

УДК 519.6:512.972, 004.27

І.А. Клименко¹, канд. техн. наук, доцент,
В.В. Ткаченко², канд. техн. наук, доцент,
Є.В. Красовська¹, канд. техн. наук¹Національний авіаційний університет, м. Київ, Україна
iklyumenko@yandex.ua, pretty1001@yandex.ru²Національний технічний університет «КПІ», м. Київ, Україна
tkvalentina@yandex.ru

Співпроцесор для виконання арифметичних операцій з плаваючою комою на ПЛІС

Розглянуто можливу реалізацію обчислювальної системи-на-кристалі для рішення задач управління в реальному часі. Означені основні напрямки підвищення продуктивності систем оглянутого класу та, як один із ефективних засобів, запропоновано співпроцесор на ПЛІС для виконання арифметичних операцій з плаваючою комою. Розроблено модуль співпроцесора для виконання операції додавання з плаваючою комою на ПЛІС Cyclon II компанії Altera. Результати моделювання показали значне збільшення швидкості обчислень на рівні апаратного модуля співпроцесора в порівнянні з програмною реалізацією додавання багатобайтних слів у форматі з плаваючою комою на серійному мікроконтролері.

Ключові слова: ПЛІС, програмовані логічні інтегральні схеми, система-на-кристалі, паралельні обчислення, обчислювальна система управління в реальному часі, співпроцесор, арифметичний розширювач, арифметика з плаваючою комою

Вступ

Одним з ефективних напрямків підвищення продуктивності обчислювальних систем (ОС) є перехід до паралельних обчислень, як на програмному, так і на архітектурному рівні. З іншого боку сучасний рівень розвитку мікроелектроніки дозволяє значно підвищити продуктивність обчислень за рахунок вдосконалення елементної бази. Сьогодні програмовані логічні інтегральні схеми (ПЛІС) є гідною альтернативою замовленим інтегральним схемам класу ASIC за всіма характеристиками для кінцевої розробки найскладніших спеціалізованих цифрових пристроїв. Ріст можливостей сучасних ПЛІС дозволяє розробникам реалізувати складні мультипроцесорні системи на одному кристалі [1], а провідні компанії-розробники ПЛІС, в свою чергу, пропонують можливості застосування вбудованих програмованих та апаратних ядер процесорів, блоків пам'яті, арифметичних модулів, периферії та схем зв'язку. Можливість динамічної реконфігурації є однією з потужніших і перспективних якостей технології ПЛІС, яка забезпечує гнучкість проектування, масштабування продуктивності, адаптацію до конкретних вимог вирішуваної задачі.

Дослідження присвячені розробці ОС на базі ПЛІС класу система-на-кристалі (ОС ПЛІС), які в контексті виконуваних ними специфічних задач підпадають під розряд нестандартного,

спеціалізованого електронного обладнання. Для таких ОС застосування ПЛІС пропонує велику кількість переваг [1], на підставі чого слід означити основні проблеми досліджуваної області: обґрунтування життєздатності та перспектив, технології ПЛІС для побудови ОС широкого застосування; розробка та втілення методів та засобів підвищення продуктивності ОС та ПЛІС. Необхідність вирішення означених проблем обґрунтовують актуальність та доцільність наукових досліджень в галузі розробки як ОС, так і різноманітних вузлів нестандартного електронного обладнання на ПЛІС.

Постановка задачі

Зважаючи на найбільш розповсюджені завдання, що вирішуються ОС класу система-на-кристалі, а саме – задачі управління складними технологічними процесами в реальному часі, які характеризуються переважною кількістю операцій обміну даними під час циклу управління, та потребують високої продуктивності й надійності/В роботах [2, 3] в якості найбільш ефективної архітектури для реалізації ОС ПЛІС визначена відкрита архітектура із загальною шиною, побудована за модульним принципом. Окрім втілення паралелізму на апаратному рівні та застосування ефективної елементної бази, необхідно розробляти різноманітні засоби та методи вдосконалення архітектури, що сприятимуть підвищенню продуктивності та

надійності систем-на-кристалі. Зокрема, для ОС із загальною шиною, що функціонують в режимі реального часу, в контексті досягнення високої продуктивності актуальним стає вирішення питань зменшення тривалості циклу управління, підвищення ефективності обробки зовнішніх переривань, вирішення конфліктів під час доступу до загального системного ресурсу, апаратна реалізація складних спеціалізованих функцій.

В роботі [4] запропонована можлива реалізація ОС ПЛІС. Основні алгоритми обчислень, протоколи обміну даними, синхронізація роботи процесорів виконуються на програмному рівні. Для забезпечення високої продуктивності обчислень у склад системи запропоновано додавати апаратні співпроцесори, призначені для високопродуктивної реалізації складних спеціалізованих функцій, на обчислення яких на програмному рівні потребується значний час. Однотипність структури інтерфейсів для зв'язку з системною шиною та способу обміну даними між співпроцесорами та процесорами ОС, що вимагає модульний принцип архітектури, дозволяє досить просто додавати до складу ОС співпроцесори різного функціонального призначення, таким чином здійснювати реконфігурацію системи відповідно вимогам вирішуваних задач. До складу ОС [4] входить процесорне ядро, співпроцесор, основана пам'ять, розподілений контролер переривань, контролер прямого доступу до пам'яті (ПДП), зовнішні пристрої. Обмін даними між процесором та співпроцесором відбувається через загальну пам'ять під управлінням контролеру ПДП.

