

УДК 004.31

А.П. Здолбіцький, В.В. Місюк

Луцький національний технічний університет

СИСТЕМА НА КРИСТАЛІ AMBER ARM В БАЗИСІ ПЛІС

Здолбіцький А.П., Місюк В.В. Система на кристалі Amber ARM в базисі ПЛІС. У рамках підходу до розробки систем на кристалі реалізовано RISC-ядро з набором інструкцій ARMv2 на базі програмованої логічної схеми Cyclone III фірми Altera.

Ключові слова: FPGA, система на кристалі (СНК), програмована логічна схема.

Форм. 0. Рис. 2. Літ. 10.

А.П. Здолбіцький, В.В. Мисюк. Система на кристалі AMBER ARM в базисі ПЛІС. В рамках підходу до розробки систем на кристалі реалізовано RISC-ядро з набором інструкцій ARMv2 на базі програмуваної логічної схеми Cyclone III фірми Altera.

Ключевые слова: FPGA, система на кристалле (СНК), программируемая логическая схема.

A.P. Zdolbitskiy, V.V. Misyuk. System on Chip AMBER ARM in the base FPGA. As part of the approach to the development of systems-on-chip implemented with RISC-core instruction set ARMv2 based on programmable logic circuit Cyclone III company Altera.

Keywords: FPGA, system-on-chip (SoC), programmable logic circuit.

Аналіз систем та порівняльна характеристика FPGA. Програмовані логічні інтегральні схеми (ПЛІС – Programmable Logic Devices – PLD) являють собою нову елементну базу, яка володіє гнучкістю замовних ВІС і доступністю традиційної «жорсткої» логіки [3,10]. Головною відмінною властивістю ПЛІС є можливість їх налаштування на виконання заданих функцій самим користувачем. Сучасні ПЛІС характеризуються низькою вартістю, високою швидкістю (до 3,5 нс), значними функціональними можливостями (одна ПЛІС може замінити кілька сотень корпусів традиційної «жорсткої» логіки), багатократністю перепрограмування, низькою споживаною потужністю (що дозволяє використовувати їх у виробках з живленням від батарейок), гнучкістю архітектури та ін.

Підтвердженням перспективності нової елементної бази служить щорічна поява нових поколінь ПЛІС, що мають більш досконалу архітектуру, а також постійно зростаючий обсяг випуску ПЛІС провідними зарубіжними виробниками мікросхем: Advanced Micro Devices (AMD), Altera, Xilinx, Atmel, Intel, Texas Instruments та ін [3,10].

Процес проектування цифрового пристрою на основі ПЛІС полягає в описі його функціонування на вхідній мові використовуваного програмного засобу, виконанні автоматизованого синтезу, проведенні моделювання та налаштування обраної ПЛІС за допомогою програматора. При цьому час розробки навіть досить складних проектів може становити всього кілька годин. Для того щоб змінити алгоритм роботи пристрою, достатньо перепрограмувати ПЛІС.

Розробка пристроїв на основі ПЛІС являє собою нову технологію проектування електронних схем, включаючи їх виготовлення і супроводження.

Абревіатура FPGA означає Field Programmable Gate Array – програмована користувачем вентильна матриця. Програмована користувачем вентильна матриця, ПКВМ (англ. Field-Programmable Gate Array, FPGA) – напівпровідниковий пристрій, що може бути налаштований виробником або розробником після виготовлення; звідси назва: «програмується користувачем». ПКВМ програмується шляхом зміни логіки роботи принципової схеми, наприклад, за допомогою вихідного коду мовою проектування (типу VHDL), на якому можна описати цю логіку роботи мікросхеми. ПКВМ є однією з архітектурних різновидів програмованих логічних інтегральних схем (ПЛІС).

ПКВМ можуть бути модифіковані практично в будь-який момент у процесі їх використання. Вони складаються з конфігурованих логічних блоків, подібних перемикачів з безліччю входів і одним виходом (логічні вентиля або gates). У цифрових схемах такі перемикачі реалізують базові виконавчі операції AND, NAND, OR, NOR і XOR. У більшості сучасних мікропроцесорів функції логічних блоків фіксовані і не можуть модифікуватися. Принципова відмінність ПКВМ полягає в тому, що і функції блоків, і конфігурація з'єднань між ними можуть змінюватися за допомогою спеціальних сигналів, що посилюються схемою. У деяких спеціалізованих інтегральних схемах (ASIC) використовуються логічні матриці, аналогічні ПКВМ за структурою, однак вони конфігуруються один раз в процесі виробництва, у той час як ПКВМ можуть постійно перепрограмувати і міняти топологію з'єднань в процесі використання. Однак, така гнучкість вимагає істотного збільшення кількості транзисторів мікросхеми.

