

Tytuł: Szachy

Autorzy: Dawid Mironiuk, Michał Malara

Ostatnia modyfikacja: 2.09.2024

Spis treści

1.	REPOZYTORIUM GIT.....	1
2.	WSTĘP	1
3.	SPECYFIKACJA	2
3.1.	OPIS OGÓLNY ALGORYTMU	2
3.2.	TABELA ZDARZEŃ	4
4.	ARCHITEKTURA	5
4.1.	MODUŁ: TOP	5
4.1.1.	Schemat blokowy.....	5
4.1.2.	Porty	5
4.1.3.	Interfejsy	5
4.2.	ROZPROWADZENIE SYGNAŁU ZEGARA.....	8
5.	IMPLEMENTACJA.....	8
5.1.	LISTA ZIGNOROWANYCH OSTRZEŻEŃ VIVADO.	8
5.2.	WYKORZYSTANIE ZASOBÓW	9
5.3.	MARGINESY CZASOWE.....	10
6.	FILM.	10

1. Repozytorium git

<https://github.com/Dudixu/projekcik>

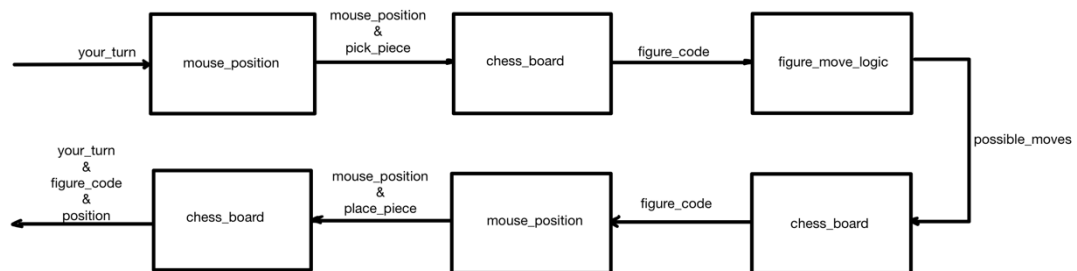
2. Wstęp

W ramach projektu została stworzona gra w szachy dla dwóch osób, każda z osób do gry używa osobnej płytki BASYS3. Szachy to najpopularniejsza gra planszowa na świecie, w samej tylko Polsce szacuje się że zasady zna od 4 do 6 milionów Polaków. W grze zostały zaimplementowane wszystkie podstawowe zasady gry w szachy takie jak promocja piona, roszada czy sprawdzenie poprawności ruchu poszczególnych figur i zakomunikowanie na jakie pola może ruszyć się wybrana przez gracza figura.

