

# Filtro de Média Móvel

Laboratório de Sistemas Digitais

Luíz Fernando, Miguel Vila  
(107535) luiz.ferreira@ua.pt, (107276) miguelovila@ua.pt

UNIVERSIDADE DE AVEIRO

Aveiro, junho 2022

# Introdução

Para este trabalho foi proposto o desenvolvimento de um filtro de média móvel, um componente frequente em sistemas de processamento de sinal e ou imagem, responsável pela suavização de um sinal de entrada na FPGA Terasic DE2-115.

Este filtro, de largura 4, irá suavizar um sinal de entrada disponível numa ROM de 256x8 bits. Já o sinal de saída deverá ser guardado numa RAM com as mesmas dimensões.

## Manual de Utilização

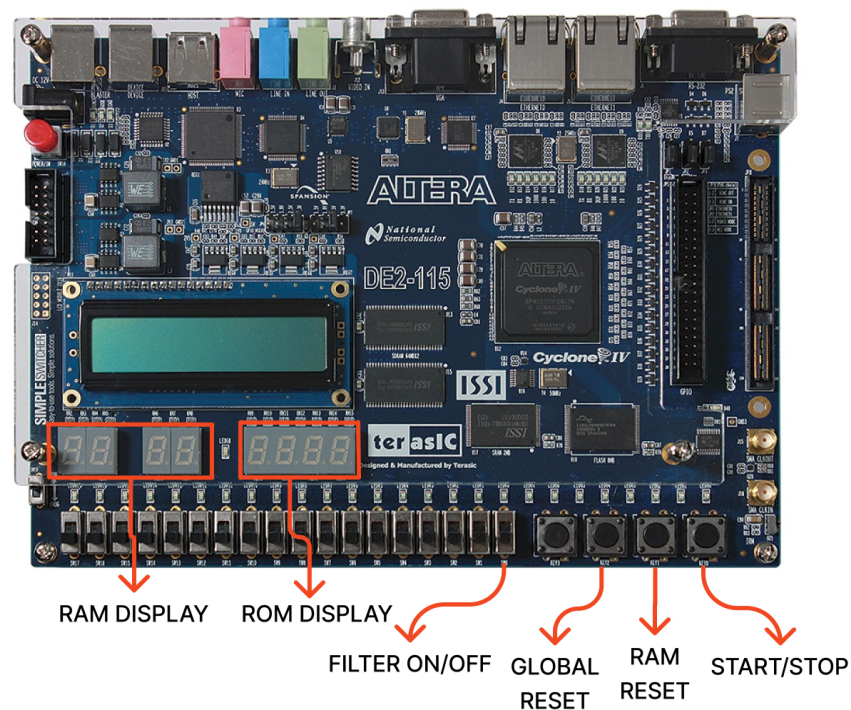


Figura 1: Funções e visualização de dados na Terasic DE2-115

O filtro possui as seguintes funcionalidades:

- RAM Display: é onde pode ser feita a visualização dos valores guardados

na memória RAM no endereço atual.

- ROM Display: é onde pode ser feita a visualização do valor presente na memória ROM no endereço atual.
- Filter ON/OFF: caso esteja ligado, o valor armazenado na memória RAM resulta de uma média entre os dois valores anteriores, o valor atual e o próximo valor. Caso esteja desligado, o valor armazenado na memória RAM corresponde ao valor atual lido da memória ROM.
- RAM Reset: efetua o reset da memória RAM escrevendo todos os endereços com x"00". Permite fazer reset à memória e continuar a execução a partir do endereço anterior.
- Global Reset: efetua o reset dos componentes essenciais e da memória RAM. Reinicia o processo de filtragem a partir do endereço "00000000".
- Start/Stop: dependendo do estado atual do sistema, para ou inicia o processo de filtragem de valores.

