Wrocław, 02.05.2019

Stanisław Drelich 235902 Jan Kamuda 218202

UCiSW 2 - PROJEKT

Gra Pong

Spis treści

[1. Wprowadzenie 2](#_Toc7473845)

[1.1. Założenia projektowe 2](#_Toc7473846)

[1.2. Użyty sprzęt 2](#_Toc7473847)

[1.3. Moduły wewnętrzne i zaimplementowane 2](#_Toc7473848)

[2. Realizacja 3](#_Toc7473849)

[2.1 Ekrany gry 3](#_Toc7473850)

[2.1.1 Widok planszy podczas regularnej rozgrywki 3](#_Toc7473851)

[2.1.2 Widok planszy po zdobyciu punktu. 3](#_Toc7473852)

[2.1.3 Widok planszy podczas pojawienia się pułapki 4](#_Toc7473853)

[2.1.4 Widok planszy po wygranej. 4](#_Toc7473854)

[2.2 Schemat główny gry 5](#_Toc7473855)

[2.3. Opis modułów 6](#_Toc7473856)

[2.4.1 Grafika 6](#_Toc7473857)

[2.4.2 Mechanika 8](#_Toc7473858)

[2.4.3 Gracze 9](#_Toc7473859)

[2.4.4 KbdDecode 10](#_Toc7473860)

[3. Podsumowanie 11](#_Toc7473861)

# Wprowadzenie

## Założenia projektowe

Celem projektu było zaimplementowanie gry zręcznościowej Pong z wykorzystaniem języka opisu sprzętu VHDL w układzie FPGA.

Zaprojektowana gra powinna dać możliwość przeprowadzenia rozgrywki dla dwóch osób, polegającej na odbijaniu piłki z wykorzystaniem klawiatury. Projekt zakłada również szereg właściwości takich jak zliczanie punktów, zmiana szybkości poruszania się piłki, zmiana rozmiaru i szybkości poruszania się paletki w zależności od postępów gracza, zmiana rozmiaru pola gry w zależności od ogólnej liczby zdobytych punktów, podświetlanie pola gracza tracącego punkt, podświetlanie pola zwycięzcy całej rozgrywki, ustawianie pułapki, resetowanie gry, pauzowanie.

## Użyty sprzęt

W celu umożliwienia rozgrywki wykorzystano następujący sprzęt:

* Zestaw dydaktyczny Spartan 3E – Starter
* Klawiatura
* Monitor LCD

## Moduły Dra Sugiera i zaimplementowane

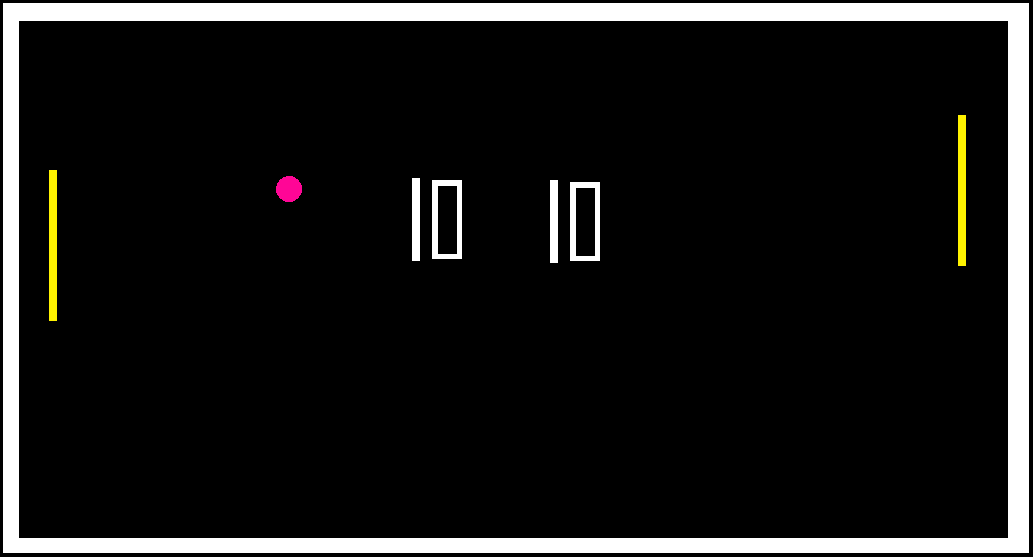
W celu umożliwienia rozgrywki wykorzystano następujące moduły:

