

2003. – P. 280–283. 4. Haithcoat T., Song W., Hipple J. Automated Building Extraction and Reconstruction from LIDAR Data // R&D Program for NASA/ICREST Studies Project Report. 09/16/01, 2001. – P. 27. 5. Rottensteiner F., Briese Ch. A new method for building extraction in urban areas from high-resolution LIDAR data // Institute of Photogrammetry and Remote Sensing, Vienna University of Technology. – 2003. – 7 p. 6. Morgan M., Habib A. Interpolation of LIDAR data and automatic building extraction // ACSM-ASPRS 2002 Annual conference proceedings. – 2002. – 10 p. 7. [www.istar.com](http://www.istar.com) 8. Ємець В., Карпін О., Імад І.А. Сбієх. Система реконструкції тривимірних об'єктів за невпорядкованими даними аерозображень // Моделювання та інформаційні технології: Зб. наук. пр. – К., 2005. – Вип. 35. – С. 120–127. 9. Форсайт, Девід А., Понс, Жан. Комп'ютерне зречення. Современный подход: Пер. с англ. – М.: Изд. дом “Вильямс”, 2004. – 928 с.; Парал. тит. англ. 10. Ємець В.Ф., Мороз І.В. Алгоритм відокремлення країв на растрових зображеннях тривимірних об'єктів // Вісн. Нац. ун-ту “Львівська політехніка”. – 2003. – № 492. – С. 75–80. 11. Скворцов А. Триангуляція Делоне и её применение. – Томск: Изд-во Том. ун-та, 2002. – 128 с. 12. Skala V. Precision of iso-surface extraction from volume data and visualization // Proceedings of ALGORITHM 2000: Conference on Scientific Computing 2000. – 2000. – P. 368–378. 13. Эйнджер Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. – 2-е изд.: Пер. з англ. – М.: Изд. дом “Вильямс”, 2001. – 592 с. 14. The MT (Multi-Tesselation) Package by Leila De Florian et al. // C++ library, University of Genova, <http://www.disi.unige.it/person/MagilloP/MT/>. – 2003.

УДК 681.3

Є.О. Кириченко, Н.Я. Павич\*, Я.С. Парамуд, Ю.М. Спіченко\*\*,  
Національний університет “Львівська політехніка”  
кафедра електронних обчислювальних машин,  
\*кафедра програмного забезпечення автоматизованих систем,  
\*\*проектно-конструкторське об'єднання “Політехніка”

## ФАЙЛОВА СИСТЕМА СПЕЦІАЛІЗОВАНОГО ФЛЕШ-НАКОПИЧУВАЧА

© Кириченко Є.О., Павич Н.Я., Парамуд Я.С., Спіченко Ю., 2006

**Розглянуто побудову файлової системи спеціалізованого флеш-накопичувача, який дає змогу приймати дані від різних джерел інформації, упаковувати їх у типові файли в реальному часі, обмінюватися з комп'ютером через інтерфейс USB.**

**The file system construction of specialized Flash-based Mass Storage Device is examined. The device allows taking information from various data sources, packing it into typical files in real-time mode, and exchanging them with computer through the standart USB interface.**

### Вступ

Одним із перспективних напрямів розвитку зовнішніх запам'ятовувальних пристроїв є широке впровадження флеш-накопичувачів [1]. Сучасний розвиток елементної бази для флеш-пам'яті [2–4] створює умови для побудови спеціалізованих флеш-накопичувачів. Актуальною прикладною задачею є приймання комп'ютером інформації від різних джерел для подальшого її опрацювання. У багатьох випадках джерела інформації використовують інтерфейси, через які безпосередньо, без спеціальних апаратно-програмних засобів, неможливо передати дані до універсального або персонального комп'ютера. Тоді доцільно використовувати спеціалізований флеш-накопичувач із вбудованою файловою системою, який забезпечував би узгодження різних

інтерфейсів та формував файли даних, прийнятні для стандартних операційних систем. Тому дослідження щодо побудови таких накопичувачів є актуальними.

