ZASTOSOWANIA PROCESORÓW SYGNAŁOWYCH - PROJEKT

Harmonogram projektu

Terminy realizacji i prowadzący zajęcia:

- T1 – wtorek, 8.15–10.00, NE 237 - dr inż. Grzegorz Szwoch (pok. 732)
- T2 – wtorek, 8.15–10.00, NE 237 - dr inż. Piotr Odya (pok. 730)
- T0 – poniedziałek, 16.15–18.00, NE 237 - mgr inż. Adam Korzeniewski (pok. 732)

N	T1	T2	T0	Etap projektu
0	25.02			Zajęcia organizacyjne **NE Aud 1P** (**obecność obowiązkowa!**)
1	5.03	12.03	11.03	Wprowadzenie do DSP (bez pkt., ale **obecność obowiązkowa!**)
2	19.03	26.03	25.03	Projekt i implementacja filtru FIR (5 pkt)
3	2.04	9.04	8.04	Implementacja i testowanie filtru IIR (5 pkt)
4	16.04	7.05	6.05	Implementacja algorytmu FFT (5 pkt)
5	14.05	21.05	20.05	Implementacja alg. detekcji częstotliwości (5 pkt)
6	28.05	4.06	3.06	Testowanie algorytmów (5 pkt)
7	11.06	13.06	17.06	Zaliczenie projektu - raport końcowy (25 pkt)

Materiały pomocnicze do realizacji projektu

- Prezentacje z wykładów ZPS: http://multimed.org/student/materialy.html#zps
- A. Leśnicki: *Technika cyfrowego przetwarzania sygnałów*.Wyd. PG, Gdańsk 2014.
- *TMS320C55x DSP Library Programmer's Reference* (SPRU422J): http://www.ti.com/lit/ug/spru422j/spru422j.pdf
- *TMS320C55x Optimizing C/C++ Compiler User's Guide* (SPRU281G): http://www.ti.com/cn/lit/pdf/spru281
- *TMS320C5535 Fixed-Point Digital Signal Processor* (różne materiały): http://www.ti.com/product/TMS320C5535/technicaldocuments
- *C5000 Teaching ROM:* https://e2e.ti.com/group/universityprogram/educators/w/wiki/2040.c5000-teaching-rom
- *Spectrum Digital eZdsp5535*: http://support.spectrumdigital.com/boards/ezdsp5535/revc/
- *Kurs języka C*: https://pl.wikibooks.org/wiki/C

Organizacja zajęć

- W trakcie semestru odbywa się 6 spotkań projektowych (po 2 godz.) oraz zajęcia wprowadzające.
- Projekt jest realizowany w dwuosobowych grupach.
- Grupa zajmuje zawsze to samo stanowisko komputerowe.
- Każda grupa otrzymuje na zajęciach jedną płytkę DSP.
- Grupa posiada swój numer i powinna na każdych zajęciach używać tej samej płytki, wg numeru podanego na pudełku.
- Nie ma możliwości wypożyczania płytek poza zajęciami.
- Wszelkie usterki i nieprawidłowości należy zgłaszać prowadzącemu.
- Studenci odpowiadają za uszkodzenia sprzętu wynikające wyłącznie z ich winy.
- Pomoc w realizacji projektu można uzyskać u prowadzącego, w terminie jego konsultacji oraz, po uprzednim uzgodnieniu, w dodatkowych terminach ustalonych przez prowadzącego.

Przygotowanie do zajęć

- Jest to projekt – nie laboratorium!
- Oznacza to, że studenci mają samodzielnie wykonać postawione zadania w trakcie semestru.
- Wstępem teoretycznym do zajęć był wykład z przedmiotu.
- Zajęcia projektowe dają możliwość skorzystania z DSP do implementacji i testowania algorytmów, ale jest to zbyt krótki czas na wykonanie całości projektu.
- Część projektu, którą można zrealizować bez dostępu do DSP, należy wykonać samodzielnie pomiędzy zajęciami. Dotyczy to np. projektowania filtrów, opracowania koncepcji algorytmu, przeczytania dokumentacji, przygotowania kodu “na sucho” itp.
- Harmonogram projektu może zakładać wykonanie pewnych prac przed przyjściem na zajęcia – będzie to kontrolowane przez prowadzącego.

