

Tytuł: Jump King B1

# Autorzy: Miłosz Płonczyński, Tomasz Machnik

Ostatnia modyfikacja: 11.06.2025

## Spis treści

1. Repozytorium git.....	1
2. Wstęp .....	1
3. Specyfikacja .....	1
3.1. Opis ogólny algorytmu.....	1
3.2. Tabela zdarzeń .....	3
4. Architektura.....	2
4.1. Moduł: top .....	4
4.1.1. Schemat blokowy .....	4
4.1.2. Porty.....	5
4.1.3. Interfejsy.....	6
4.2. Rozprowadzenie sygnału zegara .....	8
5. Implementacja .....	9
5.1. Lista zignorowanych ostrzeżeń Vivado.....	9
5.2. Wykorzystanie zasobów .....	9
5.3. Marginesy czasowe .....	9
6. Konfiguracja sprzętu. ....	10

## 1. Repozytorium git

Adres repozytorium GITa:

<https://github.com/TMTomaszMachnik/JumpKingB1.git>

## 2. Wstęp

Pomysł na grę był zaczerpnięty z popularnej gry „Jump King”. Celem było stworzenie nieco cięższej mechanicznie gry, która będzie w pełni wykorzystywać realistyczną fizykę skoków oraz odbijanie się od krawędzi i platform.

