УДК 621.396.2

Гринкевич Г. О. *(Державний університет інформаційно-комунікаційних технологій)*

СТВОРЕННЯ АЛГОРИТМУ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є ДЛЯ АПАРАТНОЇ РЕАЛІЗАЦІЇ MIMO-МОДЕМУ

Гринкевич Г. О. Створення алгоритму швидкого перетворення Фур'є для апаратної реалізації MIMO-модему. Проведена розробка алгоритмів керуючих програм прямого та зворотнього швидкого перетворення Фур'є для MIMO-модему передачі даних на 1024 вхідних/вихідних відліків. Апаратна реалізація модему орієнтована на застосування сучасних та потужних цифрових сигнальних процесорів, що дозволяє значно зменшити час обчислення операцій та збільшити швидкість передачі даних MIMO-системи.

Ключові слова: МОДЕМ, СИГНАЛЬНИЙ ПРОЦЕСОР, ПЕРЕТВОРЕННЯ ФУР'Є, MIMO

Гринкевич А. А. Создания алгоритма быстрого преобразования Фурье для аппаратной реализации MIMO-модема. Проведена разработка алгоритмов управляющих программ прямого и обратного быстрого преобразования Фурье для MIMO-модема передачи данных на 1024 входящих/исходящих отсчетов. Аппаратная реализация модема ориентирована на применение современных и мощных цифровых сигнальных процессоров, что позволяет значительно уменьшить время вычисления операций и увеличить скорость передачи данных MIMO-системы.

Ключевые слова: МОДЕМ, СИГНАЛЬНЫЙ ПРОЦЕССОР, ПРЕОБРАЗОВАНИЕ ФУРЬЕ, MIMO

Grynkevych G. O. Creations algorithm fast Fourier transformation for hardware representation of MIMO-modem. Development algorithms of control program is conducted direct and reverse fast Fourier transform for MIMO-modem of communication data on a 1024 incoming/outgoing counting out. Hardware representation of modem is oriented to application modern and powerful digital signal processors, that allows considerably to decrease time of calculation of operations and rev up communication of data of MIMO-system.

Key words: MODEM, SIGNAL PROCESSOR, FOURIER TRANSFORMATION, MIMO

Вступ. Важливим інструментом для підвищення фізичної швидкості та вірогідності передачі даних у безпровідних мережах є розширення смуги пропускання спектральних каналів. В [1, 2] показано, що найбільша ефективність передачі даних здійснюється в каналах з ортогональним частотним мультиплексуванням (ОЧМ).

В представленій статті проведена розробка алгоритмів ОЧМ-модуляторів/демодуляторів із застосуванням сучасних та потужних цифрових сигнальних процесорів (ЦСП) серії ADSP-21160 фірми Analog Devices. Запропоновані алгортми дозволяють значно зменшити час обчислення операцій, смугу частот, що займає сигнал та збільшити швидкість передачі даних в системах технології MIMO (Multiple Input Multiple Output – множинний вхід множинний вихід). Для створення керуючої програми MIMO-модему використано програмне середовище Analog Devices VisualDSP++ [2...4], яке призначене для розробки і відладки цифрових сигнальних процесорів серії ADSP-21160.

Розробка алгоритмів перетворення Фур'є. Блок-схема MIMO-модему на основі ОЧМ із використанням прямого (ШПФ) та зворотнього (ЗШПФ) швидкого перетворення Фур'є показана на рис. 1.

Спосіб ОЧМ має такі переваги:

• велика спектральна ефективність модуляції завдяки максимально близькому розташуванню субканалів;

• висока завадостійкість інформаційного сигналу завдяки використанню великої кількості несучих в широкому діапазоні частот;

• можливість реалізації повністю цифрових високо-

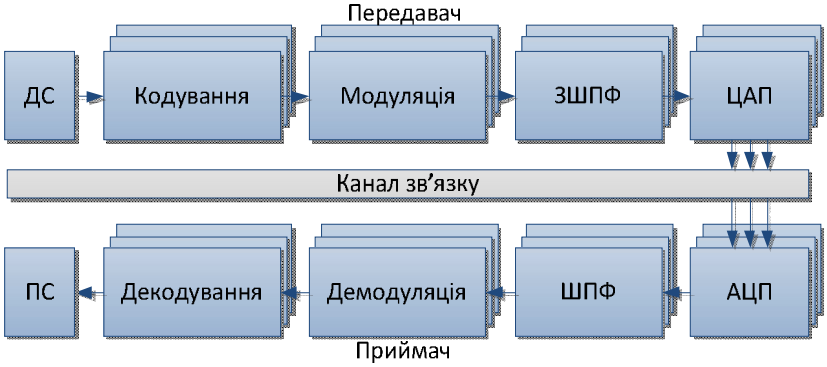

Рис. 1. Блок-схема MIMO-модему

ефективних алгоритмів модуляції-демодуляції, основаних на використанні дискретного перетворення Фур'є (ДПФ).

