



Osciloscópio baseado em FPGA

IISE – Projeto 4

Diogo Miguel Cunha Fernandes, PG47150
José Tomás Lima de Abreu, PG47386

Orientação:

Professor Doutor Jorge Cabral
Professor Doutor Rui Machado
Professora Sofia Paiva

Projeto Integrador em Eletrónica Industrial e Computadores
Universidade do Minho 2021/2022



Índice

Problema e contexto

Especificação

Implementação

Resultados

Calendário de tarefas

Problema e contexto

01 Problema e contexto

02 Objetivos e resultados esperados

Problema e contexto

- **Aumento da complexidade dos sistemas digitais [2]:**
 - Maior dificuldade no desenho de *hardware*;
 - Maior tempo de desenvolvimento.
- **Algoritmos de processamento digital de sinal (PDS):**
 - Complexos;
 - Difíceis de implementar em *hardware*.

Objetivos e resultados esperados

- Explorar técnicas de PDS recorrendo a *High Level Synthesis* (HLS);
- **Osciloscópio básico capaz de:**
 - Amostrar sinais;
 - Aplicar filtros digitais ao sinal amostrado;
 - Apresentação dos sinais num *display* através de interface HDMI;

Especificação

01 Diagrama de Blocos

02 XADC IP

03 Filters IP

04 HDMI IP

Diagrama de Blocos

Diagrama de Blocos

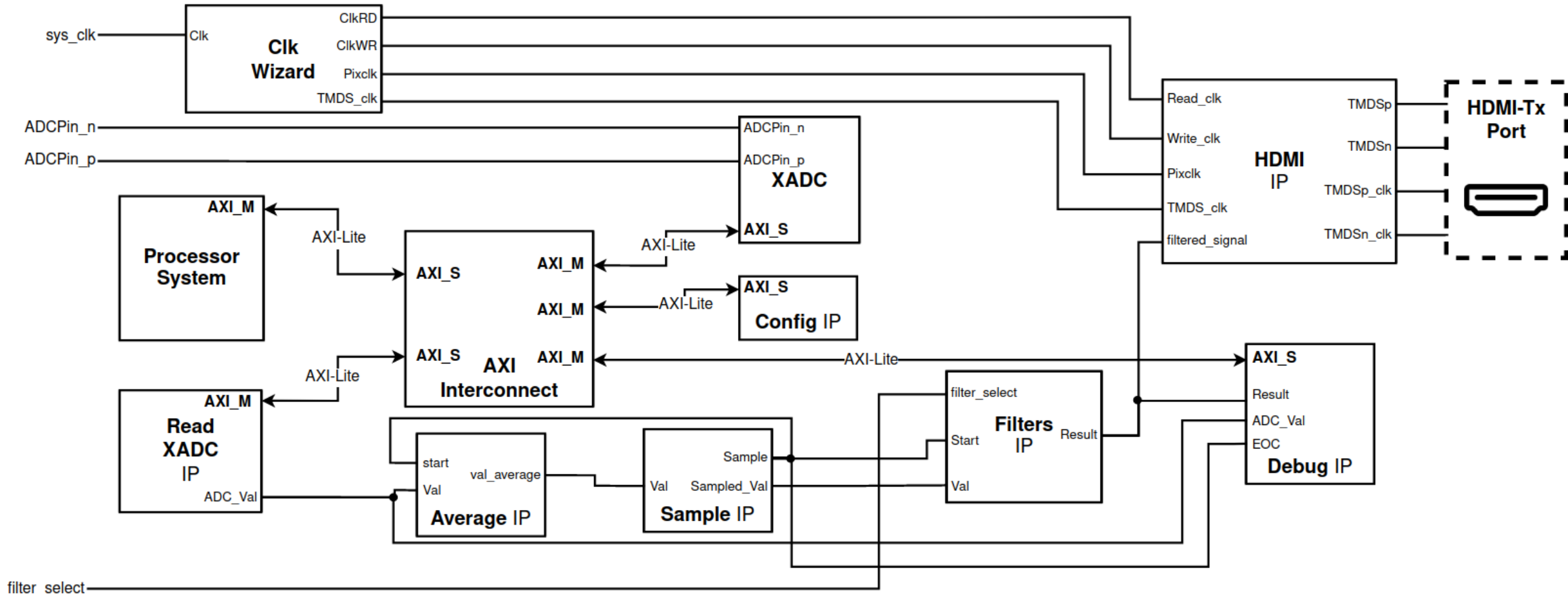
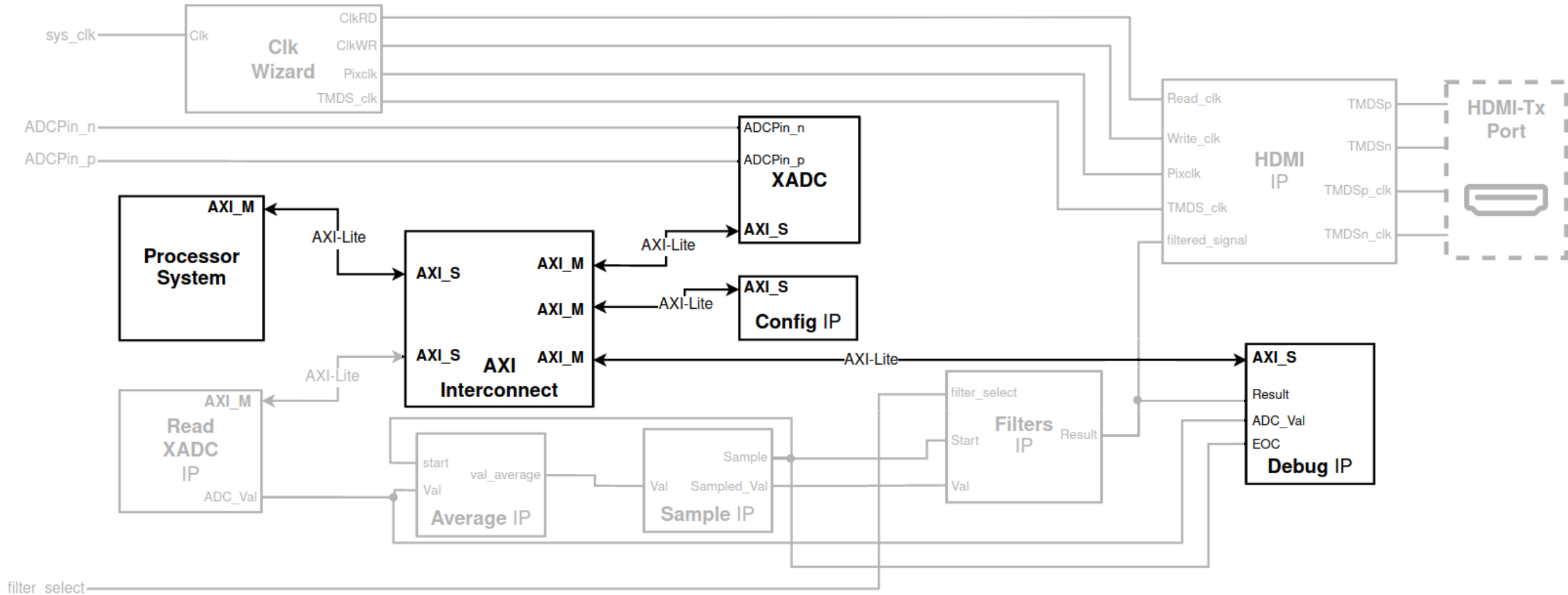


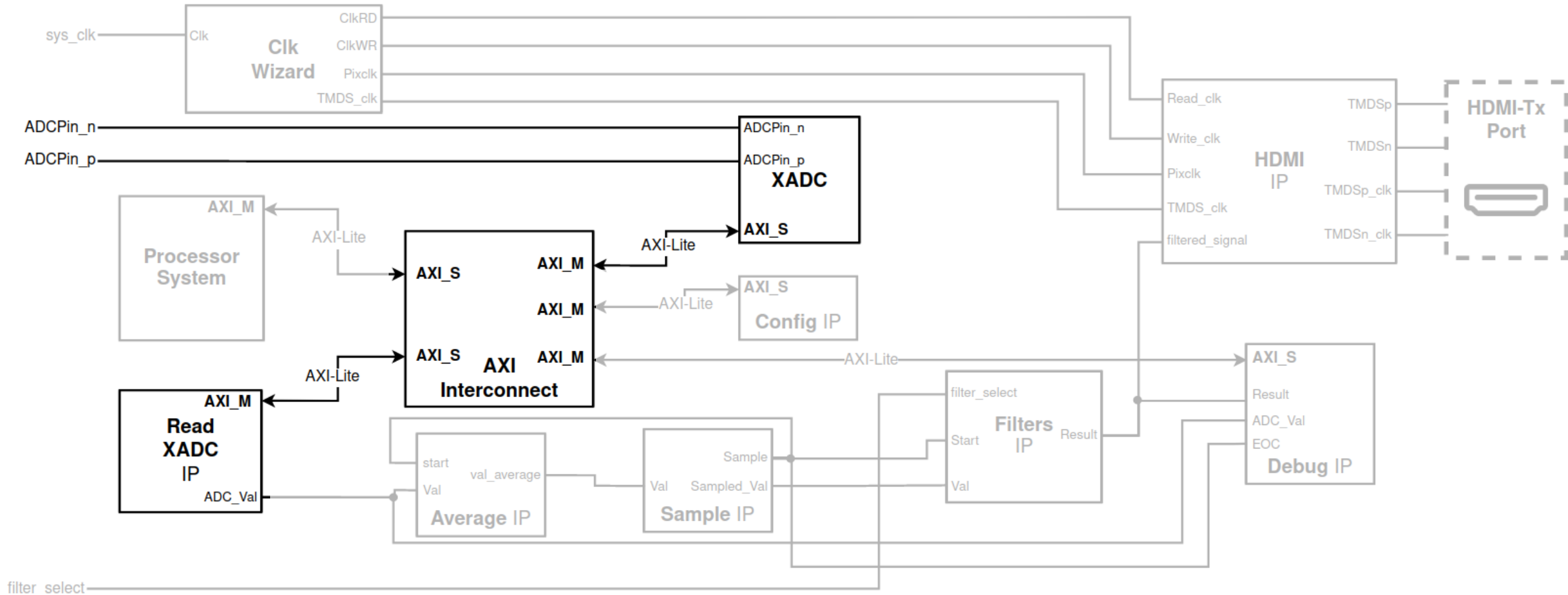
Figura 1 – Diagrama de blocos geral do sistema.

