Информационные системы и технологии

УДК 004.272.26, 004.274

І.А. Клименко, канд. техн. наук
Національний технічний університет України «Київський політехнічний інститут»,
вул. Політехнічна, 16, корпус 12, м. Київ, 03056, Україна.

Вдосконалення засобів синхронізації процесів в мультипроцесорних системах-на-кристалі

Розроблені та досліджені засоби автоматичного управління паралельними обчисленнями та синхронізації, що дозволяють вдосконалити відомі технології проектування, які пропонують відомі виробники ПЛІС, та підвищити ефективність синхронізації процесів в мультипроцесорних системах-на-кристалі. Бібл. 6, рис. 6.

Ключові слова: *мультипроцесорна система-на-кристалі; автоматична синхронізація; обчислення керовані потоком даних; ПЛІС.*

Вступ

Розповсюджений спосіб побудови систем-на-кристалі базується на застосуванні уніфікованої технології проектування (*design flow*), що пропонують виробники ПЛІС [2, 5, 4]. Побудова мультипроцесорних систем на ПЛІС у такий спосіб має обмеження, які пов'язані з застосуванням жорстких архітектурних рішень та програмною реалізацією функціональних процесів. Це обмежує можливості реконфігурації архітектури, вдосконалення функціональних процесів і негативно впливає на швидкодію обчислень. Особливо критичні ці обмеження стають в процесі проектування вузькоспеціалізованого устаткування, якщо на його роботу накладаються часові обмеження, зокрема в режимі реального часу. В таких системах важлива ефективна реалізація обміну даними, синхронізації процесів, обробки зовнішніх впливів. Вищесказане обґрунтовує актуальність та доцільність виконаних досліджень і обумовлює необхідність вдосконалення існуючих технологій проектування мультипроцесорних систем на кристалі.

Огляд відомих рішень

Дослідження ефективності синхронізації процесів в мультипроцесорних системах-на-кристалі, побудованих засобами доступними в межах уніфікованої технології проектування від виробників ПЛІС, в ряді робіт виконані із загальним висновком, щодо необхідності їх вдосконалення. В роботі [2] показано, що засоби *Mutex Core Altera* для застосування в мультипроцесорних середовищах, характеризуються значними витратами часу, які є критичними за великої кількості процесорних ядер. Для синхронізації процесів в системах на базі процесорних ядер *MicroBlaze Xilinx* в [5] досліджена можливість застосування операційної системи реального часу *Xilkernel Xilinx* та вбудованих механізмів *POSIX threads*, з висновком, що ці засоби не ефективні для мультипроцесорних реалізацій, виявлені проблеми застосування ядра *Xilinx Mutex*, пов'язані з його орієнтацію на монопроцесорні системи-на-кристалі. В якості альтернативних рішень розглянуті апаратне ядро *Hardware Mutex для рішення проблем синхронізації процесів,* яке підключається до загальної шини, та мультишинна архітектура з синхронізацією на базі ядер *XilinxMutex*. Таким чином, всі засоби синхронізації в межах уніфікованої технології проектування від виробників ПЛІС реалізовані на програмному рівні, що негативно впливає на швидкодію обчислень. Спеціально розроблені архітектури та вдосконалення засобів синхронізації, альтернативні уніфікованому підходу, часто є специфічними до вирішуваних задач, що призводить до обмеженого використання систем, складнощів їх проектування та програмування.

Постановка задачі

Уніфікована технологія проектування від компанії Altera на базі інструментального середовища SOPC Builder (System on a Programmable Chip Builder), забезпечує автоматичну генерацію систем-на кристалі [4]. Основними функціональними елементами є модулі процесорних ядер Nios II, шини Avalon та набір HDL-модулів (*Hardware Description Language*): контролери пам'яті, інтерфейси, периферійне устаткування, тощо. Основна конфігурація системи – загальна шина з обміном даними через загальну пам'ять.

