



УКРАЇНА

(19) UA (11) 90766 (13) C2
(51) МПК (2009)
G06F 3/00
H03K 19/0175

МІНІСТЕРСТВО ОСВІТИ
І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ДЕПАРТАМЕНТ
ІНТЕЛЕКТУАЛЬНОЇ
ВЛАСНОСТІ

ОПИС ДО ПАТЕНТУ НА ВІНАХІД

(54) ПРОГРАМНО-АПАРАТНИЙ КОНТРОЛЕР ІНТЕРФЕЙСУ

1

2

(21) а200806604

(22) 15.05.2008

(24) 25.05.2010

(46) 25.05.2010, Бюл.№ 10, 2010 р.

(72) МАЙКІВ ІГОР МИРОСЛАВОВИЧ, КОЧАН РО-
МАН ВОЛОДИМИРОВИЧ, КОЧАН ВОЛОДИМИР
ВОЛОДИМИРОВИЧ

(73) МАЙКІВ ІГОР МИРОСЛАВОВИЧ, КОЧАН РО-
МАН ВОЛОДИМИРОВИЧ, КОЧАН ВОЛОДИМИР
ВОЛОДИМИРОВИЧ

(56) EP 0296739 A2; 28.12.1988

DE 19616547; 31.10.1996

US 6748456 B1; 08.06.2004

US 6828822 B1; 07.12.2004

US 7078929 B1; 18.07.2006

US 4125869; 14.11.1978

WO 9740450; 30.10.1997

CA 1128667; 27.07.1982

Беломытцев В., Кашин. М Программируемые логические современные схемы в модулях фирмы Fastwel. // Современные технологии автоматизации. - 2001. - №3

(57) Програмно-апаратний контролер інтерфейсу, що містить послідовно з'єднані мікроконтролер,

програмовану логічну матрицю і драйвер лінії зв'язку, який відрізняється тим, що в програмованій логічній матриці реалізовано схему, в яку входять три тригери, детектор початку повідомлення і послідовно з'єднані два подільники частоти генератора тактових імпульсів мікроконтролера, лічильник квантів часу і регістр зсуву, паралельний вихід якого підключено до шини даних мікроконтролера, а послідовний вхід - до виходу драйвера, до якого підключено також вхід детектора початку повідомлення, вихід якого підключено до першого входу встановлення першого тригера, вихід якого підключено до першого входу переривання мікроконтролера і входів скидання другого подільника і лічильника квантів часу, вихід якого підключено також до входу запису другого тригера і другого входу переривання мікроконтролера, причому виходи мікроконтролера підключені до блокуючого входу детектора початку повідомлення, другого входу встановлення і входу скидання першого тригера, інформаційного входу і входу запису третього тригера, вихід якого підключено до інформаційного входу другого тригера, вихід якого підключено до входу драйвера.

Запропонований пристрій відноситься до засобів обчислювальної техніки, зокрема локальних комп'ютерних мереж, автоматизованих систем управління технологічними процесами, інформаційно-вимірювальних систем, особливо систем з розподіленими обчислювальними засобами (ресурсами).

Сучасні системи з розподіленими обчислювальними ресурсами часто реалізують як багаторівневі системи [1], в яких обмін даними між окремими вузлами здійснюється цифровим способом. Зазвичай їх будують на базі послідовних інтерфейсів і процес взаємодії (обміну даними) між елементами систем в загальному відповідає еталонній моделі взаємодії відкритих систем ISO/OSI [2]. Більшість промислових систем організовані як локальні мережі, тому, для ефективної взаємодії між їх елементами, достатньо забезпечити підтримку

не всіх семи рівнів, які передбачені моделлю ISO/OSI [3, 4], а лише 3-х рівнів (прикладного, канального і фізичного).

Розподіл функцій між елементами вузла обробки даних в такій системі здійснюють наступним чином - функції прикладного рівня реалізує центральний процесор (ЦП), а функції канального і фізичного рівнів реалізує інтерфейс. Під поняттям інтерфейсу розуміють набір уніфікованих апаратних, програмних і конструктивних засобів для реалізації ефективної взаємодії елементів мережі (системи) [5]. Сукупність апаратних і конструктивних засобів реалізує мережевий адаптер, який включає: 1) власне контролер інтерфейсу (KI); 2) драйвер фізичної лінії; 3) розняття.

