

ДОСЛІДЖЕННЯ ХАРАКТЕРИСТИК ЗАСОБІВ ГЕНЕРУВАННЯ ПРОГРАМНИХ МОДЕЛЕЙ СПЕЦІАЛІЗОВАНИХ ПРОЦЕСОРІВ У САМОКОНФІГУРОВНІЙ КОМП'ЮТЕРНІЙ СИСТЕМІ

© Мельник В., 2012

Проаналізовано особливості засобів генерування програмних моделей обчислювальних пристроїв, порівняно характеристики цих засобів та запропоновано рекомендації щодо застосування цих засобів під час побудови самоконфігуровної комп'ютерної системи.

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

An overview and analysis of the features of the IP Cores generation facilities and a comparison of the characteristics of these facilities are presented in the paper. Recommendations in regard of their use in self-configurable computer system design are proposed.

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

Вступ

Протягом останнього десятиліття відбулися значні зміни у галузі проектування та мікроелектронного виробництва комп'ютерних засобів. Було розроблено технологію проектування програмних моделей обчислювальних пристроїв [1], яка описує процес розроблення архітектури обчислювальних пристроїв на рівні міжрегістрових передач з використанням мов опису апаратних засобів (*VHDL* [2], *Verilog* [3] та ін.) та їх реалізації у вигляді надвеликих інтегральних схем на базі програмованих логічних інтегральних схем (ПЛІС). Поширення ПЛІС сприяло відновленню інтересу до розроблення високопродуктивних апаратних прискорювачів, що функціонують у взаємодії з універсальними, зокрема й персональними, комп'ютерами. Апаратні прискорювачі, побудовані на базі ПЛІС (їх ще називають реконфігуровними прискорювачами [4]), мають істотні переваги порівняно з традиційними, а саме – вони можуть бути багаторазово перепрограмовані та мають значно нижчу вартість, ніж прискорювачі на базі замовних спеціалізованих надвеликих інтегральних схем (НВІС) у випадку несерійного виробництва.

Комп'ютерні системи, в яких універсальні комп'ютери об'єднані з реконфігуровним середовищем на основі мікросхем програмованої логіки, називають реконфігуровними комп'ютерними системами. Об'єднання універсального комп'ютера з реконфігуровним середовищем в єдину систему дає змогу виконувати обчислювальні завдання значно швидше, що забезпечується створенням в цьому середовищі спеціалізованих процесорів для апаратного виконання обчислювально найскладніших частин завдань.

Поряд з високою продуктивністю, яка забезпечується реконфігуровними комп'ютерними системами (РККС) на основі універсальних процесорів та реконфігуровних прискорювачів, є й деякі проблеми, пов'язані з їх застосуванням. Це зокрема їхня недостатня «динамічність» – перед тим, як виконувати обчислювальне завдання в РККС, її необхідно сконфігурувати, тобто синтезувати в її реконфігуровному середовищі спеціалізований процесор, який реалізує цей алгоритм.

Аналіз останніх досліджень

У роботі [5] запропоновано новий клас прискорювачів обчислень в комп'ютерах – самоконфігуровні прискорювачі та концепцію їх побудови. Цю концепцію покладено в основу побудови самоконфігуровних комп'ютерних систем (СККС), які позбавлені притаманних РККС проблем. СККС – це комп'ютерна система, яка містить універсальний комп'ютер, реконфігуровне середовище, побудоване на основі ПЛІС або інших типів інтегральних схем програмовної логіки, а також програмні засоби, які 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 процесором. У роботі [6] описано метод самоконфігурування на основі запропонованої концепції.