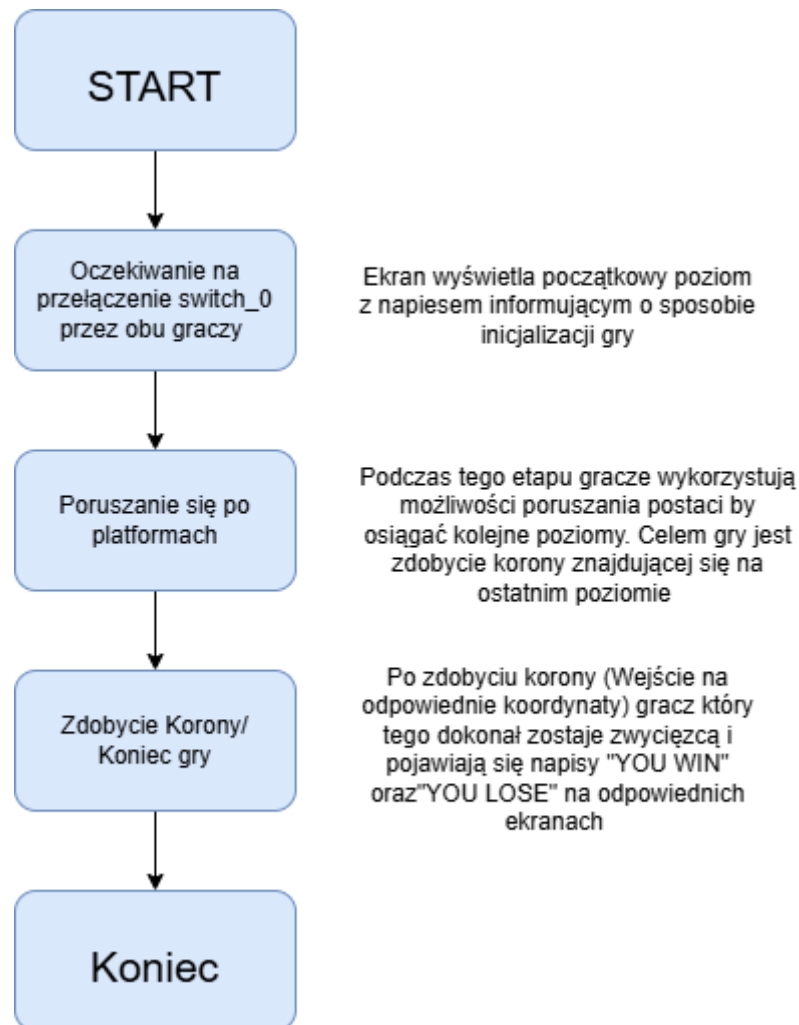
## 3. Specyfikacja

### 3.1. Opis ogólny algorytmu

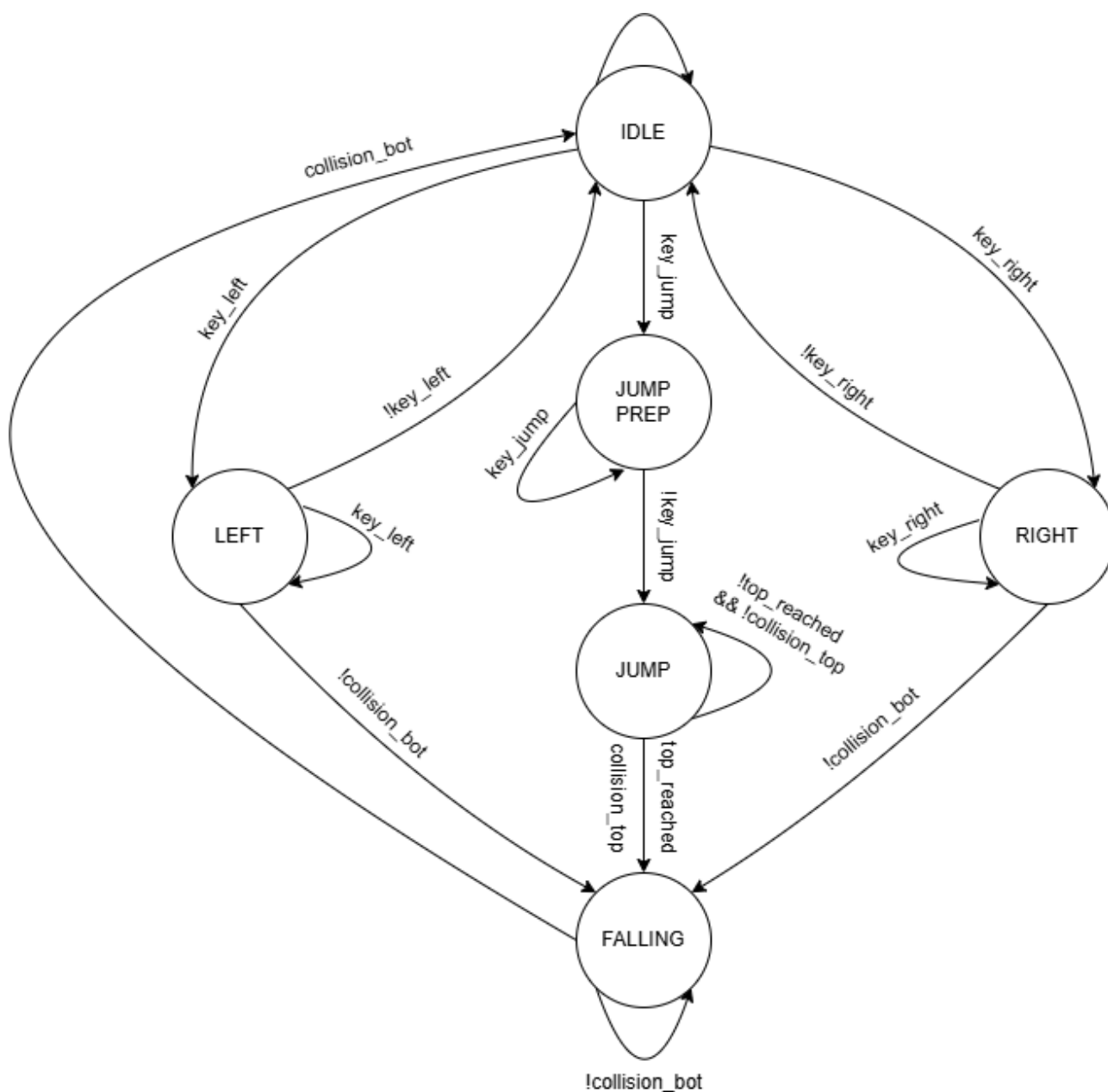
Gra rozpoczyna się w gdy gracz zgodnie z instrukcją podaną na ekranie startowym, przełączy switch\_0 swojego basysa by zsynchronizować ze sobą wyświetlacz VGA. Gra polega na rywalizacji graczy, który w krótszym czasie osiągnie punkt końcowy (oznaczony jako złota korona na ostatnim poziomie). Rozgrywka skupia się na skakaniu po platformach wygenerowanych w module draw\_background.sv. Skakanie oraz poruszanie się na boki jest realizowane za pomocą klawiatury (klawisze a,d,spacja) podłączonej do płytki Basys3 poprzez USB, interfejsowanej za pomocą protokołu PS2. Główną mechaniką jest zwiększanie się siły skoku proporcjonalnie do czasu, w którym przytrzymujemy spację. Należy odpowiednio dobrać kierunek oraz siłę wyskoku aby przeskoczyć na następne platformy i ominąć przeszkody.