Основна конфігурація архітектури ОС ПЛІС – процесорне ядро. Найбільш широко застосовують програмовані процесорні ядра, що випускаються відомими компаніями *Xilinx* та *Altera* [2]. В роботі [5] запропоноване власне процесорне ядро з мікропрограмним управлінням, яке може бути застосовано у складі ОС [4]. В якості переваг: процесорне ядро легко модифікується в напрямку підвищення розрядності оброблюваних слів, за рахунок модульної структури арифметико-логічного пристрою (АЛП); адаптація до рішення різних класів прикладних задач відбувається досить легко на рівні зміни системи команд і завантаження їх у пам'ять мікрокоманд без будь-яких змін в архітектурі пристрою. Процесорне ядро може бути застосоване також для побудови мультипроцесорної системи-на-кристалі.

В роботах [6, 7] запропоновані засоби підвищення ефективності обробки зовнішніх переривань, актуальні для систем управління в реальному часі. Запропонована система

переривань із забезпеченням динамічних пріоритетів обробки запитів, що сприятиме рівномірному обслуговуванню запитів від зовнішніх пристроїв та процесорів. В роботах [4, 8] запропоновані спеціалізовані апаратні обчислювачі на ПЛІС для вбудовування у склад ОС. Застосування таких обчислювачів є ефективним засобом для підвищення продуктивності обчислень.

Актуальним питанням підвищення продуктивності обчислень є реалізація операцій із плаваючою комою (ПК), які зазвичай в серійних мікропроцесорах (МП) виконуються на програмному рівні та займають значний час загального обчислювального процесу. В роботі запропонований спеціалізований обчислювач з мікропрограмним управлінням на ПЛІС для реалізації операцій з ПК. Обчислювач може бути вбудований в запропоновану авторами роботи [4] систему-на-кристалі. Розроблена модель обчислювача для виконання операцій додавання в форматі з ПК засобами системи автоматизації проектування (САПР) *Qwartus II* компанії *Altera*.

Математичне обґрунтування

Нехай за умови поставленої задачі мантиси змінюються в межах $1/2 \leq M_i < 1$.

Суму двох чисел $X = 2^{P_x} M_X$ і $Y = 2^{P_y} M_Y$, поданих у форматі з ПК, можна записати у вигляді

$$2^{P_x} M_X + 2^{P_y} M_Y = 2^{P_z} M_Z.$$

Для додавання чисел із ПК необхідно привести їх до загального порядку $P_{заг}$, в якості якого зручно обрати більший порядок з двох доданків $P_{заг} = \max(P_X, P_Y)$.

При цьому мантиса числа з меншим порядком підлягає зменшенню за рахунок зсуву праворуч. В протилежному випадку виникає переповнення розрядної сітки мантиси числа, що перетворюється. Після цього суму двох чисел можна подати у вигляді

$$2^{P_{заг}} M_X + 2^{P_{заг}} M'_Y = 2^{P_{заг}} (M_X + M'_Y),$$

де за M'_Y прийнято перетворену мантису числа з меншим порядком.

Виконання операцій додавання та віднімання чисел із ПК у загальному вигляді складається з наступних етапів: вирівнювання порядків; підсумовування мантис; визначення порядку результату; нормалізація результату; округлення результату; кінцева нормалізація результату [9].

Узагальнена структура спеціалізованого обчислювача

Узагальнена структура спеціалізованого обчислювача для виконання операцій з ПК подана на рис. 1. Пристрій підключається до системної шини, яка складається з трьох груп провідників – шини адреси (ША), шини даних (ШД) та шини управління (ШУ). У склад інтерфейсу системної шини (ІСШ) входить регістр стану, в якому співпроцесор виставляє ознаки, в тому числі ознаку готовності прийняття даних для чергового обчислення; стек даних із декількох регістрів, в який в програмному режимі записуються операнди (X, Y) та командне слово або видається результат (Z) після виконання обчислень. Перед процедурою обміну даними із співпроцесором процесор зчитує вміст регістру стану для аналізу ознаки готовності. Після чого ініціює процедуру передачі даних в стек даних співпроцесора. Командне слово містить інформацію щодо типу виконуваної операції, належить до управляючої інформації і передається на входи блоку мікропрограмного управління (БМУ) локальною шиною управління (ЛШУ). На підставі команди здійснюється перехід до відповідних мікропідпрограм у пам'яті мікропрограм. По закінченні обчислень співпроцесор ініціює переривання до процесора шляхом встановлення відповідного біту в регістрі стану. Процесор перериває виконання основної програми та зчитує результати обчислень в основну системну пам'ять. Обмін даними між процесором та співпроцесором відбувається в режимі ПДП. Управляючі сигнали між співпроцесором та процесором або контролером ПДП передаються по шині управління (ШУ).

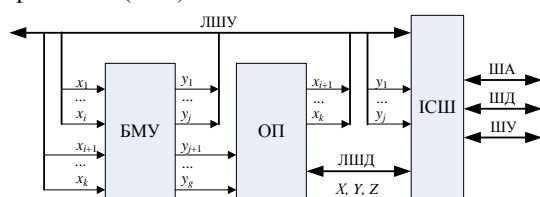


Рисунок 1 – Узагальнена структура спеціалізованого обчислювача для виконання операцій з ПК

БМУ приймає від ІСШ логічні умови ($x_1...x_i$), які управляють розгалуженням мікропрограм у ПМК, та формує управляючі сигнали ($y_1...y_i$) для ІСШ. Також БМУ приймає логічні умови ($x_{i+1}...x_j$) від операційного пристрою, та формує для нього управляючі сигнали ($y_{i+1}...y_g$). Управляючі сигнали між функціональними блоками співпроцесора передаються ЛШУ.

