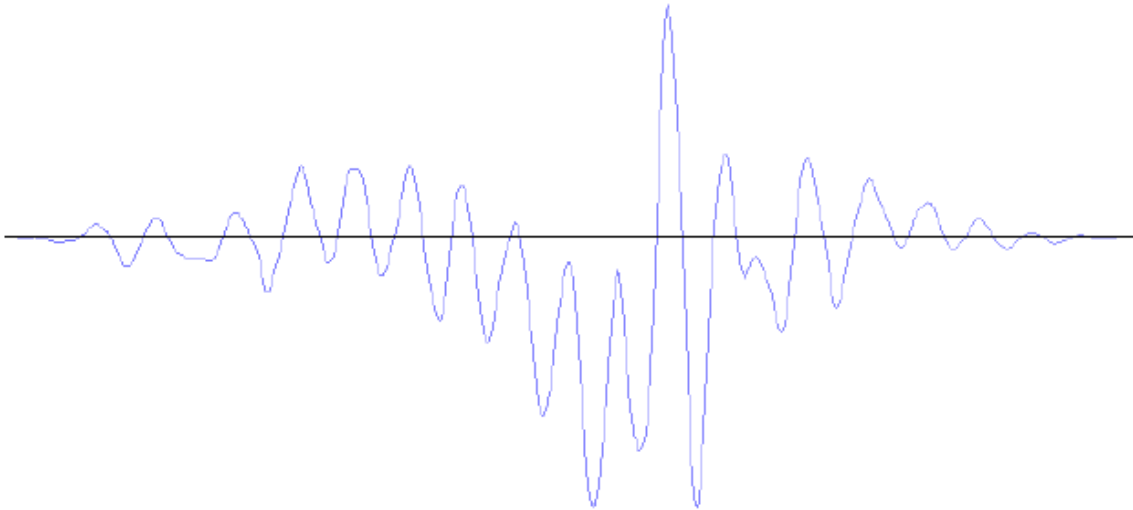


## Lab.13. Podstawy cyfrowego przetwarzania sygnałów

Ćwiczenie polega na użyciu kodera / dekodera audio (coder / decoder - CODEC) na karcie DE1-SoC lub DE2-115. Ćwiczenie polega na podłączeniu mikrofonu do kodeka audio w celu zapewnienia dźwięku wejściowego, zmiany odbieranego dźwięku, odfiltrowania szumów, a następnie odtworzenia powstałego dźwięku przez głośniki / słuchawki. Oprócz płyty głównej z serii DE potrzebujesz mikrofonu i głośników lub słuchawek.

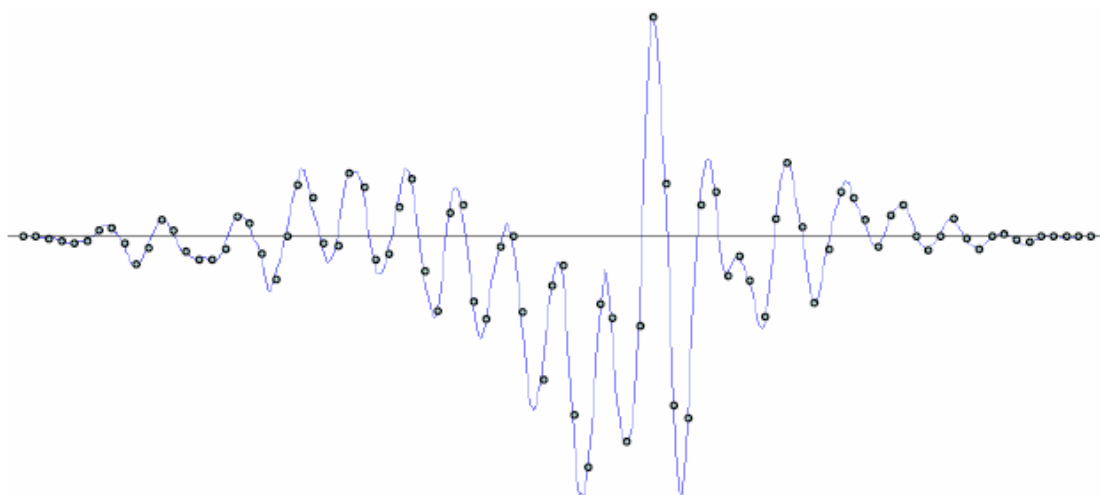
### Podstawy

Dźwięki, takie jak mowa i muzyka, są sygnałami, które zmieniają się z czasem. Amplituda sygnału określa głośność, z jaką słyszymy dźwięk. Zmieniając sygnał w czasie, określa się rodzaj dźwięków, które słyszymy. Na przykład dźwięk "ah" jest reprezentowany przez kształt fali pokazany na rys. 1.



