

УДК 004.383, 004.4'24

В. А. МЕЛЬНИК

Національний університет «Львівська політехніка», Львів

ЗАСОБИ АВТОМАТИЧНОГО ГЕНЕРУВАННЯ ПРОГРАМНИХ МОДЕЛЕЙ СПЕЦІАЛІЗОВАНИХ ПРОЦЕСОРІВ У САМОКОНФІГУРОВНІЙ КОМП'ЮТЕРНІЙ СИСТЕМІ

Анотація. В статті запропоновано структуру і розроблено принципи функціонування та алгоритм роботи засобів автоматичного генерування програмних моделей спеціалізованих процесорів, що входять до складу самоконфігуровної комп'ютерної системи.

Ключові слова: Самоконфігуровна комп'ютерна система, засоби генерування програмних моделей процесорів, програмні моделі спеціалізованих процесорів.

Аннотация. В статье предложена структура и разработаны принципы функционирования и алгоритм работы средств автоматического генерирования программных моделей специализированных процессоров, входящих в состав самоконфигурируемой компьютерной системы.

Ключевые слова: Самоконфигурируемая компьютерная система, средства генерирования программных моделей процессоров, программные модели специализированных процессоров.

Abstract. This paper proposes specialized processors IP Cores automatic generation facilities: their structure, principles of work and algorithm of functioning. Proposed facilities are intended to be used in the self-configurable computer system.

Key words: Self-configurable computer system, IP Cores generation facilities, specialized processors IP Cores..

Вступ

На протязі останнього десятиліття відбулися значні зміни в галузі проектування та мікроелектронного виробництва комп'ютерних засобів. Отримала подальший розвиток технологія проектування програмних моделей обчислювальних пристроїв [1], яка описує процес розробки архітектури обчислювальних пристроїв на рівні міжрегістрових передач з використанням мов опису апаратних засобів (VHDL [2], Verilog [3], ін.) і їх реалізації в мікросхемах програмовної логіки, найчастіше – програмовних логічних інтегральних схемах (ПЛІС).

Покращення технічних характеристик ПЛІС та їх невисока вартість посприяли відновленню інтересу до напрямку розробки реконфігурованих комп'ютерних систем (РККС), в яких універсальні комп'ютери об'єднані з реконфігурованим середовищем на основі мікросхем програмовної логіки. Об'єднання універсального комп'ютера з реконфігурованим середовищем в єдину систему дозволяє виконувати обчислювальні завдання значно швидше, ніж без нього, що забезпечується створенням в цьому середовищі спеціалізованих процесорів для апаратного виконання найбільш обчислювально складних частин завдань. Наявність на ринку реконфігурованих прискорювачів – пристроїв, які містять реконфігуроване середовище і засоби взаємодії з хост-комп'ютером через один або декілька стандартних інтерфейсів, робить можливою реалізацію РККС на базі довільного універсального, у тім числі й персонального комп'ютера.

Поряд з високою продуктивністю, яка забезпечується реконфігурованими комп'ютерними системами (РККС), є й ряд проблем, пов'язаних з їх застосуванням. Це, зокрема, значні часові затрати на виконання розподілу обчислювального навантаження, часта відсутність необхідних для реалізації в реконфігурованому середовищі РККС програмних моделей спеціалізованих процесорів, що викликає необхідність розробки цих моделей від початку, та високі додаткові вимоги до кваліфікації користувачів РККС, оскільки вони, крім моделювання і програмування, повинні виконувати системний аналіз та розподіляти обчислювальне навантаження, розробляти архітектуру спеціалізованих процесорів, здійснювати їх синтез та реалізацію в реконфігурованому середовищі.

Аналіз останніх досліджень

Реконфігуровні комп'ютерні системи, їх будову та організацію функціонування на різних рівнях висвітлено в роботі [4]. В результаті аналізу цієї роботи можна зробити висновок про значну інертність таких систем, що є наслідком необхідності виконання їх конфігурування / реконфігурування перед використанням. В роботі [5] запропоновано концепцію побудови самоконфігурованих прискорювачів, а в роботі [6] – метод їх самоконфігурування, згідно з якими всі кроки із здійснення конфігурування виконуються комп'ютерною системою автоматично.

