

УДК 004.451

І.М.Майків

Тернопільський національний економічний університет

## ПРОГРАМНО-АПАРАТНИЙ МЕТОД РЕАЛІЗАЦІЇ КОНТОЛЕРІВ ПОСЛІДОВНИХ ІНТЕРФЕЙСІВ

Запропоновано програмно-апаратний метод побудови контролерів послідовних інтерфейсів, який передбачає роздільне виконання функцій каналного і фізичного рівнів. При цьому функції каналного рівня реалізуються програмно з допомогою мікроконтролера (МК), а функції фізичного рівня – апаратно, на програмованій логічній матриці (ПЛМ). Запропонований метод забезпечує оптимальне співвідношення апаратних і програмних затрат на реалізацію контролера інтерфейсу (КІ) та ефективне використання обчислювальної потужності МК.

Ключові слова: контролер інтерфейсу, каналний рівень, фізичний рівень, мікроконтролер, програмована логічна матриця.

Вступ. Сучасні вимірювально-керуючі системи (ВКС) реалізуються як багаторівневі локальні мережі із розподіленими обчислювальними ресурсами на базі поширених промислових шин, у яких виділяють декілька ієрархічних рівнів [1-3]. На даний час існує більше десятка послідовних інтерфейсів (RS232, RS485, LIN, CAN, 1-Wire, Ethernet та ін.), що широко використовуються для реалізації розподілених ВКС. Це обумовлено великим різноманіттям виробничих задач, які характеризуються різними умовами експлуатації, об'ємами потоків даних, допустимими затратами на реалізацію. Більшість вказаних інтерфейсів запропоновані фірмами виробниками промислового обладнання для вирішення множини задач у певній проблемній області, а отже, оптимізовані для відповідних умов експлуатації. При цьому жоден із них не домінує в даній області.

Тому при розробці універсального обладнання для ВКС фірмам-виробникам необхідно забезпечити підтримку широкого набору послідовних інтерфейсів, що веде до зростання кінцевої вартості виробів.

Метою роботи дослідження нового методу реалізації послідовних інтерфейсів, що дозволить зменшити вартість обладнання для ВКС.

### 1. Задачі контролерів інтерфейсів та аналіз відомих методів їх реалізації

Процес обміну даними між окремими вузлами розподіленої ВКС в цілому відповідає моделі ISO/OSI [4], однак для ефективного функціонування локальної мережі достатньо забезпечити підтримку лише трьох рівнів (фізичного, каналного та прикладного) [5]. При цьому формується єдине програмно-апаратне середовище обміну інформацією (рис.1), де у кожному окремому вузлі: 1) обробку даних (функції прикладного рівня) реалізує мікроконтролер (МК) або процесор; 2) обмін даними (функції каналного та фізичного рівнів) реалізує підсистема вводу/виводу, яка забезпечує підтримку набору необхідних інтерфейсів. Саме поняття інтерфейсу об'єднує сукупність уніфікованих програмних та апаратних засобів [6] які реалізують мережевий адаптер, що включає: 1) інтегральну мікросхему (ІМС) контролера інтерфейсу (КІ); 2) драйвер фізичної лінії; 3) роз'єм. При цьому на інтерфейс покладається забезпечення інформаційної, електричної та конструктивної сумісності між елементами розподіленої системи [6].

