

# UKŁADY CYFROWE I SYSTEMY WBUDOWANE 2

PROJEKT

ORGANY Z POZYTYWKĄ

Maja Bojarska, 241287

Damian Koper, 241292

30 kwietnia 2020

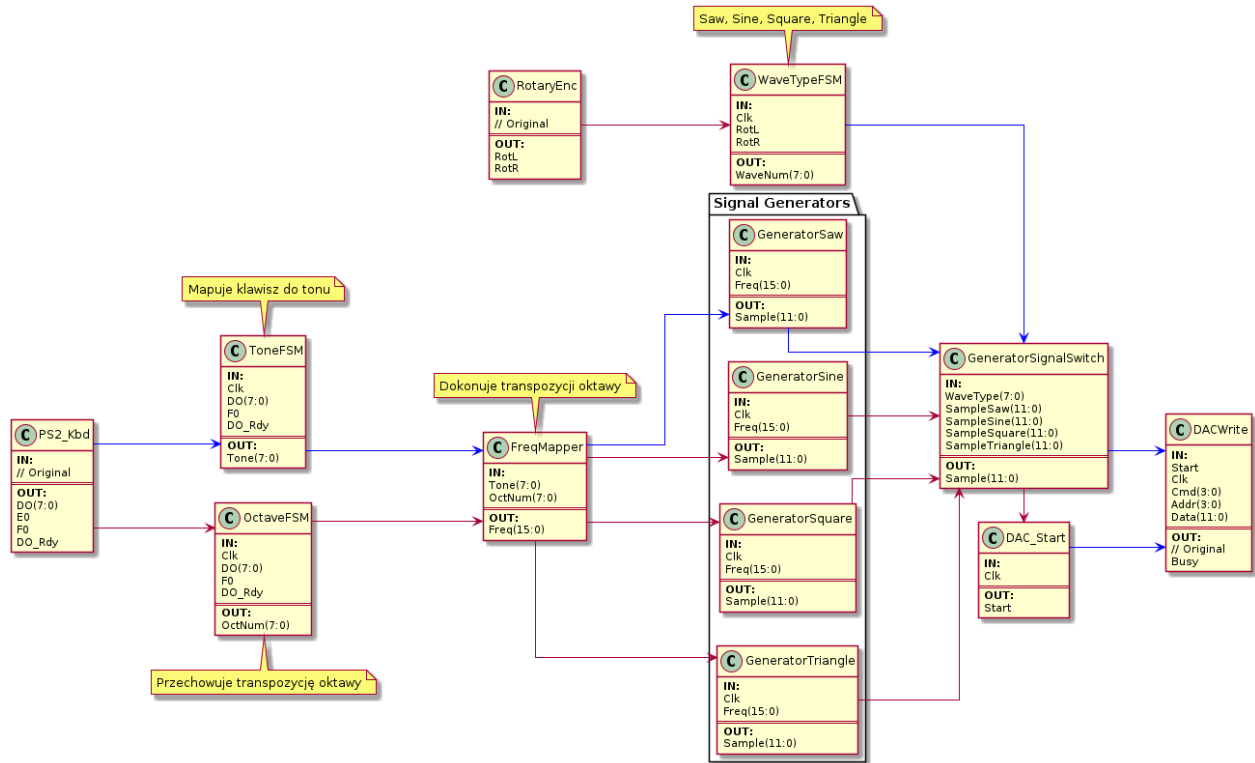
# 1 Cel projektu

Celem projektu było wykonanie układu realizujący działanie organów sterowanych za pomocą klawiszy klawiatury podłączonej z wykorzystaniem interfejsu PS2. Rozszerzeniem działania układu było odtwarzanie sekwencji dźwięków zapisanej w pliku tekstowym odczytanych z karty pamięci SD.

## 1.1 Założenia wstępne

Początkowy projekt układu zakładał podział kolejnych funkcjonalności na możliwie małe moduły według zasady pojedynczej odpowiedzialności. Układ mapował klawisze klawiatury na właściwe tony. Klawisze były przypisane w przybliżeniu zgodnie z układem klawiszy pianina. Klawisze A – J odpowiadały tonom C – H, a klawisze W – E i T – U półtonom kolejno C# – D# i F# – A#. Oktawa zmieniana była za pomocą klawiszy strzałek w zakresie od 0 do 8.

