УДК 004.032.2

В.С. Глухов, О.В. Глухова
Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин

ПРОЕКТУВАННЯ ЗАСОБІВ КЕРУВАННЯ ПРОГРАМНИМИ СТЕНДАМИ ДЛЯ ВІДЛАГОДЖЕННЯ ФУНКЦІОНАЛЬНИХ ВУЗЛІВ ПЛІС

Описано технологію розроблення засобів керування програмними стендами для відлагодження функціональних вузлів на ПЛІС. Програмні стенди є описаними мовами описів апаратних засобів моделями фізичних стендів, на яких перевіряють моделі функціональних цифрових вузлів. Пропоновані засоби призначені для перетворення поданих у табличному вигляді тестових впливів, що повинні подаватися з боку стенда на досліджуваний об'єкт, на вирази цих впливів на мові опису апаратних засобів. Запропоновані засоби входять до складу системи проектування ПЛІС.

Ключові слова: програмний стенд, ПЛІС, мова опису апаратних засобів, САПР.

The article describes FPGA functional units' testbench control software development technology. Testbenches are hardware languages described natural stands models which used for digital component functional models tests. Proposed tools designed to convert presented in tabular form test influences to the ones hardware description language form that must be submitted on the object. Proposed tools are part of the FPGA system design tools.

Key words: testbench, FPGA, HDL, CAD.

Вступ

Останнім часом для створення цифрових систем керування, систем збору та накопичення наукової інформації широко використовуються мікросхеми програмованих логічних матриць. Під час створення ПЛІС для опису цифрових схем, що містяться всередині ПЛІС (ядер, ядро – Core), використовуються спеціалізовані мови описів апаратних засобів, зокрема мова VHDL. Як правило, ядра мають стандартний інтерфейс для під'єднання їх до стандартних мікропроцесорів або мікроконтролерів. Для перевіряння правильності створених VHDL-описів ядер здійснюється їхнє моделювання. Для подачі тестових впливів на модель об'єкта тестування (ядро) та аналізу результатів його реакції на ці впливи додатково (також у вигляді опису на мові VHDL) створюється програмний стенд (testbench). Тестові впливи та еталонні результати реакції об'єкта входять як складові до опису програмного стенда. Тестування проводить розробник об'єкта, який чудово розбирається в VHDL-описах об'єкта тестування, але не має можливості приділяти велику увагу і багато часу розробленню складного програмного стенда, тобто описати мовою VHDL велику послідовність тестових наборів, яка точно відтворює реальну послідовність вхідних сигналів. Розробник може викласти у вигляді простої таблиці послідовність потрібних йому тестових впливів. Метою цієї роботи є створення елемента системи автоматизованого проектування ПЛІС, який перетворює табличний вигляд тестових впливів на VHDL-описи програмного стенда.

1. Окреслення проблеми

Для перевіряння правильності створених VHDL-описів цифрових схем (ядер) здійснюється їхнє моделювання. Проблемою є створення VHDL-опису великої послідовності вхідних тестових наборів, яка точно відтворює реальну послідовність вхідних сигналів. Мета цієї роботи – створення елемента системи автоматизованого проектування ПЛІС, який перетворює табличний вигляд тестових наборів на VHDL-описи програмного стенда для перевіряння ядер ПЛІС і самої ПЛІС.

2. Аналіз останніх досліджень та публікацій

Мікросхеми програмованих логічних матриць (ПЛІС) широко використовуються для створення цифрових систем керування, систем збору та накопичення наукової інформації (СЗНІ)

[1]. Особливості сучасних ПЛІС наведено у роботах [2, 3]. Для опису цифрових схем, що містяться всередині ПЛІС (ядер), використовуються спеціалізовані мови описів апаратних засобів, зокрема мова VHDL [4]. Як правило, ядра ПЛІС мають стандартні паралельні інтерфейси для під'єднання їх до стандартних мікропроцесорів або мікроконтролерів, зокрема побудованих на основі процесорів ARM з 16- [5] і 32-бітними шинами [6], і стандартні послідовні інтерфейси, такі як CAN [7] та SciWay [8], для під'єднання до віддалених периферійних пристроїв та побудови спеціалізованих комп'ютерних мереж. Для перевіряння правильності створених VHDL-описів ядер здійснюється їхнє моделювання. Для подачі тестових впливів на модель об'єкта тестування (ядра) та аналізу результатів його реакції на ці впливи додатково (також у вигляді опису на мові VHDL) створюється програмний стенд (testbench [9]).

