

1. Założenia projektu

Celem projektu było wykonanie układu cyfrowego pozwalającego na modyfikację sygnału fonicznego w czasie rzeczywistym za pomocą algorytmów cyfrowego przetwarzania sygnałów. Urządzenie miałooby znaleźć zastosowanie w modyfikacji brzmienia elektrycznych instrumentów muzycznych.

Podstawowe funkcjonalności, które muszą zostać spełnione przez gotowe urządzenie zostały przedstawione poniżej:

- modelowanie naturalnie występujących zjawisk akustycznych:
 - echo,
 - efekt Dopplera.
- nakładanie efektów nie bazujących na naturalnych zjawiskach akustycznych:
 - zmiana częstotliwości próbkowania,
 - wprowadzanie nowych harmonicznych.
- uniezależnienie pracy urządzenia od źródła sygnału,
- interfejs użytkownika oparty o aplikację na telefon z systemem android pozwalający na włączanie poszczególnych efektów oraz zmianę ich parametrów,
- zachowanie latencji sygnału poniżej 5 ms.

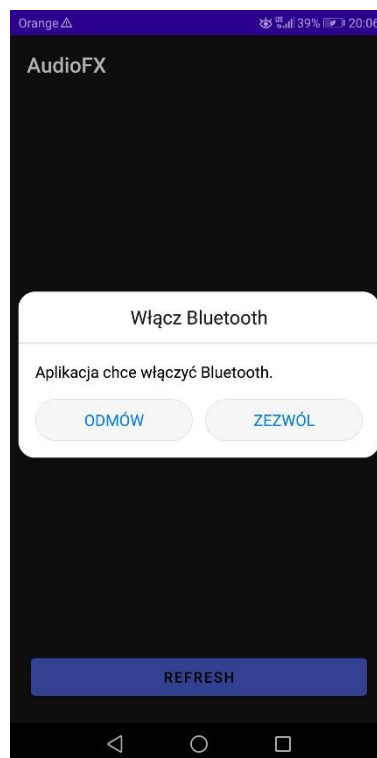
Do realizacji projektu zgodnie z założeniami zostały wybrane następujące narzędzia oraz podzespoły sprzętowe:

- **platforma ewaluacyjna Basys 3 oparta o układ FPGA Xilinx Artix-7 XC7A35T** – realizacja cyfrowego przetwarzania sygnału.
- **środowisko Active HDL** – konfiguracja układu FPGA przy użyciu języka opisu sprzętu VHDL.
- **układy scalone CS5434 i CS4345** – przetworniki ADC oraz DAC do konwersji sygnału z postaci analogowej na cyfrową i odwrotnie. Sygnał cyfrowy jest transmitowany w oparciu o interfejs I2S.
- **układ scalony AD8656** – wzmacniacz operacyjny do kondycjonowania sygnału analogowego.
- **język programowania Python w środowisku PyCharm** – rozwój i testowanie algorytmów przetwarzania sygnału.
- **język programowania Kotlin w środowisku AndroidStudio** – stworzenie aplikacji na urządzenie mobilne umożliwiającej sterowanie urządzeniem.
- **moduł HC-06 ZS-040** oparty o układ scalony BC417 umożliwiający komunikację między telefonem i układem FPGA za pomocą interfejsu Bluetooth.

2. Opis funkcjonalny

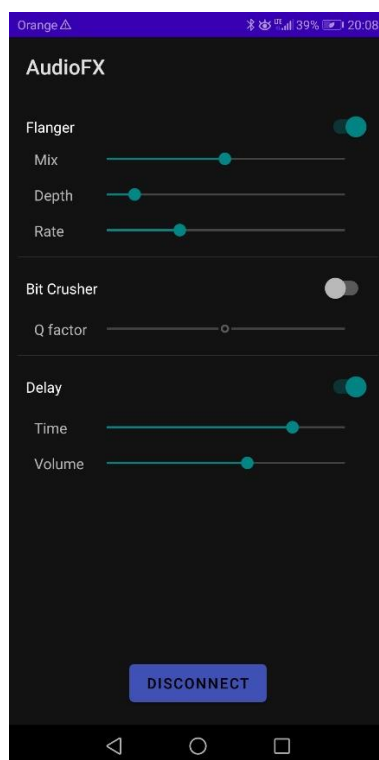
Źródło sygnału oraz odbiornik zostają podłączone do urządzenia za pomocą złącza Jack 6.3 mm., a platforma musi zostać zasilona poprzez gniazdo USB.

Obsługa efektu odbywa się za pomocą aplikacji na urządzenie z systemem Android. Po uruchomieniu aplikacji użytkownik zostaje poproszony o włączenie modułu Bluetooth i połączenie się z urządzeniem (Rysunek 1).



Rysunek 1

Po ustanowieniu połączenia przez interfejs Bluetooth użytkownik może włączać lub wyłączać poszczególne efekty oraz zmieniać ich parametry za pomocą zestawu przełączników i suwaków widocznych na ekranie (Rysunek 2).



Rysunek 2

Ustawienia efektów są automatycznie aktualizowane po wprowadzeniu zmian przez użytkownika. Kolejność efektów w oknie aplikacji odpowiada kolejności w jakiej efekty wprowadzane są w sygnale dźwiękowym. W obecnej wersji zostały wprowadzone następujące efekty:

Flanger – do sygnału dodawana jest jego kopia przesunięta w czasie o niewielki, stopniowo zmieniający się okres. Symulacja działania głośnika Leslie. Użytkownik ma kontrolę nad następującymi parametrami:

Mix - stosunek amplitudy sygnału opóźnionego do amplitudy sygnału oryginalnego.

Depth – zakres zmian czasu opóźnienia.

Rate – częstotliwość zmian czasu opóźnienia.