Операційний пристрій може бути структурований для виконання різноманітних операцій з ПК – додавання, множення, ділення, обчислення квадратного кореню, тощо. Для виконання кожної з них до складу пристрою додаються відповідні обчислювальні модулі. Обчислення відбуваються під управлінням мікропідпрограм у ПМК, таким чином під час розширення функціоналу пристрою у ПМК додаються спеціальні мікропідпрограми. За використання технології ПЛІС для реалізації співпроцесора зміна функціональних можливостей пристрою не є складною задачею. Така зміна може виконуватись досить швидко, в тому числі внутрисистемно – без вилучення пристрою зі складу системи та повернення до потоку проектування [2]. Таким чином запропонований пристрій втілює основну і найбільш актуальну на сьогодні ідею застосування ПЛІС – це можливість їх динамічної реконфігурації в контексті адаптації під класи вирішуваних задач.

У загальному випадку операційний пристрій для виконання операції додавання із ПК складається з трьох функціональних частин: операційного пристрою порядків (ОПП), що виконує етап вирівнювання порядків і отримання порядку результату, операційного пристрою для підсумовування мантис (СММ) та операційного пристрою нормалізації мантиси (ОПН) результату з подальшим його округленням (рис. 2). На рис. 2 застосовані наступні позначення: X, Y – операнди; MX, MY – мантиси операндів після виконання вирівнювання порядків; MZ_0, PZ_0 – мантиса та порядок ненормалізованого результату, Z – результат обчислення.

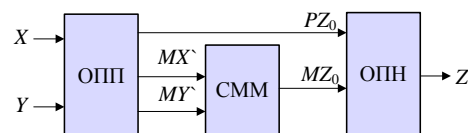


Рисунок 2 – Операційний пристрій для додавання чисел із ПК

Розробка операційної частини співпроцесора*Блок вирівнювання порядків*

Операційна схема блоку для виконання операції вирівнювання порядків для підсумовування чисел із ПК подана на рис. 3 [10].

Блок складається з регістрів $RGPX, RGPY$ для зберігання порядків та RGX, RGY для зберігання мантис чисел із ПК. Регістри порядків та мантис мають розрядність $(n+2)$ та $(m+2)$, де n та m розрядність порядку та мантиси відповідно, по два розряди додаються для зберігання знакових розрядів порядків та мантис. У регістрі $RG\Delta$ запам'ятовується різниця порядків Δ . Також у

Операційна схема пристрою складається з регістрів *RGPZ*, *RGZ* – відповідно регістрів порядку та мантиси результату, записаної у модифікованому доповняльному коді. Комбінаційна схема *KC* застосовуються для виявлення порушення нормалізації мантиси результату. За цього перевіряються умови порушення нормалізації та генеруються управляючі сигнали *D*, *L* та *R* зсуву регістру *RGZ* ліворуч або праворуч та додавання або віднімання одиниці в регістрі *RGPZ*.

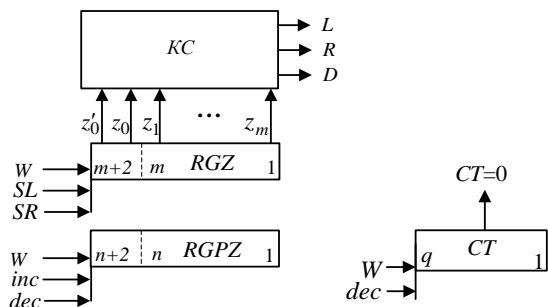


Рисунок 5 – Операційна схема блоку виконання операції нормалізації модифікованого доповняльного коду

Лічильник *CT* застосовується для обмеження кількості зсувів числом розрядів мантиси результату, бо процес зсувів під час ППН може виявитись нескінченним, якщо в результаті додавання мантис отримано нуль. Розрядність лічильника дорівнює $q = \lceil \log_2 m \rceil$, де *m* – розрядність мантиси. У вихідному стані у лічильник заноситься значення *m*.

Синтез комбінаційної схеми детально розглянутий в роботі [10]. Під час синтезу отримані логічні вирази для побудови *KC*, які розглядаються в якості умов порушення нормалізації вліво, вправо, та за формування значення мантиси 11,000...00 відповідно:

$$L = z'_0 \overline{z_0} \vee \overline{z'_0} z_0,$$

$$R = z'_0 z_0 z_1 \vee z'_0 \overline{z_0} z_1,$$

$$D = z'_0 z_0 (z_1 \vee \dots \vee z_m).$$

Змістовний МА роботи пристрою для виконання операції нормалізації подано на рис. 6.

Виконання нормалізації результату виконується шляхом зсуву регістру *RGZ*, що зберігає мантису результату, ліворуч (або праворуч) і віднімання (або додавання) одиниці від регістру порядку результату *RGPZ* при кожному зсуві до тих пір, поки будуть зберігатися вказані комбінації цифр.