## Arquitetura e Implementação

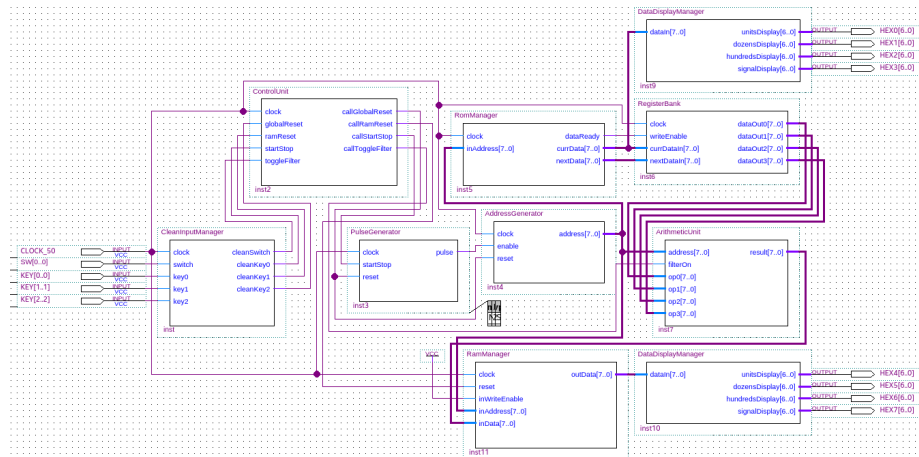


Figura 2: Diagrama de funcionamento do filtro de média móvel.

De uma forma simplista, o projeto é controlado por uma unidade central de controle (uma máquina de estados) que controla um gerador de endereços ligado diretamente à ROM, um registo, responsável por armazenar momentaneamente os dados obtidos da ROM e enviá-los para uma unidade aritmética capaz de

efetuar os cálculos de filtragem do sinal recebido da ROM. Em seguida, este sinal devidamente filtrado é guardado na RAM.

Tanto os valores armazenados na ROM como os na RAM passam por módulos que os preparam para serem mostrados nos displays de 7 segmentos do kit Terasic DE2-115 como números inteiros com sinal em base 10.

## CleanInputManager

Componente responsável pela sincronização e remoção de interferências mecânicas provenientes dos sinais das keys e dos switches. Optou-se por juntar o DebounceUnit e o sincronizador de forma a simplificar o top-level do projeto.

## ControlUnit

Componente responsável pelo comportamento das restantes entidades. Este trata-se de uma máquina de estados que suporta os seguintes estados:

- `t_GlobalReset`: efetua o reset dos componentes essenciais (PulseGenerator e AddressGenerator) e avança para o estado `t_RAMRESET`.
- `t_RAMRESET`: efetua o reset da memória RAM escrevendo todos os endereços com `x"00"` preservando o endereço anterior. Se for executado enquanto o sistema estiver no estado `t_STOPPED`, mantém esse mesmo estado após o reset. Se for executado enquanto o sistema estiver no estado `t_RUNNING`, mantém esse mesmo estado após o reset. Numa primeira execução (quando o estado anterior é `t_GlobalReset`) passa ao estado `t_RUNNING`.
- `t_RUNNING` e `t_STOPPED`: permite a geração ou bloqueio da geração de endereços.

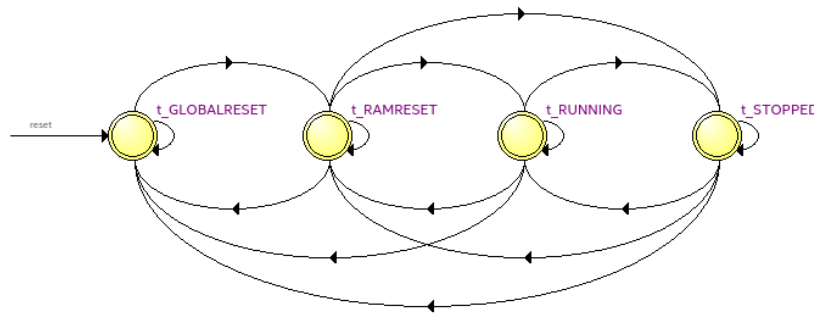


Figura 3: Diagrama de estados da ControlUnit.

## RomManager

Entidade responsável por obter o valor do endereço atual e o valor do próximo endereço. Tem como base uma máquina de estados de 4 estados:

- `t_IDLE`: mantém-se neste estado até haver uma alteração de endereço. Quando esta mudança ocorre, avança para `t_CURRADDRESS`.
- `t_CURRADDRESS`: obtém o valor guardado na memória ROM no endereço atual, avançando para o estado `t_NEXTADDRESS`.
- `t_NEXTADDRESS`: gera o próximo endereço e obtém o valor guardado na memória ROM nesse endereço, avançando para o estado `t_DATAREADY`.
- `t_DATAREADY`: gera um pulso indicando que já obteve o valor atual e o próximo valor e regressa ao estado `t_IDLE`.