* Moduły Dra Sugiera
  + PS2\_Kbd
* Moduły zaimplementowane
  + Vga\_driver
  + Grafika
  + Mechanika
  + Gracze
  + KbdDecode

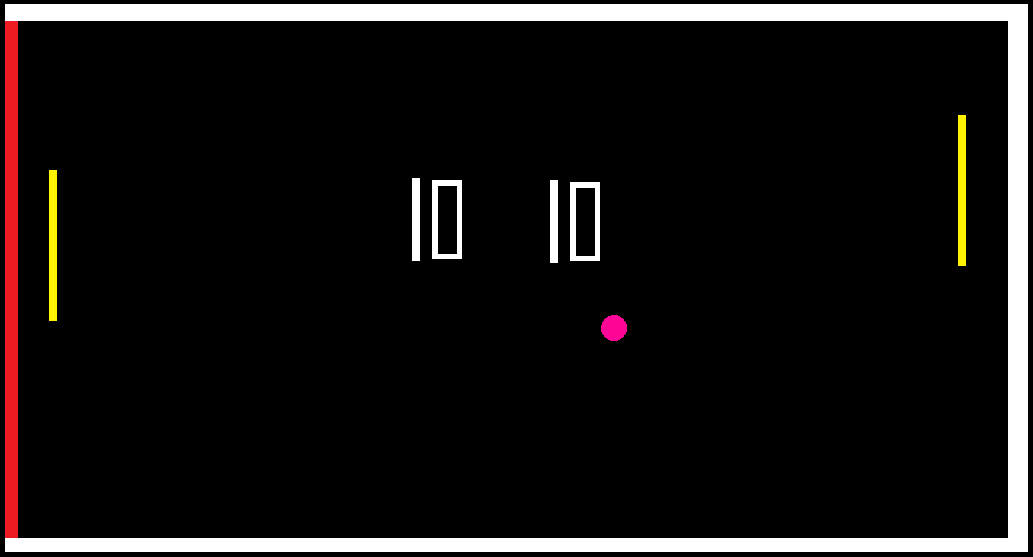
# Realizacja

## 2.1 Ekrany gry

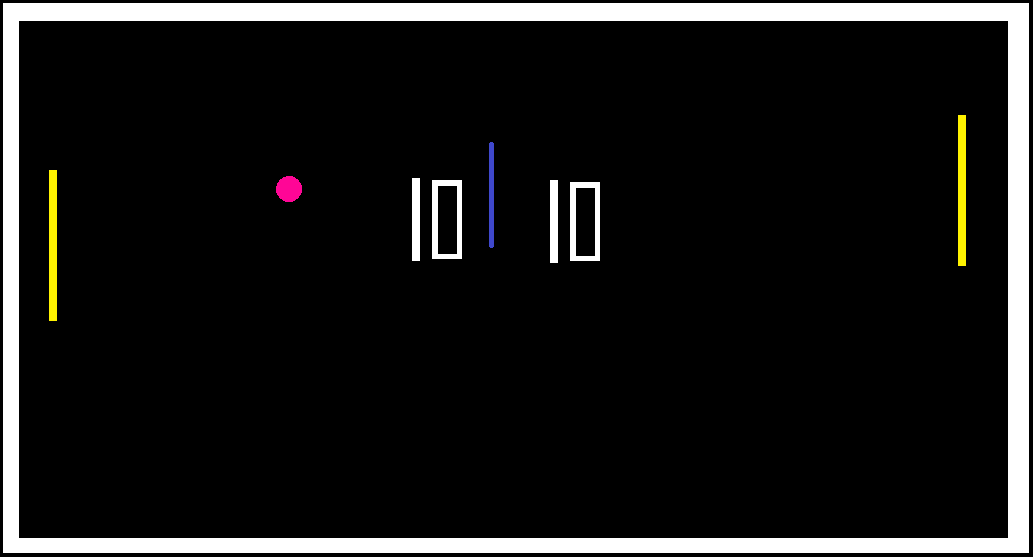
### 2.1.1 Widok planszy podczas regularnej rozgrywki

Rys. 1. Pole planszy podczas regularnej rozgrywki.