### Огляд літературних джерел

Флеш-накопичувач – це один із найперспективніших периферійних пристроїв персональних комп'ютерів (ПК) [1]. Провідні фірми виготовляють велику кількість флеш-накопичувачів з різними технічними характеристиками. Переважна більшість таких накопичувачів забезпечує можливість взаємодії з ПК через інтерфейс USB (Universal Serial Bus) [5–9]. Однак в окремих випадках потрібно запам'ятати інформацію в реальному часі від джерела, яке безпосередньо не має зв'язку з комп'ютером. Після запам'ятовування інформацію переважно опрацьовують на ПК. Тоді доцільно використовувати спеціалізований портативний флеш-накопичувач. Сучасна елементна база [2–4] дає змогу побудувати такий накопичувач. Для забезпечення ефективного запам'ятовування та подальшого опрацювання інформацію доцільно упакувати у типові файли. Важливим компонентом спеціалізованого флеш-накопичувача стає файлова система (*filesystem*) [10, 11]. Загалом поняття “файлова система” містить:

- ◆ сукупність усіх файлів на запам'ятовувальному пристрої;
- ◆ набори службових структур даних, що використовуються для керування файлами, таких як, наприклад, каталоги файлів, дескриптори файлів, таблиці розподілу вільного і зайнятого запам'ятовувального простору;
- ◆ комплекс програмних засобів, що реалізують керування файлами, зокрема операції для створення, іменування, знищення, запису, читання файлів, установки атрибутів та рівнів доступу, пошуку та ін.

Файлова система накопичувача має бути сумісною з типовими операційними системами ПК, які найчастіше будуються на основі файлових таблиць FAT (File Allocation Table) [10]. Використання FAT забезпечує простоту файлової системи, надійність зберігання інформації на зовнішніх запам'ятовувальних пристроях.

### Постановка завдання

Розробити принципи побудови спеціалізованого флеш-накопичувача та його файлової системи, які б дали змогу приймати інформацію від різних джерел формувати в реальному часі типові файли та надалі передавати файли в комп'ютер через стандартний інтерфейс.

### Основні результати дослідження

На основі аналізу функціонального призначення спеціалізований флеш-накопичувач має приймати в реальному часі вхідні дані через інтерфейс джерела інформації, формувати файлову систему та забезпечити подальше передавання їх через шину USB на ПК під керуванням типової операційної системи (Win2k, XP, Linux). Пристрій автономно (без участі користувальницьких апаратних і програмних рішень) має забезпечити швидкісний запис даних на портативний флеш-накопичувач і надалі передавання їх на персональний комп'ютер. На пристрій доцільно покласти додаткову функцію цифрового опрацювання сигналів (ЦОС) під час введення інформації, оскільки вона є досить поширеною в сучасних прикладних задачах.

Запропоновану структурну схему пристрою наведено на рис. 1.

Пристрій забезпечує приймання даних в реальному часі через контролери зовнішніх інтерфейсів і під керуванням мікропроцесора записує дані до буферної пам'яті. Дані за відповідним алгоритмом формуються у файлову структуру на внутрішньому Flash-накопичувачі. Контролер USB через універсальну послідовну шину забезпечує передавання даних на персональний комп'ютер. Пристрій повинен функціонувати в реальному часі за найбільшого темпу надходження вхідних даних. Тому як мікропроцесор доцільно використати DSP-процесор, орієнтований на первинну цифрову обробку сигналів і забезпечує високу продуктивність опрацювання вхідних даних.