Diagrama de Blocos



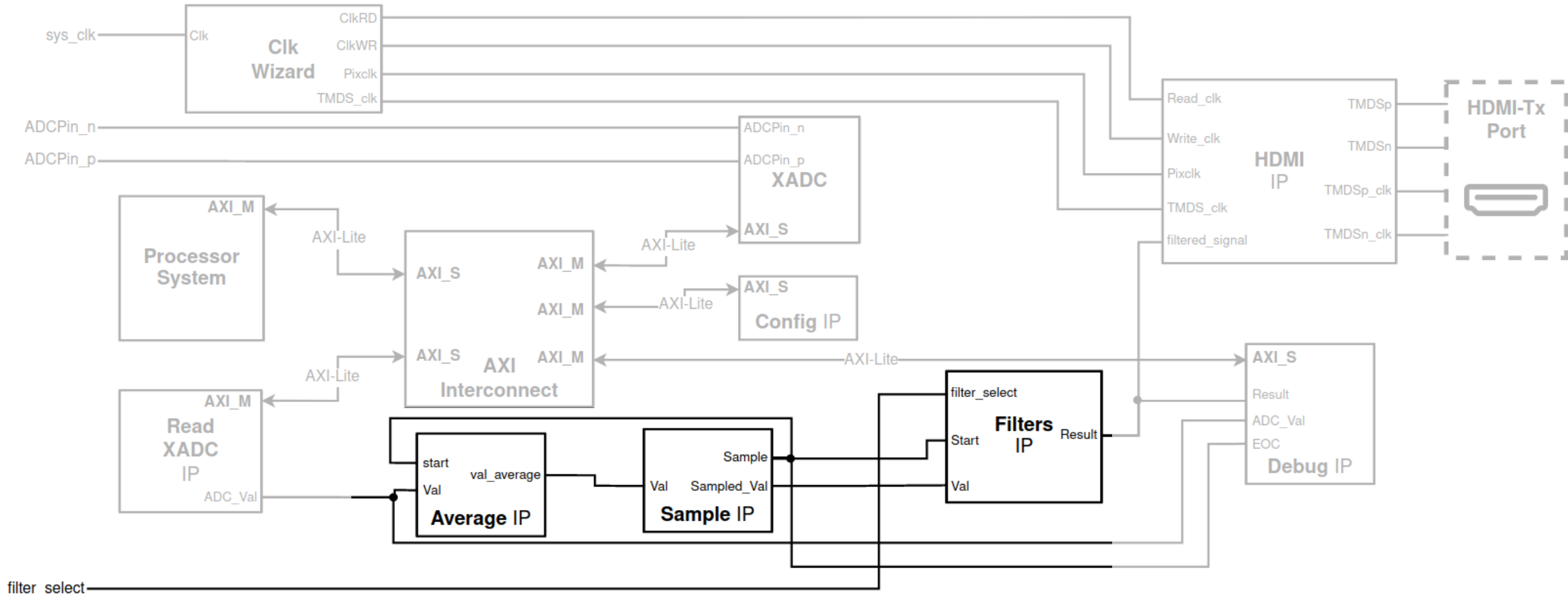
- **PS – Processing System:** para *debug* e interface de configuração;

Diagrama de Blocos



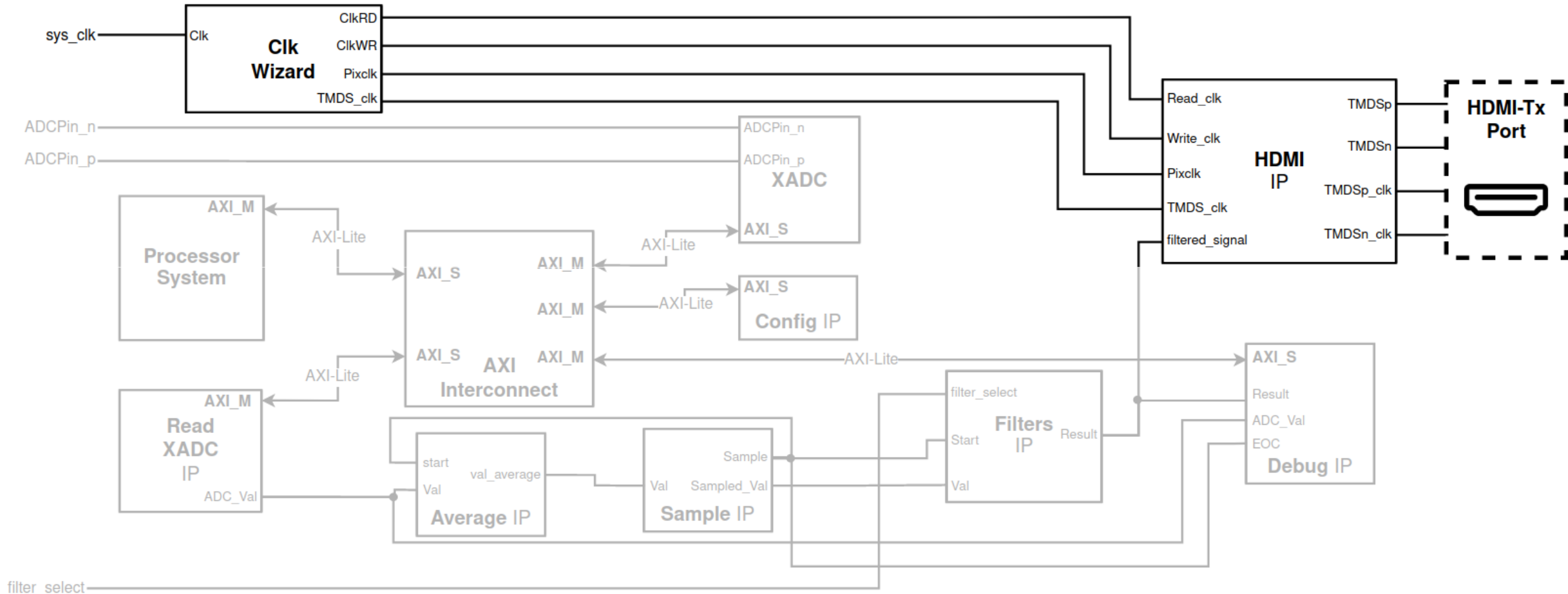
- **Read XADC IP:** leitura dos valores convertidos pelo XADC;

Diagrama de Blocos



- **Filters IP**: aplicação do filtro digital escolhido pelo utilizador ao sinal de entrada;

Diagrama de Blocos



- **HDMI IP:** apresentação do sinal filtrado numa interface HDMI;

XADC IP

XADC IP

1. Realiza uma leitura ao *ConfigIP*;
2. Verifica se foi configurado;
3. Se sim, efetua leituras consecutivas ao XADC.

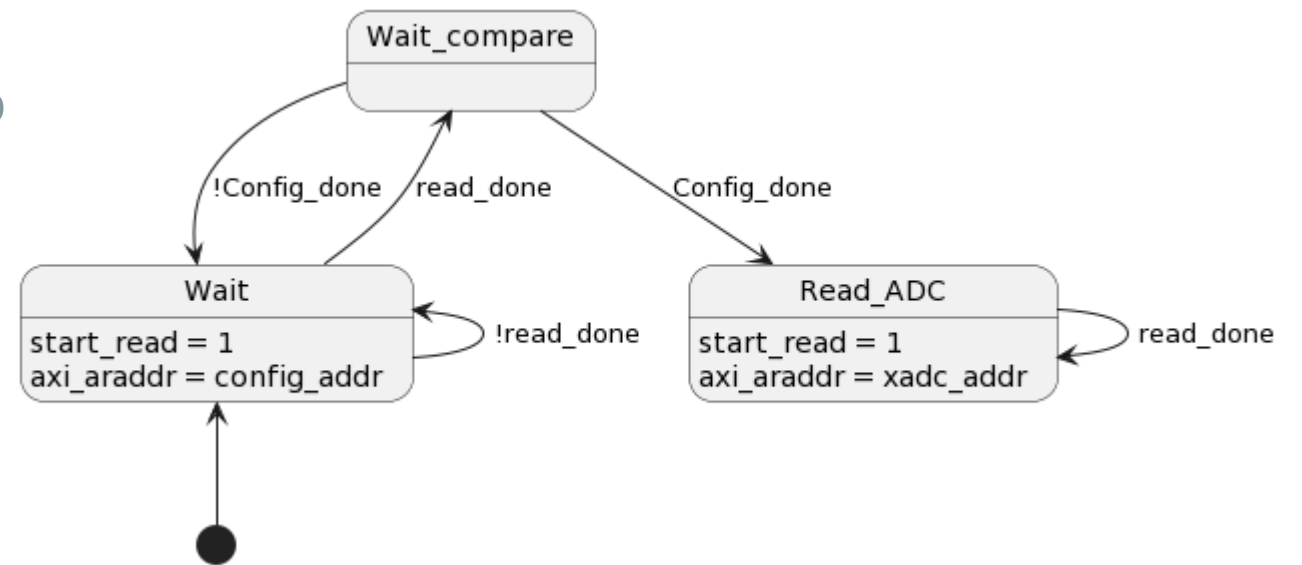


Figura 2 – Diagrama de estados – XADC IP.

Filters IP

Filters IP

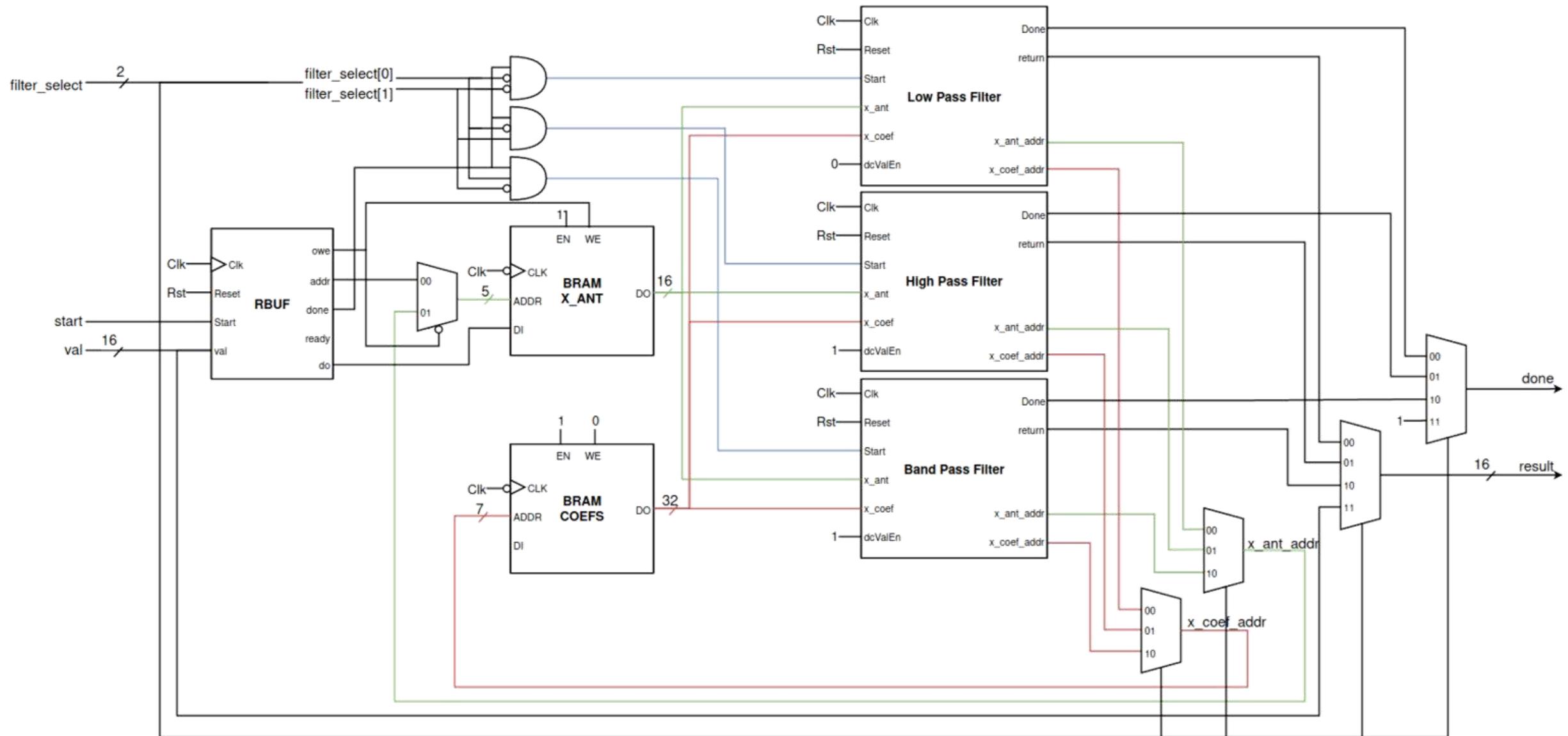


Figura 3 – Diagrama de blocos – Filters IP.

