УДК 681.324

О.П. КУРГАЄВ, І.В. САВЧЕНКО

Інститут кібернетики ім. В.М. Глушкова НАНУ, Київ

ПРОЕКТУВАННЯ УПРАВЛЯЮЧОЇ ЧАСТИНИ IP-БЛОКІВ НА ПЛІС

Запропоновано визначення процесу проектування управляючої частини IP-блоків в середовищі САПР та методика розробки специфікації пристрою управління, який реалізований у вигляді цифрового скінченого автомата (ЦСА). Розглянуто методи кодування ЦСА. Синтез специфікації ЦСА в структуру логічних елементів виконано за допомогою інструментального засобу Synplify Pro®. Основні результати роботи використано для побудови пристрою управління процесора баз знань.

Ключові слова: *hdl-модель, діаграма станів, пристрій обробки даних, пристрій управління, специфікація, файл обмежень.*

Вступ

Задача проектування систем на програмованих логічних інтегральних схемах (ПЛІС) на всіх етапах включає в себе комбінування використання системи автоматизованого проектування (САПР) та методології проектування складних систем [1]. Процес проектування систем на ПЛІС можна поділити на чотири етапи: 1) розробка архітектури на системному рівні; 2) залежно від архітектури підбір апаратних, мікропрограмних та програмних блоків (IP-блоків); 3) проектування необхідних IP-блоків; 4) інтеграція всіх блоків шляхом проектування і верифікації архітектури на системному рівні та організації інтерфейсу поміж її складовими.

Більшість IP-блоків можна умовно поділити на дві частини: пристрій обробки даних (ПОД) та пристрій управління (ПУ) [2]. ПУ звичайно реалізують за формою цифрового скінченного автомата (ЦСА).

ЦСА широко застосовуються в: пристроях ділення, конвертування даних з binnary в BSD формат, лічильниках періоду [3], контролерах послідовного зв'язку UART [4], контролерах шини AMBA [5], RISC мікропроцесорах [6] та ін.

Для розробки ЦСА на ПЛІС достатньо лише розробити специфікацію ЦСА та виконати її кодування [7]. Наступні процедури синтезу, оптимізації та розміщення ПУ на кристалі автоматично виконує САПР. Вирішення задачі проектування ЦСА на ПЛІС присвячено роботи [2, 3, 7]. Однак, в цих роботах не достатньо повно висвітлено процес розробки специфікації у вигляді діаграми станів та методи кодування ЦСА.

Задача статті – визначити процес проектування ПУ в САПР, запропонувати методику процесу розробки специфікації ЦСА у вигляді діаграми станів, розглянути методи кодування ЦСА.

1. Вибір мови специфікації ПУ

Проектування ПУ починається з визначення специфікації ПУ у вигляді текстового опису або алгоритму функціонування.

В процесі створення специфікації ПУ (СПУ) для САПР виникає необхідність представлення її у вигляді, що забезпечить її «легке» читання та розуміння. Це стає можливим завдяки вибору мови специфікації. Крім того, мова специфікації забезпечує спрощений перехід до наступного етапу проектування. До найбільш розповсюджених мов специфікацій можна віднести: функціональні схеми (ФС), граф-схеми алгоритмів (ГСА) та діаграми станів або графи переходів (ГП) [8].

ГП в порівнянні з ФС та ГСА: надають більш компактніше та наочніше представлення поведінки автомата; забезпечують можливість розгляду тільки існуючих станів автомата; дають можливість за формальним ознаками визначити повноту та несуперечність специфікації; в ГП кожна дуга переходу автомата позначається булевою формулою тільки тих змінних, які забезпечують даний перехід; дає можливість визначити всі значення вихідних сигналів в кожному стані; забезпечують легкість переходу до опису поведінки автомата на мові високого рівня.

2. Побудова графу переходів ПУ

1. На основі заданої специфікації формується множина станів ПУ та зображується на площині у вигляді вершин графу. Для більшої наочності біля кожної вершини графа позначається назва стану.