Постановка завдання

Однією з компонент СККС є система генерування, функцією якої є автоматичне створення програмних моделей спеціалізованих процесорів на основі високорівневого опису обчислювальних алгоритмів. 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 [7–12]. У роботі [13] сформовано вимоги до системи генерування програмних моделей спеціалізованих процесорів, на основі яких можна проаналізувати особливості наявних засобів генерування щодо ефективності їх застосування в СККС. Разом з тим, питання використання цих засобів для побудови системи генерування програмних моделей спеціалізованих процесорів в СККС, яка відповідала б висунутим до неї вимогам, досі не опрацьовувалось. Саме це питання і є предметом нашого наукового дослідження.

1. 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 [18]. Функцію системи генерування можна описати таким виразом:

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

де GS – система генерування; PR^{SCCS} – програма для апаратного виконання в реконфігурованому середовищі СККС, яка описує обчислювальні алгоритми $A_i, i = \overline{1...z}$; $PMSP$ – програмна модель спеціалізованого процесора для виконання обчислювальних алгоритмів $A_i, i = \overline{1...z}$; z – кількість обчислювальних алгоритмів.

2. Типи засобів генерування програмних моделей обчислювальних пристроїв

До засобів високорівневого проектування програмних моделей обчислювальних пристроїв, які пропонується використати для реалізації системи генерування, належать:

1. Генератори програмних моделей процесорів на основі бібліотек програмних моделей процесорів та процесорних компонент [13];
2. Генератори програмних моделей процесорів на основі конфігурованих програмних моделей процесорів [11, 12];
3. Системи автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня [8–10].

Проаналізуємо особливості застосування зазначених типів програмних засобів в системі генерування СККС відповідно до сформульованих в роботі [14] вимог, та визначимо, як задовольняють ці засоби такі вимоги в частині:

- 1) функціональної повноти системи генерування;
- 2) технічних характеристик генерованих системою генерування моделей процесорів;
- 3) архітектури генерованих системою генерування моделей процесорів.

3. Особливості застосування генераторів на основі бібліотек програмних моделей процесорів

Генератор на основі бібліотеки програмних моделей процесорів можна подати двома складовими: власне бібліотекою, реалізованою, наприклад, як база даних, та засобами пошуку програмної моделі процесора в цій бібліотеці. Пошук програмної моделі в бібліотеці виконується відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора [13]. Бібліотека містить готові програмні моделі процесорів, описані мовою опису апаратних засобів. Вона може бути як типовою і містити моделі процесорів для виконання наборів стандартизованих прикладних алгоритмів та найчастіше використовуваних обчислювальних операцій, так і виконаною на замовлення і орієнтованою на задачі певної галузі [15].

Отже, можна зробити такі висновки щодо відповідності генераторів на основі бібліотек програмних моделей процесорів поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі бібліотек програмних моделей процесорів внаслідок обмеженості бібліотек не може згенерувати модель спеціалізованого процесора для виконання довільного обчислювального алгоритму, тому не може бути функціонально повною.

2. Оскільки програмні моделі процесорів, що входять до складу бібліотеки, створюються заздалегідь, існує можливість врахування в цих моделях на етапі їх створення особливостей архітектури цільової ПЛІС, зокрема ПЛІС реконфігурованого середовища СККС, кількості виділеного в ПЛІС місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним комп'ютером та реконфігурованим середовищем. Це дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Необхідно зазначити, що бібліотека може вмішувати не програмні моделі процесорів, а файли конфігурації цільової ПЛІС, отримані в результаті логічного синтезу цих процесорів. Це дасть змогу скоротити та спростити процес компіляції програми в СККС, уникнувши етапу логічного синтезу процесорів.

Час генерування за допомогою генераторів на основі бібліотек програмних моделей процесорів можна визначити за виразом:

$$t_{gen}(G_{LPPM}) = t_{search} + t_{out}, \quad (2)$$

де t_{search} – час пошуку програмної моделі процесора в бібліотеці; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

4. Особливості застосування генераторів на основі бібліотек програмних моделей процесорних компонент

Генератор на основі бібліотеки програмних моделей процесорних компонент можна подати трьома складовими: бібліотекою, засобами пошуку програмних моделей відповідних компонент в цій бібліотеці і засобами компонування програмних моделей процесорів з компонент та генерування файлів опису верхнього рівня. Пошук програмних моделей у бібліотеці та компонування виконуються відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора.