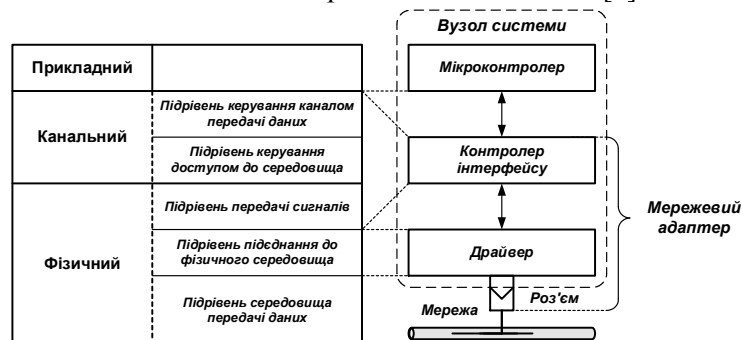


Рис. 1. Розподіл рівнів моделі ISO/OSI між компонентами вузла

Традиційно КІ реалізують програмним або апаратним методом. Програмний метод реалізації передбачає послідовне виконання набору підпрограм (ПП), які реалізують протокол обміну даними заданого інтерфейсу і забезпечують прийом/передачу кожного окремого біта та повідомлення в цілому. Такий підхід забезпечує високу гнучкість системи, оскільки у випадку необхідності змінити інтерфейс потрібно перейти до виконання іншого набору ПП, розміщених в постійному запам'ятовуючому пристрої (ПЗП) МК, і за потреби замінити драйвер фізичної лінії зв'язку. Таке рішення є найбільш ефективним за необхідності мінімізувати апаратні затрати, однак не забезпечує високу достовірність прийому даних, а також ефективне використання обчислювальної потужності МК. Її підвищення, шляхом ускладнення алгоритму обробки даних, не завжди можливе, оскільки набір відповідних ПП необхідно виконувати в реальному масштабі часу, що не завжди можливе через обмежену продуктивність не дорогих МК. Прикладом такого підходу до реалізації набору поширених послідовних інтерфейсів є мережевий прикладний процесор (МПП) [7], виконаний як двопроцесорна система, в якій процеси обробки та обміну даними виконуються на окремих МК AT89C52.

Апаратний метод реалізації передбачає використання спеціалізованих ІМС КІ, кожна з яких реалізує один інтерфейс. Такий підхід дозволяє досягнути: 1) високої швидкості обміну даними; 2) високої достовірності прийому даних; 3) мінімального завантаження МК процесами, пов'язаними із обробкою повідомлень. Прикладом використання такого підходу є МПП фірми Esensors [8], що підтримує шість послідовних інтерфейсів (RS232, RS485, ТП, 1-Wire, SPI, I2C, IEEE-1451.4). Однак при цьому зростає апаратна складність просторою, що веде до зменшення його надійності, зростання енергоспоживання, габаритів та кінцевої вартості.

Усунути частину із вище вказаних недоліків апаратного методу реалізації КІ можна, використавши програмовані логічні матриці (ПЛМ), які користувач може самостійно конфігурувати відповідно до своїх потреб. Прикладом такого підходу до реалізації КІ є програмовані модулі вводу-виводу UNIO [9] фірми Fastwel, виконані на ПЛМ фірми Xilinx, що можуть містити від однієї до чотирьох ПЛМ, кожна з яких реалізує окремий КІ. Очевидними переваги вказаного підходу є: 1) відсутність обмежень, пов'язаних із функціональними можливостями наявної елементної бази, що дозволяє реалізувати практично будь-який інтерфейс; 2) висока гнучкість, що дозволяє багаторазово вносити зміни у схему без зміни конструкції виробів; 3) спрощений процес відлагодження, що дозволяє скоротити тривалість процесу розробки та впровадження у виробництво; 4) зменшення габаритних розмірів. Означений підхід дозволяє реалізувати режим "динамічної рекофігурації" ПЛМ – можливість змінювати набір задач, які вирішуються без зупинки систем, що є особливо важливим при створенні відмовостійких систем. Наступним прикладом використання даного підходу є багатоканальний контролер UART інтерфейсу на базі ПЛМ [10].

Аналогічний підхід використовують при реалізації інтерфейсів у МК. Так у МК серії AVR фірми ATMEL [11] реалізовано вузол універсального послідовного інтерфейсу (Universal Serial Interface – USI), який, залежно від налаштувань, реалізує SPI або I<sup>2</sup>C інтерфейс. Аналогічний вузол, що підтримує один з інтерфейсів SPI, I<sup>2</sup>C, UART, реалізовано в МК серії MSP430 фірми Texas Instruments [12].

Аналіз відомих рішень показав, що сучасний підхід передбачає використання уніфікованих апаратних засобів, які шляхом реконфігурації реалізують один із набору необхідних інтерфейсів. Однак результати, наведені в [13, 14], показують, що апаратні затрати на реалізацію КІ на базі ПЛМ є приблизно співрозмірними із затратами на реалізацію soft-мікроконтролера PicoBlaze [15]. Це наочно демонструє надлишковість апаратного методу, що є особливо важливим у випадку створення вбудованих систем на базі ПЛМ. Узагальнений перелік переваг та недоліків програмно та апаратного методів реалізації КІ наведено в таблиці 1.