Згідно моделі ISO/OSI, каналний рівень відповідає за організацію коректного обміну даними (прийом і передача повідомлень) через фізичний

(13) C2

(11) 90766

(19) UA

рівень, і реалізує наступні функції: 1) формування повідомлень для передачі; 2) виявлення і корекція помилок в прийнятих повідомленнях; 3) управління потоком даних; 4) адресація вузлів; 5) реалізація механізмів арбітражу (доступу) до середовища передачі даних (лінії зв'язку) [2].

Фізичний рівень забезпечує обмін даними між вузлами мережі [2]. Його головною функцією є побітове перетворення цифрових даних (логічного значення) в електричні сигнали, а також їх передача/приймач через фізичне середовище передачі даних. Модель ISO/OSI розділяє фізичний рівень на три підрівні: 1) інтерфейс залежний від середовища (Medium Dependent Interface - MDI); 2) пристрою фізичного середовища (Physical Medium Attachment - PMA); 3) фізичної передачі даних (Physical Signaling - PLS). Перший підрівень (MDI) визначає механічні та електричні характеристики роз'язки та лінії зв'язку. Другий підрівень (PMA) визначає характеристики електричних сигналів, що передаються в мережу (вимоги до драйвера фізичної лінії). Третій підрівень (PLS) визначає процеси передачі/прийому бітів, що передбачає реалізацію трьох наступних функцій: 1) кодування/декодування біта (Bit Encoding/Decoding); 2) тактової синхронізації біта (Bit Timing); 3) синхронізація потоку даних (Synchronization). Ефективна реалізація вказаних функцій гарантує високу достовірність прийому даних на рівні біта.

На даний час існує більше десятка інтерфейсів, що знайшли широке застосування в системах з розподіленими обчислювальними ресурсами. Їх поява зумовлена широким різноманіттям умов експлуатації, потужності потоку даних і допустимих затрат на реалізацію. Тому системи в загальному не можуть задовольнитись одним універсальним, економічно ефективним рішенням. Більшість інтерфейсів стандартизовані. Це вимагає від виробників обладнання забезпечити підтримку цілого набору поширених інтерфейсів.

Традиційно набір інтерфейсів реалізують апаратно або програмне.

Відомі рішення, де набір інтерфейсів реалізують апаратно - використовуючи набір спеціалізованих мікросхем контролерів інтерфейсу та драйверів фізичної лінії [6]. Цей підхід до реалізації набору інтерфейсів має ряд переваг: 1) висока швидкість обміну даними; 2) висока достовірність прийому даних; 3) мінімальне завантаження процесора вузла мережі процесами, пов'язаними із обміном повідомленнями. Однак: 1) значно зростають апаратні затрати і площа друкованої плати, які є затратами постійними (входять в кожен обчислювальний вузол окремо); 2) зростає енергоспоживання та габарити вузла, що теж можна віднести до постійних затрат. В результаті таке рішення є найдорожчим.

Забезпечити підтримку набору вживаних інтерфейсів можна шляхом використання мікроконтролера із необхідним набором вбудованих інтерфейсів [7]. Однак зазвичай мікроконтролери не мають широкого набору інтерфейсів, а ті, що мають їх багато - дорогі. Тому таке рішення дешевше за попереднє, однак ціна таких мікроконтролерів

висока і досить суттєво впливає на ціну обчислювального вузла.

Іншим рішенням є програмна реалізація набору інтерфейсів - набір функцій каналного рівня та підрівня фізичної передачі даних кожного інтерфейсу програмне виконує процесор вузла мережі. Він виконує набір підпрограм, кожна із яких реалізує одну із функцій контролера інтерфейсу. Перевагою такого підходу є мінімальні апаратні затрати, оскільки необхідно використати лише набір мікросхем-драйверів фізичної лінії [8, 9]. Зміна інтерфейсу вимагає лише переходу на інший набір підпрограм, що може бути розміщений в постійній пам'яті мікроконтролера або завантажених із сервера мережі. При цьому різко зростає складність програмного забезпечення - підпрограми інтерфейсу мусять виконуватися в реальному часі. Проте затрати на нього - разові (під час розробки вузла), тому таке рішення найдешевше. Однак воно дозволяє реалізувати з допомогою широко розповсюджених дешевих мікроконтролерів тільки низькі швидкості обміну даними. Підвищення продуктивності інтерфейсів (якщо і можливе) вимагає використання швидкісних мікроконтролерів, що суттєво збільшує постійні затрати.