Принциповими перевагами застосування бібліотеки програмних моделей процесорних компонент над бібліотекою програмних моделей процесорів є розширення функціональності генератора та зменшення об'єму бібліотеки [13]. Це однак не позбавляє цих засобів їх природної обмеженості в частині функціональної повноти. Можна зробити такі висновки щодо відповідності генераторів на основі бібліотек програмних моделей процесорних компонент поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі бібліотек програмних моделей процесорних компонент не може бути функціонально повною.

2. Існує можливість врахування в моделях процесорних компонент на етапі їх створення особливостей архітектури ПЛІС реконфігуровного середовища СККС, кількості виділеного в ній місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним комп'ютером та реконфігуровним середовищем, що дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Час генерування за допомогою генераторів на основі бібліотек програмних моделей процесорних компонент можна визначити з виразу:

$$t_{gen}(G_{LPCPM}) = t_{search} + t_{comp+TLGen} + t_{out}, \quad (3)$$

де t_{search} – час пошуку програмних моделей компонент процесора в бібліотеці; $t_{comp+TLGen}$ – час компонування моделі процесора та генерування файлів опису верхнього рівня; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

5. Особливості застосування генераторів на основі конфігуровних програмних моделей процесорів

Генератор на основі конфігуровних програмних моделей процесорів можна подати двома складовими: множиною конфігуровних програмних моделей, описаних мовою опису апаратних засобів, та засобами їх конфігурування. Конфігурування виконується відповідно до отриманих ззовні конфігураційних кодів, що задають параметри процесора [11, 12].

Аналогічно до генераторів на основі бібліотек засобам генерування на основі конфігуровних програмних моделей процесорів також притаманна природна властивість функціональної обмеженості. Це пояснюється тим, що розробити таку конфігуровну програмну модель спеціалізованого процесора, яку можна сконфігурувати для виконання довільного обчислювального алгоритму, зберігаючи при цьому апаратну орієнтацію архітектури цього процесора, є нереальною задачею.

На основі цього можна зробити такі висновки щодо відповідності генераторів на основі конфігуровних програмних моделей процесорів поданим у роботі [14] вимогам до системи генерування:

1. Система генерування на основі конфігуровних програмних моделей процесорів не може бути функціонально повною.

2. Існує можливість врахування в конфігуровних програмних моделях процесорів на етапі їх створення особливостей архітектури ПЛІС реконфігуровного середовища СККС, кількості виділеного в ній місця під реалізацію процесорів, а також можливостей інтерфейсу між універсальним процесором та реконфігуровним середовищем, що дасть змогу досягти максимально можливої на обладнанні даної ПЛІС продуктивності процесора.

Необхідно зазначити, що можливості мов опису апаратних засобів в частині створення конфігуровних програмних моделей є достатніми для розроблення простих пристроїв, однак для складніших пристроїв, зокрема, спеціалізованих процесорів, цих можливостей часто буває замало [16]. Саме це й стало причиною розроблення інших, ефективніших методів конфігурування, зокрема й методу на основі бібліотеки процесорних компонент.