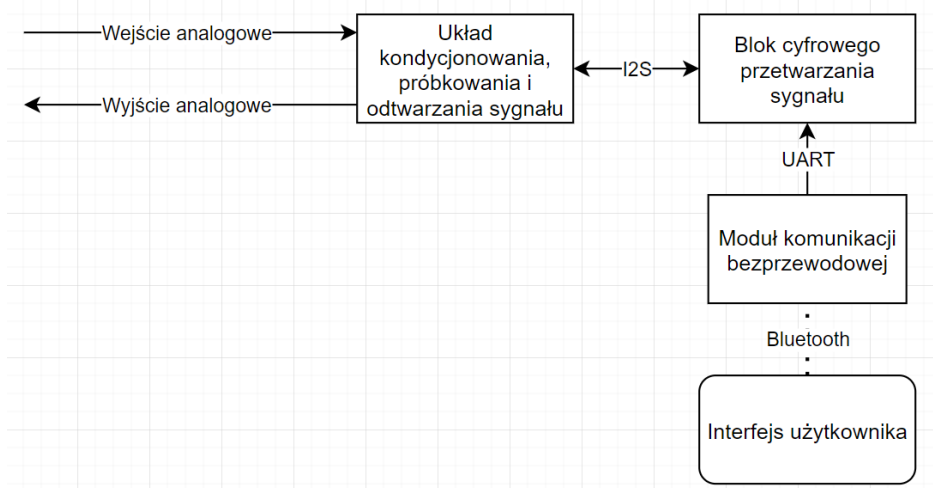
Dla skrajnie dużych wartości parametru *Mix* wprowadzany efekt to Vibrato – okresowa modulacja częstotliwości.

Bit Crusher – ograniczenie częstotliwości zmian chwilowych wartości sygnału – efekt charakterystyczny dla gatunku muzyki chiptune. Parametr *Q factor* określa współczynnik podziału częstotliwości próbkowania.

Delay – symulacja zjawiska echa. Do sygnału dodawana jest jego kopia przesunięta w czasie o czas zależny od parametru *Time*. Głośność echa jest zależna od parametru *Volume*.

3. Schemat ideowy

Schemat ideowy został przedstawiony poniżej:

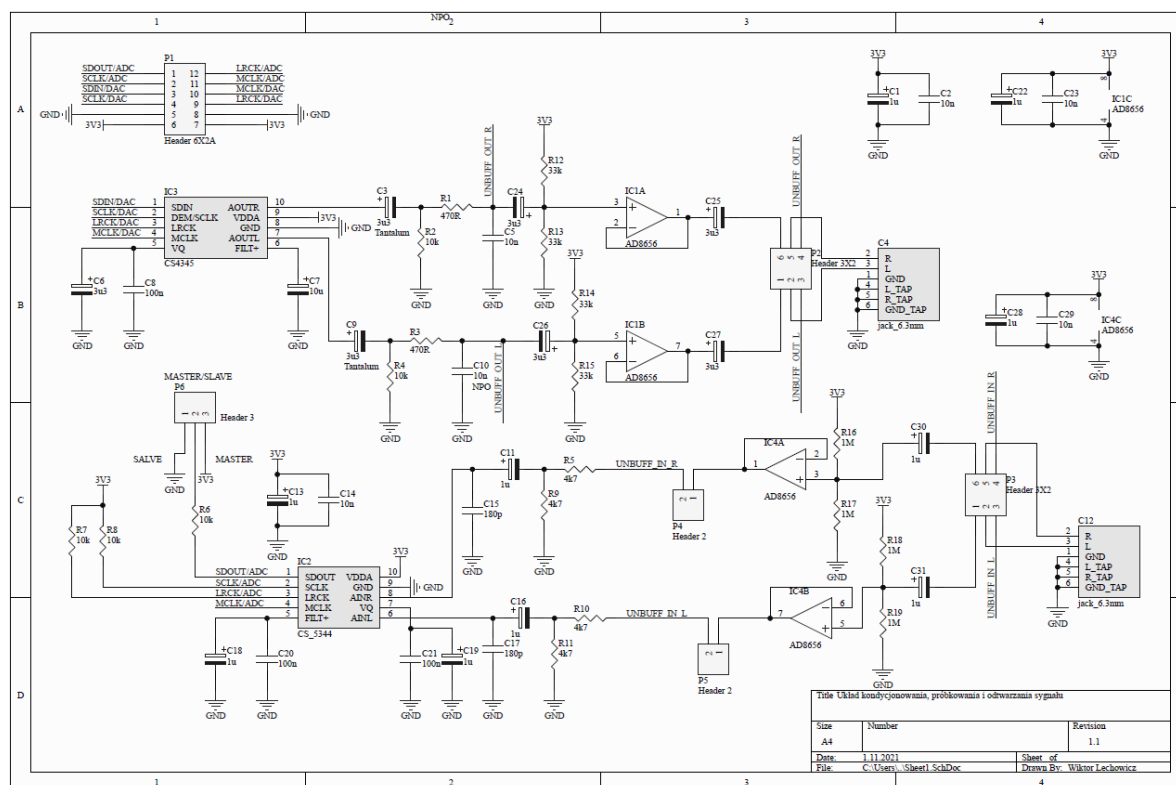


Rysunek 3

Układ kondycjonowania, próbkowania i odtwarzania sygnału jest wykonany w postaci osobnego obwodu drukowanego i zostanie opisany w punkcie 4. Blok cyfrowego przetwarzania sygnału został zrealizowany za pomocą układu FPGA, a jego opis znajduje się w punkcie 5. Do komunikacji bezprzewodowej został użyty gotowy moduł HC-06, natomiast interfejs użytkownika został opisany w punkcie 2.