Запропоновано в роботі [5] концепцію взято за основу при побудові самоконфігуровної комп'ютерної системи, яка позбавлена проблем застосування, притаманних РККС. Самоконфігуровна комп'ютерна система (СККС) – це комп'ютерна система, яка містить універсальний комп'ютер, реконфігуроване середовище, побудоване на основі ПЛІС або інших типів інтегральних схем програмовної логіки, а також програмні засоби, які автоматично розподіляють обчислювальне навантаження між універсальним комп'ютером і реконфігурованим середовищем та самостійно виділяють з комп'ютерної програми фрагменти, виконання яких в реконфігурованому середовищі прискорює роботу комп'ютерної системи, самостійно створюють в цьому середовищі пристрої (спеціалізовані процесори) для виконання виділених фрагментів програми та самостійно організують їх функціонування та взаємодію з універсальним процесом-

ром. Проблема інертності, притаманна РККС і викликана витратами часу на їх конфігурування, в СККС знімається завдяки зміні способу опрацювання нею інформації.

Постановка завдання

Однією з компонент СККС є система генерування, функцією якої є автоматичне створення програмних моделей спеціалізованих процесорів на основі опису виконуваних ними обчислювальних алгоритмів мовою програмування високого рівня. Під програмною моделлю спеціалізованого процесора розуміють його представлення у вигляді програми мовою опису апаратних засобів на рівні міжрегістрових передач або нижчому.

Сьогодні існує значна кількість методів та програмних засобів генерування програмних моделей спеціалізованих процесорів. На ринку наявні засоби автоматичної генерації програмних моделей [7, 8], бібліотеки програмних моделей [9], а також системи високорівневого проектування програмних моделей спеціалізованих процесорів на основі опису алгоритмів їх роботи мовою високого рівня [10–12]. В роботі [13] сформовано вимоги до системи генерування програмних моделей спеціалізованих процесорів, на основі яких можна провести аналіз особливостей існуючих засобів генерування на предмет ефективності їх застосування в СККС. Разом з тим, питання розроблення структури цієї системи та методики її функціонування досі не розглядалися. Саме ці питання і є предметом наукового дослідження, висвітленого в даній статті.

Функція системи генерування програмних моделей спеціалізованих процесорів

Система генерування програмних моделей спеціалізованих процесорів (далі – система генерування) є одним з базових програмних засобів, що входять до складу СККС і надають їй властивості самоконфігуровності. Функцією системи генерування в СККС є автоматичне створення програмної моделі процесора для виконання обчислювальних алгоритмів, отриманих від іншої компоненти СККС – системи розподілу обчислювального навантаження. Функцію системи генерування можна описати наступним виразом:

$$PMSP(A_i, i = \overline{1 \dots z}) = GS(PR^{SCCS}(A_i, i = \overline{1 \dots z})) \quad (1)$$

де GS – система генерування; PR^{SCCS} – подана деякою мовою програмування програма, що описує обчислювальні алгоритми $A_i, i = \overline{1 \dots z}$, які має виконувати реалізований в реконфігурованому середовищі СККС спеціалізований процесор (далі – програма спеціалізованого процесора (СП)); $PMSP$ – програмна модель спеціалізованого процесора для виконання обчислювальних алгоритмів $A_i, i = \overline{1 \dots z}$; z – кількість обчислювальних алгоритмів.

Типи засобів генерування програмних моделей обчислювальних пристроїв

Сьогодні на ринку присутні наступні типи програмних засобів, які виконують функцію автоматичного створення програмних моделей обчислювальних пристроїв:

1. генератори на основі конфігурованих програмних моделей процесорів [7, 8];
2. генератори на основі бібліотек програмних моделей процесорів та процесорних компонент [9];
3. системи автоматизованого проектування програмних моделей процесорів з мови високого рівня [10 – 12].