HDMI IP

HDMI IP

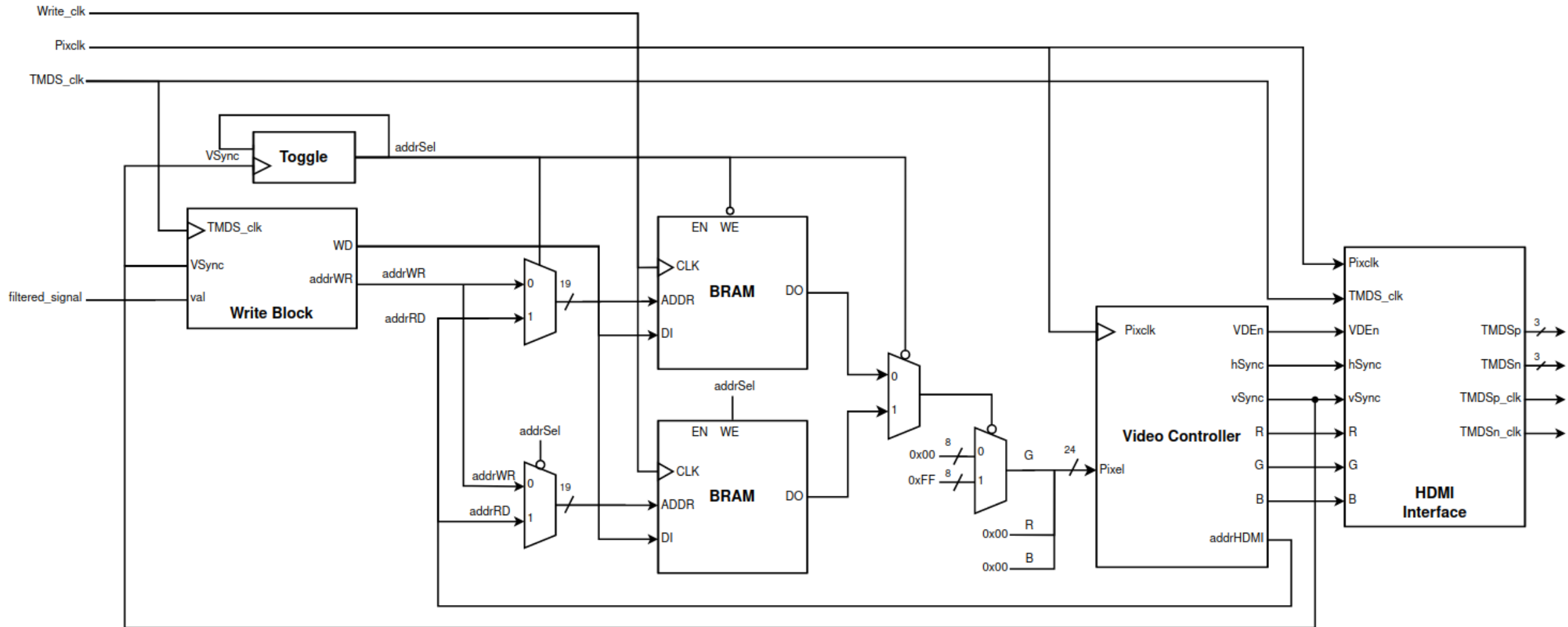


Figura 4 – Diagrama de blocos – HDMI IP.

HDMI IP – Main FSM

1. Limpa a *frame*;
2. Escreve nas coordenadas indicadas pelo valor obtido do filtro (*valIndex*);
3. Espera que a outra *frame* seja lida.

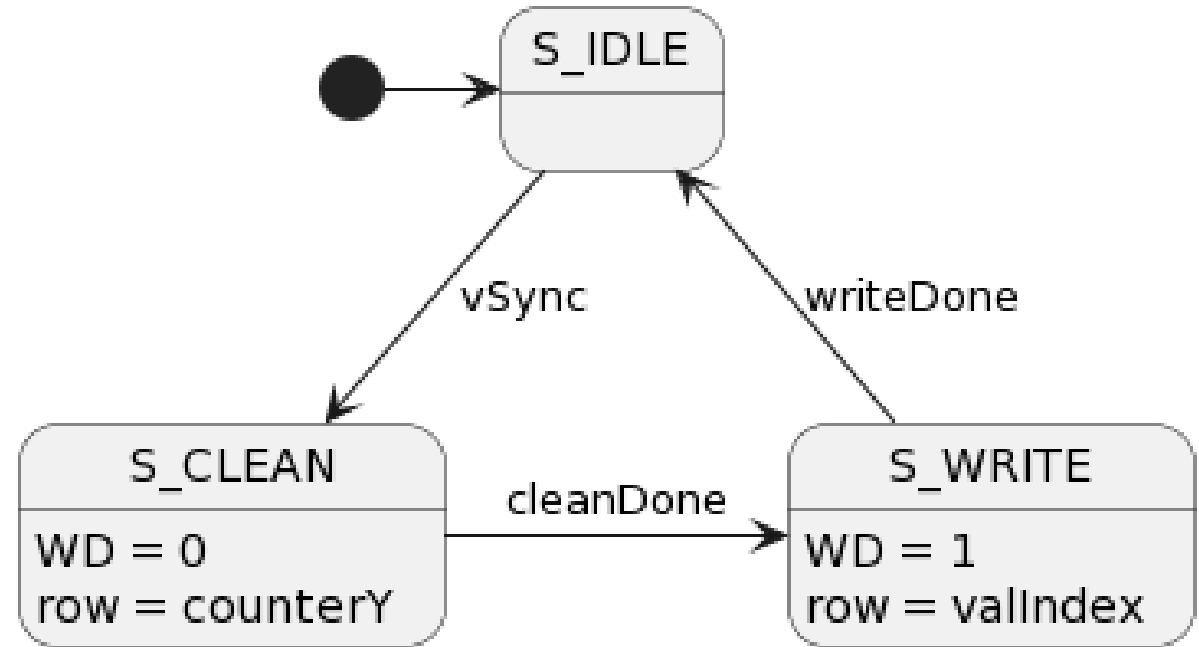


Figura 5 – Diagrama de estados do HDMI IP - Main FSM.

HDMI IP - Write FSM

- Usada nos estados S_CLEAN e S_WRITE da main FSM;

1. Determina qual o endereço (*addrWR*) a escrever com base no valor obtido do filtro:

$$valIndex = \frac{(FFFFh - val) \cdot (height - 1)}{FFFFh}$$

$$addrWR = row.width + x$$

2. Escrita no *addrWR* da BRAM.

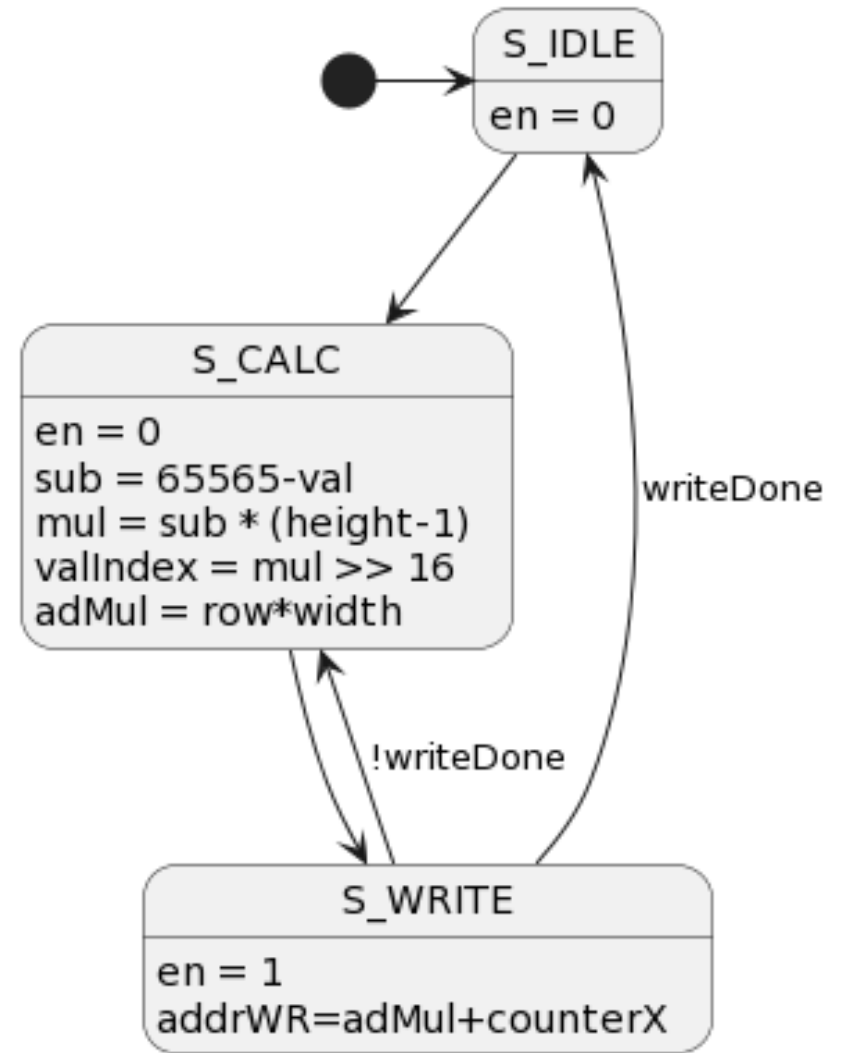


Figura 6 – Diagrama de estados do HDMI IP - Write FSM.

HDMI IP - Read FSM

- Leitura do *addrRd* da BRAM;
- Troca de BRAM quando acabar de ler uma *frame*.

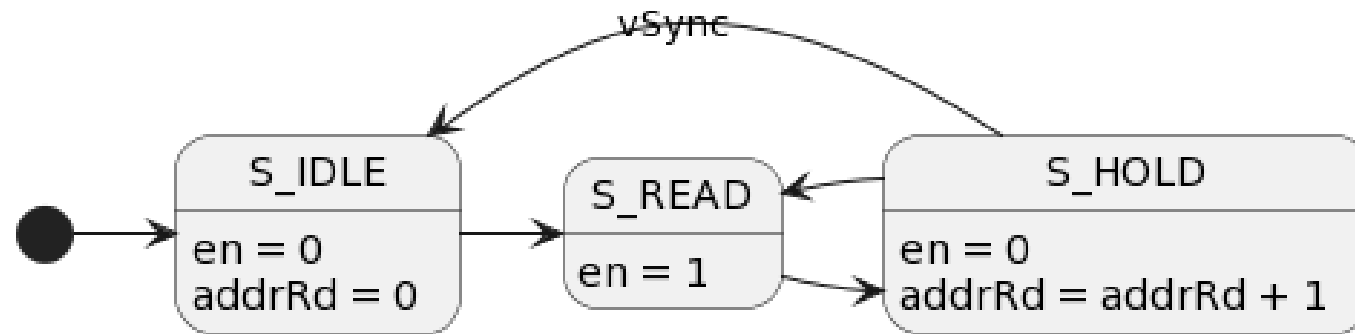


Figura 7 – Diagrama de estados do HDMI IP - Read FSM.

Implementação

01 *Block Design*

02 XADC IP

03 Filters IP

04 HDMI IP

Block Design

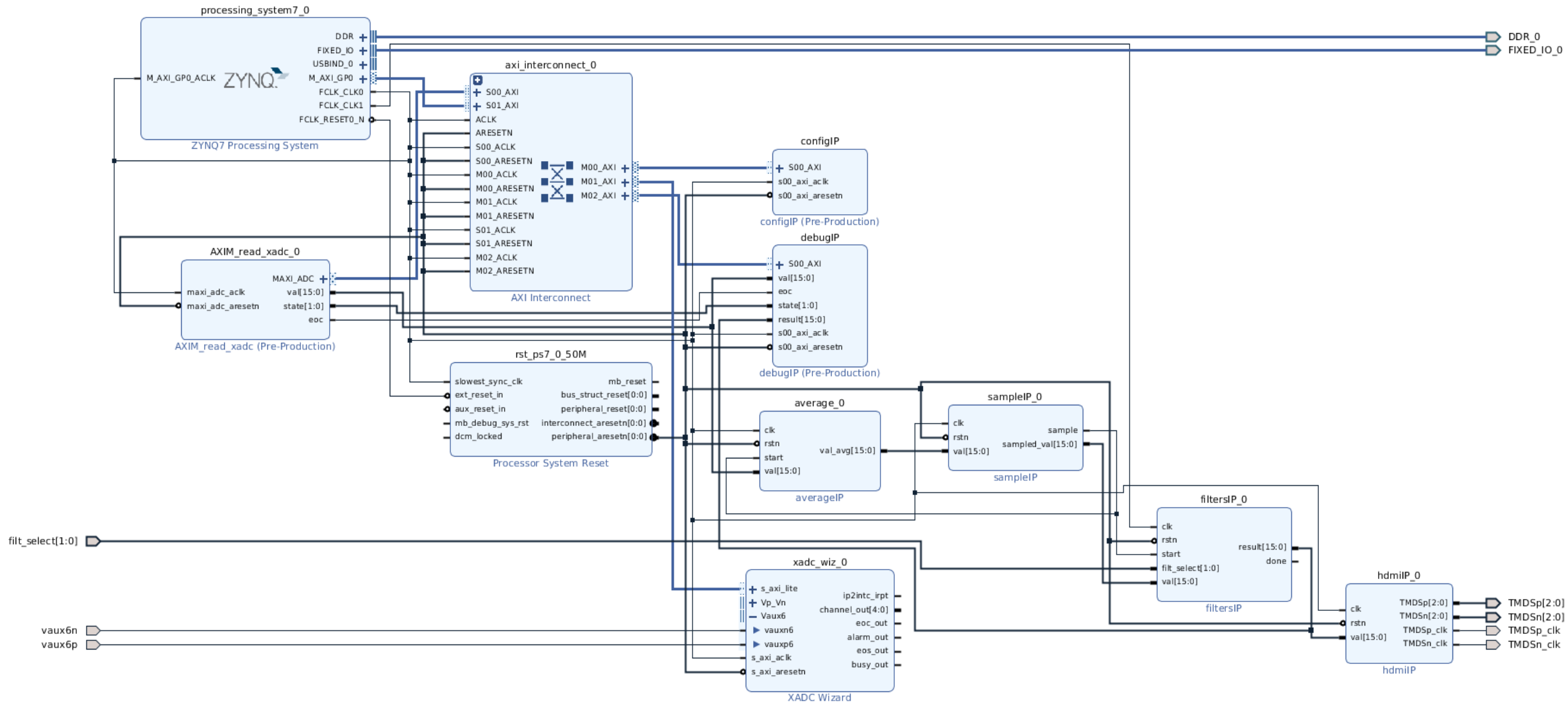
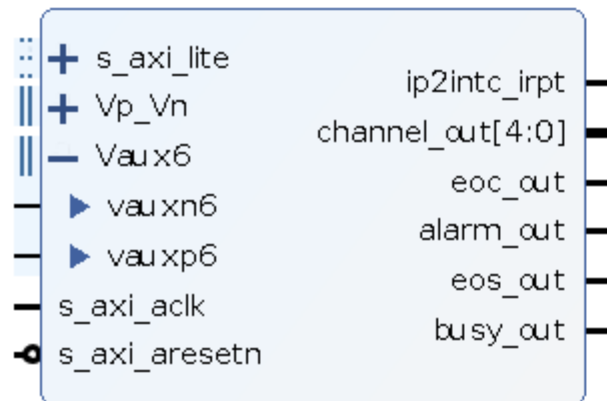


