

# Tytuł: **FPGA\_Warships**

**Autorzy: Natalia Kapuścińska(NK),  
Paweł Zięba(PZ)**

Ostatnia modyfikacja: 31.08.2023

## **Spis treści**

1.Repozytorium git.....	1
2.Wstęp .....	2
3.Specyfikacja .....	2
3.1.Opis ogólny algorytmu.....	2
3.2.Tabela zdarzeń.....	4
4.Architektura .....	4
4.1.Moduł: top.....	4
4.1.1.Schemat blokowy .....	5
4.1.2.Porty .....	6
4.1.3.Interfejsy .....	7
4.1.4.Sygnały.....	10
4.2.Rozprowadzenie sygnału zegara .....	12
5.Implementacja .....	13
5.1.Lista zignorowanych ostrzeżeń Vivado. ....	13
5.2.Wykorzystanie zasobów .....	14
5.3.Marginesy czasowe .....	14
6.Film .....	14
7.Schemat połączeń projektu .....	15
8.LOG Warning_Summary .....	16

## 1. Repozytorium git

Adres repozytorium GITa:

[https://github.com/PawelZieba01/FPGA\\_WARSHIPS](https://github.com/PawelZieba01/FPGA_WARSHIPS)

Branch'e, które zostały zmergowane do maina w trakcie tworzenia projektu, były na bieżąco usuwane aby zachować klarowność repozytorium. Historia rozgałęzień projektu widoczna jest w historii commitów gita.

W ramach dokumentacji prowadzona była strona na portalu Confluence:

<https://fpgastatki2023.atlassian.net/wiki/spaces/SD/overview> (zaproszono [kaczmarczyk@agh.edu.pl](mailto:kaczmarczyk@agh.edu.pl))

