

УДК 004.415

І. Білоусов¹, канд. техн. наук; В. Кочан², канд. техн. наук; І. Майків¹

¹Тернопільський національний економічний університет

²Тернопільський державний технічний університет імені Івана Пулюя

ПРОЕКТНИЙ АНАЛІЗ МЕТОДІВ РЕАЛІЗАЦІЇ КОНТРОЛЕРІВ ПОСЛІДОВНИХ ІНТЕРФЕЙСІВ

У статті наведено результати теоретичного аналізу та багатоваріантної реалізації обчислювальної системи на базі програмовано логічної матриці, що включає мікроконтролер та контролер інтерфейсу I-Wire. Визначено оптимальні рішення для випадків: 1) мінімізації апаратних витрат; 2) мінімального завантаження мікроконтролера; 3) оптимізації програмно-апаратних витрат; 4) скорочення затрат часу на проектування системи.

Ключові слова: контролер інтерфейсу, мікроконтролер, програмована логічна матриця.

I. Bilousov, V. Kochan, I. Maykiv

THE PROJECT ANALYSIS OF METHOD OF REALIZATION OF CONTROLLERS OF SERIAL INTERFACES

The results of the theoretical analysis and multiple implementation of the computing system which is based on FPGA and includes the microcontroller and the controller of the I-Wire interface are submitted in article. From the obtained results the set of optimal decisions are determined for cases: 1) minimization of hardware expenses; 2) the minimal loading of the microcontroller; 3) optimization of hardware-software expenses; 4) the reduction of expenses of time for system design.

Key words: interface controller, microcontroller, programmable logic device.

Умовні позначення

K – кількість апаратно реалізованих вузлів контролера інтерфейсу;
 M – кількість програмно реалізованих вузлів контролера інтерфейсу;
 $N_{МК}$ – апаратні витрати на реалізацію мікроконтролера, макрокомірки;
 $N_{КІ}$ – апаратні витрати на реалізацію контролера інтерфейсу, макрокомірки;
 N_I – апаратні витрати на реалізацію одного розряду шини даних в контролері інтерфейсу;
 N_i – апаратні витрати на реалізацію i -го функціонального вузла контролера інтерфейсу;
 $N_{ЦА}$ – апаратні витрати на реалізацію цифрового автомата в контролері інтерфейсу;
 $O_{ПП}$ – обсяг прикладної програми, байт;
 $O_{П}$ – обсяг програми підтримки інтерфейсу, байт;
 $O_{ЦА}$ – обсяг програми, що реалізує цифровий автомат контролера інтерфейсу, байт;
 O_i – обсяг програми, що реалізує i -й функціональний вузол контролера інтерфейсу, байт;
 $R_{ШД}$ – розрядність шини даних контролера інтерфейсу;
 $t_{біт}$ – час передачі біта;
 t_{CRC} – час підрахунку контрольної суми;
 $t_{обм}$ – час, витрачений на виконання програмної частини контролера інтерфейсу;
 $z_{ап}$ – апаратні витрати на реалізацію системи;
 $z_{пр}$ – програмні витрати на реалізацію системи, байт;
 $z_{сум}$ – сумарні витрати на реалізацію системи;
 KI – контролер інтерфейсу;
 $ЛЕ$ – логічний елемент;
 $МК$ – мікроконтролер;
 $м.к.$ – макрокомірка;
 $ПЛК$ – програмований логічний контролер;
 $ПЛМ$ – програмована логічна матриця;
 $РЗП$ – регістр загального призначення;
 $ЦА$ – цифровий автомат;
 $ШД$ – шина даних.

Актуальність питання. Сучасні вимірювально-керуючі системи реалізуються як багаторівневі системи з розподіленими обчислювальними ресурсами на базі поширених інтерфейсів (Ethernet, CAN, LIN, 1-Wire та ін.) [1, 2]. Їх базовим елементом є *ПЛК*, до якого висувається ряд вимог, головними з яких є: 1) поточна обробка даних у реальному часі; 2) підтримка набору поширених інтерфейсів. Традиційно *ПЛК* реалізують на базі промислових комп'ютерів, що дозволяє забезпечити одночасну реалізацію вищевказаних функцій [3]. Однак кінцева вартість таких *ПЛК* є достатньо високою. Перспективним напрямком є реалізація *ПЛК* на базі *МК*, вартість яких постійно зменшується одночасно із розширенням їх функціональних можливостей. Однак більшість недорогих *МК* підтримують обмежений набір інтерфейсів. Тому на практиці необхідний інтерфейс реалізують програмно або апаратно.