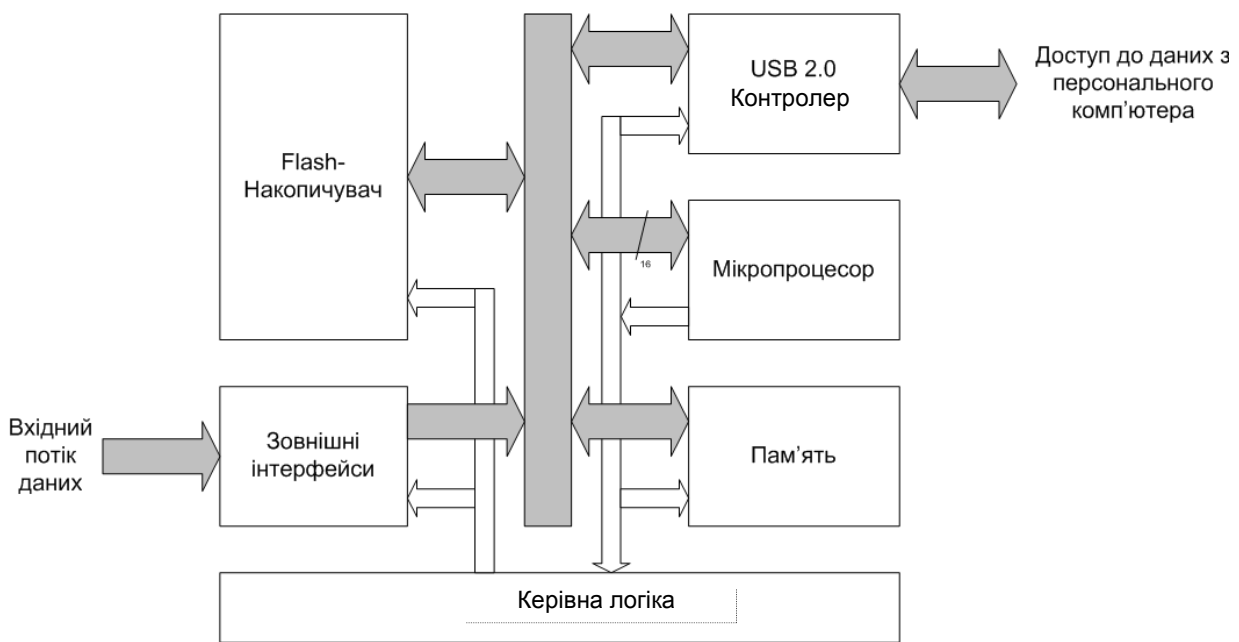


Рис. 1. Структурна схема пристрою

Алгоритм функціонування пристрою наведено на рис. 2. Із вмиканням живлення відбувається самотестування та ініціалізація всіх його складових: процесора, USB-контролера, Flash-накопичувача та контролерів зовнішніх інтерфейсів. Завантажується інформація про файлову структуру, яка на цей час є у Flash-накопичувачі. Після цього мікропроцесор переходить в режим очікування переривання від USB-контролера або контролерів зовнішніх інтерфейсів. У разі виникнення переривання від зовнішнього інтерфейсу програма починає збирати дані з буферів визначеного контролера зовнішнього інтерфейсу й обробляти їх. Оброблені дані DSP складає до FIFO-буфера у своїй пам'яті даних. Коли FIFO-буфер заповниться, управління передається підпрограмі формування файлової структури, яка за короткий час надає прийнятим даним вигляду структури файла, який згодом перескерують до буфера Flash-накопичувача. Після цього Flash-накопичувачу подається команда для запису даних з буфера, і процес починається спочатку. Алгоритм передбачає фоновий контроль за правильністю функціонування пристрою, і у разі виявлення помилки алгоритм повертається на відповідний блок. У випадку триразового повторення однієї помилки пристрій інформує користувача про несправність.

У цій блок-схемі алгоритму важливими є елементи щодо формування структури файлів та реалізація протоколу USB Mass Storage. Їх доцільно розглянути детальніше.

Блок-схему алгоритму формування структури файлів зображено на рис. 3. Після заповнення FIFO-буфера аналізують дані завантажувального сектора, обробленого під час ініціалізації. Саме в ньому прописано всю інформацію про склад та будову файлової системи, яка в цей час є на Flash-накопичувачі. Визначивши номер сектора, з якого починаються дані, та їхню кількість, програма переходить до аналізу шляху та структур директорій, через які цей шлях проходить. Залежно від запиту в цих структурах відбувається пошук або заданого переліку символів (назви файла), або вільного місця для створення нового файла. Якщо операція пройшла успішно, то ініціюється відповідно читання з носія або запис на нього. Алгоритм містить підпрограми формування структури файлів. Частина операцій для отримання інформації про місткість носія та про будову файлової системи на ньому проводять під час ініціалізації самого пристрою. Надалі підпрограма визначає шлях, за яким знаходиться потрібний нам файл. Шлях визначають за допомогою файлової таблиці (FAT), в якій зазначено всю структуру файлів та директорій, які містяться на диску (носії). Перебираючи сектор за сектором, підпрограма або знаходить потрібне місце і, залежно від команди, здійснює запис чи читання, або повертає на відповідний блок після виявлення помилки виконання операції. Виконання цього алгоритму підтримується самоконтролем. Про наявність

некоректної роботи можна взяти з глобального ідентифікатора, якому присвоюється номер помилки, виявленої засобами контролю.