Для запобігання можливих переповнень та зникнень порядку під час корекції порядку

необхідно заздалегідь на етапі проектування пристрою розрахувати розрядність регістру порядку з урахуванням максимального модуля порядку. Розрядність регістру порядку має дорівнювати кількості розрядів для запису суми двох максимальних модулів порядків та знакових розрядів.

Під час нормалізації результату шляхом зсуву праворуч відкидання молодшого розряду може привести до накопичування великої додатної похибки. Для виключення цього застосовується операція округлення. Суматор *SM* та регістр *RGZ* мають додатковий розряд зі сторони молодших розрядів. Округлення здійснюється додаванням у цей розряд одиниці й наступним відкиданням його вмісту.

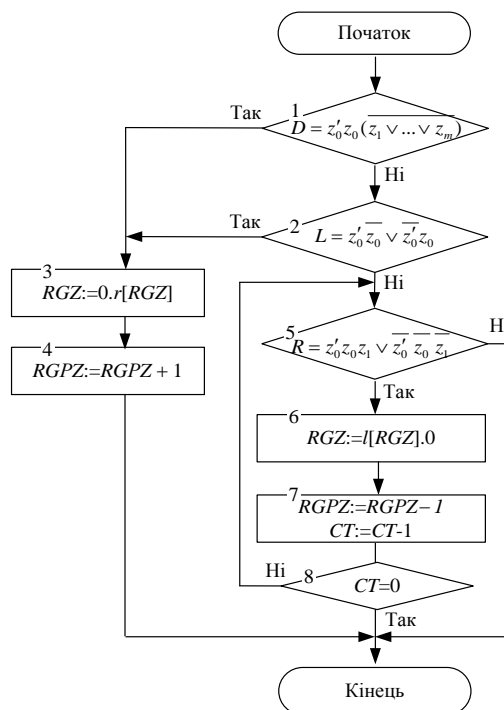


Рисунок 6 – Змістовний МА виконання операції нормалізації модифікованого коду

Розробка блоку мікропрограмного управління

Схема БМУ з примусовою адресацією подана на рис. 7. Теоретичні основи побудови БМУ викладені в роботі [10].

Основні функціональні частини БМУ: РАМК – регістр адреси мікрокоманди (МК), призначений для зберігання адреси A_i в поточному *i*-му такті; ПМК – пам'ять МК; РМК – регістр МК, призначений для зберігання МК на протязі часу її виконання; *clk* – синхросигнал; $\{x_j\}$ – логічні умови, що надходять на вхід мультиплексора умов (МУ) і управляють розгалуженням мікропрограм.

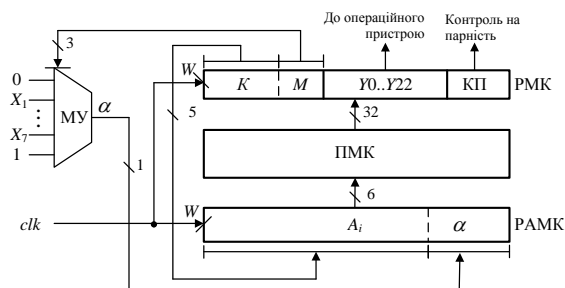


Рисунок 7 – Структурна схема БМУ з примусовою адресацією

МК розміщуються у пам'яті мікрокоманд. Формат МК має вигляд наведений на рис. 8.

МК:	Адресна інформація		Управляюча інформація	Контроль на парність
	К	М	УС	
(32)	(5)	(3)	(22)	(1)

Рисунок 8 – Формат мікрокоманди

Слово МК складається з трьох зон: зона формування адреси наступної МК, що

00	1	00000	111	000	0000	0000	0011	1100	1111	838003CF
01	1	00001	000	000	0000	0000	0000	0011	0000	84000030
02	1	00001	111	000	0000	1101	0000	0000	0000	8780D000
03	1	00010	010	000	0000	1000	0000	0000	0000	89008000
04	1	00011	000	000	0000	1000	0000	0000	0000	8C008000
05	1	00011	111	000	0000	1010	0000	0000	0000	8F80A000
06	1	00100	001	000	0000	1000	0000	0000	0000	90808000
07	1	00101	001	000	0000	1010	0000	0000	0000	9480A000
08	1	00011	000	000	0000	1100	0001	0000	0000	8C00C100
09	1	00101	111	000	0000	0000	0000	0000	0000	97800000
0A	1	00011	111	000	0000	1110	0000	0100	0000	8F80E040
0B	0	00110	000	101	0000	0000	1100	0000	0000	18500C00
0C	1	00111	011	000	0000	0000	0000	0000	0000	9D800000
0D	1	01001	101	000	0000	0000	0000	0000	0000	A6800000
0E	0	01010	111	110	0000	0000	0100	0000	0000	2BE00400
0F	0	01001	000	000	1111	0000	0000	0000	0000	240F0000
10	0	00110	111	100	1001	0000	0000	0000	0000	1BC90000
12	0	01010	000	000	0000	0000	0000	0000	0000	28000000
13	0	01000	100	000	0000	0000	0000	0000	0000	22000000
14	0	01010	000	000	0000	0000	0000	0000	0000	28000000
15	0	01010	000	000	0000	0000	0000	0000	0000	28000000
16	0	01010	000	000	0011	0000	0000	0000	0000	28030000

Рисунок 9 – Управляюча програма в ПМК

Модельовання роботи співпроцесора

Проект співпроцесора для додавання чисел з ПК розроблений в системі автоматизованого проектування *Quartus II* компанії *Altera* [11].