При апаратній реалізації інтерфейсу використовують спеціалізовану мікросхему *КІ*, що дозволяє забезпечити високу швидкість обміну та достовірність прийому даних, а також низький рівень завантаження *МК* процесами, пов'язаними із обробкою повідомлень. Однак при цьому зростає апаратна складність, енергоспоживання, габарити та вартість обладнання.

Програмний метод реалізації інтерфейсів є ефективним щодо мінімізації апаратних витрат та підвищення гнучкості системи, однак не забезпечує високу достовірність прийому даних. Її підвищення шляхом ускладнення алгоритмів обробки не завжди можливе через обмежену продуктивність дешевих *МК*.

У багатьох випадках при реалізації *КІ* доцільно використовувати переваги як апаратного, так і програмного методів, що дозволяє оптимізувати витрати на розробку системи.

До ухвалення рішення про реалізацію проекту необхідно розглянути всі його аспекти, в тому числі використання типових проектних рішень як для окремих вузлів, так і системи в цілому. Це дає можливість знизити трудомісткість проектування, зменшити вартість системи та підвищити її надійність.

Мета дослідження. Метою даних досліджень є аналіз співвідношення витрат апаратних та програмних ресурсів при реалізації *КІ* та вироблення проектних рішень для їх подальшого використання при розробці конкретних проектів. Це вимагає проведення попереднього аналізу витрат на реалізацію *КІ* та вибору оптимального варіанту реалізації *КІ* для кожного конкретного випадку в залежності від вибраної умови: 1) мінімізації апаратних витрат; 2) мінімального завантаження *МК*; 3) оптимізації апаратно-програмних витрат; 4) скорочення витрат часу на проектування системи.

Результати досліджень. Сумарні витрати при реалізації системи є функцією апаратних і програмних витрат:

$$z_{\text{сум}} = f(z_{\text{ан}}, z_{\text{пр}}).$$

При цьому апаратні витрати є функцією використаних апаратних ресурсів на реалізацію *МК* та *КІ*:

$$z_{\text{ан}} = f_1(N_{\text{МК}}, N_{\text{КІ}}) = N_{\text{МК}} + N_{\text{КІ}}, \text{ де } N_{\text{КІ}} = R_{\text{шд}} \cdot N_1 + N_{\text{ца}} + \sum_{i=1}^K N_i.$$

У випадку програмної реалізації *КІ* $N_{\text{КІ}} = 0$, оскільки *ЦА* та всі функціональні вузли *КІ* реалізуються програмно.

Програмні витрати визначаються обсягом прикладної програми та програми, що реалізує підтримку *КІ*:

$$z_{\text{пр}} = f_2(O_{\text{ПП}}, O_{\text{П}}) = O_{\text{ПП}} + O_{\text{П}}, \text{ де } O_{\text{П}} = O_{\text{ца}} + \sum_{i=1}^M O_i.$$

У випадку апаратної реалізації *КІ* $O_{\text{ца}} = 0$.

Узагальнена структурна схема системи, (рис.1), включає *МК* - виконує обробку даних, та *КІ* – забезпечує інформаційну сумісність обладнання при обміні даними [4]. Обмін даними між *МК* та *КІ* виконується по шині даних, а керування роботою *КІ* здійснюється по шині керування. Якщо обмін даними між *МК* та *КІ* здійснюється по перериваннях, то, отримавши повідомлення, *КІ* генерує сигнал переривання (*INT*), який поступає на *МК*.