Rys. 1: Waveform dla dźwięku 'ah'.

Waveform to sygnał analogowy, który można zapisać cyfrowo za pomocą stosunkowo małej liczby próbek, które reprezentują wartości analogowe w określonych punktach czasowych. Proces tworzenia takich sygnałów cyfrowych nazywany jest próbkowaniem.

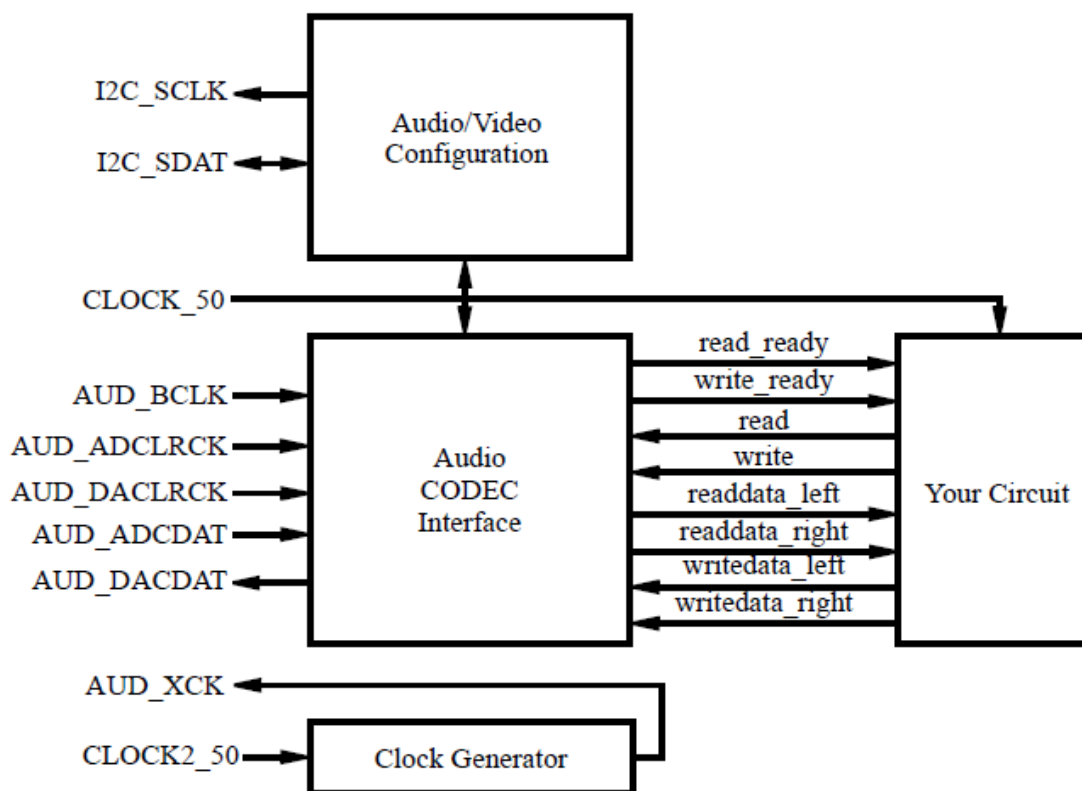


Rys. 2: Wybrany kształt fali dźwięku "ah".

Punkty na rys. 2 opisują wybrany kształt fali. Wszystkie punkty są rozmieszczone równomiernie w czasie i śledzą oryginalny przebieg.

Płyty DE1-SoC i DE2-115 są wyposażone w kodek audio (audio CODEC), zdolny do próbkowania dźwięku z mikrofonu i wykorzystania go jako wejścia do układów. Domyślnie CODEC dostarcza 48000 próbek na sekundę, co jest wystarczające do dokładnego odwzorowania dźwięków słyszalnych.