2. Біля кожної вершини графу визначається кортеж значень всіх вихідних сигналів.

3. Визначаються всі допустимі переходи поміж вершинами графу, які позначаються відповідними

орієнтованими дугами. Стійкий стан позначають інцидентною петлею.

4. Кожна дуга графа позначається булевою формулою на підмножині вхідних сигналів ПУ.

5. Для кожної вершини графу забезпечується несуперечність переходів з однієї вершини в іншу, завдяки визначенню пріоритетів переходів.

6. Для забезпечення повноти переходів можуть бути введені додаткові переходи поміж вершинами. Такі переходи також позначаються відповідними булевими функціями.

7. Якщо в побудованому графі присутні генеруючі контури (перехід без умови), то вони усуваються.

8. Якщо деякі вершини графа мають однакові кортежі вихідних сигналів, то вони можуть бути суміщені.

9. Якщо граф семантично коректний, то діаграму станів побудовано.

3. Розробка схеми зв'язків ПУ – ПОД

Як вже зазначалося вище, структура цифрової синхронної системи складається з двох пристроїв: ПУ та ПОД (рис. 1).

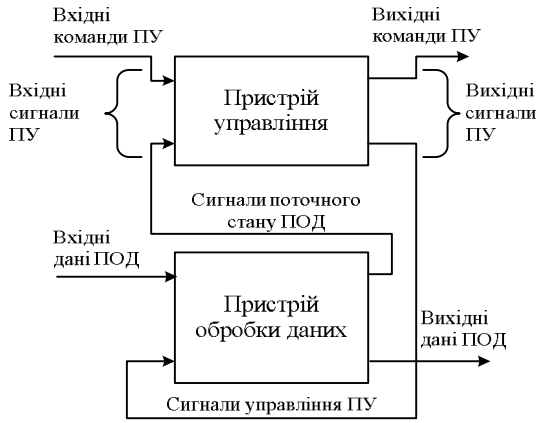

Рис. 1. Схема зв'язку «ПУ – ПОД»

ПУ представляє ЦСА з вхідними (вхідні команди ПУ та сигнали поточного стану ПОД) та вихідними сигналами (вихідні команди, сигнали управління). Вхідні сигнали визначають команду, яку повинен виконати ПУ; вихідні сигнали керують роботою ПОД та інших підсистем.

ПОД виконує операції над даними та складається з: 1) функціональних блоків; 2) регістрів, елементів пам'яті для тимчасового зберігання даних; 3) шин для обміну даними поміж внутрішніми блоками ПОД. Зовнішні дані можуть бути отримані через шину вхідних даних. Результати обробки даних передаються по шині вихідних даних.

Таким чином, виходячи із змістовної специфікації ПУ, формуються множини вхідних і вихідних сигналів ПУ та їх власні підмножини: сигналів поточного стану ПОД та сигналів управління ПУ відповідно. Дві останні підмножини повністю визначають структуру взаємозв'язків між ПУ та ПОД (чи ПУ – ПОД).

4. Вибір структурної моделі ПУ

На практиці широко застосовуються автомати двох типів: Мілі та Мура [3]. Автомати Мура формують дещо інший клас моделей в порівнянні з автоматами Мілі, але для кожного автомата Мура може бути побудований еквівалентний автомат Мілі [9]. Відносно ж виразності вони еквівалентні – їх виразні можливості обмежені регулярними мовами.

Узагальнену структуру ЦСА з виходами по типу автоматів Мілі та Мура представлено на рис. 2. Пам'ять станів складається з набору тригерів (або блоку пам'яті із регістром адресу), в яких зберігається поточний стан автомата. Пам'ять станів підключено до тактового сигналу, тому перехід в наступний стан відбувається з кожним новим тактом. Логіка переходів визначає наступний стан автомата та є функцією від вхідної команди та сигналу поточного стану. Вихідний сигнал визначається за допомогою вихідної логіки та є функцією від вхідної команди ПУ та сигналу поточного стану (автомат Мілі) чи тільки від поточного стану (автомат Мура).