складається з поля адресної інформації (К) та поля управління мультиплексором (М); зона кодування управляючих сигналів (УС); зона контролю слова МК на парність, яка містить один розряд.

У кожному такті за синхросигналом *clk* адреса МК поновлюється у РАМК і надходить на адресний вхід ПМК. За поточною адресою відповідна МК видається на вихід даних ПМК. Слово МК записуються у РМК за зворотним перепадом синхросигналу *clk*.

У склад розробленої моделі БМУ входить ПМК, що складається з 64 слів. Розрядність адреси дорівнює семи розрядам, тоді розрядність поля адресної інформації (К) дорівнює п'яти. Під час побудови загальної структурно-функціональної схеми обчислювача та алгоритму управління визначена кількість управляючих сигналів (УС) для виконання операції додавання із ПК, що дорівнює двадцяти трьом сигналам – $Y_{22} - Y_0$, та кількість логічних умов – шість. Для кодування логічних умов необхідно трирозрядне поле для управління мультиплексором (МУ). Таким чином отримуємо розрядність МК, що дорівнює тридцяти двом розрядам (рис. 8).

Фрагмент управляючої мікропрограми в ПМК представлений на рис. 9:

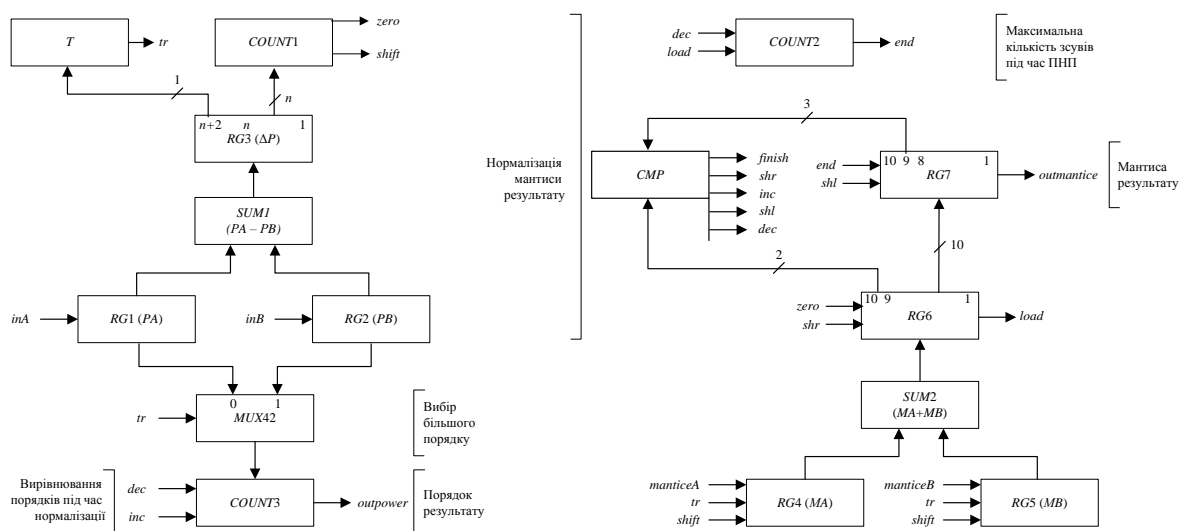


Рисунок 10 – Базова схема додавання чисел з ПК

Далі наведений опис алгоритму обчислень з урахуванням функціонального призначення кожного з модулів проекту.

1. У регістри $RG1$ й $RG2$ записуються вихідні порядки – PA , PB відповідно, а в $RG4$ й $RG5$ – вихідні мантиси MA , MB .

2. На суматорі $SUM1$ здійснюється віднімання порядків ($PA - PB$), результат фіксується в регістрі $RG3$, старший знаковий розряд результату зберігається в тригері T , а значущі розряди в лічильнику $COUNT1$.

3. На тригері T встановлюється сигнал Tr , поточний стан тригеру вказує на більший порядок ($Tr=0$, якщо $A>B$ та $Tr=1$, якщо $A<B$); лічильник $COUNT1$ формує управляючий сигнал $Shift$ зсуву вправо вмісту регістрів $RG4$ або $RG5$, враховуючи значення Tr ; зсув вправо відбувається до тих пір поки значення лічильника не дорівнюватиме нулю – встановлення сигналу $zero=1$.

4. За встановлення сигналу $zero = 1$ мантиси підсумуються на $SUM2$ і результат зберігається в регістр $RG6$; вихідні значення порядків з регістрів $RG1$ та $RG2$ надходять на мультиплексор $MUX42$ та за управлінням сигналу Tr більше значення порядку зберігається в $COUNT3$.

5. Знакові розряди отриманої суми мантис з регістру $RG6$ надходять на компаратор CMP для запобігання порушення нормалізації ліворуч, якщо вони не рівні – виконується зсув праворуч $RG6$ за сигналом Shr і інкремент $COUNT3$ по сигналу inc , а на управляючому виході $RG6$ формується сигнал $load$, за яким відбувається запис в $COUNT2$.

6. Інформація з регістру $RG6$ записується в $RG7$.

7. Знаковий розряд та старший значущий розряд вмісту регістру $RG7$ надходить у

компаратор CMP для запобігання порушення нормалізації праворуч, якщо вони рівні – виконується зсув ліворуч $RG7$ за сигналом shl , декремент $COUNT2$ по сигналу dec ; інакше – видається сигнал end .