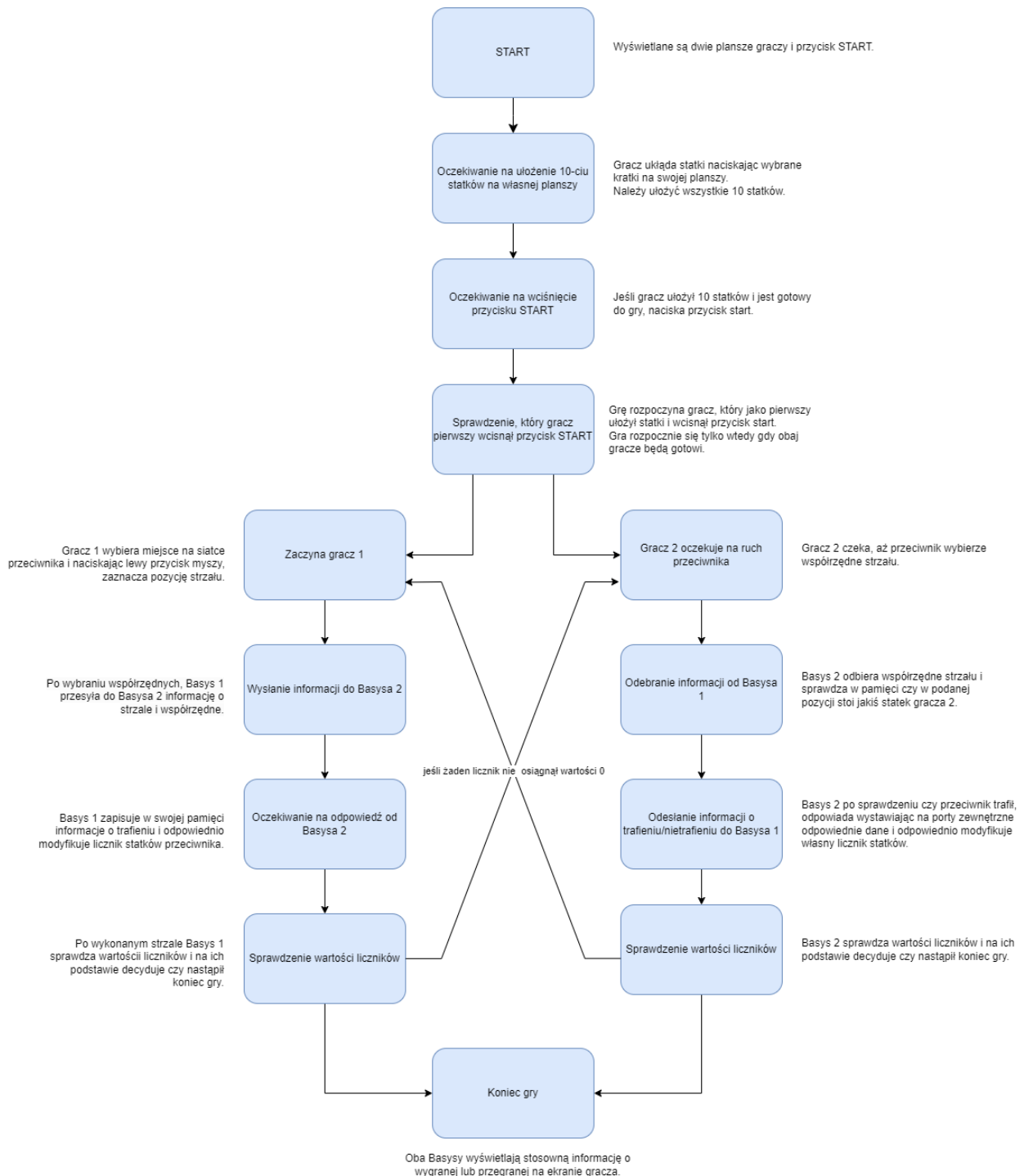
\*jest to tylko uzupełnienie raportu

## 2. Wstęp

Pomysł na statki FPGA narodził się jako odpowiedź na liczne oszustwa w tej popularnej grze, w wersji vivado wszystkie trafienia są odpowiednio oznakowane, nie ma też żadnych wątpliwości na temat historii trafień gdyż ta jest przechowywana do samego końca.

### 3. Specyfikacja

#### 3.1. Opis ogólny algorytmu



Gra rozpoczyna się od ułożenia przez każdego z graczy dziesięciu statków na własnej planszy. Podczas układania statków gracz może obserwować pozostałą ilość do położenia, na wyświetlaczu. Po zakończeniu układania gracz wciska przycisk „START” (gracz, który jako pierwszy wcisnął przycisk – rozpoczyna jako pierwszy). Rozpoczynający gracz wybiera z obszaru siatki przeciwnika pole, w które chce strzelić i wciska LPM. Po wykonanym ruchu układy Basys3 zapisują odpowiednie informacje i na ekranach wyświetlają się odpowiednie kolory pól, w zależności czy statek został trafiony czy też nie. W trakcie całej gry gracz może obserwować stan liczników statków na wbudowanym wyświetlaczu 7-

segmentowym swojego Basyska. Po osiągnięciu przez jeden z liczników wartości równej zero, gra informuje o wygranej i przegranej odpowiednich graczy, przez wyświetlenie stosownej informacji na ekranie – Koniec gry.

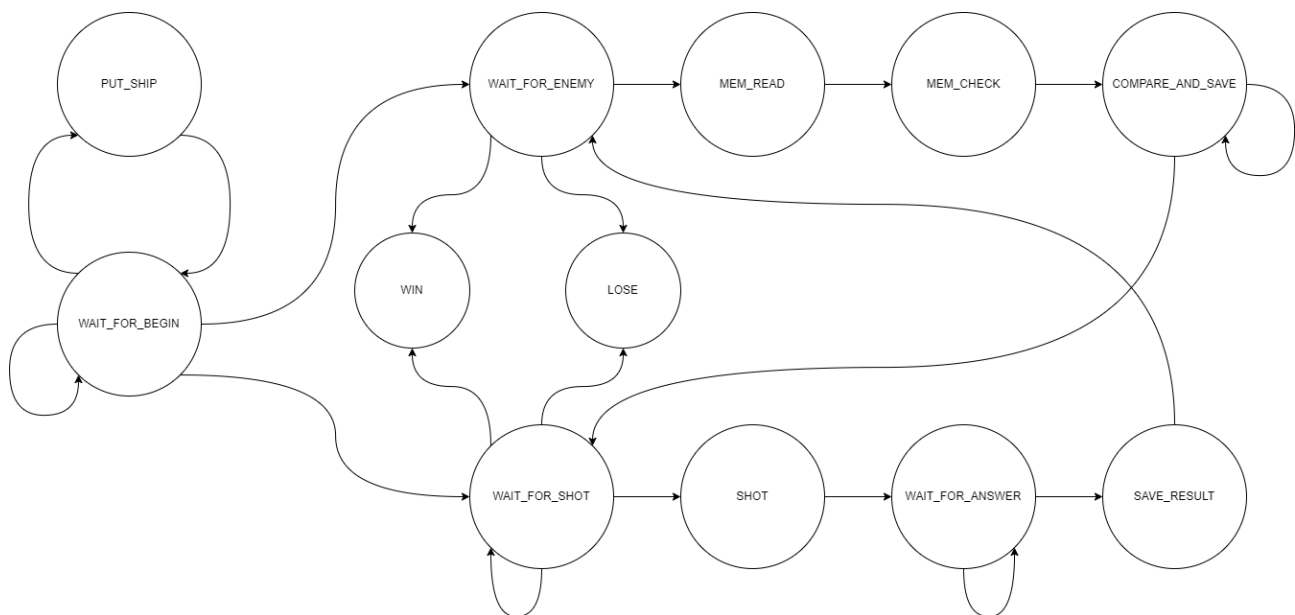
*Komunikacja obu układów oparta jest na dwóch równoległych magistralach 8-bitowych służących jako wyjście i wejście adresu siatki. Dodatkowo użyto 4 sygnałów sterujących ready1, hit1 jako wyjścia oraz ready2, hit2 jako wejścia. Sygnały sterujące służą do sygnalizowania gotowości układu i akcji strzału. Oba układy Basys3 są połączone „symetrycznie” – hit1-hit2, ready1-ready2 itd.*