Для рішення задачі синхронізації в середовищі *SOPC Builder*, використовується механізм *Mutex*, який є суто програмною реалізацією й потребує попередньої підготовки програмного коду. Альтернативним засобом від компанії *Altera* є апаратні блоки *MailBox*, які застосовуються для обміну інформацією між кожною парою процесорів. Але для забезпечення атомарності операції зміни вмісту *MailBox* застосовується механізм Mutex, що залишає значну програмну компоненту реалізації. Для реалізації *MailBox* використовується коштовний ресурс внутрішньої пам'яті ПЛІС. За збільшення кількості процесорів стрімко зростає кількість блоків *MailBox*. В системі с шістьма процесорами під час обміну масивами даних з максимальним об'ємом 2 Кб реалізація *MailBox* потребує 20% внутрішньої пам'яті ПЛІС.

Проблемою, що впливає на швидкодію синхронізації процесів є застосування традиційних механізмів синхронізації процесів під час доступу до загальних ресурсів – прапорцях, які встановлюються загальному адресному просторі [1], непродуктивно збільшуючи кількість звернень до загальної шини. Кількість таких звернень до загальної шини, може коливатися в широких межах – від двох звернень, для перевірки та встановлення прапорця, до невизначеної кількості звернень, в стані очікування дозволу передачі даних.

Отже стає задача, вдосконалення відомих засобів синхронізації потоків, шляхом зменшення кількості звернень до загальної шини, запобігання стану невизначеного часу очікування, зберігання ресурсів внутрішньої пам'яті ПЛІС, збільшення швидкодії синхронізації процесів за збільшення кількості процесорних ядер.

Вдосконалення архітектури

Для вдосконалення базової архітектури загальна шина послідовності проектування *SOPC Builder*, в статті запропонована використання моделі обчислень керованої потоком даних. Для реалізації такого способу управління обчисленнями до складу системи, на відміну від базової, додано апаратний пристрій автоматичного розподілу задач і синхронізації (ПРС), який є практичною реалізацією на ПЛІС відомого методу автоматичного динамічного розпаралелювання обчислень на рівні програмних модулів і програм у паралельних системах заснованого на механізмі формування заявок під управлінням дескрипторів [1]. Централізовані засоби управління перетворюють вихідний обчислювальний алгоритм на потік акторів та вхідних дескрипторів. ПРС забезпечує автоматичне розпаралелювання та розподіл задач між процесорними ядрами.

Пристрій розподілу задач реалізований на апаратному рівні у вигляді окремого модуля, який підключається до загальної шини і виконує свої функції автоматично та автономно від процесу функціонування програмного шару операційної системи. Готові заявки, які автоматично формуються в загальному середовищі формування заявок, зберігаються в локальній буферній пам'яті блоку і виконуються вільними процесорами. Узагальнена структура блоку, та його функціональні зв'язки з основними складовими системи-на-кристалі представлено на рис.1. В загальному адресному просторі ПРС представлений адресами пари регістрів адреси та даних. Загальна кількість звернень до загальної шини, що відбувається під час рішення деякої задачі дорівнює $M_{ЗШ} = n + d_{IN} + n_{IN}(1 + d_i + 2)$, де n_{IN} та d_{IN} – кількість акторів та вхідних дескрипторів відповідно, d_i – кількість вхідних дескрипторів кожного актора $N_i \mid (i = \overline{1,n})$.

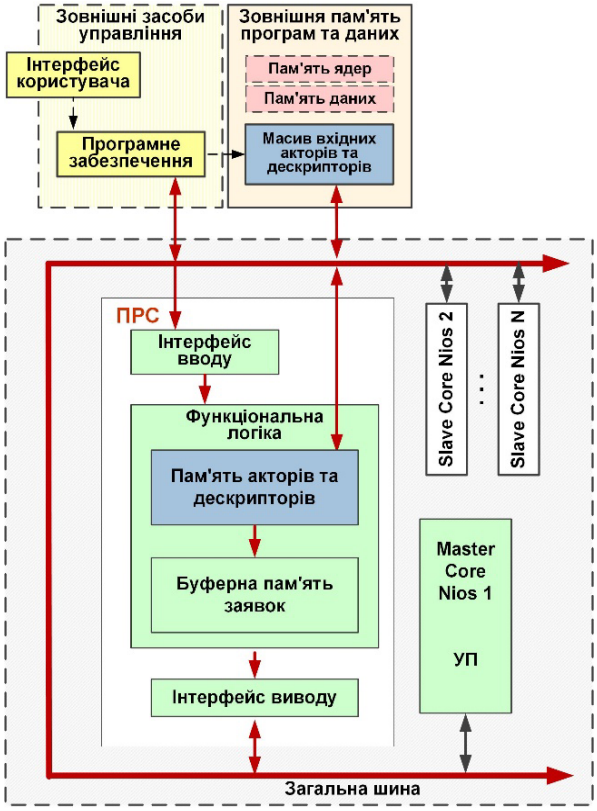


Рис.1. Структура пристрою автоматичного розподілу задач та синхронізації у складі системи-на-кристалі