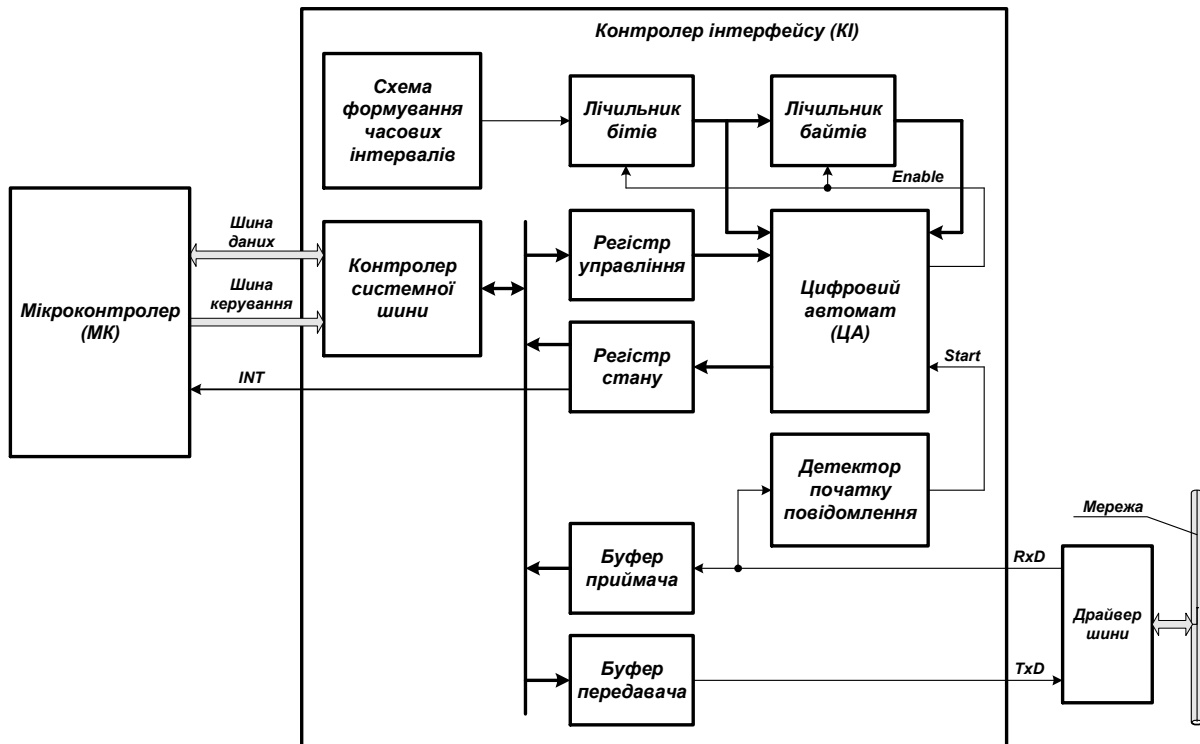


Рисунок 1 - Узагальнена структурна схема системи

Також на рис.1 представлено узагальнену структурну схему *КІ*. *ЦА* реалізує процес передачі/прийому повідомлення. Контролер системної шини керує потоками даних між *МК* та внутрішніми регістрами *КІ*. Схема формування часових інтервалів генерує набір сигналів, необхідних для синхронізації процесів прийому та передачі. Лічильник бітів здійснює підрахунок прийнятих/переданих бітів. Лічильник байтів застосовується у тому випадку, якщо довжина повідомлення є достатньо великою (декілька десятків бітів або більше). Регістр управління містить необхідну інформацію про дані, прийняті від *МК* для передачі, та режим обміну даними в мережі. Регістр стану містить інформацію про режим роботи *КІ* та стан процесу обміну даними в мережі. Буфер приймача та передавача містять дані, що були прийняті або які необхідно передати. Детектор початку повідомлення фіксує ознаку початку повідомлення, після чого генерує сигнал *Start*, яким переводить цифровий автомат *КІ* в режим прийому повідомлення. Останній сигналом *Enable* включає лічильники бітів та байтів і переходить у режим прийому повідомлення.

