

РЕАЛІЗАЦІЯ ЦИФРОВИХ ФІЛЬТРІВ НА МІКРОКОНТРОЛЕРАХ STM32 З ВИКОРИСТАННЯМ КІЛЬЦЕВИХ БУФЕРІВ

доцент, к.т.н. Зубков О.В., студент Яковенко О.С.

Харківський національний університет радіоелектроніки,
кафедра мікропроцесорних технологій і систем, м. Харків, Україна
e-mail: oleh.zubkov@nure.ua, oleksandr.iakovenko@nure.ua

Abstract. The analysis of digital filters in electronics and their software implementation on general-purpose microcontrollers was completed in this work. A real-time digital filtering algorithm has been developed for STM32 microcontrollers with support for DSP instructions and the use of ring buffers. A study of the proposed algorithm performance was carried out using the example a digital low-pass filter with a finite impulse response implementation. A comparative analysis of performance measurement results and comparisons with the results of using the standard library proved the effectiveness of the proposed algorithm.

Ключові слова: цифровий фільтр, мікроконтролер, буфер, швидкодія.

Вступ. Наразі розробка цифрових фільтрів на мікроконтролерах залишається актуальною та корисною в багатьох сферах. Цифрові фільтри застосовуються для обробки сигналів у реальному часі в різноманітних пристроях, від медичних приладів до комунікаційних систем. В контексті "Інтернету речей" (IoT) цифрові фільтри застосовуються для фільтрації шумів, видалення спотворень та покращення якості отриманих сигналів. В медичних пристроях, наприклад, в електрокардіографах (ЕКГ) чи електроенцефалографах (ЕЕГ), використання цифрових фільтрів дозволяє відокремлювати корисний сигнал від шумів. У бездротових комунікаційних системах, таких як Wi-Fi, Bluetooth, цифрові фільтри можуть використовуватися для покращення якості сигналу та підвищення швидкості передачі даних. Враховуючи зростаючі можливості мікроконтролерів та їх обробки, розробка цифрових фільтрів на мікроконтролерах залишається актуальною та значущою у багатьох сферах, де потрібна обробка сигналів у реальному часі.

Основна частина. Усі сучасні процесори за їх апаратними можливостями реалізації цифрової фільтрації умовно можна поділити на 2 групи: спеціалізовані та загального призначення. Спеціалізовані процесори, такі як Shark, мають високу вартість і їх використовують при створенні професійної віщальної, локаційної, навігаційної і т.д. апаратури. Менші за ціною процесори загального призначення, такі як STM32, завдяки вбудованому DSP модулю можуть бути використані для реалізації цифрових фільтрів [1, 2]. Математичні підходи до синтезу цифрових фільтрів дуже добре відомі і описані у сучасній літературі [3]. Такі математичні пакети, як Matlab, мають навіть вбудовані спеціалізовані

утиліти для візуального проектування цифрових фільтрів [4]. Основною проблемою реалізації цифрових фільтрів на мікроконтролерах загального призначення є необхідність виконувати багато інших різноманітних задач крім цифрової фільтрації. Якщо для зберігання коефіцієнтів фільтру, відліків сигналу, що обробляється, і результатів фільтрації достатньо незначного об'єму оперативної пам'яті (близько 1 кбайт при порядках фільтра до 100), то об'єм математичних операцій значний і необхідно мати значну швидкодію процесора для втілення алгоритму цифрової фільтрації або розробляти оптимальні за часом виконання алгоритми обробки, які використовують апаратні можливості сучасних процесорів і мов програмування. Тому метою дослідження був аналіз швидкодії існуючих алгоритмів та досягнення зменшення часу обробки.

Найбільш поширеними на практиці є фільтри з кінцевою імпульсною характеристикою (FIR) [3, 4]. FIR фільтри мають скінчену відповідь на імпульс, що означає, що їх відповідь на вхідний сигнал залежить лише від обмеженої кількості минулих вхідних сигналів. Це дає їм хорошу лінійність та стійкість до коливань. Загальна структура таких фільтрів наведена на рис. 1, а математичний опис сигналу на виході фільтра представлено формулою 1.

