

Tytuł: **Super PONG**

**Autorzy: Dominik Bachurski (DB),
Wiktor Dziedzic (WD)**

Ostatnia modyfikacja: 28.08.2023

Spis treści

1. Repozytorium git.....	1
2. Wstęp	1
3. Specyfikacja	2
3.1. Opis ogólny algorytmu.....	2
3.2. Tabela zdarzeń	2
4. Architektura.....	3
4.1. Moduł: top	3
4.1.1. Schemat blokowy	3
4.1.2. Porty.....	4
a) top_pong_basys, input	4
b) top_pong_basys, output	3
4.1.3. Interfejsy	3-10
4.2. Rozprowadzenie sygnału zegara	10
5. Implementacja	11
5.1. Lista zignorowanych ostrzeżeń Vivado.....	11
5.2. Wykorzystanie zasobów	11
5.3. Marginesy czasowe	11
6. Film.	11

1. Repozytorium git

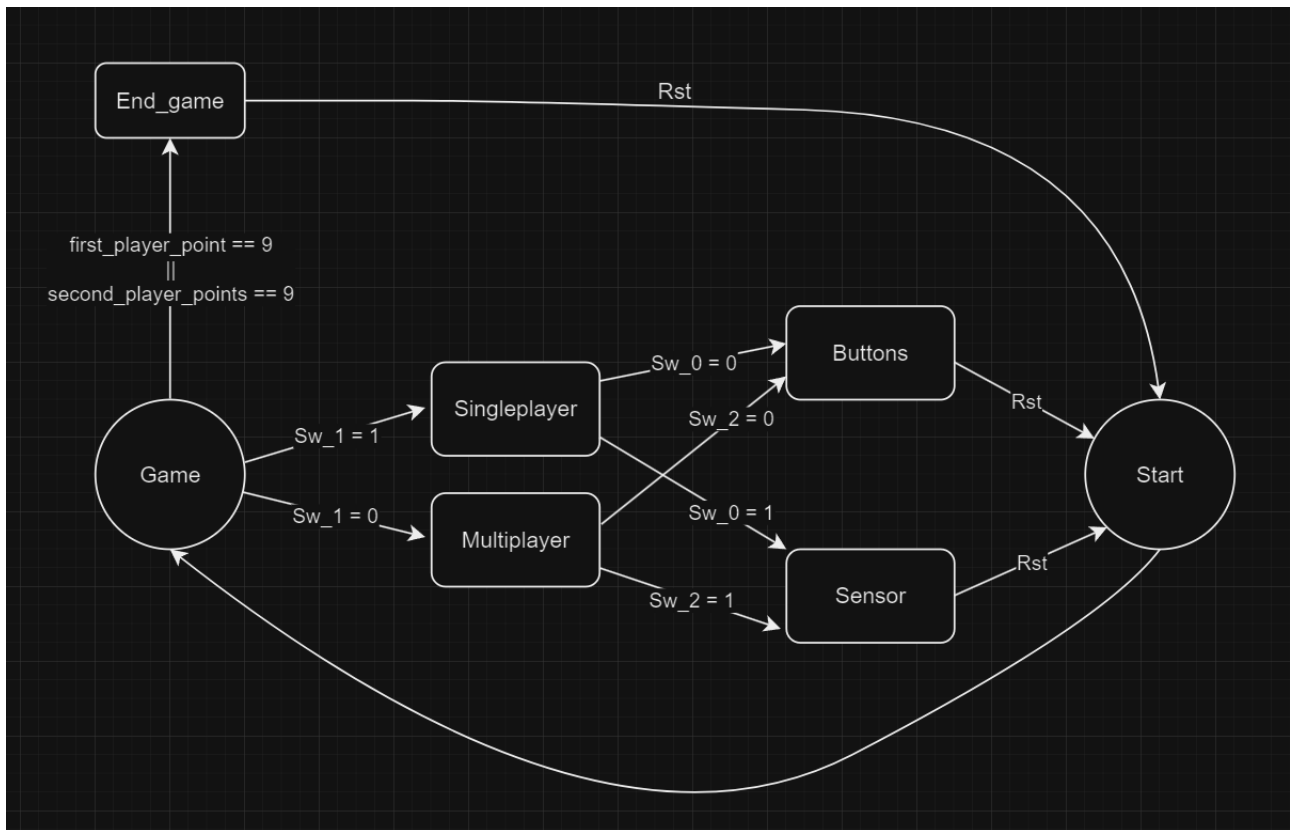
<https://github.com/bachurson/PONG-FPGA>

