

UKŁADY CYFROWE I SYSTEMY WBUDOWANE 2

PROJEKT

ORGANY Z POZYTYWKĄ

Maja Bojarska, 241287

Damian Koper, 241292

4 maja 2020

1 Cel projektu

Celem projektu było wykonanie układu realizującego działanie organów, sterowanych za pomocą klawiszy klawiatury, podłączonej poprzez interfejs PS2. Rozszerzeniem działania układu było odtwarzanie sekwencji dźwięków odczytanej z pliku tekstowego, zapisanego na karcie pamięci typu SD.

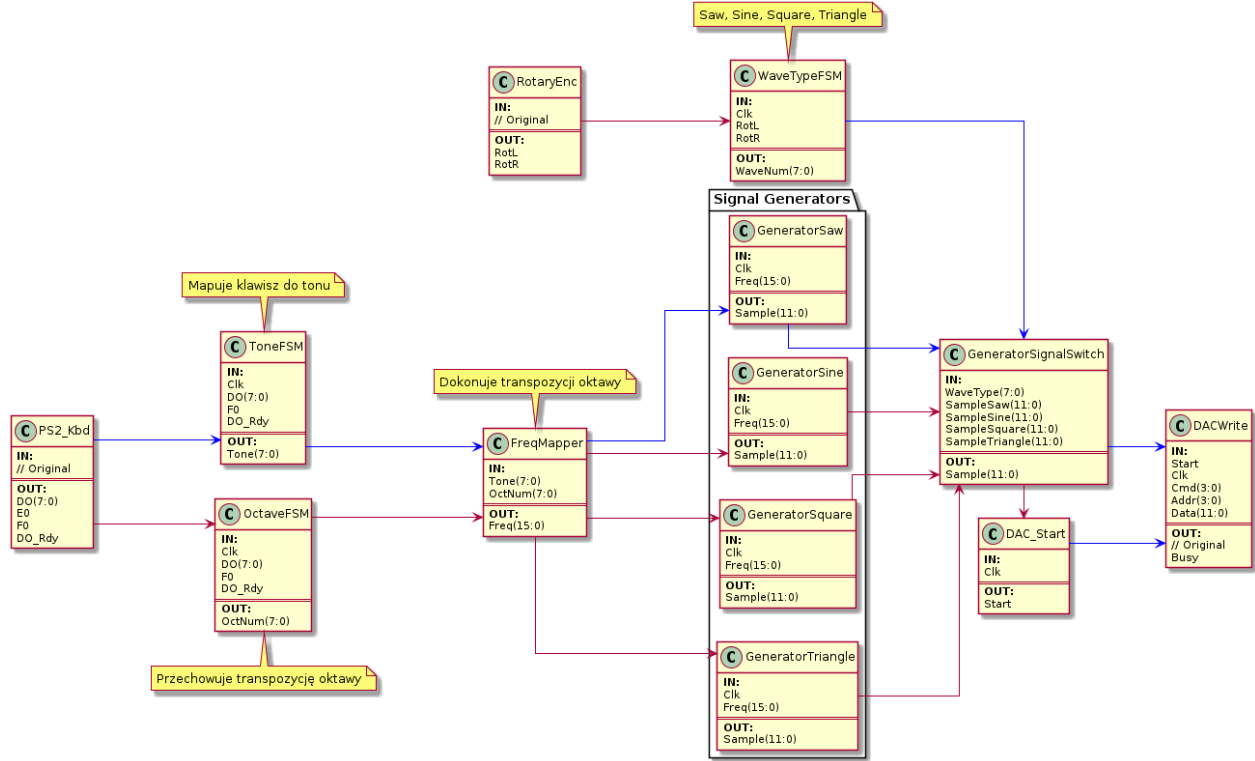
1.1 Założenia wstępne

Początkowy projekt układu zakładał podział kolejnych funkcjonalności na możliwie małe moduły, według zasady pojedynczej odpowiedzialności. Układ mapował klawisze klawiatury na odpowiadające im dźwięki. Klawisze były przypisane do dźwięków, zgodnie z układem przybliżonym do klawiszy jednej oktawy pianina. Oktawa zmieniana była za pomocą klawiszy strzałek w zakresie od 0 do 8.

