

Tytuł: Oscyloskop

Autorzy: Jakub Zając (JZ),

Paweł Mozgowiec (PM)

Ostatnia modyfikacja: 31.08.2023

Spis treści

1. Repozytorium git	1
2. Wstęp	1
3. Specyfikacja	1
3.1. Opis ogólny algorytmu	1
3.2. Tabela zdarzeń	2
4. Architektura	2
4.1. Moduł: top	2
4.1.1. Schemat blokowy	2
4.1.2. Porty	3
a) mou – mouse_ctl, input	3
b) vga – vga_ctl, output	3
4.1.3. Interfejsy	3
a) m2c – mouse_ctl to core	3
4.2. Rozprowadzenie sygnału zegara	3
5. Implementacja	4
5.1. Lista zignorowanych ostrzeżeń Vivado.	4
5.2. Wykorzystanie zasobów	4
5.3. Marginesy czasowe	4
6. Film.	4

1. Repozytorium git

Adres repozytorium GITa (jeżeli używane):

<https://github.com/Zajcu14/oscilloscope.git>

W przypadku repozytorium prywatnego należy zaprosić użytkownika zewnętrznego o adresie mailowym: kaczmarczyk@agh.edu.pl

2. Wstęp

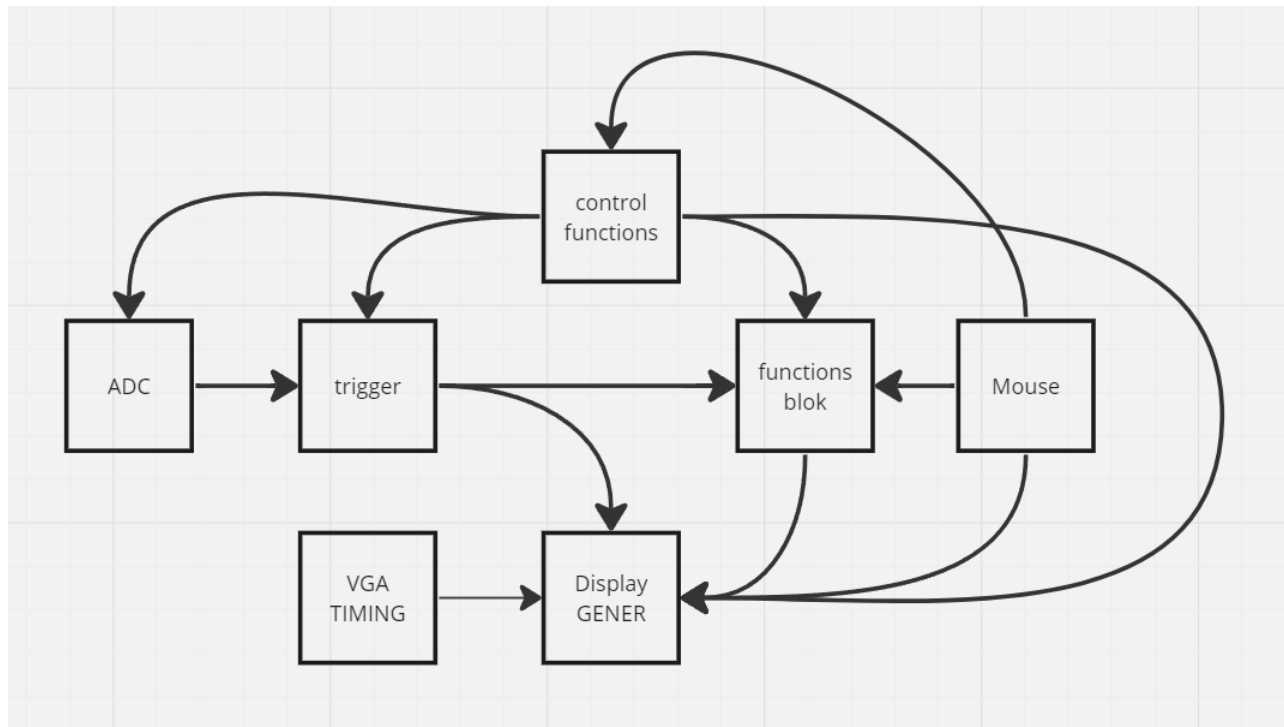
Pomysł wziął się z naszego zainteresowania dziedziną elektroniki cyfrowej i przetwarzania sygnałów. Chcieliśmy również nabrać doświadczenia przy pracy z zewnętrznymi urządzeniami akwizycji sygnału jak i późniejszej ich obróbki. Celem projektu było stworzenie cyfrowego oscyloskopu z opcjami filtrowania i analizy widmowej.