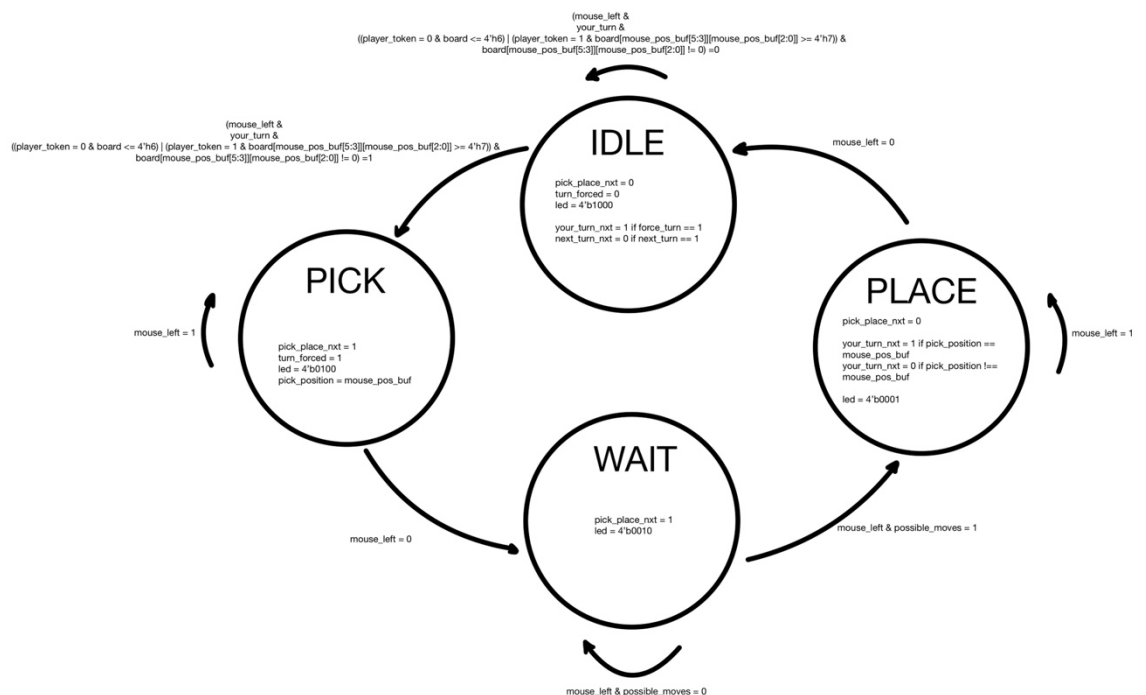
3. Specyfikacja

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.