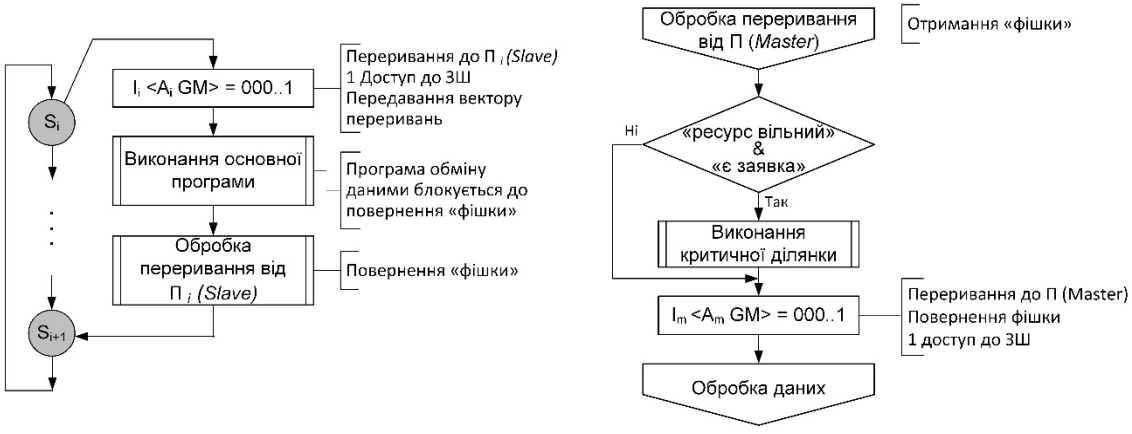
Модель задачі

В якості цільових задач в статті розглядаються задачі змішаного типу паралелізму [6]. При цьому вузли графа обчислювальної задачі є макрозадачами, які виконуються засобами процесорних ядер, що знаходяться в вузлах потокової системи.

В якості практичної задачі для дослідження та експериментів виконується задача галузі управління технічними системами – рішення системи лінійних алгебраїчних рівнянь від великої кількості змінних. В якості методу для реалізації на паралельних цифрових системах обраний метод Гауса. Якщо функції на які розгалужується задача є достатньо елементарні і обраховуються швидше ніж обмін даними виникають проблеми зі швидкодією навіть за достатньої кількості процесорів. Для приведення розмірності задачі до відповідності структури системи використовується спосіб блочного розпаралелювання алгоритму рішення СЛАУ методом Гауса, коли під блоком розуміють об'єднання декількох рядків в один блок [3]. Матриця розподіляється між процесорними ядрами по-блочно, відповідно до їх кількості.

Засоби автоматичної синхронізації

Для підвищення ефективності обміну даними в межах вдосконаленої архітектури системи-на-кристалі пропонується реалізація відомого способу автоматичної синхронізації описаному в роботі [1]. Спосіб базується на пересуванні «фішки», яка інтерпретується доступністю ресурсу S, через вершини S_i графа ($i = \overline{1,(N-1)}$), де N – кількість процесорів в системі. Якщо «фішка» знаходиться в вершині S_i і є вимога доступу до загального ресурсу від процесора P_i, виконується операція виділення ресурсу. Процесор отримує доступ до виконання критичної ділянки, після виконання якої повертає «фішку», звільнюючи ресурс S_i. Функції автоматичної синхронізації процесів покладаються на розроблений ПРС (рис. 1). Програма синхронізації виконується засобами управляючого процесору. Програмний режим опитування, обумовлює витрати продуктивності, пов'язані з безперервною перевіркою наявності «фішки» навіть за застосування комунікаційної пам'яті. За реалізації управління потоком даних в централізованій системі, коли підлеглі процесори рівноправні та виконують лише обробку даних, необхідно додатково здійснювати синхронізацію сигналів вимоги доступу до загального ресурсу і готовності заявки на виконання, інакше заявка може бути виконана іншим процесором, який раніш отримає фішку від управляючої програми. Окрім того реалізація відомого способу потребує наявності локальної пам'яті у складі кожного процесорного модуля, що не передбачено архітектурою засобів проектування SOPC Builder. З метою рішення означених вище проблем запропонована модифікація відомого способу автоматичної синхронізації для забезпечення ефективної синхронізації в рамках вдосконаленої архітектури. Алгоритм реалізації модифікованого способу зображений на рис. 2 а, б.



