

УДК 004.451.47: 65.011.46 (043.2)

Пащенко Н. В.

ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ ДЛЯ РЕАЛІЗАЦІЇ КОНТРОЛЕРА PCI ІНТЕРФЕЙСУ

Інститут комп'ютерних технологій Національного Авіаційного Університету

У статті аналізуються можливості, які надаються сучасному розробнику для реалізації своїх рішень із використанням шини PCI. Розглядаються альтернативи й вартісні характеристики існуючих підходів. Наводяться основні переваги ПЛІС типу FPGA при реалізації контролера PCI інтерфейсу.

Вступ

Складність цифрових обчислювальних пристроїв постійно зростає, а разом з нею ростуть і вимоги до передачі даних. Тому в цей час питання про можливість застосування ПЛІС для реалізації пристроїв передачі даних є досить актуальним.

Мета роботи

У даній статті аналізуються можливості, надавані сучасному розробнику при реалізації своїх рішень із використанням шини PCI, розглядаються альтернативи й вартісні характеристики існуючих підходів.

Постановка задачі

З моменту своєї розробки шина PCI здобуває все більшу популярність серед виробників комп'ютерної техніки, повністю витісняючи шину ISA з більшості сучасних материнських плат. Така популярність обумовлена, по-перше, високою продуктивністю в режимі блокових передач (до 528 Мбайт/сек для PCI 64-битий/66-МГц), по-друге, підтримкою протоколу *Plug&Play*, гнучким арбітражем шини, а також компактністю й надійністю комутаційних з'єднань (*Compact PCI*), підтримкою *JTAG IEEE1149.1*. І хоча в наш час проектуються й більш продуктивні системні шини, шині PCI за прогнозами ще довго буде віддаватися найбільший пріоритет.

Методика рішення

Як правило, контролер шини PCI виконується у вигляді окремої СБІС, наприклад, фірм *AMCC*, *PLX*, що реалізує автомати станів для керуючих і контрольних сигналів шини, набір регістрів, відповідальних за конфігураційний простір контролера, що надає розробнику певний інтерфейс користувача. У цьому випадку, розробник не дотримується строгих вимог специфікації шини з боку системного інтерфейсу, оскільки СБІС контролера забезпечує гарантовані часові характеристики й основний увагу приділяє розробці свого конкретного додатка. У той же час, для узгодження контролера із пристроєм, наприклад, через *FIFO*, розробнику необхідно використати високошвидкісну логіку додаткового керування, оскільки частина користувача стандартних СБІС контролерів не дозволяє прямо сполучати довільні пристрої, у тому числі, і сучасні процесори цифрової обробки сигналів (*DSP*). У більшості випадків, у якості "обв'язки" застосовуються мікросхеми програмованої логіки. Таким чином, недоліком даного підходу є надлишкова кількість корпусів ІС СБІС контролера.

Для багатосерійного виробництва пристроїв PCI рівня користувача, наприклад, відео- або мережних карт деякі фірми часто використовують замовлені інтегральні схеми, що реалізують на одному кристалі як безпосередньо сам контролер шини, так і специфічний додаток корис-

тувача. Основний недолік даного підходу полягає у високій вартості підготовки виробництва замовленої схеми, що часом обчислюється десятками тисяч доларів. До недоліків також варто віднести тривалий час розробки й виробництва замовленої схеми і низьку гнучкість пристрою, неможливість наступних доробок схеми без додаткових матеріальних і часових витрат.

Наступним підходом, який позбавлений недоліків обох попередніх підходів, є побудова пристрою *PCI* на базі приладів програмованої логіки. У наш час вже створені програмовані користувачем логічні інтегральні схеми з дуже високим ступенем інтеграції, застосування яких, разом з використанням сучасних мікропроцесорів і мікроконтролерів, дозволяє в короткий термін виконати розробку й почати виробництво електронної апаратури нового покоління [1].

У наш час виробники ПЛІС пропонують широкий спектр виробів, параметри яких як по швидкодії, так і по обсягу внутрішньої логіки цілком дозволяють реалізацію складного пристрою на одному кристалі. Так, наприклад, контролер *PCI* шини розрядністю 64 біт, із системною частотою 66 МГц, займає в ПЛІС фірми *Xilinx* *XCV1000* (понад 1 млн. вентилів), усього 3% від загального обсягу. Контролер *PCI* 32 біт/33 МГц має обсяг порядку 10 тис. вентилів і на невеликих ПЛІС фірми *Xilinx* займає 20-30% обсягу кристала, залишаючи розробнику значні ресурси мікросхеми. У цьому випадку, виражений системний підхід до розробки пристрою користувача інтеграція контролера й додатка користувача на одному кристалі (*System-On-Chip*) (рис. 1).