*Komunikacja i cały algorytm jest w pełni „symetryczny” – oba układy Basys3 mają wgrany ten sam program, jedyną różnicą są stany w jakich się znajdują w danej chwili.*

*Kolory pól:*

- zielony – mój statek
- niebieski – nietrafiony statek
- biały – puste pole
- czerwony – trafiony statek

Główny algorytm gry można również przedstawić przy pomocy **uproszczonego** diagramu głównej maszyny stanów.



### 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 dotyczący różnych stanów działania programu.*

Zdarzenie	Kategoria	Reakcja systemu
Reset/Uruchomienie części sprzętowej	Start	Zainicjalizowanie wszystkich rejestrów i pamięci wartościami domyślnymi i wyświetlenie na ekranie obszaru gry.
LPM w obszarze siatki gracza	Start	Sprawdzenie czy nie położono już 10-ciu statków na planszy, jeśli nie to zapisanie do pamięci pozycji statku gracza i zaktualizowanie ekranu.
LPM w obszarze przycisku „START”	Start	Jeśli położono 10 statków na planszy gracza – Ustawienie sygnału ready1 na 1, ukrycie przycisku „START” i oczekiwanie aż przeciwnik wciśnie swój przycisk.
LPM w obszarze siatki przeciwnika	Tryb strzelania	Wysłanie do drugiej płytki Basys3 informacji o strzale i współrzędnych, które wybrał gracz.
Licznik przeciwnika == 0	Sprawdzanie stanu liczników	Zakończenie gry, wyświetlenie informacji o wygranej.
Licznik gracza == 0	Sprawdzanie stanu liczników	Zakończenie gry, wyświetlenie informacji o przegranej.
Sygnały hit2 i ready2 = 1	(zdarzenie zewnętrzne) Oczekiwanie na ruch przeciwnika	Odebranie współrzędnych strzału i sprawdzenie w pamięci czy trafiono w statek oraz odesłanie stosownej informacji
Sygnał ready2 = 1 hit2 = 0 lub 1	(zdarzenie zewnętrzne) Oczekiwanie na odpowiedź przeciwnika	Odebranie informacji o trafieniu (hit2=1) lub nietrafieniu (hit2=0) i zapisanie do pamięci stosownej informacji.

