

4. Солоха А.А. Математическая модель шагового двигателя / А.А. Солоха // Матер. XVII междунар. научн. конф «Математические методы в технике и технологиях», Кострома, 2004. – Т.5.

5. Нейдорф Р.А. Задачи квазиоптимального быстрогодействия управления шаговым двигателем / Р.А. Нейдорф // Матер. XVIII междунар. научн. конф. «Математические методы в технике и технологиях», Кострома, 2005. – Т.2.

6. Зайцев Г.Ф. Теорія автоматичного управління / Г.Ф. Зайцев, В.К. Стеклов, О.І. Брицький ; під ред. проф. Зайцева Г.Ф. – К. Техніка. 2002. – 688 с.

7. Труш О.В. Аналіз можливостей підвищення якості сервосистем управління накопичувачів на жорстких магнітних і оптичних дисках / О.В. Труш // Збірник наукових праць ВІКНУ. –2010. – № 26.

УДК 621.396.2

Макаренко А.О., к.т.н. (Державний університет інформаційно-комунікаційних технологій)

РОЗРОБКА OFDM-АЛГОРИТМУ ДЛЯ РЕАЛІЗАЦІЇ СУЧАСНИХ БЕЗПРОВОДОВИХ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Макаренко А.О. Розробка OFDM-алгоритму для реалізації сучасних безпроводових телекомунікаційних систем. Проведена розробка алгоритмів керуючих програм прямого та зворотного швидкого перетворення Фур'є для безпроводного модему передачі даних на 1024 вхідних/вихідних відліків. Апаратна реалізація модему орієнтована на застосування потужних цифрових сигнальних процесорів і дозволяє суттєво зменшити час обчислення операцій та збільшити швидкість передачі даних безпроводної системи.

Ключові слова: OFDM, МОДЕМ, СИГНАЛЬНИЙ ПРОЦЕСОР, ПЕРЕТВОРЕННЯ ФУР'Є.

Макаренко А.А. Разработка OFDM-алгоритма для реализации современных беспроводных телекоммуникационных систем. Проведена разработка алгоритмов управляющих программ прямого и обратного быстрого превращения Фурье для беспроводного модема передачи данных на 1024 входных/выходных отсчетов. Аппаратная реализация модема ориентирована на применение мощных цифровых сигнальных процессоров и позволяет существенно уменьшить время вычисления операций и увеличить скорость передачи данных беспроводной системы.

Ключевые слова: OFDM, МОДЕМ, СИГНАЛЬНЫЙ ПРОЦЕССОР, ПРЕОБРАЗОВАНИЕ ФУРЬЕ.

Makarenko A.O. Development of OFDM-algorithm is for realization of modern wireless of the telecommunication systems. Conducted development of algorithms of control program of direct and reverse rapid Fourier transformation for the off-wire modem of communication of data on 1024 input-output counting out. Hardware representation of modem is oriented to application of powerful digital alarm processors and allows considerably to decrease time of calculation of operations and rev up communication of data of the off-wire system.

Keywords: OFDM, MODEM, SIGNAL PROCESSOR, FOURIER TRANSFORMATION

Вступ. Важливим інструментом для підвищення фізичної швидкості та достовірності передачі даних у безпроводних мережах є розширення смуги пропускання спектральних каналів. В [1...3] показано, що найбільша ефективність передачі даних здійснюється в каналах з ортогональним частотним мультиплексуванням (ОЧМ, OFDM).

В представленій статті проведена розробка алгоритмів ОЧМ-модуляторів/демодуляторів із застосуванням сучасних потужних цифрових сигнальних процесорів (ЦСП) серії ADSP-21160 фірми Analog Devices. Запропоновані алгоритми дозволяють значно зменшити час обчислення операцій, смугу частот, що займає сигнал та збільшити швидкість передачі даних в безпроводних системах, орієнтованих в першу чергу для передачі даних. Для створення керуючої програми модему використано програмне середовище Analog Devices Visual DSP++ [2, 4...6], яке призначене для розробки і від лагодження пристроїв на основі цифрових сигнальних процесорів серії ADSP-21160.

Розробка OFDM-алгоритмів. Блок-схема безпроводного OFDM-модему на основі ОЧМ із використанням прямого (ШПФ) та зворотного (ЗШПФ) швидкого перетворення Фур'є

представлена в [1, 2, 8]. Спосіб ОЧМ характеризується рядом переваг, а саме[3]: 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 високоефективних алгоритмів модуляції-демодуляції, основаних на використанні дискретного перетворення Фур'є (ДПФ).