Pierwotny projekt zawierał również podziały na wiele typów fal, których zmiana dokonywana być miała pomocą sygnałów z enkodera. Diagram przepływu danych wstępnego projektu przedstawia rysunek 1.



**Rysunek 1:** Bazowy projekt organów. Niebieskimi strzałkami oznaczono podstawową i wykonaną jako pierwszą funkcjonalność.

## 1.2 Założenia rozszerzone

Rozszerzeniem funkcjonalności organów sterowanych za pomocą klawiatury było uzyskanie możliwości odtwarzania wcześniej zapisanej sekwencji dźwięków o zmiennej długości. Wykorzystano do tego możliwość wczytania danych z karty pamięci i moduł *SDC\_FileReader*. Przykład zapisu dźwięku w pliku tekstowym:

a40000000101001101

gdzie:

$a$  – Ton

4 – Oktawa

0000000101001101 – Czas trwania [ $x * 3ms$ ]

Jeden dźwięk jest definiowany zawsze z wykorzystaniem 18 znaków, więc zapis nie wymaga stosowania separatorów. Czas trwania dźwięku zapisany jest z wykorzystaniem liczby binarnej zapisanej tekstowo na 16 bitach i definiuje czas trwania jako wielokrotność  $3ms$ .

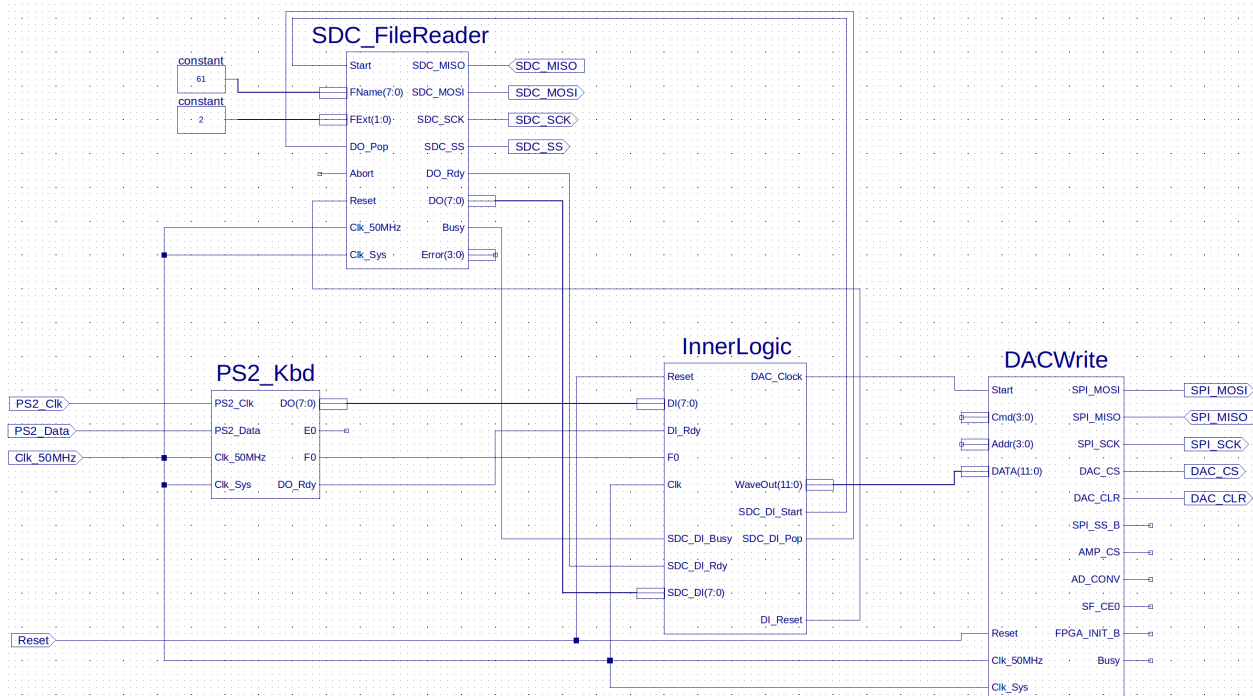
## 2 Struktura układu

### 2.1 Schemat najwyższego poziomu

Schemat najwyższego poziomu przedstawiony na rysunku 2 zawiera wszystkie wewnętrzne moduły odpowiedzialne za komunikację z urządzeniami peryferyjnymi. Są to:

- *SDC\_FileReader*
- *PS2\_Kbd*
- *DACWrite*

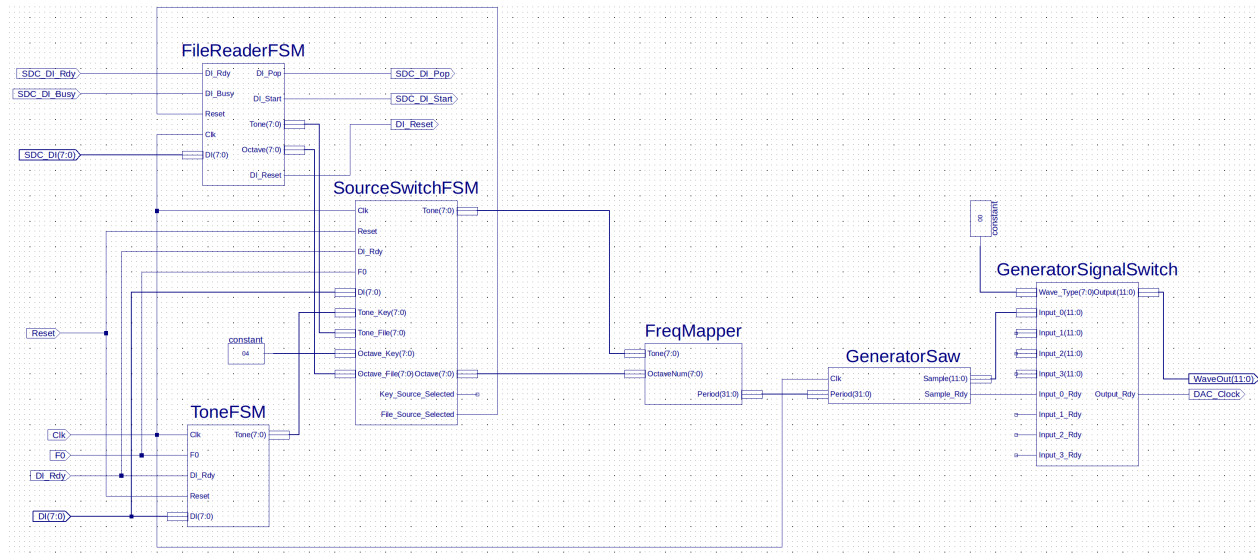
Wszystkie moduły komunikują się z modulem *InnerLogic*.



Rysunek 2: Schemat najwyższego poziomu.

## 2.2 InnerLogic

Schemat InnerLogic przedstawiony na rysunku 3 zawiera wszystkie moduły odpowiedzialne za generowanie sygnału wyjściowego na podstawie danych wejściowych zebranych z klawiatury i karty SD. Przedstawione tutaj moduły mają swoje częściowe odwzorowanie we wstępnym projekcie przedstawionym na rysunku 1.



Rysunek 3: Schemat InnerLogic.

