

Laboratório de Sistemas Digitais
Ano Letivo 2022/23
Projeto Final – enunciado 3

Filtro de Mediana Móvel

1. Introdução

Um filtro de mediana móvel é um componente frequente em sistemas de processamento de sinal e ou imagem. O sinal de saída, como consequência do resultado da filtragem, é uma versão “suavizada” do sinal de entrada em que os efeitos de variações bruscas são de alguma forma minimizados tal como se pretende representar na figura 1. Neste trabalho, pretende projetar-se e implementar na FPGA e testar no kit Terasic DE2-115 um sistema de filtragem de mediana móvel de largura 5. Para efeitos de teste o sinal de entrada estará disponível numa ROM de 256x8 bits e o sinal de saída deverá ser escrito numa RAM com as mesmas dimensões.

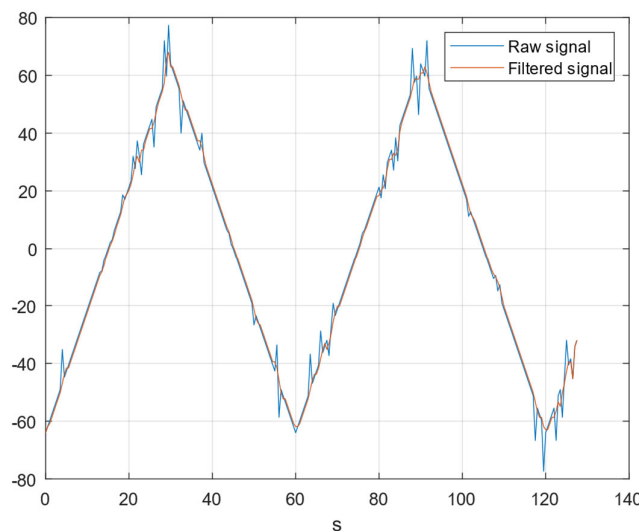


Figura 1: Filtro de média móvel de largura 4.

2. Descrição do funcionamento e requisitos

De uma forma geral, este trabalho consiste em modelar em VHDL, simular, sintetizar, implementar na FPGA e testar no kit Terasic DE2-115 um sistema digital de acordo com as principais especificações seguintes:

- O sistema deve ler um sinal de teste escrito numa ROM 256x8 bits. Este sinal é baseado numa onda triangular com ruído impulsivo. O período da onda triangular é de 60s. A frequência de amostragem é de 2Hz e como tal a duração total do sinal gravado na ROM é de 128 s. Este componente será previamente fornecido.
- O conteúdo da ROM é do tipo *signed* com 8 bits.

- A RAM deverá ter as mesmas dimensões da ROM, com 1 porto para escrita síncrona e leitura assíncrona.
- O sistema deve mostrar sequencialmente em 4 displays Hexadecimais o conteúdo da ROM (sinal de entrada)
- O sistema deve mostrar sequencialmente em 4 displays Hexadecimais o conteúdo da RAM (sinal filtrado)
- Cada amostra do sinal filtrado y_k deverá ser calculada por uma unidade de ordenação que permita obter o valor da mediana (valor central da ordenação) de grupos de 5 amostras do sinal de entrada x

$$y_k = \text{median}(x_{k-2}, x_{k-1}, x_k, x_{k+1}, x_{k+2}), k = 2, \dots, 253 \quad (1)$$

$$y_0 = x_0, y_1 = x_1, y_{254} = x_{254}, y_{255} = x_{255}$$

O sistema deverá ter os seguintes sinais de controlo:

- START: KEY0
- FILTER_ON: SW(0),
- RESET_RAM: KEY1, reset da RAM
- RESET: KEY2, reset geral do sistema
- HEX3...HEX0: Conteúdo da ROM (sinal da entrada)
- HEX7...HEX4: Conteúdo da RAM (sinal de saída)
- O *top-level* do circuito deverá ser implementado com recurso a representação estrutural em VHDL.

3. Sugestões para implementação

A implementação deste sistema deve seguir uma estratégia faseada, de acordo com a descrição que se sugere e de acordo com a proposta de arquitetura da figura 2:

Fase 1 (2 valores): Implementar um subsistema de geração de endereços que permita ler na forma *signed* o conteúdo da ROM.

Utilize o ficheiro auxiliar `ImpNoisyTriangSignal_ROM_256x8.vhd` donde poderá ler o sinal de entrada.