W trakcie gry na ekranie gracza pokazują się również postać sterowana przez przeciwnika. Umożliwia to komunikacja między układami przy użyciu trzech interfejsów UART przechowujących w buforze danych 8 bitów. Dane te odpowiadają za odpowiednią pozycję przeciwnego gracza oraz jego położenie na płaszczyźnie poziomów.

***Z racji na kombinacyjny i nieliniowy charakter rozgrywki schemat blokowy (Rys 1.) jest mocno uproszczony natomiast prawidłowa struktura głównej maszyny stanów została pokazana na diagramie stanów (Rys 2.)***



Rys 1. Uproszczony schemat blokowy gry



Rys 2. Uproszczony Diagram Stanów głównego FSM (jump king ctl sv)

### 3.2. Tabela zdarzeń

Zdarzenie	Kategoria	Reakcja systemu
Reset/Uruchomienie części sprzętowej	Start	Zainicjalizowanie wszystkich rejestrów i pamięci wartościami domyślnymi i wyświetlenie na ekranie początkowego ekranu gry.
Przełączenie przełączników przez oboje graczy	Start	Ekran startowy znika. Pojawia się postać gracza. Gra gotowa do rozpoczęcia
Przytrzymanie przycisku „spacja”	Poruszanie postacią	Postać przechodzi do stanu ładowania skoku. Zwiększana jest wartość zmiennej przechowującej wartość początkową prędkości skoku.
Zwolnienie przycisku „spacja”	Poruszanie postacią	Postać wykonuje skok w górę w ruchu parabolicznym.
Wciśnięcie przycisku „A”	Poruszanie postacią	Podczas trzymania klawisza „A” postać porusza się w lewo ze stałą prędkością.

Wciśnięcie przycisku „D”	Poruszanie postacią	Podczas trzymania klawisza „D” postać porusza się w prawo ze stałą prędkością.
Sygnal collision_bot == 0	Fizyka ruchu horizontalnego	Przy ruchu horyzontalnym postać zaczyna spadać w dół
Sygnal top_reached == 1	Fizyka spadania	Postać zaczyna spadać spadkiem swobodnym.
Sygnal collision_top == 1	Fizyka spadania	Postać odbija się od elementu otoczenia i zaczyna spadać spadkiem swobodnym.
Sygnal colision_bot == 1	Fizyka spadania	Postać zatrzymuje się na napotkanym elemencie otoczenia.
Postać gracza pojawia się w „obszarze korony”	Koniec	Na ekranie wyświetlają się grafiki determinującego gracza wygranego i przegranego.