Прототипом запропонованого пристрою є програмовані модулі вводу-виводу UNIO фірми Fastwel [10]. Ці модулі реалізовані на базі програмованих логічних матриць (ПЛМ), які користувач може запрограмувати під один із необхідних інтерфейсів, використовуючи бібліотеку готових підпрограм (розміщених в постійній пам'яті мікроконтролера або завантажених із сервера мережі). При необхідності, користувач може розробити свою програму, що реалізує необхідний інтерфейс. Таке рішення поєднує переваги програмного і апаратного, оскільки у вузлі є апаратно реалізований контролер інтерфейсу, але його можна легко переналадити на потрібний інтерфейс, як і при програмній реалізації. Однак в прототипі контролер інтерфейсу виконує і функції каналного рівня, і підрівня фізичної передачі даних (PLS), що вимагає реалізації набору незалежних цифрових автоматів, кожен із яких реалізує один із вказаних процесів. Це веде до зростання апаратної складності контролера інтерфейсу, що вимагає ПЛМ значного об'єму і ціни.

Метою винаходу є мінімізація затрат при реалізації універсального контролера послідовного інтерфейсу на програмованій логічній матриці за рахунок оптимального розподілу функцій каналного рівня і підрівня фізичної передачі даних між мікроконтролером і програмованою логічною матрицею, що дало можливість мінімізації сумарних апаратних затрат на реалізацію обчислювального вузла.

Суть запропонованого пристрою полягає в тому, що в програмованій логічній матриці програмно-апаратного контролера інтерфейсу (який складається з послідовно з'єднаних мікроконтролера, програмованої логічної матриці та драйвера лінії зв'язку) реалізовано схему, куди входять три тригери, детектор початку повідомлення і послідовно з'єднані два подільники частоти генератора тактових імпульсів мікроконтролера, лічильник квантів

часу і реєстр зсуву. Паралельний вихід реєстра зсуву підключено до шини даних мікроконтролера, а вхід послідовних даних - до виходу драйвера лінії зв'язку, до якого підключено також вхід детектора початку повідомлення. Вихід останнього підключено до першого входу встановлення першого тригера, вихід якого підключено до першого входу переривання мікроконтролера і входів скидання другого подільника і лічильника квантів часу. Вихід лічильника квантів часу підключено також до входу запису другого тригера і другого входу переривання мікроконтролера. При цьому відповідні виходи мікроконтролера підключені до блокуючого входу детектора початку повідомлення, другого входу встановлення і входу скидання першого тригера, інформаційного входу і входу запису третього тригера. Вихід третього тригера підключено до інформаційного входу другого тригера, вихід якого підключено до входу драйвера лінії зв'язку.

В запропонованому програмно-апаратному контролері інтерфейсу мікроконтролер реалізує функції каналного рівня, а також підрівня фізичної передачі даних (PLS), що дозволяє розглядати процеси прийому і передачі повідомлень як набір окремих і незалежних процесів, які взаємодіють між собою. Два із них реалізують обробку і формування повідомлень в цілому (каналний рівень), а два інші обробку і формування біта (підрівень фізичної передачі даних). Тривалість вказаних процесів є детермінованою в часі і визначається типом інтерфейсу та заданою швидкістю обміну даними.

Фізичний рівень визначає спосіб фізичного з'єднання вузлів. На фізичному рівні виконуються процеси передачі та/або прийому послідовного потоку бітів, що передбачає реалізацію трьох наступних функцій: 1) кодування/декодування біта (Bit Encoding/ Decoding); 2) тактової синхронізації біта (Bit Timing); 3) синхронізація потоку даних (Synchronization). Ефективна реалізація вказаних функцій у запропонованому контролері інтерфейсу гарантує високу достовірність прийому даних на рівні біта.