Результати аналізу показують, що недоліки програмного та апаратного методів створення КІ є взаємно виключними. Це вказує на те, що при проектуванні КІ не враховується характер процесів, які реалізує КІ, що веде до виникнення внутрішніх протиріч.

Таблиця 1

Переваги та недоліки програмного і апаратного методів реалізації контролерів послідовних інтерфейсів

	Програмний	Апаратний
Переваги	<ol style="list-style-type: none"> <li>1. Легко реалізувати алгоритм.</li> <li>2. Використані апаратні ресурси залишаються постійними при зростанні складності протоколу.</li> <li>3. Багатократне використання логічних ресурсів, добре підходить для реалізації логічних функцій.</li> </ol>	<ol style="list-style-type: none"> <li>1. Висока швидкість обміну даними, що практично не залежить від складності протоколу обміну.</li> <li>2. Мінімальне завантаження МК процесами обробки даних.</li> <li>3. Мінімальне використання оперативної та постійної пам'яті.</li> <li>4. Одночасне виконання процесів прийому та передачі даних (Режим обміну – duplex).</li> </ol>
Недоліки	<ol style="list-style-type: none"> <li>1. Низька швидкість обміну даними, що зменшується при ускладненні протоколу обміну.</li> <li>2. Об'єм використаної оперативної пам'яті та постійної пам'яті зростає при ускладненні протоколу обміну.</li> <li>3. Почергове виконання процесів прийому та передачі даних (Режим обміну – simplex).</li> </ol>	<ol style="list-style-type: none"> <li>1. Паралельне виконання процесів утруднює реалізацію та відлагодження.</li> <li>2. Великі апаратні затрати, які зростають при ускладненні протоколу обміну.</li> </ol>

2. Аналіз процесів обробки даних в контролерах послідовних інтерфейсів при прийомі та передачі

Незалежно від типу інтерфейсу, КІ, як елемент системи вводу/виводу, забезпечує формування, передачу, а також прийом і обробку повідомлень. Вказані функції відносяться до задач каналного рівня та підрівня передачі сигналів, фізичного рівня (див. рис.1). Процеси формування і обробки повідомлень реалізують функції каналного рівня, а саме: виявлення і корекції помилок, адресування, арбітражу, керування потоком даних та ін. Реалізуючи вищевказані функції каналного рівня, КІ забезпечує логічну організації процесу обміну даними і гарантує прийом даних на рівні повідомлення із заданою достовірністю. Одночасно КІ реалізує набір функцій підрівня передачі сигналів: кодування і декодування біта, тактової синхронізації біта, синхронізації потоку даних, що забезпечує прийом/передачу окремих бітів як складової частини повідомлення. Повний перелік функцій, що реалізує КІ в процесі прийому/передачі повідомлення, визначається стандартом на інтерфейс.

Окрім вищевказаних функцій, які реалізують обмін даними через мережу, КІ, незалежно від способу реалізації (програмного чи апаратного), взаємодіє із МК (див. рис.1), що реалізує прикладні функції.

Отже, множина функцій, які реалізує КІ, розділяється на дві групи: функції внутрішньосистемного обміну та функції мережевого обміну даними, які, в свою чергу розділяють на групу функцій каналного рівня, на якому виконується обробка повідомлення в цілому, та групу функцій фізичного рівня, які реалізують обробку біта як складової частини повідомлення (рис.2).

Такий підхід дозволяє розглядати процеси прийому та передачі повідомлень як множину окремих процесів, що взаємодіють між собою [16].

Приймач, виконуючи прийом повідомлення, реалізує процеси прийому біта та обробки повідомлення (рис.3). Процес прийому біта здійснює перетворення фізичного сигналу, що поступає з мережі і характеризується логічним значенням та тривалістю, в біт, що характеризується логічним значенням. При цьому вказаним процесом реалізуються функції: 1) виявлення ознаки початку повідомлення; 2) квантування часового інтервалу  $t_{bit}$ ; 3) синхронізація до вхідного потоку даних; 4) фіксація логічного значення шини та визначення логічного значення прийнятого біта.



