Escuela Politécnica Superior,
Grado en Informática

**Asignatura: Diseño Automático de Sistemas**

# Práctica 5 Implementación de un filtro digital

3 de Mayo de 2022

UNIVERSIDAD NEBRIJA

UNIVERSIDAD
NEBRIJA

**Pablo Sarabia Ortiz**

# Índice/Tabla de contenidos

# 1. Introducción

## 1.1. Presentación

Se va a implementar un filtro FIR (Finite Impulse Response). Los filtros digitales tienen diversas aplicaciones entre ellas modificar las señales de audio para adecuarlas y eliminar ruidos o sonidos no deseados. El filtro FIR es de los más sencillos y habituales en los diseños digitales consta de sumas y multiplicaciones.

## 1.2. Bibliografía recomendada

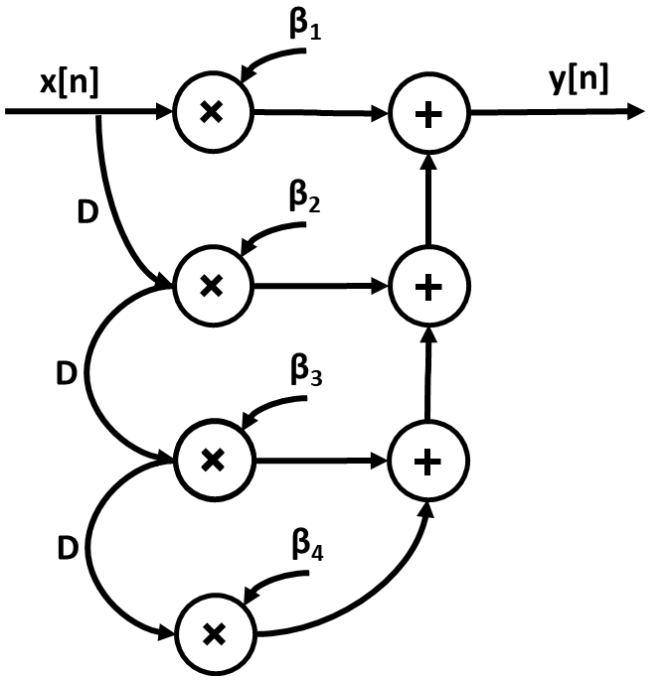
Podéis consultar las diapositivas y cualquier articulo en internet sobre los filtros FIR y su implementación en VHDL.

## 1.3. Objetivo

Demostrar una implementación real de los filtros vistos en clase. En esta práctica se valorará la implementación y no es necesario entregar un testbench. Se calificará también las técnicas aplicadas de pipeline y agrupación de operandos para mejorar el diseño del sistema. Si se han realizado estas técnicas por favor adjuntarlas en el informe final.

# 2. Filtro FIR

El filtro que se desea implementar tiene 4 etapas tal y como se muestra en la siguiente figura;




La entrada de los datos va a ser de ancho 8 bits y los coeficientes son también de 8 bits. Un aspecto importante para recordar es que el ancho en bits de una multiplicación es M+N siendo M y N la longitud en bits de los factores. Y el ancho en bits de una suma es N+1 siendo N la longitud del sumando.

Se recomienda hacer pipeline tras cada operación e ir registrando cada resultado. Cómo el resultado solo tiene 10 bits es necesario tomar solo los 10 bits más significativos.

Para reducir el tamaño de las señales intermedias se puede implementar el filtro en cascada, consiste en agrupar los sumandos/factores para que el ancho de las palabras esté equilibrado.

Se proporciona la interfaz en el fichero fir.vhd que contiene el siguiente código:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity fir_filter is
port (
      clk         :in std_logic;
      rst         :in std_logic;
      -- Coeficientes
      beta1 :in std_logic_vector(7 downto 0);
      beta2 :in std_logic_vector(7 downto 0);
      beta3 :in std_logic_vector(7 downto 0);
      beta4 :in std_logic_vector(7 downto 0);
      -- Data input 8 bit
      i_data      :in std_logic_vector(7 downto 0);
      -- Filtered data
      o_data      :out std_logic_vector(9 downto 0)
      );
end fir_filter;
```

Para facilitar la implementación se recomienda utilizar tipos propios con arrays para mediante un bucle for realizar la multiplicación o suma.

# 3. Evaluación y entrega

## 3.1. Grupos

Se utilizarán los mismos grupos que en la anterior práctica.

## 3.2. Puntuación

Este hito se corresponde con un **10%** del valor del proyecto final. Para su evaluación basta con enseñar el código, explicarlo y si se han realizado optimizaciones (pipeline, retiming o agrupación de opeaciones) adjuntar los correspondientes diagramas y explicaciones. El código será revisado por el profesor en la siguiente sesión de prácticas. Para la revisión es necesario llevar el código del prov