Figura 8 – Block Design implementado no Vivado.

XADC IP

- Comunica através de AXI-Lite;
- *Single Channel – Vaux6*;
- Taxa de amostragem: 1 MS/s.



Interface Options

☒ AXI4Lite ☐ DRP ☐ None

Startup Channel Selection

☐ Simultaneous Selection
☐ Independent ADC
☒ Single Channel
☐ Channel Sequencer

AXI4STREAM Options

Timing Mode

☒ Continuous Mode ☐ Event Mode

DRP Timing Options

☒ Enable DCLK

DCLK Frequency(MHz)

ADC Conversion Rate(KSPS)

Acquisition Time (CLK)

Clock divider value = 4

ADC Clock Frequency(MHz) = 25.00

Actual Conversion Rate(KSPS) = 961.54

Figura 9 – Configuração do XADC no Vivado.

Filters IP

- IP gerado pelo Vivado HLS;
- 4 DSPs para realizar operações aritméticas.

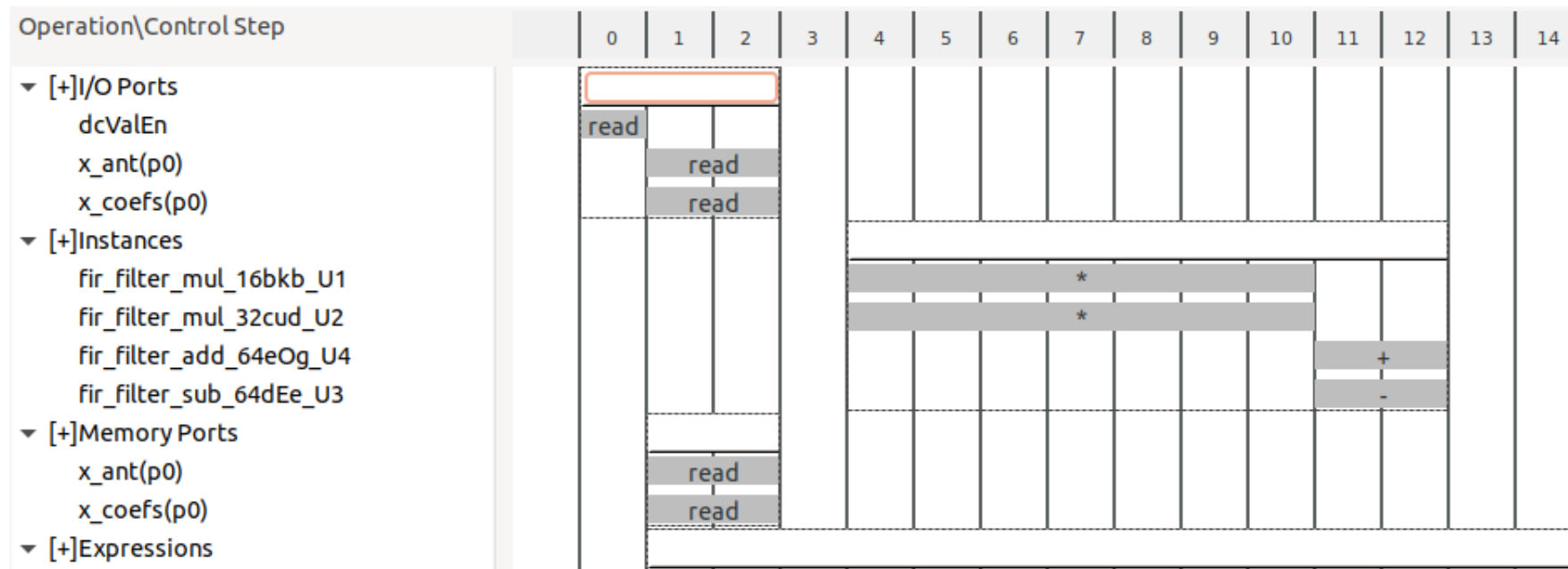
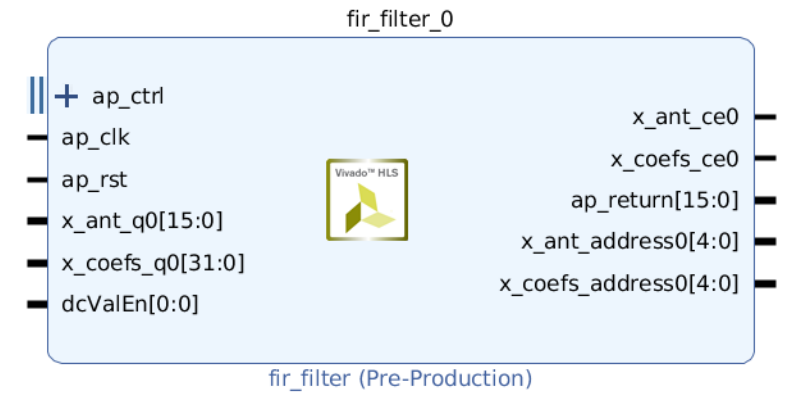


Figura 10 – Perspetiva de análise no Vivado HLS.

Resultados

01 XADC IP

02 Filters IP

03 HDMI IP

XADC IP

- AXI master (AXIM_read_xadc) que faz a leitura do slave XADC;
- AXI slave (my_slave) teste, com 2 registos:
 - endereço 44A00000h: simula o registo de configuração;
 - endereço 44A00004h: simula o registo do XADC ;

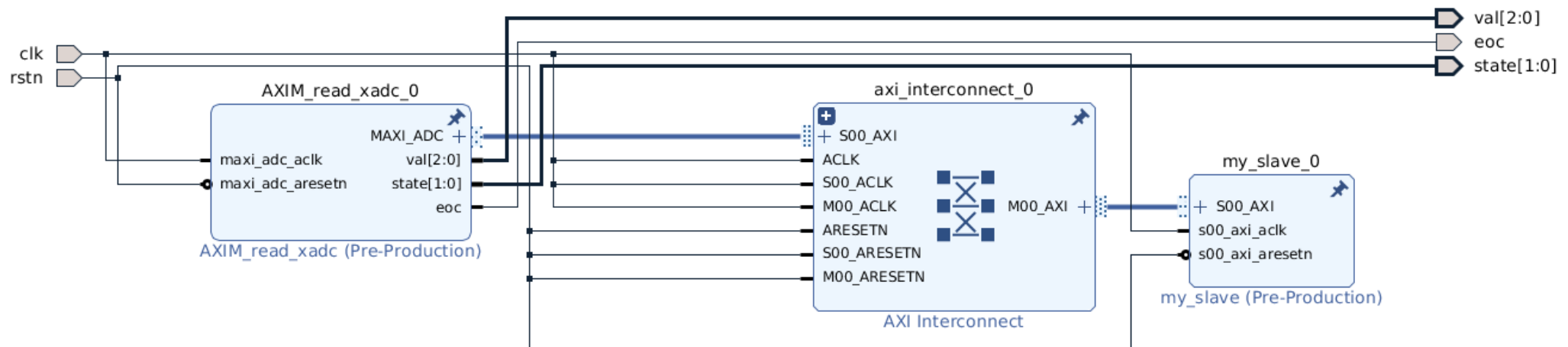


Figura 11 – Block Design do teste ao AXI master de leitura do XADC.

XADC IP

- *Master* realiza uma leitura ao registo de configuração;
- Com o registo de configuração a 1, são realizadas leituras consecutivas ao XADC;
- A cada leitura, o sinal de EOC é ativo e o valor lido é colocado na saída.



Figura 12 – Resultado do *testbench* do AXI master de leitura do XADC.

Filters IP

Filters IP – Resultados Esperados

- *Golden vectors* para validação dos filtros.

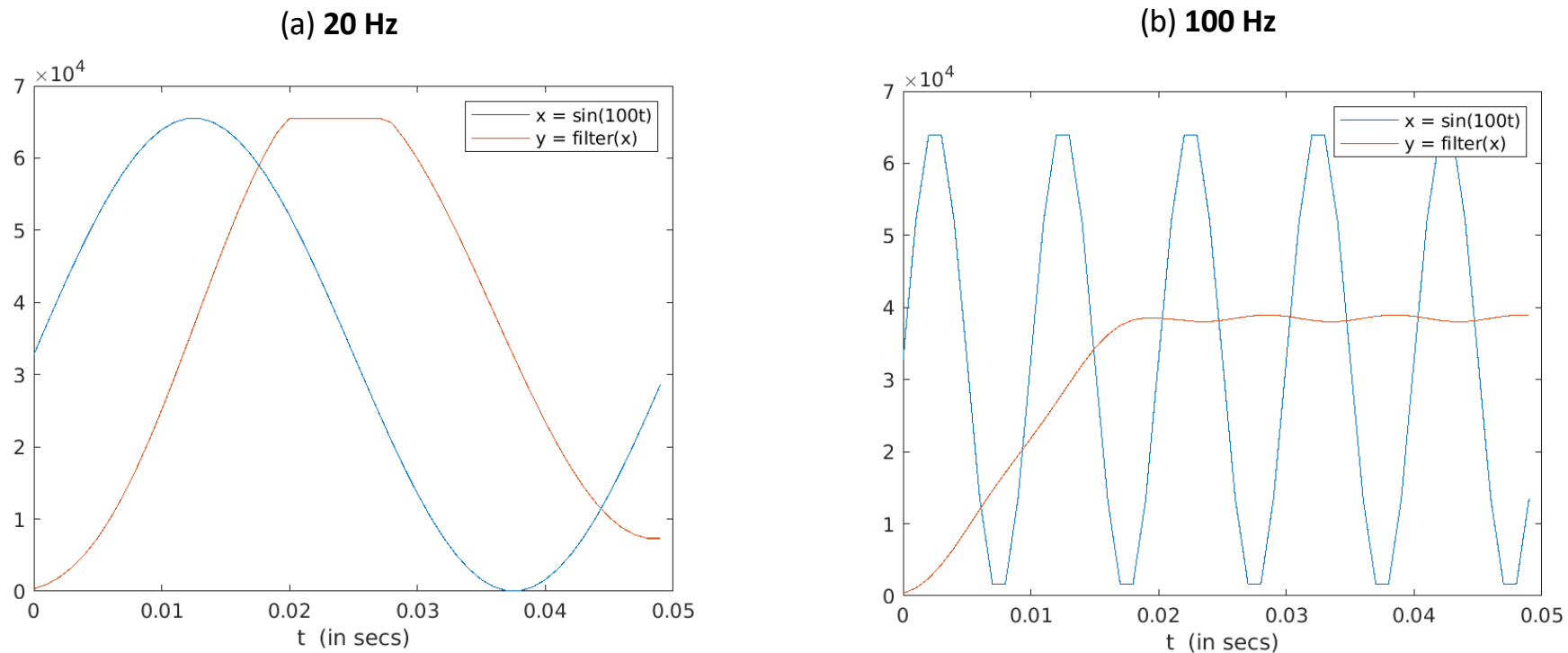


Figura 13 – Saída do LPF em MATLAB, a vermelho, em função da entrada a azul.