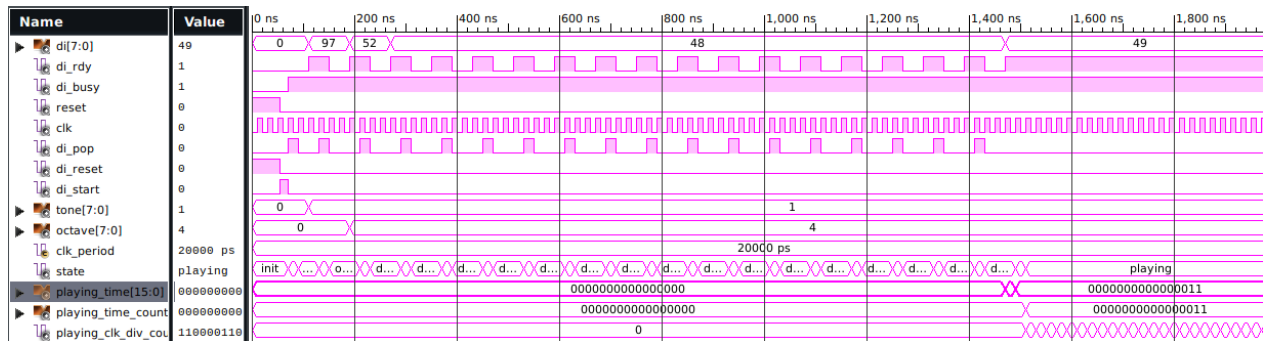
## 2.3 FileReaderFSM

Moduł `FileReaderFSM` realizuje maszynę stanów, która odpowiedzialna jest za interakcje z modułem `SDC_FileReader`. Odpowiada on za dostarczanie numeru tonu i oktawy przez określony czas, gdzie wszystkie te dane odczytywane są z karty SD.

Moduł `SDC_FileReader` umożliwia obsługę odczytu jako obsługę kolejki FIFO. Moduł `FileReaderFSM` w procesie odczytu danych jednego dźwięku odczytuje najpierw znak tonu, potem oktawy, a następnie czas jego trwania wpisując tę wartość do licznika. Po zakończonym odczycie 18 znaków licznik jest uruchamiany, a zmapowane kody tonu i oktawy są widoczne na wyjściu dopóki licznik się nie wyzeruje. Wczytanie tonu 1 i oktawy 4 przedstawia symulacja na rysunku 4.

Ton i oktawa podawane są na wyjście zaraz po ich odczytaniu, a licznik uruchamiany jest

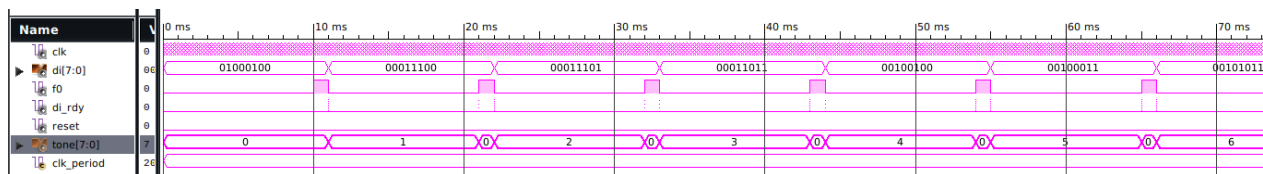
po wczytaniu całego słowa określającego długość dźwięku. Skutkuje to pomijalnie małym wydłużeniem czasu trwania dźwięku.



Rysunek 4: Symulacja modułu FileReaderFSM.

## 2.4 ToneFSM

ToneFSM realizuje maszynę stanów, której stan określa aktualnie odtwarzany ton. Kody od 1 do xd odpowiadają wszystkim tonom i półtonom jednej oktawy. Kod 0 odpowiada ciszy, czyli stanowi, kiedy żaden przycisk nie jest wciśnięty. Stan maszyny zmieniany jest w momencie naciśnięcia lub puszczenia przycisku na klawiaturze i stan ten następnie jest mapowany na odpowiedni kod tonu. Kolejne wciśnięcia przycisków (a, w, s, e, d, r, f) przedstawia symulacja na rysunku 5.