ПКВМ включають в себе три головних програмованих елементи: незкомутовані програмовані логічні блоки (ПЛБ), блоки вводу-виводу (БВВ) і внутрішні зв'язки. ПЛБ є функціональними елементами для побудови логіки користувача, БВВ забезпечують зв'язок між контактами корпусу і внутрішніми сигнальними лініями. Програмовані ресурси внутрішніх зв'язків забезпечують управління шляхами сполучення входів і виходів ПЛБ і блоків вводу-виводу на відповідні мережі. Всі канали трасування мають однакову ширину (однакова кількість провідників). Більшість блоків БВВ вписуються або в один рядок (по висоті), або в один стовпець (по ширині) масиву вентилів.

Логічний блок класичної ПКВМ складається з таблиці пошуку на 4 входи і тригера. В останні роки виробники почали переходити на таблиці пошуку з 6 входами у високопродуктивних частинах схеми, пояснюючи це необхідністю підвищення продуктивності.

Логічний блок має таблицю пошуку на 4 входи і вхід синхронізації (clock). Вихід блоку тільки один, це може бути регістрова або нерегістрова вихідна таблиця пошуку. Оскільки сигнали синхронізації в комерційних ПКВМ (а часто й інші сигнали, розпаралелюючих на велику кількість входів – high-fanout signals) трасуються особливим чином спеціальними трасувальними ланцюгами, управління цими сигналами робиться окремо.

Входи розташовані на окремих сторонах логічного блоку, вихідний контакт може трасуватися у двох каналах: або праворуч від блоку, або знизу. Вихідні контакти кожного логічного блоку можуть з'єднуватися з трасувань сегментами в суміжних каналах. Аналогічно, контактний майданчик блоку вводу-виводу (pad) може з'єднуватися з трасуванням елементом у будь-якому суміжному каналі. Наприклад, верхній контактний майданчик чіпа може з'єднуватися з будь-яким з W провідників (де W – ширина каналу) в горизонтальному каналі, розташованому безпосередньо під ним.

Як правило, трасування ПКВМ несегментоване, тобто кожен сегмент провідника з'єднує лише один логічний блок з перемикаючим блоком. Через обхід програмованих перемикачів у блоці перемикача трасування виходить більш довгим. Для збільшення швидкості внутрішньосистемних з'єднань, в деяких архітектурах ПКВМ використовуються більш довгі трасувальні з'єднання між логічними блоками.

У місці перетину вертикальних і горизонтальних каналів створюються перемикаючі блоки. При такій архітектурі для кожного провідника, що входить у перемикальний блок, існують три програмованих перемикача, які дозволяють йому підключатися до трьох інших провідників в суміжних сегментах каналу. Модель чи топологія вимикачів, використовувана в цій архітектурі, є планарною або доменною топологією перемикальних блоків. У цій топології провідник траси номер один підключається тільки до провідника траси номер один у суміжних каналах, провідник траси номер 2 підключається тільки до провідників траси номер 2 і так далі.

Сучасні сімейства ПКВМ розширюють перераховані вище можливості і включають вбудовану функціональність високого рівня. Маючи ці загальні функції, вбудовані в кремній, можна скоротити площу кристала, до того ж ці функції будуть працювати швидше, ніж якщо їх створювати на базі примітивів. До таких засобів належать мультиплексори, блоки цифрової обробки сигналів, вбудовані процесори, швидка логіка вводу-виводу і вбудована пам'ять.

ПКВМ також широко застосовуються для систем перевірки придатності, у тому числі в докремнієвій і післякремнієвій перевірці придатності, а також при розробці вбудованих програм. Це дозволяє компаніям-виробникам інтегральних схем перевіряти працездатність своїх пристроїв до виготовлення їх на заводі, скорочуючи час виходу виробу на ринок.

Загальну структуру кристала FPGA-мікросхеми наведено на рис. 1.2. По периферії верхнього шару кристала розміщуються блоки вводу/виводу (БВВ), що можуть бути запрограмованими для виконання функцій буферів: вхідного, вихідного, із запам'ятовуванням стану та ін. У деяких серіях FPGA-ІМС рівень напруги на двох БВВ може відрізнитись, що дає змогу поєднувати різні за рівнем живлення інтерфейси.

