

ПІДХОДИ ДО НИЗЬКОРІВНЕВОГО МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ

В статті проаналізовано підходи до моделювання комп'ютерних систем (КС) на основі програмованих логічних інтегральних схем (ПЛІС). Виявлено, що основними моделями є абстрактна та низькорівнева. Виокремлено завдання, переваги та недоліки кожної з них. Отримано висновок, що застосування низькорівневих моделей в більшості випадків більш виправдане. Проте низька швидкість їх роботи створює передумови розроблення двох моделей, кожна з яких використовують для відповідного кола завдань. Розроблено рекомендацію щодо використання моделей.

Конкретизовано завдання відлагодження вузлів та блоків на основі низькорівневого моделювання комп'ютерних систем на основі ПЛІС, а саме: 1) використання моделі системи як тестової оболонки при відладці кожного з пристроїв; 2) імітація аварійних ситуацій, виявлених при тестуванні реальних пристроїв; 3) створення поточкових тестів для виявлення помилок при взаємодії пристроїв; 4) збір статистики для оцінки алгоритмів арбітражу і управління потоками даних; 5) оцінка продуктивності системи. Проведений аналіз кожного підзавдання на предмет призначення, можливостей, переваг та недоліків.

Ключові слова: низькорівневе моделювання, комп'ютерна система, програмована логічна інтегральна схема.

A.S. LYAPANDRA

Ternopil national economic university

APPROACHES TO LOW-LEVEL MODELING OF FIELD-PROGRAMMABLE GATE ARRAYS COMPUTER SYSTEMS

In the article approaches to modeling field-programmable gate array (FPGA) based computer systems is analyzed. Found that the basic are low-level and abstract models. A task, advantages and defects, is distinguished each of them. A conclusion is got, that application of low-level models in most cases is more justified. However low rate of speed of their work creates pre-conditions of development of two models, each of which is used for the respective range of tasks. Recommendation is worked out in relation to the use of models.

The task of debugging units and blocks is specified on the basis of low-level design of the computer systems on the basis of FPGA, namely: 1) use of model of the system as a test shell at debugging each of devices; 2) imitations of emergency situations, real devices educed at testing; 3) creations of thread tests are for the exposure of errors at co-operation of devices; 4) collections of statistics are for the estimation of algorithms of arbitration and management of data threads; 5) estimation of the system productivity. The conducted analysis of every subtask is for the purpose a purpose, possibilities, advantages and defects.

Keywords: low-modeling, computer system, duration of implementation of algorithm, field-programmable gate array.

Постановка задачі

Останнім часом все більшої популярності набуває методика проектування з використанням програмованих логічних інтегральних схем. Малий час проектування, вартість розробки, простота моделювання, а також можливість оперативного виправлення помилок на етапі реалізації робить цю методику перспективною. До недавнього часу відносно скромні характеристики програмованих схем в порівнянні з напівзавомними і замовленими кристалами дозволяли використовувати ПЛІС лише в невеликих проектах і при розробці нескладних пристроїв [1].

Проте ПЛІС останніх поколінь значно розширили сферу застосування цієї технології. Зокрема, часові характеристики сучасних ПЛІС задовольняють вимогам, що пред'являються як до периферійних, системних контролерів, так і контролерів високошвидкісних інтерфейсів введення-виведення. Розміри сучасних ПЛІС дають можливість досягти високого рівня інтеграції.

Отже, сучасні ПЛІС дають можливість реалізувати схеми високої складності і швидкодії шляхом використання лише ПЛІС-технології. Типовим прикладом подібної системи може бути множина ПЛІС, що становлять набір системної логіки материнської плати. Проте, враховуючи значні зміни характеристик ПЛІС, задача вдосконалення підходів до низькорівневого моделювання комп'ютерних систем на основі програмованих логічних інтегральних схем є актуальною.

Аналіз досліджень та публікацій

Задачі проектування комп'ютерних систем присвячені численні роботи, серед яких відзначимо роботи В.Воеводіна та Вл.Воеводіна [2], Шпаковського Г.І. [3]. У цих працях розглянуто особливості проектування комп'ютерних систем, проведено аналітичний аналіз їх параметрів.