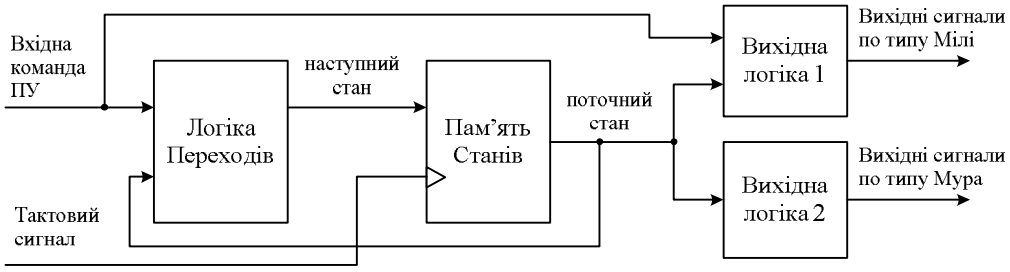

Рис. 2. Структура ЦСА

5. Реалізація HDL-моделі ПУ

HDL-модель розробляється у вигляді текстового файлу та складається з: декларативної та архітектурної частини. В декларативній частині описуються параметри входів та виходів ПУ, це перш за все специфікація інтерфейсу пристрою. В архітектурній частині визначаються функції ПУ, особливості формування вихідних сигналів на основі вхідних сигналів та поточного стану ПУ.

Для викладення специфікації ПУ на мові VHDL використовують:

1) *Поведінковий* опис. Внутрішня структура ПУ не специфікується в явній формі, задаються функціональний опис поведінки та особливості формування вихідних сигналів ПУ залежно від його вхідних сигналів та станів.

2) *Структурний* опис. Описується в явній формі внутрішня структура ПУ, як множина функціональних блоків та зв'язків між ними. Поведінка ПУ, особливості формування вихідних сигналів при надходженні вхідних сигналів визначаються неявно – складом та зв'язками поміж блоками, що входять до складу ПУ.

3) *Структурно-поведінковий* опис. Застосовується комбінація першого та другого методів [1], переважно у випадках мов, вище регулярного рівня.

6. Розробка файлу обмежень

В файлі обмежень (constraint file) за допомогою Tcl команд вказуються додаткові параметри для засобів автоматичного синтезу: часові затримки, загальні атрибути [10]. При проектуванні скінченних автоматів, в файлі обмежень слід вказати параметри атрибута syn_encoding (тип кодування автомата). Параметри атрибута syn_encoding представлено в табл. 1. [10].

Таблиця 1

Параметри атрибута syn_encoding

Параметр syn_encoding	Особливості застосування	Пояснення
sequential	Мінімальна кількість апаратних ресурсів, кількість станів < 5	Більш ніж один з бітів регістра стану може змінюватись та містити «1»
onehot	Максимальна швидкодія, застосовується велика кількість тригерів, кількість станів 5-40	Тільки два біти регістра стану можуть змінюватись (один переходить в «0», інший в «1»), та не більш одного може містити «1»
gray	Великий декодер на виході, кількість станів > 40	Тільки один біт регістра стану може змінюватись та більш ніж один містити «1»
safe	Скидання автомата при переході в неіснуючий стан	Ускладнюється реалізація, може використовуватись одночасно з: sequential, onehot, gray

7. Компіляція файлів та верифікація проекту

Завершальним етапом проектування є: перевірка синтаксису файлів проекту, їх компіляція, генерування логічної структури (ЛС) ПУ та верифікація. Такі процедури можна реалізувати за допомогою інструментальних засобів САПР: Synplify Pro® та ModelSim®. Synplify Pro®, що міститься в комплекті САПР Libero IDE фірми Actel, отримуючи текстовий опис цифрової системи або її окремих елементів на мові високого рівня VHDL або Verilog, здатний згенерувати ЛС у вигляді файлу, що містить списки всіх логічних елементів схеми та таблицю їх з'єднання [10].