Filters IP -

Resultados em Simulação

- Aplicação de LPF (*filt_select* = 0) a um sinal de entrada de 20 Hz.

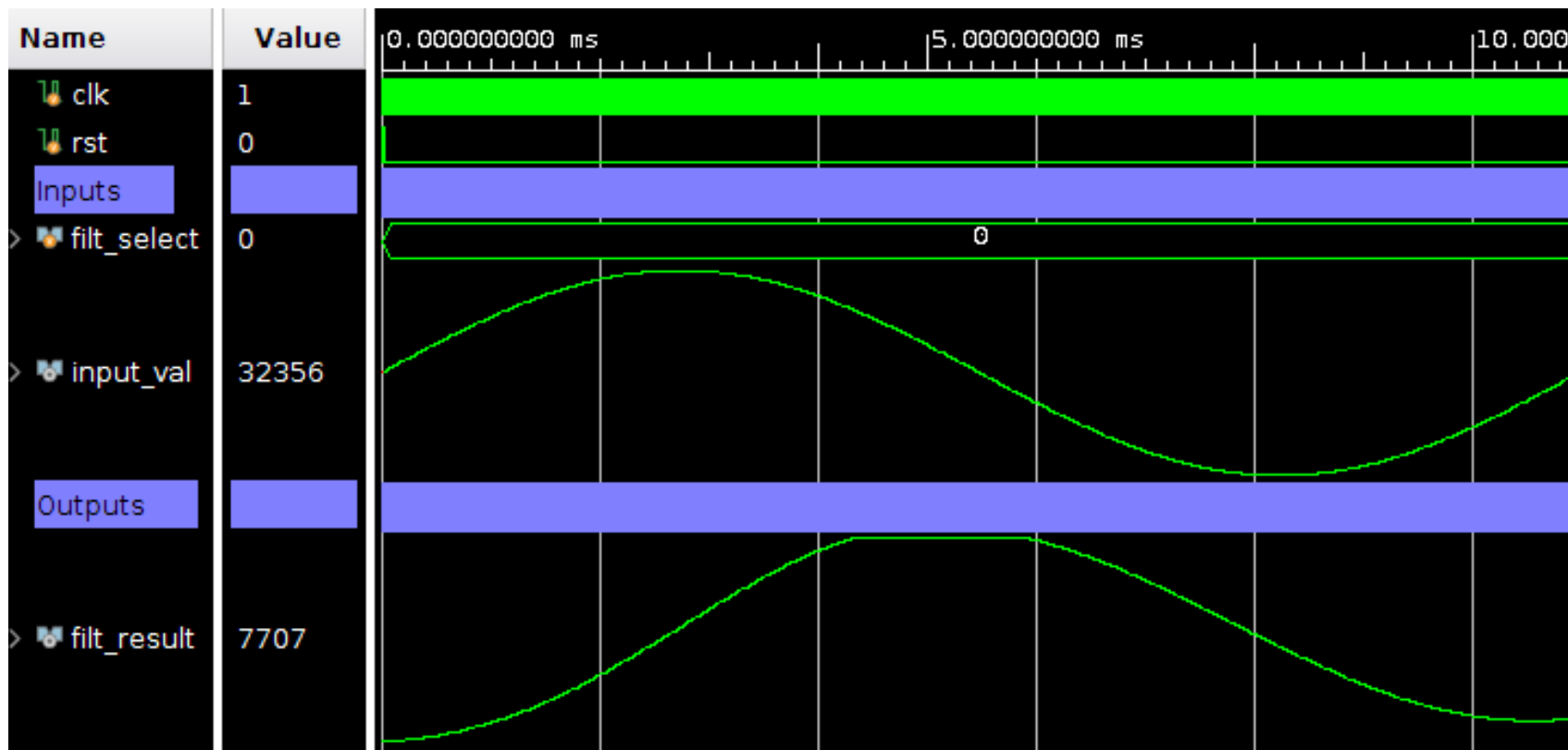


Figura 14 - Simulação comportamental do bloco de filtros aplicando um LPF a uma onda de entrada de 20 Hz.

Filters IP - Resultados em Simulação

- Aplicação de LPF (*filt_select* = 0) a um sinal de entrada de 100 Hz.

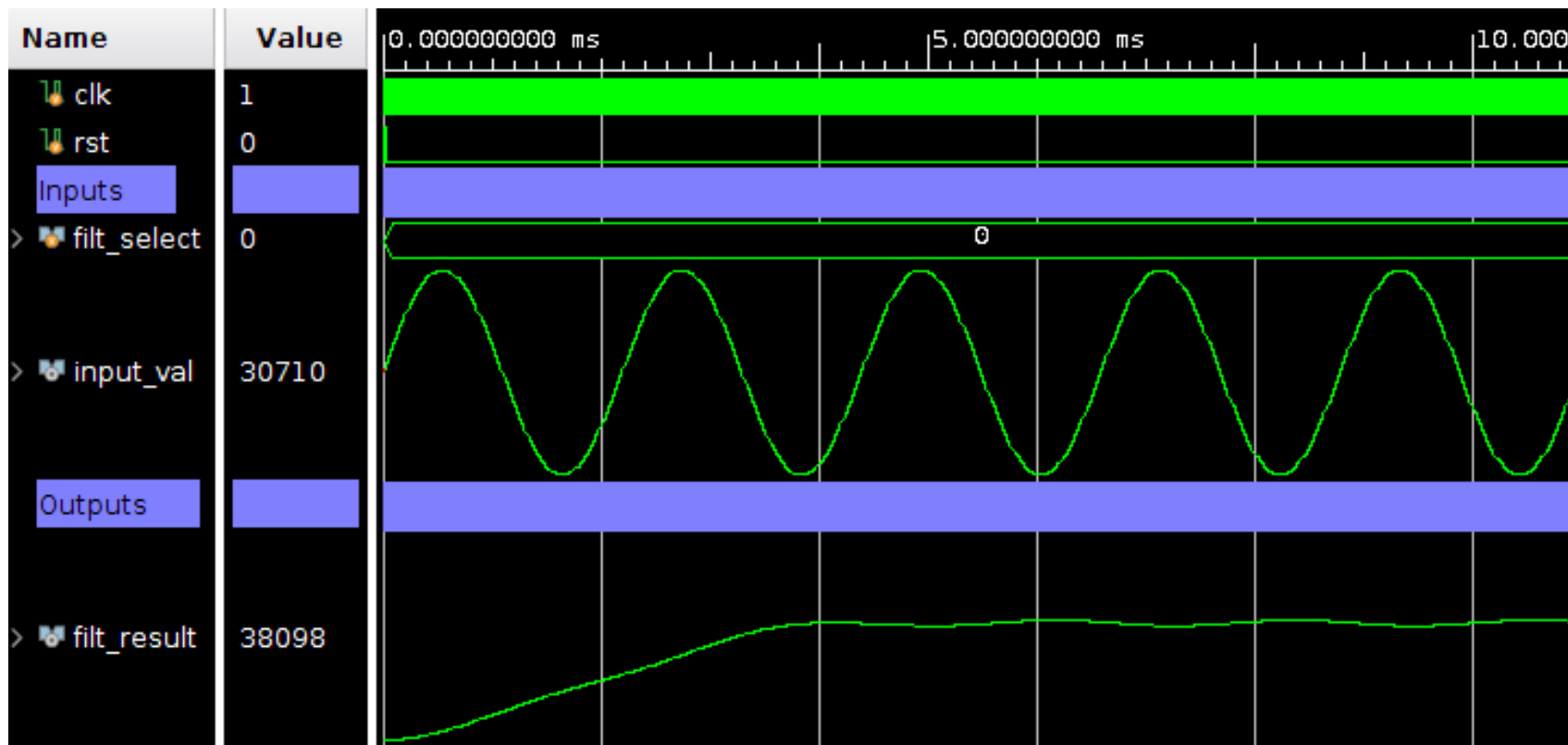
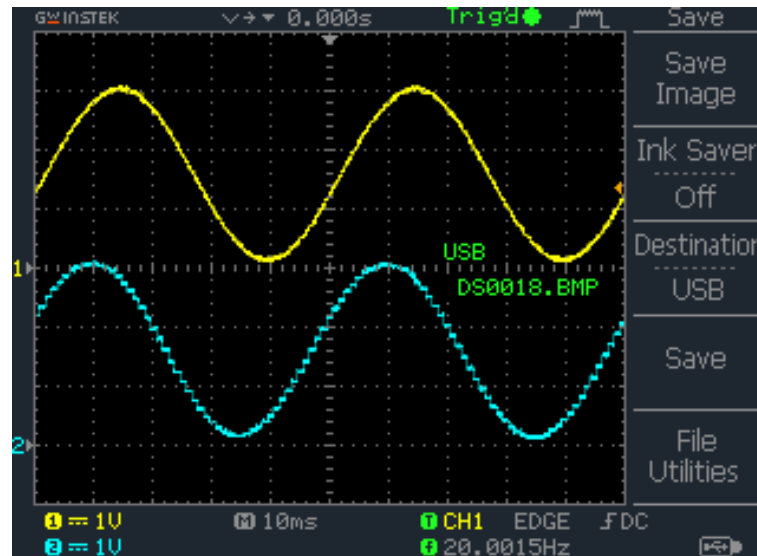


Figura 15 - Simulação comportamental do bloco de filtros aplicando um LPF a uma onda de entrada de 100 Hz.

Filters IP - Resultados Experimentais

- Resultados na STM32;
- Utilização do DAC para visualização externa do sinal filtrado.

(a) 20 Hz



(b) 100 Hz

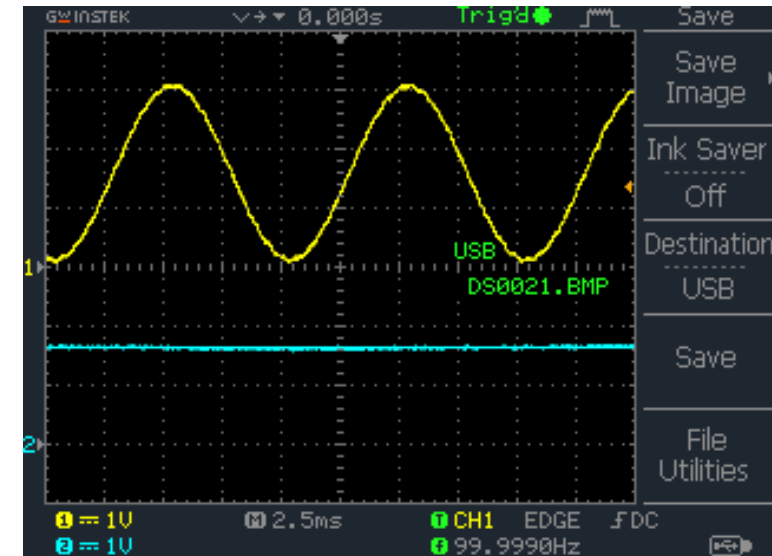


Figura 16 - Saída do LPF na STM32, a azul, em função da entrada, a amarelo.

Filters IP - Resultados Experimentais

- Resultados na Zybo;
- Utilização do *debugIP* para disponibilizar na PS os valores obtidos pelo filtro.

