

Dokumentacja projektu - Technika cyfrowa (lab)

Zespół 3

Semestr 4 (letni 2021)

1 Podstawowe informacje o projekcie

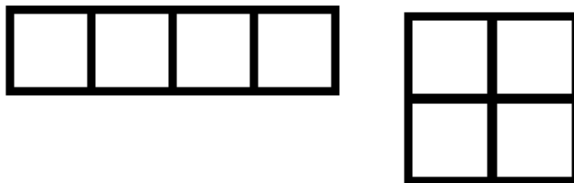
- Realizowany temat:
Realizacja gry "Tetris" w programie Logisim
- Pożądana ocena: 5
- Skład zespołu nr 3:
 - Filip Bochniak 144456
 - Maurycy Kujawski 144452
 - Daniel Różycki 144471
- Podział zadań:
 - Filip
 - * sterowanie klawiaturą
 - Maurycy
 - * całe GPU + logika
 - Daniel
 - * reprezentacja logiczna + logika

2 Funkcjonalności

2.1 Przewidywane

2.1.1 Bazowe

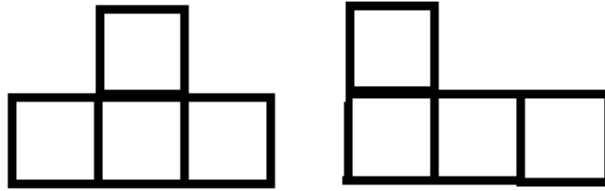
- Ideą projektu jest implementacja gry typu Tetris w środowisku Logisim na ekranie LED 16 na 8 diód. Gra polega na tym, że pojawiają się u góry ekranu klocki. Gracz kontroluje orientację klocka oraz jego pozycję na osi X ekranu. Zadaniem gracza będzie ustawienie klocków na ekranie o szerokości ośmiu klocków taki sposób, aby cały rząd był wypełniony klockami. Wypełniony rząd następnie znika, powodując opadnięcie klocków wyżej, o pozycję niżej. Planujemy zaimplementować w pierwszej kolejności następujące klocki.



Gracz będzie miał możliwość rotacji klocka. Do bazowej funkcjonalności planowana jest również przegrana, która nastąpi w momencie gdy na samej górze ekranu będzie klocek. Gra powinna wyświetlić na ekranie ilość rzędów które zostały wyeliminowane przez gracza, które będą wynikiem jaki osiągnął.

2.1.2 Dodatkowe

- Gdy po zaimplementowaniu funkcjonalności bazowej zostanie trochę czasu do zdania projektu planowana jest implementacja dodatkowych klocków.



Dodatkową funkcją będzie też stopniowe zwiększanie się prędkości opadania klocków co 10 klocków.

2.2 Uzgadnianie

- reprezentacja logiczna

2.3 Zrealizowane

- GPU - czyli transfer pamięci logicznej na ekran

3 Problemy

3.1 Bieżące

3.2 Rozwiązane

- Maurycy: początkowo był problem z zegarem, gdy nieoptymalnie używaliśmy jego sygnału i zmiana wartości pixela w pamięci wymagała 4 taktów aż, rozwiązanie -> zmiana paru rzeczy żeby lepiej działało
- Daniel: Kompletnie zła logika gry, zasugerowałem się GPU i chciałem żeby logika się ustawiała w kolejności iterującej (ponadto pixel po pixelu, a nie jako obiekt), co prowadziło do tak wielu złych i nieoptymalnych rozwiązań że to się w głowie nie mieści
Rozwiązanie: zły pomysł do kosza, jest o wiele lepszy bardziej "obiektyowy" na horyzoncie, ale to nowe rozwiązanie logiki gry skonsultujemy z całą grupą, żeby móc się optymalnie podzielić i uzgodnić wszystko

4 BONUS - opis każdej funkcji (bloku)

- 1IO buffer.circ
podtrzymuje jedynki w słowie 8-bitowym po zadaniu sygnału (włącznie z tymi kolejno dostarczonymi), następnie ponownie podtrzymuje ostatnią wartość, aż do otrzymania sygnału reset
- 8IO buffer.circ
używa 8 blozków 1IO buffer, dzięki czemu rozszerza działanie do 8 słów 8-bitowych
- 7IO tflop.circ
T-Flop siedmiobitowy
- GPU.circ
otrzymuje na wejście słowo 8-bitowe i po otrzymaniu odpowiedniego sygnału ustawia w wewnętrznej pamięci odpowiadającej pixelom na wyświetlaczu w stan pozytywny, a następnie po innym sygnale wyświetla aktualną zawartość pamięci na wyświetlaczu. przycisk reset resetuje całą pamięć na wartości 0
- logika
Cała logika gry

5 DODATKOWE - do zrobienia

- NIEAKTUALNE blok wykrywający czy wartość na pikselem jest oznaczona jako blok ruchomy (wartość 2)
- NIEAKTUALNE blok wykorzystujący informację z powyżej aby ustawić siebie na zapalony pixel (ponadto stan bloku nieruchomego czyli 01)
- NIEAKTUALNE blok iterator który bierze tablicę pamięci przechowującą ledy i iteruje od dolnego prawego do górnego lewego parę pionową pixeli, a na wyjściu daje oba te pixele w celu dalszej analizy