(а) **(б)**

Рис. 2. Модифікований спосіб автоматичної синхронізації: а) – алгоритм виконуваний управляючим процесором; б) – алгоритм виконуваний підлеглим процесором

Модифікований спосіб забезпечує синхронізацію управляючих сигналів вимоги доступу та наявності заявки та функціонує в режимі управління за каналами зв'язку, що

запобігає накладним витратам продуктивності процесорів. Обмін управляючими даними вимагає лише двох звернень до загальної шини (ЗШ), при цьому відсутні непродуктивні використання ресурсу процесорів в стані очікування. Реалізація модифікованого способу базується на застосуванні базової системи переривань процесора Nios II Altera.

Розробка архітектури та програмного забезпечення

Система-на-кристалі розроблена та досліджена на базі ПЛІС Cyclone II *EP2C35F672C6* компанії *Altera,* загальною логічною ємністю близько 33 216 логічних комірок та вбудованою пам'яттю об'ємом близько 59Кб. Використаний стандартний потік проектування SOPC Builder. Для реалізації обрані процесорні ядра типу Nios II/s (Small core size), які мають незначні вимоги до об'єму апаратних ресурсів, що становлять 4% логічного ресурсу кристалу, і високу продуктивність. Для кожного ядра виділені 10Кб загальної пам'яті із складу вбудованої пам'яті RAM. Для виконуваних програм виділено 256Кб зовнішньої пам'яті SDRAM. Система з одним процесорним ядром займає 27% логічного ресурсу мікросхеми та 20% вбудованої пам'яті (близько 12Кб). Пристрій автоматичного розподілу задач та синхронізації працює під управлінням Master-процесора підключеного до шини Avalon. Пристрій синтезований на мові Verilog, як апаратна надбудова процесорного ядра.

Найбільша проблема, що впливає на обчислювальну потужність досліджуваної системи це обмежений об'єм внутрішньої пам'яті кристалу, що не дозволяє розмістити на одному кристалі велику кількість процесорних ядер. За рахунок розміщення пам'яті процесорів і пам'яті проміжних даних розрахунків на зовнішній SRAM був значно зменшений задіяний процесорами об'єм вбудованої пам'яті. При цьому, зважаючи на можливість забезпечення при цьому достовірності результатів експериментів, а також з причини однакового підходу до всіх досліджуваних архітектур, ми нехтуємо негативною складовою часу, що вносить застосування зовнішньої пам'яті.

Під час здійснення розрахунків використані вбудовані стандартні апаратні засоби для обчислень з плаваючою крапкою (Floating Point Hardware, FPH). Виконані порівняльні дослідження в цьому напрямку показали збільшення швидкодії на порядок та підвищення точності результатів.

Моделювання та дослідження часових характеристик

Дослідження проводились для конфігурації систем-на-кристалі з кількістю процесорних ядер від трьох до шести. Порівняні дві архітектури – базова архітектура із загальною шиною, яка побудована засобами середовища SOPC Builder, та удосконалена архітектура на базі моделі керованої потоком даних. В першому випадку управління розподілом задач та синхронізація та здійснюється програмно-апаратними засобами MailBox та Mutex Core., в досліджуваній архітектурі – засобами розробленого ПРС.