ModelSim® – це засіб верифікації проектів, реалізованих на мовах HDL. Для верифікації потрібно додатково в графічному редакторі ModelSim® розробити модель тестових сигналів (testbench) для існуючого дизайну на мові HDL та виконати процес верифікації [11].

8. Приклад проектування ПУ

Розглянемо детальніше основні етапи процесу проектування для прикладу, в якості якого приймемо ПУ, що керує процесом пошуку запису в таблиці записів (ТЗ), скінченного, але наперед не визначеного розміру. ТЗ містить записи $R_1, R_2, R_3, ..., R_N$ із ключами $K_1, K_2, K_3, ..., K_N$ відповідно, де $N \geq 1$. ПУ використовує послідовний алгоритм пошуку запису із заданим ключем K. Пошук починається з перевірки ключа першого запису K_1 і триває доти, поки не буде знайдено запис із ключем K_i, де $K_i = K$.

8.1. Побудова графу переходів ПУ

Нехай специфікація ПУ задана алгоритмом:

1. На вхід «SEARCH» ззовні поступає активний сигнал, що є ознакою початку роботи алгоритму.

2. ПУ генерує одиничний активний імпульс на виході «CLR_RD».

3. Якщо на вхід «FULL» поступає активний сигнал від ПОД, то ПУ генерує одиничний активний імпульс на виході «N_FOUND» та переходить до п. 7. Інакше ПУ переходить до п. 4.

4. Якщо на вхід «EMPTY» поступає активний сигнал від ПОД, то ПУ генерує одиничний активний імпульс на виході «N_FOUND» та переходить до п. 7. Інакше ПУ генерує одиничний активний імпульс на виході «RD».

5. ПУ генерує одиничний активний імпульс на виході «CMP».

6. Якщо на вхід «RESULT» поступає активний сигнал від ПОД, то ПУ генерує одиничний активний імпульс на виході «FOUND» та переходить до п. 7. В іншому випадку перехід до п. 4.

7. Робота ПУ завершена.

На основі цієї специфікації побудовано варіант графу переходів ПУ з виходами по типу автомата Мура (рис. 3).