4. Układ kondycjonowania, próbkowania i odtwarzania sygnału

Do realizacji funkcjonalności tego układu niezbędne było wykonanie zewnętrznego modułu dołączanego do platformy uruchomieniowej z układem FPGA. Schemat elektryczny tego modułu został przedstawiony poniżej:



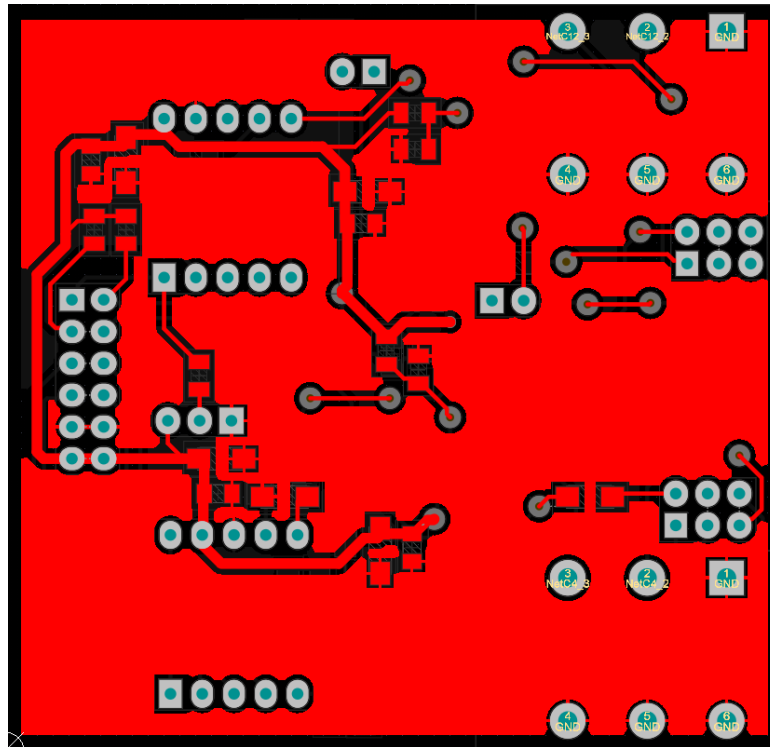
Rysunek 4

Układ jest podłączany do płytki z układem FPGA za pomocą listwy P1. Do poprawnej pracy wymaga zasilenia napięciem stabilizowanym 3.3V. Przetwornik DAC CS5345(IC3) oraz przetwornik ADC CS5344(IC2) komunikują się z układem FPGA za pośrednictwem interfejsu I2S. Elementy pasywne w torach sygnału analogowego pełnią funkcję filtrów środkowoprzepustowych. Wzmacniacze operacyjne AD8656(IC1, IC4) pozwalają na poprawną pracę układu ze źródłem sygnału o impedancji do 100kΩ oraz odtwarzanie sygnału wyjściowego bezpośrednio za pomocą słuchawek. Zworki P2-P5 służą do wyboru trybu pracy MONO/STEREO oraz włączania wtórników napięciowych w tor sygnałowy. Zworka P6 służy do wyboru trybu pracy przetwornika ADC w trybie master lub slave.

Projekt obwodu drukowanego został wykonany z myślą o wykonaniu płytki obwodu drukowanego w warunkach domowych. W kolejnej wersji PCB planuje się rozdzielić masę cyfrową i analogową oraz użyć footprintów umożliwiających bezpośredni montaż przetworników ADC i DAC w obudowach TSSOP-10.

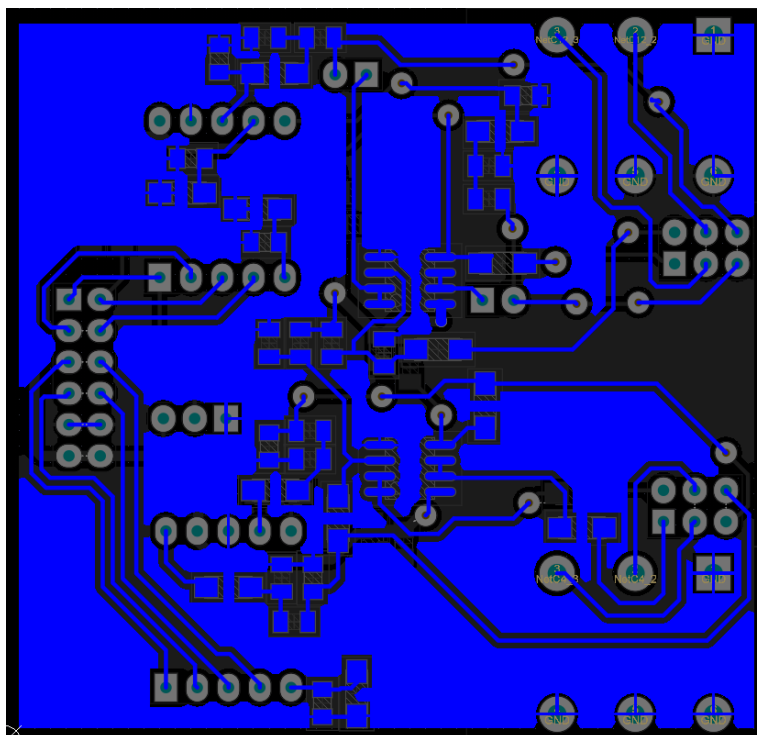
Projekt obwodu drukowanego:

Warstwa Top:



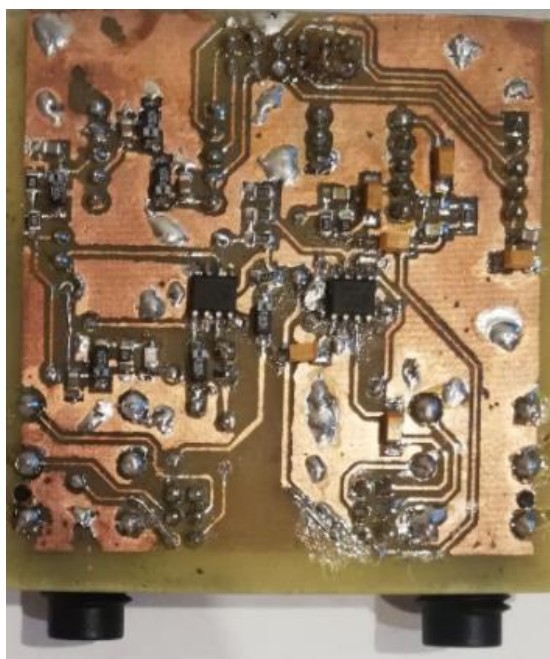
Rysunek 5

Warstwa Bottom:

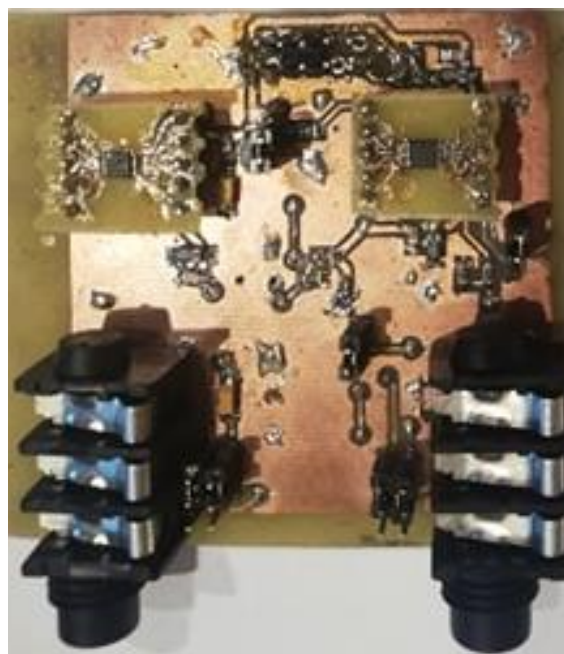


Rysunek 6

Zmontowany prototyp PCB:



Rysunek 7

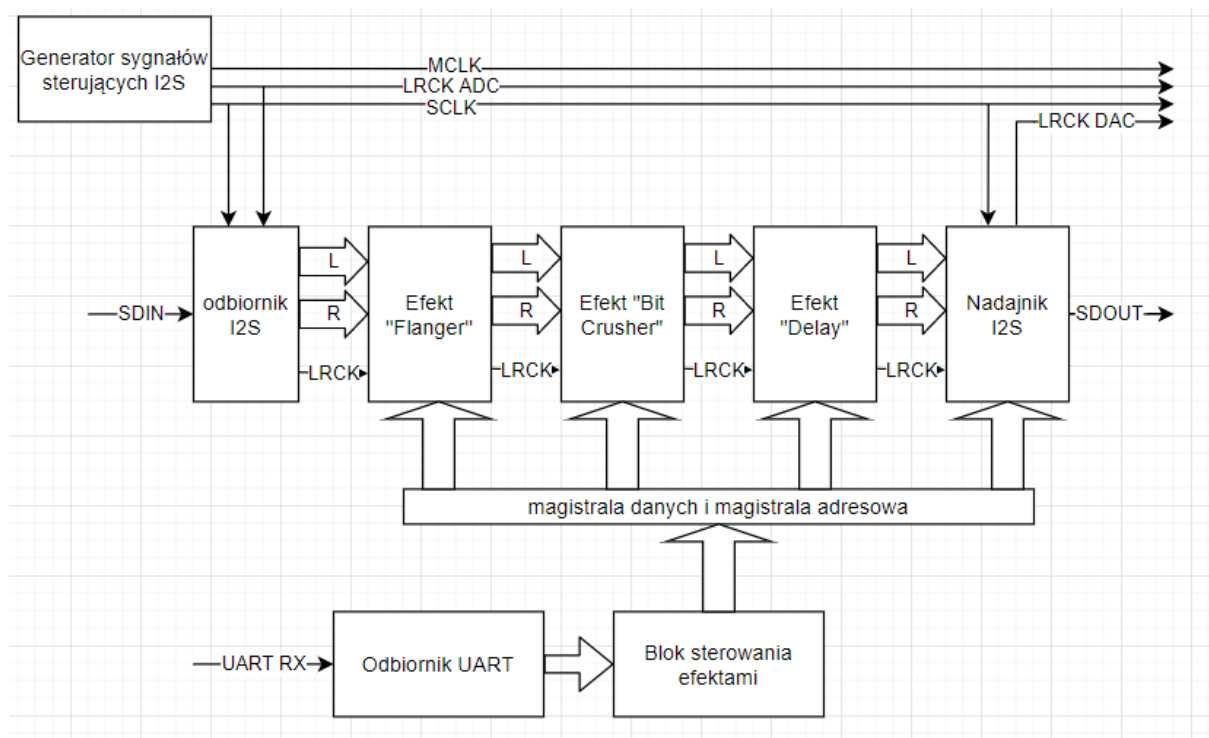


Rysunek 8

5. Blok cyfrowego przetwarzania sygnałów

5.1 Schemat blokowy

Blok cyfrowego przetwarzania sygnałów jest głównym elementem składowym tego projektu. Jest to układ cyfrowy wykonany w oparciu o układ FPGA Artix-7 XC7A35T. Odpowiada on za komunikację z przetwornikami DAC i ADC poprzez interfejs I2S oraz wykonywanie algorytmów przetwarzania sygnału na ciągach próbek. Kod źródłowy znajduje się w pliku *top.vhd*. Schemat blokowy zrealizowanego układu cyfrowego został przedstawiony poniżej:



Rysunek 9

Na zewnątrz układu FPGA zostały wyprowadzone następujące sygnały:

- MCLK – sygnał zegarowy taktujący przetworniki ADC i DAC,
- LRCK ADC – sygnał wyboru kanału dla przetwornika ADC,
- LRCK DAC – sygnał wyboru kanału dla przetwornika DAC,
- SCLK – sygnał zegarowy dla transmisji I2S wspólny dla obydwu przetworników,
- SDIN – linia danych z przetwornika ADC,
- SDOUT – linia danych nadawanych do przetwornika DAC,
- UART RX – wejście danych z konwertera Bluetooth/UART.

Generator sygnałów sterujących I2S wytwarza sygnały, w uzależnieniu od których pracują pozostałe elementy. Wszystkie elementy układu są taktowane wspólnym sygnałem zegarowym. Wartości kolejnych próbek są przekazywane pomiędzy blokami za pomocą magistrali L i R o szerokości 24 bitów. Zmiana wartości logicznej na linii LRCK informuje o nadejściu kolejnej próbki na magistrali L (zmiana z wartości '1' na '0') lub R (zmiana z '0' na '1'). Częstotliwość próbkowania sygnału audio to 48kHz. Parametry poszczególnych efektów zadawane są za pomocą danych odbieranych przez odbiornik UART. Mechanizm zmiany parametru jest następujący: otrzymany jako pierwszy bajt odpowiadający wartości parametru jest podtrzymywany na magistrali danych. Następnie, po otrzymaniu bajtu adresu, adres jest wystawiany na magistralę adresową na czas trwania jednego cyklu zegara CLK. Jeśli adres zostanie rozpoznany przez któryś z bloków efektów, zostanie do niego załadowana wartość z magistrali danych. Szczegółowy opis zawartości modułów znajduje się w kolejnych podpunktach.