## 4. Architektura

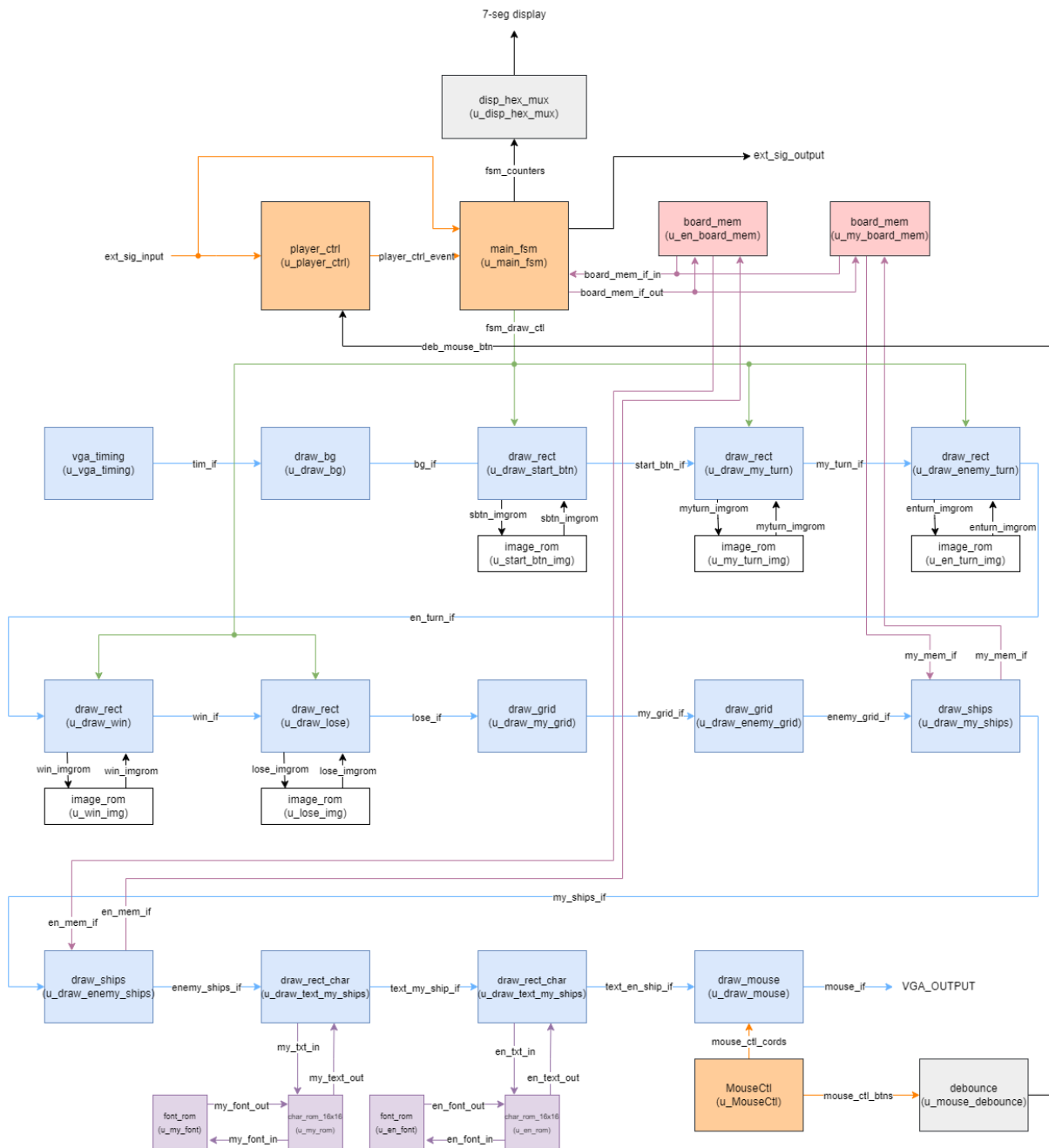
### 4.1. Moduł: top

Osoba odpowiedzialna: PZ, NK

Moduł top składa się z głównego toru VGA odpowiedzialnego za generowanie sygnałów dla monitora. Tor VGA odczytuje z dwóch bloków pamięci informacje o siatkach graczy (czy dane pole jest puste, trafione, nietrafione itd.). Do modułów pamięci dane są zapisywane i odczytywane przez główną maszynę stanów main\_FSM. Maszyna stanów pełni funkcję głównej logiki sterującej, wspomaganej przez moduł player\_ctrl, który ma za zadanie monitorować działania użytkownika (sprawdzanie pozycji myszy i interakcji obiektów). Dodatkowo moduł top wyposażono w blok obsługujący wyświetlacz 7-segmentowy na którym wyświetlane są wartości liczników gracza i przeciwnika (liczniki są integralną częścią maszyny stanów).

### 4.1.1. Schemat blokowy

Schemat blokowy modułu top: top\_warships



\*z powodu nieprawidłowości i błędnego zachowania vivado oraz projektu, zrezygnowano ze zmiany nazw sygnałów w module top na poprzedzone nagłówkiem interface'u

#### 4.1.2. Porty

##### a) *mou – mouse\_ctl, inout*

nazwa portu	opis
ps2_data	szeregowe wejście/wyjście danych myszki
ps2_clk	zegar myszki

##### b) *vga – vga\_ctl, output*

nazwa portu	opis
vga_vs	sygnał synchronizacji pionowej VGA
vga_hs	sygnał synchronizacji poziomej VGA
vga_r[3:0]	sygnał czerwonego koloru VGA
vga_g[3:0]	sygnał zielonego koloru VGA
vga_b[3:0]	sygnał niebieskiego koloru VGA