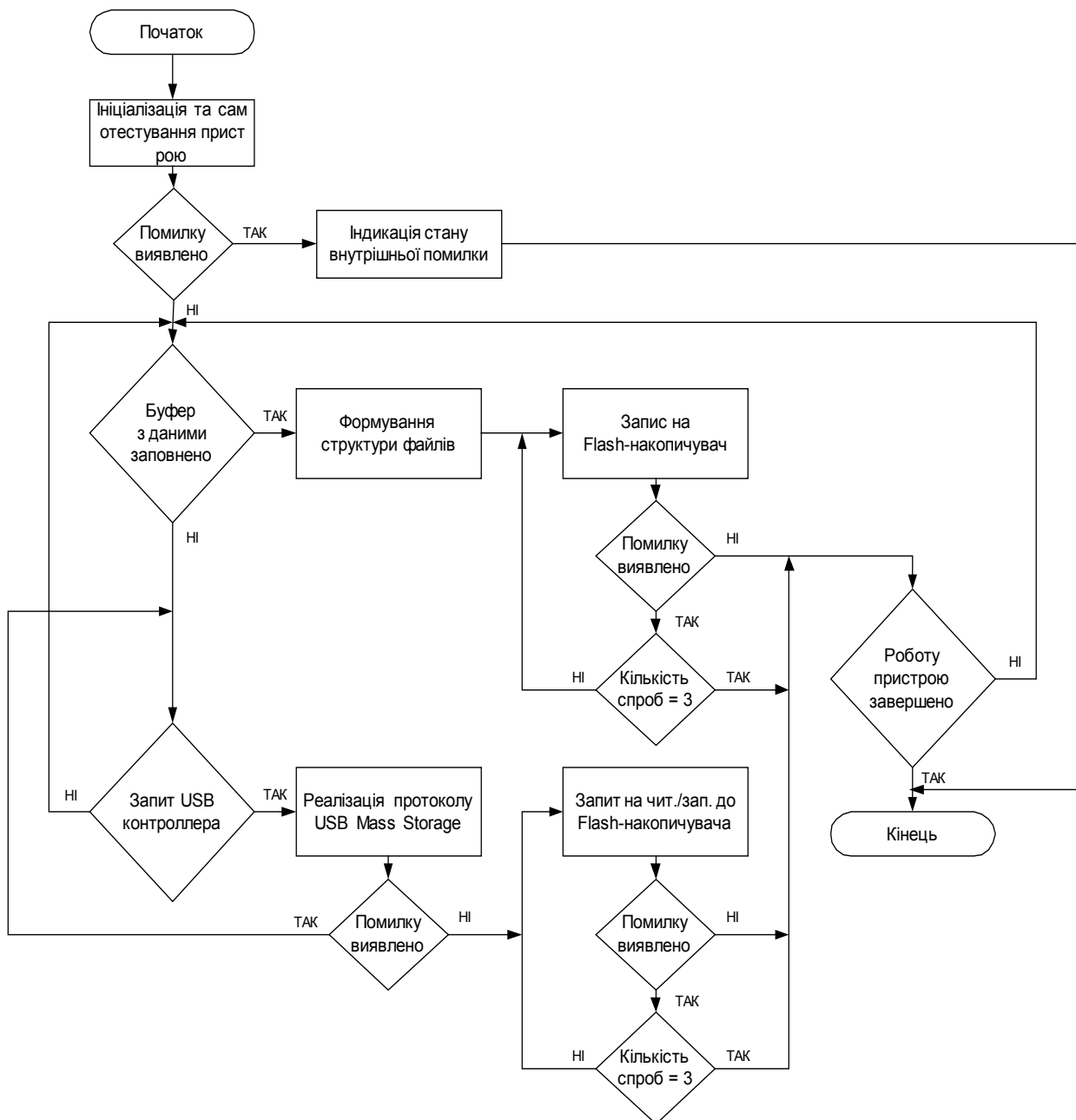


Рис. 2. Блок-схема алгоритму функціонування пристрою

У випадку виникнення переривання від USB-контроллера програма переходить в режим обробки протоколу USB Mass Storage. Цей процес відбувається за алгоритмом, блок-схему якого наведено на рис. 4. Спочатку аналізують CBW-пакет (Command Block Wrapper), який відсилає хост.