Таким чином, мета винаходу в запропонованій схемі досягнута за рахунок роздільного виконання функцій каналного рівня і підрівня фізичної передачі даних, тобто спеціалізації функцій компонентів відповідно до їх особливостей. Оскільки функції каналного рівня пов'язані із логічною обробкою послідовності бітів, то їх виконують програмне (їх реалізація покладена на процесор мікроконтролера). В той же час функції підрівня фізичної передачі даних пов'язані із процесами передачі/прийому в реальному часі кожного окремого біта - їх реалізовано апаратно на ПЛМ. Таке рішення одночасно:

1. спрощує схему, яку потрібно реалізувати в ПЛМ, через те, що ПЛМ виконує функції в реальному часі, які власне легко реалізуються апаратно, і не виконує логічної обробки, яка їй невластива;

2. різко знижує вимоги до часу реакції мікроконтролера на поступлення повідомлення (тобто до його швидкодії) і, тим самим, також спрощує програмне забезпечення, оскільки мікроконтролер

отримує до оцінки ряд відліків після прийому біта, а не створює ці відліки;

3. дозволяє реалізувати різні типи послідовних інтерфейсів, оскільки апаратна частина, яка важко реконфігурується, займається прийомом і передачею тільки окремих бітів (цей процес однаковий для всіх інтерфейсів). А формування повідомлень в цілому ("складанням" бітів) займається програмна частина, яка легко адаптується під довільний тип послідовного інтерфейсу.

Суть винаходу пояснює структурна схема запропонованого програмно-апаратного контролера інтерфейсу, представлена на Фіг.1, а на Фіг.2 показана внутрішня структурна схема, яку треба реалізувати в ПЛМ.

В склад структурної схеми запропонованого програмно-апаратного контролера інтерфейсу (Фіг.1) входять мікроконтролер МК, програмована логічна матриця ПЛМ і драйвер лінії зв'язку ДЛЗ. ДЛЗ розділяє потоки даних на дані, які поступають при прийомі (лінія RxD) і дані, які передаються (лінія TxD). МК тісно співпрацює з ПЛМ через свої виходи управління Упр, входи переривання IRQ і шляхом прийому результатів тестування біта через шину даних DB.

На Фіг.2 показана структурна схема, реалізована в ПЛМ. В неї входять два послідовно з'єднані подільники частоти генератора тактових імпульсів мікроконтролера Пд1 і Пд2, лічильник квантів часу ЛК, реєстр зсуву Рг3, детектор початку повідомлення ДПП і тригери Тг1, Тг2, і Тг3. Розглянемо процес прийому біта схемою Фіг.2. Спочатку на вході детектора початку повідомлення ДПП появляється лог. 0, який детектор визначає і його вихідний сигнал перекидає тригер Тг1. Тг1 знімає сигнал скидання з входів подільника Пд2 і лічильника квантів часу ЛК, які починають працювати під дією вихідних імпульсів подільника Пд1. Одночасно вихід Тг1 поступає на вхід переривання мікроконтролера IRQ МС для підготовки його до прийому повідомлення. Вихідні імпульси ЛК (кванти тестування біта) поступають на вхід зсуву реєстр зсуву Рг3, на вхід послідовних даних якого подано біт, що приймається. В Рг3, під дією імпульсів ЛК, що поступають на вхід зсуву, записуються значення лог. 0 і 1, які в момент імпульсів ЛК присутні на вході даних (проходить багатократне визначення значення біта). Останній імпульс ЛК поступає на другий вхід переривання мікроконтролера IRQ МС для прийому чергового біта. Мікроконтролер МК посилає на Рг3 сигнал RD, який вмикає вихід Рг3 (без дії сигналу RD вихід Рг3 знаходиться в третьому стані) і стан Рг3 поступає на шину даних МК DB МС, з якої його зчитує МК. МК, за співвідношенням лог. 0 і 1 визначає значення прийнятого біта і поміщає його в відповідну комірку пам'яті, де зберігається проміжні значення бітів прийнятого повідомлення. ЛК переповнюється, скидається в нуль і починає прийом наступного біта. Після прийому останнього біта повідомлення заданого формату (що задається підпрограмою обробки повідомлення МК) МК своїм сигналом МС3 скидає Тг1, що закінчує роботу ПЛМ. Тг1 своїм виходом утримує Пд2 і ЛК у скинутому стані.