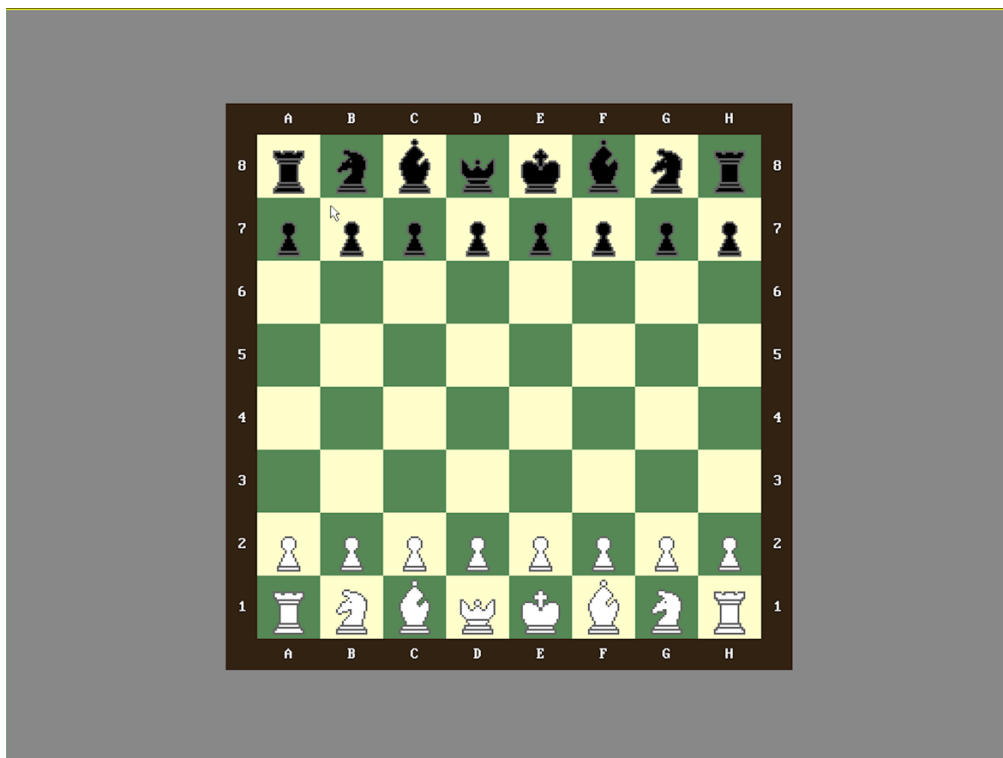


Schemat 1. Uproszczony schemat blokowy działania implementowanego algorytmu



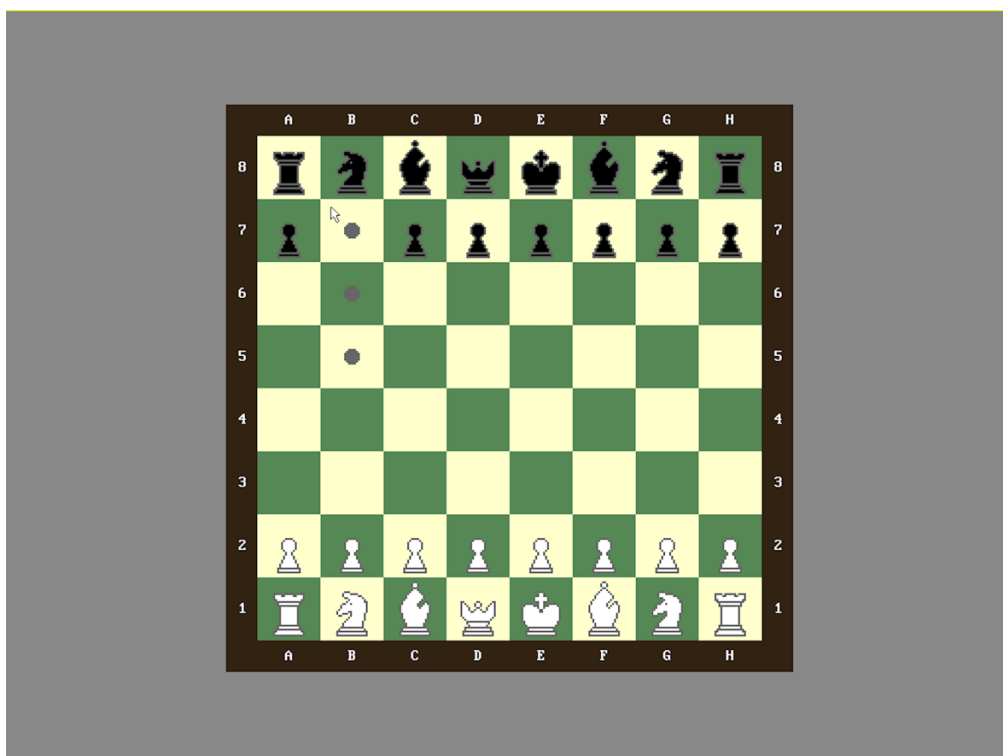
Schemat 2. Maszyna stanu przedstawiająca działanie programu.

- Po uruchomieniu płytek z wgranym bitstreamem szachów na ekranie pojawia się plansza z ustawionymi figurami szachowymi. Gra rozpocznie się gdy któryś z graczy ruszy się białą figurą i wcześniejszym przełączeniu switcha na start.



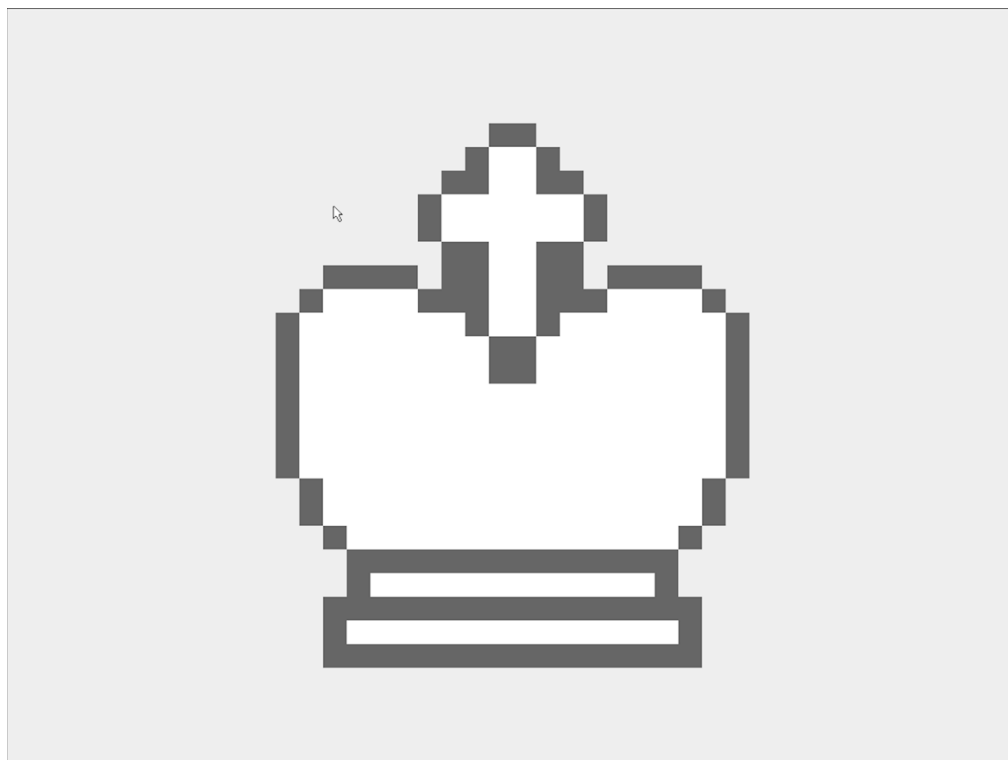
Obraz 1. Wygląd planszy podczas startu.