Генератором програмних моделей процесорів називають програмну систему, яка містить конфігуровану програмну модель процесора та засоби автоматичного виконання її конфігурування. Генератори автоматично створюють програмні моделі процесорів, виходячи з значень вхідних конфігураційних параметрів. Кількість генерованих програмних моделей процесорів залежить від кількості k конфігураційних параметрів $P_i, i = \overline{1 \dots k}$, розрядності n_{CCP_i} конфігураційного коду кожного параметра, яка рівна $\lceil \log_2 N_{P_i} \rceil$, де N_{P_i} – кількість значень цього параметра, $\lceil \rceil$ – більше ціле значення, та кількості взаємовиключних комбінацій між параметрами. Результатом роботи генератора є програмні моделі процесорів, представлені однією або кількома мовами опису апаратних засобів, системи тестування програмних моделей процесорів, командні скрипти для виконання їх компіляції, симуляції та синтезу, документація користувача та ін.

Принципи побудови генератора програмних моделей процесорів та його реалізація можуть бути різними. Найпростішим є створення програмних моделей цифрових елементів на основі їх конфігурованих програмних моделей мовами VHDL або Verilog. Саме таким способом реалізовано системи IP Core Generator фірми Aldec [7] та CORE Generator System фірми Xilinx [8]. Разом з тим, ефективність

цього підходу зменшується з підвищенням складності обчислювального пристрою. Тому для генерації складних пристроїв використовують інші, ефективніші підходи, зокрема, підхід на основі бібліотек програмних моделей процесорів та їх компонент.

Генератор на основі бібліотеки програмних моделей процесорів [9] складається з бібліотеки програмних моделей, описаних мовою опису апаратних засобів, та засобів керування бібліотекою. Засоби керування бібліотекою отримують від користувача конфігураційні коди, що задають параметри програмної моделі процесора, та вибирають з множини наявних в бібліотеці програмних моделей процесорів таку модель, яка відповідає заданим параметрам.

Генератори на основі бібліотеки процесорних компонент [9] «конструюють» програмну модель, використовуючи для цього розміщені в бібліотеці компоненти, представлені на певному рівні деталізації процесора. В таких генераторах присутні засоби конфігурування, які, крім вибирання потрібних компонент з бібліотеки, виконують їх компоновку і генерують файли опису верхнього рівня мовою опису апаратних засобів для кожного з рівнів ієрархії.

Середовища проектування, які дозволяють автоматично створювати програмні моделі потрібних процесорів з мови високого рівня, в більшості реалізацій використовують підхід виконання компіляції високорівневого опису алгоритму роботи обчислювального пристрою в код опису його архітектури мовою опису апаратних засобів. Для створення високорівневого опису алгоритму роботи обчислювального пристрою більшість засобів використовує модифікований варіант мови C, наприклад Mentor Graphics використовує IEEE standard System C, Celoxica – Handle C, Nallatech – DIME C, Impulse – Impulse-C, Mitronics – Mitrion-C, деякі засоби використовують ANSI C.

Особливості застосування засобів автоматичного генерування програмних моделей обчислювальних пристроїв в СККС

Відповідно до результату досліджень, висвітлених в роботі [13], засоби системи генерування в СККС повинні задовольняти вимогам а) в частині функціональної повноти системи генерування; б) в частині технічних характеристик генерованих системою генерування програмних моделей процесорів; та в) в частині архітектури генерованих системою генерування програмних моделей процесорів.

Як показали дослідження, єдиним типом засобів генерування програмних моделей процесорів, які характеризуються функціональною повнотою, є системи автоматизованого високорівневого проектування. Більше того, цей тип засобів є єдиним, де функціональні характеристики генерованого процесора задаються описом алгоритму його роботи на мові програмування високого рівня, що є необхідною умовою застосування в СККС. Тому як головний елемент системи генерування доцільно використовувати систему автоматизованого високорівневого проектування. Разом з тим, цей тип засобів генерування в більшості не відповідає вимогам в частині архітектури та в частині технічних характеристик процесорів; цим вимогам відповідають генератори на основі бібліотек та конфігурованих програмних моделей.