3. Specyfikacja

3.1. Opis ogólny algorytmu

Uproszczony schemat blokowy działania implementowanego algorytmu. Co się dzieje po starcie, jak wygląda przebieg działania, kiedy i pod jakimi warunkami się kończy.

Ewentualnie przykładowe screen-shoty tego, co w przybliżeniu chcielibyśmy uzyskać.



3.2. Tabela zdarzeń

Opis zdarzeń występujących podczas działania programu/urządzenia, zarówno zewnętrznych (interakcje z użytkownikiem), jak i wewnętrznych (specyficzne stany w algorytmie). Zdarzenia podzielone są na kategorie dotyczące różnych stanów działania programu. Kategorie powinny odpowiadać stanom ze schematu z pkt. 2.1.

Zdarzenie	Kategoria	Reakcja systemu
podłączenie sygnały wejściowego	ADC	przetworzenie sygnału na cyfrowy - próbka
sygnał sinusoidalny	trigger	uruchomienie wyzwolenia - zapis próbek do pamięci buffora
zbocze narastające clk_65Mhz	trigger -> functions blok	wpisane próbek i późniejsza interpretacja przez moduły wewnętrzne
zbocze narastające clk_65Mhz	trigger -> Display_gener	wpisane próbek i późniejsze narysowanie wykresu
zbocze narastające clk_65Mhz	functions blok-> Display_gener	wpisane próbek i późniejsze narysowanie wykresu oraz wartości liczbowych (parametrów)
zbocze narastające clk_65Mhz	control fun.=> function_blok	konfiguracja bloku i jego parametrów
ruch myszką	mouse	zmiana koordynatów myszki
zbocze narastające clk_65Mhz	mouse => function_blok	zmiana wartości wewnętrznych uwzględniających położenie oraz wciśnięte klawisze myszki

zbocze narastające clk_65Mhz	control fun.=>trigger	zmiana szybkości próbkowania oraz wartości trigger_level
zbocze narastające clk_65Mhz	control fun.=>ADC	zmiana szybkości próbkowania
zbocze narastające clk_65Mhz	mouse fun.=>Display_gen	rysowanie myszki na ekranie
zbocze narastające clk_65Mhz	control fun.=>Display_gen	rysowanie myszki na ekranie
zbocze narastające clk_65Mhz	Vga_timing =>Display_gen	wygenerowanie sygnałów kalibrujących
kliknięcie myszki w pierwszej ćwrtce ekranu	Display_gen	kontrola próbkowania trigger, przycisk lewy odjęcie wartości, prawy dodanie wartości, poprzez przytrzymanie środkowego przycisku zwiększamy szybkość zmiany
kliknięcie myszki w drugiej ćwrtce ekranu	Display_gen	kontrola trigger+lvl, przycisk lewy odjęcie wartości, prawy dodanie wartości, poprzez przytrzymanie środkowego przycisku zwiększamy szybkość zmiany
ruch myszką po wykresie i przytrzymanie lewego przycisku	mouse =>Display_gen	poruszanie wykresem
przyciśnięcie RST na płytkę (BTNC)	płytką	reset całego programu przywrócenie ustawień nominalnych

4. Architektura

Uwaga: dobrze zrobiony projekt zawiera tylko moduły strukturalne (zbudowane z innych modułów) i funkcjonalne (zawierające bloki proceduralne always @). Staramy się nie generować bloków mieszających te dwa typy, o ile to możliwe.