В процесі прийому повідомлення мікроконтролер МК виконує також ряд функцій, пов'язаних із обробкою повідомлення на каналному рівні, наприклад перевірка адреси, підрахунок контрольної суми або перевірку на парність (непарність). Конкретний перелік функцій визначається протоколом обміну даними даного інтерфейсу.

При передачі повідомлення сигнал MC1 мікроконтролера МК блокує ДПП, а сигнал MC2 примує до перекидає Tг1, який знімає скидання з Пд2 і ЛК. В тригер Tг3 записується значення біта, який має бути переданий першим, шляхом виставлення значення цього біта (сигнал MC4) і подачі імпульсу на вхід запису (сигнал MC5). Останній імпульс ЛК поступає на вхід запису тригера Tг2, в який записується значення біта з Tг3. З виходу Tг2 значення біта поступає, через вихід TxD, на драйвер лінії зв'язку. Останній імпульс ЛК поступає також на другий вхід переривання мікроконтролера IRQ MC для запису в Tг3 значення наступного біта, який буде переданий після переповнення ЛК. Процес передачі закінчується подачею сигналу MC3, що скидає Tг1. Tг1 своїм виходом утримує Пд2 і ЛК у скинутому стані.

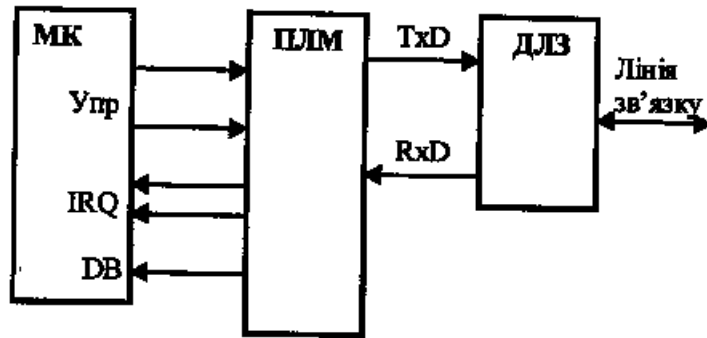
Наявність в схемі Фіг.2 детектора початку повідомлення ДПП полегшує роботу мікроконтролера МК шляхом подавлення коротких імпульсів завад, які явно не можуть бути стартовими імпульсами чергового повідомлення (довші імпульси завади відкидає мікроконтролер шляхом аналізу співвідношення прийнятих лог. 0 і 1). ДПП може бути виконаний як на базі аналогових схем (тригерів Шмідта і конденсаторних фільтрів), так і на базі їх комбінації з цифровими схемами (багатократного підтвердження наявності стартового імпульсу). В останньому випадку для роботи ДПП потрібні імпульси, які поступають з подільника Пд1 частоти тактового генератора МК. Пд1 не скидається, тому імпульси на ДПП поступають постійно, що забезпечує спрацювання ДПП в довільний момент часу. Швидкість обміну в запропонованому програмно-апаратному контролері інтерфейсу задається коефіцієнтами поділу подільників Пд1, Пд2 і лічильника квантів часу ЛК. Для її зміни при налагодженні системи необхідно перепрограмувати ПЛМ з допомогою зовнішнього програматора. Можлива також зміна швидкості обміну в процесі роботи, тоді ПЛМ перепрограмує мікроконтролер МК. Тоді він повинен зберігати в своїй постійній пам'яті прошивку ПЛМ, що дуже суттєво збільшує її об'єм.

Слід відзначити універсальність запропонованого рішення. При необхідності зміни протоколу обміну даними або інтерфейсу (при використанні аналогічної лінії зв'язку), то це вимагає зміни тільки програмного забезпечення. Апаратне забезпечення, реалізоване ПЛМ, є тільки мінімально необхідним для прийому і передачі біта. Однак, на відміну від контролерів інтерфейсу реалізованих програмне [8, 9, 11], в запропонованому ефективні-

ше використовується МК, оскільки для формування часових затримок при прийомі та передачі біта використані не програмні, а апаратні засоби. Це дозволяє значно (майже на порядок) підвищити граничну швидкодію інтерфейсу для даного типу мікроконтролера, або використати час, який витрачав мікроконтролер на реалізацію програмного таймера, для виконання прикладного програмного забезпечення.