Виділення невирішених частин

Перевагою комп'ютерних систем на основі ПЛІС є можливість конфігурування їх структури відповідно до алгоритму розв'язання задачі та структури даних. Проте питання вдосконалення підходів до низькорівневого моделювання комп'ютерних систем на основі ПЛІС у вищенаведених працях розкрито не достатньо повно [2-3].

Формулювання цілей

Ціллю циклу робіт є отримання комплексної моделі, на основі якої в подальшому провести проектування вузлів реконфігурованої обчислювальної системи.

Ціллю цієї статті є вдосконалення підходів до низькорівневого моделювання комп'ютерних систем на основі програмованих логічних інтегральних схем.

Абстрактне і низькорівневе моделювання

В процесі проектування систем, що складаються з множини електронних пристроїв, важлива роль відводиться етапу моделювання системи. При розробці моделей керуються однією з таких концепцій: 1) використання початкових описів пристроїв на мові моделювання апаратури; 2) створення абстрактних описів пристроїв на довільній мові високого рівня. 3) враховують завдання, що вирішуються в процесі моделювання системи.

Моделювання з використанням першої концепції є низькорівневим, на протизагу абстрактному моделюванню. До переваг першого методу відносяться 1.1.1) адекватність поведінки пристроїв при моделюванні реальній роботі пристроїв (при моделюванні використовуються ті ж описи, що і при синтезі); 1.1.2) простота діагностики помилок, виявлених в роботі пристроїв (засоби моделювання мов опису апаратури надають повний доступ до внутрішніх структур пристроїв). З іншого боку очевидні такі недоліки методу: 1.2.1) низька швидкість моделювання (через свої особливості, мови моделювання апаратури сильно програють в швидкості універсальним мовам); 1.2.2) вузька спеціалізація мов моделювання апаратури (мови моделювання апаратури не надають досить засобів для розробки програм; в результаті написання різноманітних сервісних модулів ускладнене).

Розглянемо також переваги і недоліки абстрактного моделювання. До переваг можна віднести: 2.1.1) високу швидкість роботи; 2.1.2) універсальність (мови високого рівня зазвичай надають широкий спектр різних засобів роботи з графікою, статистичної обробки і об'єктно-орієнтованого опису). До недоліків: 2.2.1) відсутність точної відповідності абстрактного опису пристроїв опису, призначеному для фізичної реалізації (що утрудняє діагностику помилок, допущених в описі самих пристроїв, і вимагає додаткових зусиль із доказу адекватності результатів, отриманих при абстрактному моделюванні); 2.2.2) опис пристроїв на мові високого рівня більш трудомістко, оскільки мови моделювання апаратури надають широкий набір засобів для вирішення саме цього завдання [4].

Розглянемо завдання, що вирішуються в процесі моделювання системи, і використання кожної з концепцій у кожному конкретному випадку: 3.1) використання моделі системи як тестової оболонки при відлагодженні кожного з пристроїв. Рішення цієї задачі неможливе методами абстрактного моделювання, оскільки основна робота в процесі моделювання при відладці пристроїв полягає саме у виявленні і діагностиці помилок, допущених в описі пристроїв. В той же час, низькорівнева модель добре підходить для вирішення цього завдання, оскільки налагоджувальні тести зазвичай невеликі і швидкість моделювання не є вирішальною; 3.2) створення потокових тестів для виявлення помилок при взаємодії пристроїв. Для вирішення цього завдання абстрактна модель також погано пристосована, оскільки результатом роботи потокових тестів часто є виявлення помилок в пристроях, що становлять систему. Діагностика і корекція подібних помилок в абстрактній моделі ускладнена. Застосування низькорівневої моделі, яка надає широкий набір засобів для діагностики і корекції помилок пристроїв, є неефективним через низьку швидкість моделювання, оскільки потокові тести мають великий обсяг; 3.3) збір статистики для оцінки алгоритмів арбітражу і управління потоками даних. Для вирішення цього завдання краще підходить абстрактна модель, оскільки для збору достовірних статистичних даних потрібний великий обсяг даних. Обробка статистичних даних засобами універсальної мови високого рівня також є прийнятною. Помилки, виявлені при рішенні цієї задачі, відносяться до досліджуваних алгоритмів і тому не вимагають діагностики в межах окремого пристрою. Проте низькорівнева модель забезпечує достовірний потік даних між пристроями і, як наслідок, - достовірніші статистичні дані, і лише низька швидкість моделювання не дає можливість використовувати її для вирішення подібних завдань; 3.4) оцінка продуктивності системи. Багато в чому це завдання схоже з попереднім завданням, оскільки полягає у зборі та обробці статистичних даних. Тому усе, що говорилося про завдання оцінки алгоритмів арбітражу і управління потоками даних, відноситься і до цього випадку. Проте, на відміну від попереднього завдання, завдання оцінки продуктивності системи на реальних навантаженнях вимогливіше до достовірності поведінки моделі і, як наслідок, - використання низькорівневої моделі є більш прийнятнішим. Але низька швидкість моделювання не дає використовувати переваги цього підходу; 3.5) верифікація протоколів взаємодії пристроїв. Для вирішення цього завдання наявність детально описаних пристроїв системи не є необхідним. Тому вона часто вирішується паралельно з розробкою самих пристроїв. Оскільки протоколи взаємодії пристроїв зазвичай добре формалізовані, то вони без зусиль описуються на будь-якій мові високого рівня. До моменту, коли можливе використання початкових описів пристроїв для моделювання системи, це завдання вже не коштує. Таким чином, низькорівневі моделі зазвичай не використовуються для верифікації протоколів взаємодії пристроїв. Проте, якщо подібне завдання виникає на пізнішому етапі проектування (наприклад, при розширенні вже розробленої системи або при заміні застарілого протоколу на новіший), то її можна успішно вирішувати за допомогою низькорівневої моделі. Тести верифікації зазвичай мають невеликий об'єм, і, як наслідок, швидкість моделювання не є критичною. Необхідно також помітити, що при проектуванні систем часто