#### 4.1.3.

##### a) *disp – 7seg display output*

nazwa portu	opis
disp_sseg[6:0]	Segmenty wyświetlacza 7-segmentowego (katody)
disp_an[3:0]	Anody wyświetlacza 7-segmentowego

##### b) *ctrl – control signals, input*

nazwa portu	opis
ctrl_ship_cords_in[7:0]	Sygnał współrzędnych siatki od drugiego układu Basys3
ready2	Sygnał gotowości od drugiego układu Basys3
hit2	Sygnał strzału/trafienia od drugiego układu Basys3

##### c) *ctrl – control signals, output*

nazwa portu	opis
ctrl_ship_cords_out[7:0]	Wyjście sygnału współrzędnych siatki dla drugiego układu Basys3
ready1	Sygnał gotowości dla drugiego układu Basys3
hit1	Sygnał strzału/trafienia dla drugiego układu Basys3

**4.1.4. Interfejsy****a) *tim\_if – timing vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblink	sygnał horyzontalny blank VGA
vblink	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**b) *lose\_if – draw\_lose vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblink	sygnał horyzontalny blank VGA
vblink	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**c) *win\_if – draw\_win vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblink	sygnał horyzontalny blank VGA
vblink	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**d) *bg\_if – draw\_bg vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblink	sygnał horyzontalny blank VGA
vblink	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA



**e) start\_btn\_if – draw\_start\_btn vga output**

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**f) my\_turn\_if – draw\_my\_turn vga output**

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**g) en\_turn\_if – draw\_en\_turn vga output**

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**h) my\_grid\_if – draw\_my\_grid vga output**

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

***i) enemy\_grid\_if – draw\_enemy\_grid vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

***j) my\_ships\_if – draw\_my\_ships vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

***k) enemy\_ships\_if – draw\_enemy\_ships vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

***l) text\_my\_ships\_if – draw\_text\_my\_ship vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**m) *text\_en\_ships\_if – draw\_text\_en\_ship vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**n) *mouse\_if – draw\_mouse vga output***

<b>nazwa sygnału</b>	<b>opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**4.1.5. Sygnały****a) *mouse***

<b>nazwa sygnału</b>	<b>opis</b>
mouse_x_pos [11:0]	pozycja X myszy z kontrolera
mouse_y_pos [11:0]	pozycja Y myszy z kontrolera
mouse_left	sygnał lewego klawisza myszy (LPM)
mouse_left_db	odfiltrowany sygnał LPM – reakcja tylko przez jeden cykl zegara

**b) *start\_btn\_icon***

<b>nazwa sygnału</b>	<b>opis</b>
rgb_pixel_start_btn [11:0]	kolor pixela z modułu image_rom
rgb_pixel_addr_start_btn [13:0]	adres pixela dla modułu image_rom
start_btn	sygnał dla kontrolera interakcji z graczem (player_ctrl) informujący o wciśnięciu przycisku LPM
start_btn_en	sygnał enable dla moduły rysującego ikonę (rysuj/nie rysuj)

**c) enemy\_turn\_icon**

<b>nazwa sygnału</b>	<b>opis</b>
rgb_pixel_enemy_turn [11:0]	kolor pixela z modułu image_rom
rgb_pixel_addr_enemy_turn [13:0]	adres pixela dla modułu image_rom
en_en	sygnał enable dla moduły rysującego ikonę (rysuj/nie rysuj)

**d) my\_turn\_icon**

<b>nazwa sygnału</b>	<b>opis</b>
rgb_pixel_my_turn [11:0]	kolor pixela z modułu image_rom
rgb_pixel_addr_my_turn [13:0]	adres pixela dla modułu image_rom
my_en	sygnał enable dla moduły rysującego ikonę (rysuj/nie rysuj)

**e) win\_icon**

<b>nazwa sygnału</b>	<b>opis</b>
rgb_pixel_my_turn [11:0]	kolor pixela z modułu image_rom
rgb_pixel_addr_my_turn [13:0]	adres pixela dla modułu image_rom
my_en	sygnał enable dla moduły rysującego ikonę (rysuj/nie rysuj)

**f) lose\_icon**

<b>nazwa sygnału</b>	<b>opis</b>
rgb_pixel_lose_turn [11:0]	kolor pixela z modułu image_rom
rgb_pixel_addr_lose_turn [13:0]	adres pixela dla modułu image_rom
lose_en	sygnał enable dla moduły rysującego ikonę (rysuj/nie rysuj)