- Po rozpoczęciu gry gracze wykonują ruchy na zmianę, wysyłając między sobą informacje o wykonanym ruchu.
- Po wybraniu przez gracza figury którą chce się ruszyć moduł `figure_move_logic` oblicza możliwość ruchu na podstawie obecnego rozstawienia figur na planszy, które jest zapisane w module `chess_board` i wyświetla na ekranie szare koła na polach na które gracz może się ruszyć wybraną figurą.



Obraz 2. Wygląd pól na których możemy postawić figure.

- Jeśli gracz odstawi figurę na pole z którego ją podniósł, może ponownie wykonać ruch inną figurą. Natomiast jeśli zostanie wykonany ruch na jedno z oznaczonych pól kołami kod figury zostaje wpisany na wybrana pozycje w chess_board oraz współrzędne i kod figury wraz z sygnałem your_turn zostają wysłane do przeciwnika.
- Gra kończy się w momencie zbitia jednego z króli na planszy a na ekranie wyświetla się menu końca gry na którym wyświetla się król wygranego gracza.



Obraz 3. Wygląd menu końcowego.

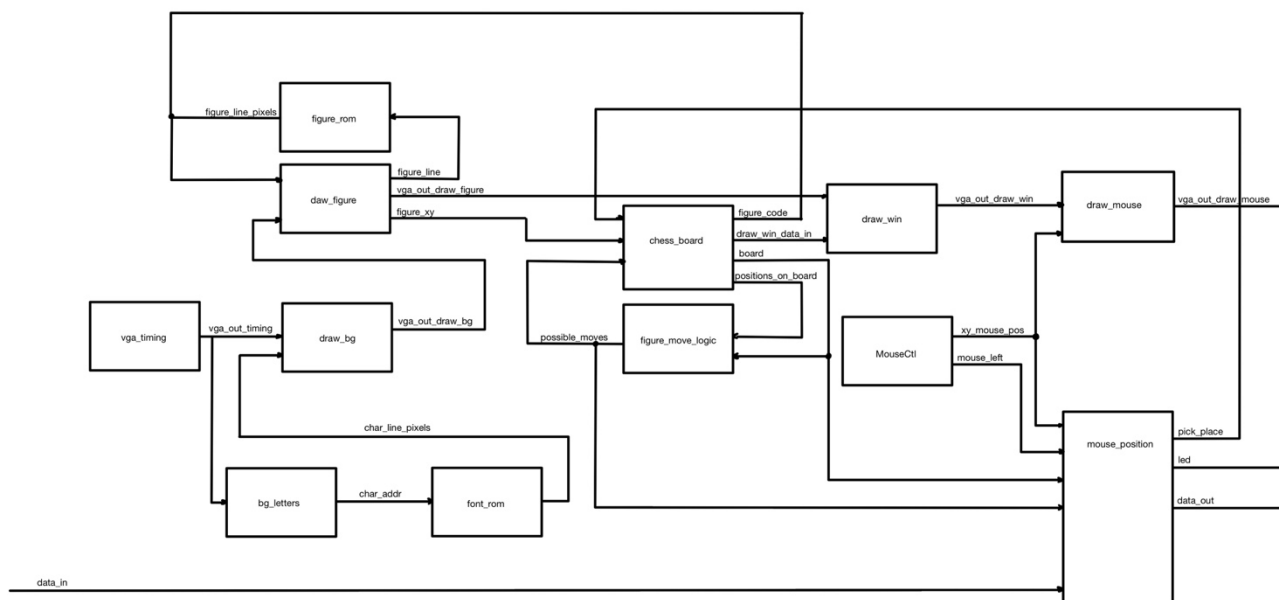
3.2. Tabela zdarzeń

Zdarzenie	Kategoria	Reakcja systemu
LPM w obszarze planszy, na polu z białą figurą, przy sygnale twój ruch	PICK	Uruchomienie gry, podniesienie figury
LPM na jedno z pól oznaczonych jako możliwość ruchu	PLACE	Położenie figury
LPM na pole z wrogą figurą	PLACE	Zbitie figury przeciwnika
LPM na pole z wrogim królem	PLACE	Przejsście do menu końca gry (draw_win)
Your_turn = 0	IDLE	Oczekiwanie na ruch przeciwnika.
Your_turn = 1	IDLE	Nasz ruch.