Рис. 2. Множина функції КІ

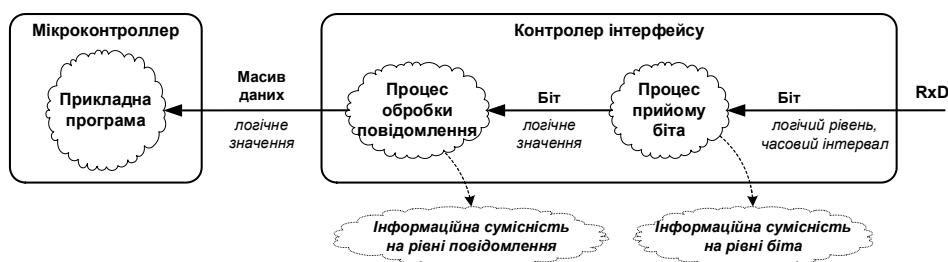


Рис. 3. Сукупність процесів при прийомі повідомлення

Одночасно приймач виконує процес обробки повідомлення, що реалізує наступні функції: 1) прийняття рішення про початок нового повідомлення; 2) формування повідомлення із прийнятих бітів; 3) перевірка адреси (фільтрація вхідних повідомлень); 4) виявлення та, по можливості, корекція помилок в повідомленні; 5) виділення масиву даних із прийнятого повідомлення, що поступають на обробку в МК (рис. 3).

Аналогічним чином передавач, виконуючи передачу повідомлення, реалізує процеси формування повідомлення та передачі біта (рис. 4). Процес формування повідомлення здійснює перетворення масиву вихідних даних, що поступають з МК, в послідовний потік бітів, які характеризуються логічним значенням. При цьому означеним процесом реалізуються функції: 1) прийому масиву даних від МК; 2) формування повідомлення; 3) побітової передачі повідомлення.

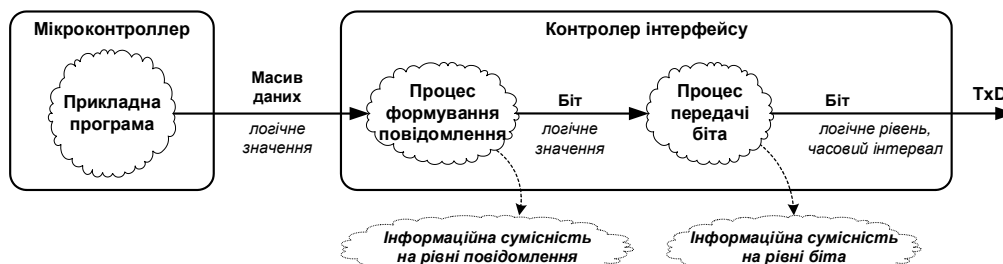


Рис. 4. Сукупність процесів при передачі повідомлення

Одночасно передавач виконує процес передачі біта, який полягає в утриманні на виході передавача логічного рівня, що відповідає його логічному значенню біта, протягом проміжку часу передачі біта ( $t_{bit}$ ).

#### 4. Програмно-апаратний метод реалізації контролерів послідовних інтерфейсів

Аналіз показав, що при обміні даними КІ виконує чотири процеси, з яких два процеси нижнього рівня виконують прийом та передачу окремого біта як складової частини повідомлення, а два інші процеси верхнього рівня реалізують обробку повідомлення в цілому. При цьому

характер процесів принципово відрізняється. Процеси формування і обробки повідомлення реалізують множину логічних функцій, які забезпечують формування (обробку) додаткових полів повідомлення. В той же час процеси прийому і передачі біта пов'язані з обробкою інформації в реальному масштабі часу. Тривалість цих процесів є детермінованою в часі і визначається типом інтерфейсу та заданою швидкістю передачі даних. При цьому процеси обробки повідомлень реалізуються цифровими автоматами (ЦА), інтервал роботи яких відповідає тривалості прийому/передачі всього повідомлення, а кожний наступний стан визначається після завершення прийому/передачі чергового біта. В той же час процеси прийому/передачі окремого біта виконують фактично інші ЦА, інтервал роботи яких відповідає проміжку часу  $t_{bit}$ .

Кожен із процесів можна реалізувати як програмно, так і апаратно, що дає чотири можливих варіанти комбінації процесів для приймача і передавача та шістнадцять комбінацій для КІ.

