

**ELECTRÓNICA II**  
**3º Trabalho de Laboratório**  
**FILTRO DIGITAL FIR**

José T. de Sousa e José Gerald

**Instituto Superior Técnico**  
**Departamento de Engenharia Electrotécnica e de Computadores**  
**Área Científica de Electrónica**

## Objectivo:

O objetivo deste trabalho é projetar e simular um filtro FIR (*Finite Impulse Response*), de modo a familiarizar o aluno com o fluxo de projeto de circuitos digitais e emulação em FPGA.

O filtro é primeiro projetado conceptualmente utilizando a ferramenta *Matlab* ou a alternativa gratuita *Octave*.

O circuito é depois descrito na linguagem Verilog. Utilizar-se-ão as ferramentas de projecto de FPGA da empresa Xilinx e placa Basys2 da empresa Digilent.

## Conteúdo:

Este trabalho é disponibilizado com os seguintes pastas e ficheiros:

- eiilab3.[odt,pdf]: este guia.
- octave: pasta do código Matlab/Octave
  - fir.m: *script* de Matlab/Octave para geração do sinal de entrada e coeficientes do filtro
  - fir\_analysis.m: *script* de Matlab/Octave para análise do resultado de simulação.
- verilog: pasta do código Verilog
  - fir\_tap.v: descrição Verilog de uma unidade de ponderação e atraso do filtro FIR
  - fir\_path.v: descrição Verilog do filtro FIR
  - freq\_synt.v: descrição Verilog de um gerador de onda quadrada de frequência variável
  - detector.v: descrição Verilog de um detector de amplitude simples
  - top.v: sistema contendo o filtro, o gerador de onda quadrada e o detector de amplitude
  - Makefile: ficheiro de comandos para simular em ambiente Linux com o simulador Icarus Verilog
  - tb: pasta dos testes (*testbenches*)
    - freq\_synt\_tb.v: teste para o gerador de onda quadrada
    - detector\_tb.v: teste para o detector de amplitude
    - fir\_path\_tb.v: descrição Verilog do teste do filtro FIR
    - top\_tb.v: teste para o sistema
- fpga: pasta dos ficheiros necessários para implementação em FPGA
  - README: instruções básicas para utilização do programa ISE da Xilinx
  - top.ucf: ficheiro de restrições para o dispositivo FPGA (localizações dos pinos e limite inferior da frequência de funcionamento)
  - basys2\_ref\_manual : manual da placa

# 1 Análise

## 1.1 Previsão teórica

1. Execute o script *fir.m* utilizando os programas Matlab/Octave e explique para que serve.
2. Usando os programas Matlab ou Octave obtenha a resposta em frequência do filtro gerado no ponto anterior.
3. Usando os mesmos programas e a equação de recorrência obtenha a resposta no tempo a um impulso unitário, escalão unitário e ao sinal gerado no ponto 1.

## 1.2 Descrição Verilog

1. Explique o módulo *fir\_tap*.
2. Explique o módulo *fir\_path* e desenhe o esquema do circuito. Em que forma está o filtro FIR?
3. Explique o módulo *freq\_synt*.
4. Explique o módulo *detector*.
5. Explique o módulo *top* e desenhe o esquema do circuito total. Em particular, explique como se obtém a frequência de amostragem de 48kHz a partir do sinal relógio de 50MHz presente à entrada da FPGA.

# 2 Simulação

1. Simule o módulo *fir\_path* usando o programa ISE. Note que os ficheiros *x.hex* e *h.hex* gerados pelo script *fir.m* devem ser colocados na diretoria do projecto ISE. Garanta que carrega no botão de Start depois de aberto o simulador ou não será gerado o ficheiro de saída. Alternativamente e se usa Linux poderá usar o simulador gratuito *Icarus Verilog* através da Makefile disponibilizada e visualizar as ondas com a ferramenta *Gtkwave*. (Use o Google para saber mais)
2. Execute o script *fir\_analysis.m* para analisar o resultado obtido.
3. Explique a forma de onda da resposta no tempo, em particular a distorção observada.
4. Explique o gráfico do espectro do sinal de saída e leia dele um valor aproximado do ganho DC e a atenuação da componente de 12kHz.
5. Obtenha o atraso temporal do filtro por análise da resposta no tempo e compare com o previsto teoricamente.
6. Alterando os ficheiros que achar necessários, simule novamente o filtro aplicando à entrada um impulso e um escalão unitários. Obtenha os gráficos das respostas no tempo a estes dois sinais. Obtenha os gráficos da resposta na frequência do filtro ao impulso unitário e mostre que confirmam os resultados obtidos para o ganho DC e o atraso.
7. Simule o sistema (*top*) completo e interprete os resultados.

### 3 Implementação em FPGA

1. Implemente o filtro usando o programa ISE e obtenha o número de elementos lógicos utilizadas (LUTs), o número de multiplicadores, e o número de blocos de memórias (BRAMs, se aplicável); obtenha iterativamente a máxima frequência de trabalho variando o período mínimo especificado no ficheiro .ucf.
2. Implemente agora o sistema total e obtenha novamente os recursos de hardware utilizados. Explique porque razão o tamanho do sistema completo é inferior ao do filtro isolado.
3. Programe a FPGA da Basys2 com o sistema *top*. Altere a frequência do sinal à entrada do filtro utilizando os interruptores na placa. Registe qualitativamente a intensidade da luz no LED0 para as frequências testadas.

### 4 Projeto

Nesta parte deverá alterar o filtro dado de modo a obter um filtro FIR com as seguintes características:

1. Atenuação inferior a 3dB para 1kHz
2. Atenuação superior a 40dB para 12kHz

Nas alterações a efectuar, poderá variar as frequências de amostragem e corte e a ordem do filtro desde que ainda caiba na FPGA.

Repita os pontos 1 a 3 para o filtro que projetou.

### 5 Comentário

Compare os resultados teóricos, de simulação e experimentais.

Compare os filtros dado e projetado em termos de:

1. Tamanho do circuito
2. Frequência de trabalho
3. Potência, sabendo que esta é proporcional ao tamanho do circuito e à frequência de trabalho.