З огляду на це при побудові системи генерування доцільно використовувати декілька типів засобів генерування, що дасть можливість досягти максимального задоволення вимог до неї, зберігаючи при цьому її ортогональність. Крім того:

1. для виконання часто вживаних та критичних за часом виконання обчислень та прикладних алгоритмів доцільно використати наперед розроблені, добре оптимізовані програмні моделі процесорів, які можуть бути отримані з використанням генераторів на основі бібліотек;
2. програмні моделі процесорів в бібліотеках доцільно представляти на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС;
3. використання генераторів на основі конфігурованих програмних моделей процесорів не виглядає доцільним з огляду на те, що вони не мають жодних переваг над генераторами на основі бібліотек процесорних компонент, але поступаються їм ефективністю механізмів конфігурування;
4. оскільки функціональні характеристики процесорів для генераторів на основі бібліотек програмних моделей процесорних компонент задаються не програмою, а конфігураційними кодами, взаємодію такого генератора з системою розподілу обчислювального навантаження необхідно організувати шляхом прямої передачі команд від користувача до системи генерування.

Структура засобів генерування програмних моделей спеціалізованих процесорів

Виходячи з наведених вище міркувань, можна визначити основні компоненти системи генерування спеціалізованих процесорів, яка відповідає сформованим в роботі [13] вимогам. Цими компонентами є:

1. система проектування програмних моделей процесорів з мови високого рівня;
2. генератор на основі бібліотек програмних моделей процесорів, представлених на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС;
3. генератор на основі бібліотек програмних моделей процесорних компонент;
4. блок аналізу коду програми.

Структуру такої системи генерування наведено на рис. 1. В наведеній структурі генератор G_{LPPM} на основі бібліотек програмних моделей процесорів містить множину бібліотек $LPPM_i$, $i = \overline{1..n}$, де n – кількість бібліотек, а кожна i -та бібліотека складається з множини програмних моделей процесорів PPM_j , $j = \overline{1..z}$, де z – кількість програмних моделей процесорів в бібліотеці, причому $z = k$ при $i = 1$; $z = l$ при $i = n$, тобто:

$$\begin{aligned} G_{LPPM} &= \left\{ \{LPPM_i, i = \overline{1..n}\}, SEM \right\}, \\ LPPM_i &= \{PPM_j, j = \overline{1..z}\}, \end{aligned} \quad (2)$$

де SEM – засоби пошуку і вибірки відповідної вхідним конфігураційним кодам програмної моделі процесора з бібліотеки. Кожна з програмних моделей PPM_j представлена у вигляді файлу конфігурації ПЛІС реконфігуровного середовища СККС.