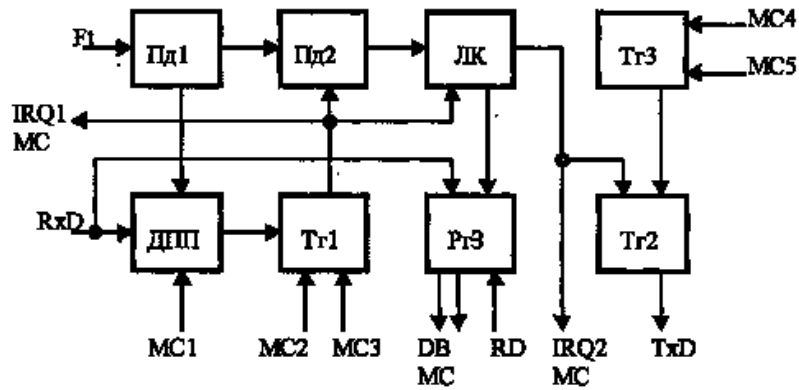
Джерела інформації:

1. К. Кругляк Промышленные сети: цели и средства. - Современные технологии автоматизации, №4, 2002. - с.6-17
2. Компьютерные сети. 4-е изд. / Э. Таненбаум. СПб.: Питер, 2003. - 992с.
3. Е.Карпенко Возможности CAN-протокола. - Современные технологии автоматизации, №4, 1998.-с. 16-20
4. С. Хвоц, Х. Амаду Промышленные сети на базе стандарта MIL-STD-1553B. - Современные технологии автоматизации, №1, 1999.- 42-45с.
5. Интерфейсы систем обработки данных: Справочник / А.А. Мячев, В.Н. Степанов, В.К. Щербко; Под ред. А.А. Мячева. - М.: Радио и связь, 1989. - 416с.
6. D. Wobshall. An Implementation of IEEE 1451 NCAP for Internet Access of Serial Port-Based Sensors, Proceedings of second Sensor for Industry Conference Sicon/02, 19-21 November 2002, Houston, Texas, ISBN 1-55617-834-4, pp. 157-160.
7. LPC2292/LPC2294. 16/32-bit ARM microcontrollers; 256 kB ISP/IAP flash with CAN, 10-bit ADC and external memory interface. Data sheet. Philips, Rev. 0.3 - 26 September, 2005. [http://www.nxp.com/#/homepage/cb=\[t=p,p=50809/45994,i=LPC2292_2294_6\]pp=\[v=p,t=pip,i=LPC2292_2294_6,fi=45994,ps=0\]](http://www.nxp.com/#/homepage/cb=[t=p,p=50809/45994,i=LPC2292_2294_6]pp=[v=p,t=pip,i=LPC2292_2294_6,fi=45994,ps=0])[3]
8. 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. May 18-20, pp.1188-1193.
9. R. Kochan, V. Kochan, A. Sachenko, A. Stepanenko, I. Maykiv "Interface and Reprogramming Controller for Dynamically Reprogrammable Network Capable Application Processor (NCAP)" Proceedings of third IEEE international workshop on Intelligent Data Acquisition and Advancing Computing Systems (IDAACS'2005), September 5-7, 2005, Sofia, Bulgaria, pp.639-642.
10. В. Беломытцев, М. Калаш Программируемые интегральные схемы в модулях фирмы Fastwel. - Современные технологии автоматизации, №3, 2001. - с.80-84
11. Микропроцессорные средства производственных систем / В.Н. Алексеев, А.М. Коновалов, В.Г. Колосов и др.; Под общ. ред. В.Г. Колосова. - Л.: Машиностроение. Ленингр. Отд-ние, 1988. - 287с.



Структурна схема пропонованого програмно-апаратного контролера інтерфейсу

Фіг. 1



Внутрішня структурна схема, реалізована в ПЛМ

Фіг. 2