Всі вищенаведені переваги ОЧМ стали можливими завдяки апаратній реалізації ЗШПФ та ШПФ. Переваги системи ОЧМ проявляються при великій кількості несучих. Без такого кроку неможлива реалізація ОЧМ, адже в іншому випадку пряме апаратне формування ОЧМ-сигналу вимагало б величезних схемотехнічних витрат у вигляді тисяч генераторів і модуляторів в передавачі і такого ж числа детекторів в приймачі. Маловірогідно, що така схема була б реалізована.

Реалізація ЗШПФ та ШПФ [1, 5] базується на формулах (1) і (2), відповідно:

$$x(n)=\frac{1}{N}\sum_{m=0}^{N-1}X(m)e^{j2\pi nm/N} \qquad \textbf{(1),} \qquad X(m)=\frac{1}{N}\sum_{n=0}^{N-1}x(n)e^{-j2\pi nm/N} \qquad \textbf{(2),}$$

де m – індекс ДПФ в частотній області; n – індекс ДПФ в часовій області; N – кількість вхідних/вихідних відліків ДПФ.

Однак, оскільки значення $e^{j2\pi nm/N}$ з (1) та $e^{-j2\pi nm/N}$ з (2) є постійними для заданого числа N, то вираз $X(m)=\sum_{n=0}^{(N/2)-1}x(2n)\left(W_{N/2}\right)^{nm}+\left(W_N\right)^m\sum_{n=0}^{(N/2)-1}x(2n+1)\left(W_{N/2}\right)^{nm}$ можна обчислити і використовувати готові значення при створенні програмного забезпечення конкретного MIMO-модему передачі даних ($W_N = e^{j2\pi/N}$). Використання даної можливості та розбиття N вхідних/вихідних відліків ДПФ на дві та більше частини дозволяє зменшити кількість обчислювальних операцій до $(N/2)\log_2 N$ [6].

У загальному випадку, вимоги по використовуваній пам'яті для N-точкового ШПФ такі: N комірок для дійсних та N комірок для уявних даних і N комірок для коефіцієнтів повороту.

Як показано в [2, 3, 6], результати порівняння реалізації алгоритмів ШПФ по основі 2 на різних процесорах такі:

- ADSP-2189M (16 розрядів, фіксована точка, 453 мкс, 1024 точки ШПФ);
- ADSP-21160 (32 розряди, плаваюча точка, 90 мкс, 1024 точки);
- ADSP-TS001 (16 розрядів, режим з фіксованою точкою, 7,3 мкс, 256 точок); або (32 розряди, режим з плаваючою точкою, 69 мкс, 1024 точки).

Для забезпечення функціонування в реальному часі повний розрахунок ШПФ повинен виконуватися в проміжку, що відповідає часу накопичення одного пакету даних. Передбачається, що, поки робиться обчислення ШПФ поточного пакету даних, ЦСП накопичує дані для наступного пакету. Безперервне отримання даних полегшується завдяки можливостям гнучкої адресації даних в ЦСП у поєднанні з використанням різних каналів прямого доступу до пам'яті.

Розглянемо ЦСП процесор ADSP-21160, який обчислює 1024-точкове 32-розрядне комплексне ШПФ з плаваючою точкою за 90 мкс. Очевидно, що максимальна частота дискретизації дорівнює 1024/90 = 11,38 Мо/с. Тут сигнал має ширину смуги частот менше 5,7 МГц. Також передбачається, що немає додаткових витрат процесорного часу, пов'язаних з ШПФ, або обмежень, пов'язаних з передачею даних.

Наведений приклад дає оцінку максимальної ширини смуги сигналу, який може бути оброблений даним ЦСП з урахуванням характеристик реалізованого на ньому ШПФ. Число точок ШПФ також визначає мінімальний рівень шуму ШПФ відносно рівня широкосмугового шуму, і це також має бути враховано при виборі числа точок ШПФ. На рис. 2 представлені співвідношення між рівнем сигналу, що відповідає повному динамічному діапазону системи, рівнем широкосмугового шуму (виміряного в ширині смуги від 0 до $f_s/2$) і мінімальним рівнем шуму ШПФ.