5.2 Odbiornik I2S

Moduł opisany w pliku *I2S_Receiver.vhd* służy do odbierania danych z przetwornika ADC CS5344 poprzez interfejs I2S w wersji Left-Justified zgodnie z poniższym diagramem. Wartości próbek są zapisane na 24 bitach w kodzie U2.

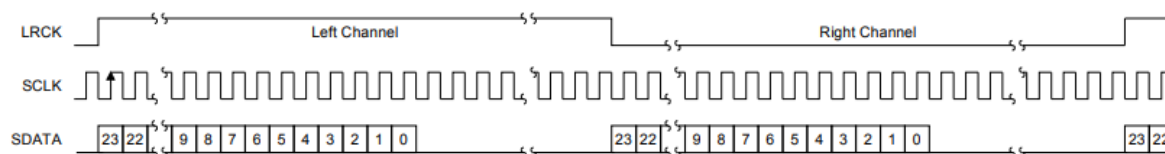


Figure 5. CS5344 Left-Justified Serial Audio Interface

Rysunek 10. Źródło: nota katalogowa układu CS5344

Tabela portów WE/WY:

Nazwa	Kierunek [WE, WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnal taktujący 100MHz
CE	WE	1	Sygnal zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
LRCK_IN	WE	1	Wejście sygnału LRCK z przetwornika ADC, wyznaczającego początek nowego strumienia bitów na linii SDIN
LRCK_OUT	WY	1	Wyjście informujące kolejny element o gotowości danych na magistrali L_OUT lub R_OUT
SDIN	WE	1	Szeregowe wejście danych z przetwornika ADC
SCLK	WE	1	Wejście sygnału wyznaczającego momenty odczytu wartości logicznych na linii SDIN
L_OUT	WY	24	Magistrala odebranych danych odpowiadających próbkom z lewego kanału audio
R_OUT	WY	24	Magistrala odebranych danych odpowiadających próbkom z prawego kanału audio

Mechanizm działania modułu jest następujący:

W każdym cyklu zegara taktującego wartości logiczne na liniach SCLK i LRCK są zapisywane w dwuelementowych rejestrach przesuwanych. Po wykryciu zmiany na linii LRCK, podczas następujących po sobie zboczy narastających na linii SCLK, wartości z linii SDIN są zapisywane w wewnętrznym rejestrze. Po odczytaniu 24 bitów wartość zapisana w tym rejestrze jest przekazywana na magistralę L_OUT lub R_OUT w zależności od stanu wejścia LRCK.

5.3 Nadajnik I2S

Moduł opisany w pliku *I2S_Transmitter.vhd* służy do nadawania danych do przetwornika DAC CS4345 w tym samym standardzie co moduł odbiornika I2S.

Tabela portów WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnał taktujący 100MHz
CE	WE	1	Sygnał zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
LRCK	WE	1	Wejście informujące o pojawieniu się nowych danych na magistrali L_IN lub R_IN
R_IN	WE	24	Magistrala danych prawego kanału
L_IN	WE	24	Magistrala danych lewego kanału
SCLK	WE	1	Wejście sygnału zegarowego transmisji szeregowej
SDOUT	WY	1	Wyjście danych transmisji szeregowej

Po detekcji zbocza na linii LRCK kolejne 24 bity z magistrali R_IN lub L_IN są wystawiane na wyjściu SDOUT w momentach występowania zboczy opadających na wejściu SCLK.

5.4 Odbiornik UART

Moduł znajdujący się w pliku *UART_Receiver.vhd* służy do odbioru danych w standardzie UART z prędkością 9600 bodów (wartość generyczna ustawiana w entity modułu). Kontrola poprawności odebranych danych na podstawie bitu parzystości nie została wykonana w obecnej wersji.

Tabela portów WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnał taktujący 100MHz
CE	WE	1	Sygnał zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
RX	WE	1	Szeregowe, asynchroniczne wejście danych interfejsu UART.
D_VALID	WY	1	Wyjście informujące o odebraniu nowego bajtu danych
D_OUT	WY	8	Magistrala danych odebranych przez interfejs

Podobnie jak inne moduły, odbiornik UART wykrywa początek transmisji na linii RX(zbocze opadające) poprzez zapamiętywanie wartości na linii RX w kolejnych cyklach zegara w dwuelementowym rejestrze przesuwym oraz porównywanie zawartości tego rejestru z wartością 10_{BIN} . W celu zminimalizowania ryzyka wystąpienia stanu metastabilnego, częstotliwość sprawdzania wartości na linii RX została zmniejszona 100 razy względem zegara taktującego CLK. Zabieg ten nie powoduje błędów w odbiorze danych ze względu na niską prędkość transmisji 9.6 Kb/s. Momenty odczytu kolejnych bitów z linii RX wyznaczone są za pomocą wewnętrznego licznika, skonfigurowanego na podstawie generycznego parametru entity, uzależnionego od szybkości transmisji.