Варто відмітити, що даний напрямок одержав великий розвиток, особливо з ростом внутрішніх обсягів ПЛІС і різкого падіння цін на них. Висока надійність однокристалного рішення і, як наслідок, невисока складність друкованої плати, дає цьому підходу всі переваги реалізації

складних пристроїв на замовлених БІС, зберігаючи при цьому гнучкість системи й можливість її швидкої реконфігурації. Це і є значною перевагою даного підходу.



Рис. 1. Системна інтеграція на ПЛІС *Xilinx*

Завдяки можливості швидкого перепроектування ПЛІС, розробник може надалі в найкоротший термін провести необхідну модернізацію свого пристрою, уже переданого замовнику, не працюючи на місці безпосередньо із самим пристроєм, а виславши користувачу файл прошивання ПЛІС, що визначає порядок функціонування пристрою. Останнім часом провідними виробниками програмованої логіки в засоби проектування ПЛІС все частіше включається підтримка модернізації ПЛІС через Інтернет.

Сучасні ПЛІС надають розробнику багато архітектурних і функціональних особливостей, що роблять можливим побудову на кристалі цифрових пристроїв різного призначення й складності. Такими особливостями є:

- великий обсяг внутрішніх логічних ресурсів (до 3,2 млн. вентилів в *XCV3200E*);
- високі робочі частоти (до 210 МГц);
- наявність у ПЛІС типу *FPGA* (*Field Programmable Gate Array*) швидкодіючого (до 210 МГц) блокового й розподіленого ОЗП;
- внутрішні шини із третім станом і мінімальними затримками на перемикання;

- високошвидкісні буфери введення/виведення цифрових даних (до 310 Мбіт/виведення);
- підтримка різних периферійних стандартів з різноманітними граничними рівнями;
- убудований порт *JTAG*.

Варто згадати наявність широкого спектра корпусів різного виконання з більшим числом зовнішніх двонаправлених виводів, що дозволяє розробнику організувати широкі шини передачі даних і безліч цифрових портів.

У наш час майже кожен виробник приладів програмованої логіки поставляє той або інший варіант бібліотек інтерфейсів *PCI* або засобів їхньої генерації (наприклад, *GUI* для *LogiCORE PCI*), які використовуються як надбудови для засобів розробки ПЛІС. Вартість даних бібліотек досить велика й припускає використання їх для серійного виробництва. У вітчизняних умовах через недостатність обсягів випуску продукції й нерозвиненості виробництва даний варіант не завжди прийнятний, що змушує шукати інші шляхи рішення даної проблеми.

Існує два напрями: або реалізувати свій варіант контролера інтерфейсу *PCI*, відштовхуючись від положень специфікації *PCI* і власного досвіду, що зробити досить непросто. З іншого боку, можна звернутися до відповідного досвіду вітчизняних розробників, що пропонують свої продукти, як правило, значно дешевше західних постачальників. До того ж, бібліотеки, запропоновані західними розробниками, в основному, допускають повну модифікацію внутрішньої структури контролера *PCI*, що в більшості випадків надає надлишкові можливості, що виявляються незатребуваними.

Прикладом виходу з подібного положення може служити розроблене фірмою "Скан Інженерінг-телеком" ядро контролера *PCI* 32 біт/33 МГц *Slave*. Ядро являє собою бібліотеку для засобів проектування *Xilinx*, що містить символ конт-

ролера *PCI* із "закритим" внутрішнім умістом й асоційованим *netlist*-файлом [2]. Розробник здійснює підключення свого додатка до контролера в схемотехнічному редакторі або на *VHDL*, не вникаючи у внутрішній уміст контролера, заздалегідь оптимізованого для функціонування на шині *PCI* і з визначеними часовими й топологічними характеристиками.

Слід відпомітити, що сторона користувача контролера надає розробнику пристрою дуже гнучкий інтерфейс, що дозволяє сполучати з *PCI* пристрої різного функціонального призначення. Далі виробляється спільне функціональне моделювання інтегрованого ядра контролера з "обв'язкою" користувача і наступне трасування проекту. На рис. 2 наведений узагальнений маршрут проектування *PCI* на ПЛІС *Xilinx* з використанням пакета проектування *Foundation*.

У процесі трасування незадіяні внутрішні виводи контролера з боку додатка користувача і відповідна незадіяна логіка контролера піддаються оптимізації, що дозволяє заощадити деякий обсяг внутрішніх ресурсів ПЛІС.