Як вже говорилося вище, інтегроване середовище VisualDSP++ є основним засобом розробки і відладки додатків для процесорів компанії Analog Devices і підтримує процесори типу TigerSHARC, SHARC і Blackfin. Середовище VisualDSP++ може бути завантажене з веб-сайту компанії та після закінчення реєстрації користувачеві висилається серійний номер, що дає можливість працювати з пакетом в тестовому режимі впродовж 90 днів [3].

Синтез керуючої програми. Перейдемо до опису процесу складання проекту керуючої програми у VisualDSP++ [3...5]. Найбільш типовий процес роботи над проектом в цьому середовищі полягає в наступному.

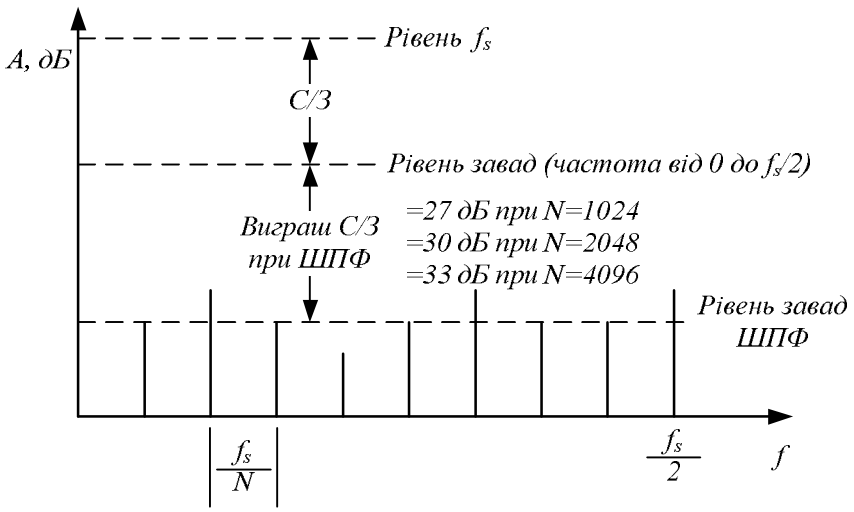

Рис. 2. Виграш у відношенні сигнал/завада при ШПФ

На першому етапі користувач створює файл проекту (.dpj) і настроює його параметри (тип процесора, налаштування асемблера, компілятора, компонувальника і т. д.).

Наступний етап полягає в додаванні в проект існуючих або створенні нових початкових файлів. Початкові файли програми можуть бути написані на мові асемблера процесора (що мають розширення .asm, .s або .dsp) і/або високорівневих мовах програмування C/C++ (.c/.cpp/.cxx). Обидва ці варіанти мають свої переваги та недоліки і недоліки. На рис. 3 показано вікно завантаженої програми, як частини проекту.

ifft

```
#include <complex.h>
#include <iostream>
#include <fstream>

#include "idft.h"

using namespace std;

namespace SPUC {
void ifft(complex<double> *y, int m)
{
  complex<double> u,temp,tm;
  int wptr;
  int i,j,k,l,le,windex;
  float scale;

  /*  n = 2**m = inverse fft length */
  int n = 1 << m;
  le = n/2;
  complex<double> *w = new complex<double> [n+1];
  complex<double> *x  = new complex<double> [n+1];

  /*  calculate the w values recursively */
  complex<double> w_inc = expj(TWOPI/n);
  complex<double> w_x = w_inc;
```

Рис. 3. Частина проекту реалізації MIMO-модему передачі даних на основі ЦСП

Також на цьому етапі в проект можуть бути додані (якщо це не було зроблено автоматично за допомогою Project Wizard) два допоміжні файли: файл *run-time* заголовка і файл опису лінкера.

Коли усі початкові і допоміжні файли готові, можна приступити до збірки проекту. Цей процес відбувається в два кроки. Спочатку початкові файли і CRT піддаються компіляції і асемблюванню за допомогою утиліт «компілятор» і «асемблер», відповідно. В результаті формується один або декілька об'єктних файлів в стандартному форматі ELF, які мають розширення .doj і містять в собі структуровані секції коду і даних. Потім викликається утиліта «лінкер» (компонувальник), яка на підставі інформації, що міститься у файлі опису лінкера (.ldf), аналізує об'єктні файли, що підключаються користувачем бібліотеки (.dlb) і визначає, які з об'єктів необхідно помістити в той або інший сегмент доступної процесору внутрішньої або зовнішньої пам'яті. Результатом роботи лінкера є виконуваний файл у

форматі ELF з розширенням .dxe. В деяких випадках лінкер також формує файли вмісту спільно використовуваної пам'яті (для проектів, що виконуються у багатопроцесорних системах) з розширенням .sm і файли оверлейної пам'яті (для систем, в яких окремі незалежні частини коду оперативно підвантажуються із зовнішньої пам'яті у внутрішню пам'ять процесора) з розширенням .ovl.