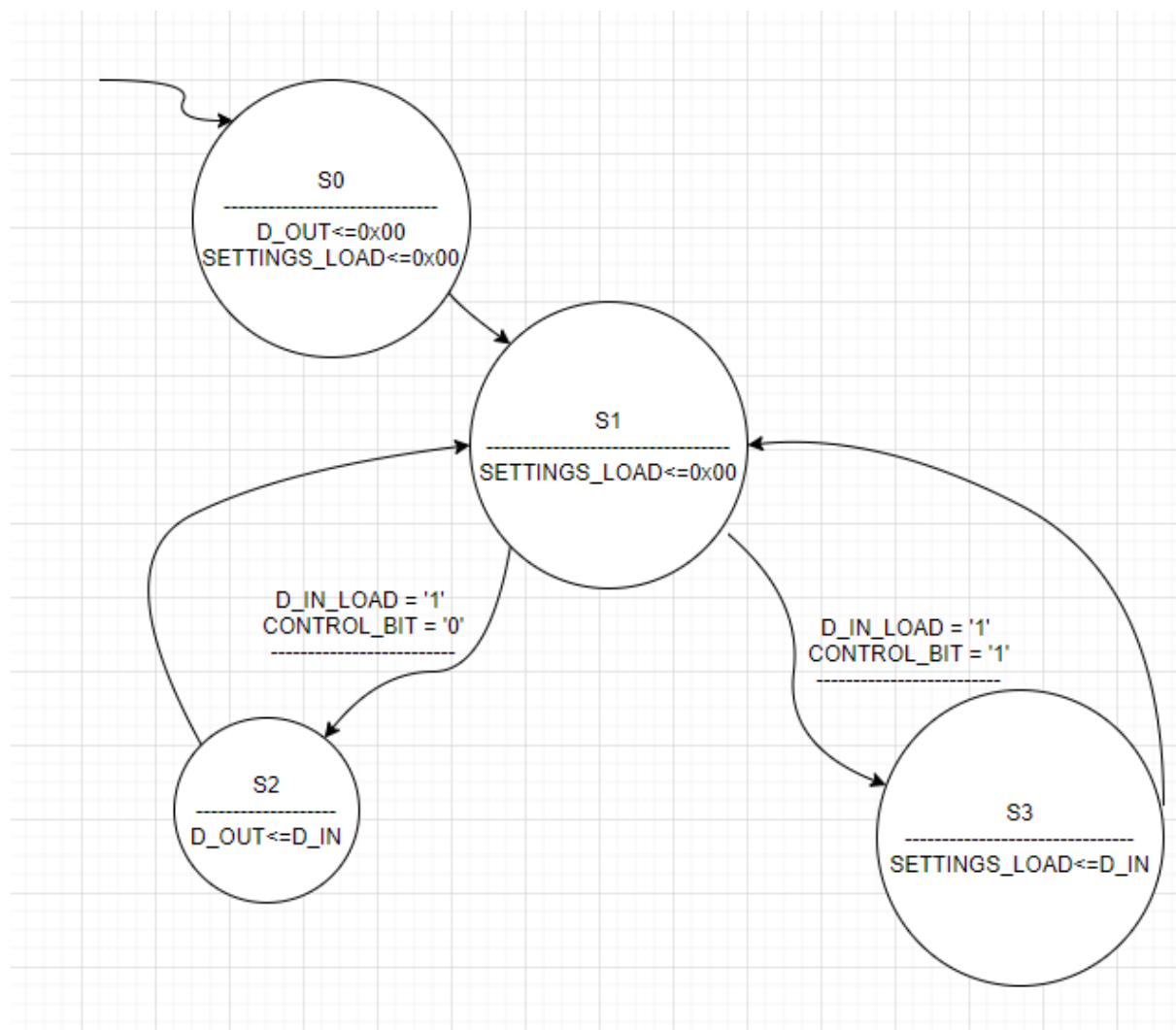
5.5 Blok sterowania efektami

Ten blok został opisany w pliku *settings_loader.vhd*. Odpowiada za dekodowanie rozkazów zmiany ustawień efektów oraz propagowanie odpowiednich instrukcji sterujących do odpowiednich modułów efektów.

Tabela portów WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnał taktujący 100MHz
CE	WE	1	Sygnał zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
CONTROL_BIT	WE	1	Wartość logiczna na tym wejściu określa czy słowo obecne na magistrali D_IN_LOAD jest wartością parametru(stan wysoki), czy adresem parametru(stan niski). Ten port powinien zostać dołączony do MSB magistrali danych odebranych z interfejsu UART.
D_IN	WE	8	Magistrala danych odbieranych od modułu odbiornika UART.
D_IN_LOAD	WE	1	Wejście wymuszające odczyt z magistrali D_IN
SETTINGS_LOAD	WY	8	Magistrala adresowa parametrów
D_OUT	WY	8	Magistrala danych parametrów

Algorytm działania modułu został przedstawiony na poniższym diagramie:



Rysunek 11

Dla poprawnej współpracy modułu z resztą układu należy zapewnić następujący standard komunikacji:

- jako pierwszy wysyłany jest bajt z nową wartością parametru,
- kolejno, wysyłany jest bajt z adresem parametru.

Adresy zostały zapisane w postaci stałych w modułach poszczególnych obiektów. Lista wykorzystanych adresów znajduje się poniżej.

Moduł efektu	Nazwa parametru w interfejsie użytkownika	Nazwa stałej w pliku .vhd	Adres
Flanger	Włącz/wyłącz	c_on_off_setting	0x21
	Mix	c_mix_setting	0x22
	Depth	c_depth_setting	0x23
	Rate	c_rate_setting	0x24
Bit Crusher	Włącz/wyłącz	c_on_of_setting	0x01
	Q factor	c_frame_len_setting	0x02
Delay	Włącz/wyłącz	c_on_off_setting	0x41
	Time	c_time_setting	0x42
	Volume	c_volume_setting	0x43

5.6 Efekt „Flanger”

Ten moduł odpowiada za dodawanie efektu *Flanger* do sygnału dźwiękowego. Kod vhd znajduje się w pliku *flanger.vhd*.