Час генерування за допомогою генераторів на основі конфігуровних програмних моделей процесорів можна визначити за виразом:

$$t_{gen}(G_{CPM}) = t_{search} + t_{conf} + t_{out}, \quad (4)$$

де t_{search} – час пошуку конфігуровної програмної моделі в множині цих моделей; t_{conf} – час конфігурування моделі; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

6. Особливості застосування систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня

Сьогодні багато світових фірм працюють у напрямі створення систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня та пропонують свої продукти, зокрема *Celoxica*, *Impulse*, *Nallatech*, *Synfora*, *Synopsys*, *Intron*, *Mentor Graphics* [8–10]. Принциповою відмінністю таких систем від генераторів іншого типу є те, що вхідними даними для них є не конфігураційні коди, що задають параметри процесора, а високорівневий алгоритмічний опис

роботи процесора у вигляді програми мовою високого рівня. Більшість засобів використовують для цього модифікований варіант мови C, наприклад, *Mentor Graphics* використовує *IEEE standard System C*, *Celoxica – Handle C*, *Nallatech – DIME C*, *Impulse – Impulse-C*, *Mitrionics – Mitrion-C*.

У зв'язку з тим, що в СККС високорівневий алгоритмічний опис роботи процесора формується системою розподілу обчислювального навантаження, застосування систем автоматизованого високорівневого проектування програмних моделей процесорів в СККС ставить умови використання стандартної мови *ANSI C* як вхідної. З відомих систем цій вимозі сьогодні відповідають системи *Chameleon* фірми *Intron*, *CHiMPS* фірми *Xilinx* та *Carte* фірми *SRC*.

Архітектурні підходи, покладені в основу побудови різних систем високорівневого автоматизованого проектування, значною мірою визначають продуктивність генерованих ними спеціалізованих процесорів. При цьому розробники деяких систем наголошують на високому ступені адаптованості кінцевого процесора до алгоритму, що дає змогу значно прискорити виконання задачі та коригувати співвідношення «продуктивність – затрати обладнання», при цьому не висуваючи вимог до кодів вхідних програм. Розробники інших систем ставлять до написання коду вхідної програми ряд вимог, виконання яких дають змогу досягти вищих показників продуктивності. Засоби ще інших виробників орієнтовані на роботу з алгоритмами визначених галузей (цифрова обробка сигналів, криптографія тощо) і забезпечують високу продуктивність кінцевих процесорів для виконання завдань саме у цих галузях.

На основі сказаного вище можна зробити такі висновки щодо відповідності систем автоматизованого високорівневого проектування програмних моделей процесорів сформованим в роботі [14] вимогам до системи генерування:

1. Системи автоматизованого високорівневого проектування програмних моделей процесорів можуть згенерувати модель спеціалізованого процесора для виконання довільного обчислювального алгоритму, що описується мовою програмування високого рівня, тому є функціонально повними.

2. Можливість врахування особливостей архітектури ПЛІС реконфігурованого середовища СККС у генерованих програмних моделях процесорів залежить від особливостей конкретної системи автоматизованого високорівневого проектування. За інформацією авторів, таку можливість сьогодні надає система *CHiMPS* фірми *Xilinx*, вихідний *VHDL*-код якої оптимізований для ПЛІС цієї фірми; засоби інших фірм такої можливості не надають. Разом з тим, жодна з відомих систем проектування не враховує особливостей інтерфейсу між універсальним комп'ютером та ПЛІС.

3. Продуктивність створених спеціалізованих процесорів залежить від архітектурних підходів, покладених в основу побудови конкретної системи автоматизованого високорівневого проектування. Зрозуміло, що за умови застосування вдалих архітектурних рішень машинна оптимізація архітектури обчислювального пристрою з метою досягнення ним максимальної продуктивності потенційно є ефективнішою ніж виконана інженером оптимізація.

4. Деякі засоби автоматизованого високорівневого проектування дають можливість коригувати співвідношення «продуктивність – затрати обладнання». Ефективне використання обладнання ПЛІС реконфігурованого середовища СККС, що є однією з вимог до системи генерування, викликає необхідність розроблення та реалізації методу взаємодії системи генерування на основі засобів автоматизованого високорівневого проектування з засобами логічного синтезу процесорів. Цей метод повинен забезпечити можливість налаштування системи генерування на таке співвідношення «продуктивність – затрати обладнання», у результаті якого створений спеціалізований процесор займатиме максимум з виділеного в ПЛІС місця під його реалізацію.