~	!	@	#	\$	%	^	&	*	()	-	=	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	
	C#	D#	F#	G#	A#								
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	Enter	
	C	D	E	F	G	A	H			:	'		
Shift	Z	X	C	V	B	N	M	<	>	?		Shift	
								,	.	/			
Ctrl	Win	Alt								Alt	Win	Menu	Ctrl

Rysunek 1: Przypisanie tonów oktawy do klawiszy klawiatury QWERTY. Poniżej każdej litery klawisza znajduje się odpowiadający mu ton.

Pierwotny projekt zakładał również podział na wiele rodzajów fal. Rodzaj fali miał być wybierany poprzez obracanie enkodera cyfrowego w lewo (poprzedni) lub prawo (następny). Diagram przepływu danych wstępnej wersji projektu, został przedstawiony na rysunku 2.



Rysunek 2: Wstępny 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 – Klawisz na klawiaturze *QWERTY*

4 – Oktawa

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

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

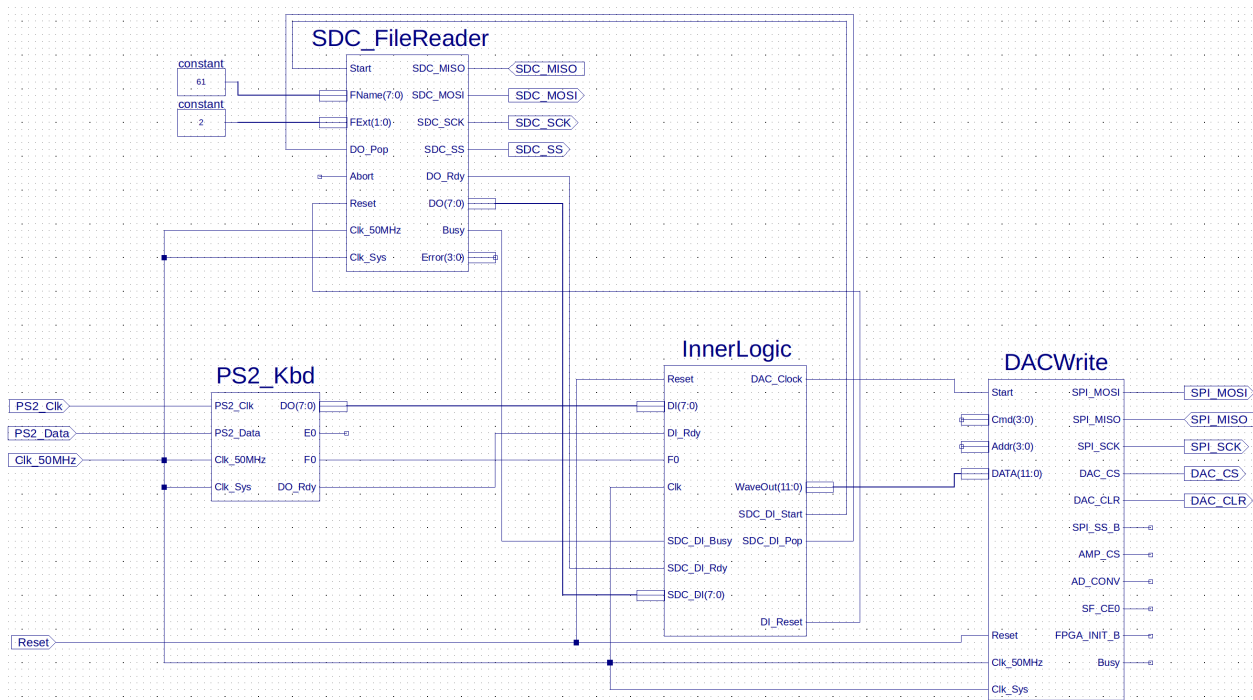
2 Struktura układu

2.1 Schemat najwyższego poziomu

Schemat najwyższego poziomu, przedstawiony na rysunku 3, zawiera wszystkie zewnętrzne moduły odpowiedzialne za komunikację z urządzeniami peryferyjnymi. Są to:

- *SDC_FileReader*
- *PS2_Kbd*
- *DACWrite*

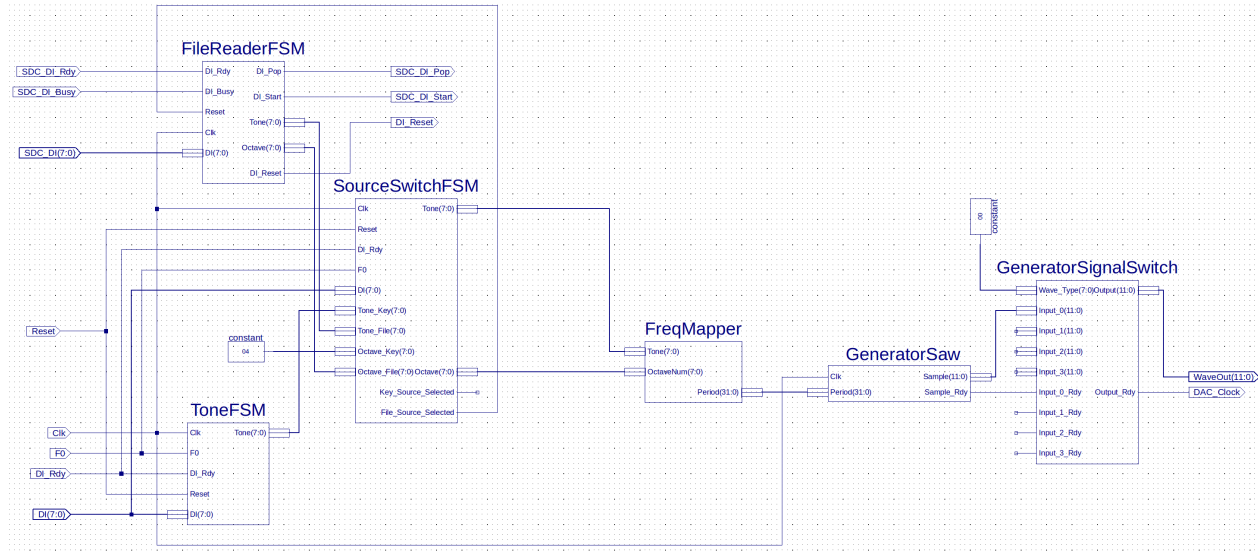
Wszystkie moduły komunikują się z modulem *InnerLogic*.



Rysunek 3: Schemat najwyższego poziomu.

2.2 InnerLogic

Schemat InnerLogic, przedstawiony na rysunku 4, 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 2.



Rysunek 4: Schemat InnerLogic.

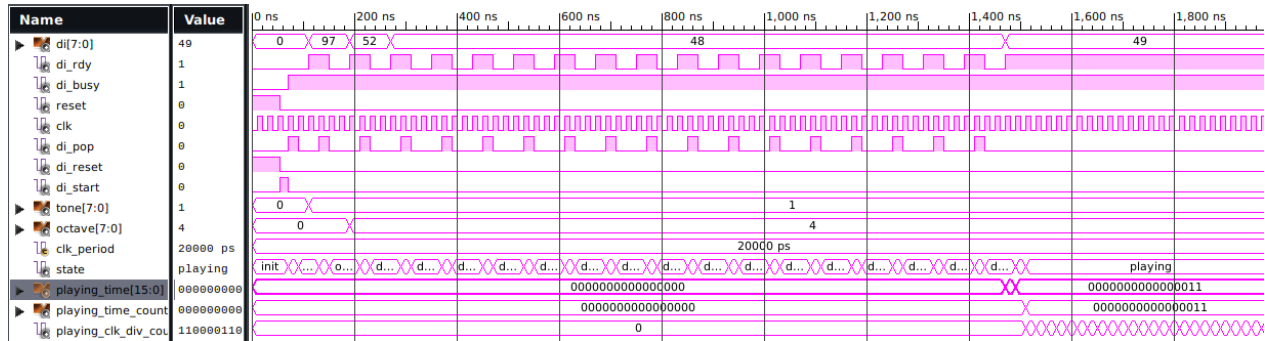
2.3 FileReaderFSM

Moduł `FileReaderFSM` realizuje maszynę stanów, która jest odpowiedzialna za interakcję 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 odczyt wartości z pliku podobnie do kolejki FIFO. Moduł `FileReaderFSM` w procesie odczytu danych jednego dźwięku najpierw odczytuje znak tonu, potem oktawy, a następnie czas jego trwania, wpisując tę wartość do licznika. Po zakończonym odczycie 18 znaków, licznik jest uruchamiany, a wartości zmapowanych kodów tonu i oktawy są obecne na wyjściach modułu, dopóki licznik się nie wyzeruje. Wczytanie tonu 1 i oktawy 4 przedstawia symulacja na rysunku 5.