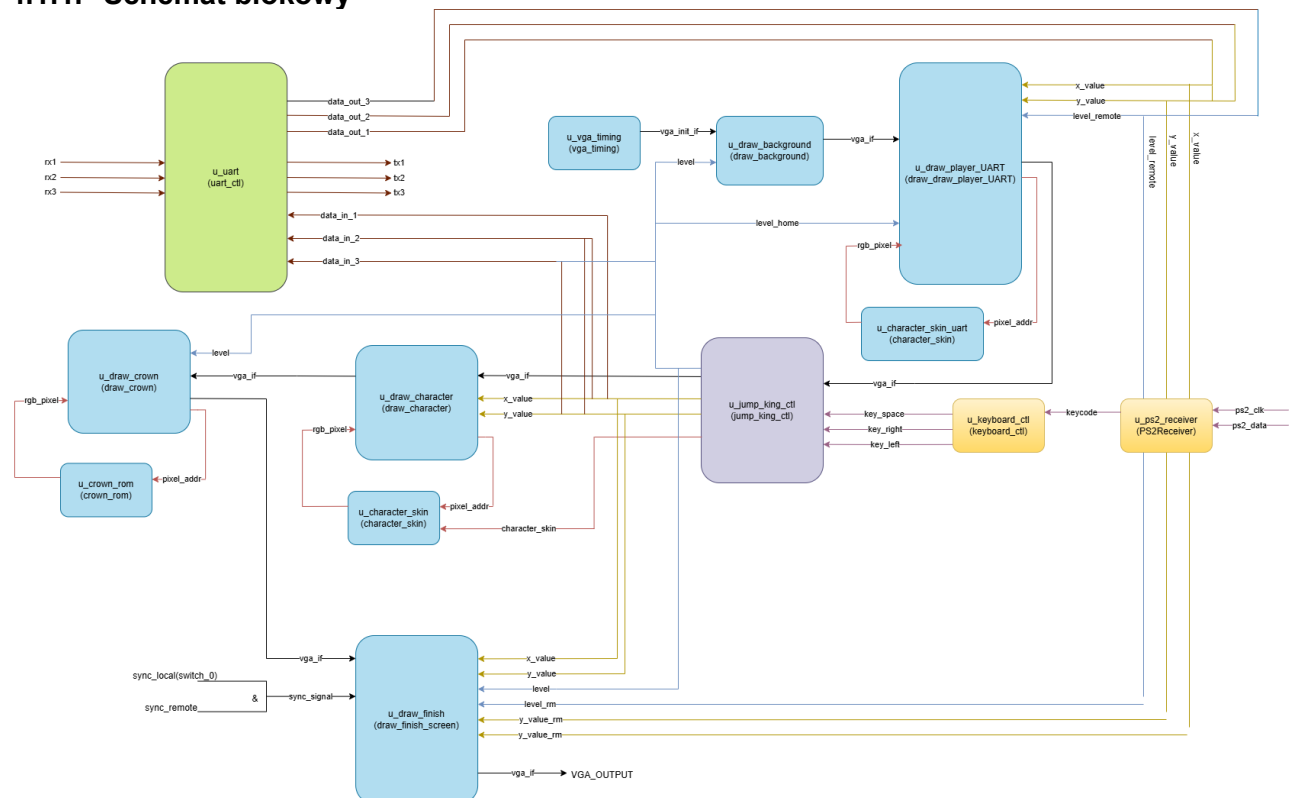
## 4. Architektura

### 4.1. Moduł: top

Osoba odpowiedzialna: MP, TM

Moduł Top składa się z podmodułów połączonych za pomocą zmiennych typu wire realizujących poszczególne funkcjonalności.