Przebieg zajęć

- Obecność na zajęciach jest obowiązkowa.
- Na początku zajęć grupa pobiera „swoją” płytkę DSP oraz przewód USB.
- Każda grupa jest zobowiązana posiadać na zajęciach własne słuchawki z wtykiem mini-jack (najlepiej własne słuchawki dla każdej osoby).
- W trakcie zajęć studenci *samodzielnie* realizują zadany temat. Prowadzący zajęcia służy pomocą w odpowiedzi na *konkretne* pytania.
- Aby uzyskać punkty, należy na zakończenie zajęć przedstawić prowadzącemu wyniki realizacji danego etapu. Nie ma możliwości oddawania wyników pracy w późniejszym terminie.
- Należy pamiętać o dokumentowaniu wykonywanych prac w celu umieszczenia ich w końcowym raporcie. Dotyczy to zapisywania projektów filtrów, robienia zrzutów ekranu, notowania wniosków, itp.
- Na zakończenie zajęć należy pamiętać o skopiowaniu katalogu projektu na własny nośnik USB. Utrata plików może spowodować niemożność wykonania projektu.
- Sprzęt należy zwrócić w takiej samej formie, w jakiej został pobrany.

Punktacja za etapy realizacji projektu

- Za realizację etapów projektu na kolejnych zajęciach studenci otrzymują punkty w skali od 0 do 5.
- Nie ma możliwości „odrobienia" zaległych punktów na kolejnych zajęciach.
- Ocenie podlega:
    - zrealizowanie danego etapu wg. harmonogramu,
    - poprawność realizacji i uzyskanych wyników,
    - przygotowanie do zajęć – jeżeli dany etap to zakładał.
- Ocena może zostać zmniejszona na skutek:
    - rażącego braku przygotowania do zajęć,
    - braku odpowiedniej aktywności na zajęciach (np. zajmowania się innymi rzeczami niż projekt),
    - braku umiejętności odpowiedzi na pytania prowadzącego, w przypadku podejrzenia o niesamodzielną pracę,
    - notorycznego spóźniania się na zajęcia.
- Wszelkie wykryte przypadki niesamodzielnej pracy będą skutkowały odejmowaniem punktów. Student realizujący projekt musi wykazać się zrozumieniem zasady działania zaimplementowanego algorytmu.

Oceny i warunki zaliczenia

- W trakcie realizacji projektu można uzyskać do 50 punktów.
- Na ostatnich zajęciach studenci prezentują prowadzącemu efekty projektu i odpowiadają na ew. pytania oraz oddają raport końcowy i kod źródłowy programu.
- Punktacja za poszczególne elementy zajęć jest następująca:
    - 25 pkt. do uzyskania na kolejnych zajęciach (5 x 5 pkt.)
    - 5 pkt. za prezentację końcowych wyników pracy,
    - 10 pkt za raport końcowy,
    - 10 pkt za wykonany kod programu.
- Warunkiem zaliczenia zajęć jest uzyskanie min. 26 punktów.
- W raporcie końcowym ocenie podlega:
    - opisanie wszystkich istotnych etapów projektu,
    - forma i czytelność prezentacji,
    - własne wnioski i komentarze.
- W kodzie programu ocenie podlega:
    - poprawność zaimplementowanego kodu,
    - organizacja kodu,
    - czytelność kodu,
    - komentarze wyjaśniające funkcje bloków kodu,
    - brak niepotrzebnych elementów w kodzie.

ZADANIA PROJEKTOWE

Zajęcia nr 1 – Wprowadzenie do DSP

Przygotowanie do zajęć

- Należy obowiązkowo przynieść na zajęcia własne słuchawki z wtykiem mini-jack.

Zadania do wykonania

- W ramach zajęć studenci zapoznają się z metodą uruchamiania projektów na DSP, zgodnie ze wskazówkami prowadzącego.
- Poruszone tematy:
  - tworzenie projektu w Code Composer Studio,
  - uruchamianie i debugowanie programu,
  - analiza zawartości kodu w języku C,
  - proste modyfikacje kodu,
  - implementacja i uruchomienie prostego filtru FIR z biblioteki DSPLIB,
  - korzystanie z dokumentacji DSPLIB.

Wymagane wyniki: brak.