Всі вищенаведені переваги ОЧМ стали можливими завдяки апаратній реалізації ЗШПФ та ШПФ. Найбільш суттєво 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:

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

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

Однак, оскільки значення $e^{j2\pi n/N}$ з (1) та $e^{-j2\pi n/N}$ з (2) є постійними для заданого числа N , то вираз $X(m) = \sum_{n=0}^{(N/2)-1} x(2n)(W_{N/2})^{nm} + (W_N)^m \sum_{n=0}^{(N/2)-1} x(2n+1)(W_{N/2})^{nm}$ можна обчислити і використовувати готові значення при створенні програмного забезпечення конкретного МІМО-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 величини $(N/2)\log_2 N$ [7].

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

Як вже говорилося вище, інтегроване середовище Visual DSP++ є основним засобом розробки додатків для процесорів компанії Analog Devices і підтримує процесори типу Tiger SHARC, SHARC і Blackfin. Середовище Visual DSP++ може бути завантажено з веб-сайту

компанії та після закінчення реєстрації користувачеві висилається серійний номер, що дає можливість працювати з пакетом в тестовому режимі впродовж 90 днів [4].

Синтез керуючої програми. Перейдемо до опису процесу складання проекту керуючої програми у Visual DSP++ [4...6]. Послідовність операцій відображена на рис. 1. Найбільш типовий процес роботи над проектом в цьому середовищі полягає в наступному.

На першому етапі користувач створює файл проекту (.dprj) і настраює його параметри (тип процесора, налаштування асемблера, компілятора, компоувальника 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. 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.



Рис. 1. Етапи роботи над проектом

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 .obj і містять в собі структуровані секції коду і даних. Потім викликається утиліта «лінкер» (компоувальник), 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 оперативного підвантажуються із зовнішньої пам'яті у внутрішню пам'ять процесора) з розширенням .ovl.

Усі необхідні для формування виконувального 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 відлагодження. У разі успішної збірки проекту користувач настраює 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 або з обмеженими можливостями відлагодження. Працездатність отриманого в результаті збірки виконувального файлу необхідно повторно досліджувати у складі платформи.

Оскільки формат виконувального 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 [4, 6] або на платі власної розробки за допомогою утиліти Flash Programmer, яка призначена для внутрішньо-схемного програмування мікросхем пам'яті безпосередньо з середовища Visual DSP++.

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

Робота алгоритму ЗШПФ. Проведемо опис роботи алгоритму ЗШПФ, який показаний на рис. 2.