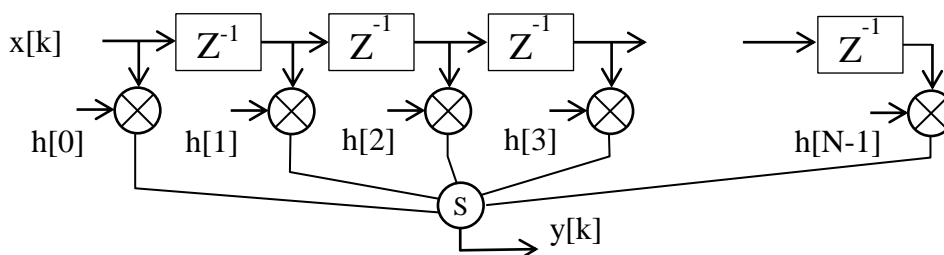


Рисунок 1 – Узагальнена схема цифрового FIR фільтра

$$y[k] = \sum_{i=0}^{N-1} h(i) \cdot x[k - i], \quad (1)$$

де N – порядок фільтру; $h(i)$ – коефіцієнти фільтру; $x[k-i]$ – відліки вхідного сигналу.

Задача створення алгоритмів цифрової фільтрації сигналів на мікроконтролерах з ARM архітектурою не є новою. Існують спеціалізовані бібліотеки CMSIS, які містять набір функцій реалізації цифрової фільтрації. Наприклад, для реалізації FIR фільтрів існують функції: `arm_fir_f32`, `arm_fir_q31`, `arm_fir_q15`. Перша з функцій призначена для виконання обчислень з плаваючою комою. При таких обчисленнях поведінка фільтра завжди стабільна і результат реальної фільтрації відповідає теоретичним розрахункам. Дві інші функції призначені для реалізації обчислень з фіксованою комою. Вони мають більшу швидкодію, але результат фільтрації може значно відрізнятись від очікуваного. Кожна з функцій має 3 вхідних параметри: набір коефіцієнтів фільтру, масив відліків вхідного сигналу і масив для формування результатів фільтрації.

Функції забезпечують високу швидкодію завдяки використанню усіх 32 регістрів загального призначення ARM ядра. У багатьох Internet джерелах є порівняльний аналіз швидкодії цих функцій при заданій частоті дискретизації та кількості коефіцієнтів фільтру. Але реальна реалізація фільтрації сигналу значно складніша, бо передбачає взаємодію з аналого-цифровим (АЦП) та цифро-аналоговим (ЦАП) перетворювачем.

У таких сучасних мікроконтролерах, як STM32 існує вбудований контролер прямого доступу до пам'яті, який забезпечує паралельно з роботою ядра процесору передавання даних між оперативною пам'яттю та периферійним пристроєм. Найбільша швидкодія передавання даних досягається при використанні кільцевих буферів разом з контролером DMA та АЦП і ЦАП. З урахуванням апаратних можливостей мікроконтролерів STM32 були запропоновані наступні: структурна реалізація цифрової фільтрації та алгоритм програмної реалізації фільтрації.

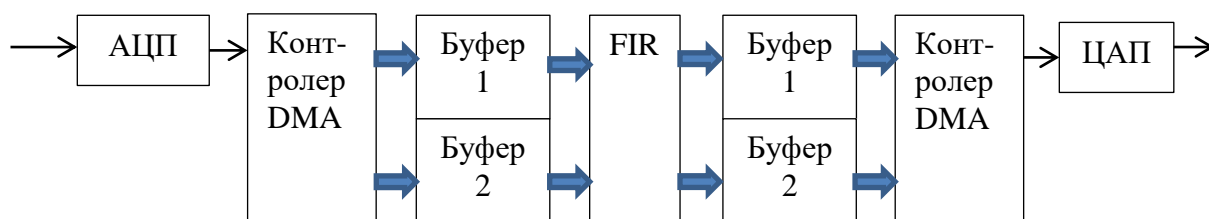


Рисунок 2 – Апаратна реалізація цифрової фільтрації у мікроконтролерах STM32

В системі на рис. 2 АЦП виконує періодичні перетворення вхідного сигналу у цифрову форму, а контролер DMA передає ці значення у 2 буфера. Спочатку накопичуються дані у перший буфер. Коли починається накопичення у другий виконується цифрова фільтрація, що використовує кінцеву частину другого буфера і значення із першого буфера. Результат фільтрації одразу записується у перший вихідний буфер для їх відтворення за допомогою контролеру DMA та ЦАП. АЦП та ЦАП працюють синхронно, а синхронізацію забезпечує на апаратному рівні таймер, що подає на них команди запуску перетворень. Після заповнення буферу 2 йде фільтрація частини даних із буферу 1 і даних із буферу 2, а у цей час відтворюються дані із першого буфера. Далі процес повторюється циклічно по колу. Для досліджень були створені дві програми. Перший алгоритм фільтрації був реалізований у обробниках переривань від контролеру DMA по закінченню заповнення кожного з буферів. Він використовував безпосередньо результати аналого-цифрового перетворення із буферів. Друга програма використовувала стандартні функції бібліотеки CMSIS-DSP для фільтрації. Але для їх використання необхідно було додатково формувати масив вхідних даних бо функції не