Відповідно до описаної в праці [17] послідовності виконання етапів автоматичного синтезу спеціалізованих процесорів від алгоритму до рівня міжрегістрових передач час генерування за допомогою систем автоматизованого високорівневого проектування програмних моделей процесорів можна визначити виразом:

$$t_{gen}(HLDS) = t_{IMCT} + t_{ASA} + t_{SPOS} + t_{out}, \quad (5)$$

де t_{IMCT} – час перетворення високорівневого опису вхідного алгоритму в проміжний код, який надає можливість подальшого опрацювання структури алгоритму; t_{ASA} – час визначення та аналізу структури алгоритму; t_{SPOS} – час виконання оптимізаційного синтезу схеми спеціалізованого процесора; t_{out} – час подачі програмної моделі процесора на вихід системи генерування.

7. Порівняльний аналіз засобів генерування програмних моделей спеціалізованих процесорів у системі генерування

Аналізуючи описані виразами (2)–(5) часові характеристики розглянутих засобів генерування, бачимо, що мінімальним часом виконання генерування характеризуються генератори на основі бібліотек програмних моделей процесорів, оскільки їхній час генерування $t_{gen}(G_{LPPM})$ визначається як сума двох величин: часу t_{search} пошуку в бібліотеці та часу t_{out} подачі файлів програмної моделі процесора на вихід. Дещо більшим є час роботи генераторів на основі бібліотек програмних моделей процесорних компонент $t_{gen}(G_{LPCPM})$, оскільки, крім зазначених складових, він містить час $t_{comp+TLGen}$ компонування моделі процесора та генерування файлів опису верхнього рівня.

Час генерування за допомогою генераторів на основі конфігурованих програмних моделей процесорів $t_{gen}(G_{CPM})$ є співмірним з часом генерування за допомогою генераторів на основі бібліотек програмних моделей процесорних компонент $t_{gen}(G_{LPCPM})$, оскільки він по суті, відрізняється лише однією складовою, а саме – замість часу $t_{comp+TLGen}$ компонування моделі процесора та генерування файлів опису верхнього рівня містить час t_{conf} виконання конфігурування моделі. Зрозуміло, що тривалість цих часів є одного порядку.

На відміну від розглянутих, час генерування за допомогою систем автоматизованого високорівневого проектування програмних моделей процесорів $t_{gen}(HLDS)$ визначається зовсім іншими складовими, за винятком часу t_{out} подачі програмної моделі процесора на вихід системи генерування. Ці складові відображають тривалість виконання завдань перетворення високорівневого опису вхідного алгоритму в проміжний код, визначення та аналізу структури алгоритму та виконання оптимізаційного синтезу схеми спеціалізованого процесора. Зрозуміло, що ці завдання по своїй суті є значно складнішими від завдання пошуку в бібліотеці чи генерування файлів опису верхнього рівня. Очевидно, що час $t_{gen}(HLDS)$ є більшим, ніж час роботи засобів генерування інших типів, що можна записати виразом:

$$t_{gen}(HLDS) > t_{gen}(G_{LPCPM}) \approx t_{gen}(G_{CPM}) > t_{gen}(G_{LPPM}). \quad (6)$$

На основі проведеного вище аналізу особливостей застосування засобів генерування програмних моделей процесорів сформуємо таблицю відповідності цих засобів вимогам до системи генерування. Також врахуємо значення часу роботи цих засобів, присвоївши їм швидкісні індекси від найшвидшого (1) до найповільнішого (3) згідно з виразом (6) (таблиця).