#### 4.1.1. Schemat blokowy



Rys 3. Schemat blokowy modułu głównego (top\_jk.sv)

**4.1.2. Porty****a) *keyboard – keyboard\_ctl, input***

<b>nazwa portu</b>	<b>opis</b>
ps2_data	szeregowe wejście danych z klawiatury (PS2)
ps2_clk	zegar klawiatury (PS2)

**b) *vga – vga\_ctl, output***

<b>nazwa portu</b>	<b>opis</b>
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

**c) *uart – uart\_ctl, input***

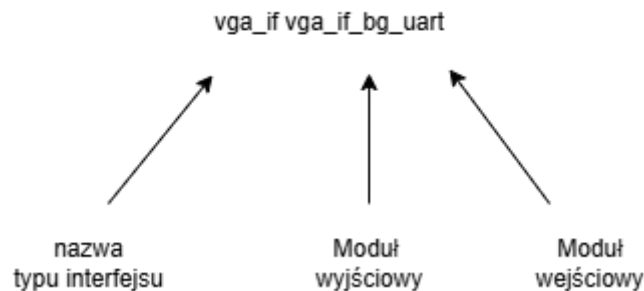
<b>nazwa portu</b>	<b>opis</b>
sync_remote	Sygnał synchronizacyjny odbierany od drugiego gracza
sw0	Sygnał synchronizacyjny uruchamiany za pomocą przełącznika sw0
rx1	Wejściowy sygnał odbiornika UART (kanał 1)
rx2	Wejściowy sygnał odbiornika UART (kanał 2)
rx3	Wejściowy sygnał odbiornika UART (kanał 3)

**d) *uart – uart\_ctl, output***

<b>nazwa portu</b>	<b>opis</b>
sync_remote	Sygnał synchronizacyjny odbierany od drugiego gracza
tx1	Wyjściowy sygnał odbiornika UART (kanał 1)
tx2	Wyjściowy sygnał odbiornika UART (kanał 2)
tx3	Wyjściowy sygnał odbiornika UART (kanał 3)

#### 4.1.3. Interfejsy

Interfejsy w grze pozwalają na przysyłanie sygnałów kalibracyjnych i funkcjonalnych związanych z wyświetlaczem VGA między modułami. By poprawić czytelność i ustrukturyzować kod przyjęta została następująca konwencja nazywania interfejsów:



Rys 3. Przykładowy interfejs VGA używany w grze

#### Lista Interfejsów:

- vga\_if vga\_if\_t\_bg
- vga\_if vga\_if\_bg\_uart
- vga\_if vga\_if\_uart\_ctl
- vga\_if vga\_if\_ctl\_r
- vga\_if vga\_if\_r\_fin
- vga\_if vga\_if\_fin\_out

##### a) ***vga\_if t\_bg – vga timing to draw background***

nazwa sygnału	Opis
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

##### b) ***vga\_if bg\_uart – vga draw background to draw player\_uart***

nazwa sygnału	Opis
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

**c) *vga\_if\_uart\_ctl - draw\_player\_uart to jump\_king\_ctl***

<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

**d) *vga\_if\_ctl\_r - jump\_king\_ctl to draw\_character***

<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

**e) *vga\_r\_crown - draw\_character to draw\_crown***

<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) *vga\_crown\_fin - draw\_crown to draw\_finish\_screen***