3. Цілі статті

Метою цієї роботи є створення елемента системи автоматизованого проектування ПЛІС, який перетворює табличний вигляд тестових впливів на VHDL-описи програмного стенда.

4. Основи тестування цифрових схем

Загальну схему тестування ядер ПЛІС подано на рис. 1, де показано ядро (об'єкт тестування) і програмний стенд (testbench). До складу останнього входять генератор тестових послідовностей, еталон для порівняння з результатами роботи ядра і вузол порівняння (аналізатор результатів роботи ядра). Усі складові описано мовою VHDL. Особливістю програмного стенда є зворотний зв'язок, який дає змогу змінювати тестову послідовність залежно від результатів роботи ядра.

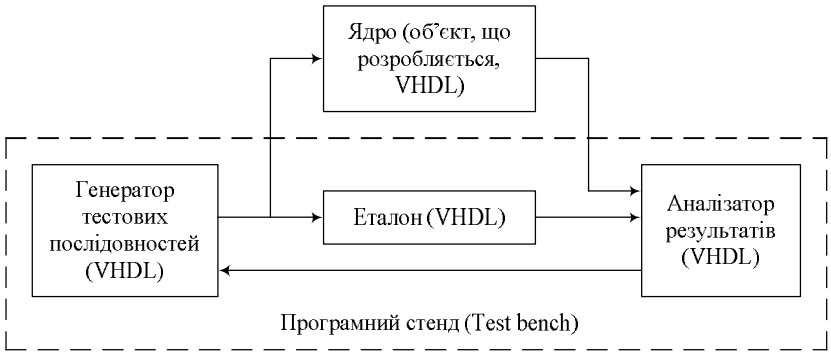

Рис. 1. Загальна схема тестування

Якщо ядро має вихід на стандартний інтерфейс, під час його тестування за схемою рис. 1 виникає задача опису мовою VHDL послідовності обміну даними цією шиною з урахуванням усіх її сигналів і часових параметрів. Одним з варіантів розв'язання цієї задачі є введення до програмного стенда моделі мікропроцесора (мікроконтролера), який керує шиною (рис. 2). За такою схемою мікроконтролер бере на себе функції генерації тестових послідовностей і аналізу результатів роботи тестованого ядра. При цьому не треба описувати послідовності обміну даними шиною, але виникає задача створення моделі (ядра) мікроконтролера (яку, своєю чергою, спочатку також треба буде протестувати).

Також треба взяти до уваги існування великої кількості тестових послідовностей, зафіксованих у табличному вигляді, створених і відпрацьованих ще до впровадження VHDL-описів, а також існування готових VHDL-описів різноманітних циклів шин мікропроцесорів.

Для розв'язання задачі пропонується використовувати в складі програмного стенда спрощену модель мікроконтролера, яка описує тільки його інтерфейсну частину (тільки обмін даними шиною), але не містить описів його внутрішньої структури. Послідовність циклів обмінів даними задається у табличному вигляді. Пропоноване рішення полягає у перетворенні табличного опису послідовності обміну даними на VHDL-опис інтерфейсної частини мікроконтролера, який використовується у складі програмного стенда (рис. 3). Для забезпечення цього рішення потрібно створити відповідний транслятор.

Перевагою запропонованого рішення є можливість використання наявних тестових послідовностей, поданих у табличному вигляді.

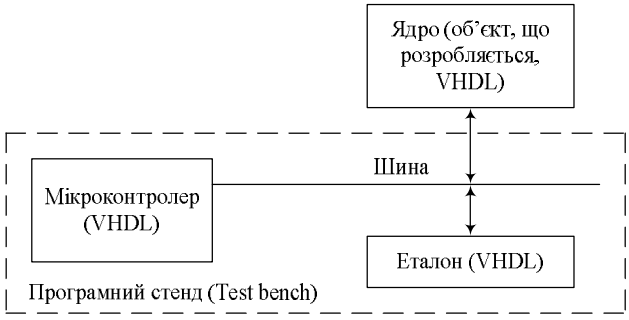

Рис. 2. Програмний стенд з мікроконтролером

5. Транслятор описів послідовностей циклів обмінів шиною

Для створення транслятора необхідно проаналізувати можливі типи обмінів шиною. За основу для аналізу взято мікроконтролери на основі ARM з 16- [5] та 32-бітними [6] шинами.