2. Wstęp

Nasz pomysł na projekt jest wynikiem kombinacji wcześniejszego pomysłu wykorzystania czujników odległości do innej aplikacji i stworzeniu ciekawej gry. Gra Pong w której występuje możliwość sterowania czujnikiem wydawała się dla nas najciekawszą opcją.

3. Specyfikacja

3.1. Opis ogólny algorytmu



3.2. Tabela zdarzeń

Zdarzenie	Kategoria	Reakcja systemu
BtnC	Start programu	Uruchomienie gry
Piłka uderza w ścianę	Gra	Zmiana kierunku ruchu piłki
Piłka uderza w lewą krawędź ekranu	Gra	Punkt dla bota lub drugiego gracza
Piłka uderza w prawą krawędź ekranu	Gra	Punkt dla pierwszego gracza
Piłka uderza w środek rakietki	Gra	Piłka odbija się z wysoką precyzyjnością zgodną z zasadami fizyki
Piłka uderza w okolicach odległości $\frac{1}{4}$ długości rakietki od środka	Gra	Piłka odbija się z średnią precyzyjnością w losowym kierunku
Piłka uderza w górne krawędzie i boki rakietki	Gra	Piłka odbija się z małą precyzyjnością w losowym kierunku ze zmienną prędkością
Przełączenie switch 0 w pozycje 1	Program	Gra z wykorzystaniem czujników