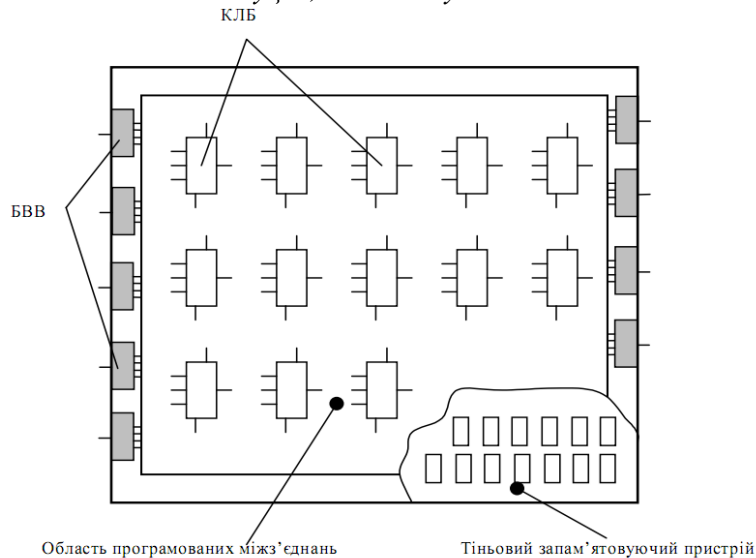


Рис. 1. – Узагальнена структура FPGA-мікросхем

У центрі кристала у вигляді матриці розміщено конфігуровані логічні блоки (КЛБ). Швидкодія мікросхем визначається часою затримкою “вхід-вихід” одного КЛБ. Структура КЛБ залежить від серії мікросхем, так, наприклад, в ІМС XC2000 кожен КЛБ має один елемент пам’яті (тригер), 2 виходи, 4 входи загального призначення та спеціальний вхід синхронізації (тактовий вхід). КЛБ ІМС цієї серії може генерувати будь-яку логічну функцію чотирьох змінних або дві логічні функції трьох змінних. Змінні для логічних функцій можуть надходити з чотирьох входів та виходу елемента пам’яті.

Область між конфігурованими логічними блоками називається областю програмованих міжз’єднань і являє собою розвинену ієрархію металічних ліній зв’язку, в місцях перетину яких розміщено спеціальні швидкодіючі транзистори. Функція області міжз’єднань полягає в забезпеченні зв’язку між будь-якими виводами КЛБ та БВВ.

Необхідний маршрут міжблокових з’єднань у FPGA-ПЛИС реалізується комутацією відповідних ліній за допомогою транзисторів. Нижній шар кристала займає тіньовий запам’ятовуючий пристрій, інформація в елементах якого і визначає логічні функції КЛБ, конфігурацію БВВ та маршрути міжз’єднань.

У таблиці 1.1 наведено основні характеристики мікросхем FPGA виробництва Xilinx Inc., зокрема, однієї з останніх розробок серії Virtex та масової серії Spartan.

Таблиця 1. – Основні характеристики мікросхем FPGA

Характеристика	Серія Virtex			Серія Spartan	
	XCV50	CV1000	XCS40/XL	XCS20/XL	XCS05/XL
Системна частота (MHz)	200	200	80	80	80
Технологія (мкм)	0.22	0.22	0.35/0.5	0.35/0.5	0.35/0.5
Напруга живлення ядра (В)	2.5	2.5	3.3/5.5	3.3/5.5	3.3/5.5
Кількість системних вентилів (шт.)	57906	112402	13К-40К	7К-20К	2К-5К
Кількість логічних комірок (шт.)	1728	27648	1862	950	238
Максимальне число портів вводу/виводу (шт.)	180	514	205	160	77

Широкий діапазон ІМС FPGA-технології дозволяє проектувати на їх основі широкий спектр електронних пристроїв, серед яких: засоби поєднання різних за живленням інтерфейсів, перетворювачі кодів, периферійні контролери, мікропрограмні пристрої керування, скінченні автомати, універсальні та спеціалізовані процесори, пристрої цифрової обробки сигналів тощо.