Fase 2 (2 valores): Implementar um subsistema de geração de endereços que permita ler na forma *signed* o conteúdo da RAM. Implemente o modo de funcionamento RESET_RAM que permita preencher a RAM com `x"00"`. Neste modo, utilize a frequência de 50 MHz para a geração dos endereços da RAM.

Fase 3 (4 valores): Implemente um subsistema intermédio de armazenamento que tenha como saída as 5 amostras do sinal de entrada envolvidas na determinação da mediana. Sugere-se que estes valores sejam sequencialmente lidos da Rom e escritos num banco de 5 registos que proporcionem um mecanismo de deslocamento ao nível das amostras de 8 bits. Na prática, trata-se duma linha de atraso em modo *serial-in – parallel-out* para palavras de 8 bits.

Fase 4 (4 valores): Uma unidade de ordenação deverá ler em paralelo o conteúdo dos 5 registos e implementar a fórmula (1). O resultado deverá ser escrito na RAM com endereçamento adequado. Sugere-se uma implementação iterativa do algoritmo “Bubble Sort” a partir de unidades elementares de ordenação de 2 valores.

FASE 5 (4 valores): Implementar uma unidade de controlo (máquina de estados) que permita o funcionamento de acordo com a interface proposta. O sistema deve necessariamente começar por fazer RESET à RAM e só depois evoluir para o processo de filtragem.

Fase 6 (4 valores): Interligar os blocos anteriores (*top-level*) e visualizar a filtragem em tempo real. Para controlo de qualidade, as sequências geradas devem seguir as sequências que constam no ficheiro auxiliar: `LSD_MedF_Signals.xlsx`

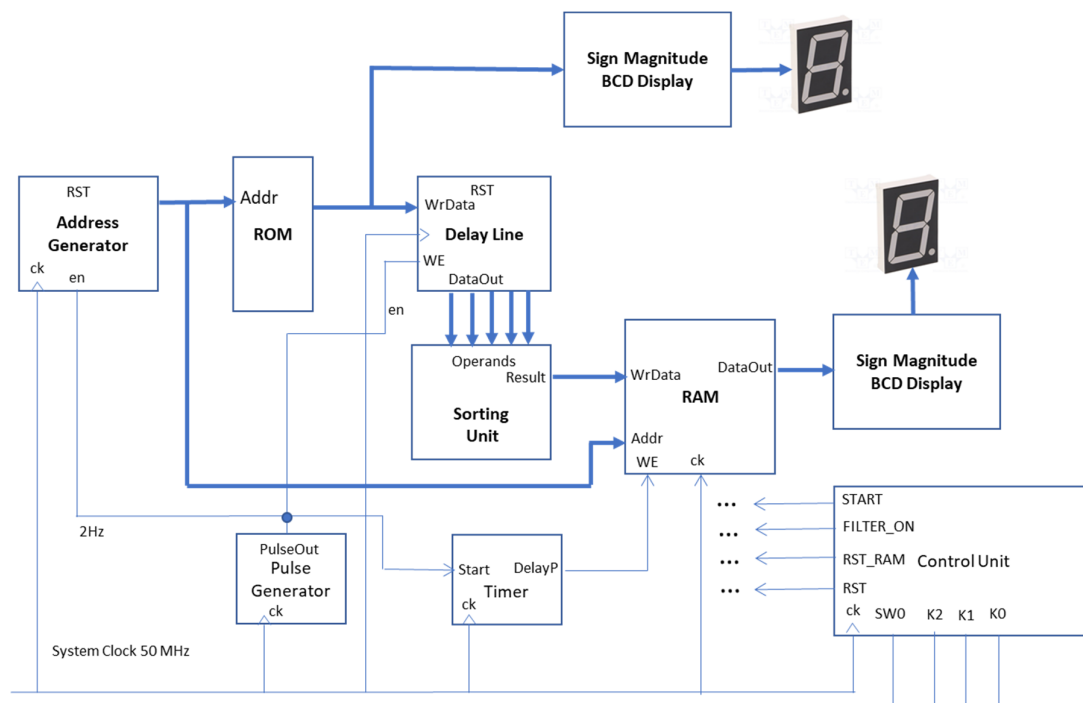


Figura 2: Proposta de arquitetura (ligações incompletas)