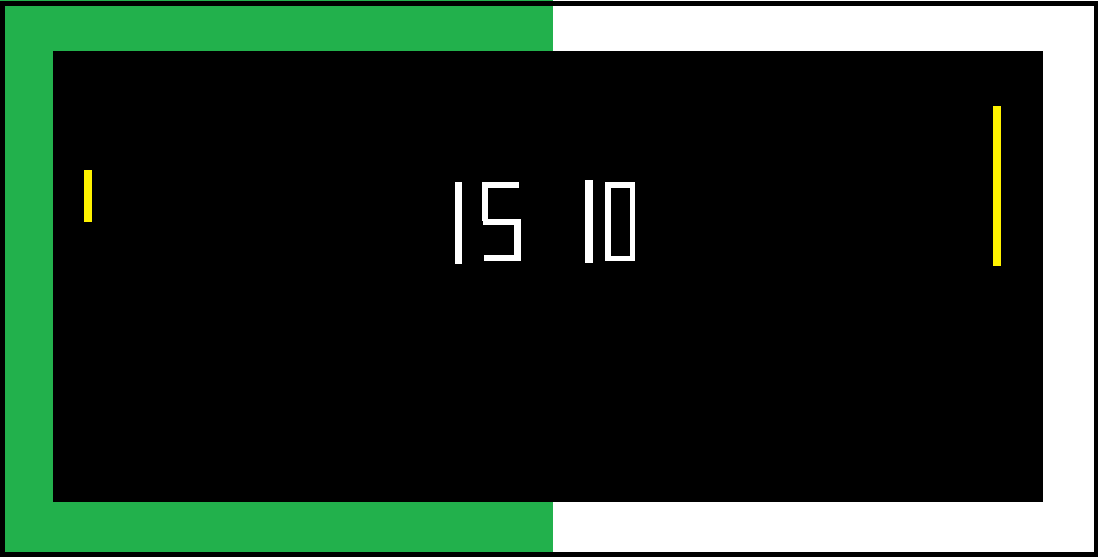
### 2.1.2 Widok planszy po zdobyciu punktu.

Rys. 2. Pole planszy po zdobyciu punktu ( czerwone pole mruga ).