На діаграмі (рис. 3) зображено загальну позитивну динаміку збільшення ефективності функціонування досліджуваних конфігурації системи під час виконання обчислень від різної кількості аргументів СЛАР. Це є природньо за реалізації розгалуження обчислювального процесу. Показник ефективності (Е) розраховується як $E = K_i / N \times 100\%$, де K_i – показник прискорення обчислення ($K_N = T_N / T_1$), T_N, T_1 – час обчислення, N – кількість процесорних ядер.

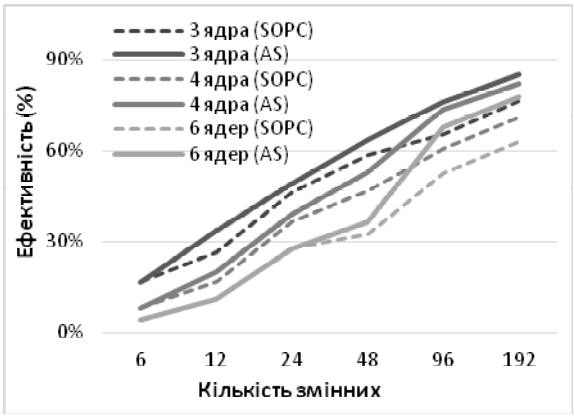


Рис. 3. Графічне зображення ефективості досліджуваних засобів синхронізації

Для аналізу інтенсивності приросту ефективності застосуємо показник абсолютного приросту ефективності: $\Delta E_i = E_i - E_{i-1}$, де $(i-1)$ показник попереднього виміру. Для оцінки середніх показників приросту ефективності застосуємо усереднені відносно кількості процесорів показники ефективності. З діаграми (рис. 4) видно, що взагалі застосування автоматичної синхронізації надає більш інтенсивний приріст ефективності в середньому на 2%. Як видно система забезпечує менший приріст ефективності у випадку неефективного використання ресурсів процесорних ядер або

переважної кількості операцій обміну даними. За ідеальних умов, коли кількість операцій передавання даних менша, або порівняна з кількістю обчислювальних операцій за рівномірного завантаження процесорних ядер інтенсивність приросту ефективності за запропонованого способу автоматичної синхронізації дорівнює 5%.

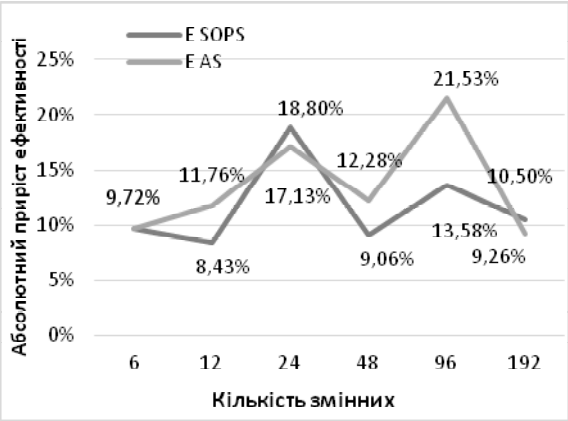


Рис. 4. Графічне зображення абсолютного приросту ефективності досліджуваних засобів синхронізації

Доцільно розглянути також показники інтенсивності прискорення обчислення, розраховані як $K_i = (K_i / K_{i-1}) \times 100\%$. На підставі графіків залежностей (рис. 5) видно, що інтенсивність прискорення обчислення за застосування засобів автоматичної синхронізації більше на 65% в середньому, ніж за застосування відомих засобів синхронізації.