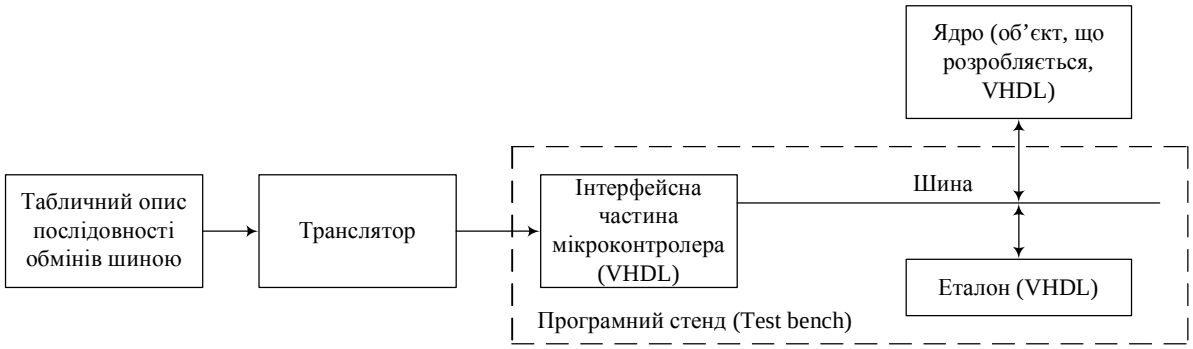

Рис. 3. Пропонована схема тестування ядер

За результатами аналізу варіантів обміну 16- та 32-бітними шинами процесорів ARM виділено декілька циклів обміну, особливості яких зведено у табл. 1 та табл. 2. Умовне позначення циклу складається з трьох частин: <тип циклу> <адреса> [<дані для запису>]. У циклах читання даних для запису немає.

Для кожного табличного рядка формується VHDL-опис виклику відповідної процедури циклу шини відповідно до табл. 1, де показано змінну частину опису виклику процедури, і табл. 2, де наведено постійну частину цього опису (AAAA, BB, DDDD, DDDDDDDD – довільні значення старших і молодших розрядів адреси та шини даних).

Таблиця 1

Описи циклів шини

Типи циклів шини				Змінна частина VHDL-опису виклику процедури циклу шини				
Позначка циклу	розрядність шини	швидкий або повільний обмін	запис або читання	команда	старші біти адреси	молодші біти адреси	дані для запису	приймач результатів читання
w	16	швидкий	запис	REGWRITE16	(X"AAAA",	X"BB",	X"DDDD",	
r	16	швидкий	читання	REGread16	(X"AAAA",	X"BB",		REG_RD,
v	16	повільний	запис	REGWRITE16s	(X"AAAA",	X"BB",	X"DDDD",	
q	16	повільний	читання	REGread16s	(X"AAAA",	X"BB",		REG_RD,
l	32	швидкий	запис	REGWRITE	(X"AAAA",	X"BB",	X"DDDDDDDD",	
s	32	швидкий	читання	REGread	(X"AAAA",	X"BB",		REG_READ,

У ході роботи транслятор аналізує правильність табличних описів, у разі виявлення помилки в описі якогось циклу шини формується повідомлення про помилку.

Приклади табличних описів окремих циклів подано нижче:

w 123456 78 – швидкий запис шістнадцяткового числа 78 до адреси шістнадцяткової адреси 123456;

s 9abcde – швидке читання з шістнадцяткової адреси 9abcde.

Виклик окремих VHDL-процедур веде до їх виконання і формування на вході тестованого ядра відповідної послідовності інтерфейсних сигналів.

