

# Systemy Dedykowane w Układach Programowalnych

## Projekt

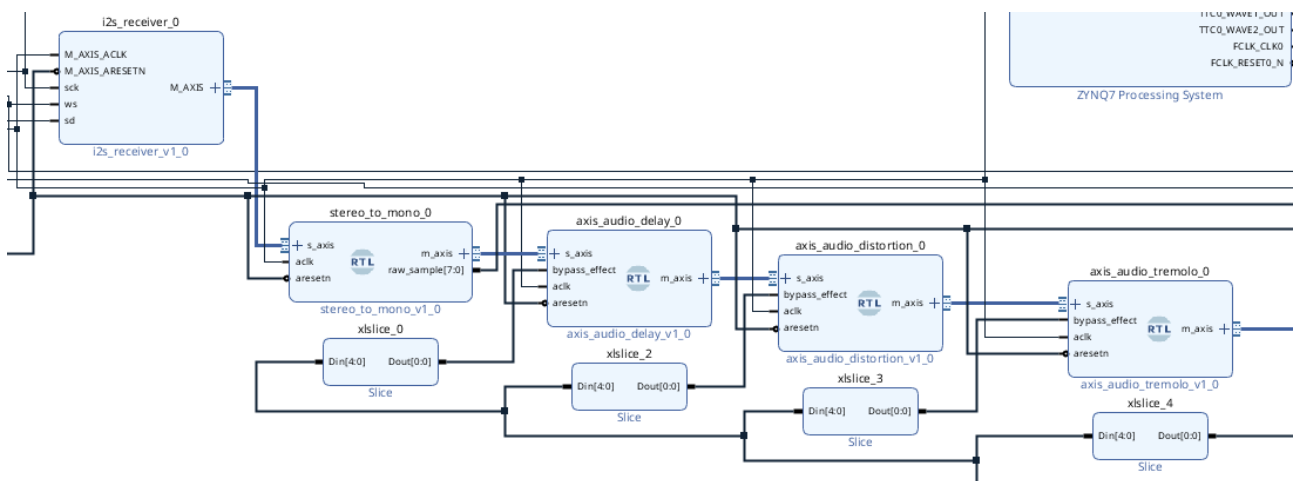
### Skrzynka z efektami audio

## 1. Wstęp do projektu

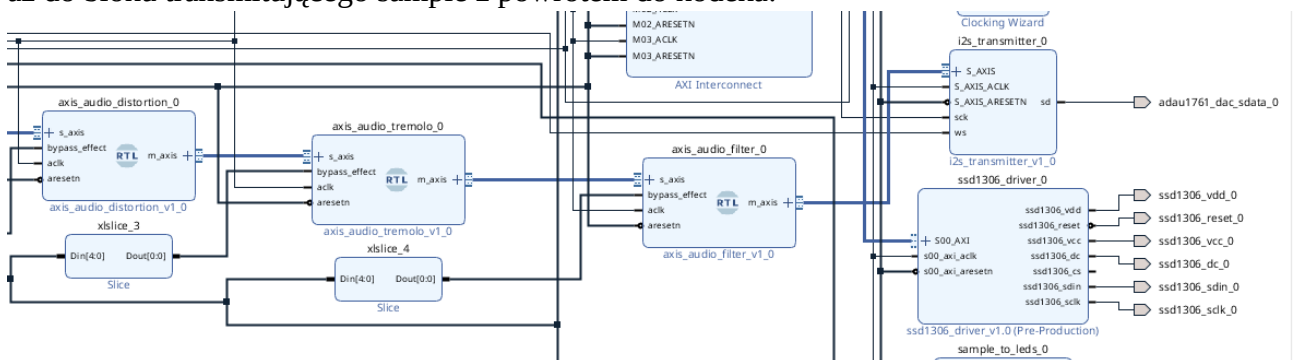
Głównym celem projektu było wykorzystanie możliwości układu FPGA do nakładania efektów na strumień audio w czasie rzeczywistym. Z uwagi na ograniczony czas, postanowiono na ograniczenie się do czterech prostych efektów, a jako interfejs użytkownika wykorzystano obecny na płycie ZedBoard wyświetlacz OLED oraz 3 przyciski sterujące.

## 2. Sposób przetwarzania audio

Cały proces przetwarzania odbywa się sprzętowo – od odbioru sampli z kodeka ADAU1761 do ich ponownego wysłania do odtworzenia na wyjściu słuchawkowym. Poniższy fragment diagramu prezentuje przepływ kolejnych sampli przez system...



