

ЯДРО 16-РОЗРЯДНОГО RISC-ПРОЦЕСОРА

Розглянута розробка шістнадцятирозрядного ядра RISC-процесора, реалізованого в ПЛІС, яке має високу швидкість та невеликі апаратні витрати. Система команд ядра адаптована для виконання алгоритмів компресії та керування.

A soft IP core of the 16-bit RISC microprocessor for the FPGA implementation is considered, which distinguished by low hardware volume, and high speed. The core instruction set is adapted to implement the lossless compression, and control algorithms.

Ключові слова: ПЛІС, RISC, LISA, хеш-функція.

1. Вступ

Програмовані логічні інтегральні схеми (ПЛІС) дають змогу реалізувати процесори зі спеціалізованою системою команд, завдяки чому мінімізуються апаратні витрати, енергоспоживання і зростає продуктивність. У [1] показана ефективність реалізації 8-розрядного мікропроцесора у ПЛІС, у якого мінімізована система команд. Тепер розробка процесорних ядер зі спеціалізованою системою команд підтримується засобами САПР, такими як Synopsys Processor Designer, який використовує методику LISA [2].

Рівень розвитку ПЛІС дає змогу реалізувати в них великі багатопроцесорні системи [1,2]. Але нові серії ПЛІС мають ряд особливостей, що впливають на вибір архітектури модулів, які в них конфігуруються. У нових поколіннях ПЛІС, якщо число транзисторів збільшується учетверо, то об'єм ресурсів для трасування сигналів збільшується лише удвічі. Затримки у лініях зв'язку стали в 1–3 рази більшими за затримки у логічних таблицях (ЛТ) та інших логічних елементах. Як результат, конфігуровані модулі мають бути компактними, щоби їх внутрішні сигнали далеко не поширювались.

Вимозі компактності не відповідають ядра мікропроцесорів з поширеною архітектурою, такі як Nios, Microblaze, Mico32, OpenRISC, MIPSFPGA [3]. Для досягнення прийнятної тактової частоти ці процесори мають п'ять та більше ступенів конвеєра команд. Для виконання кількох незалежних програмних потоків

у такому ядрі реалізуються система переривання та зміни контексту, віртуальна пам'ять, кеш-ОЗП, швидкісний інтерфейс до зовнішньої динамічної пам'яті.

Множина архітектур таких процесорів обмежена 8- та 32-розрядними процесорами. Крім того, більшість процесорів орієнтована на виконання кількох обчислювальних процесів, включаючи роботу операційної системи. На виконання цих процесів потрібні суттєві часові та апаратні витрати [3].

Метою роботи є побудова процесорного ядра яке займає мало місця в ПЛІС і в той же час є швидкодіючим та має зручну систему команд для виконання логічних операцій над послідовностями символів. Для цього був розроблений 16-розрядний процесор RISC-ST2, який є процесорним елементом такої системи.

2. Особливості архітектури процесорного ядра

Архітектура процесорного ядра RISC-ST2 орієнтована на виконання однієї задачі, яка має багато логічних операторів та обробку потоків даних з полями змінної довжини. Це, наприклад, пакування даних за алгоритмом LZW. При необхідності виконання кількох паралельних задач кожна з них може виконуватись на окремому ядрі. Кілька ядер можуть бути об'єднані у систему через свої регістри вводу-виводу та систему переривань.

За основу було взяте ядро RISC-ST, яке описане в [4]. При формуванні структури приймалися до уваги рекомендації проекту-

3. Результати реалізації ядра

Ядро мікропроцесора описане мовою VHDL і не має обмежень для синтезу та конфігурування у ПЛІС будь-якої серії. Модель процесора має вбудований дизасемблер, який спрощує тестування та відлагодження програм. Розроблено програму кросасемблера на мові Java, виходом якої є VHDL-файли блоків пам'яті даних та програм процесора.

У таблиці 1 показані результати синтезу процесора для різних серій ПЛІС та їх порівняння з відомими 16-розрядними процесорами.

Табл.1. Результати конфігурування процесорів у ПЛІС

ПЛІС	Апаратні витрати, ЛТ	Тактова частота, МГц	Прим.
Xilinx Kintex-7	508	237	
Xilinx Artix-7	513	163	
Xilinx Spartan-6	643	144	
Xilinx Spartan-3	861	77	
Altera Max-10	1212	121	
Altera Cyclone-V	495	161	
Altera Stratix-V	546	235	
Xilinx Kintex-7	672	142	1
Xilinx Kintex-7	1387	150	2
Xilinx Spartan-3	3602	74	3

Примітки: 1. Процесор RISC-ST2 з ускладненими командами. 2. Процесор OpenMSP430 [5]. 3. Процесор NanoBlaze [3].

За статистикою, у сучасних ПЛІС на один блок пам'яті припадає, в середньому, 300 ЛТ.

Отже, процесор має доволі невеликі апаратні витрати – стільки, скільки припадає на 2 – 3 блоки пам'яті. Ці витрати є значно меншими, ніж у аналога та у семеро разів менше, ніж у поширеного 32-розрядного MicroBlaze [3]. Наприклад, у порівняно дешевій ПЛІС Artix-35T вміщується до 45 процесорних ядер.

Тактова частота процесора з доданими складними командами є нижчою на 67% від частоти ядра з мінімальним набором команд. До таких складних команд належить команда циклу DJNZ, сигнали при виконанні якої проходять критичним шляхом структури ядра. Але такі команди замінюють собою кілька простіших команд і спрощують програмування.

Висновки

Розроблено ядро RISC-процесора зі спеціалізованою системою команд. Завдяки тому, що процесор орієнтований на виконання лише одного обчислювального процесу та обробки переривань, він має невеликі апаратні витрати та високу швидкодію. Він призначений для виконання алгоритмів безвтратної компресії та керування. При зміні системи команд, яка є нескладною, процесор може бути використаний для інших потреб, наприклад, для цифрової обробки сигналів, розпізнавання образів тощо.

Перелік посилань

1. Сергієнко А.М., Лепеха В.Л. Деякі особливості проектування мікроконтролерів для СНК / А.М. Сергієнко, В.Л. Лепеха // Вісник НТУУ «КПІ», сер. Інформатика, управління та обчислювальна техніка. – Т. 50. – 2009. – с. 70-73.
2. Synopsys Processor Designer. Automating the Design and Implementation of Custom Processors and Programmable Accelerators// Synopsys, Inc. – 2010. – 3 P. – Available at www.synopsys.com.
3. Meyer-Baese U. Digital Signal Processing with Field Programmable Gate Arrays. 4-th Ed. – Springer. – 2014. – 930 p.
4. Сергиенко А.М. VHDL для проектирования вычислительных устройств. –К.: –"ДиаСофт". –2003. –210 с.
5. OpenMSP430 / O. Girard // – Rev. 1.15, May, 19, 2015. –129 P. – Available at www.opencores.org