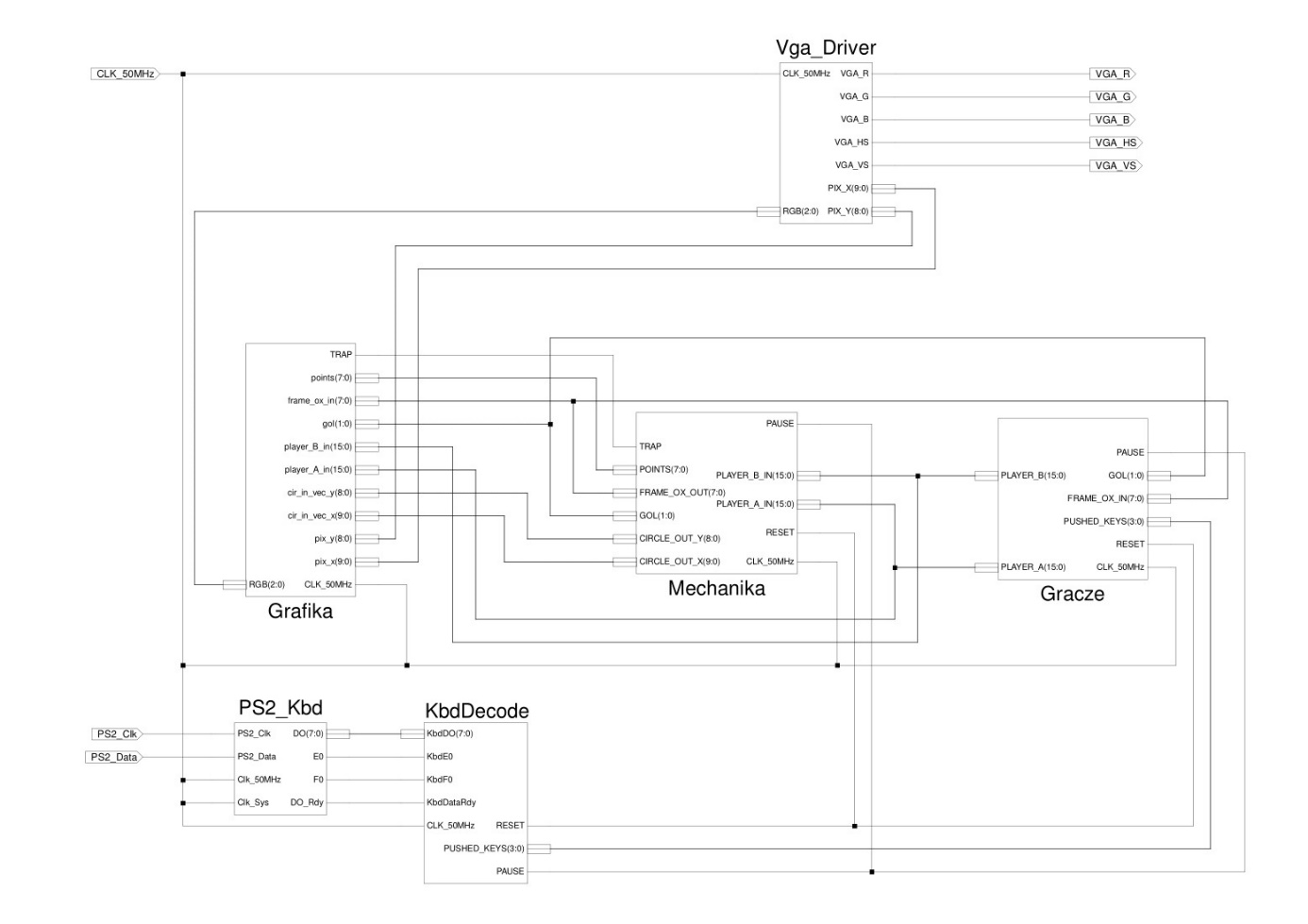
### 2.1.3 Widok planszy podczas pojawienia się pułapki

Rys. 3. Pole planszy podczas pojawienia się pułapki.

### 2.1.4 Widok planszy po wygranej.

Rys. 4. Pole planszy po wygranej ( zielone pole mruga ).

## 2.2 Schemat główny gry



## 2.3. Opis modułów

### 2.4.1 Grafika

Moduł odpowiedzialny za tworzenie i wyświetlanie elementów gry na ekranie tj. :

- tło, na które składa się pole gry oraz ramka podświetlająca się w zależności od zdobytych punktów.

- paletki graczy

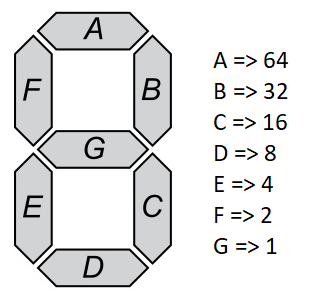
- piłka (realizacja opiera się na wykorzystaniu funkcji obliczającej pierwiastek kwadratowy)

- pułapka ( w zależności od ustalonego prawdopodobieństwa

- punkty (wyświetlanie punktów zbliżone co do zasady działania wyświetlacza 7-segmetnowego)

(Poniżej schemat działania)

Przykład :