Система на кристалі Amber ARM. Ядро процесора Amber з відкритим вихідним кодом є ARM-сумісним 32-бітним RISC-процесором. Саме ядро, вихідний код, знаходиться на сайті [4,5] і є частиною бібліотек з відкритим вихідним кодом. Ядро Amber повністю сумісне з набором команд ARMv2 і підтримується набором інструментів GNU. З даним набором команд сумісне також вільнорозповсюджене ядро STORM Open Soft Core [4]. Ця версія набору інструкцій ARM підтримується вільнорозповсюджуваними компіляторами, оскільки вона не захищена патентами і може бути реалізованою без ліцензії ARM Holdings, на відміну від деяких проектів з відкритим вихідним кодом. Проект Amber є повністю вбудованою в FPGA системою, що включає Amber ядро і деякі периферійні пристрої, наприклад інтерфейси UARTs, таймери, Ethernet MAC.

Є дві версії ядра, передбачені в проекті Amber. Amber 23 має 3-ступінчастий конвеєр, єдиний кеш команд і даних, містить Wishbone інтерфейс і має продуктивність до 0,75 DMIPS на МГц. Версія ядра Amber 25 має 5-ступінчастий конвеєр, окремі кеші для даних та інструкцій, містить Wishbone інтерфейс і має продуктивність 1,0 DMIPS на МГц. Обидва ядра реалізовані з однаковим інтерфейсом і на 100% програмно сумісні між собою.