Ćwiczenie obejmuje projekt kilku układów, które pobierają dane wejściowe z mikrofonu za pośrednictwem CODECa, nagrywają i przetwarzają te dane audio, a następnie odtwarzają je za pomocą głośników. Aby uprościć zadanie, prosty system, który może nagrywać i odtwarzać dźwięki na płycie z serii DE, jest dostarczany jako "zestaw startowy". Układ pokazany na rys. 3, zawiera generator zegara, interfejs audio CODEC i moduły konfiguracji audio / wideo. Ten interfejs jest uproszczoną wersją University Program Audio IP Core, który jest dostępny na stronie internetowej Intel's FPGA University Program.



Rys. 3: System audio do omawianego ćwiczenia.

Lewa strona rysunku 3 pokazuje wejścia i wyjścia systemu. Te porty wejścia / wyjścia zapewniają wejścia zegara, a także podłączają CODEC AUDIO i moduły konfiguracji audio / video do odpowiednich urządzeń peryferyjnych na płytach Intel DE1-SoC i DE2-115. Na środku rysunku pokazano grupę sygnałów do i z modułu interfejsu AUDIO CODEC. Sygnały te pozwalają układowi pokazanemu po prawej stronie na rejestrowanie dźwięków z mikrofonu i odtwarzanie ich przez głośniki.

System działa w następujący sposób. Aby zresetować konfigurację audio / video, rozpoczyna się sekwencja autoinicjalizacji. Sekwencja ustawia urządzenie audio na wejście demonstracyjne mikrofonu z częstotliwością 48 kHz i wyprowadzanie sygnału przez głośniki na tej samej częstotliwości. Po zakończeniu automatycznej inicjalizacji CODEC AUDIO rozpoczyna odczytywanie danych z mikrofonu 48 000 razy na sekundę i wysyła je do rdzenia interfejsu AUDIO CODEC w systemie. Następnie jak otrzymano próbkę, jest ona przechowywana w buforze o 128 elementach w rdzeniu interfejsu AUDIO CODEC. Pierwszy element bufora jest zawsze widoczny na wyjściach `readdata_left` i `readdata_right`, gdy ustawiony jest sygnał `read_ready`. Następny element można odczytać, ustawiając sygnał odczytu, który pobiera bieżącą próbkę, a nowa próbka pojawia się jeden lub więcej cykli zegara później, jeśli ustawiony jest sygnał `read_ready`.

Aby wyprowadzić dźwięk przez głośniki, wykonywana jest podobna procedura. Twój układ powinien obserwować sygnał `write_ready`, a jeśli potwierdzono zapis próbki, AUDIO CODEC dostarcza go na wejściach `writedata_left` i `writedata_right` i jest on potwierdzany przez sygnał

zapisu. Ta czynność przechowuje próbkę w wewnętrznej części bufora interfejsu AUDIO CODEC, która wyśle próbkę do głośników we właściwym czasie.

Pakiet startowy zawierający ten projekt stanowi część tego ćwiczenia.

**Zadanie 1.** Prześlij dane wyjściowe kodeka na jego wejście. Skompiluj układ i załaduj go do Intel DE1-SoC lub DE2-115. Podłącz mikrofon i głośniki do portów Mic i Line Out płyty i mów do mikrofonu, aby usłyszeć swoją wypowiedź poprzez głośniki.

W tej części ćwiczenia należy wykonać prostą modyfikację w dostarczonym układzie pakietu startowego, aby przesłać dane wejściowe z mikrofonu do głośników. Należy zachować ostrożność, aby odczytywać dane i zapisywać dane w interfejsie AUDIO CODEC tylko wtedy, gdy ustawione są gotowe sygnały.

**Zadanie 2.** Dodaj generator szumu do sygnału wejściowego. Ustaw odpowiednią wartość szumu przy pomocy zmiany szerokości i miejsca bitów licznika w sygnale szumu. Wykonaj symulację. Zaprojektuj jednostkę sterującą generatorem szumu i sygnałami sterującymi AUDIOCODECa (read, write). Szum ma być włączany przełącznikiem SW0.