У випадку програмної реалізації КІ МК реалізує функції як ЦА з обробки повідомлень (функції каналного рівня), так і ЦА з обробки бітів (функції фізичного рівня). Реалізація логічних функцій (ЛФ) каналного рівня, пов'язаних із обробкою повідомлення, легко вирішується МК. В той же час реалізація набору часових інтервалів, які формують  $t_{bit}$ , при обмеженому числі таймерів, змушує реалізовувати останні програмно, у вигляді "пустих" циклів, що в цілому веде до неефективного використання МК. Також не можливо одночасно реалізувати прийом та передачу повідомлень.

У випадку апаратної реалізації КІ процеси прийому/передачі біта реалізуються на базі лічильників –подільників та нескладного ЦА. Це дозволяє легко сформувати необхідний набір часових інтервалів, які формують біт в цілому. Однак реалізація множини ЛФ каналного рівня в цілому веде до значного зростання складності схеми керування і самого ЦА.

Виходячи із вищесказаного, пропонується новий, програмно-апаратний метод реалізації КІ із розділним виконанням процесів прийому/передачі біта та обробки/формування повідомлення. В цьому випадку обробку повідомлень доцільно виконувати програмно, а реалізацію часових залежностей – апаратно. Узагальнена структурна схема системи із програмно-апаратним КІ представлена на рис. 5.

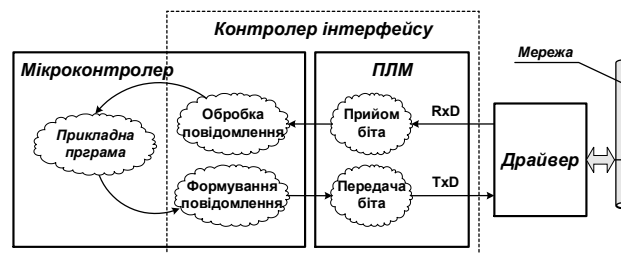


Рис. 5. Структурна схема системи з програмно-апаратним контролером інтерфейсу

Система включає МК, який програмно реалізує процеси обробки/формування повідомлення, та реалізовану на ПЛМ апаратну частину, яка реалізує процеси прийому/передачі біта. Перевагою запропонованої структури є можливість перепрограмування як МК, так і ПЛМ, що в цілому дозволяє підвищити ефективність системи за рахунок адаптації її до вимог замовника.

Оцінку ефективності запропонованого методу проведено в [13, 17, 18], в кожній із яких розглянуто декілька варіантів реалізації КІ поширених послідовних інтерфейсів на базі ПЛМ. Оцінка ефективності проводилась за наступними критеріями: 1) апаратні затрати на реалізацію; 2) програмні затрати на реалізацію - об'єм програм та кількість використаних регістрів загального призначення; 3) часові затрати – час затрачений МК на підтримку процесу обміну даними. Отримані результати показали, що програмно-апаратний метод порівняно із апаратним методом дозволяє на 30-40 % зменшити апаратні затрати на реалізацію КІ. В той же час часові затрати МК на підтримку процесу обміну даними зменшуються на декілька порядків порівняно із програмно реалізованим КІ. Так, для КІ 1-Wire [13] часові затрати зменшились у 160 раз, а для КІ I<sup>2</sup>C [18] у 22 рази.

Отримані результати показують, що даний метод є особливо ефективним для інтерфейсів із низькою швидкістю обміну даними, оскільки витрати часу, пов'язані із реалізацію процесів прийому та передачі біта, покладаються на апаратну частину КІ (див. рис.5), а отриманий надлишок часу МК може використати для розв'язання більш складних прикладних задач.

## Висновки

В роботі проведено аналіз процесів прийому та передачі даних, що реалізуються контролерами послідовних інтерфейсів. Результати аналізу показали, що процеси обробки даних в КІ слід розглядати як сукупність послідовних процесів, які відрізняються між собою за характером обробки даних. Процеси прийому/передачі біта, як складової повідомлення, пов'язані з обробкою інформації в часі. В той же час процеси формування/обробки повідомлення пов'язані із реалізацією множини функцій, які передбачають логічну обробку даних.