8. Виконання пункту 7 триває, поки в $COUNT2$ не дорівнюватиме нулю або поки не виданий сигнал end .

Значення управляючих сигналів наступні: керуючі входи регістрів: $Y0 - shr$ $RG1$; $Y1 - load$ $RG1$; $Y2 - shr$ $RG2$; $Y3 - load$ $RG2$; $Y4 - shr$ $RG3$; $Y5 - load$ $RG3$; $Y6 - shr$ $RG4$; $Y7 - load$ $RG4$; $Y8 - shr$ $RG5$; $Y9 - load$ $RG5$; $Y10 - shr$ $RG6$; $Y11 - load$ $RG6$; $Y12 - aload$; $Y13 - updown=1$; $Y14 - inc/dec$; тригер T : $Y15 - C$; лічильник $COUNT2$: $Y18 - aset$; $Y19 - dec$; лічильник $COUNT3$: $Y20 - aload$; $Y21 - updown=1$; $Y22 - inc/dec$. При цьому shr/shl – зсув праворуч/ліворуч, $load$ – завантаження регістрів, $aload$ – завантаження лічильників, inc/dec – інкремент/декремент.

Далі наведені значення логічних умов ($X1 - X7$), що надходять на інформаційні входи (1 – 7) мультиплексора умов БМУ, за якими виконується розгалуження мікропрограм: 0 ($X1$) – логічний «0», 1 ($X2$) – встановлення значення 0 на лічильнику порядку $COUNT1$; 2 ($X3$) – встановлення 1 на тригері T ; 3 – ($X4$) – ПНП ($RG6[9]=RG6[8]$); 4 ($X5$) – ПНЛ ($RG6[8]\neq RG6[7]$); 5 ($X6$) – встановлення значення 0 на лічильнику мантиси; 7 ($X7$) – логічна «1».

У процесі побудови схеми проекту були застосовані мегафункції САПР *Quartus II*. За допомогою мегафункцій були побудовані такі модулі: пам'ять мікрокоманд, регістр зі зсувом праворуч; регістр зі зсувом ліворуч; мультиплексор умов $MUX41$ (сім однорозрядних

інформаційних входи, трирозрядний управляючий вхід та однорозрядний інформаційний вихід); мультиплексор вибору більшого порядку *MUX42* (два шестирозрядних інформаційних входи, однорозрядний та шестирозрядний вихід). Після створення та налагодження всі модулі створені засобами графічного проектування були

конвертовані в текстові файли на мові опису апаратури *Verilog*.

Для прикладу наведемо фрагмент проекту, що містить схему БМУ, розроблену засобами графічного проектування у САПР *Quartus II* (рис. 11).