Таблиця 2

Змінна частина опису виклику процедури

постійна частина VHDL-опису виклику процедури циклу шини									
синхро-імпульси	швидка виборка	повільна виборка	вибір 1-го байта	вибір 3-го байта	сигнал читання	сигнал запису	шина адреси	шина даних	сигнал очікування
clk,	cs_f,	cs_s,	bs1,	bs3,	rd,	wr,	A,	D,	wait);

Нижче наведено перетворення табличних описів послідовностей циклів 16-бітної шини на часові діаграми сигналів програмного стенда.

Приклад 1 (
рис. 4):

v 01800A 0000; – повільний запис;

q 018008; – повільне читання до регістра REF_READ.

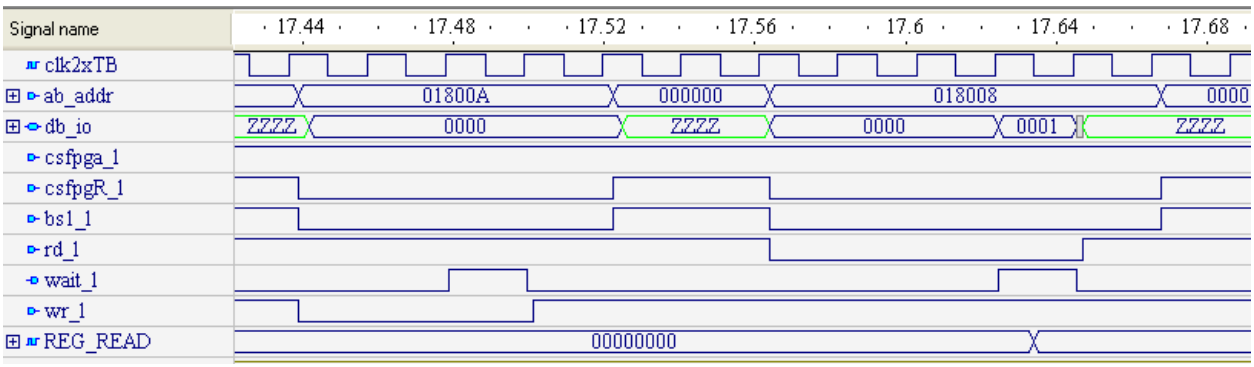

Рис. 4. Повільний запис, повільне читання

Приклад 2 (
рис. 5):

r 01800E; – швидке читання;

w 018020 0018; – швидкий запис;

v 018022 0000; – повільний запис.

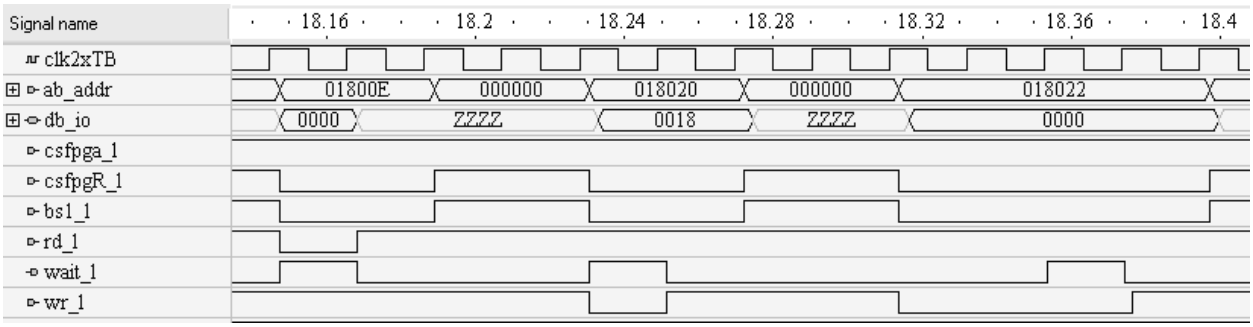

Рис. 5. Швидке читання, швидкий запис, повільний запис

Програма транслятора написана мовою С.

6. Подальший напрям роботи

Для під'єднання ядер до інших вузлів комп'ютерних систем на кристалі ПЛІС крім паралельних шин, також використовують послідовні шини: SPI, I2C, RS, CAN та інші. Запропонований підхід до тестування ядер на програмних стендах, реалізований для паралельних шин, пропонується застосувати і для перевіряння взаємодії ядер з іншими вузлами за допомогою вказаних послідовних шин.