Jeżeli chcemy narysować liczbę, sumujemy wartości segmentów zgodnie z powyższym rysunkiem, następnie przekazujemy tę wartość do funkcji DrawSegment. Funkcja ta porównuje przyjętą wartość z kolejnymi potęgami liczby 2 zaczynając od 64 ( 2^liczba segmentów = 64 ), kończąc na 1, z każdym krokiem odejmując wartość w przypadku kiedy okazuje się ona być większa od porównywanego segmentu. Ten sposób pozwala na dokładne zdekodowanie i wyświetlenie właściwych segmentów. W celu rozjaśnienia działania tego sposobu, przedstawię go na przykładzie.

W celu wyświetlenia liczby „4” sumujemy wartości segmentów A, B i C, D i G, co daje nam wartość 51. Następnie przekazujemy tę wartość do funkcji DrawSegmnt. Funkcja zaczyna swoje działanie od porównania tej liczby z 64, która odpowiada za wyświetlanie segmentu A. Okazuje się ona większa i nie zostaje odjęta od 51 co skutkuje nienarysowaniem tego segmentu. Następnie funkcja przechodzi do kolejnej potęgi – 32, która okazuje się mniejsza od 51. W tym momencie a wartość zostaje odjęta od 51, wartość zostaje odjęta (51-32=19), a segment B narysowany. W kolejnych krokach schemat działania wykonuje się w ten sam sposób, co finalnie skutkuje wyświetleniem wszystkich segmentów odpowiedzialnych za narysowanie cyfry „4”.

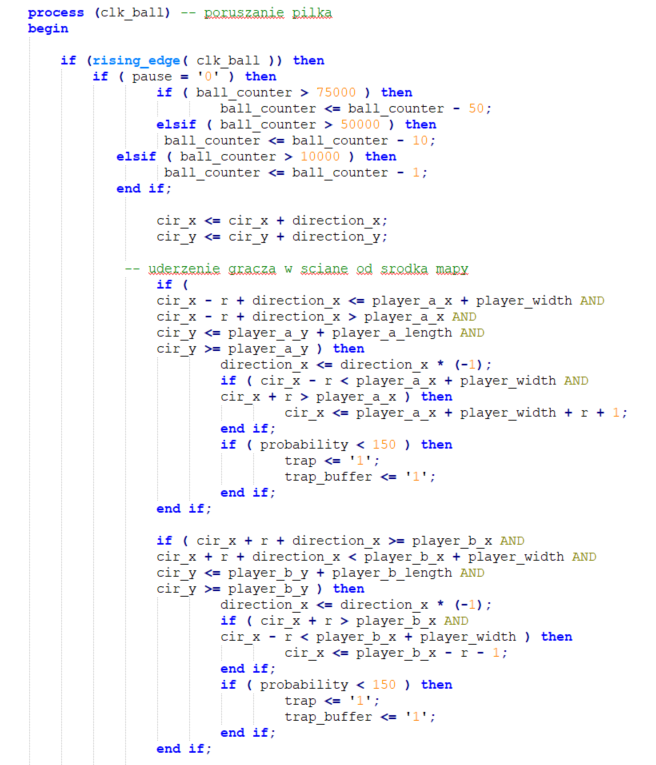


### 

### 2.4.2 Mechanika

Najobszerniejszy moduł, odpowiedzialny za poruszanie się paletek i piłki. W celu jak najdokładniejszej realizacji obejmuje kilkanaście przypadków, w których może znaleźć się piłka tj. dotknięcie ścian, paletek, pułapek.