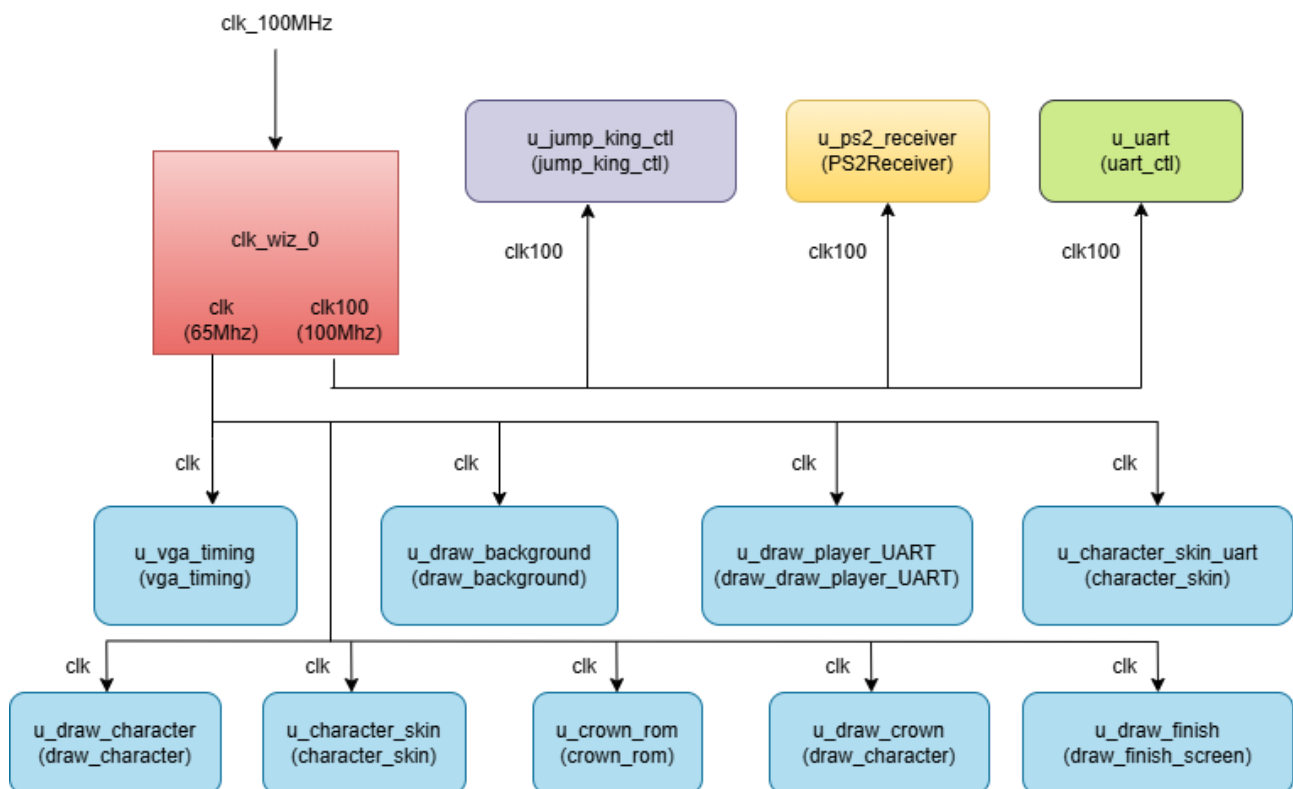
<b>nazwa sygnału</b>	<b>Opis</b>
hcount [10:0]	horyzontalny licznik VGA
vcount [10:0]	wertykalny licznik VGA
hsync	sygnał synchronizacji poziomej VGA
vsync	sygnał synchronizacji pionowej VGA
hblnk	sygnał horyzontalny blank VGA
vblnk	sygnał wertykalny blank VGA
rgb [11:0]	sygnał koloru rgb VGA

**g) vga\_if\_fin\_out– draw\_finish\_screen to output**

nazwa sygnału	Opis
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

**4.2. Rozprowadzenie sygnału zegara**

Osoba odpowiedzialna: MP,TM.





## 5. Implementacja

### 5.1. Lista zignorowanych ostrzeżeń Vivado.

Identyfikator or ostrzeżenie	Liczba wystąpień	Uzasadnienie
38-282	1	W ostrzeżeniu została zawarta informacja o nie spełnieniu przez projekt warunków czasowych. Wynika to ze skomplikowanej logiki używanej w jednym z bloków kombinacyjnych (dzielenie, potęgowanie, mnożenie). Podjęto próby naprawy (np. poprzez „pipeline”), jednak nie przynosiły one zamierzonych efektów, a działanie gry było zaburzone. Podejrzewa się, że na warunki czasowe mogły wpłynąć zegary, nie będące swoimi wielokrotnościami. Ze względu na stopień zaawansowania projektu nie podjęto próby zamiany zegara kontrolującego logikę gry.
35-328	1	Ostrzeżenie wynikające z powyższego.
8-7080	1	Projekt jest zbyt mały, przez co nie spełnia warunków równoległej syntezy