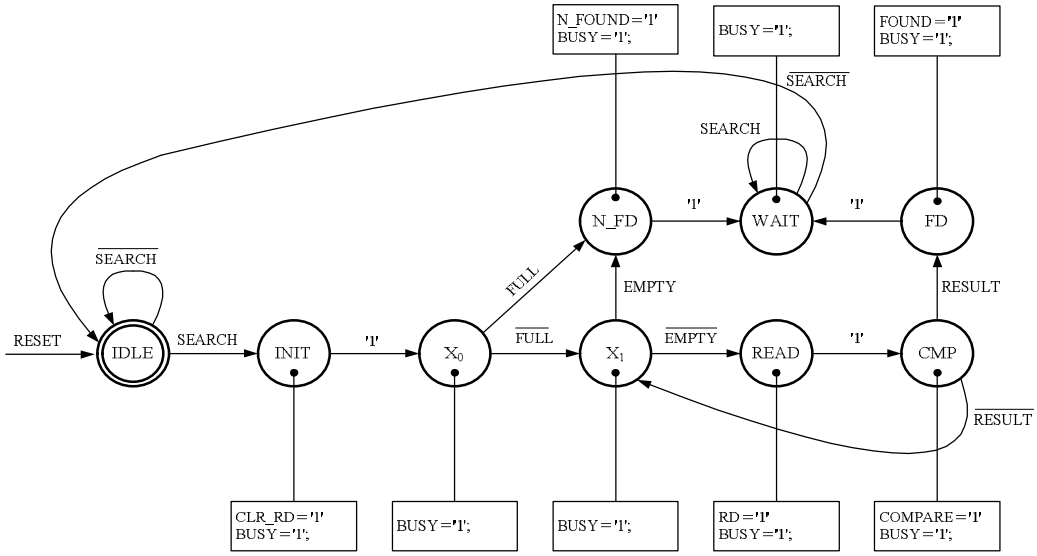

Рис. 3. Граф переходів ПУ

В будь-який момент ПУ може перебувати в одному з дев'яти станів: IDLE, INIT, X_0, X_1, N_FD, READ, WAIT, CMP, FD. Стан IDLE – одночасно початковий і кінцевий. Наявність вихідних сигналів (CLR_RD, RD, CMP, FOUND, N_FOUND, BUSY) залежить від поточного стану автомата. На рис. 3 навпроти кожного із станів автомата в рамках зображено тільки активні вихідні сигнали. Вихідні сигнали, які відсутні в рамках, приймають неактивне значення. Граф спроектовано таким чином, що при наявності сигналу RESET (низький рівень) автомат переходить в стан IDLE.

8.2. Розробка схеми зв'язків ПУ – ПОД

Схему ПУ – ПОД представлено на рис. 4.

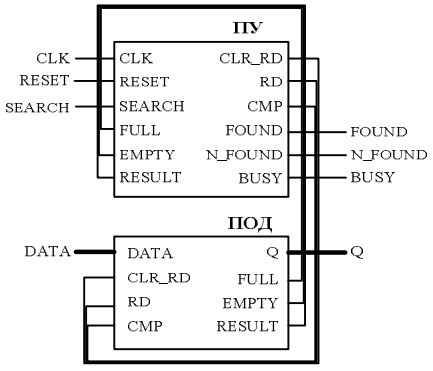

Рис. 4. Схема зв'язків ПУ – ПОД

На вхід ПУ надходять вхідна команда SEARCH та сигнали поточного стану ПОД (FULL, EMPTY, RESULT). З виходів ПУ надходять: сигнали управління (CLR_RD, RD, CMP) на входи ПОД; вихідні команди управління (FOUND, N_FOUND, BUSY) надходять на інші підсистеми. Сигнали CLK, RESET є тактовим сигналом та сигналом скидання відповідно. Розглянемо вищевказані сигнали детальніше:

SEARCH – зовнішній командний сигнал, за яким починається робота ПУ;

FULL – сигнал, за допомогою якого ПУ перевіряє наявність порожніх записів в ТЗ. FULL = «1», якщо в ТЗ не міститься жодного порожнього запису;

EMPTY – дорівнює «1», якщо в ТЗ не міститься жодного заповненого запису або в результаті пошуку були прочитані всі записи;

RESULT – дорівнює «1», якщо запис із заданим ключем знайдений;

CLR_RD – команда на підготовку до зчитування першого запису;

RD – команда зчитати запис та підготуватися до зчитування наступного;

CMP – ініціює процес порівняння ключів K_i та K ;

FOUND – встановлюється в «1», якщо шуканий запис знайдено;

N_FOUND – встановлюється в «1», якщо шуканий запис не знайдено;

BUSY – встановлюється в «1» під час пошуку запису.

DATA – шина вхідних даних ПОД.

Q – шина вихідних даних ПОД.

8.3. Розробка файлу обмежень

ПУ має 13 станів (див. HDL-модель ПУ), тому для атрибута syn_encoding вибираємо параметр one-hot (оскільки 13 знаходиться в інтервалі [5, 40].

8.4. Компіляція файлів та верифікація проекту

Синтез проекту ПУ виконано в середовищі Synplify Pro®. Проект реалізовано для FPGA M1AGL600V2 фірми Actel. Всього використано 27 логічних комірок (ЛК). Також додатково схема містить 6 вхідних та 6 вихідних буферів. Максимальна тактова частота схеми F_{MAX}=82,9 МГц.

Висновки

В роботі запропоновано: визначення процесу проектування управляючої частини IP-блоків в середовищі САПР; запропоновано методику процесу розробки специфікації ЦСА у вигляді діаграми станів; розглянуто методи кодування ЦСА. Синтез функціональної специфікації ПУ в структуру логічних елементів виконано за допомогою інструментального засобу Synplify Pro®.