Ten układ to prosty licznik, którego wartość należy interpretować jako **wartość ze znakiem**. Układ musi być zsynchronizowany z zegarem 50 MHz, a sygnał zezwolenia musi być kontrolowany przez poziom wysoki, kiedy moduł AUDIO CODEC może zarówno wyprodukować, jak i przyjąć nową próbkę.

```
module noise_generator (clk, enable, Q);
    input clk, enable;
    output [23:0] Q;
    reg [2:0] counter;

    always@(posedge clk)
        if (enable)
            counter = counter + 1'b1;

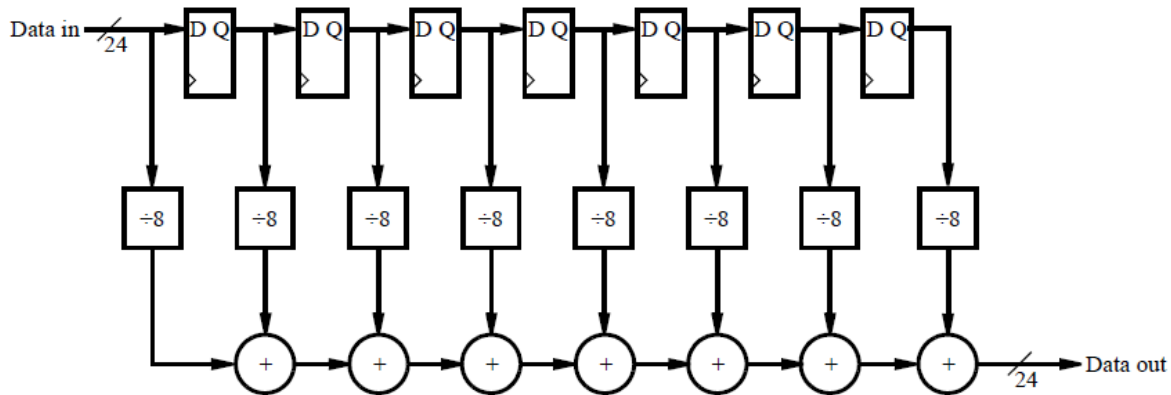
    assign Q = {{10{counter[2]}}, counter, 11'd0};
endmodule
```

Rys.4. Układ generacji szumu

Aby usłyszeć efekt generatora szumów, dodaj wartości wytwarzane przez układ do każdej próbki audio z CODEC AUDIO w układzie z zadania 1.

**Zadanie 3.** Zaimplementuj filtr cyfrowy uśredniający z 8 próbek. Wykonaj symulację. Sprawdź jego działanie dla sygnału przed i po dodaniu szumu z generatora z zadania 2. Dodaj do jednostki sterującej funkcję sterowania sygnałem włączenia filtra przełącznikiem SW1.

Filtrowanie to proces korekcji sygnału, na przykład usuwanie szumów. Szum w fali dźwiękowej jest reprezentowany przez małe, ale częste zmiany amplitudy sygnału. Prostym układem logicznym, który realizuje zadanie filtrowania szumów, jest filtr uśredniający o skończonej odpowiedzi impulsowej (FIR). Schemat filtra przedstawiono na rys. 5.



Rys.5. Prosty filtr uśredniający **FIR**.

Filtr uśredniający, podobny do filtra pokazanego na rys. 5, usuwa szumy z dźwięku, uśredniając wartości sąsiednich próbek. W tym przypadku usuwa niewielkie odchylenia dźwięku, obserwując zmiany w sąsiednich 8 próbkach. Podczas korzystania z mikrofonów niskiej jakości ten filtr powinien usuwać zakłócenia powstające podczas mówienia do mikrofonu, dzięki czemu dźwięk mowy będzie wyraźniejszy.