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

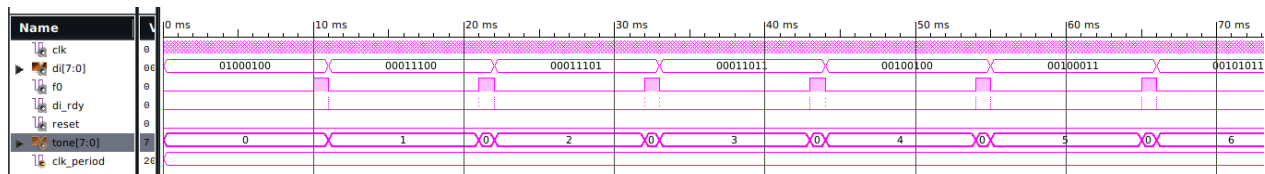
po wczytaniu całego słowa określającego długość dźwięku. Skutkuje to pomijalnie małym wydłużeniem czasu trwania dźwięku.



Rysunek 5: Symulacja modułu FileReaderFSM.

2.4 ToneFSM

ToneFSM realizuje maszynę stanów, której stan określa aktualnie odtwarzany ton. Kody od 1 do 12 odpowiadają wszystkim tonom 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 jest następnie mapowany na odpowiedni kod tonu. Kolejne wciśnięcia przycisków (A, W, S, E, D, R, F) przedstawia symulacja na rysunku 6.



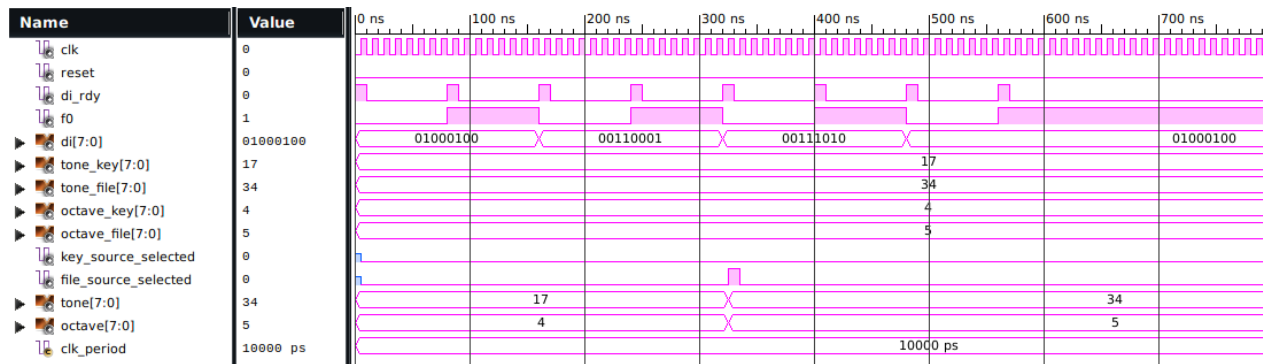
Rysunek 6: Symulacja modułu ToneFSM.

2.5 OctaveFSM

xd

2.6 SourceSwitchFSM

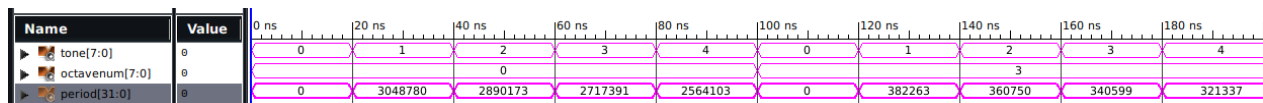
Moduł SourceSwitchFSM odpowiedzialny jest za wybór źródła dźwięku i za restartowanie odczytu dźwięków z karty pamięci w przypadku wyboru tego źródła. Klawisz M zmienia źródło na klawiaturę, a klawisz N na kartę pamięci. Na symulacji z rysunku 7 widać zmianę domyślnego źródła klawiatury na kartę pamięci i wysłanie impulsu wystąpienia zdarzenia zmiany źródła. Zdarzenie to obsługiwane jest przez moduł *FileReaderFSM*, który restartuje proces odczytu.



Rysunek 7: 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 oraz 3 przedstawia rysunek 8.



Rysunek 8: Symulacja modułu FreqMapper.