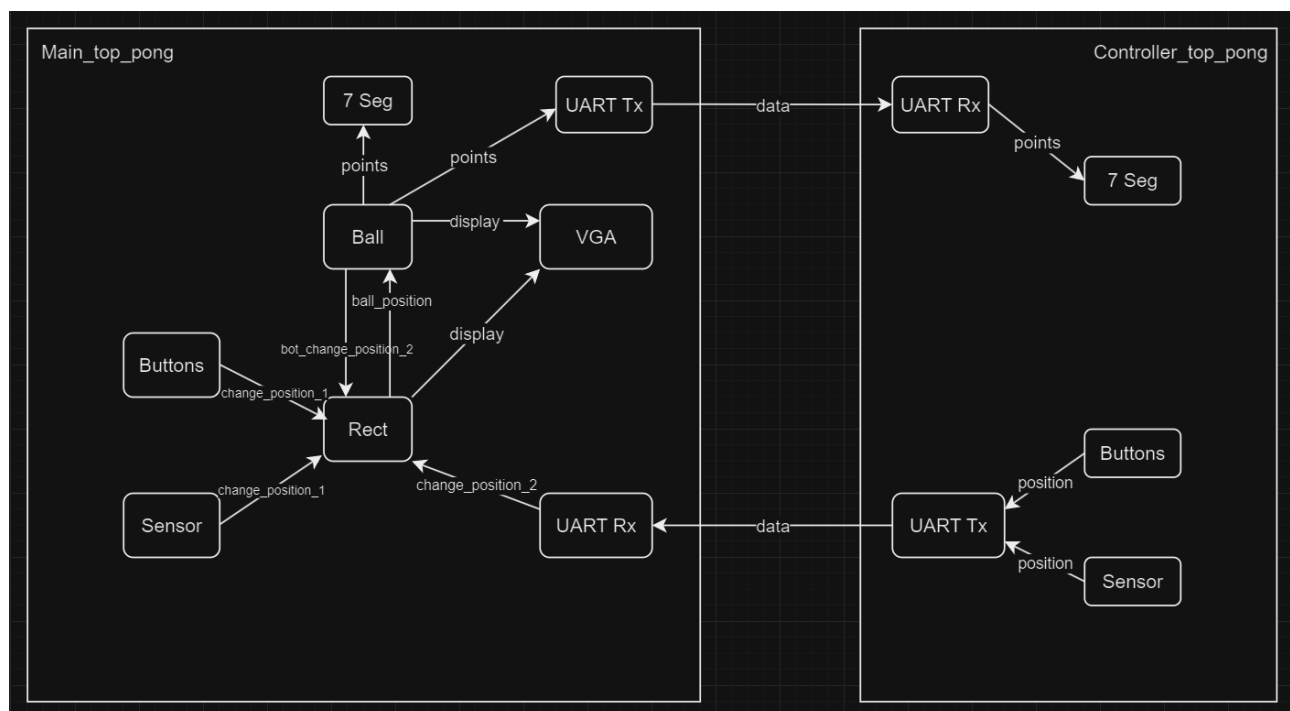
Przełączenie switch 0 w pozycje 0	Program	Gra z wykorzystaniem przycisków płytki
Przełączenie switch 1 w pozycje 1	Program	Gra na bota
Przełączenie switch 1 w pozycje 0	Program	Gra multiplayer
Przełączenie switch 2 w pozycje 1	Program	Oczekiwanie na sygnał z czujnika przesłany przez UART z drugiej płytki
Przełączenie switch 2 w pozycje 0	Program	Gra multiplayer z drugim graczem za pomocą przycisków
Przytrzymanie BtnU	Gra	Przesunięcie rakiety o 10 pikseli w górę co 10ms
Przytrzymanie BtnD	Gra	Przesunięcie rakiety o 10 pikseli w dół co 10ms
Zmiana odległości obiektu od czujnika	Gra	Przesunięcie rakiety proporcjonalnie do zmierzonej odległości
Zdobycie punktu	Program	Wyświetlenie aktualnego wyniku na płycie ustawienie piłki w pozycji startowej
Zdobycie 9 punktów przez któregoś z graczy	Program	Pileczka przestaje się ruszać program oczekuje na sygnału resetu do ponownego rozpoczęcia gry

4. Architektura

4.1. Moduł: top

Osoba odpowiedzialna: DB WD

4.1.1. Schemat blokowy



4.1.2. Porty

a) *Top_pong_basys, in*

nazwa portu	opis
clk	Sygnał zegarowy
btnC	Przycisk C
btnD	Przycisk D
btnU	Przycisk U
echo	Sygnał echo – odległość obiektu od czujnika
echo_second	Sygnał echo 2 – odległość obiektu od drugiego czujnika
RxD	Wejście szeregowe dla danych odbiornika
sw[2:0]	Przełączniki SW[2:0] – zmiana trybu programu

b) *Top_pong_basys, out*

nazwa portu	opis
Vsync	Sygnał synchronizacji pionowej
Hsync	Sygnał synchronizacji poziomej
vgaRed[3:0]	Składowa czerwona sygnału VGA
vgaGreen[3:0]	Składowa zielona sygnału VGA
vgaBlue[3:0]	Składowa niebieska sygnału VGA
JA1	Sygnał wyjścia JA1 zegara
an[3:0]	Wyjścia aktywacji segmentów na wyświetlaczu 7-SEG
seg[6:0]	Sygnały segmentów wyświetlacza 7-SEG
trig	Sygnał wyzwalania czujnika
trig_second	Sygnał 2 wyzwalania drugiego czujnika
RxData[7:0]	Wyświetlanie na ledach odebranej wartości bitowo z drugiej płytki
TxD	Sygnał wyjściowy do wyświetlania aktualnego wyniku na drugiej płytce
reset	Wspólny sygnał resetu na obu płytkach

c)