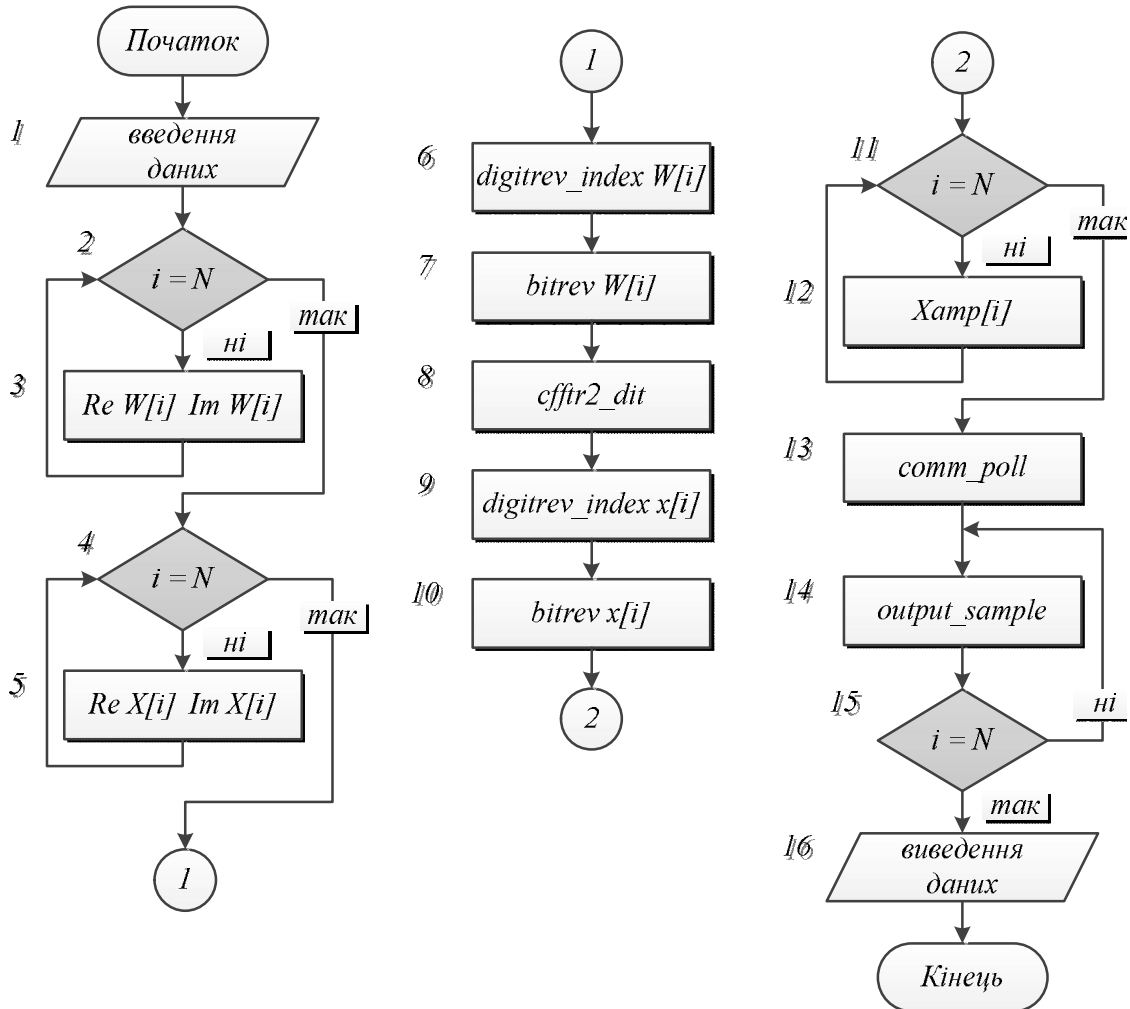


Рис. 2. Алгоритм ЗШПФ для 1024 вихідних відліків

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 [3, 8] (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 6-му алгоблоці виконується цифрова обробка повертаючих множників W за основою 2 для їх подальшої біт-реверсивної перестановки (7-й блок). Суть операції біт-реверсивної перестановки полягає в розбитті вхідної послідовності відліків на парні та непарні відліки.

Блок 8 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 безпроводного модему передачі даних на 64 відліки [3, 8], 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 порівнянні 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. Макаренко А. О. Багатоканальна телекомунікаційна система на основі мережі електроживлення: дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.12.02 – Телекомунікаційні системи та мережі / А. О. Макаренко. – К.: ДУІКТ, 2011. – 156 с.
4. Visual DSP++ Development Software Release 5.0 [Електронний ресурс] // – Режим доступу : <http://www.analog.com>
5. Проектирование с использованием процессоров Analog Devices [Електронний ресурс] // – Режим доступу : <http://kit-e.ru>
6. Вальпа О.Д. Разработка устройств на основе цифровых сигнальных процессоров фирмы Analog Devices с использованием VisualDSP++ / О.Д. Вальпа. – М.: Горячая линия-Телеком, 2007. – 270 с.
7. Лайонс Р. Цифровая обработка сигналов / Р. Лайонс ; пер. с англ. – [2-е изд.]. – М.: ООО «Бином-Пресс», 2006 г. – 656 с.
8. Гринкевич Г.О. Розробка імітаційної моделі та алгоритмів функціонування МІМО-системи / Г. О.Гринкевич, А.О.Макаренко // Зв'язок. – 2012. – №2. – С. 29-33.

УДК 62-231.3.:621.313.8

Шворов С.А., д.т.н. (Нац. унів.-т біоресурсів і природокористування України)

МЕТОД ОПТИМАЛЬНОГО УПРАВЛІННЯ УНІВЕРСАЛЬНИМИ РОБОТАМИ-МАНІПУЛЯТОРАМИ

Шворов С.А. Метод оптимального управління універсальними роботами-маніпуляторами. Пропонується метод управління універсальними роботами-маніпуляторами, за допомогою якого в інтелектуальній системі управління забезпечується виконання найбільшої кількості оперативних завдань при обмеженнях на часові витрати.

Ключові слова: РОБОТ-МАНІПУЛЯТОР, СИСТЕМА УПРАВЛІННЯ. ОПЕРАТИВНЕ ЗАВДАННЯ

Шворов С.А. Метод оптимального управления универсальными роботами-манипуляторами. Предлагается метод управления универсальными роботами-манипуляторами, с помощью которых в интеллектуальной системе управления обеспечивается выполнение наибольшего количества оперативных заданий при ограничениях на временные затраты.

Ключевые слова: РОБОТ-МАНИПУЛЯТОР, СИСТЕМА УПРАВЛЕНИЯ. ОПЕРАТИВНОЕ ЗАДАНИЕ