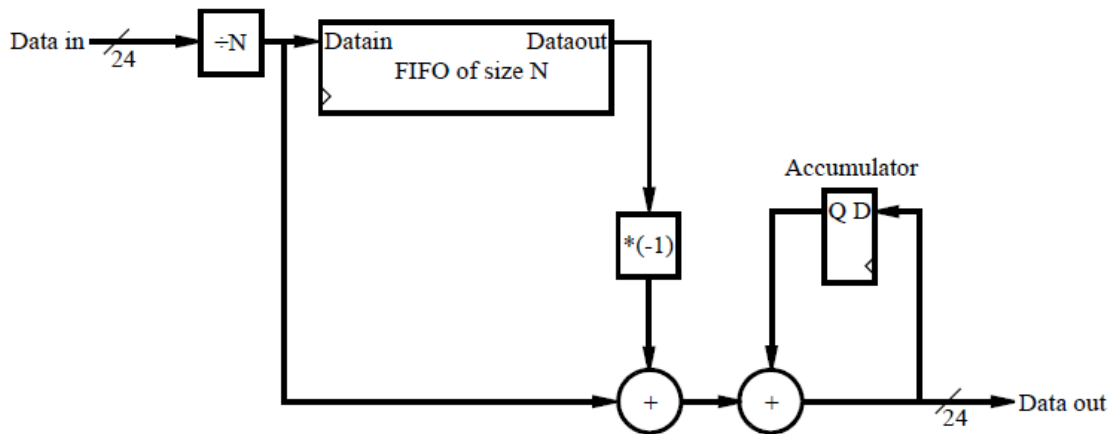
Musisz zaimplementować układ, który pokazano na rys. 5, aby przetworzyć dźwięk z mikrofonu i wyprowadzić przefiltrowany dźwięk przez głośniki. Czy zauważasz jakąkolwiek różnicę między jakością dźwięku w tej części w porównaniu do pierwszej części ćwiczenia?

Uwaga: Możesz zastosować wysokiej jakości mikrofon z funkcją redukcji szumów. W takich okolicznościach prawdopodobnie nie będzie słychać żadnego efektu korzystania z tego filtra. Jeśli tak, sugerujemy wprowadzenie szumu do dźwięku, dodając wyjście układu z rys. 4 do próbki wyprodukowanej przez CODEC AUDIO.

**Zadanie 4.** Zaimplementuj N-próbkowy filtr uśredniający FIR. Wykonaj symulację. Sprawdź działanie filtra na płycie dla różnych wartości N będących potęgą liczby 2 ( $N = 4, 8, 16, 32$ ). Dodaj możliwość włączania szumu i filtra przełącznikami SW0 i SW1.

Realizacja filtra uśredniającego z zadania 2, może być skuteczna w usuwaniu części szumu wytwarzanego przez generator szumu. Jednakże, jeśli mikrofon jest słabej jakości, albo zwiększymy szerokość licznika generatora szumu, filtr z drugiej części będzie niewystarczający, aby usunąć szum. Powodem tego jest to, że filtr z drugiej części, obserwuje tylko bardzo krótki okres czasu, podczas którego zmienia się kształt fali dźwiękowej. Można to poprawić, tworząc większy filtr, biorąc średnią liczbę kolejnych próbek.

W tej części, należy poeksperymentować z wielkością filtra do określenia liczby próbek, w których trzeba uśrednić wejście audio do usuwania szumu tła. Aby to zrobić skutecznie, wykorzystaj do uśredniania filtr FIR, pokazany na rys. 6.



Rys. 6: N-próbkowy filtr uśredniający **FIR**.

Aby obliczyć średnią liczbę ostatnich  $N$  próbek, układ najpierw dzieli próbkę wejściową przez  $N$ . Następnie uzyskana wartość jest przechowywana w buforze (FIFO - First-In First-out) o długości  $N$  i dodawana do akumulatora. Aby upewnić się, że wartość w akumulatorze jest średnią z ostatnich próbek  $N$ , obwód odejmuje wartość, która opuszcza FIFO, która reprezentuje  $(n + 1)$ -szą próbkę.

Zaimplementuj, skompiluj i załaduj układ na płytę Intel DE1-SoC. Podłącz mikrofon i głośniki do portów Mic i Line Out i mów do mikrofonu, aby usłyszeć swoją wypowiedź przez głośniki. Eksperymentuj z różnymi wartościami  $N$ , aby zobaczyć, co dzieje się z Twoją mową i szumem w tle, pamiętając o dzieleniu próbek przez odpowiednią wartość. Zalecamy eksperymentowanie z wartościami  $N$ , które są potęgą 2, aby ułatwić dzielenie.

Jeśli masz przenośny odtwarzacz audio ze złączem, dzięki któremu możesz wprowadzać sygnał wejściowy do układu przez port Mic, spróbuj eksperymentować z różnymi rozmiarami filtrów i ich wpływem na utwór, który odtwarzasz.