Fundación FULGOR


# Diseño Digital Avanzado

## Unidad 2 - Implementación de Filtros

**Dr. Ariel L. Pola**
ariel.pola@mi.unc.edu.ar
September 19, 2025

# Tabla de Contenidos

# Implementación de Filtros

# Implementación de Filtros

## Simulación de Filtros FIR e IIR

### Efectos de cuantización

- Ejecutar los script de python con el objetivo de comprender los efectos de cuantización de los coeficientes.
  - fir_filter_direct_form.ipynb
  - iir_filter_direct_form.ipynb
  - IIR_Filter_Design.ipynb
- Instanciar y ejecutar el testbench de cada uno de los filtros.
  - filtro_fir.v, tb_filtro_fir.v
  - iir.v, filter_tb.v
  - iir_top.v, iir_filter.v, coeffSec1.v, coeffSec2.v, coeffSec3.v, tb_iir_filter.v

## Primer modelo

- Implementar en FPGA el filtro FIR según los siguientes archivos
  - top_design.v
  - signal_generator.v
  - filtro_fir.
  - SarTruncFP.v
- Agregar los IPs VIO e ILA para controlar en forma remota el diseño

## Laboratorio

- Considerar un sistema de transmisión compuesto por una señal senoidal y un filtro pasa bajo con las siguientes características:
  - Señal senoidal compuesta por dos frecuencias $f_1 = 17kHz$ ($A = 0.5$) y $f_2 = 1.5kHz$ ($A = 1.0$)
  - Frecuencia de muestreo $f_s = 48kHz$
  - Filtro pasa bajo con frecuencia de corte $f_{cut} = 8khz$

## Laboratorio

- Desarrollo del modelo
  1. Utilizando el scripts de Python coeff.ipynb, determinar los coeficientes del filtro para una frecuencia de corte de $f_{cut} = 8khz$. El filtro debe tener una longitud de 15 coeficientes.
  2. Realizar el diagrama en bloques del filtro.
  3. Generar un proyecto con los archivos entregador por la cátedra con la herramienta Vivado.
  4. Configurar el archivo mem.hex con las señales senoidales especificadas previamente utilizando el script genmem.py.
  5. Generar los coeficientes del filtro utilizando el script coeff.ipynb para los siguientes valores de frecuencias de corte $f_{cut} = 0.5khz, 8khz, 18khz$.
  6. Configurar el filtro en verilog con los valores de los coeficientes cuantizados (sintetizar cada filtro por separado).
  7. Implementar en FPGA y graficar las señales senoidales pre y pos filtradas.