У пакеті знаходиться інформація, яка вказує на напрямок передавання, на кількість потрібних даних та унікальну сигнатуру пакета, завдяки якій відбувається перевірка на збіжність процесу передавання даних. Також там знаходиться SCSI-команда, в якій сказано, що саме треба робити з Flash-накопичувачем (читання, запис, визначення місткості тощо). Після виконання SCSI-команди програма відсилає хосту відповідь у вигляді CSW-пакета (Command Status Wrapper), в якому повідомляє про виникнення помилки або про успішне завершення операції. Надалі програма повертається у вихідне положення.

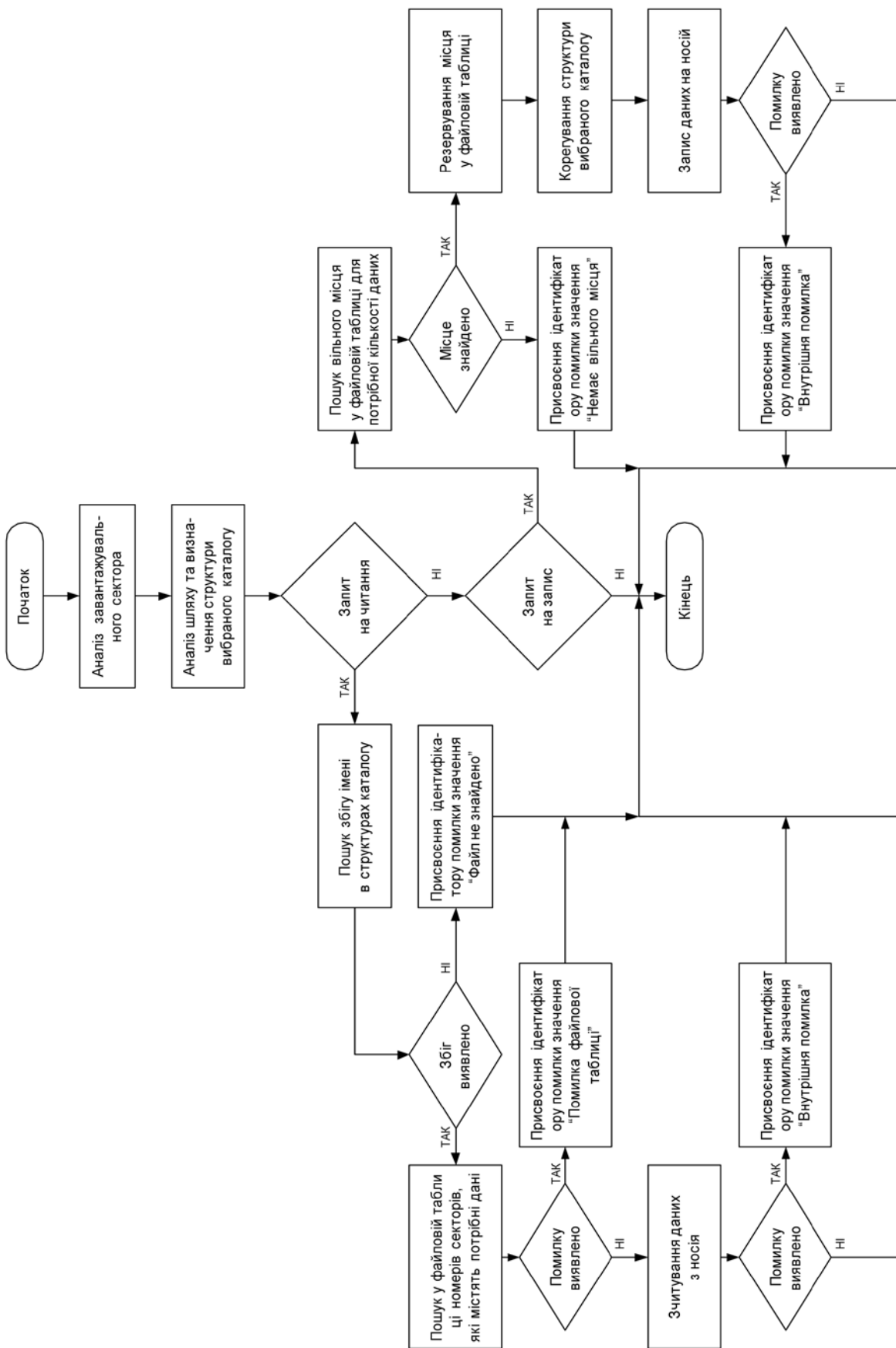


Рис. 3. Блок-схема алгоритму формування структури файлів

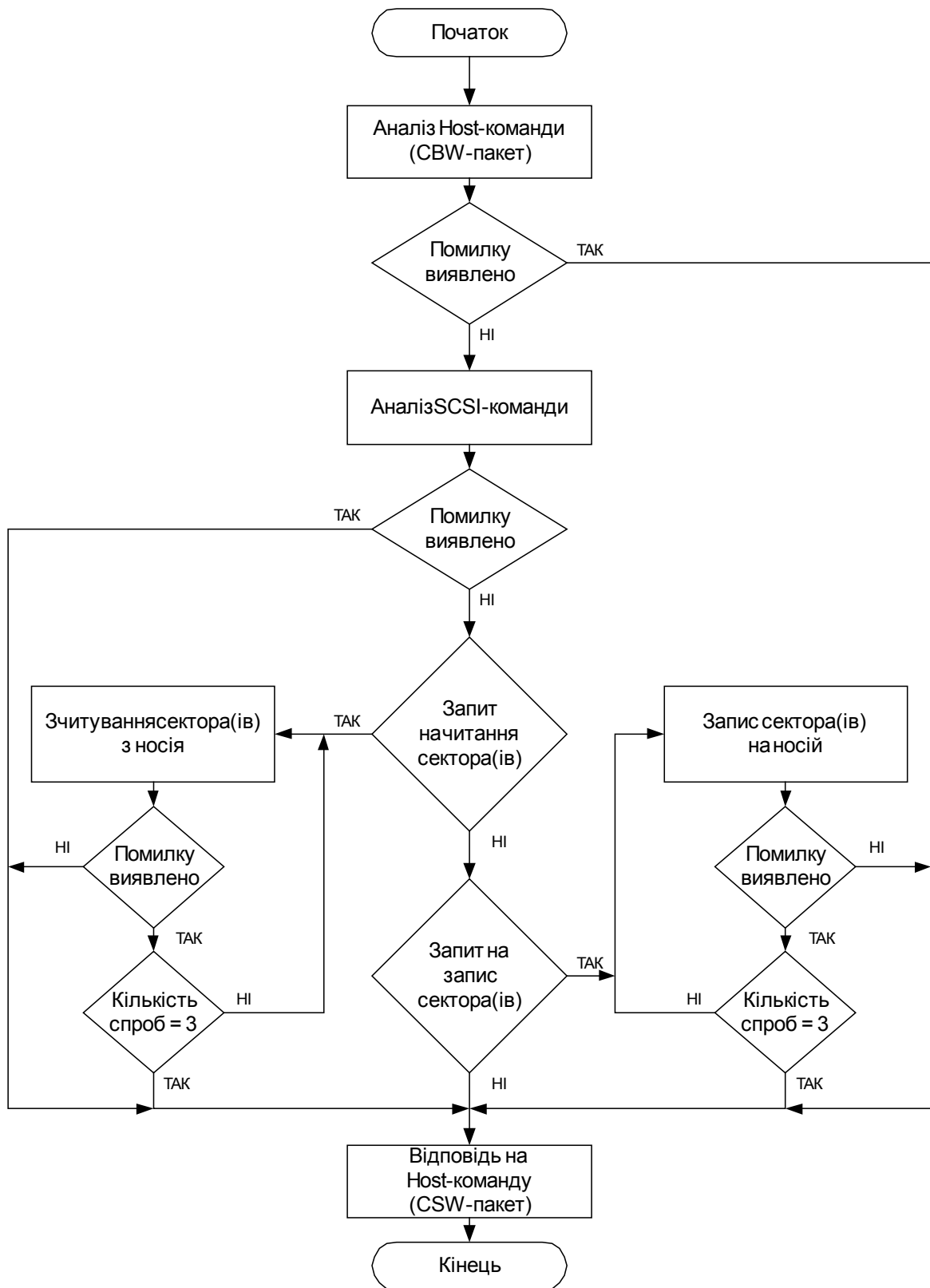


Рис. 4. Блок-схема алгоритму реалізації протоколу USB Mass Storage

Наведені алгоритми перевірено на ефективність відповідним програмним забезпеченням. Програмне забезпечення розроблено за сучасними технологіями [12, 13]. Модель програмного забезпечення розроблена із використанням середовища VisualDSP++ фірми Analog Device. Процес налагодження після написання коду програми й успішної компіляції містив три основні етапи:

моделювання (Simulation), оцінку (Evaluation) і емуляцію (Emulation). На етапі моделювання працює моделююча програма (симулятор), що імітує роботу процесора. Симулятор використовують для перевірки і налагодження програмного коду до виготовлення плати з процесором. На другому етапі використовували відлагоджувальну плату EZ-KIT для того, щоб визначити, який процесор найкраще підходить для розв'язання задачі. Плату під'єднують до комп'ютера за допомогою кабелю через USB-порт. Сьогодні відлагоджувальні плати існують для всіх типів процесорів, починаючи з ADSP-2181 і закінчуючи новітнім процесором BlackFin. На третьому етапі, коли макет пристрою було виготовлено, плати тестували за допомогою спеціального апаратно-програмного модуля – емулятора. Цей модуль керує цифровим сигнальним процесором через JTAG-інтерфейс і дає змогу докладно відстежити виконання програмного коду. Експериментальними дослідженнями підтверджено правильність запропонованих рішень.

### Висновки

Розглянута структурна схема, запропоновані алгоритми функціонування дають змогу будувати спеціалізовані флеш-накопичувачі, призначені для приймання даних від різних джерел інформації, упаковувати їх у типові файли в реальному часі, обмінюватися інформацією з комп'ютером через інтерфейс USB. Сучасні елементна база, засоби проектування та відлагодження забезпечують широкий спектр характеристик пристроїв залежно від вибору конкретних мікросхем накопичувачів та мікропроцесорів.