4.1.3. Interfejsy

a) *Top_pong, in*

nazwa portu	opis
btnd	Przycisk D
btneu	Przycisk U
RxD	Wejście szeregowe dla danych
echo	Sygnał echo
echo_second	Drugi sygnał echo
mode	Przełącznik tryb pracy: czujnik – przyciski gracz nr 1
mode_2	Przełącznik tryb pracy: multiplayer - bot

mode_3	Przełącznik tryb pracy odbieranie sygnału z czujników bądź przycisków od drugiego gracza
--------	------------------------------------------------------------------------------------------

b) Top_pong, out

nazwa portu	opis
vs	Sygnał synchronizacji pionowej
hs	Sygnał synchronizacji poziomej
r[3:0]	Składowa czerwona sygnału VGA
g[3:0]	Składowa zielona sygnału VGA
b[3:0]	Składowa Niebieska sygnału VGA
an[3:0]	Wejścia aktywacji segmentów na wyświetlacza 7-SEG
seg[6:0]	Sygnały segmentów wyświetlacza 7-SEG
trig	Sygnał wyzwalania
trig_second	Drugi sygnał wyzwalania
RxData	Przekazywana wartość odebrana z drugiej płytki
TxD	Sygnał posiadający informacje o wyniku przekazywany drugiej płytce

c) Debounce, in

nazwa portu	opis
btn	Przycisk poddawany debouncingowi

d) Debounce, out

nazwa portu	opis
btn_out	Wyjściowy sygnał przycisku

e) Draw_rect, in

nazwa portu	opis
btn_up	Przycisk sterujący rakieta w górę
btn_down	Przycisk sterujący rakieta w dół
pos[7:0]	Pozycja podawana z czujnika
mode	Tryb pracy czujniki - przyciski
vga	Interfejs VGA

f) Draw_rect, out

nazwa portu	opis
y_position[10:0]	Pozycja Y rakietki
vga_out	Interfejs VGA

g) Draw_rect_2, in

nazwa portu	opis
ball_y_pos[10:0]	Pozycja Y piłki
mode_2	Tryb gry bot - multiplayer
pos[7:0]	Pozycja z czujnika
vga	Interfejs VGA
mode_3	Tryb pracy czujniki – przyciski drugiego gracza

h) Draw_rect_2, out

nazwa portu	opis
y_position[10:0]	Pozycja Y
vga_out	Interfejs VGA

i)**j) Random_4, out**

nazwa portu	opis
Q[1:N]	Wyjście sekwencji pseudolosowej

k) Ball_ctl, in

nazwa portu	opis
rect_y_pos	Pozycja pionowa pierwszej rakietki
rect2_y_pos	Pozycja pionowa drugiej rakietki
random_4	Losowa sekwencja czterobitowa

l) Ball_ctl, out

nazwa portu	opis
xpos	Pozycja pozioma piłki
ypos	Pozycja pionowa piłki
point_first_player	Punkty pierwszego gracza
point_second_player	Punkty drugiego gracza
score_flag	Flaga dla sygnału transmitującego o zdobytym punkcie przez któregoś z graczy

m) Draw_ball, in

nazwa portu	opis
x_position	Pozycja pozioma piłki
y_position	Pozycja pionowa piłki
vga	Interfejs VGA

n) Draw_ball, out

nazwa portu	opis
vga_out	Interfejs VGA

o) Seg7_display, in

nazwa portu	opis
points_first_player	Wynik pierwszego gracza liczba 0-9
points_second_player	Wynik drugiego gracza liczba 0-9

p) Seg7_display, out

nazwa portu	opis
an	Sygnał sterujący pozycja wyświetlacza 7-SEG
seg	Sygnał sterujący segmentami wyświetlacza 7-SEG

q) Uart_rx, in

nazwa portu	opis
RxD	Sygnał wejściowy UART
mode	Tryb pracy odbieranie przyciski – czujnik od drugiego gracza

r) Uart_rx out

nazwa portu	opis
RxData	Odebrany sygnał 8 bitowy

s) Sensor, in

nazwa portu	opis
echo	Sygnał odebrany od czujnika

t) Sensor, out

nazwa portu	opis
trig	Sygnał wyzwalający do czujnika
distance	Zmierzona odległość

