Tytuł: Duck Hunt - multiplayer

Autorzy: Łukasz Gąsecki (ŁG), Oliwia Szewczyk (OS)

Ostatnia modyfikacja: 05.06.2025

Spis treści

[1. Repozytorium git 1](#__RefHeading___Toc1311_2642343945)

[2. Wstęp 1](#__RefHeading___Toc375_832888478)

[3. Specyfikacja 1](#__RefHeading___Toc377_832888478)

[3.1. Opis ogólny algorytmu 1](#__RefHeading___Toc379_832888478)

[3.2. Tabela zdarzeń 2](#__RefHeading___Toc381_832888478)

[4. Architektura 2](#__RefHeading___Toc383_832888478)

[4.1. Moduł: top 2](#__RefHeading___Toc385_832888478)

[4.1.1. Schemat blokowy 2](#__RefHeading___Toc387_832888478)

[4.1.2. Porty 3](#__RefHeading___Toc389_832888478)

[a) mou – mouse\_ctl, input 3](#__RefHeading___Toc391_832888478)

[b) vga – vga\_ctl, output 3](#__RefHeading___Toc393_832888478)

[4.1.3. Interfejsy 3](#__RefHeading___Toc453_832888478)

[a) m2c – mouse\_ctl to core 3](#__RefHeading___Toc395_832888478)

[4.2. Rozprowadzenie sygnału zegara 3](#__RefHeading___Toc1150_832888478)

[5. Implementacja 4](#__RefHeading___Toc1313_2642343945)

[5.1. Lista zignorowanych ostrzeżeń Vivado. 4](#__RefHeading___Toc1315_2642343945)

[5.2. Wykorzystanie zasobów 4](#__RefHeading___Toc1317_2642343945)

[5.3. Marginesy czasowe 4](#__RefHeading___Toc1319_2642343945)

[6. Film. 4](#__RefHeading___Toc1249_1436539759)

# Repozytorium git

Adres repozytorium GITa:

<https://github.com/LGasecki/UEC2_MTM_Project_Duck_Hunt.git>

W przypadku repozytorium prywatnego należy zaprosić użytkownika zewnętrznego o adresie mailowym: [kaczmarczyk@agh.edu.pl](mailto:kaczmarczyk@agh.edu.pl)

# Wstęp

*Skąd się wziął pomysł i co w ramach tego projektu robimy.*

# Specyfikacja

## 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.*

## 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** |
| LPM w obszarze napisu START GAME | Ekran startowy | Przejście do stanu czekania na przeciwnika, informacja przeciwnika o gotowości gry |
| LPM w obszarze START GAME przez przeciwnika | Czekanie na start przeciwnika | Rozpoczęcie gry |
| Animacja startowa | Gra | Animacja wchodzącego psa i wskakującego za trawę |
| Losowanie startowej pozycji X celu | Gra | Ustawienie celu na wysokości startowej oraz losowej pozycji wzdłuż linii tej wysokości |
| Cel uderza w ścianę | Gra | Zmiana kierunku lotu celu z losową prędkością wektorową |
| Trafienie celu (LPM) | Gra | Dodanie punktu do naszego wyniku, animacja zabicia celu, odjęcie naboju, animacja psa trzymającego zabity cel |
| Strzelenie poza celem / Pudło (LPM) | Gra | Odjęcie naboju, gra toczy się dalej |
| Przeładowanie magazynku (RPM) | Gra | Załadowanie maksymalną liczbę naboi (3) do magazynku, aktualizacja dostępnych wszystkich naboi |
| Liczba wszystkich naboi = 0 | Gra | Koniec gry, przejście do stanu czekania na koniec gry przeciwnika |
| Przeciwnik skończy grę (liczba jego naboi = 0) | Czekanie na koniec gry przeciwnika | Porównanie wyników, przejście do ekranu końca gry. |
| Wyświetlanie wyniku | Ekran końcowy | Wyświetlanie wyniku graczy wraz z informacją kto wygrał |

# 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 (głównym kryterium powinna być czytelność kodu).

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

## Moduł: top

Osoba odpowiedzialna: ŁG

### Schemat blokowy

Uwaga: Schemat blokowy to nie jest schemat z Vivado! Nie zawiera on sygnałów, tylko interfejsy. Interfejs oznacza tutaj grupę sygnałów. Schemat blokowy pokazuje moduły składowe, oraz łączące je interfejsy.

*Miejsce na 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).*

### Porty

#### mouse – mouse\_ctl, input

|  |  |
| --- | --- |
| **nazwa portu** | **opis** |
| mou\_si | szeregowe wejście danych |
|  |  |

#### vga – vga\_ctl, output

|  |  |
| --- | --- |
| **nazwa portu** | **opis** |
| Vsync | sygnał synchronizacji pionowej VGA |
| Hsync | Sygnał synchronizacji poziomej VGA |
| vgaGreen[3:0] | Sygnał natężenia koloru zielonego VGA |
| vgaBlue[3:0] | Sygnał natężenia koloru niebieskiego VGA |
| vgaRed[3:0] | Sygnał natężenia koloru czerwonego VGA |

### Interfejsy

#### m2c – mouse\_ctl to core

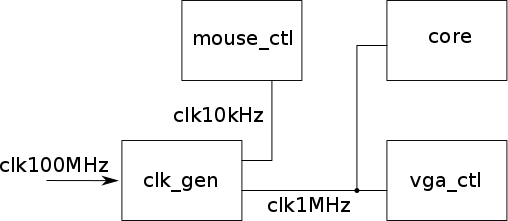
|  |  |
| --- | --- |
| **nazwa sygnału** | **opis** |
|  |  |
|  |  |
|  |  |

## Rozprowadzenie sygnału zegara

Osoba odpowiedzialna: JK

*Informacja 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).*

# Implementacja

## Lista zignorowanych ostrzeżeń Vivado.

|  |  |  |
| --- | --- | --- |
| Identyfikator ostrzeżenia | Liczba wystąpień | Uzasadnienie |
|  |  |  |
|  |  |  |

## Wykorzystanie zasobów

Tabela z wykorzystaniem zasobów z Vivado

## Marginesy czasowe

Marginesy czasowe (WNS) dla setup i hold.

# Konfiguracja sprzętu

Schemat połączenia ze sobą płytek Basys3 w trybie multiplayer.

Obraz zawierający tekst, Czcionka, linia, diagram

Zawartość wygenerowana przez AI może być niepoprawna.

Schematy podłączenia dodatkowych urządzeń peryferyjnych.

Konfiguracja zworek, przełączników, itp., jeśli inna niż domyślna.

# Film.

Link do ściągnięcia filmu:

https://