Tabela WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnał taktujący 100MHz
CE	WE	1	Sygnał zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
SETTING	WE	8	Magistrala danych parametrów efektu
SETTINGS_LOAD	WE	8	Magistrala adresowa parametrów efektów
L_IN	WE	24	Magistrala danych wejściowych dla lewego kanału
R_IN	WE	24	Magistrala danych wejściowych dla prawego kanału

L_OUT	WY	24	Magistrala danych wyjściowych dla lewego kanału
R_OUT	WY	24	Magistrala danych wyjściowych dla prawego kanału
LRCK_IN	WE	1	Wejście informujące o nowej wartości na magistrali L_IN(1 _B) lub R_IN(0 _B)
LRCK_OUT	WY	1	Wyjście informujące o nowej wartości na magistrali L_OUT(1 _B) lub R_OUT(0 _B)

Do realizacji tego [efektu](#), dla każdego z kanałów został wykorzystany bufor kołowy o długości 256 próbek oraz tablica LUT zawierająca 32768 wartości pierwszego kwadrantu okresu funkcji

$$f(x) = 1 - \cos(x)$$

zapisane w 16-bitowej precyzji. Przetwarzanie sygnału polega na dodaniu do sygnału wejściowego jego kopii sprzed K okresów próbkowania, gdzie K jest liczbą rzeczywistą wyznaczoną na podstawie zawartości tablicy LUT oraz zadanych parametrów: *depth* oraz *rate*.

$$K = depth \cdot f\left(\frac{p}{4 \cdot 32768} \cdot 2 \cdot \pi\right)$$

p – indeks w tablicy LUT. Wraz z nadejściem kolejnej próbki sygnału indeks p jest zwiększany o wartość parametru *rate*.

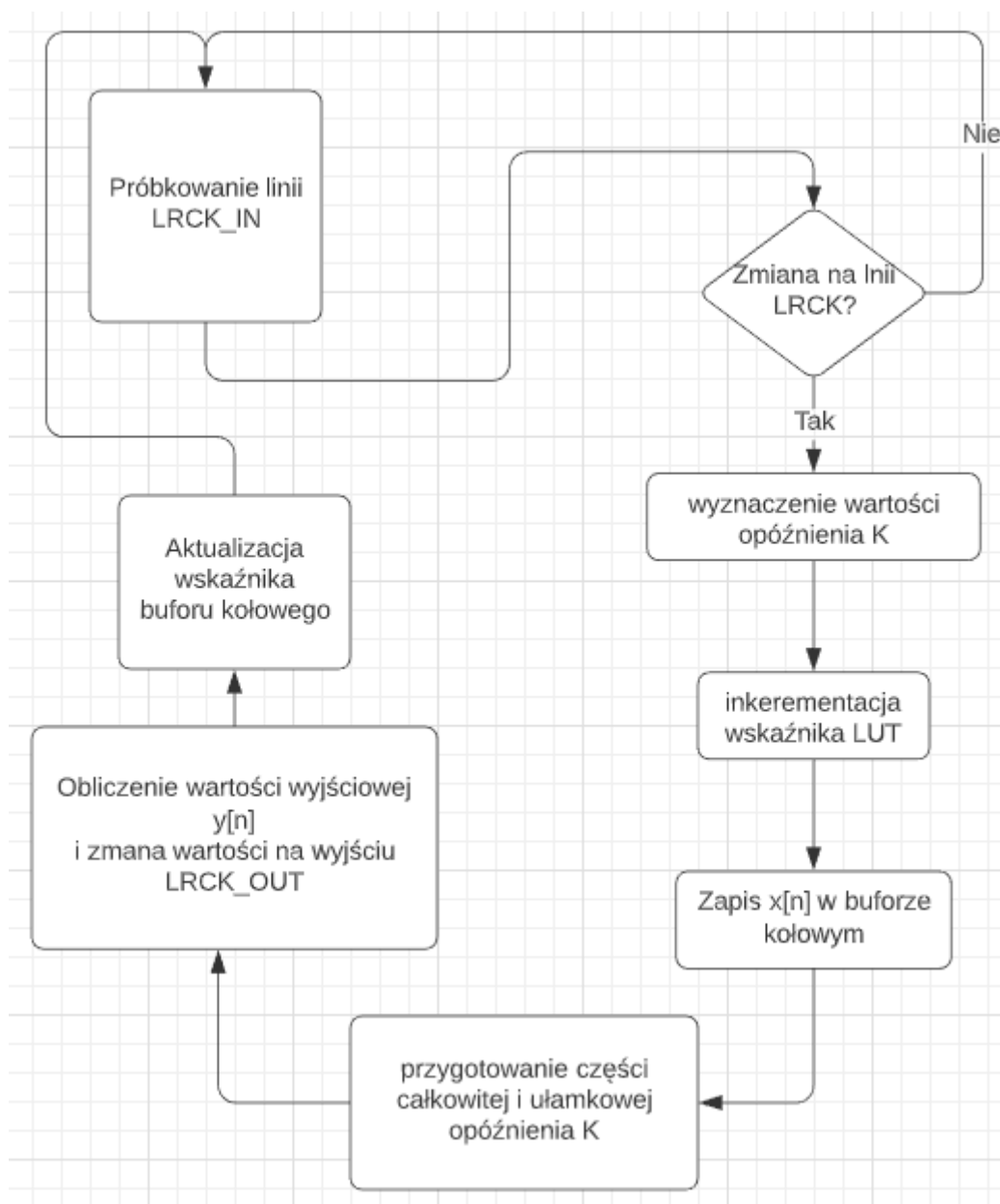
Ponieważ K jest liczbą rzeczywistą, sygnał wyjściowy $y[n]$ jest wyznaczany poprzez interpolację liniową zgodnie ze wzorem:

$$y[n] = (1 - mix) \cdot x[n] + mix \cdot (K_f \cdot x[n - (K_i + 1)] + (1 - K_f) \cdot x[n - K_i])$$

K_i – część całkowita opóźnienia K ,

K_f – część ułamkowa opóźnienia K

Algorytm działania modułu został przedstawiony poniżej:



Rysunek 12

Operacje mnożenia zostały wykonane w oparciu o gotowe moduły DSP znajdujące się w układzie FPGA. Bufory kołowe służące do przechowywania przeszłych próbek z obydwu kanałów zostały zaimplementowane w pamięci BRAM. Tablica LUT funkcji $f(x)$ została zaimplementowana w pamięci DRAM. Algorytm jest realizowany osobno dla kanału lewego i prawego. Wewnątrz modułu znajduje się również blok odpowiedzialny za odczyt rozkazów zmiany parametrów.

5.7 Efekt „Bit crusher”

Moduł tego efektu jest opisany w pliku *bit_crusher.vhd*.