**g) my\_board\_memory**

<b>nazwa sygnału</b>	<b>opis</b>
my_board_read2_addr [7:0]	adres do odczytu danych z zegarem vga
my_board_read2_data [1:0]	dane z pamięci odczytywane z zegarem vga
my_board_read1_write1_addr[7:0]	adres do zapisu i odczytu z zegarem control
my_board_read1_data [1:0]	dane z pamięci odczytywane z zegarem control
my_board_write1_data [1:0]	dane wpisywane do pamięci z zegarem control
my_board_write_nread	wybór trybu wpisywania lub odczytywania

**h) enemy\_board\_memory**

<b>nazwa sygnału</b>	<b>opis</b>
enemy_board_read2_addr [7:0]	adres do odczytu danych z zegarem vga
enemy_board_read2_data [1:0]	dane z pamięci odczytywane z zegarem vga
enemy_board_read1_write1_addr[7:0]	adres do zapisu i odczytu z zegarem control

enemy_board_write1_data [1:0]	dale wpisywane do pamięci z zegarem control
enemy_board_write_nread	wybór trybu wpisywania lub odczytywania

**i) draw\_text\_my\_ships**

<b>nazwa sygnału</b>	<b>opis</b>
ms_char_pixels [7:0]	dane wybranej linii pixeli z pamięci font_rom
ms_char_line [3:0]	numer linii do pamięci font_rom
ms_char_xy [7:0]	adres litery do pamięci char_rom_16x16
ms_char_code [6:0]	kod znaku do pamięci font_rom

**j) draw\_text\_enemy\_ships**

<b>nazwa sygnału</b>	<b>opis</b>
e_char_pixels [7:0]	dane wybranej linii pixeli z pamięci font_rom
e_char_line [3:0]	numer linii do pamięci font_rom
e_char_xy [7:0]	adres litery do pamięci char_rom_16x16
e_char_code [6:0]	kod znaku do pamięci font_rom

**k) counters**

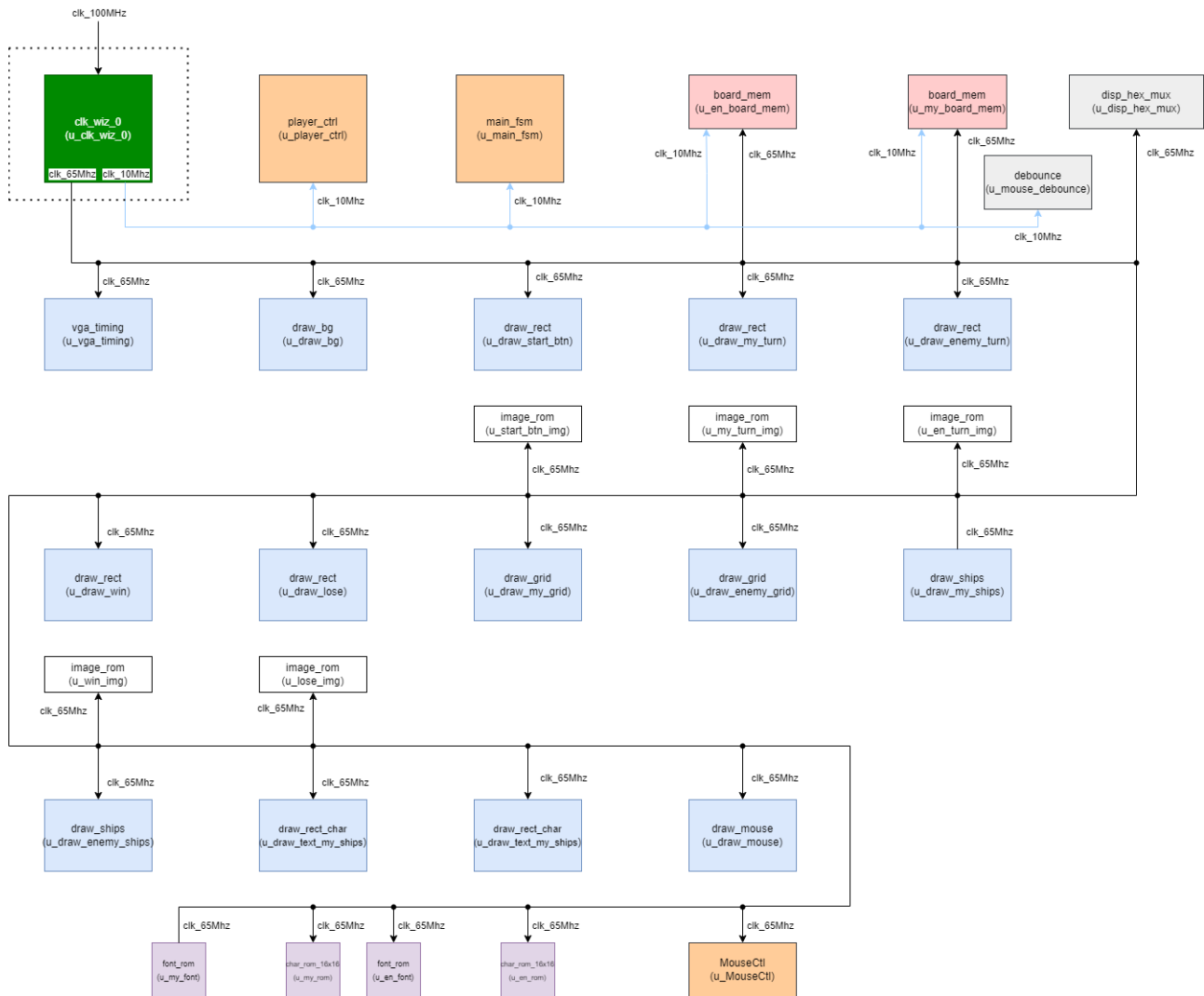
<b>nazwa sygnału</b>	<b>opis</b>
en_ctr [3:0]	sygnał licznika statków gracza
my_ctr [3:0]	sygnał licznika statków przeciwnika