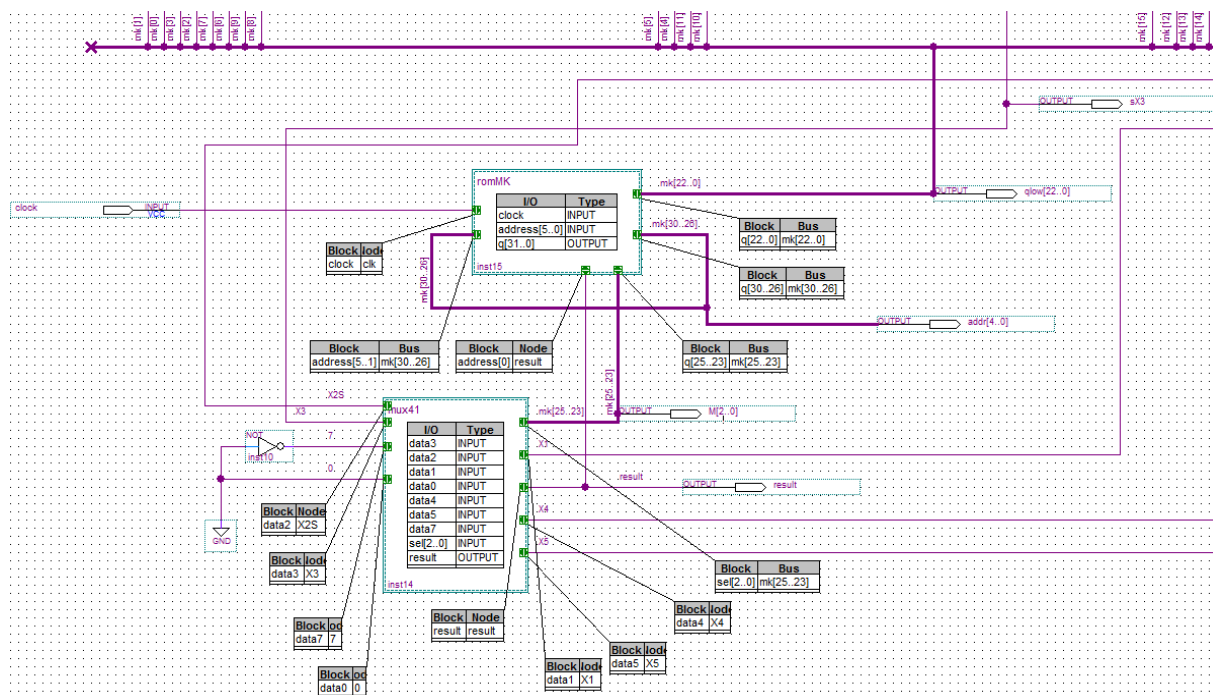


Рисунок 11 – Схема БМУ у САПР *Quartus II*

БМУ містить 2 основні частини: мультиплексор умов *mux41* і пам'ять МК *romMK*. За примусової адресації (рис. 9) інформаційний розряд *result* з виходу МП умов, є молодшим розрядом адреси наступної МК *address[0]*, інші розряди адреси *address[5..1]* поступають на адресний вхід *romMK* безпосередньо із поля К попередньої МК *mk[30..26]*. Прийняття адреси модуля відбувається за фронтом вхідного сигналу *clock*, тому немає необхідності в реалізації спеціального регістру (СФАМК) для збереження адреси на вході ПМК, так само як і регістру на виході ПМК для збереження слова МК (РМК). Сигнали на управляючі входи МП умов надходять з поля *M* слова МК *mk[25..23]*. Управляючі сигнали *mk[22..0]* поступають на операційний пристрій.

Функціональне моделювання роботи пристрою додавання з ПК виконано засобами моделюючої програми *ModelSim* компанії *MentorGraphics*. Часове моделювання пристрою у

САПР *Quartus II* виконується після розміщення та трасування проекту з урахуванням реальних затримок і часу розповсюдження сигналів на мікросхемі *EP2C35F672C6N* сімейства *Cyclone II* фірми *Altera* [12]. Часові параметри за цього на 90% наближені до реальних. Час на виконання обчислень отриманий у САПР *Quartus II* дорівнює 220 нс. У порівнянні з часом виконання операції додавання трибайтових двійкових чисел у форматі з ПК (мантиса – два байта, порядок – один байт) на програмному рівні засобами серійного восьмибітного мікроконтролера *AT89C51* фірми *ATMEL*, що дорівнює 217 мкс, маємо підвищення швидкодії приблизно на три порядки.

Фрагмент звіту компілятора щодо розміщення пристрою на мікросхемі *Cyclone II* наведений на рис. 12. Зі звіту видно, що розроблена модель займає 2% логічного ресурсу мікросхеми та 1% об'єму убудованої пам'яті. Отримані показники дозволяють зробити висновок про можливість розширення функціональних можливостей співпроцесора.

Fitter Summary	
Fitter Status	Successful - Mon Jun 14 09:28:13 2010
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	float_adder
Top-level Entity Name	float_adder
Family	Cyclone II
Device	EP2C8F256C6
Timing Models	Final
Total logic elements	130 / 8,256 (2 %)
Total combinational functions	129 / 8,256 (2 %)
Dedicated logic registers	73 / 8,256 (< 1 %)
Total registers	73
Total pins	48 / 182 (26 %)
Total virtual pins	0
Total memory bits	1,984 / 165,888 (1 %)
Embedded Multiplier 9-bit elements	0 / 36 (0 %)
Total PLLs	0 / 2 (0 %)

Рисунок 12 – Фрагмент звіту компілятора щодо розміщення пристрою на мікросхемі Cyclone II

Висновки

Багатотактні арифметичні операції множення та ділення, процедури обробки порядків, нормалізації та округлення результату, що мають місце під час виконання операцій з ПК потребують значного часу виконання на програмному рівні, особливо це актуально для різноманітних спеціалізованих ОС, в тому числі систем управління в реальному часі, де традиційно застосовуються серійні мікроконтролери з невеликою розрядністю оброблюваних слів. Реалізація складних багатотактних операцій на рівні апаратних засобів значно підвищує швидкість ОС. Запропонована апаратна реалізація обчислювальних блоків на ПЛІС у складі спеціалізованої ОС сприятиме підвищенню продуктивності обчислень за рахунок зменшення часу виконання обчислень та можливості динамічної реконфігурації від розрядності оброблюваних слів до зміни функціональних можливостей відповідно вимог вирішуваних задач.

Запропоноване рішення дозволяє реалізувати операції додавання, множення та ділення з плаваючою комою на співпроцесорі, що входить до складу обчислювальної системи. Синхронізація центрального процесора та

співпроцесора здійснюється на програмному рівні.

Системи управління для рішення задач в реальному часі належать до нестандартної вузькоспеціалізованої цифрової апаратури, для реалізації якої на сьогодні доцільно використовувати технології ПЛІС. Програмовані логічні інтегральні схеми на ряду з порівняними технічними характеристиками замовлених інтегральних мікросхем, мають ряд переваг – розробка цифрових пристроїв будь-якої складності, аж до багатопроцесорних ОС, при цьому забезпечення високої швидкодії і низького енергоспоживання; простота проектування, налагоджування та вдосконалювання; низькі затрати на виробництво; можливість динамічної реконфігурації архітектури відповідно вимогам вирішуваних задач.

Засоби для реалізації арифметики з ПК на базі ПЛІС дозволяють вдосконалити логічну структуру ОС та, за рахунок високої швидкодії елементної бази, підвищити продуктивність обчислень.

Результати моделювання роботи співпроцесора показали зменшення на три порядки часу обчислення у порівнянні з програмною реалізацією обчислень засобами серійних мікроконтролерів.