Відповідність засобів генерування програмних моделей процесорів вимогам до системи генерування

Вимога до системи генерування	Тип засобів генерування програмних моделей процесорів*			
	1	2	3	4
Функціональна повнота	–	–	–	+
Врахування особливостей архітектури ПЛІС	+	+	+	–/+**
Врахування особливостей інтерфейсу між універсальним комп'ютером та ПЛІС	+	+	+	–
Максимальна на даному типі ПЛІС продуктивність процесора	+	+	+	–/+**
Максимальна ефективність використання обладнання ПЛІС	+	+	+	–/+***
Індекс часу виконання генерування	1	2	2	3

* 1 – на основі бібліотек програмних моделей процесорів.

* 2 – на основі бібліотек програмних моделей процесорних компонент.

* 3 – на основі конфігурованих програмних моделей процесорів.

* 4 – системи автоматизованого високорівневого проектування програмних моделей процесорів.

** – позитивно тільки для засобів *CHiMPS* фірми *Xilinx*.

*** – позитивно тільки для засобів, які дають можливість коригування співвідношення «продуктивність – затрати обладнання».

Проаналізуємо дані таблиці.

Система генерування СККС повинна бути функціонально повною. З даних таблиці видно, що 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 мовою 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, з даних таблиці видно, що цей тип засобів генерування переважно 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 для побудови 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 можна отримати 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 за ефективністю конфігурування;

4. Оскільки функціональні характеристики процесорів для генераторів на основі бібліотек програмних моделей процесорних компонент задаються не програмою, а конфігураційними кодами, взаємодію такого генератора із системою розподілу обчислювального навантаження необхідно організувати шляхом прямої передачі команд від користувача до системи генерування.

Висновки

1. Проаналізовано засоби генерування програмних моделей процесорів на основі бібліотек програмних моделей процесорів та процесорних компонент, на основі конфігурованих програмних моделей процесорів, а також систем автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня в контексті їх застосування в СККС для побудови системи генерування. Досліджено відповідність цих засобів вимогам до системи генерування в частині функціональної повноти, в частині технічних характеристик моделей процесорів та в частині архітектури моделей процесорів.

2. Встановлено, що єдиним типом засобів генерування програмних моделей процесорів, які характеризуються функціональною повнотою, є системи автоматизованого високорівневого проектування, а також те, що цей тип засобів є єдиним, в якому функціональні характеристики генерованого процесора задаються описом алгоритму його роботи мовою програмування високого рівня, що є необхідною умовою застосування в СККС.

3. Встановлено, що вимогам в частині архітектури та в частині технічних характеристик процесорів відповідають генератори на основі бібліотек програмних моделей процесорів та процесорних компонент, а також на основі конфігурованих програмних моделей процесорів. При цьому програмні моделі процесорів в бібліотеках доцільно представляти на рівні файлів конфігурації ПЛІС реконфігурованого середовища СККС, що дасть змогу скоротити та спростити процес компіляції програми в СККС, уникнувши етапу логічного синтезу процесорів.

4. Досліджено часові характеристики засобів генерування програмних моделей процесорів, в результаті якого встановлено, що мінімальним часом генерування характеризуються генератори на основі бібліотек програмних моделей процесорів, дещо більшим – генератори на основі бібліотек програмних моделей процесорних компонент та на основі конфігурованих програмних моделей процесорів, і найбільшим – системи автоматизованого високорівневого проектування програмних моделей процесорів з мови високого рівня.

5. Встановлено, що використання генераторів на основі конфігурованих програмних моделей процесорів є недоцільним з огляду на те, що вони не мають жодних переваг над генераторами на основі бібліотек процесорних компонент, але поступаються їм за ефективністю конфігурування.