вміють працювати із кільцевими буферами, тобто було необхідно копіювати значення із буферів у додатковий масив. Для аналізу швидкодії розробленого алгоритму було синтезовано цифровий фільтр у середовищі Matlab із наступними характеристиками: полоса пропускання 7.5кГц, частота дискретизації 50кГц, гранична частота полоси затримування 9кГц, порядок фільтру 63 (кількість коефіцієнтів 64). Результати дослідження представлені на рис. 3.

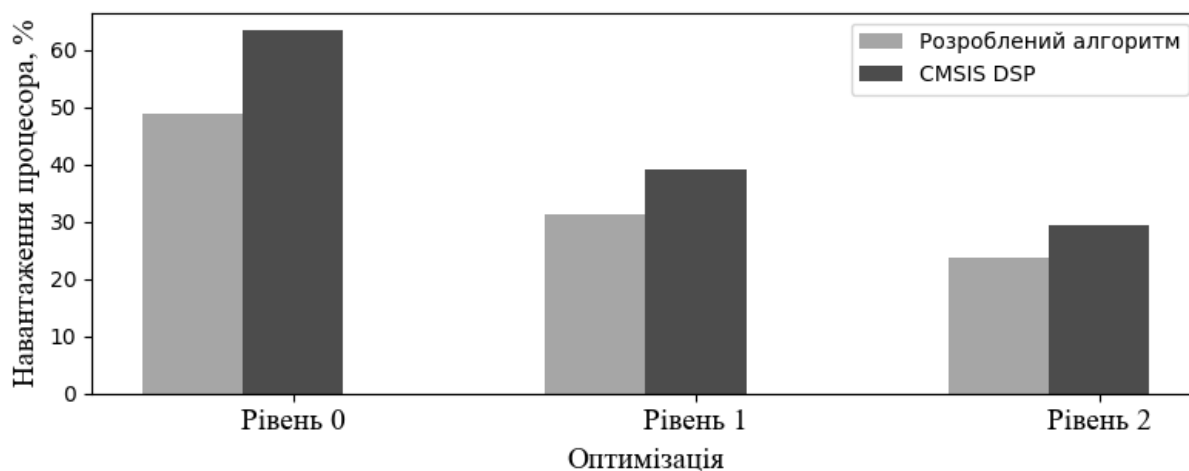


Рисунок 3 – Результати аналізу швидкодії алгоритмів реалізації DSP фільтрів

Аналіз отриманих даних показує, що при відсутності оптимізації запропонований алгоритм дає вигравш у 14.5% від загальної швидкодії мікроконтролера. При використанні першого рівня оптимізації програмного коду вигравш зменшується до 5.6%, але все одно є достатньо значним.

Висновки. Запропонований алгоритм цифрової фільтрації сигналу з безпосереднім використанням кільцевих буферів АЦП та ЦАП забезпечує швидкодію більшу ніж у функцій бібліотеки CMSIS DSP, бо ця бібліотека не враховує апаратних можливостей мікроконтролерів серії STM32F4 та вищих. Завдяки запропонованому алгоритму вдалося досягти зростання швидкодії до 14.5%.

Список використаних джерел.

1. Simona-Daiana Sim; Zsófia Lendek; Petru Dobra Implementation and Testing of Digital Filters on STM32 Nucleo-64P 2022 IEEE International Conference on Automation, Quality and Testing, Robotics pp.1-6.

2. Програмування мікроконтролерів STM32 в середовищі STM32CubeIDE в прикладах і задачах: Навч. посіб. / О. В. Зубков, І. В. Свид, О. В. Воргуль, В. В. Семенець. Дніпро : ЛІРА ЛТД, 2022. 144 с.

3. Проектування цифрових фільтрів: навчальний посібник / Г.Є. Філатова. Х. : НТУ «ХП», 2017. 120 с.

4. Lars Wanhammar, Tapio Saramäki Digital Filters Using MATLAB 1st ed.- Springer, 2020, 821 p.