використовуються стандартні протоколи взаємодії систем. В цьому випадку подібне завдання не виникає взагалі [5].

З усього вищевикладеного можна зробити висновок, що застосування низькорівневих моделей в більшості випадків більш виправдане і лише низька швидкість їх роботи примушує використовувати абстрактні моделі, незважаючи на їх недоліки. Тому рекомендовано за допомогою низькорівневої моделі проводити відлагодження вузлів та блоків КС, а за допомогою абстрактної - верифікацію протоколів і оцінку ефективності алгоритмів. Завдання, що залишилися, вирішувати відповідно до специфіки конкретної розробки і переваг розробників за допомогою того або іншого методу [6].

Завдання низькорівневого моделювання комп'ютерних систем на основі ПЛІС

При розробці систем, що містять множину ПЛІС-пристроїв, виникає необхідність об'єднання описів усіх пристроїв з метою створення моделі усєї системи. За допомогою такої моделі можна вирішувати широкий спектр завдань, серед яких виділимо такі: 1) використання моделі системи як тестової оболонки при відладці кожного з пристроїв; 2) імітація аварійних ситуацій, виявлених при тестуванні реальних пристроїв; 3) створення поточкових тестів для виявлення помилок при взаємодії пристроїв; 4) збір статистики для оцінки алгоритмів арбітражу і управління потоками даних; 5) оцінка продуктивності системи.

При розробці великої системи з множиною ПЛІС пристроїв виникає необхідність налагодження кожного з них. Для цього необхідно помістити кожен пристрій в тестове середовище. Для кожного пристрою таке середовище буде індивідуальне, але при цьому можна виділити загальні елементи: по-перше, має бути присутнім джерело навантаження та модуль опрацювання. Цей модуль відповідає за формування вхідних дій на тестований пристрій і реакцію на його вихідні дії. По-друге, має бути присутнім блок моніторингу і статистики, який визначатиме некоректні стани інтерфейсу пристрою і, якщо це необхідно, збирати статистику роботи пристрою.

При використанні системної моделі як тестової оболонки для усіх пристроїв, відпадає необхідність у використанні штучних генераторів навантаження для кожного пристрою, оскільки джерелом навантаження для будь-якого пристрою зазвичай служить інший тестований пристрій. У моделі системи потрібний лише один генератор навантаження для усєї системи. Один модуль моніторингу і статистики потрібний для кожного інтерфейсу, що при моделюванні систем з інтерфейсами, підтримувальними більше двох пристроїв, дасть вигоду в порівнянні з індивідуальним тестуванням пристроїв [7].