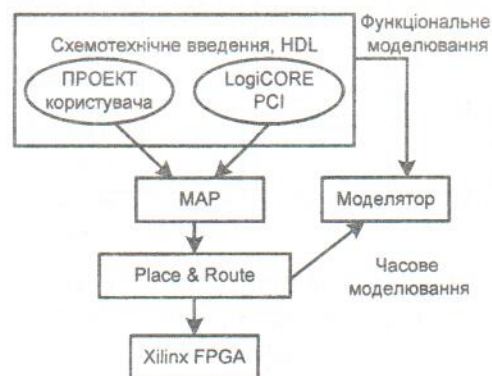


Рис. 2. Маршрут проектування ПЛІС *Xilinx* із застосуванням ядра *PCI*

Після трасування виконується тимчасове моделювання проекту з витягнутими затримками, і формується файл прошивки ПЛІС. У підсумку, розробник реалізує своє нестандартне рішення на базі *PCI* практично на одному кристалі ПЛІС, обмежуючись підключенням до

ПЛИС своїх цільових пристроїв: DSP пам'яті, портів і так далі.

Як останній аргумент на користь застосування ПЛИС *Xilinx* для реалізації інтерфейсу *PCI*, розглянемо порівняльну вартість даного рішення в порівнянні з рішенням на базі стандартної мікросхеми контролера *PCI*. Умови порівняння виберемо з метою забезпечення приблизно рівних функціональних можливостей реалізованих пристроїв.

З одного боку, виберемо СБІС *PCI Slave AMCC5920* фірми *Applied Micro Circuits* з конфігураційною *EEPROM* (16 Кбит *Atmel*), яка має зовнішнє *FIFO* 16 x 1k (високошвидкісний ОЗП з керуванням через *CPLD*), інтерфейсну *CPLD Xilinx XC95288* обсягом 6,4 тис. вентилів.

З іншого боку, ПЛИС *Xilinx* серії *Spartan XCS20* (20 тис. вентилів) з конфігураційною *PROM* плюс вартість логічного ядра контролера, перенесена на кількість виробів, що випускають (рис. 3).

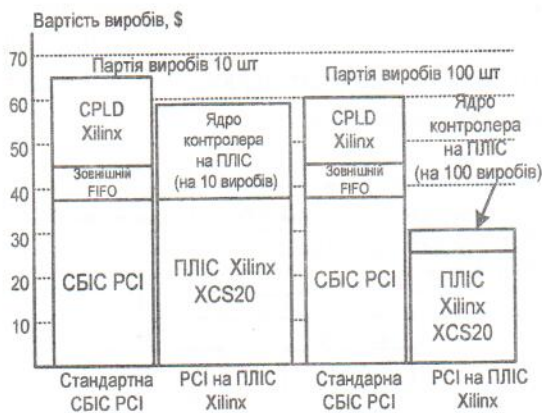


Рис. 3. Порівняльні вартісні характеристики двох різних підходів реалізації *PCI* інтерфейсу

Аналіз рис. 3 показує, що при рівних функціональних можливостях наведених рішень значну частку вартості рішення на ПЛИС при малих партіях вносить ядро *PCI* контролера. Однак перенесена на один виріб вартість ядра контролера зменшується пропорційно росту обсягу випуску, і вже на партії в сто виробів проявляється значна перевага підходу реалізації *PCI* на ПЛИС *Xilinx*. Крім того, ядро контролера, завдяки гнучкій конфігурації збоку додатка користувача, припускає використання в широкій номенклатурі виробів на базі *PCI* інтерфейсу (відповідний ріст обсягу випуску), що веде до швидкої самоооплатності ядра.

Висновки

Розглянуте інтегроване рішення *PCI* інтерфейсу на ПЛИС фірми *Xilinx* надає користувачу високу гнучкість апаратного ядра, всі переваги однокристалного рішення і при невисокій вартості вже на дрібносерійному виробництві робить даний підхід прийнятним рішенням для сучасного вітчизняного розробника.

Список літератури

1. В. Капитанов, В. Мистюков, С. Довгань. Контроллер *PCI* інтерфейса на ПЛИС фірми *Xilinx* // *ChipNews* №3, 2003 г.
2. В. Бродин, А. Калинин, Л. Хохлов, И. Шагурин. Комплекс средств для обучения проектированию цифровых устройств на ПЛИС *Altera* с использованием системы *Max+plusII* // *ChipNews* №2, 2000 г.