На представленій схемі наведено мінімальний набір функціональних вузлів, що необхідні для реалізації будь-якого *КІ*. В залежності від складності протоколу обміну їх кількість може змінюватись і розширюватись. Кожен із вищевказаних вузлів може бути реалізований як програмно, так і апаратно, що в цілому дає достатньо велику множину варіантів реалізації *КІ*. Виходячи із цього, в [5] проведено аналіз процесів, які реалізує *КІ*. Результати аналізу показали, що процеси прийому та передачі можна розглядати як набір незалежних процесів, які взаємодіють між собою. Два процеси верхнього рівня виконують: 1) обробку повідомлення при прийомі; 2) формування повідомлення при передачі. Одночасно два інших процеси нижнього рівня здійснюють обробку біта при прийомі/передачі як складової частини повідомлення. Вказані групи процесів є

принципово різними. Процеси прийому та передачі біта пов'язанні з обробкою інформації в часі i , відповідно моделі ISO/OSI [6], реалізують функції фізичного рівня. Одночасно процеси обробки/формування повідомлення пов'язані з їх логічною обробкою i , відповідно моделі ISO/OSI, реалізують набір функцій канального рівня.

Виходячи із вищесказаного, в [5] запропоновано програмно-апаратний метод реалізації *KI*, який передбачає програмну реалізацію функцій канального рівня та апаратну реалізацію функцій фізичного рівня.

Дослідження ефективності різних методів реалізації *KI* виконано на прикладі інтерфейсу 1-Wire [7], на базі “Spartan-3E Development Kit” [8], що містить ПЛМ серії Spartan-3E обсягом 500 тис. ЛЕ. Реалізацію здійснено на мові VHDL з використанням ядра 8-ми розрядного RISC МК PicoBlaze [9], наданого фірмою XILINX. В роботі [10] представлено п'ять варіантів обчислювальної системи, що включає МК та *KI* 1-Wire, який реалізовано наступним чином:

- програмний *KI* ;
- апаратний *KI*;
- програмно-апаратний *KI*, із 8-ми розрядною ШД;
- програмно-апаратний *KI*, із однорозрядною ШД;
- програмно-апаратний *KI* на базі контролера інтерфейсу RS232.

Аналіз отриманих результатів. Результати, отримані при реалізації вище перерахованих варіантів систем на базі ПЛМ, представлено в таблиці. В стрічках таблиці наведено перелік параметрів, які відображають витрати на реалізацію системи. Дані, наведені в таблиці, відображають абсолютне (чисельник) та відносне (знаменник) значення. Базовим варіантом для оцінки всіх наступних рішень є система із програмно реалізованим *KI*, в якій найбільші програмні і найменші апаратні витрати.

Таблиця - Витрати на реалізацію системи

	Параметр	Програм-ний	Апарат-ний	Програмно-апаратний		
				із 8-ми розр. ШД	із одно-розр. ШД	на базі RS232
		1	2	3	4	5
Програмні витрати						
1	Загальний обсяг програми – z_{np} (байт)	376 / 1	313 / 0,83	309 / 0,82	319 / 0,85	331 / 0,88
2	Обсяг програми для підтримки інтерфейсу – $z_{пв}$ (байт)	107 / 1	31 / 0,29	29 / 0,27	39 / 0,36	51 / 0,47
3	Використано РЗП із 16 доступних	7 / 1	3 / 0,42	4 / 0,57	6 / 0,85	7 / 1
Апаратні витрати						
4	К-сть використаних м.к. - $z_{ан}$	97 / 1	175 / 1,8	153 / 1,57	138 / 1,42	140 / 1,44
5	К-сть використаних тригерів	78 / 1	177 / 2,26	131 / 1,68	107 / 1,37	131 / 1,67
Часові результати						
6	Час, витрачений МК на один цикл обміну даними - $t_{обм}$, (мкс),	6560 / 1	5,7 / 0,00086	5,2 / 0,00079	41 / 0,00625	44,5 / 0,0067
7	Максимальна робоча частота (МГц)	120,98 / 1	119,33 / 0,985	120,48 / 0,99	120,65 / 0,99	86,55 / 0,71

Із представлених результатів видно, що система із програмно реалізованим інтерфейсом має найменші апаратні витрати – 97 м.к. і найбільший обсяг програми - 376 байт, із яких 107 байт (28,4% від загального обсягу) використано для реалізації набору підпрограм підтримки інтерфейсу. Також найнижчою є ефективність

використання *МК*, оскільки час, витрачений *МК* на виконання команди *Read_ROM*, відповідає часу виконання самої команди (6560 мкс).