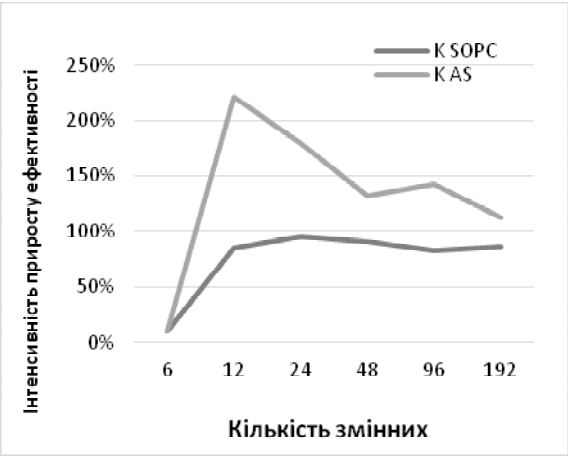


Рис. 5. Графічне зображення дослідження інтенсивності приросту ефективності

Абсолютний приріст ефективності досліджуваних підходів, розрахований за виразом $\Delta E_{AS} = E_{SOPC} - E_{AS}$, зображує позитивну динаміку збільшення ефективності (рис. 6), при цьому знову прослідковуються критичні ділянки на яких рівень ефективності зменшується, що характеризуються збільшенням кількості операцій обміну даними і невідповідністю розмірності задачі кількості процесорних ядер. Відносний показник приросту ефективності, розрахований за виразом $K_{AS} = (E_{SOPC} - E_{AS}) / E_{AS} \times 100\%$, в результаті зображує приріст ефективності за застосування запропонованого способу автоматичної синхронізації у середньому на 10,25%.

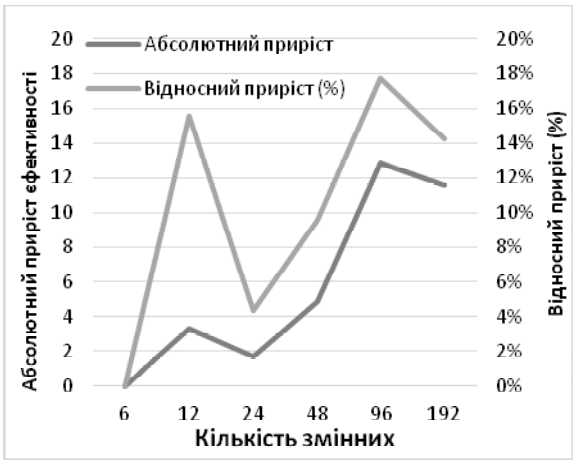


Рис. 6. Графіки показників приросту ефективності

Дослідження розроблених засобів показують, що за сумірної кількості процесорів та розмірності задачі спостерігається збільшення показників ефективності. За збільшення кількості процесорів на ефективність впливає степінь їх завантаження обчислювальними операціями. Абсолютний показник, що зображує динаміку зміни ефективності, зростає на ділянках з ідеальною організацією обчислювального процесу – сумірними параметрами обчислювальної системи та вирішуваної задачі, і зменшується в зворотному випадку. При цьому зберігається позитивна динаміка збільшення ефективності на всьому проміжку дослідження. Найбільшим негативним чинником впливу на ефективність в досліджуваній системі є операції обміну даними, коли їх час їх виконання переважає над часом обчислення. Дослідження показують різке зниження ефективності за високої кількості невідомих, з причини значного збільшення об'єму оброблюваних даних і відповідно часу на їх передавання.

Висновки

Розроблені засоби дозволяють забезпечити автоматичне управління паралельними обчис-

леннями та синхронізацію процесів прозоро для програмного шару обчислювальної системи, зменшуючи навантаження на центральні управляючі засоби та забезпечуючи апаратне прискорення реалізації функції управління.

Реалізація модифікованого способу автоматичної синхронізації показала прискорення обчислень та зберігання ресурсу вбудованої пам'яті мікросхеми ПЛІС, у порівнянні з засобами синхронізації процесів уніфікованої послідовності проектування компанії Altera. За результатами експериментів ефективність обчислень збільшилась у середньому на 10,25%. При цьому спостерігається збільшення інтенсивності приросту ефективності під час масштабування обчислювальної потужності системи-на-кристалі на 65%.

Дослідження підтвердили, що адаптація структури задачі під структуру обчислювальної системи, за шляхом зміни зернистості, забезпечує підвищення ефективності обчислень. При цьому застосування розроблених засобів дозволяє збільшити інтенсивність приросту ефективності від 2% до 5% за умови, якщо розмірність задачі відповідає кількості процесорних ядер.