Усі необхідні для формування виконуваного файлу утиліти запускаються системою VisualDSP++ автоматично. Як правило, на цьому етапі збірка проекту відбувається в конфігурації Debug, в якій відключена оптимізація компілятора і в код додаються можливості відладки. У разі успішної збірки проекту користувач настроює налагоджувальну сесію, після чого виконуваний файл завантажується через цільовий об'єкт в програмну модель процесора або в реальний процесор для відлагодження.

Коли проект успішно проходить функціональне відлагодження, він може бути повторно зібраний в конфігурації Release. Ця конфігурація дає оптимізований по продуктивності код без можливостей відлагодження (чи з обмеженими можливостями). Працездатність отриманого в результаті збірки виконуваного файлу необхідно повторно досліджувати у складі платформи. Оскільки формат виконуваного файлу не годиться для безпосереднього завантаження в процесор в автономно працюючій системі, він має бути перетворений утилітою loader в завантажуваний образ (.ldr). Отриманий в результаті роботи завантажувача/сплітера файл може бути записаний в мікросхему пам'яті на платі EZ-KIT/EZ–BRD [3, 5] або на платі власної розробки за допомогою утиліти Flash Programmer.

Головною виконуючою ланкою проекту реалізації MIMO-модему передачі даних на основі ЦСП є алгоритм та, як результат, програма побудови ЗШПФ та ШПФ.

Робота алгоритму. На рис. 4 показано алгоритм ЗШПФ.