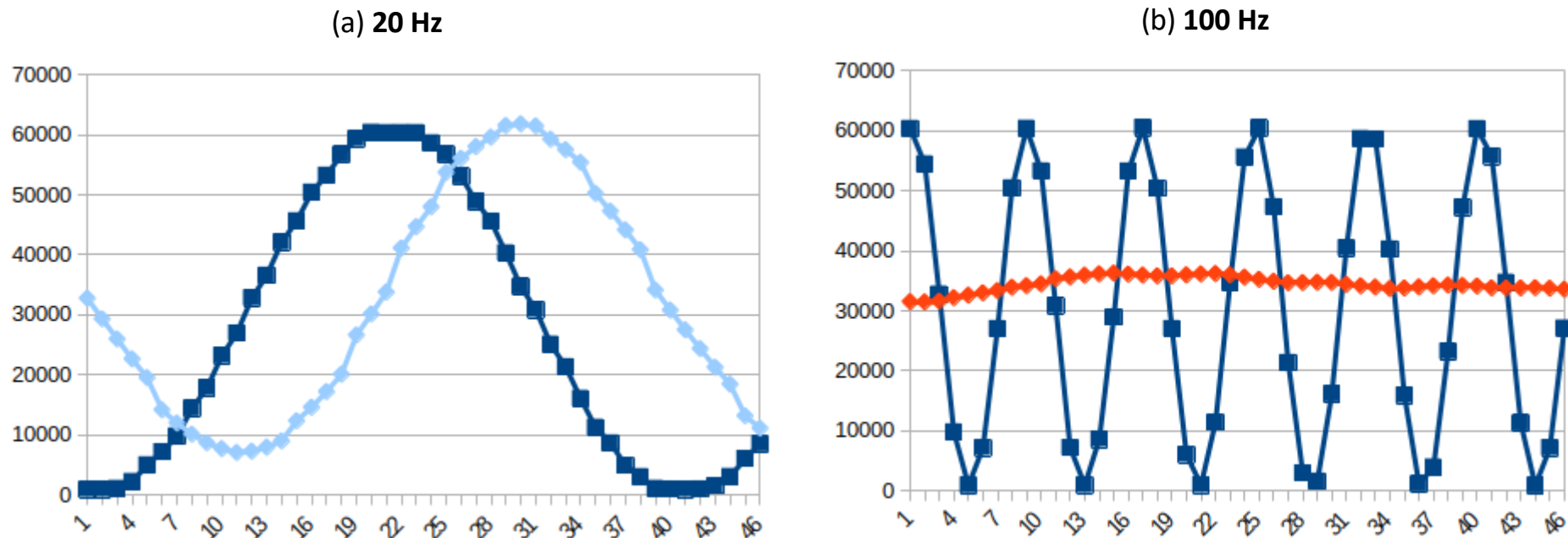


Figura 17 - Bloco de filtros com o LPF selecionado para uma entrada, a azul escuro, de:

a) 20 Hz - Saída a azul claro; b) 100 Hz - Saída a vermelho..

HDMI IP

HDMI IP - Write FSM

- Simula-se uma *frame* 4x4 e *input* constante = 65535;
- S_CLEAN: WD = 0;
- S_WRITE: WD = 1, nos endereços: 0,1,2,3;

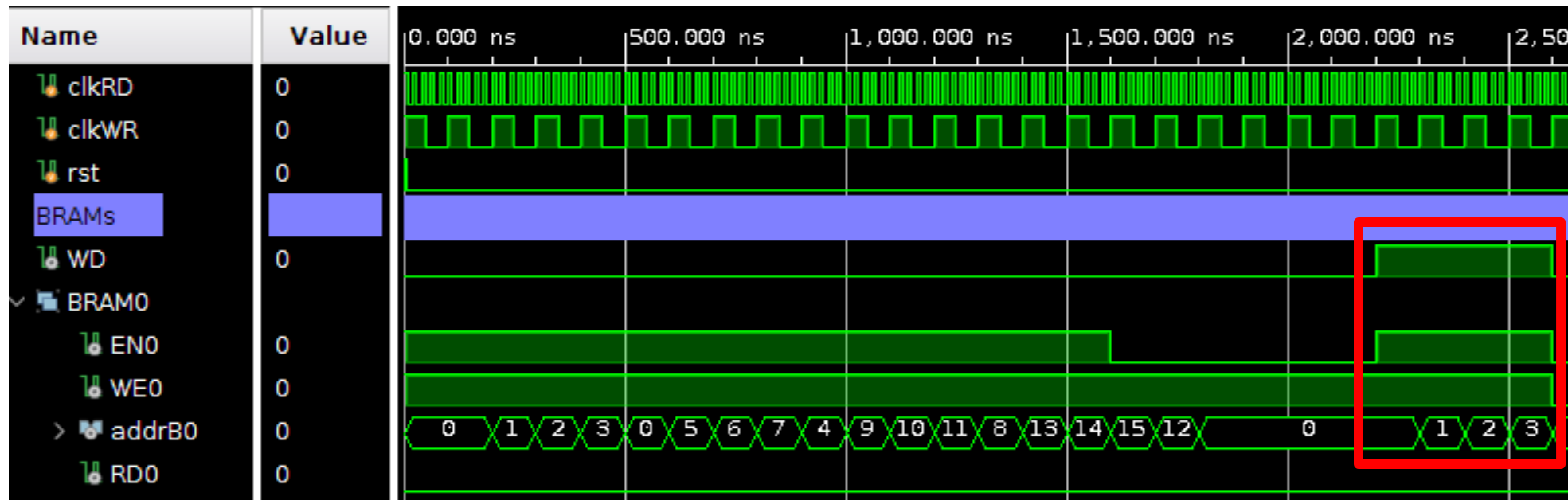


Figura 18 - Simulação comportamental do bloco HDMI - Escrita da frame.

HDMI IP – Read FSM

- Pixel a *0x00ff00* (Verde) nos endereços: 0,1,2,3

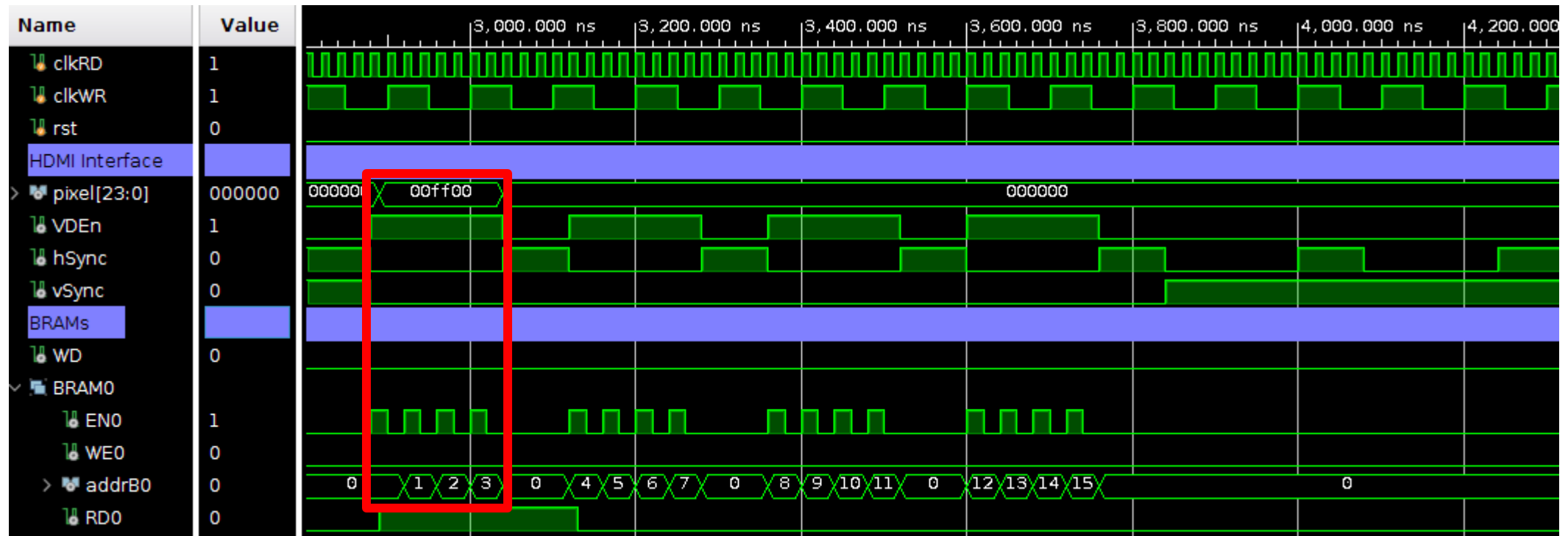


Figura 19 - Simulação comportamental do bloco HDMI - Leitura da frame.

HDMI IP – Main FSM

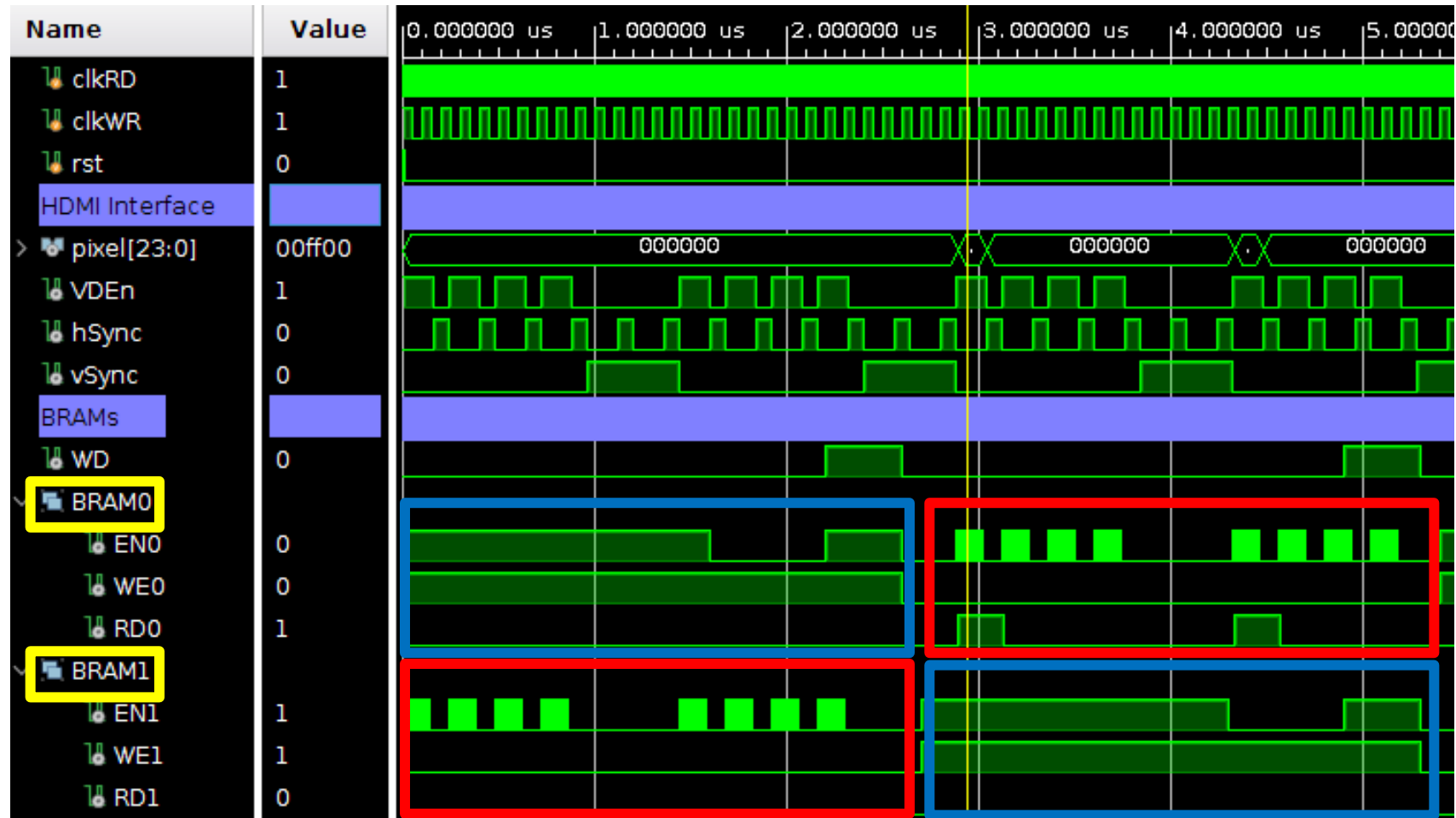


Figura 20 - Simulação comportamental do bloco HDMI.