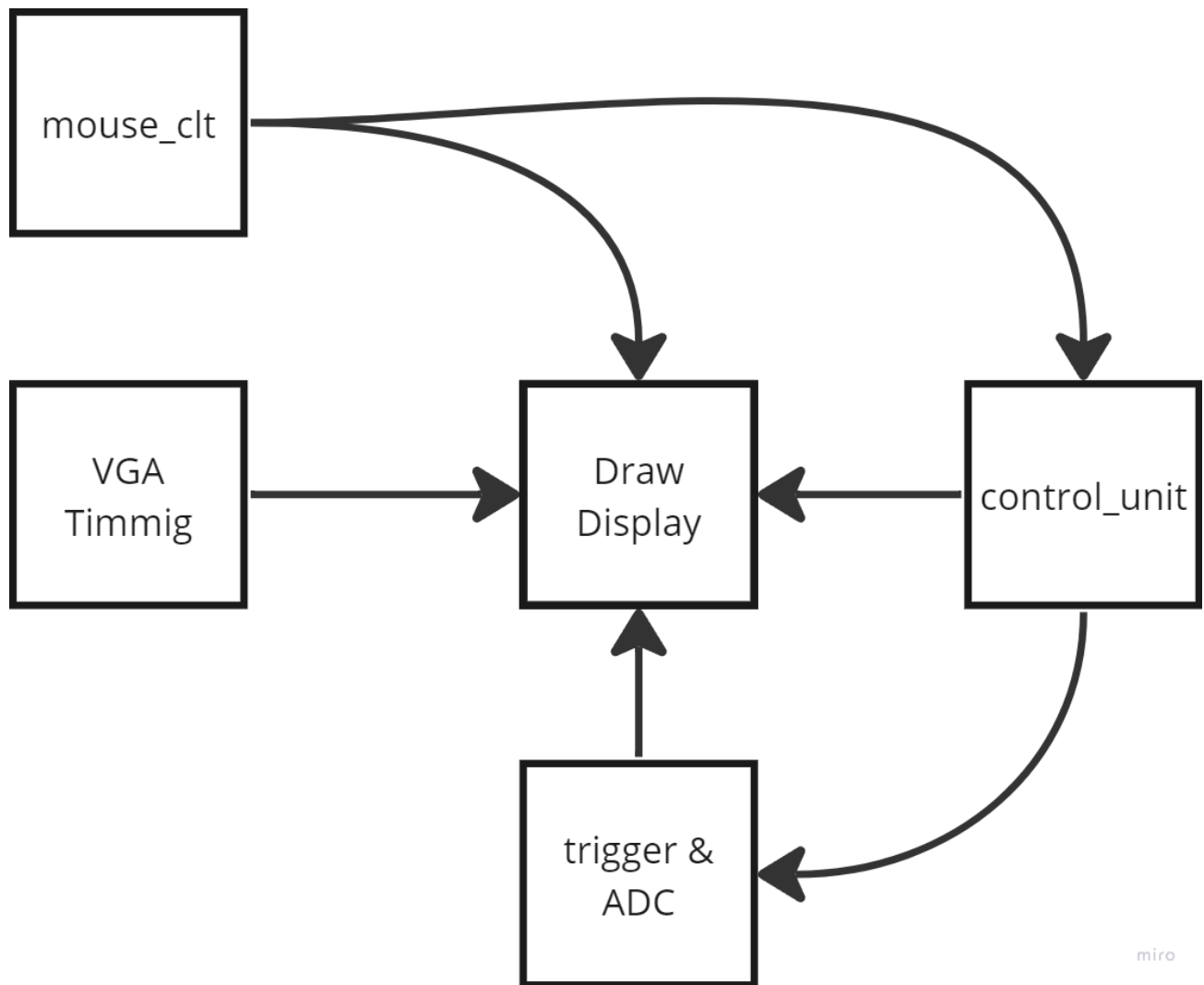
Uwaga: opisujemy architekturę **tylko głównego modułu oraz rozprowadzenie sygnału zegara**.

4.1. Moduł: top

Osoba odpowiedzialna: JZ oraz PM

4.1.1. Schemat blokowy

Przykładowy schemat blokowy modułu głównego



Uwaga:

- interfejsy dwukierunkowe rozbijamy na 2 interfejsy jednokierunkowe
- nazwa interfejsu stanowi prefiks nazwy sygnałów składowych
- w interfejsach nie uwzględniamy sygnałów globalnych (np. clk i rst).

4.1.2. Porty

a) ps2– MouseCtl, input

nazwa portu	opis
ps2_data	szeregowe wejście danych
ps2_clk	taktowanie komunikacji z myszką

b) vga – vga_ctl, output

nazwa portu	opis
vs	sygnał synchronizacji pionowej VGA
hs	sygnał synchronizacji poziomej VGA

4.1.3. Interfejsy

a) vga_if – vga_ctl to core

nazwa sygnału	opis
---------------	------

[10:0] vcount	wertykalna pozycja wyświetlanego piksela na ekranie
[10:0] hcount	horyzontalna pozycja wyświetlanego piksela na ekranie
[11:0] rgb	wartość koloru w kodzie rgb
vsync	sygnał synchronizacji wertykalny
hsync	sygnał synchronizacji horyzontalny
vblnk	sygnał wygaszania wertykalny
hblnk	sygnał wygaszania horyzontalny