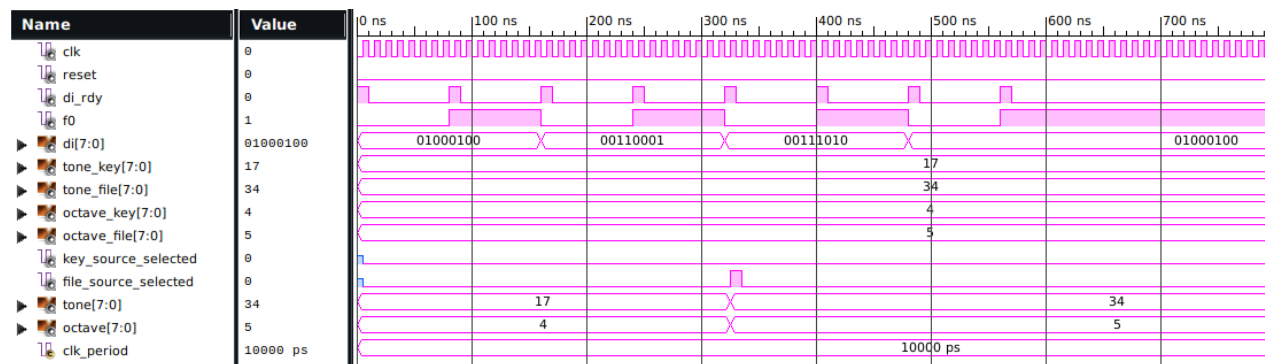
Rysunek 5: Symulacja modułu ToneFSM.

## 2.5 OctaveFSM

xd

## 2.6 SourceSwitchFSM

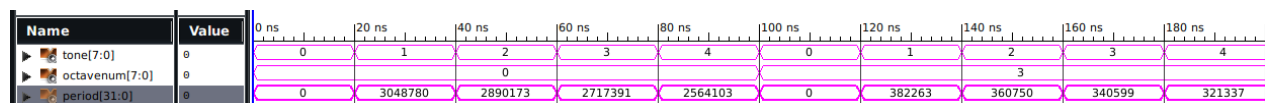
Moduł SourceSwitchFSM odpowiedzialny jest za wybór źródła dźwięku i za restartowanie odczytu dźwięków z karty pamięci w przypadku wyboru tego źródła. Klawisz M zmienia źródło na klawiaturę, a klawisz N na kartę pamięci. Na symulacji z rysunku 6 widać zmianę domyślnego źródła klawiatury na kartę pamięci i wysłanie impulsu wystąpienia zdarzenia zmiany źródła. Zdarzenie to obsługiwane jest przez moduł *FileReaderFSM*, który restartuje proces odczytu.



Rysunek 6: Symulacja modułu SourceSwitchFSM.

## 2.7 FreqMapper

FreqMapper obsługuje proces mapowania oktawy i tonu na liczbę cykli zegara o częstotliwości 50MHz, która odpowiada okresowi fali danego dźwięku. Ton 0 mapowany jest zawsze na wartość 0, co w dalszym procesie generowania sygnału oznacza ciszę. Symulację dla oktawy 0 i 3 przedstawia rysunek 7.



Rysunek 7: Symulacja modułu FreqMapper.

## 2.8 GeneratorSaw

xd

## 2.9 GeneratorSignalSwitch

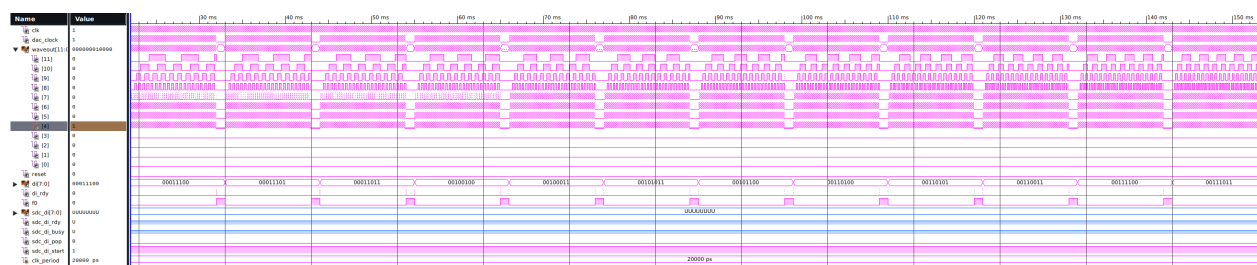
xd

### 3 Symulacja InnerLogic

Oba warianty wejść zostały odpowiednio przetestowane w symulacji. Rysunki 8 i 9 przedstawiają symulacje działania modułu InnerLogic i falę generowaną przez ten moduł.