2.8 GeneratorSaw

xd

2.9 GeneratorSignalSwitch

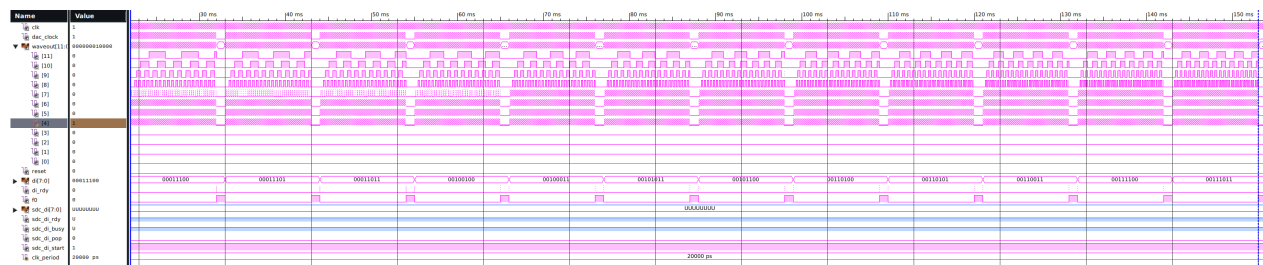
xd

3 Symulacja InnerLogic

Oba warianty wejść zostały odpowiednio przetestowane w symulacji. Rysunki 9 i 10 przedstawiają symulacje działania modułu InnerLogic i falę generowaną przez ten moduł.

3.1 Wejście z klawiatury

Rysunek 9 przedstawia symulację działania modułu InnerLogic i falę generowaną przez ten moduł. Symulacja obejmuje odtworzenie wszystkich tonów w jednej oktawie, poprzez wcisnięcie odpowiadających im klawiszy. Przeplatane jest to chwilą ciszy, co widać na symulacji.

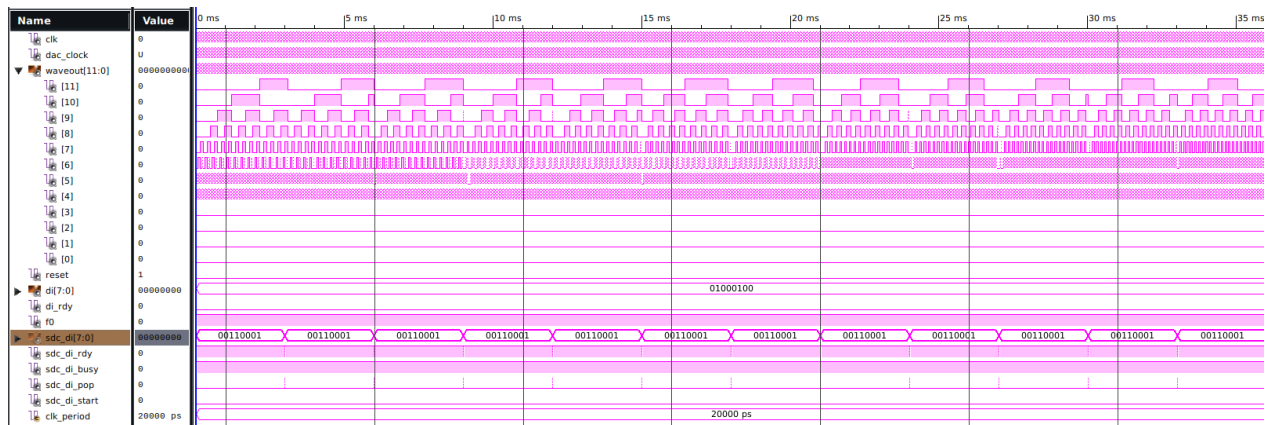


Rysunek 9: Symulacja modułu InnerLogic. Wejście z klawiatury.

3.2 Wejście z karty pamięci

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

```
1 a400000000000000000001w40000000000000000001s40000000000000000001  
    ↳ e400000000000000000001d400000000000000000001f40000000000000000001  
    ↳ t4000000000000000000001g400000000000000000001y400000000000000000001  
    ↳ h40000000000000000000001u4000000000000000000001j400000000000000000001
```

Rysunek 10: Symulacja modułu InnerLogic. Wejście z karty pamięci.

4 Podsumowanie

4.1 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)

```

4.2 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. Proces generowania fali w pozostałych generatorach odbywałby się podobnie, poprzez użycie innego zachowania liczników lub podawanie na wyjście stabilizowanych wartości.