Resultados Finais

Resultados Finais

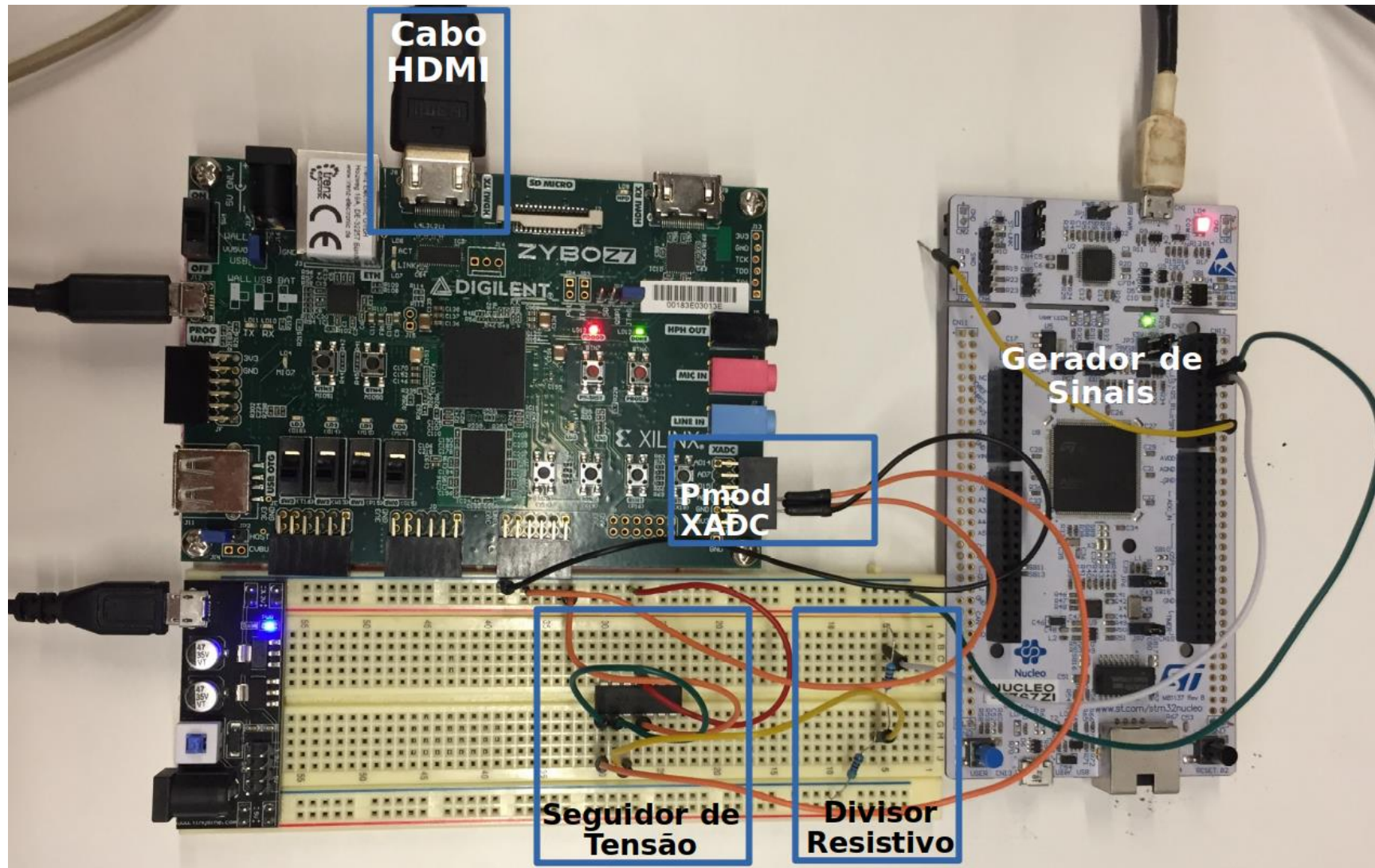


Figura 21 – Montagem realizada para os testes finais.

Resultados Finais

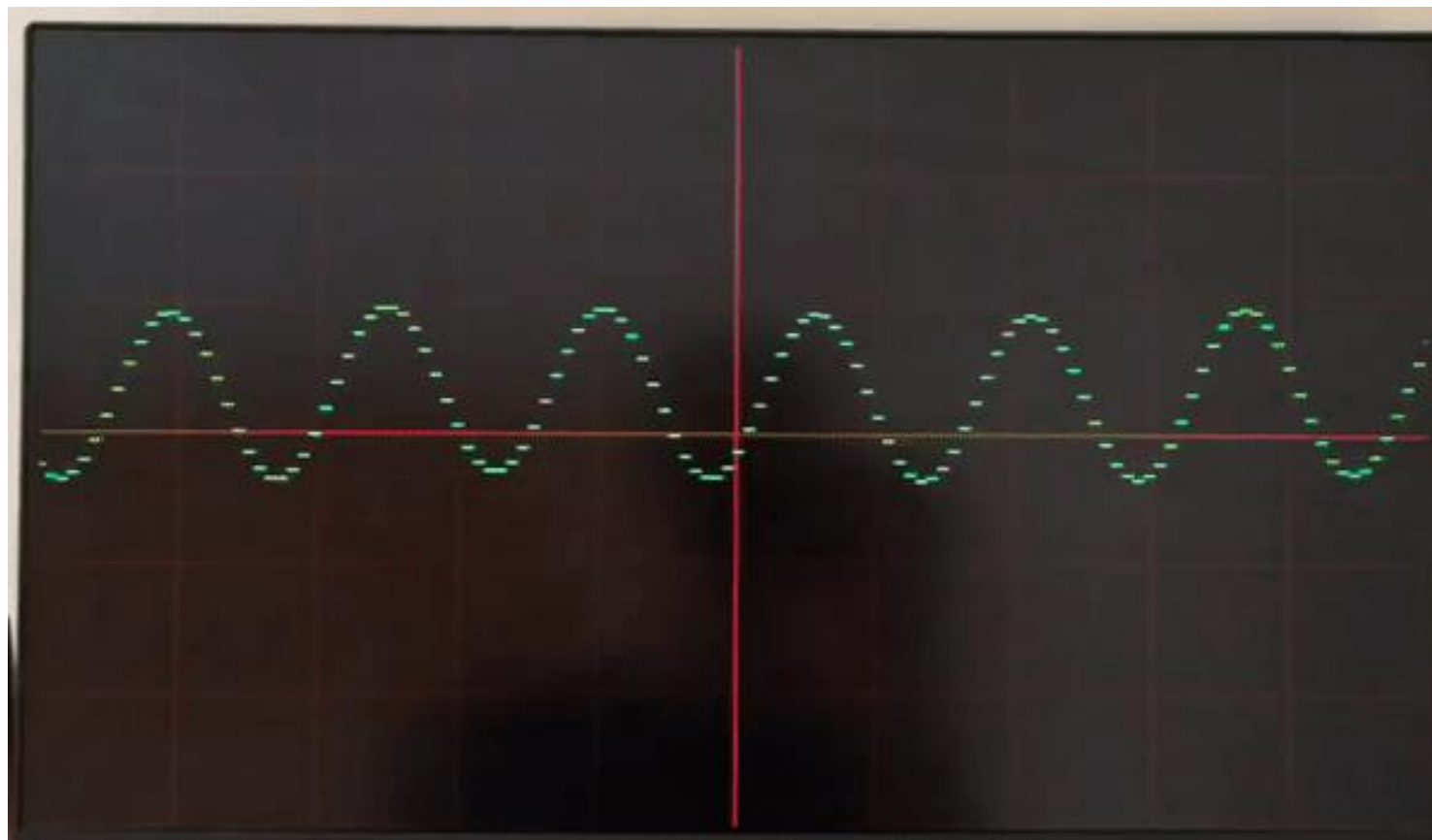
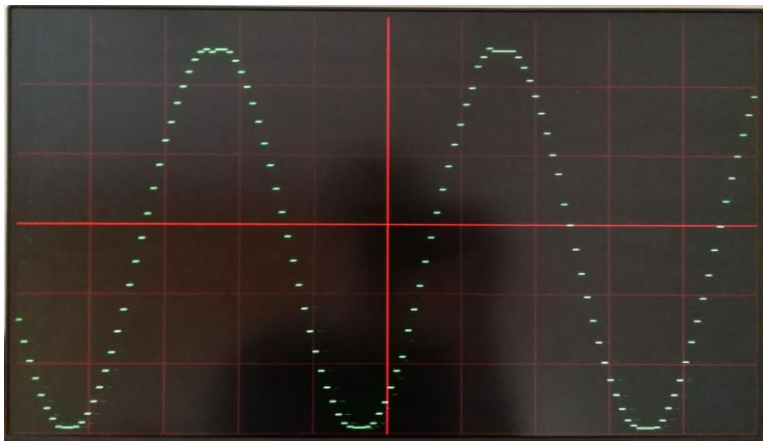


Figura 22 – Funcionamento do sistema com o LPF selecionado a uma frequência de entrada de 50 Hz e 20 Hz de entrada.

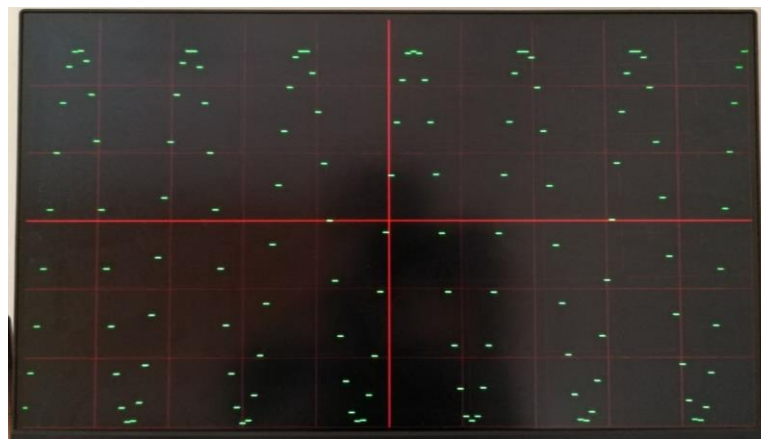
Resultados Finais

Sem filtro aplicado

(a) 20 Hz



(b) 50 Hz



(c) 100 Hz

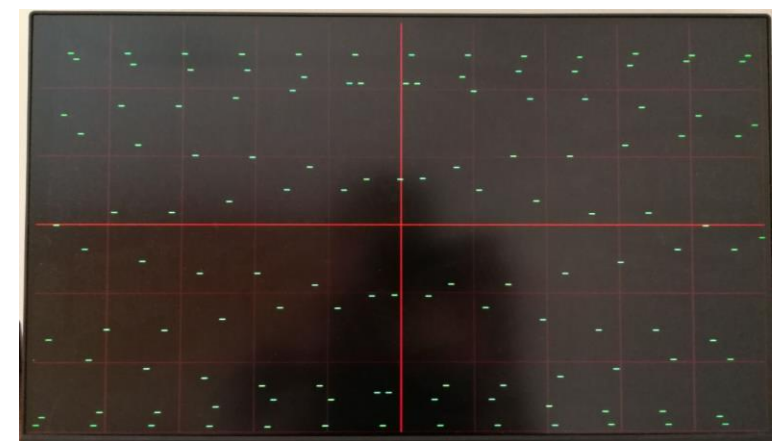


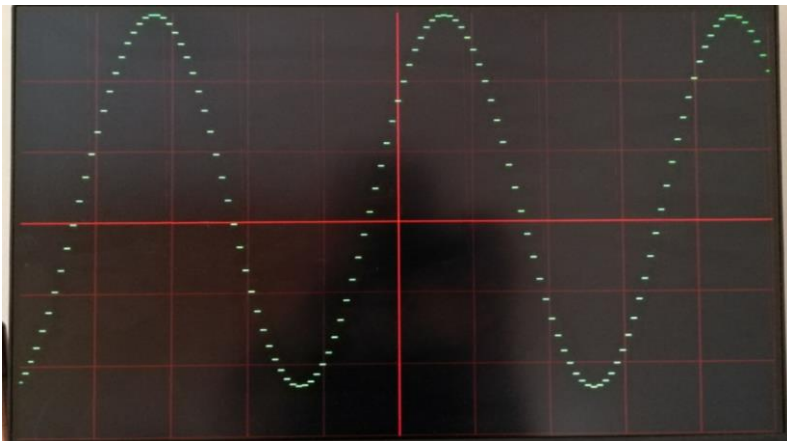
Figura 22 – Visualização do sinal de entrada a diferentes frequências.