u) Sensor_second, in

nazwa portu	opis
echo_second	Sygnał odebrany od drugiego czujnika

v) Sensor_second, out

nazwa portu	opis
trig_second	Sygnał wyzwalający do drugiego czujnika
distance_second	Zmierzona odległość od drugiego czujnika

w) Buffor, in

nazwa portu	opis
input_data	Dane przekazane z odbiornika RxD
mode	Buforowanie sygnału odebranego czujniki - przyciski

x) Buffor, out

nazwa portu	opis
data_buffored	Zbuforowane dane z UARTA

y) Buffor_2, in

nazwa portu	opis
input_data	Buforowanie drugi raz danych z bufor
mode	Buforowanie sygnału odebranego czujniki - przyciski

z) Buffor_2, out

nazwa portu	opis
data_buffered	Dane po podwójnym zbuforowaniu

aa) Compare_pos, in

nazwa portu	opis
pos	Odległość z pierwszego czujnika
pos_second	Odległość z drugiego czujnika

bb) Compare_pos, out

nazwa portu	opis
y_position	Pozycja wyjściowa dokładniejsza

cc) Vga_timing, in

nazwa portu	opis
vcount	Licznik pionowy
hcount	Licznik poziomy
vsync	Sygnał synchronizacji pionowej
hsync	Sygnał synchronizacji poziomej
vblnk	Sygnał blankingu pionowego
hblnk	Sygnał blankingu poziomego
rgb	Dane koloru

dd) Vga_timing, out

nazwa portu	opis
vcount	Licznik pionowy
hcount	Licznik poziomy
vsync	Sygnał synchronizacji pionowej
hsync	Sygnał synchronizacji poziomej
vblnk	Sygnał blankingu pionowego
hblnk	Sygnał blankingu poziomego
rgb	Dane koloru

ee) TxD, in

nazwa portu	opis
flag_transmit	Flaga do wysyłania sygnału w momencie zmiany stanu punktów
points_first_player [3:0]	Ilość punktów pierwszego gracza wysyłana do drugiej płytki
points_second_player [3:0]	Ilość punktów drugiego gracza wysyłana do drugiej płytki

ff) TxD, out

nazwa portu	opis
TxD	Sygnał transmitujący

gg) Draw_bg, in

nazwa portu	opis
vga_in	Interfejs VGA

hh) Draw_bg, out

nazwa portu	opis
vga_out	Interfejs VGA

4.2. Rozprowadzenie sygnału zegara

Osoba odpowiedzialna: DB, WD

W projekcie przyjęliśmy łączną częstotliwość zegara wynoszącą 65 MHz, która jest używana we wszystkich modułach. Głównym źródłem sygnału zegarowego jest moduł generatora zegara, umieszczony w module głównym projektu. Moduł generatora zegara umieszczamy w module głównym projektu. W pozostałych modułach używamy tylko i wyłącznie sygnały zegara wygenerowane przez ten moduł.



5. Implementacja

5.1. Lista zignorowanych ostrzeżeń Vivado.

Identyfikator ostrzeżenia	Liczba wystąpień	Uzasadnienie
Synth 8-7129	24	Sygnal RGB interfejsu vga_if nie jest przekazywany z vga_timing do vga_bg, jako że nie jest to konieczne, postanowiliśmy nie pisać zbędnego kodu.
Synth 8-7080	1	Ten warning jest powszechny i nie wpływa na poprawne działanie programu.

5.2. Wykorzystanie zasobów

Graph Table			
Resource	Utilization	Available	Utilization %
LUT	1453	20800	6.99
LUTRAM	2	9600	0.02
FF	829	41600	1.99
IO	48	106	45.28
BUFG	2	32	6.25
MMCM	1	5	20.00

5.3. Marginesy czasowe

WNS - hold: 0.345ns

WNS - setup: 0.056ns

6. Film.

Link do ściągnięcia filmu:

https://drive.google.com/file/d/1km_EPiU1aDkN2u0Ax9Xkq_xUpd3KV2_t/view?usp=sharing