4. Architektura

4.1. Moduł: top

4.1.1. Schemat blokowy



Schemat 3. Uproszczony schemat blokowy modułu top.

4.1.2. Porty

a) *mou – mouse_ctl, input*

nazwa portu	opis
mou_si	Szeregowe wejście danych

b) *vga – vga_ctl, output*

nazwa portu	opis
vga_vs	sygnał synchronizacji pionowej VGA
vga_hs	sygnał synchronizacji poziomej VGA
vga_red	Sygnał odpowiadający za poziom czerwieni (red) dla każdego piksela na ekranie.
vga_blue	Sygnał odpowiadający za poziom niebieskiego (blue) dla każdego piksela na ekranie.
vga_green	Sygnał odpowiadający za poziom zieleni (green) dla każdego piksela na ekranie.

4.1.3. Interfejsy

a) *Xy_mouse_pos – mouse_ctl to draw_mouse & mouse_position*

nazwa sygnału	opis
xpos[9:0]	Horyzontalna pozycja kursora myszy na ekranie

ypos[9:0]	Wertykalna pozycja kursora myszy na ekranie
-----------	---

b) vga_out_timing – vga_timing to draw_bg & bg_letters

nazwa sygnału	opis
vsync	Sygnał synchronizacji pionowej (vertical sync) informujący monitor o zakończeniu jednej klatki i rozpoczęciu następnej.
vcount [10:0]	Licznik pionowy (vertical count) określający bieżącą linię poziomą, na której odbywa się wyświetlanie lub blanking w pionie.
vblnk	Sygnał blankingu pionowego (vertical blanking) wskazujący na czas, gdy nie ma wyświetlania aktywnego obrazu, a elektronika monitora wraca do górnej części ekranu.
hcount [10:0]	Licznik poziomy (horizontal count) określający bieżącą pozycję pikseli w linii poziomej wyświetlania lub blankingu.
hsync	Sygnał synchronizacji poziomej (horizontal sync) informujący monitor o zakończeniu jednej linii skanowania i rozpoczęciu następnej.
hblnk	Sygnał blankingu poziomego (horizontal blanking) oznaczający czas, gdy wyświetlacz nie wyświetla aktywnego obrazu podczas przejścia do następnej linii poziomej.
rgb [11:0]	Sygnał kolorów RGB (red, green, blue) określający kolor każdego piksela wyświetlanego na ekranie, przy czym każdy kolor ma swoją 4-bitową składową.

c) vga_out_draw_bg – draw_bg to draw_figure

nazwa sygnału	opis
vsync	Sygnał synchronizacji pionowej (vertical sync) informujący monitor o zakończeniu jednej klatki i rozpoczęciu następnej.
vcount [10:0]	Licznik pionowy (vertical count) określający bieżącą linię poziomą, na której odbywa się wyświetlanie lub blanking w pionie.
vblnk	Sygnał blankingu pionowego (vertical blanking) wskazujący na czas, gdy nie ma wyświetlania aktywnego obrazu, a elektronika monitora wraca do górnej części ekranu.
hcount [10:0]	Licznik poziomy (horizontal count) określający bieżącą pozycję pikseli w linii poziomej wyświetlania lub blankingu.
hsync	Sygnał synchronizacji poziomej (horizontal sync) informujący monitor o zakończeniu jednej linii skanowania i rozpoczęciu następnej.
hblnk	Sygnał blankingu poziomego (horizontal blanking) oznaczający czas, gdy wyświetlacz nie wyświetla aktywnego obrazu podczas przejścia do następnej linii poziomej.
rgb [11:0]	Sygnał kolorów RGB (red, green, blue) określający kolor każdego piksela wyświetlanego na ekranie, przy czym każdy kolor ma swoją 4-bitową składową.