### 3.1 Wejście z klawiatury

Rysunek 8 przedstawia symulację działania modułu InnerLogic i falę generowaną przez ten moduł. Symulacja obejmuje odtworzenie wszystkich tonów poprzez wciśnięcie odpowiadających im klawiszy. Przeplatane jest to chwilą ciszy, co widać na symulacji.



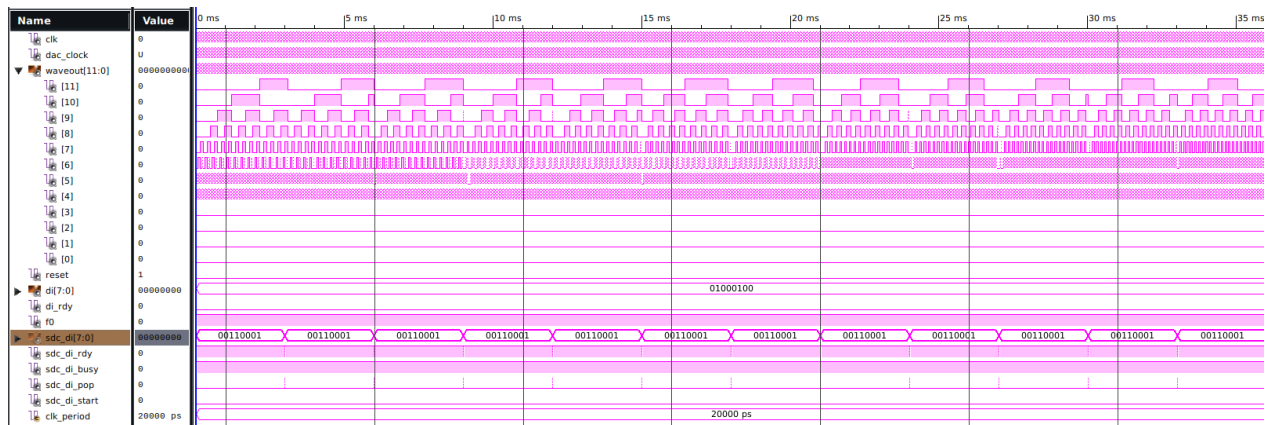
**Rysunek 8:** Symulacja modułu InnerLogic. Wejście z klawiatury.

### 3.2 Wejście z karty pamięci

Symulacja obejmuje odtworzenie melodii zdefiniowanej w pliku na karcie pamięci. Dźwięki opisane są następującym ciągiem:

[illegible]





Rysunek 9: Symulacja modułu InnerLogic. Wejście z karty pamięci.

## 4 Analiza statyczna

## 5 Podsumowanie

### 5.1 Analiza czasów

Narzędzie ISE w wygenerowanym raporcie z procesu implementacji zapewnił, że wymagania czasowe związane z częstotliwością taktowania zegara 50MHz zostaną spełnione.

```

1 Timing summary:
2 -----
3 Timing errors: 0   Score: 0   (Setup/Max: 0, Hold: 0)
4 Constraints cover 5434354 paths, 0 nets, and 7551 connections
5 Design statistics:
6   Minimum period: 17.917ns{1}   (Maximum frequency: 55.813MHz)

```

### 5.2 Zrealizowane założenia

Działanie poparte poprawnymi efektami symulacji pozwala sądzić, iż projekt został wykonany poprawnie zgodnie ze wstępnymi założeniami. Zrezygnowano jednak z pozostałych generatorów typów fal, pozostając tylko przy fali piłokształtnej, ponieważ proces generowania fali w pozostałych generatorach odbywałby się podobnie. Czy to poprzez użycie innego zachowania liczników, czy to poprzez podawanie na wyjście stabilizowanych wartości.