При проектуванні КС на основі ПЛІС частину тестування зручно проводити на реальних пристроях, але при усіх перевагах цього методу очевидні і недоліки, головними з яких є обмеженість засобів аналізу виявлених аварійних ситуацій. При тестуванні реальних пристроїв придатними для спостереження є лише інтерфейсні сигнали. Найпростіший спосіб вирішення цієї проблеми - використання незайнятих контактів корпусу кристала для додавання в інтерфейс внутрішніх сигналів. Цим методом не можна розв'язати проблему повністю, оскільки при виборі корпусу розробники керуються, зокрема, критерієм вартості мікросхеми і, як наслідок, кількість незайнятих ніжок невелика. У подібній ситуації можна використовувати модель системи, що дає можливість в точності повторити отриману в тесті реального пристрою критичну ситуацію і провести її повний аналіз засобами мови опису обладнання.

Наявність в моделі системи генератора навантаження дає можливість проводити широкий спектр різноманітних поточкових тестів, від імітації реальних режимів роботи системи до інтенсивних стрес-тестів (максимальне завантаження пристроїв). Імітація реальних режимів роботи допомагає виявляти помилки взаємодії пристроїв на стадіях від первинного налагодження до створення фізичного дизайну схем. При подібному моделюванні виникає проблема генерації навантаження, яке еквівалентне реальному. Існує декілька шляхів вирішення цієї проблеми. Найпростіший з них - використання трас роботи пристрою, що навантажує реальну систему, його аналога або його точної моделі. У разі ж відсутності подібної інформації можна використовувати випадкові параметризовані генератори, які настраюються відповідно до теоретичних описів параметрів потоку. У такому вигляді навантаження відрізнятиметься від реального, але для первинного тестування відмінність буде несуттєвою [8].

Стрес-тести потрібні для визначення пікової пропускну здатності системи і її стійкості до пікових навантажень. Подібні тести реальної системи зазвичай неможливі, на відміну від моделі, де для організації подібних тестів потрібна лише можливість генерації навантаження, відмінного від реального навантаження. Для налаштування деяких пристроїв і оцінки роботи певних алгоритмів потрібні статистичні дані про роботу системи. Можливості по збору статистичної інформації в реальній системі обмежені і, за відсутності апаратної підтримки, дають невисоку точність. Модель забезпечує вищу точність і ширші можливості, але лише за наявності генератора, здатного створювати реальне навантаження. За відсутності можливості забезпечити еквівалентність навантаження використання моделі недоцільне. В більшості випадків рекомендується використовувати поєднання методів моделювання і тестів реальної системи.

Якщо немає можливості провести оцінку продуктивності системи на реальних тестах, то можна використовувати для цих цілей модель системи. Як і у попередньому випадку необхідно мати генератор реального навантаження. Не дивлячись на те, що дослідження продуктивності системи краще проводити на реальних тестах, деякі частини цього дослідження можна проводити тільки на моделі. До цих частин відноситься вже згадуване раніше дослідження пікової продуктивності системи і дослідження продуктивності в нестандартних умовах, які неможливо відтворити в реальній системі.

Висновки

В статті проаналізовано підходи до моделювання КС на основі ПЛІС. Виявлено, що основними моделями є абстрактна та низькорівнева. Виокремлено завдання, переваги та недоліки кожної з них. Отримано висновок, що застосування низькорівневих моделей в більшості випадків більш виправдане. Проте низька швидкість їх роботи створює передумови розроблення двох моделей, кожену з яких використовують для відповідного кола завдань.

Розроблено рекомендацію за допомогою низькорівневої моделі проводити відлагодження вузлів та блоків КС, а за допомогою абстрактної - верифікацію протоколів і оцінку ефективності алгоритмів. Завдання, що залишилися, вирішувати відповідно до специфіки конкретної розробки і переваг розробників за допомогою того або іншого методу.

Конкретизовано завдання відлагодження вузлів та блоків на основі низькорівневого моделювання комп'ютерних систем на основі ПЛІС, а саме: 1) використання моделі системи як тестової оболонки при відладці кожного з пристроїв; 2) імітація аварійних ситуацій, виявлених при тестуванні реальних пристроїв; 3) створення поточкових тестів для виявлення помилок при взаємодії пристроїв; 4) збір статистики для оцінки алгоритмів арбітражу і управління потоками даних; 5) оцінка продуктивності системи. Проведений аналіз кожного підзавдання на предмет призначення, можливостей, переваг та недоліків.