**4.2. Rozprowadzenie sygnału zegara**

Osoba odpowiedzialna: PZ, NK

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

*W projekcie zastosowano dwie częstotliwości zegara, wygenerowane w module clk\_wiz\_0. Częstotliwość 65MHz odpowiada za obsługę toru VGA oraz wyświetlacza 7-segmentowego. Częstotliwość 10MHz obsługuje główną maszynę stanów, pozostałą logikę sterującą oraz komunikację z drugim układem basys3.*



## 5. Implementacja

### 5.1. Lista zignorowanych ostrzeżeń Vivado.

Identyfikator ostrzeżenia	Liczba wystąpień	Uzasadnienie
8-6014	4	Nieużywane elementy w module MouseCtl.vhd zostały automatycznie usunięte przez Vivado – nie podejmowano próby usunięcia tych ostrzeżeń aby nie modyfikować oryginalnego modułu zaczerpniętego z zajęć laboratoryjnych UEC2.
8-7129	24	Zignorowano ponieważ świadomie nie podłączono linii rgb interfejsu vga_tim w module draw_bg, gdyż moduł ten z nich nie korzysta. Pozostałe sygnały interfejsu tim_if są normalnie używane.
8-7080	1	Projekt nie spełnia wymogów równoległej syntezy – jest zbyt mały

## 5.2. Wykorzystanie zasobów

Resource	Utilization	Available	Utilization %
LUT	1931	20800	9.28
LUTRAM	35	9600	0.36
FF	2267	41600	5.45
BRAM	16	50	32.00
IO	66	106	62.26
MMCM	1	5	20.00