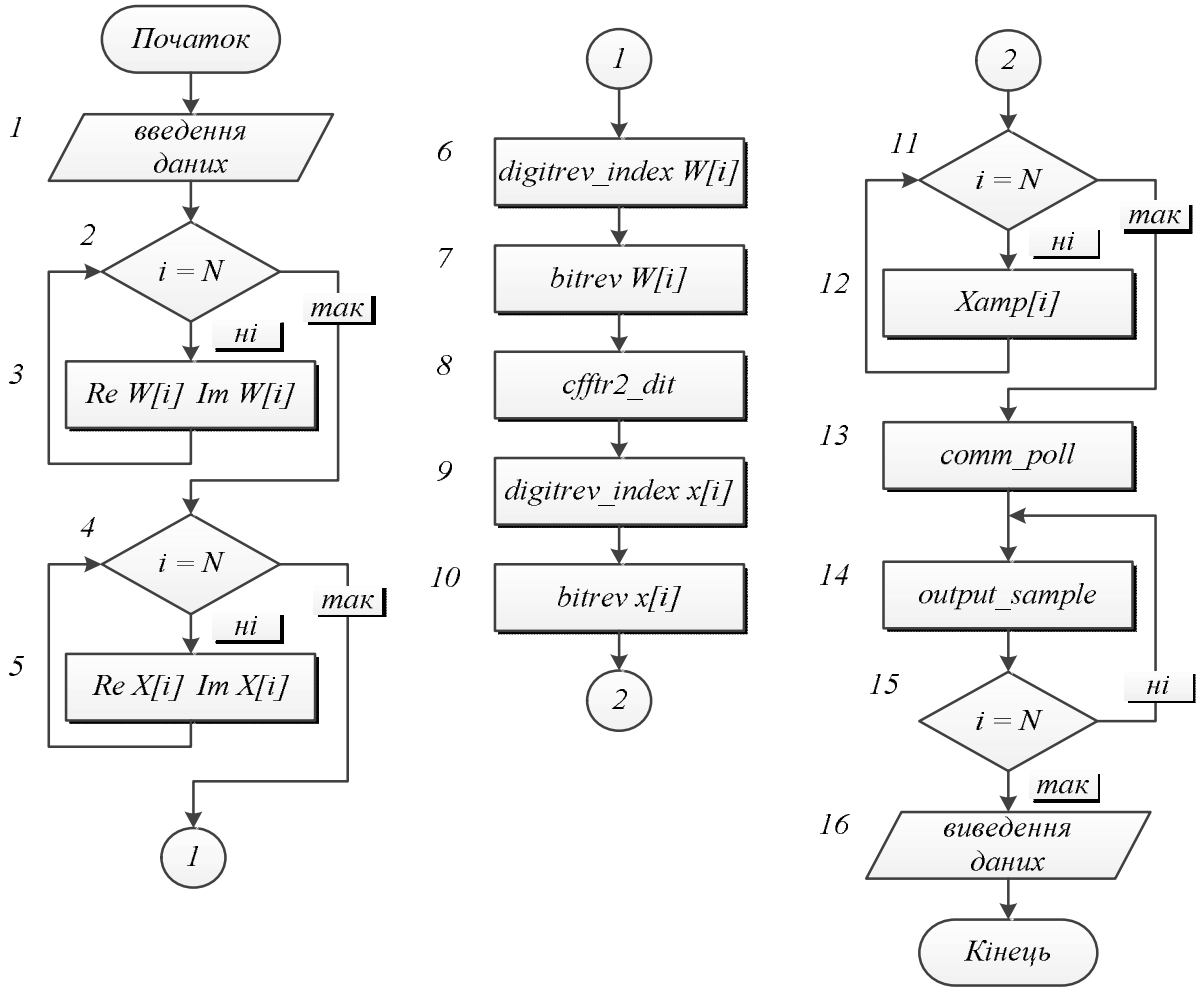

Рис. 4. Алгоритм ЗШПФ для 1024 вихідних відліків

Перший блок в алгоритмі відповідає за введення вхідних даних та визначення характеристик операндів, які будуть використовуватись в подальших розрахунках.

В блоках 2 та 3 виконуються операції обчислення дійсної та уявної складових комплексних повертаючих множників *W*, які дорівнюють $e^{j2\pi/N}$ і є постійними при заданому *N* [6, 7] (в нашому випадку *N*=1024). Комплексна та уявна складові вхідного сигналу *X(i)* виділяються за допомогою блоків 4 та 5. Варто зазначити, що операції розрахунку виконуються до моменту отримання рівності *i=N*. В шостому алгоблоці виконується цифрова обробка повертаючих множників *W* за основою 2 для їх подальшої біт-реверсивної перестановки (7 блок). Суть операції біт-реверсивної перестановки полягає в розбитті вхідної послідовності відліків на парні та непарні відліки. 8-й блок призначений для виконання операції ЗШПФ над комплексними числами за основою 2.

В блоках 9 та 10 проходять аналогічні операції, що й у алгоблоках 6 та 7, тільки для вихідних відліків *X(i)*. Операція обчислення амплітуди всіх 1024 вихідних відліків *X(i)* виконується в 11 та 12 блоках. 13-й блок виконує функцію попередньої підготовки до формування вихідного сигналу та підключення периферії на можливість виведення даних. 14-й і 15-й блоки відповідають за формування вибірки вихідного сигналу з 1024 субканалів та додаткову перевірку вибірки на наявність усіх 1024 субканалів. Виведенням даних завершується алгоритм.

Висновки. В роботі проведено удосконалення алгоритму функціонування MIMO-модему передачі даних на 64 відліки, а саме: розроблено підпрограму реалізації ЗШПФ та ШПФ з кількістю вхідних/вихідних відліків, рівною 1024. В якості пристрою обробки даних застосовано ЦСП ADSP-21160, що має кращі характеристики в порівняні з мікроконтролером ATmega128. Частотний діапазон, який займають сигнали згідно виразів (1) і (2) складає 8 МГц. Ця смуга частот найбільш наближена до параметрів стандарту передачі даних 802.16e. Тому дослідження проводились в цій смузі частот і показали ефективність обраного шляху реалізації програмного забезпечення та схемотехнічних рішень.

Література

1. Dahlman E., Parkvall S., Skold J. 4G: LTE/LTE-Advanced for Mobile Broadband. – London: ELSEVIER, 2010. – 431 p.

2. Jeffrey G. A., Arunabha G., Rias M. Fundamentals of WiMAX: Understanding Broadband Wireless Networking. – Washington: Pearson Education, Inc., 2007. – 478 p.

3. VisualDSP++ Development Software Release 5.0 [Електронний ресурс] // – Режим доступу : http://www.analog.com

4. Проектирование с использованием процессоров Analog Devices [Електронний ресурс] // – Режим доступу : http://kit-e.ru

5. Вальпа О. Д. Разработка устройств на основе цифровых сигнальных процессоров фирмы Analog Devices с использованием Visual DSP++ / О. Д. Вальпа. – М.: Горячая линия-Телеком, 2007. – 270 с.

6. Лайонс Р. Цифровая обработка сигналов / Р. Лайонс ; пер. с англ. – [2-е изд.]. – М.: ООО «Бином-Пресс», 2006 г. – 656 с.

7. Проектирование систем цифровой и смешанной обработки сигналов ; под ред. Уолта Кестера ; пер. с англ. – М.: Техносфера, 2010. – 328 с.