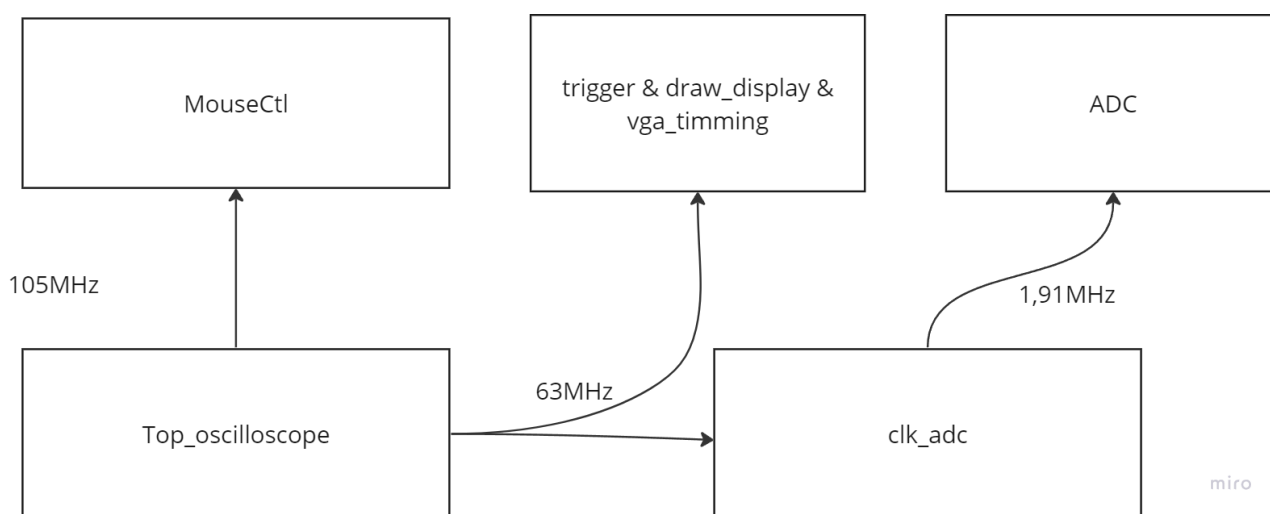
4.2. Rozprowadzenie sygnału zegara

Osoba odpowiedzialna:

PMInformacja na temat źródła sygnału zegarowego, używanych częstotliwości zegara w całym układzie.

Moduł generatora zegara umieszczamy w module głównym projektu. W pozostałych modułach używamy tylko i wyłącznie sygnały zegara wygenerowane przez ten moduł.

Uwaga: jeżeli używamy różnych częstotliwości zegara w układzie, to należy je tak dobrać, aby były wielokrotnościami siebie (umożliwia to wygenerowanie tych sygnałów z jednego IP core generatora zegara i zapobiega problemom z synchronizacją).



5. Implementacja

5.1. Lista zignorowanych ostrzeżeń Vivado.

Identyfikator ostrzeżenia	Liczba wystąpień	Uzasadnienie
[Synth 8-7080]	1	Parallel synthesis criteria is not met
RPBF #1	1	port JB[1] jest to zewnętrzna komunikacja z układem ADC
HPDR # 1	1	port JB[1] jest to zewnętrzna komunikacja z układem ADC

TIMING #1-3	3	PS2CLK ,PS2Data,btnC są komunikacją hardware
ULMTCS	1	ostrzeżenie o wykorzystaniu danych, problem polega na tym że buffer trigger posiada wielkość [11:0] data [0:511] dla uzyskania lepszej jakości obrazu, można ostrzeżenie naprawić poprzez zmniejszenie do [11:0] data [0:255]

5.2. Wykorzystanie zasobów

Tabela z wykorzystaniem zasobów z Vivado

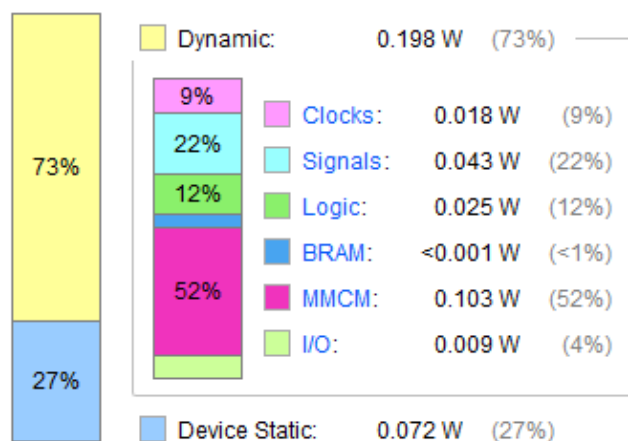
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP
✓ synth_1	constrs_1	synth_design Complete!												7502	13557	0.5	0	0
✓ impl_1	constrs_1	write_bitstream Complete!	0.343	0.000	0.064	0.000		0.000	0.270	0	5 Warn			7209	13564	0.5	0	0

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 0.27 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 26,4°C
 Thermal Margin: 58,6°C (11,7 W)
 Effective θ_{JA} : 5,0°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power



5.3. Marginesy czasowe

Marginesy czasowe (WNS) dla setup i hold.

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0,343 ns	Worst Hold Slack (WHS): 0,064 ns	Worst Pulse Width Slack (WPWS): 2,762 ns
Total Negative Slack (TNS): 0,000 ns	Total Hold Slack (THS): 0,000 ns	Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 26498	Total Number of Endpoints: 26498	Total Number of Endpoints: 13578

All user specified timing constraints are met.

6. Film.

Link do ściągnięcia filmu:

<https://>