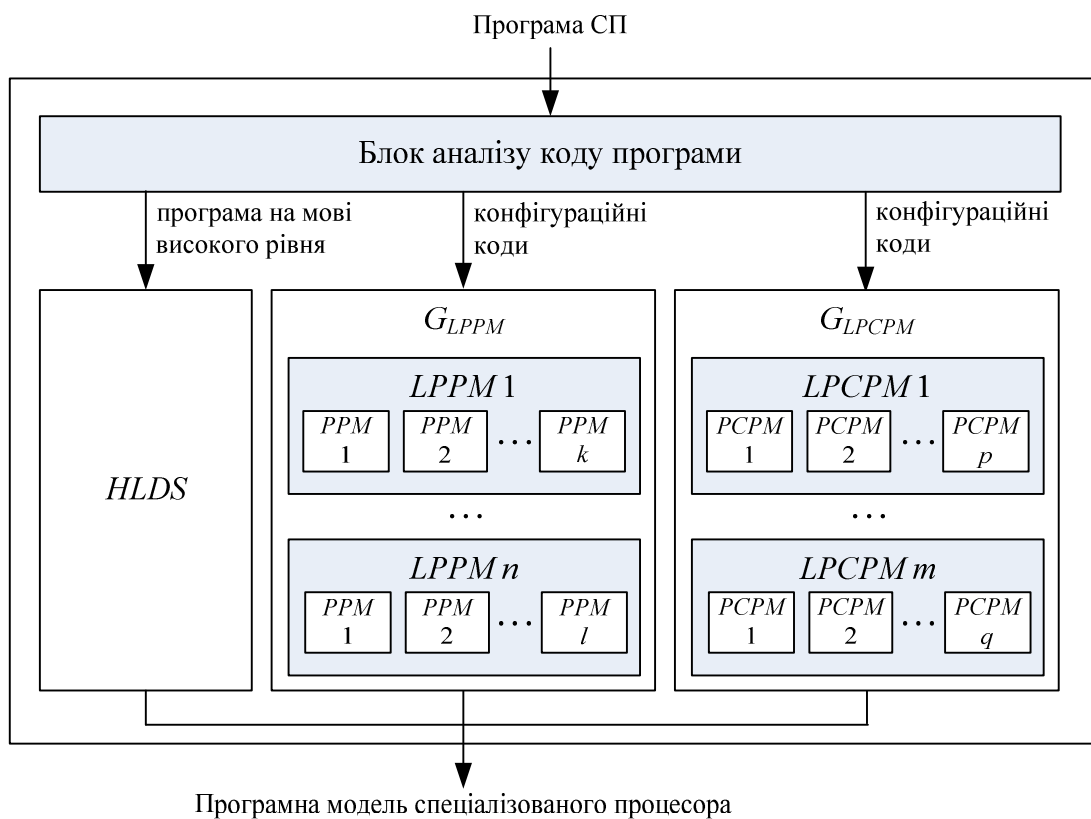


Рисунок 1 – Структура засобів генерування програмних моделей спеціалізованих процесорів в самоконфігуровній комп'ютерній системі

Аналогічно, генератор G_{LPCPM} на основі бібліотек програмних моделей процесорних компонент містить множину бібліотек $LPCPM_r$, $r = \overline{1..m}$, де m – кількість бібліотек, а кожна r -та бібліотека складається з множини програмних моделей процесорних компонент $PCPM_s$, $s = \overline{1..w}$, де w – кількість програмних моделей процесорних компонент в бібліотеці, причому $w = p$ при $r = 1$; $w = q$ при $r = m$, тобто:

$$\begin{aligned} G_{LPCPM} &= \left\{ \{LPCPM_r, r = \overline{1..m}\}, SETLGM \right\}, \\ LPCPM_r &= \{PCPM_s, s = \overline{1..w}\}, \end{aligned} \quad (3)$$

де *SETLGM* – засоби пошуку і вибірки відповідних вхідним конфігураційним кодам програмних моделей процесорних компонент з бібліотеки та генерації файлів опису верхнього рівня процесора. Кожна з програмних моделей процесорних компонент $PCPM_s$ представлена мовою опису апаратних засобів.

Методика функціонування системи автоматичного генерування програмних моделей спеціалізованих процесорів

Запропонована система автоматичного генерування функціонує під керуванням блоку аналізу коду. Цей блок визначає, котра з компонент системи генерування буде генерувати програмну модель. Алгоритм функціонування блоку аналізу коду можна описати у вигляді блок-схеми, представленої на рис. 2.



Рисунок 2 – Блок-схема алгоритму роботи блоку аналізу коду

Згідно з запропонованим алгоритмом, команди вибираються з програми СП по черзі. На початку роботи системи вказівник, який вказує поточну позицію, встановлюється на першу команду програми СП. Після вибірки команди виконується її ідентифікація, тобто визначення, чи належить дана команда до команд генераторів на основі бібліотек. В позитивному випадку виконується декодування команди, її трансляція в конфігураційний код та передача до відповідного генератора. В іншому випадку вважається, що дана команда представлена мовою програмування високого рівня, тому вона передається на входи системи автоматизованого високорівневого проектування програмних моделей процесорів. Після закін-

чення опрацювання поточної команди вказівник переводиться на наступну команду програми СП. Таким чином опрацюються всі команди програми СП.