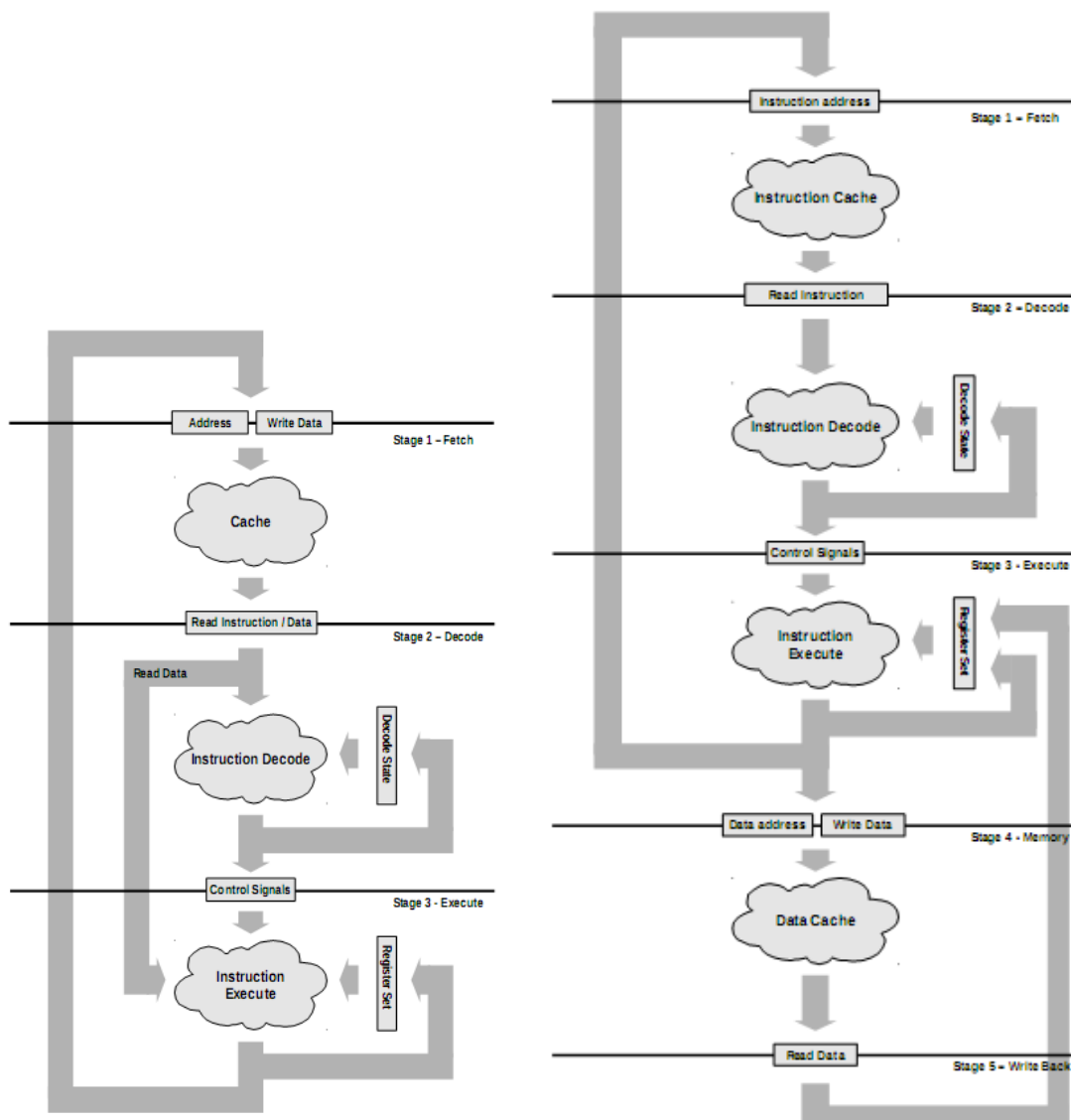


Рис. 2. – Етапи роботи ядер Amber 23 та Amber 25

Amber 23 - дуже маленьке за обсягом 32-бітне ядро, що забезпечує хорошу продуктивність. Інструкції на основі регістрів виконуються за один цикл (такт роботи), винятком є інструкції пов'язані із операціями множення. Команди завантаження і збереження вимагають три цикли. Робота ядра заходить в тупикову ситуація коли відбувається промах кеша, або коли ядро виконує одночасно читання/доступ по шині.

Ядро Amber 25 забезпечує від 30 до 40% кращу продуктивність, ніж ядро Amber 23, але також затрачає на це від 30 до 40% більше енергії і ресурсів мікросхеми. Інструкції на основі регістрів виконуються за один цикл, винятком є інструкції, пов'язані із операціями множення або складними операціями зсуву. Команди завантаження і збереження також виконуються за один цикл, якщо немає конфлікту із вибіркою наступної команди. Робота ядра заходить в тупикову ситуація коли відбувається промах в будь-якому із двох кешів, або коли ядро виконує одночасно читання/доступ по шині.

Портування ядра на кристал Altera Cyclone III. Обидва ядра були перевірені шляхом завантаження Linux 2.4. Версії ядра Linux 2.4 та більш ранні містять конфігурації підтримувані шиною ISA. У версіях ядра Linux 2.6 і пізніших явно не підтримуються ARMv2A ISA і тому потрібно кілька модифікацій вбудованого Amber ядра для запуску. Ядра не містять блок управління пам'яттю (MMU), тому вони можуть працювати тільки в невіртуальному варіанті пам'яті Linux та µClinux. Дані ядра були розроблені у версії Verilog 2001 року і оптимізовані для FPGA синтезу.

Вихідні коди модулів були розроблені для мікросхем Spartan/Virtex фірми Xilinx, що можна бачити на прикладі:

```
// Tag RAMs
`ifdef XILINX_SPARTAN6_FPGA
xs6_sram_256x21_line_en
`endif
`ifdef XILINX_VIRTEX6_FPGA
xv6_sram_256x21_line_en
`endif
`ifndef XILINX_FPGA
generic_sram_line_en
`endif
```

В мікросхемах FPGA є вбудовані блоки пам'яті. Для них існують спеціальні компоненти, в нашому випадку для портування коду на мікросхеми фірми Altera необхідно використати altsyncram. На його основі, задаючи параметри можливо створити модулі пам'яті різної разрядності та об'єму.

Висновки. Здійснено реалізацію системи на кристалі в програмованій логічній інтегральній схемі Altera Cyclone III EP3C10E144C8. Проект побудовано на основі відкритого проекту Amber ARM. В системі реалізовано процесор ARM v2a, послідовний порт, таймер, контролер переривань, контролер пам'яті SDRAM.

ЛІТЕРАТУРА

1. Amber Open Source Project. Amber 2 Core Specification. May 2013.
2. Amber Open Source Project. Amber Project User Guide. May 2013.
3. Donald E.Thomas, Philip R.Moorby. The Verilog Hardware Description Language, 5th edition. - Kluwer Academic Publishers. - 2002. – 404p.
4. <http://opencores.org>.
5. <http://opencores.org/project,amber>.
6. <http://uk.wikipedia.org/wiki/ARM>.
7. <http://www.altera.com>.
8. <http://www.marsohod.org/index.php/prodmarsohod2/amber-arm-soc>.
9. <http://www.xilinx.com>.
10. Verilog. Практика проектування цифрових пристроїв на ПЛІС: Навчальний посібник / Рябенський В.М. Ушкарєнко О.О. – Миколаїв: Вид-во "Іліон", 2007. – 324 с.