Список використаних джерел

1. *Жабин В.И.* Архитектура вычислительных систем реального времени / В.И. Жабин. – К. : ВЕК +, 2003. – 176 с.
2. *Клименко І.А.* Тенденції застосування сучасної елементної бази для побудови високопродуктивних обчислювальних систем // Проблеми інформатизації та управління: Зб.наук.пр.– К. : Вид-во нац. авіац. ун-ту «НАУ-друк», 2010. – № 1(29). – С. 90 – 103.
3. *Левченко Р.И.* Проблемы эффективности автоматического динамического распараллеливания вычислений для многопроцессорных компьютерных систем со слабой связью / Р.И. Левченко, А.А. Судаков, С.Д. Погорелый, Ю.В. Бойко // Проблеми програмування – К.: ІПС НАНУ, 2010. – № 2 – 3. – С. 178 – 184.
4. Embedded Design Handbook. Ver. 2.9. – Altera Corporation, 2011. – Режим доступу: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/nios2/edh_ed_handbook.pdf. – Дата доступу: квітень 2015.
5. *Dorta T.* Reconfigurable Multiprocessor Systems: A Review / T. Dorta, J. Jiménez, J.L. Martín, U. Bidarte, A. Astarloa // International Journal of Reconfigurable Computing. – 2010. – Vol. 2010. – P. 7:1 – 7:11.
6. Dummler J. Scalable computing with parallel tasks / J. Dummler, T. Rauber, G. Runger // Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers [MTAGS '09], (November 16, Portland, Oregon). – US, NY: ACM, 2009. – P. 9:1 – 9:10.

Поступила в редакцию 20 мая 2015 г.

УДК 004.272.26, 004.274

И.А. Клименко, канд. техн. наук

Национальный технический университет Украины «Киевский политехнический институт»,

проспект Победы, 37, г. Киев, 03056, Украина.

Усовершенствование средств синхронизации процессов в мультипроцессорных системах-на-кристалле

Разработаны и исследованы средства автоматического управления параллельными вычислениями и синхронизации, которые позволяют усовершенствовать известные технологии проектирования, предлагаемые ведущими производителями ПЛИС, и повысить эффективность синхронизации процессов в мультипроцессорных системах-на-кристалле. Библ. 6, рис.6.

Ключевые слова: *мультипроцессорная система-на-кристалле; автоматическая синхронизация; вычисления под управлением потока данных; ПЛИС.*

.

UDC 519.6:512.972, 004.27

I. Klymenko, Ph.D.

National Technical University of Ukraine "Kyiv Polytechnic Institute",

37, Prospect Peremohy, 03056, Kyiv, Ukraine.

Improvement of mechanisms of the processes synchronization in multiprocessor systems-on-chip

Mechanisms of automatic control in parallel computing and synchronization is developed and investigated. Developed mechanisms improved the standard FPGA tools from the leading manufacturers of FPGA and increased efficient of the processes synchronization in multiprocessor systems-on-chip. Reference 6, figures 6.

Keywords: *multiprocessor systems-on-chip; MPSoC; automatic synchronization; dataflow computational model; FPGA.*

References

1. *Zhabin, V. I.* (2003). The architecture of the real-time computer systems. K.: VEK+, 176 p. (Rus)
2. *Klymenko, I. A.* (2010). Tendencies of the Use of Modern Element Base for Design of the High-Performance Computing Systems. Problemy Infirmatyzacii ta Upravlinnia: Zb. Nauk. Pratsc., Vol. 1(29), pp. 90 – 103. (Ukr)
3. *Levchenko, R. I., Sudakov, O. O., Pogorelij, S. D and Bojko, Y. V.* (2010). The problems of efficient of Automatic Dynamic Paralleling of Computations for Multiprocessor Computer Systems with Weak Connection. Problemy Programuvannya, No 2 – 3, pp. 178 – 184. (Rus)
4. (2011). Embedded Design Handbook. Ver 2.9. Altera Corporation, https://www.altera.com/ content/dam/altera-www/global/en_US/pdfs/literature/hb/nios2/edh_ed_handbook.pdf.
5. *Dorta, T., Jiménez, J., Martín, J. L. and all* (2010). Reconfigurable Multiprocessor Systems: A Review. International Journal of Reconfigurable Computing, Volume 2010, pp. 7:11 – 7:11.
6. *Dummler, J., Rauber, T. and Runger G.* (2009). Scalable computing with parallel tasks. Proceeding of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers, (MTAGS '09), US, Portland, Oregon, November 14 – 20, 2009, pp. 9:1 – 9:10.