Висновки

На основі аналізу особливостей функціонування засобів високорівневого проектування програмних моделей процесорів запропоновано структуру і розроблено принципи функціонування та алгоритм роботи системи генерування, до складу якої включено декілька типів засобів, а саме – генератори на основі бібліотек програмних моделей процесорів та процесорних компонент та систему автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня. Застосування цих засобів в системі генерування при побудові самоконфігуровної комп'ютерної системи забезпечує її відповідність вимогам в частині функціональної повноти, в частині технічних характеристик моделей процесорів, та в частині архітектури моделей процесорів.

Список літератури

1. Keating M. «Reuse Methodology Manual for System-On-a-Chip Design» / M. Keating, P. Bricaud, Kluwer Academic Publishers, 1999, - 224 p.
2. IEEE, Standard VHDL Language Reference Manual. Standard 1076-1993, New York, NY: IEEE, 1993.
3. IEEE, Standard Verilog Hardware Description Language Reference Manual. Standard 1364-1995, New York, NY: IEEE, 1996
4. DeHon A. Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation / S.Nauk, A.DeHon // Morgan Kaufmann, 2008. – 944 p.
5. Мельник В. Самоконфігуровні апаратні прискорювачі обчислень в комп'ютерах / В. Мельник, З. Сарайрех // Вісник Національного університету „Львівська політехніка” «Комп'ютерні системи та мережі». - №688, Львів, 2010. -С.163 – 171.
6. Мельник В. Метод самоконфігурування апаратного прискорювача / Віктор Мельник, Зіад Сарайрех // Матеріали V Міжнародної конференції молодих вчених “Комп'ютерні науки та інженерія 2011” (CSE-2011), 24-26 листопада 2011, Україна, Львів. - С. 126-127.
7. A Proven EDA Solutions Provider makes all the difference. [Електронний ресурс] / – Режим доступу: <http://www.aldec.com/en>.
8. Xilinx Core Generator. Xilinx Inc. [Електронний ресурс] / – Режим доступу: http://www.xilinx.com/ise/products/coregen_overview.pdf - 2005.
9. Мельник В. Організація бібліотек ядер стандартизованих та замовних комп'ютерних пристроїв для високопродуктивних реконфігурованих прискорювачів / А. Мельник, В. Мельник // IV Всеукраїнська науково-практична конференція «Комп'ютерні технології: наука і освіта», Україна, м.Луцьк, 9-11 жовтня 2009 р., Луцький інститут розвитку людини Університету «Україна», - С. 113-117.
10. Genest G. Programming an FPGA-based Super Computer Using a C-to-VHDL Compiler: DIME-C / G. Genest, R. Chamberlain, R. Bruce // Adaptive Hardware and Systems, 2007. AHS 2007. Second NASA/ESA Conference, 5-8 Aug. 2007. - P. 280 – 286.
11. Chameleon – the System-Level Design Solution. [Електронний ресурс] / – Режим доступу: http://intron-innovations.com/?p=sld_chame.
12. ANSI-C to VHDL Compiler. [Електронний ресурс] / – Режим доступу: <http://www.nallatech.com/FPGA-Development-Tools/dimetalk.html>.
13. Мельник В. А. Вимоги до системи генерування моделей процесорів самоконфігуровного апаратного прискорювача / В. Мельник, З. Сарайрех // Матеріали 5-ої Міжнародної науково-технічної конференції «Сучасні комп'ютерні системи та мережі: розробка та використання» (ASCN-2011). – Львів, 2011. – С. 255 – 258.

Стаття надійшла: 20.11.12.

Відомості про автора

Мельник Віктор Анатолійович, к.т.н., доцент, доцент кафедри «Безпека інформаційних технологій» Національного університету "Львівська політехніка"; 79013, м. Львів, вул. Бандери 12; Тел: (032) 235-83-23, (067) 7631345; електронна пошта: wiktor.melnyk@gmail.com.