1. Мельник А., Мельник В. *Технологія проектування ядер комп'ютерних пристроїв* // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні системи та мережі". – 2002. – № 463. – С. 3–9.
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
4. А.О. Мельник, В.А. Мельник, Сарайрех З. Використання реконфігурованих прискорювачів для підвищення продуктивності персональних комп'ютерів // Науковий вісник Чернівецького ун-ту "Комп'ютерні системи та компоненти". – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. – Т. 1. – Вип. 1. – С. 20–25.
5. Мельник В., Сарайрех З. Самоконфігуровні апаратні прискорювачі обчислень в комп'ютерах // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні системи та мережі". – 2010. – № 688. – С.163–171.
6. Мельник В., Сарайрех З. Метод самоконфігурування апаратного прискорювача // *Мат. V Міжн. конф. молодих вчених "Комп'ютерні науки та інженерія 2011" (CSE-2011)*, 24–26 листопада 2011, Україна. – Львів. – С. 126–127.
7. Melnyk A. *IP Cores Generators in SoC Design* / A. Melnyk, W. Melnyk // *Proceedingd of the 5th international Conference for Students and Young Scientists „Telecommunication in XXI Century“*, Poland, Wólka Milanowska, 24–26 November 2005. – P. 23–28.
8. Мельник А.О., Сало А.М., Клименко В., Цигилик Л., Юрчук А. *ХАМЕЛЕОН – система високорівневого синтезу спеціалізованих процесорів* // *Наук.-техн. журнал Нац. аерокосм. ун-ту ім. М.С. Жуковського «Харківський авіаційний інститут»*. – 2009. – № 5. – С. 189–195 с.
9. *Accelerate Software Algorithms on FPGAs*. [Електронний ресурс] / – Режим доступу: <http://www.nallatech.com>
10. *ANSI-C to VHDL Compiler*. [Електронний ресурс] / – Режим доступу: <http://www.nallatech.com/FPGA-Development-Tools/dimetalk.html>
11. *Xilinx Core Generator*. Xilinx Inc [Електронний ресурс] / – Режим доступу: http://www.xilinx.com/ise/products/coregen_overview.pdf – 2005.
12. *A Proven EDA Solutions Provider makes all the difference*. [Електронний ресурс] – Режим доступу: <http://www.aldec.com/en/>
13. Мельник А. Організація бібліотек ядер стандартизованих та замовних комп'ютерних пристроїв для високопродуктивних реконфігурованих прискорювачів / А. Мельник, В. Мельник // *IV Всеукр. наук.-практ. конф. «Комп'ютерні технології: наука і освіта»*, Україна, м. Луцьк, 9–11 жовтня 2009 р., Луцький інститут розвитку людини Університету «Україна». – С. 113–117.
14. Мельник В., Сарайрех З. Вимоги до системи генерування моделей процесорів самоконфігурованого апаратного прискорювача // *Мат. 5-ої Міжнар. наук.-техн. конф. «Сучасні комп'ютерні системи та мережі: розробка та використання» (ASCN-2011)*, 29 вересня – 01 жовтня 2011, Україна, Львів. – С. 255–258.
15. Melnyk V. *Set of Symmetric Block Ciphering Soft-Cores* / V. Melnyk // *Proceedings of the VII-th International Conference CADSM 2003*, , Lviv-Slavsko., February 18–22, 2003. – P. 190–193.
16. Мельник В.А. Методи конфігурування моделей спеціалізованих процесорів / В.А. Мельник, Мохаммад Аль Хабабсах. // *Мат. IV Всеукр. наук.-практ. конф. «Комп'ютерні технології: наука і освіта»*, Україна, Луцьк, 9–11 жовтня 2009. – С. 121–125.
17. Мельник А.О. Автоматичний синтез спеціалізованих процесорів від алгоритму до рівня міжрегістрових передач / А.О. Мельник // *Мат. пробл.-наук. міжгал. Конф. «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління (ПНМК – 2010)»*, Бучач, 2010. – С. 27–29.
18. Мельник В., Степанов В., Сарайрех З. Система розподілу обчислювального навантаження між хост-комп'ютером та самоконфігурованим прискорювачем // *Наук. вісник Чернівецького університету імені Юрія Федьковича "Комп'ютерні системи та компоненти"*. – Чернівці, 2012. – Т. 3. – Вип. 1. – С. 6–16.