d) vga_out_draw_figure – draw_figure to draw_win

nazwa sygnału	opis
vsync	Sygnał synchronizacji pionowej (vertical sync) informujący monitor o zakończeniu jednej klatki i rozpoczęciu następnej.
vcount [10:0]	Licznik pionowy (vertical count) określający bieżącą linię poziomą, na której odbywa się wyświetlanie lub blanking w pionie.
vblnk	Sygnał blankingu pionowego (vertical blanking) wskazujący na czas, gdy nie ma wyświetlania aktywnego obrazu, a elektronika monitora wraca do górnej części ekranu.
hcount [10:0]	Licznik poziomy (horizontal count) określający bieżącą pozycję pikseli w linii poziomej wyświetlania lub blankingu.
hsync	Sygnał synchronizacji poziomej (horizontal sync) informujący monitor o zakończeniu jednej linii skanowania i rozpoczęciu następnej.
hblnk	Sygnał blankingu poziomego (horizontal blanking) oznaczający czas, gdy wyświetlacz nie wyświetla aktywnego obrazu podczas przejścia do następnej linii poziomej.
rgb [11:0]	Sygnał kolorów RGB (red, green, blue) określający kolor każdego piksela wyświetlanego na ekranie, przy czym każdy kolor ma swoją 4-bitową składową.

e) vga_out_draw_win – draw_win to draw_mouse

nazwa sygnału	opis
vsync	Sygnał synchronizacji pionowej (vertical sync) informujący monitor o zakończeniu jednej klatki i rozpoczęciu następnej.
vcount [10:0]	Licznik pionowy (vertical count) określający bieżącą linię poziomą, na której odbywa się wyświetlanie lub blanking w pionie.
vblnk	Sygnał blankingu pionowego (vertical blanking) wskazujący na czas, gdy nie ma wyświetlania aktywnego obrazu, a elektronika monitora wraca do górnej części ekranu.
hcount [10:0]	Licznik poziomy (horizontal count) określający bieżącą pozycję pikseli w linii poziomej wyświetlania lub blankingu.
hsync	Sygnał synchronizacji poziomej (horizontal sync) informujący monitor o zakończeniu jednej linii skanowania i rozpoczęciu następnej.
hblnk	Sygnał blankingu poziomego (horizontal blanking) oznaczający czas, gdy wyświetlacz nie wyświetla aktywnego obrazu podczas przejścia do następnej linii poziomej.
rgb [11:0]	Sygnał kolorów RGB (red, green, blue) określający kolor każdego piksela wyświetlanego na ekranie, przy czym każdy kolor ma swoją 4-bitową składową.

f) Draw_win_data_in – chess_board to draw_win

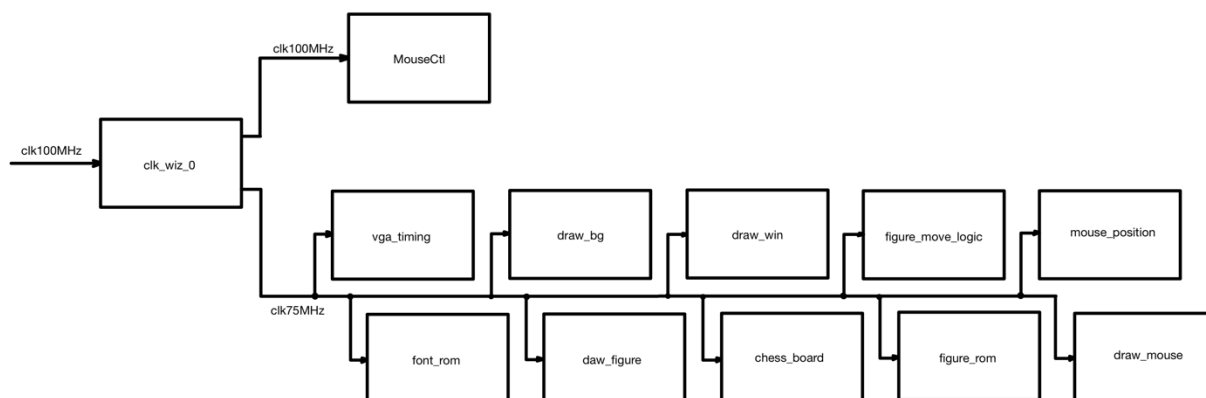
nazwa sygnału	opis
White_win	Białe figury wygrały.
Black_win	Czarne figury wygrały.

g) Positions_on_board – chess_board to figure_move_logic

nazwa sygnału	opis
Figure_taken	Kod wybranej figury przez gracza
Pp_pos	Współrzędne wybranej figury przez gracza.
White_castle	Sygnał mówiący o białej roszadzie.
Black_castle	Sygnał mówiący o czarnej roszadzie.

4.2. Rozprowadzenie sygnału zegara

W całym układzie używamy dwóch zegarów clk_75 – 75MHz oraz clk_100 – 100MHz. Zegara clk100 używa myszka pozostałe moduły używają zegara clk75.