Також частою процедурою під час тестування ядер є повторюваність деяких циклів шини, наприклад, багаторазове читання з постійної адреси. Пропонується надалі ввести до табличних описів циклів шини команди n-разового повторення деяких циклів та їх послідовностей.

До складу систем на кристалі ПЛІС входять ядро мікроконтролера і ядра функціональних вузлів системи, якими цей мікроконтролер керує. Опис роботи системи здійснюється мовою VHDL, яку потім використовують засоби проектування топології кристала ПЛІС, програми роботи мікроконтролера пишуть мовою С. Тобто під час створення сучасних систем на кристалі доводиться користуватися двома мовами – VHDL та С.

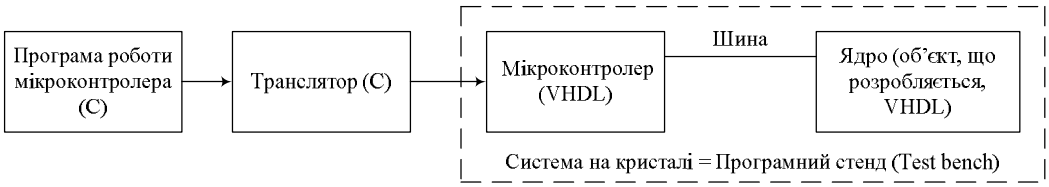

Рис. 6. Схема проектування та тестування систем на кристалі

Ця робота може стати базою для переходу під час проектування до використання розробником однієї мови С, для чого потрібні подальший розвиток і вдосконалення запропонованого транслятора. Результатом такої роботи буде схема проектування та тестування систем на кристалі (рис. 6), де вихідними даними буде програма роботи мікроконтролера та додатковий опис його структури мовою С, а результатом – опис системи на кристалі мовою VHDL, яку розуміють засоби проектування топології кристалів ПЛІС.

Висновки

У цій статті описано розв'язання задачі створення програмного стенда для тестування ядер ПЛІС. Задача зведена до розроблення транслятора – елемента системи автоматизованого проектування ПЛІС, який перетворює табличне представлення тестових впливів на VHDL-описи програмного стенда. Запропоноване рішення дає змогу прискорити процес тестування ядер за рахунок використання розроблених таблиць тестових впливів та VHDL-описів циклів стандартних шин. Як напрям подальших робіт визначено адаптацію запропонованого рішення для послідовних інтерфейсів, генерацію описів повторюваних циклів шин та інтерфейсів, складання замість табличних описів циклів шин їхніх описів мовою С (програми роботи мікроконтролерів) з подальшою трансляцією в VHDL-описи.

1. Глухов В., Лукенюк А., Шендерук С.. НВІС системи збору наукової інформації супутника "Січ-2". Матеріали 5-ї Міжнародної науково-технічної конференції "Сучасні комп'ютерні системи та мережі: розробка та використання" ACSN-2011 29 вересня – 01 жовтня, 2011, Львів, Україна. С.57-60. 2. XBLKNM 361. Constraints Guide. ISE 8.1i. www.xilinx.com 3. Spartan-6 FPGA SelectIO Resources User Guide. UG381 (v1.4). December 16, 2010. © 2009–2010 Xilinx, Inc. 4. IEEE Std 1076-2008 IEEE Standard VHDL Language Reference Manual. Approved: 26 September 2008 IEEE SA-Standards Board 5. ARM920T-based Microcontroller. AT91RM9200. © 2006 Atmel Corporation. 6. AT91SAM ARM-based Embedded MPU. SAM9G45. © 2011 Atmel Corporation. 7. CAN Specification. Version 2.0. 1991, Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart. 8. Лукенюк А.А., Шендерук С.Г.. Принципи побудови бортової системи збору та обробки наукової інформації для космічних досліджень // VI Міжнародна науково-технічна конференція "Гіротехнології, навігація, керування рухом та конструювання авіаційно-космічної техніки": збірник доповідей. Частина II. 26–27 квітня 2007 р. Київ. С.313-319. 9. VHDL Testbench Tutorial. University of Pennsylvania, Department of Electrical and Systems Engineering. Updated February 12, 2012.