### 5.2. Wykorzystanie zasobów

Resource	Utilization	Available	Utilization %
LUT	11313	20800	54.39
LUTRAM	89	9600	0.93
FF	1059	41600	2.55
BRAM	1.50	50	3.00
DSP	6	90	6.67
IO	28	106	26.42
BUFG	3	32	9.38
MMCM	1	5	20.00

### 5.3. Marginesy czasowe

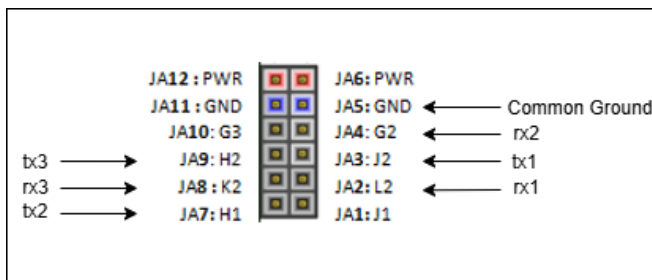
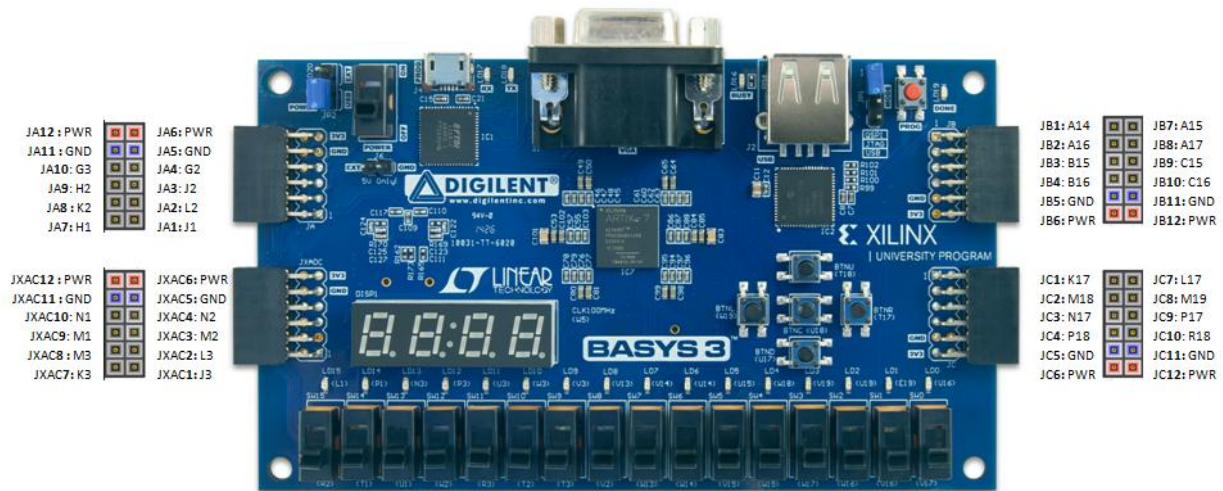
Worst Negative Slack (WNS): **-14.671 ns**

Worst Hold Slack (WHS): 0.032 ns

## 6. Konfiguracja sprzętu

Schemat połączenia ze sobą płytek Basys3 w trybie multiplayer.

**Basys3: Pmod Pin-Out Diagram**



Konfiguracja zwerek i przełączników jest domyślna, natomiast Basys3 został jedynie podłączony do klawiatury poprzez PS2 oraz za pomocą przewodów do drugiej płytki.