Schemat 4. Schemat rozprowadzenia sygnału zegara w projekcie.

5. Implementacja**5.1. Lista zignorowanych ostrzeżeń Vivado.**

Identyfikator ostrzeżenia	Liczba wystąpień	Uzasadnienie
[Synth 8-7080]	1	Brak tej optymalizacji nie jest kluczowe dla obecnej wersji projektu.
[Synth 8-5856]	1	Użycie rejestrów zamiast pamięci 3D nie wpływa negatywnie na projekt pod względem wydajności, powierzchni logicznej, czy innych zasobów sprzętowych.
[Synth 8-6014]	4	Korzystamy tylko części sygnałów, które nam są potrzebne w module pozostawiając część sygnałów w interfejsie nie używanych

5.2. Wykorzystanie zasobów

Tabela z wykorzystaniem zasobów z Vivado.

Utilization

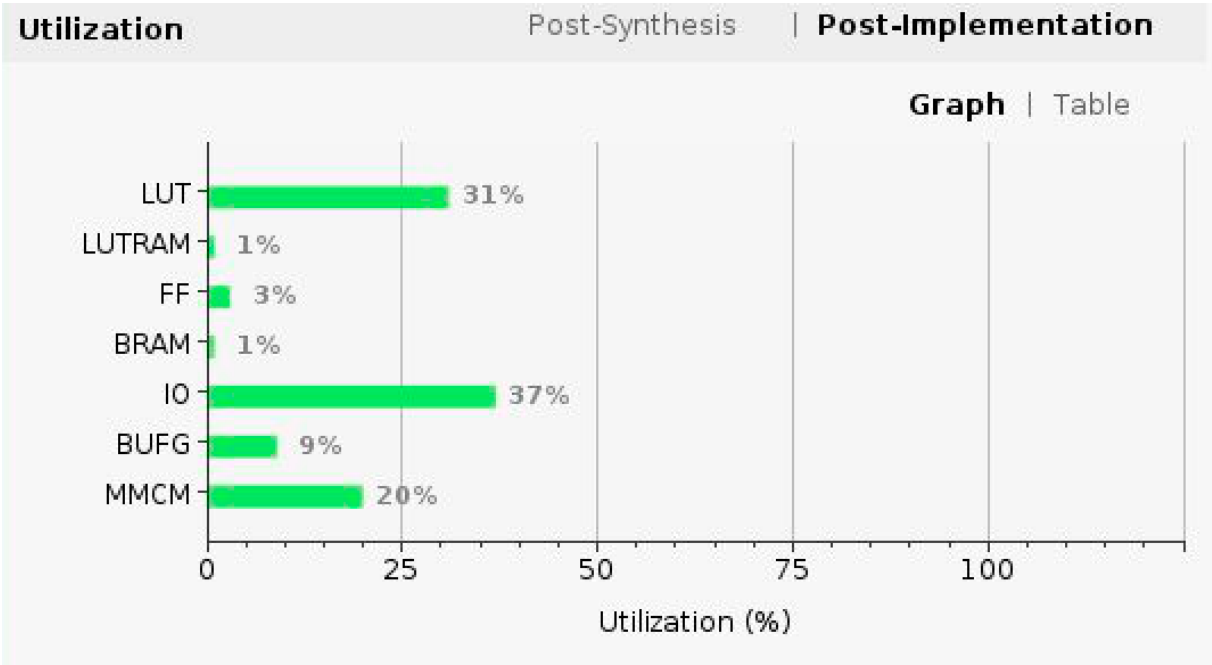
Post-Synthesis

Post-Implementation

Graph

Table

Resource	Utilization	Available	Utilization %
LUT	6385	20800	30.70
LUTRAM	2	9600	0.02
FF	1068	41600	2.57
BRAM	0.50	50	1.00
IO	39	106	36.79
BUFG	3	32	9.38
MMCM	1	5	20.00



5.3. Marginesy czasowe

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0.146 ns	Worst Hold Slack (WHS): 0.090 ns	Worst Pulse Width Slack (WPWS): 3.000
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 1679	Total Number of Endpoints: 1679	Total Number of Endpoints: 1081
All user specified timing constraints are met.		

6. Film.

Link do ściągnięcia filmu:

https://drive.google.com/file/d/1sZg7CeJTgFcqT56xBfZ7OOk9A3KuLzoQ/view?usp=drive_link