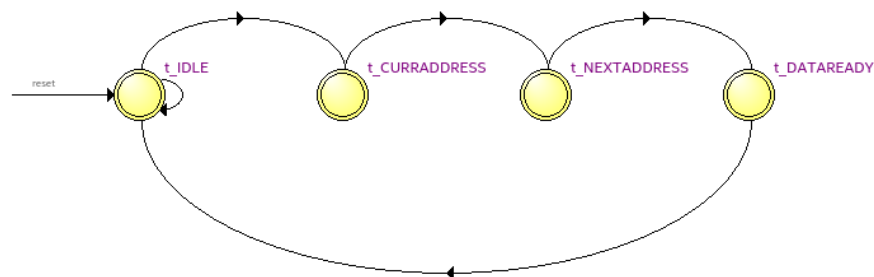


Figura 4: Diagrama de estados da RomManager.

## RamManager

Componente responsável por efetuar o reset à memória RAM através de um gerador de endereços interno, independente do gerador de endereços principal, o que permite que o endereço seja preservado.

Optou-se por encapsular a memória RAM neste componente de forma a simplificar o top-level do projeto.

## Validações

Nesta secção estão as simulações dos componentes mais importantes do processo de filtragem assim como a testagem do projeto.

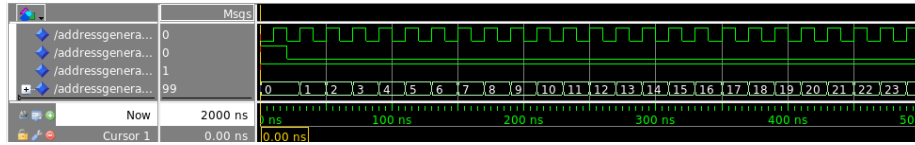


Figura 5: Simulação do gerador de endereços.

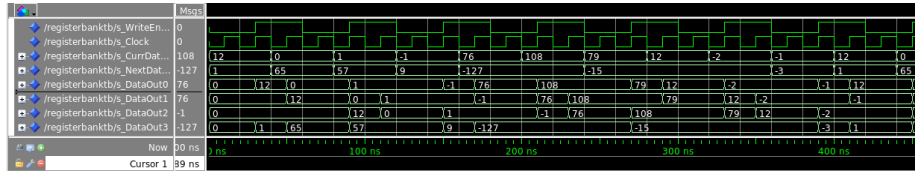


Figura 6: Simulação do RegisterBank.

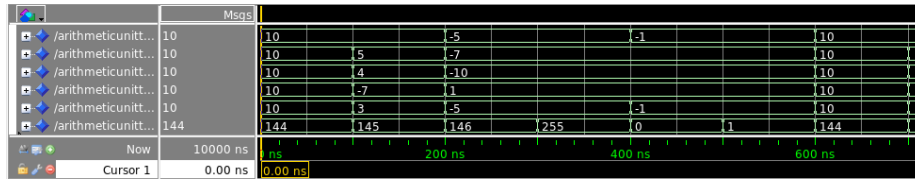


Figura 7: Simulação da ArithmeticUnit.

RAW	-87	-83	-110	-87	-92	-85	-60	-81	-87	-66	-65	-65	-49	-49	-48	-51	-41	-44	-37
FILTER	ON	ON	ON	ON	ON	ON	OFF	OFF	ON	ON	ON	ON	OFF	OFF	ON	OFF	ON	ON	ON
OUT	-87	-83	-91	-93	-93	-81	-60	-81	-73	-74	-70	-61	-57	-49	-48	-47	-41	-37	-38

Figura 8: Testagem dos valores de saída do projeto.

## Conclusão

Tal como esperado, consegui-se realizar o projeto implementando todas as funcionalidades pedidas no enunciado. Todas as dificuldades encontradas no processo de planejamento e desenvolvimento foram superadas como fruto do esforço e dedicação dos autores. Considera-se, assim, que o projeto tenha tido, no geral, boa qualidade e, por isso, autoavaliamos o projeto em 16 valores.

## Contribuições dos autores

Ambos os autores participaram ativamente e com empenho na realização deste projeto. Assim, este trabalho distribui-se tem uma percentagem de participação de aproximadamente 50% para cada um dos elementos do grupo.