Zajęcia nr 2 – Projekt i implementacja filtru FIR

Przygotowanie do zajęć

- Osoby, które nie posiadają umiejętności programowania w języku C w stopniu co najmniej podstawowym, powinny samodzielnie opanować podstawy języka:
  - składnia (średniki, itp),
  - typy danych,
  - deklarowanie zmiennych i tablic,
  - wskaźniki,
  - instrukcje warunkowe i pętle,
  - funkcje – definiowanie i wywoływanie,
  - komentarze.
- Przykładowe materiały:
  - https://pl.wikibooks.org/wiki/C
  - książka *Symfonia C++*

Zadania do wykonania

- Każda grupa otrzyma specyfikację projektową trzech filtrów: dolnoprzepustowy, środkowoprzepustowy, górnoprzepustowy.
- Należy wybrać jeden z trzech filtrów do implementacji.
- Za pomocą programu *Matlab* i narzędzia *fdatool* należy zaprojektować wybrany filtr, zgodnie ze specyfikacją. Pamiętać o zapisaniu wykresów i współczynników do późniejszego umieszczenia w raporcie.
- Wyeksportować współczynniki do Matlaba.
- Dokonać konwersji współczynników do formatu Q15, z zaokrągleniem (instrukcja *round*).
- Sprawdzić czy suma współczynników nie powoduje przepełnienia zakresu.
- Przenieść współczynniki do kodu C w formacie rozdzielonym przecinkami (można użyć funkcji *csvwrite* w Matlabie). Zapisać je w stałej tablicy.

- W dokumentacji DSPLIB odnaleźć potrzebną funkcję. Uwaga: jeżeli filtr jest symetryczny, należy użyć funkcji dla filtru symetrycznego.
- Zaimplementować filtrację sygnału z wejścia mikrofonowego w kroku nr 2. Krok 1 powinien przepuszczać sygnał bez zmian. Ustawić odpowiedni tekst na wyświetlaczu (to uwaga również do wszystkich późniejszych kroków).
- Wygenerować szum i podać go na wejście filtru, podstawiając próbki szumu zamiast sygnału z wejścia mikrofonowego. Uwaga: nie zmieniać linii uruchamiającej filtr!
- Zapisać przefiltrowane próbki do bufora.
- Wyświetlić wykres FFT bufora po filtracji.
- Ocenić poprawność działania filtru.

Wymagane wyniki

- Projekt filtru FIR wykonany zgodnie ze specyfikacją.
- Zaimplementowany filtr FIR.
- Wykres widma pokazujący poprawne działanie filtru.

Zajęcia nr 3 – Implementacja i testowanie filtru IIR

Przygotowanie do zajęć

- Zaprojektować filtr IIR zgodnie ze specyfikacją. Uwaga: należy wybrać inny typ filtru niż wykorzystany w filtrze FIR.
- Wyeksportować współczynniki do Matlaba.
- Dokonać normalizacji współczynników tak aby nie występowały przepełnienia lub inne błędy. Opis procedury można znaleźć w prezentacji z wykładu oraz w materiałach pomocniczych (*Normalizacja wzmocnienia kaskadowego filtru IIR*).
- Dokonać konwersji współczynników na format Q15.
- Sprawdzić stabilność i maksymalne wzmocnienie filtru.
- Wyeksportować współczynniki do kodu C.

Zadania do wykonania

- Odnaleźć w dokumentacji DSPLIB optymalną funkcję dla filtru IIR.
- Zaimplementować filtr IIR w kroku nr 3.
- Dokonać jego testowania w podobny sposób jak dla filtru FIR.
- Jeżeli wystąpi przepełnienie zakresu, skorygować wzmocnienie filtru.

Wymagane wyniki

- Projekt filtru IIR wykonany zgodnie ze specyfikacją, wykonany przed zajęciami.
- Zaimplementowany filtr IIR.
- Wykres widma pokazujący poprawne działanie filtru.

Zajęcia nr 4 – Implementacja algorytmu FFT

Przygotowanie do zajęć

- Obliczyć współczynniki okna Hamminga dla wybranego rozmiaru okna (takiego, które pozwoli obliczyć FFT z jego udziałem). Dokonać konwersji współczynników na format Q15 (uwaga na przepełnienie zakresu) i zapisać je w kodzie C.
- Zapoznać się z dokumentacją DSPLIB odnośnie analizy FFT.
- Przygotować koncepcję działania algorytmu.

Zadania do wykonania

- Zapoznać się z punktem „*Konfiguracja projektu do obliczania FFT*” w dokumencie „*Konfiguracja nowego projektu w CCS*” (materiały pomocnicze).
- Obliczyć widmo amplitudowe próbek pobranych z wejścia mikrofonowego, używając stosownych operacji. Kod zapisać w kroku nr 4. UWAGA: aby uzyskać maksymalną liczbę punktów, należy wykonać wszystkie obliczenia za pomocą pojedynczego bufora!
- Wyświetlić obliczone widmo używając funkcji wykresu czasowego.
- Sprawdzić poprawność analizy używając sygnału testowego zapisanego w pliku *testsignal.h* i podstawiając pojedyncze próbki tego sygnału zamiast próbek z wejścia mikrofonowego.
- Dodać okno Hamminga. Porównać dokładność obliczonego widma z oknem i bez. Przedyskutować różnice w późniejszym raporcie.