## 5.3. Marginesy czasowe

### Setup

Worst Negative Slack (WNS): 2.265 ns

### Hold

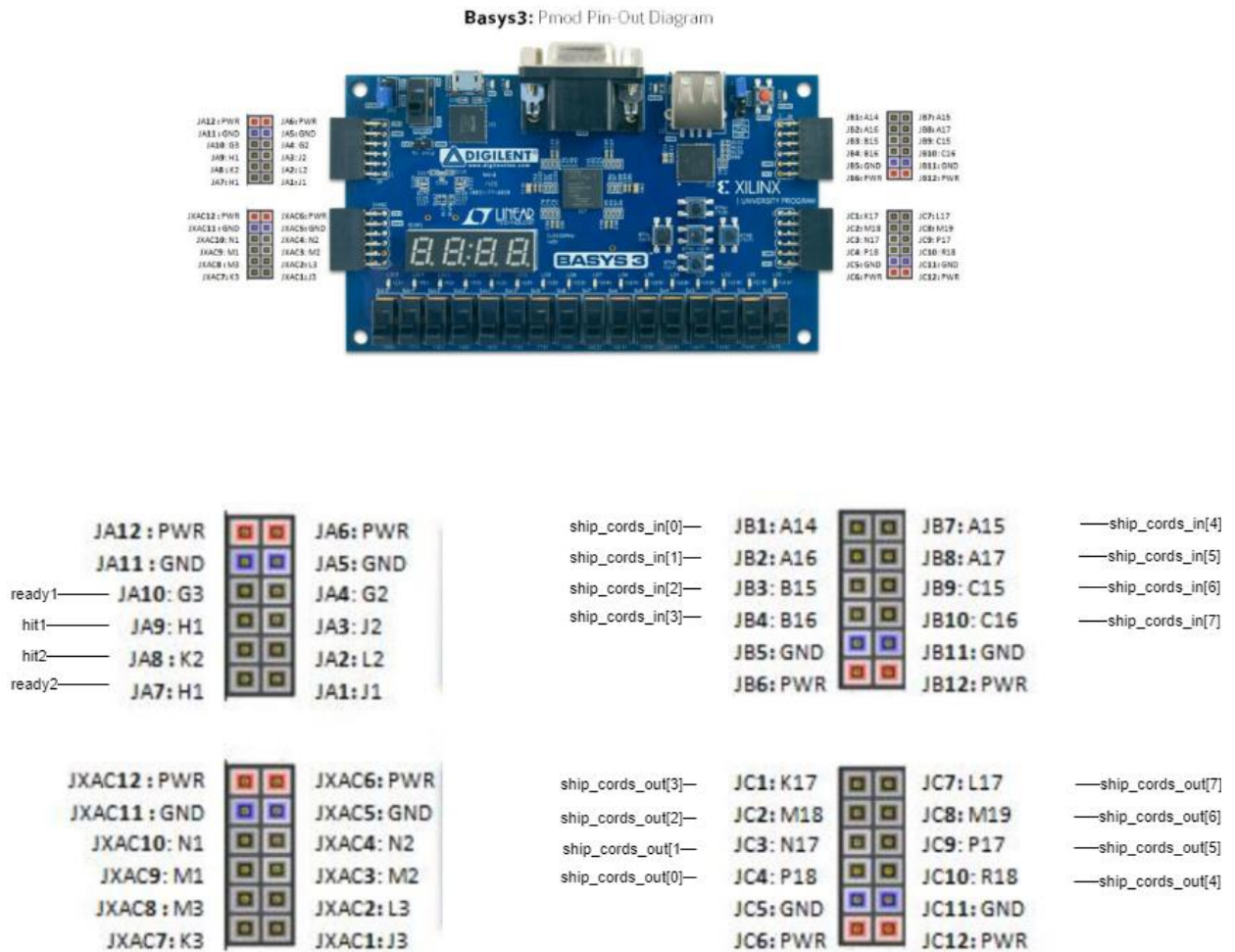
Worst Hold Slack (WHS): -0.198 ns

## 6. Film

Link do ściągnięcia filmu:

[https://drive.google.com/file/d/1rVNI9s45F6EAzhemcrMdzVRFoP97SPt7/view?fbclid=IwAR0IS4A-FiI\\_scan8XzZXFUsKb1IXHHvWTCnxh2fVmMAdyZFzxZ7nU2PVcU](https://drive.google.com/file/d/1rVNI9s45F6EAzhemcrMdzVRFoP97SPt7/view?fbclid=IwAR0IS4A-FiI_scan8XzZXFUsKb1IXHHvWTCnxh2fVmMAdyZFzxZ7nU2PVcU)

## 7. Schemat połączeń projektu



Obie płytki są połączone zgodnymi sygnałami tj. read1 (Basys1) → ready2 (Basys2), hit1 (Basys1) → hit2 (Basys2), ship\_cords\_in (Basys1) → ship\_cords\_out (Basys2) itd.



## 8. LOG Warning\_Summary

```
----SYNTHESIS----
WARNING: [Synth 8-6014] Unused sequential element set_x.inc_reg was removed. [rtl/mouse/MouseCtl.vhd:450]
WARNING: [Synth 8-6014] Unused sequential element set_x.x_inter_reg was removed. [rtl/mouse/MouseCtl.vhd:456]
WARNING: [Synth 8-6014] Unused sequential element set_y.inc_reg was removed. [rtl/mouse/MouseCtl.vhd:514]
WARNING: [Synth 8-6014] Unused sequential element set_y.y_inter_reg was removed. [rtl/mouse/MouseCtl.vhd:520]
WARNING: [Synth 8-7129] Port in\.rgb[11] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[10] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[9] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[8] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[7] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[6] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[5] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[4] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[3] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[2] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[1] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[0] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7080] Parallel synthesis criteria is not met
WARNING: [Synth 8-7129] Port in\.rgb[11] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[10] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[9] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[8] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[7] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[6] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[5] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[4] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[3] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[2] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[1] in module draw_bg is either unconnected or has no load
WARNING: [Synth 8-7129] Port in\.rgb[0] in module draw_bg is either unconnected or has no load

----IMPLEMENTATION----
CLEAR :)
```