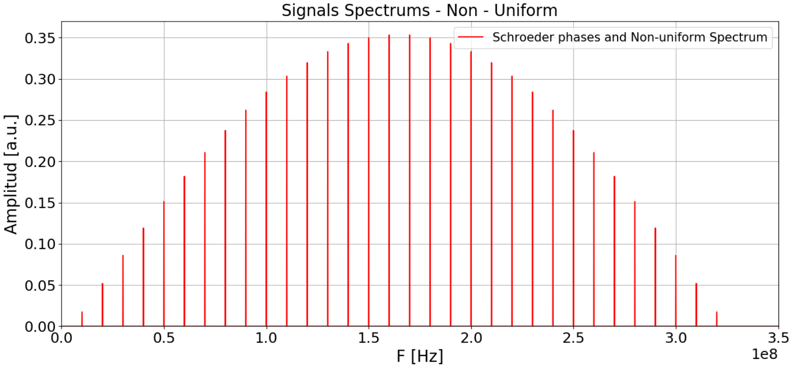


Figure: Frequency domain, 32 probe signals, optimzed but non-uniform spectrum.

Back-Up

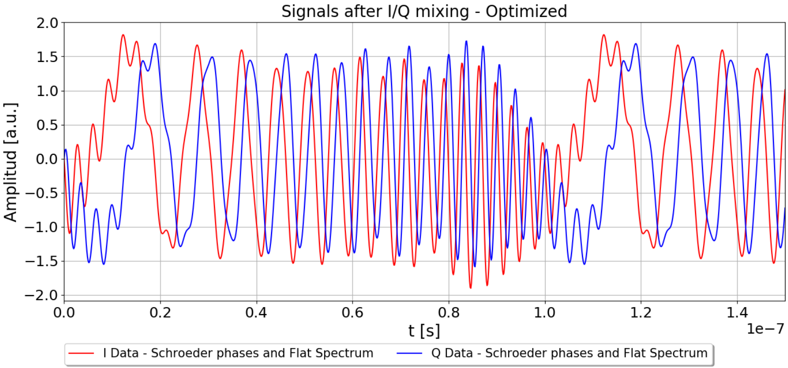


Figure: Time domain, 32 probe signals, optimzed with uniform spectrum.

Back-Up

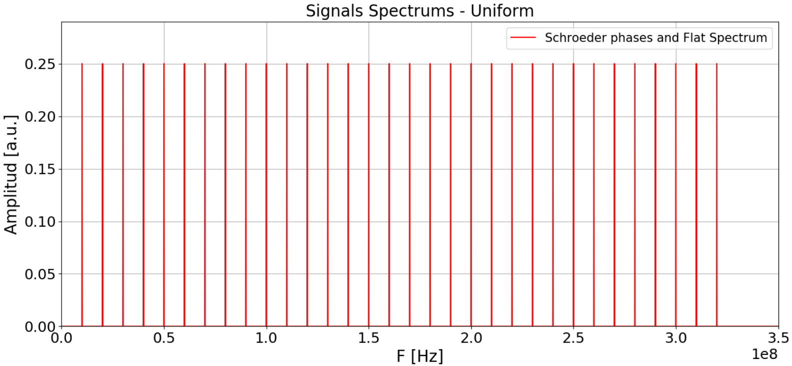


Figure: Frequency domain, 32 probe signals, non-optimzed with uniform spectrum.

Crest Factor comparisons

Crest Factor	Nº Tones	No opt.	Opt. Non-U	Opt. Uniform
$\frac{V_{max}}{V_{rms}}$	32	5.95	1.87	1.90
$\frac{V_{max}}{V_{rms}}$	400	20	1.67	1.89

Back-Up

Other approaches

- “Synthesis of discrete-interval binary signals with specified Fourier amplitude spectra”, A. V. den Bos, Int. Journal of Control, Vol. 30, No. 5, pp. 871-884, 1979,
- “Peak Factor minimization usint Time-Frequency Domain swapping algorithm”, E. V. der Ouderaa, IEEE Transactions on Instrumentation and Measurement, Vol. 37, No. 1, March 1988,
- “Optimization of multisine excitation for bioimpedance measurement device”, J. Ojarand, IEEE Proceedings Instrumentation and Measurement, pp. 829-832, May 2014,
- “An improved crest factor minimization algorithm to synthesize multisine with arbitrary spectrum”, Y. Yang, Physiol. Meas. 36 895, April 2015.