За результатами аналізу запропоновано новий, програмно-апаратний метод реалізації контролерів послідовних інтерфейсів, який передбачає програмну реалізацію функцій каналного рівня та апаратну реалізацію функцій підрівня передачі сигналів, фізичного рівня. Ефективність запропонованого методу підтверджена експериментальними дослідженнями, результати яких опубліковані в [13, 17, 18]. Наступні роботи пов'язані із створенням узагальненої методики проектування контролерів послідовних інтерфейсів на базі ПЛМ, яка дозволить створити множину альтернативних варіантів КІ та виконати попередню оцінку ефективності кожного із запропонованих варіантів.

1. Гусев С. Краткий курс в историю промышленных сетей / С. Гусев // Современные технологии автоматизации. – 2000. - №4. – С. 78-84.
2. Кругляк К. Промышленные сети: цели и средства / К. Кругляк // Современные технологии автоматизации. – 2002. - №4. – С. 6-17.
3. Кругляк К. Локальные сети Ethernet в АСУ ТП: быстрее, дальше, надежнее / К. Кругляк // Современные технологии автоматизации. – 2003. - №1. – С. 6-13.
4. M. Schleicher, F. Blasinger Digital Interfaces and Bus Systems for Communication. Practical Fundamentals. // M. K. Juchheim, Fulda, March 2001. – 157 P.
5. Интерфейсы систем обработки данных: Справочник / А.А. Мячев, В.Н. Степанов, В.К Щербо; Под ред. А.А. Мячева. – М.: Радио и связь, 1989. – 416 с.
6. R. Kochan, K. Lee, V. Kochan, A. Sachenko Development of a Dynamically Reprogrammable NCAP // Proceedings of the IEEE Instrumentation and Measurement Technology Conference IMTC'2004. – 2004, May 18-20. – Como, Italy. – P. 1188-1193.
7. Беломытцев В., Кашин М. Программируемые логические интегральные схемы в модулях фирмы Fastwel // Современные технологии автоматизации. – 2001. - №3. – С. 80-84.
8. Шаблин А. Многоканальный UART с эффективным использованием ресурсов кристалла ПЛИС / А. Шаблин // Современная электроника. – 2006. - №4. – С. 60-62.
9. Тамперт В. AVR-RISC микроконтроллеры / В. Тамперт: Пер. с нем. – К.: МК-Пресс, 2006. – 464 с.
10. Семейство микроконтроллеров MSP430x1xx. Руководство пользователя / Пер. с англ. – М.: ЗАО "Компел", 2004. – 368 с.
11. Майків І.М. Дослідження контролерів інтерфейсу 1-Wire, реалізованих на програмований логічній матриці / І.М. Майків // Проблеми інформатизації та управління: Зб. наук. пр. – К.: НАУ, 2008. - №1 (23). – С.188-194.
12. I2C Master/Slave Bus Controller Core [Електронний ресурс]= Cast Inc. – Режим доступу: [http://www.cast-inc.com/cores/i2c/cast\\_i2c-x.pdf](http://www.cast-inc.com/cores/i2c/cast_i2c-x.pdf).
13. PicoBlaze 8-bit Embedded Microcontroller. User Guide for Spartan-3, Virtex-II and Virtex-II Pro FPGA [Електронний ресурс] = UG129 / Xilinx. - Rev.1.1.2. - June 24, 2008. - Режим доступу: [http://www.xilinx.com/support/documentation/ip\\_documentation/ug129.pdf](http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf).
14. Стандарты по локальным вычислительным сетям: Справочник / В.К. Щербо, В.М. Киреичев, С. И. Самойленко; Под ред. С.И. Самойленко – М.: Радио и связь, 1990. – 304 с.
15. Майків І.М. Методи реалізації передавача інтерфейсу RS232 із використанням програмованих логічних матриць // Комп'ютерні науки та інженерія: матеріали 2-ї міжнародної конференції молодих науковців (CSE-2007). – Львів: Видавництво національного університету "Львівська політехніка", 2007. – С. 98-102.
16. Майків І.М. Исследование методов реализации контроллеров интерфейса I2C на программируемой логической матрице // Современные проблемы радиотехники и телекоммуникаций (РТ-2009): материалы 5-й международной молодежной научно-технической конференции / Севастопольский нац. технический ун-т. – Севастополь: Изд-во "Вебер", 2009. - С. 284.