Наступною за апаратними витратами є система із програмно-апаратним контролером шини 1-Wire з однорозрядною ШД. Для реалізації використано 138 м.к., що на 42,3% більше мінімального значення. Загальний обсяг програм становить 319 байт (на 15% менше максимального значення), з яких лише 39 байт (12% від загального обсягу) використано для програми підтримки інтерфейсу. Вищою є і ефективність використання *МК*, час витрачений на обробку повідомлення становить 41 мкс, або 0,6% від загального часу.

Для реалізації системи із програмно-апаратним *КІ* на базі інтерфейсу RS232 використано 140 м.к., що на 44% більше мінімального. Загальний обсяг програм становить 331 байт (на 12% менше максимального значення), з яких 51 байт (15% від загального обсягу) використано для програми обробки переривань. Зростання обсягу програми підтримки інтерфейсу пов'язане із ускладненням алгоритму обробки прийнятого біта, внаслідок чого зменшилась ефективність використання *МК*, оскільки час обробки повідомлення становить 44,5 мкс, або $\approx 0,7\%$ від загального часу.

Наступною за апаратними витратами є система із програмно-апаратним контролером шини з 8-ми розрядною ШД. Для реалізації використано 153 м.к., що на 57% більше мінімального значення. Загальний обсяг програм становить 309 байт (на 18% менше максимального значення), з яких лише 29 байт (9,3% від загального обсягу) використано для програми підтримки інтерфейсу, що реалізує підтримку інтерфейсу. Система забезпечує найвищу ефективність використання *МК*, оскільки час, витрачений на обробку повідомлення, становить 5,2 мкс, або менше 0,1% від загального часу виконання команди *Read_ROM*.

Найбільші апаратні витрати має система із незалежним, апаратно реалізованим *КІ*, який включає в себе набір службових регістрів. Для реалізації системи використано 175 м.к., що на 80,4% більше мінімального значення. Загальний обсяг програм становить 313 байт (на 16,7% менше максимального значення), з яких 31 байт ($\approx 10\%$ від загального обсягу) використано для підпрограми підтримки інтерфейсу. Зростання обсягу програми порівняно з програмно-апаратним варіантом із 8-ми розрядною ШД пов'язане з тим, що *МК* звертається до службових регістрів, реалізованих в *КІ*, на відміну від інших варіантів, в яких службові регістри реалізовані у *РЗП*. В результаті використано лише 3 *РЗП*, що становить 18,7% загального обсягу.

Отримані результати представлені у графічній формі на рис.2, де відображено розподіл описаних варіантів відповідно до затрат на їх реалізацію та ефективністю використання *МК*.

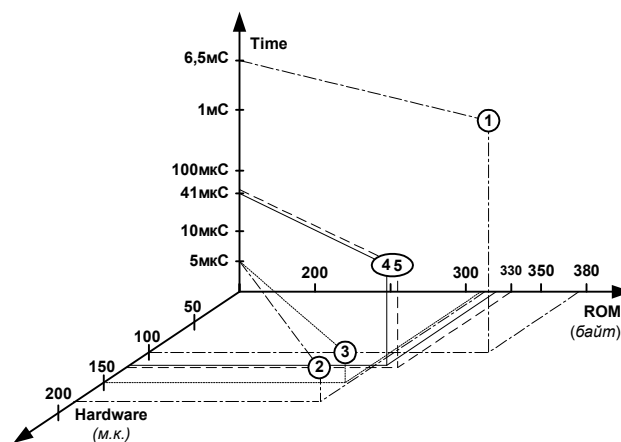


Рисунок 2 – Результати, отримані при реалізації контролерів інтерфейсу 1-Wire

Отримані результати показують, що програмний метод реалізації *КІ* при мінімальних апаратних витратах забезпечує найменш ефективне використання *МК*, оскільки $t_{ОБМ}$ відповідає часу обміну даними при виконанні команди "*Read-ROM*".