Wymagane wyniki

- Obliczenie współczynników okna Hamminga, wykonane przed zajęciami.
- Zaimplementowany algorytm obliczania widma amplitudowego.
- Wykres widma pokazujący poprawne działanie algorytmu dla testowego sygnału, z oknem i bez.

Zajęcia nr 5 – Implementacja algorytmu detekcji częstotliwości

Przygotowanie do zajęć

- Opracować koncepcję algorytmu, który za pomocą analizy FFT wykryje częstotliwość pierwszego istotnego maksimum widmowego („pierwszego prążka”). Uwaga: nie zakładać, że pierwsze maksimum będzie miało największą amplitudę. Wziąć też pod uwagę możliwość występowania szumu w sygnale.

Zadania do wykonania

- Zaprezentować prowadzącemu opracowaną koncepcję algorytmu.
- Zaimplementować przygotowany algorytm w kroku nr 4, rozszerzając kod napisany na poprzednich zajęciach.
- Przetestować działanie algorytmu na sygnale z pliku *testsignal.h*. Sprawdzić poprawność działania algorytmu.
- Zastanowić się z czego wynika ew. różnica otrzymanego wyniku w stosunku do oczekiwanej wartości 1225 Hz i opisać to w raporcie.
- Przetestować działanie algorytmu przy podawaniu różnych sygnałów typu harmonicznego na wejście mikrofonowe.

Wymagane wyniki

- Opracowanie koncepcji algorytmu, wykonane przed zajęciami.
- Zaimplementowany algorytm detekcji częstotliwości – wartość zmiennej wyświetlana w oknie debugowania.
- Poprawna detekcja częstotliwości, również w przypadku obecności szumu.

Zajęcia nr 6 – Testowanie algorytmów

Przygotowanie do zajęć

- Dodać do kodu stałą globalną (lub dyrektywę #define), która decyduje o tym czy do analizy używamy sygnału z wejścia DSP, czy „podstawionego" sygnału.
- Zmodyfikować kod tak, aby w zależności od wartości tej stałej analizował wybrany sygnał. Zmiana wartości tej stałej w kodzie (i tylko ta zmiana) powinna pozwolić na wybór: sygnał z wejścia lub sygnał zapisany w kodzie.

Zadania do wykonania

- Przetestować szczegółowo zaimplementowane algorytmy (filtry, detekcja częstotliwości) przy użyciu różnych sygnałów testowych.
- Zaimplementować generowanie sinusa. Częstotliwość (w Hz) powinna być zapisana w kodzie jako stała (którą można zmienić przed kompilacją programu).
- Przetestować działanie filtrów za pomocą sinusa o różnych częstotliwościach (w paśmie przepustowym i zaporowym). Wykonać i zapisać wykresy czasowe i widmowe wyników filtracji.
- Zaimplementować generowanie sygnału piłokształtnego o częstotliwości 500 Hz. Przetestować działanie filtrów za pomocą tego sygnału.
- Wykorzystać czas na dopracowanie algorytmów oraz testowanie przy użyciu różnych sygnałów podawanych na wejście mikrofonowe.

Wymagane wyniki

- Prezentacja poprawności działania algorytmów na sygnałach generowanych w kodzie (wykresy sygnałów po przetworzeniu przez algorytm).

Zajęcia nr 7 – Zaliczenie projektu

Przygotowanie do zajęć

- Przygotować kod do analizy sygnału podawanego na wejście DSP.
- Oczyścić kod ze zbędnych elementów. Sformatować go tak aby był czytelny. Upewnić się, że działanie głównych bloków programu jest opatrzone komentarzami (nie chodzi o komentowanie każdej osobnej instrukcji).
- Upewnić się, że poszczególne kroki mogą być wybrane po kolei za pomocą przycisków, a wyświetlacz prezentuje opis danego kroku.
- Przygotować raport końcowy.

Zadania do wykonania

- Zaprezentować prowadzącemu i omówić działanie poszczególnych algorytmów na sygnale podanym (przez prowadzącego) na wejście mikrofonowe.
- Odpowiedzieć na pytania prowadzącego.
- Oddać raport.