Основні результати роботи було використано для побудови ЦСА процесора баз знань, принцип роботи якого висвітлено в [12].

Література

1. Суворова Е.А. Проектирование цифровых систем на VHDL. / Е.А. Суворова, Ю.Е. Шейнин – СПб.: БХВ-Петербург, 2003. – 576 с.

2. Уэйкерли, Проектирование цифровых устройств: Пер. с англ. / Уэйкерли, Джон Ф. – М.: Постмаркет, 2002. – 923 с.

3. Pong P. Chu. FPGA prototyping by VHDL examples. Copyright © 2008 by John Wiley & Sons, Inc. – P. 440.

4. CoreUART Handbook v2.0. © 2009 Actel Corporation. P. 28. Avaliable at http://www.actel.com

5. AMBA™ Specification. Rev 2.0. © Copyright ARM Limited 1999. – P. 230. Avaliable at http://www.arm.com/

6. Enoch O. Hwang. Digital logic and Microprocessor design with VHDL. Copyright © Brooks / Cole 2005. – P. 513.

7. Stephen Brown, Zvonko Vranesic. Fundamentals of digital logic with VHDL, Second edition. Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2005. – P. 939.

8. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. / А.А. Шалыто – СПб.: Наука. 1998. – 628 с.

9. Брауэр В. Введение в теорию конечных автоматов: Пер. с нем. / В. Брауэр – М.: Радио и связь, 1978. – 392 с.

10. Synopsys FPGA Synthesis User Guide. Copyright © 2009 Synopsys, Inc. P. 412. Avaliable at http://www.actel.com.

11. ModelSim® User's Manual. Copyright © 1991-2008 Mentor Graphics Corporation. – P. 516. Avaliable at http://www.actel.com

12. Кургаев А.Ф. Проблемная ориентация архитектуры компьютерных систем / А.Ф. Кургаев. – К.: Сталь, 2008. – 540 с.

Надійшла в редакцію 15.02.2010

Рецензент: д.т.н., чл.-кор. НАН України, зав. відділом ІК НАН України В.П. Боюн. Інститут кібернетики імені В.М. Глушкова НАНУ, Київ.

ПРОЕКТИРОВАНИЕ УПРАВЛЯЮЩЕЙ ЧАСТИ IP-БЛОКОВ НА ПЛИС

А.Ф. Кургаев, И.В. Савченко

Предлагается определение процесса проектирования управляющей части IP-блоков в среде САПР и методика разработки спецификации устройства управления, реализованного в виде цифрового конечного автомата (ЦКА). Рассмотрены методы кодирования ЦКА. Синтез спецификации ЦКА в структуру логических элементов выполнен с помощью инструментального средства Synplify Pro®. Основные результаты работы использованы при построении устройства управления процессора базы знаний.

Ключевые слова: hdl-модель, диаграмма состояний, устройство обработки данных, устройство управления, спецификация, файл ограничений.

CONTROL UNIT PLANNING OF IP-BLOCKS ON EPLD

A.F. Kurgaev, I.V. Savchenko

Suggested in the paper are definition of designing process of a control unit of IP-blocks in the environment of CAD and a technique of working out of the specification of the control unit that is realized in the form of the digital finite state machine (FSM). The methods of code of FSM are considered. A synthesis of CU specification to a logical elements structure is executed by the Synplify Pro® tool. Main work performances were used for construction of knowledge bases processor control unit.

Key words: hdl-model, state diagram, data unit, control unit, carbon, specification, constraint file.

Кургаєв Олександр Пилипович – д.т.н., провідний науковий співробітник відділу мікропроцесорної техніки, Інститут кібернетики імені В.М. Глушкова НАНУ, Київ, e-mail: afkurgaev@ukr.net

Савченко Іван Васильович – аспірант, Інститут кібернетики імені В.М. Глушкова НАНУ, Київ, e-mail: savchenko_ivan@ukr.net