Одночасно системи з програмно-апаратним *KI* при зростанні апаратних витрат забезпечують більш ефективне використання *МК*. Так, система з програмно-апаратним *KI* з однорозрядною *ШД* при зростанні апаратних витрат на 42%, порівняно із програмно реалізованим *KI*, дає зменшення $t_{ОБМ}$ в 160 раз. Аналогічно система з програмно-апаратним *KI* з 8-ми розрядною *ШД*, при зростанні апаратних витрат на 57%, порівняно із програмно реалізованим *KI*, забезпечує зменшення $t_{ОБМ}$ в 1260 раз.

Аналіз отриманих результатів показує, що оптимальним рішенням є:

– за умови мінімізації апаратних витрат – програмна реалізація *KI*. Однак через найнижчу ефективність використання *МК* він доцільний при великому надлишку обчислювальної потужності *МК*;

– за умови мінімального завантаження *МК* – програмно-апаратний із 8-и розрядною *ШД*. Даний варіант доцільно використати, коли система розв’язує складну прикладну задачу і вимагає максимуму обчислювальної потужності *МК*;

– за умови оптимізації апаратно-програмних витрат - програмно-апаратний із одно-розрядною *ШД*. Даний варіант доцільно використати, коли при розв’язанні прикладної задачі є незначний надлишок обчислювальної потужності *МК* для програмної реалізації функцій каналного рівня (програмно-апаратна реалізація);

– за умови мінімізації витрат часу на проектування системи – програмно-апаратний на базі *KI* RS232, який, при використанні готових рішень для окремих компонентів, дозволяє скоротити час проектування системи у 2-3 рази, в залежності від кваліфікації спеціаліста.

Висновки

На прикладі інтерфейсу 1-Wire проведено проектний аналіз методів реалізації *KI*. Результати експериментів показують, що програмно-апаратний метод проектування, за рахунок роздільного виконання функцій каналного і фізичного рівнів, дозволяє оптимізувати співвідношення апаратних та програмних витрат, а також забезпечує більш ефективне використання обчислювальної потужності *МК*.

Література

1. Кругляк К. Промышленные сети: цели и средства [Текст] / К. Кругляк // Современные технологии автоматизации. - 2002. - № 4. - С. 6-17.
2. Ицкович Э.Л. Перспективные средства автоматизации и эффективность их эксплуатации [Текст] / Э.Л. Ицкович // Промышленные АСУ и контроллеры. - 2000. - № 12. - С. 1-4.
3. Перевалов А. Время думающих контроллеров уже наступило [Текст] /А. Перевалов // CHIPnews-Україна. - 2005. - №8. - С.52-56.
4. Мячев А.А. Интерфейсы систем обработки данных [Текст]: справочник / А.А. Мячев, В.Н. Степанов, В.К Щербо; под ред. А.А. Мячева. – М.: Радио и связь. - 1989. – 416 с.: ил. – ISBN 5-256-00315-1.
5. Маукив І. The Method of Software-Hardware Implementation of Serial Interfaces [Текст] / І. Маукив // Сучасні проблеми радіоелектроніки, телекомунікацій, комп’ютерної інженерії: матеріали міжнародної конференції TCSET’2008. – Львів: Вид-во нац. ун-т “Львівська політехніка. – 2008. - С.439-441. – ISBN 978-966-553-678-9
6. Буров Є. Комп’ютерні мережі / Є. Буров. - Львів: Бак, 1999. – 468 с.: іл. - ISBN 966-7065-19-7.
7. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. [Електронний ресурс] = DS18B20 / Dallas Semiconductor. – Rev. 042208. - Режим доступу: <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>.
8. Spartan-3E Starter Kit Board User Guide. [Електронний ресурс] = UG230 / Xilinx. - Rev.1.1. - June 20, 2008. – Режим доступу: http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf.
9. 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.
10. Майків І.М, Кочан В.В Оптимізація послідовних інтерфейсів, реалізованих на ПЛМ [Текст] / І.М. Майків, В.В Кочан // Вісник Хмельницького національного університету. - 2007. – Т. 1, №2. - С.117-122.

Одержано 16.01.2009 р.