Resultados Finais

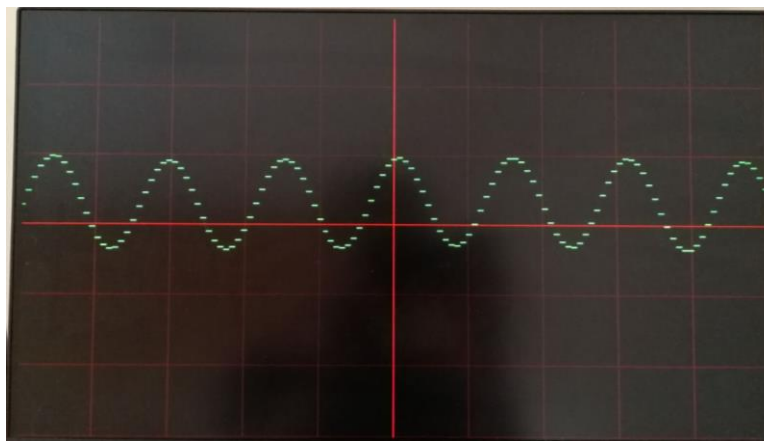
Filtro passa-baixo selecionado

- Frequência de corte do filtro 50 Hz;

(a) 20 Hz



(b) 50 Hz



(c) 100 Hz



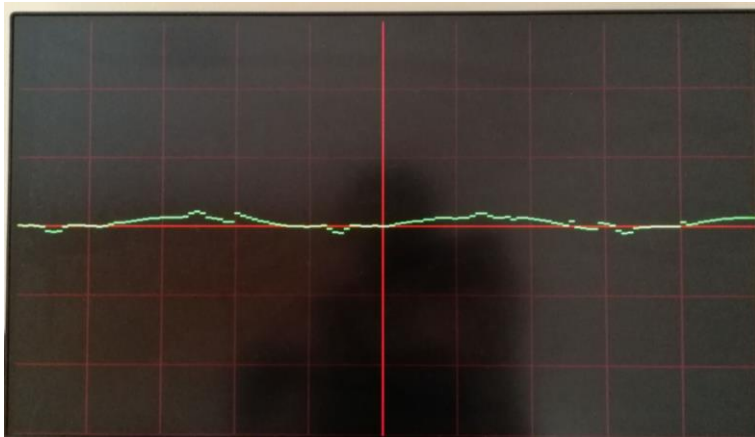
Figura 23 – Visualização da saída do LPF para vários sinais de entrada com frequências diferentes.

Resultados Finais

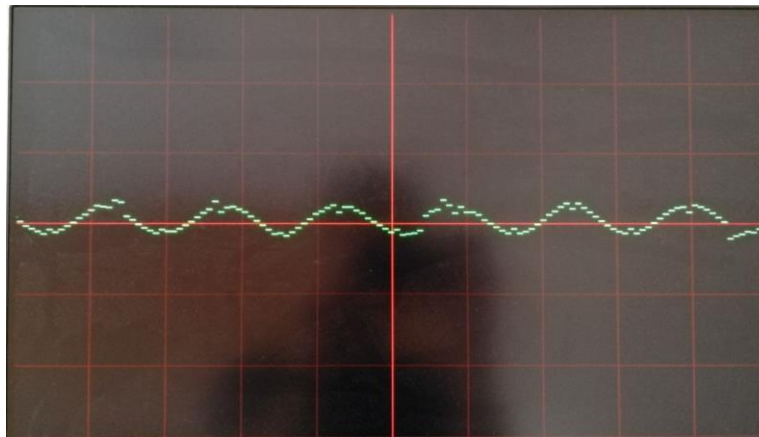
Filtro passa-alto selecionado

- Frequência de corte do filtro 50 Hz;

(a) 20 Hz



(b) 50 Hz



(c) 100 Hz

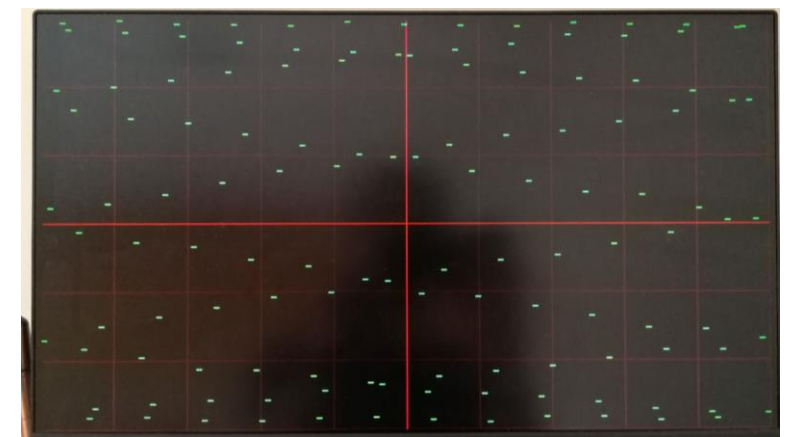


Figura 24 – Visualização da saída do HPF para vários sinais de entrada com frequências diferentes.

Resultados Finais

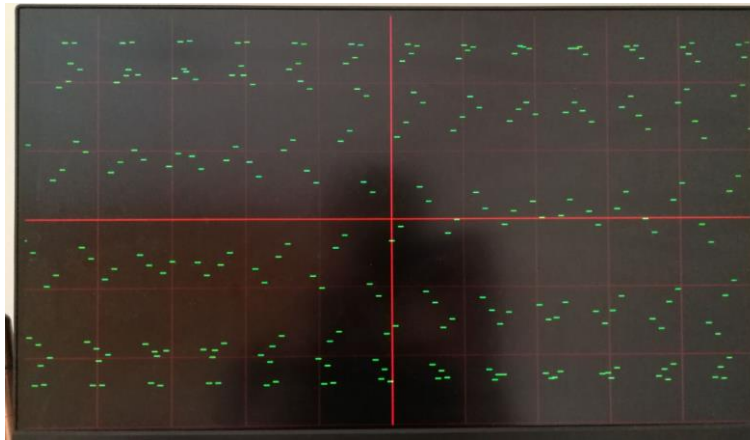
Filtro passa-banda selecionado

- Frequência de corte do filtro 60 Hz e 200 Hz;

(a) 50 Hz



(b) 120 Hz



(c) 220 Hz

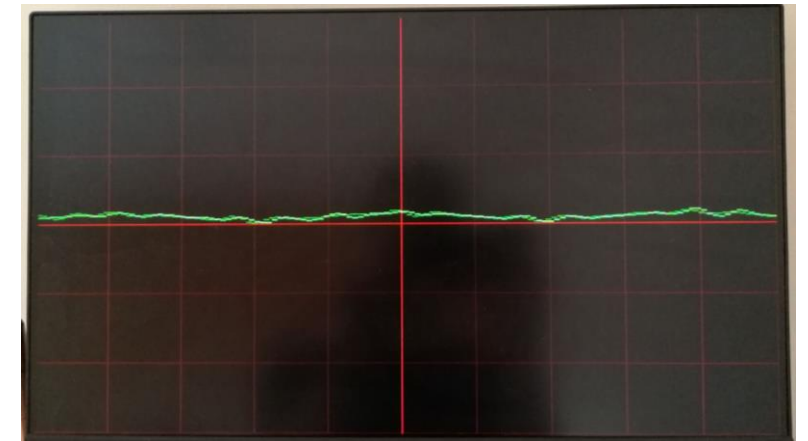


Figura 25 – Visualização da saída do BPF para vários sinais de entrada com frequências diferentes.

Conclusões

01 Conclusão

02 Trabalho Futuro

Conclusão

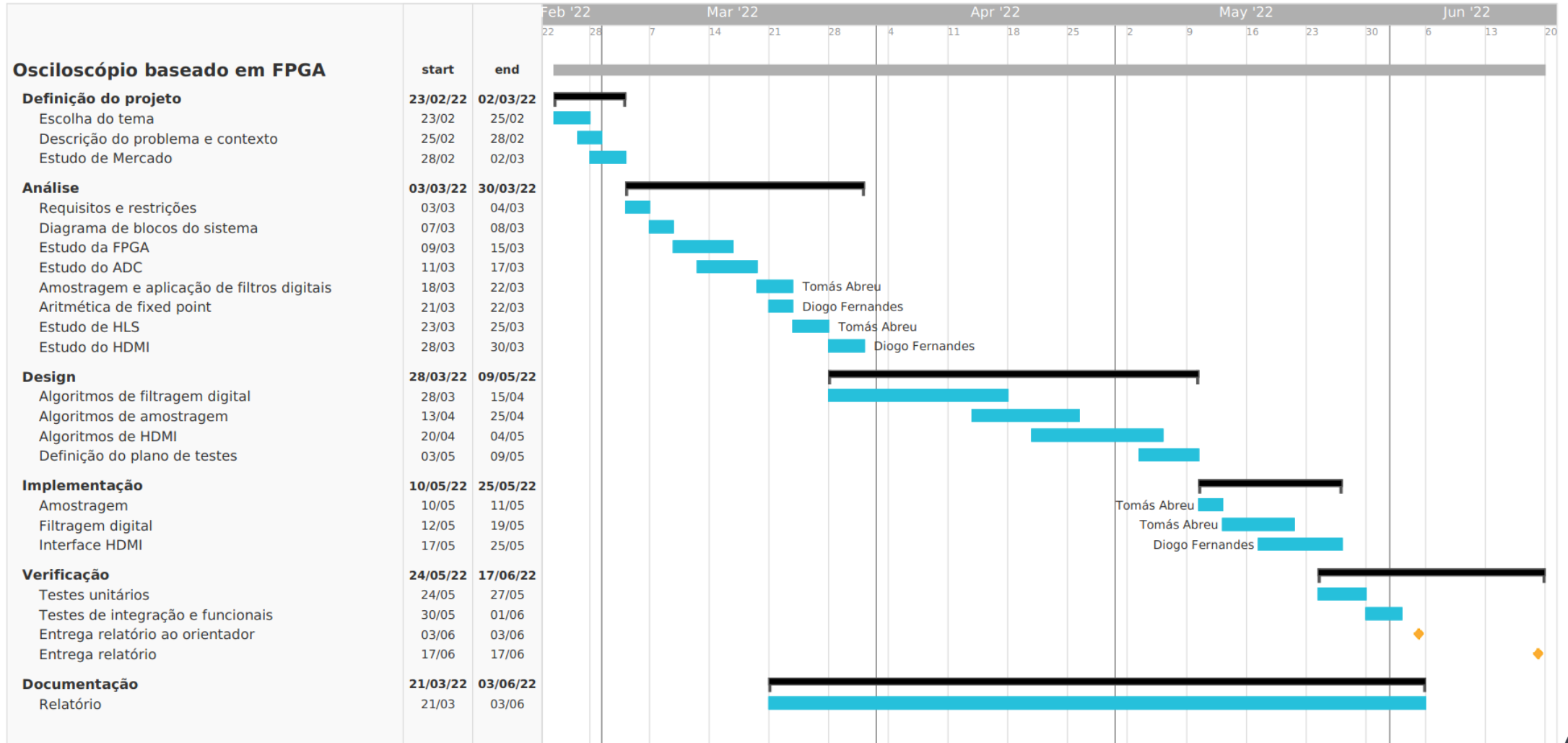
- **Osciloscópio com funcionalidades básicas:**
 - Amostragem de sinal analógico;
 - Filtros digitais implementados em HLS (técnicas DSP);
 - Controlador de vídeo que transmite o sinal filtrado por HDMI.
- **Competências adquiridas:**
 - Amostragem de sinal analógico;
 - Introdução e especialização ao desenho em HW utilizando HDL e HLS;
 - Aumento de experiência em Matlab, C e Verilog;
 - Utilização de *testbenches*.

Trabalho Futuro

- Comparação do desenvolvimento de *hardware* recorrendo a HLS com o desenvolvimento utilizando *Hardware Description Languages* (HDL): [4]
 - Qualidade do *hardware* gerado;
 - Tempo de desenvolvimento;
- Aumento da frequência de amostragem;
- Controlo da escala temporal e escala de tensão;
- Função de *trigger*;
- Utilização de mais canais de entrada ADC, representando as suas ondas na interface gráfica;
- Utilização de um ecrã tátil.

Calendário de tarefas

Diagrama de Gantt



OBRI GADO !

QUESTÕES?