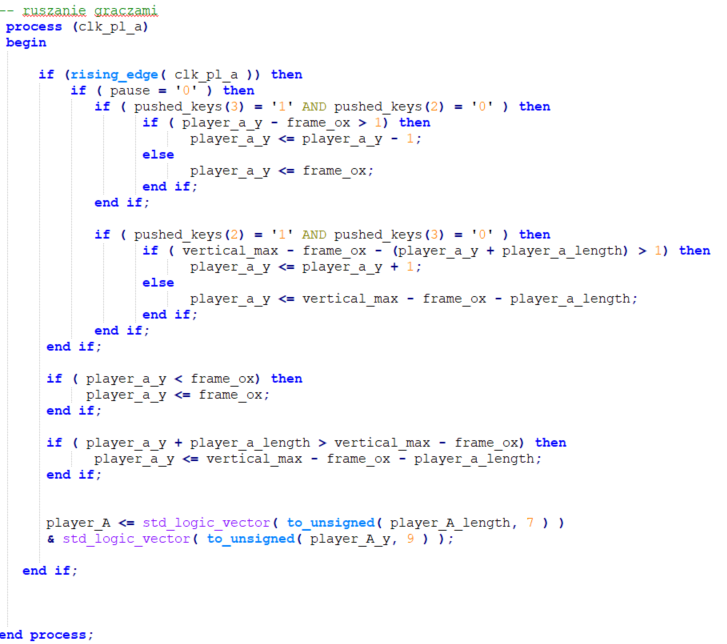
Przykład:



### 2.4.3 Gracze

Moduł odpowiedzialny za parametry gry związane ze sterowaniem paletkami. Obejmuje czynniki tj. wymiar rakietki zmniejszany o 25% oraz prędkość poruszania się jej zmniejszaną o 50% po każdym golu.

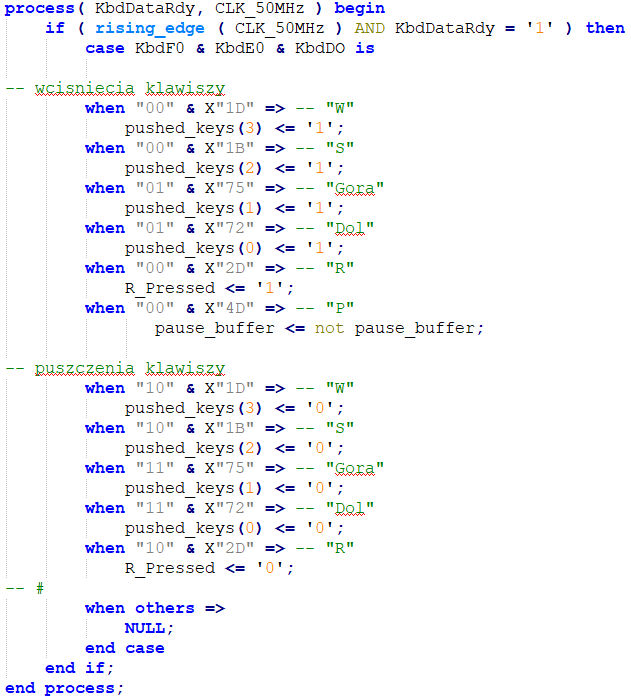
Przykład :



### 2.4.4 KbdDecode

Moduł odpowiedzialny za odczyt i reakcje na naciśnięcia przycisków biorących udział w grze tzn. kierunków góra i dół oraz klawiszy specjalnych tj. pauza oraz reset. Opiera się na odczycie i dekodowaniu kodów ASCII, a następnie na przekazanie ich do modułów odpowiadających za mechanikę.

Przykład :



# Podsumowanie

Realizacja projektu przebiegła pomyślnie i objęła wszystkie zakładane elementy gry, zaczynając od generowania obrazu, poruszanie się paletek i piłki, kończąc na wygenerowaniu pułapki, zmiany trudności gry oraz zliczaniu punktów.

Jedynym istotnym problemem, który wystąpił podczas wykonania zadania były zatrzaski występujące podczas obsługi klawiatury. Wynikały one z błędnej obsługi zdarzeń generowanych przez naciśnięcia klawiszy. Po refaktoryzacji kodu polegającej na dodaniu odczytów wciśnięcia i zwolnienia klawiszy (make code i break code ) problem ustąpił.

W obecnym momencie gra realizuje wszystkie założenia, a testy nie wykazały żadnych nieprawidłowości.