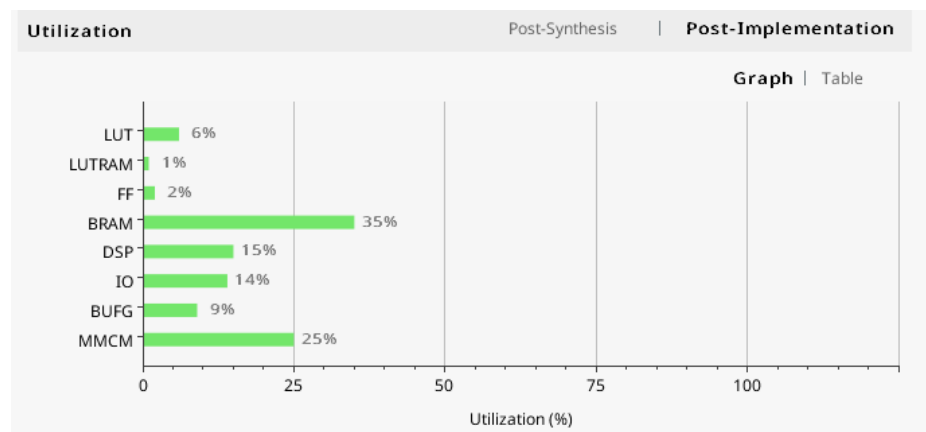
aż do bloku transmitującego sample z powrotem do kodeka:



W ramach projektu zaimplementowano:

- Blok sumujący sample stereo na dźwięk monofoniczny
  - Został wykonany jako baza dla kolejnych bloków wykorzystujących AXI-Stream, wynika z faktu że wyjście słuchawkowe kodeka ADAU1761 jest monofoniczne
- Blok opóźniający z możliwością przełączenia toru na nieopóźniony dźwięk za pomocą przycisku
  - Wykorzystuje Block RAM dostępny w układzie FPGA, daje kilkaset milisekund opóźnienia, a dzięki łatwemu przełączaniu między opóźnionym a nieopóźnionym dźwiękiem można uzyskać efekt „przeskakiwania” utworu
- Blok implementujący zniekształcenie dźwięku
  - Dodaje charakterystyczne szумы oraz efekt przesterowania sygnału
- Blok implementujący wolnozmienną obwiednię nanoszoną na sygnał, tzw. „tremolo” ze stałą częstotliwością
  - Może dawać przyjemny efekt dla niektórych instrumentów, np. gitar elektrycznych, ale z uwagi na ograniczony czas nie udało się zaimplementować sposobu synchronizacji tempa z sygnałem
- Blok implementujący prosty filtr FIR
  - Obecnie użyty jest filtr 32 rzędu z takimi samymi współczynnikami, dający filtr typu moving-average
  - Implementacja wykorzystująca bloki DSP48 dostępne w układzie FPGA

Wszystkie te bloki wykorzystują magistralę AXI Stream do przesyłu próbek, więc ich kolejność można przed procesem syntezy dowolnie zmieniać. Dodatkowo zaimplementowano prosty blok wykorzystujący 8 diod LED jako wskaźnik poziomu sygnału.



Z uwagi na ograniczony czas do obsługi zewnętrznych układów takich jak wyświetlacz OLED czy kodek audio wykorzystano następujące źródła:

- <https://github.com/zdzislaw-s/audio-processing>
- <https://github.com/zdzislaw-s/oled-driver-demo>

Oraz do obsługi grafiki na wyświetlaczu OLED w języku C++:

- <https://github.com/lexus2k/canvas>

## Obsługa urządzenia

Po uruchomieniu programu na wyświetlaczu OLED wyświetlone zostaną ikony symbolizujące dostępne efekty. Są to od lewej:

- wskaźnik poziomu sygnału na diodach LED,
- efekt „delay”,
- efekt „distortion”
- efekt „tremolo”
- filtr dolnoprzepustowy



Jeśli błądzek nad ikonką jest zamalowany, oznacza to że efekt jest włączony. Błądzek na górze ekranu pokazuje aktualnie wybrany efekt. Między efektami przełącza się za pomocą przycisków BTNL i BTNR. Włączenie lub wyłączenie wybranego efektu realizuje się przyciskiem BTNC.