1. Гук М. *Аппаратные средства IBM PC // Энциклопедия. – 2-е изд. – СПб.: Питер, 2002. – 928 с.* 2. *NAND Flash Spare Assignment Recommendation. 2003. Memory Division Samsung Electronics Co., Ltd.* 3. *NAND Flash Spare Assignment Recommendation. – 2003. Memory Division Samsung Electronics Co., Ltd.* 4. *SanDisk SDP3B FlashDisk Product Manual, Rev.7.1 © 2002 Sandisk Corp.* 5. *ISP1581 Hi-Speed Universal Serial Bus interface device. Product data. Rev. 05.* 6. *Universal Serial Bus Specification. Revision 2.0. Universal Serial Bus Mass Storage Class. Bulk-Only Transport. Revision 1.0.* 7. *Universal Serial Bus Mass Storage Class. Specification Overview. Revision 1.2.* 8. *SCSI Block Commands – 2 (SBC-2). Revision 16.* 9. *SCSI Primary Commands – 3 (SPC-3). Revision 23.* 10. *FAT: General Overview of On – Disk Format. Version 1.02. Microsoft Corp.* 11. *Файловая\_система. – <http://ru.wikipedia.org/wiki/>.* 12. *ADSP-21065L SHARC User's Manual. © Analog Devices, Inc., 2000.* 13. *Ligen/ИИ-TECH (<http://WASM.RU>). Программирование ATA устройств // Журн. "ARGC & ARGV". – 2003. – № 5. – С. 31–36.*