Список літератури

1. Клименко І.А. Тенденції застосування сучасної елементної бази для побудови високопродуктивних обчислювальних систем / І.А. Клименко // Проблеми інформатизації та управління: зб. наук. пр. – 2010. – Вип. 1 (29). – С. 90 – 103.
2. Клименко І.А. Класифікація та архітектурні особливості програмованих мультипроцесорних систем-на-кристалі / І.А. Клименко // Проблеми інформатизації та управління: зб. наук. пр. – 2012. – Вип. 1 (36). – С. 58 – 75.
3. Жабин В.И. Архитектура вычислительных систем реального времени / В.И. Жабин. – К.: ВЕК+, 2003. – 176 с.
4. Жуков І.А. Обчислювальна система для розв'язку нечітких СЛАР / І.А. Жуков, І.А. Клименко // Проблеми інформатизації та управління: зб. наук. пр. – 2011. – Вип. 3 (35). – С. 34 – 43.
5. Жуков І.А. Модуль оперативної пам'яті для RISC процесора на ПЛІС / І.А. Жуков, І.А. Клименко // Проблеми інформатизації та управління: зб. наук. пр. – 2009. – Вип.2 (28). – С. 50 – 54.
6. Клименко І.А. Реалізація контролера пріоритетних переривань для обчислювальної системи з відкритою архітектурою / І.А. Клименко, Н.А. Сакара, М.В. Федорчук // Вісник НТУУ «КПІ».

Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Видавництво «ВСК +», 2009. – №51. – С. 172 – 176.

7. Арифметичні та управляючі пристрої цифрових ЕОМ: [навчальний посібник] / В.И. Жабин, І.А. Жуков, І.А. Клименко, С.Г. Стіренко. – К.: ВСК +, 2008. – 176 с.

8. Багатопроцесорна обчислювальна система: пат. 60398, Україна, МПК G06F 15/16 (2006.01) / І.А. Жуков, І.А. Клименко, С.М. Біляєв ; заявник і патентовласник: Національний авіаційний університет. – заявл. 15.06.10; опубл. 25.06.11, Бюл. №12.

9. Цифровий обчислювач для розв'язання систем лінійних алгебраїчних рівнянь великої розмірності: пат. 51973, Україна, МПК (2006): G06F 17/10 / І.А. Жуков, Є.В. Красовська, О.О. Синельников // заявник і власник Національний авіаційний університет. – № u201001023; заявл. 01.02.10; опубл. 10.08.10, Бюл. № 15.

10. Прикладна теорія цифрових автоматів / В.И. Жабин, І.А. Жуков, І.А. Клименко, В.В. Ткаченко. – К.: НАУ, 2009. – 463 с.

11. Клименко І.А. Реалізація контролера пріоритетних переривань для обчислювальної системи з відкритою архітектурою / І.А. Клименко, Н.А. Сакара, М.В. Федорчук // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: зб. наук. пр. – 2009. – № 51. – С. 172 – 176.

12. Embedded Design Handbook Version 2.9. [Електронний ресурс]. – Altera Corporation, 2011. – Режим доступу: http://www.altera.com/literature/hb/nios2/edh_ed_handbook.pdf.

13. Cyclone II Device Handbook [Електронний ресурс]. – Altera Corporation, 2008. – Режим доступу: <http://www.altera.com/devices/fpga/cyclone2/cy2-index.jsp>.

Надійшла до редакції 30.07.2012

**І.А. КЛИМЕНКО¹, В.В. ТКАЧЕНКО²,
Є.В. КРАСОВСКАЯ¹**

¹Національний авіаційний університет,
г. Киев, Украина

²Національний технічний університет «КПІ»,
г. Киев, Украина

**І.А. KLYMENKO¹, V.V. TKACHENKO²,
I.V. KRASOVSKA¹**

¹National Aviation University, Kiev, Ukraine

²National Technical University of Ukraine “Kyiv
Polytechnic Institute”, Kiev, Ukraine

Сопроцессор для реализации арифметических операций с плавающей запятой на ПЛИС.

Рассмотрена возможная реализация вычислительной системы-на-кристалле для решения задач управления в реальном времени. Обозначены основные направления повышения производительности систем рассмотренного класса и предложен сопроцессор на ПЛИС для выполнения арифметических операций с плавающей запятой. Разработан модуль сопроцессора для выполнения операции сложения с плавающей запятой в ПЛИС Cyclon II компании Altera. Результаты моделирования показали значительное увеличение скорости вычислений на уровне аппаратного модуля сопроцессора по сравнению с программной реализацией сложения многобайтных слов в формате с плавающей запятой на серийном микроконтроллере.

Ключевые слова: ПЛИС, программируемые логические интегральные схемы, система-на-кристалле, параллельные вычисления, вычислительная система управления в реальном времени, сопроцессор, арифметический расширитель, арифметика с плавающей запятой

The FPGA Coprocessor for Implementation of Floating-Point Arithmetic Operations.

The possible implementation of a system-on-chip for control problems solution in real time is considered. We singled out basic methods of improving productivity for systems of this class and suggested a FPGA coprocessor to perform arithmetic operations with floating point. The coprocessor to perform floating-point add on FPGA Cyclon II of Altera is designed. Simulation results show a significant increase in performance at the hardware coprocessor as compared with software implementation of the addition of multibyte words with floating point at the serial microcontroller.

Keywords: FPGA, field-programmable gate array, system-on-chip, SoC, parallel computing, computer control system in real time, coprocessor, arithmetic extender, floating-point arithmetics