Література

1. Ляпандра А.С. Вплив параметрів комунікаційної частини реконфігурованої кластерної системи на тривалість виконання алгоритму / А.С. Ляпандра // Вимірювальна та обчислювальна техніка в технологічних процесах. - Хмельницький, 2014. - № 4. - С. 125-127.
2. Воеводин В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб.: БХВ-Петербург, 2002. – 608 с.
3. Шпаковский Г.И. Параллельное программирование и аппаратура / Г.И. Шпаковский. – Минск, БГУ, 2012. – 184 с.
4. Ляпандра А.С. Підхід до узагальненого оцінювання ресурсів програмованих логічних інтегральних схем / А.С. Ляпандра // Електротехнічні та комп'ютерні системи. – К.: "Техніка", 2012. – №7. – С.92–96.
5. Ляпандра А.С. Зменшення часових затримок ПЛІС у реконфігурованих комп'ютерних системах / А.С. Ляпандра // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький, 2012. – Т.193, №5. – С.147–151.
6. Ляпандра А.С. Параметри функціональної моделі апаратного забезпечення оптимізації мережевого трафіку / А.С. Ляпандра // Вісник Тернопільського національного технічного університету. – Тернопіль, 2011. – Т.16. №1. – С.161–167.
7. Lyapandra A. Load balancing reconfigurable cluster system / A. Lyapandra // Вісник Хмельницького національного університету. Технічні науки. - Хмельницький, 2015. - № 1. - С. 177-180.
8. Ляпандра А.С. Ідентифікація захворювання на основі гармонійного аналізу хемілюмінесцентних сигналів біопроб / А.С. Ляпандра // Вісник Хмельницького національного університету. – 2007. – № 6, Ч.1. – С.137–144.

References

1. Lyapandra A.S. Vpliv parametrov komunikatsionoyi chastini rekonfigurovanoi klasternoyi sistemi na trivalist vikonannya algoritmu / A.S. Lyapandra // Vimiryvalna ta obchislyvalna tekhnika v tekhnologichnikh protsesakh. - Khmelnytsky, 2014. - № 4. - pp. 125-127.
2. Voevodin V. V., Voevodin Vl. V. Parallelnye vychisleniya. – Spb.: BKHV-Peterburg, 2002. – 608 p.
3. Shpakovskiy G.I. Parallelnoe programmirovaniye i apparatura. – Minsk, BGU, 2012 g., 184 p.
4. Lyapandra A.S. Pidkhid do uzagalnenogo otsinyuvannya resursiv programovanykh logichnykh integralnykh skhem / A.S. Lyapandra // Elektrotekhnichni ta kompyuterni systemy. – K.: "Tekhnika", 2012. – №7. – pp.92–96.
5. Lyapandra A.S. Zmshennya chasovy/ikh zatrymok PLIS u rekonfigurovanykh kompyuternykh systemakh / A.S. Lyapandra // Visnyk Khmelnytskoho natsionalnoho universytetu. Technical science. Khmelnytsky. 2012. Volume 193.– Issue 5. pp.147–151.
6. Lyapandra A.S. Parametry funktsionalnoyi modeli aparatnogo zabezpechennya optymizatsiyi merezhevoogo trafiku / A.S. Lyapandra // Visnyk Ternopilskoho natsionalnoho tekhnichnoho universytetu. Ternopil, 2011. Volume 16. Issue 1. – pp.161–167.
7. Lyapandra A. Load balancing reconfigurable cluster system / A. Lyapandra // Visnyk Khmelnytskoho natsionalnoho universytetu. Technical science. Khmelnytsky. 2015. Volume 221.– Issue 1. pp. 177-180.
8. Lyapandra A.S. Identyfikatsiya zakhvoryuvannya na osnovi harmoniynogo analizu khemilyuminestsentnykh sygnaliv bioprob // Visnyk Khmelnytskoho natsionalnoho universytetu. Technical science. Khmelnytsky. 2007. Volume 100.– Issue 6. Part 1. pp.137–144.

Рецензія/Peer review : 17.5.2015 р.

Надрукована/Printed :26.6.2015 р.

Стаття рецензована редакційною колегією