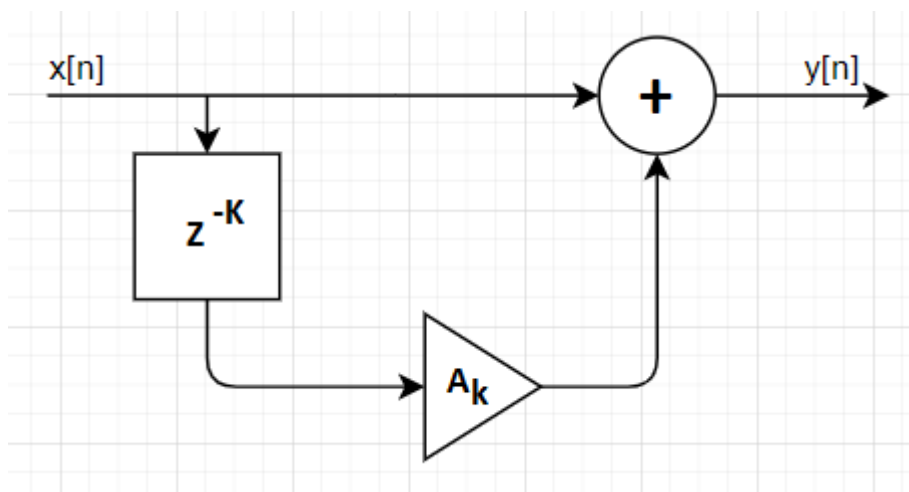
Tabela WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnal taktujący 100MHz
CE	WE	1	Sygnal zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
SETTING	WE	8	Magistrala danych parametrów efektu
SETTING_LOAD	WE	8	Magistrala adresowa parametrów efektów
L_IN	WE	24	Magistrala danych wejściowych dla lewego kanału
R_IN	WE	24	Magistrala danych wejściowych dla prawego kanału
L_OUT	WY	24	Magistrala danych wyjściowych dla lewego kanału
R_OUT	WY	24	Magistrala danych wyjściowych dla prawego kanału
LRCK_IN	WE	1	Wejście informujące o nowej wartości na magistrali L_IN(1 _B) lub R_IN(0 _B)
LRCK_OUT	WY	1	Wyjście informujące o nowej wartości na magistrali L_OUT(1 _B) lub R_OUT(0 _B)

Realizacja efektu opiera się na zmniejszaniu częstotliwości próbkowania sygnału, nie stosując standardowego filtra rekonstrukcyjnego. Zadanie to zostało zrealizowane za pomocą licznika wyznaczającego momenty odczytu wartości sygnału wejściowego. Licznik jest inkrementowany wraz z nadejściem kolejnej próbki sygnału. W momencie, w którym zawartość licznika przekracza wartość parametru *Q factor*(sygnał „*frame_len_setting*” w pliku *vhd*), licznik jest resetowany, a magistrala L_OUT(R_OUT) jest aktualizowana wartością z magistrali L_IN(R_IN).

5.8 Efekt „Delay”

Ten moduł pozwala na wprowadzenie efektu polegającego na dodaniu do strumienia próbek sygnału jego kopii przesuniętej w czasie o zadany okres. Parametr Volume odpowiada za skalowanie amplitudy opóźnionej kopii sygnału. Moduł znajduje się w pliku *delay_effect.vhd*.



Rysunek 13

Tabela WE/WY:

Nazwa	Kierunek [WE,WY, WE/WY]	Szerokość [bit]	Funkcja
CLK	WE	1	Sygnał taktujący 100MHz
CE	WE	1	Sygnał zezwolenia zegara
RESET	WE	1	Reset asynchroniczny
SETTING	WE	8	Magistrala danych parametrów efektu
SETTING_LOAD	WE	8	Magistrala adresowa parametrów efektów
L_IN	WE	24	Magistrala danych wejściowych dla lewego kanału
R_IN	WE	24	Magistrala danych wejściowych dla prawego kanału
L_OUT	WY	24	Magistrala danych wyjściowych dla lewego kanału
R_OUT	WY	24	Magistrala danych wyjściowych dla prawego kanału
LRCK_IN	WE	1	Wejście informujące o nowej wartości na magistrali L_IN(1 _B) lub R_IN(0 _B)

LRCK_OUT	WY	1	Wyjście informujące o nowej wartości na magistrali L_OUT(1 _B) lub R_OUT(0 _B)
----------	----	---	--

Przeszłe wartości próbek są zapamiętywane w buforze kołowym o rozmiarze 65536 próbek zaimplementowanym w pamięci blokowej. Wartości próbek z obydwu kanałów są zapisywane na przemian w kolejnych komórkach pamięci. Maksymalny czas opóźnienia to 700 ms. Do skalowania amplitudy sygnału został użyty rejestr przesuwany. Podejście to pozwala tylko na dzielenie wartości sygnału przez kolejne potęgi liczby 2.

6. Uwagi i doświadczenia zdobyte podczas realizacji projektu

Opis pamięci ROM o dużych rozmiarach (tablica wartości funkcji cosinus użyta w module efektu Flanger), jeśli zostaje wykonany w języku VHDL znacznie wydłuża czas syntezy projektu. W tym przypadku czas, jakiego narzędzie potrzebowało do syntezy pamięci ROM i inicjalizacji jej wartości wynosił około godziny, co znacznie utrudniało testy funkcjonalne. Aby usprawnić ten proces należy wcześniej przygotować plik z wartościami tablicy LUT przy użyciu skryptu, a następnie odczytać wartości z pliku za pomocą języka opisu sprzętu.

W przypadku występowania sygnału asynchronicznego na wejściu informacyjnym istnieje ryzyko wprowadzenia przerzutnika w stan metastabilny. Problem ten wystąpił w module odbiornika UART. Dodatkowo czas opadania zbocza na linii RX mógł okazać się duży w porównaniu do okresu zegara CLK ze względu na pojemności pasożytnicze połączeń. Aby zmniejszyć prawdopodobieństwo wystąpienia stanu metastabilnego, częstotliwość odczytu stanu linii RX została zmniejszona. Innym sposobem rozwiązania tego problemu mogło być zastosowanie dodatkowego przerzutnika D jako synchronizatora.