

ВИКОРИСТАННЯ РЕКОНФІГУРОВАНИХ ЕЛЕМЕНТІВ У БОРТОВИХ СИСТЕМАХ ЗБОРУ Й ОБРОБКИ ДАНИХ

Луценюк А. А.¹, Глухов В. С.², Шендерук С. Г.³

¹ ЛЦ ІКД НАНУ та НКАУ, м. Львів

² НУ “Львівська політехніка”, м. Львів

³ ЛЦ ІКД НАНУ та НКАУ, м. Львів

Описано варіанти побудови та можливості забезпечення надійності бортових систем збору і обробки інформації при використанні програмованих логічних інтегральних схем.

Суттєвим фактором, який характеризує сучасний стан проектування електронних засобів для космічної галузі, є використання програмованих логічних інтегральних схем (ПЛІС). Найширше використовуються радіаційно стійкі одноразово програмовані ПЛІС фірми „Actel” [1], останнім часом пропонується використовувати і багаторазово репрограмовані ПЛІС індустріального виконання фірми „Xilinx” і „Atmel” [2-4].

В залежності від об'єкта, де встановлюється система збору і обробки наукової інформації (СЗНІ), нею можуть вирішуватися різні за складністю задачі з управління і опрацювання потоків інформації. Тому складність системи і її складових частин повинна змінюватися в залежності від вирішуваних задач. У тому числі складність центрального процесора може змінюватися від простого контролера (цифрового автомата) з найпростішим арифметико-логічним пристроєм (АЛП) і RISC-мікропроцесора [5], «однокристалної» мікроЕОМ [6,7] до багаторозрядного RISC-процесора [8]. Найкраще такий процес реалізується на ПЛІС, що дозволяє без зміни конструкції виробу змінювати споживану ним потужність (кількість задіяних апаратних ресурсів) в залежності від складності вирішуваної задачі. Існуючі засоби проектування ПЛІС дозволяють реалізувати усі обрані варіанти ЦП. Для складних задач можливе використання покупних стандартних ЦП.

Вузли структурної схеми (рис. 1), які пропонується будувати на ПЛІС, позначені далі в тексті двома словами, наприклад, ПЛІС ЦП означає, що використаний центральний процесор реалізовано на ПЛІС. Тоді структура СЗНІ може бути утворена у таких варіантах:

- універсальна мікро ЕОМ + ПЛІС СО + пам'ять. Цей варіант може застосовуватись у випадках, коли космічний експеримент передбачає отримання значної кількості інформації, при цьому необхідно виконувати складні задачі управління. Очевидно, що в цьому випадку на даному етапі необхідно використати покупний потужний процесорний модуль. Що стосується задачі обробки інформації, в тому числі з метою ущільнення, то її можна виконати з використанням спецобчислювача, реалізованого на ПЛІС.

- ПЛІС ЦП і ПЛІС СО + пам'ять. Цей варіант можна буде застосувати, коли функції управління, що покладаються на ЦП, досить прості, а алгоритм обробки інформаційних сигналів є складним і вимагає відповідної апаратурної реалізації;
- ПЛІС ЦП, СО і пам'яттю. Цей варіант пропонується використати у випадку, коли функції управління, що покладаються на ЦП та алгоритм обробки інформаційних сигналів досить прості; мова йде про використання SoC (системи на одному кристалі). Цей варіант можна використати у випадках, виконання простих задач на борту КА або у випадках, коли будуть створені фінансові умови для розробки потужного обчислювального ядра на ПЛІС.

Наявність значного числа варіантів побудови СЗНІ не означає, що за ними стоїть необхідність відповідного числа модулів, субмодулів, які би дозволяли шляхом реконфігурування реалізувати перелічені варіанти. Реконфігуруваність модулів і СЗНІ в цілому в широких межах досягається шляхом перепрограмування ПЛІС при невеликому числі модулів.

На рис. 2 представлено варіант структурної схеми реконфігурованого модуля СЗНІ. Сіримися квадратиками обмежена частина схеми, що реалізується на ПЛІС. Цей модуль відіграє основну роль в забезпеченні реконфігурованості СЗНІ. На нього покладаються задачі стикування з науковими приладами через різні інтерфейси та зміна своєї інформаційної потужності в залежності від складності задач, що вирішуються. В процесі реконфігурації можна міняти номенклатури і кількість вузлів, в тому числі контролерів наукових приладів (НП), контролерів каналів зв'язку, тип контролера шини, тип і об'єм пам'яті процесора.

З'єднання усіх вузлів процесора здійснюється за допомогою локальної шини АМВА, можливості і ресурси якої можуть змінюватися при реконфігурації [8, 9].

Контролери НП можуть забезпечувати під'єднання наукових приладів через інтерфейси SPI, LVDS і CAN, використання технології LVDS, а також стиснення наукової інформації згідно з рекомендаціями [10].

Контролери каналів зв'язку забезпечують пересилання і приймання командної інформації і інформаційних даних через інтерфейси RS-232/422/485, CAN на радіоканал чи інші системи космічного корабля. При цьому забезпечується контроль передачі інформації з виправленням помилок за рахунок використання кодів Ріда–Соломона, згорткових кодів та циклічних контрольних сум згідно з рекомендаціями [11, 12].

Контролер локальної шини може забезпечувати управління вузлами реконфігурованого модуля з боку шини ISA (у конструктиві PC/104), що має місце при реалізації СЗНІ за варіантом 1, або використовуватись автономно, як ЦП при реалізації СЗНІ за варіантами 2, 3. При цьому він може бути реалізований як цифровий мікропрограмний 8-розрядний RISC-процесор, ядро однокристалльної ЕОМ чи ядро 32-розрядного RISC-процесора. Останній є "відкритим" процесором і існує у вигляді описаних на мові VHDL модулів (ядер), які можуть бути легко реалізовані на будь-якій ПЛІС.

1. Використання цифрового автомата і АЛП як ЦП, реконфігурованого модуля

Засоби проектування ПЛІС дозволяють задавати програму (алгоритм) роботи нескладних малорозрядних локальних процесорів за допомогою графа автомата зі збереженням результатів роботи у вигляді VHDL-текстів, що дозволяє у подальшому використовувати їх на інших засобах проектування. В ході роботи отримані задовільні результати заміни мікроконтролера AT89C2051 „Atmel” і схем жорсткої логіки на ПЛІС з метою зменшення габаритних розмірів і складності вузлів. При цьому алгоритми, які раніше виконував мікроконтролер, були введені у ПЛІС за допомогою графа автомата. Операційний вузол, який реалізує обмежену кількість операцій, формується зі стандартних вбудованих бібліотечних елементів ПЛІС, для яких так само формуються VHDL-коди.

2. Використання RISC-мікропроцесора як центрального процесора

Серед відомих центральних процесорних елементів, які можуть бути реалізовані на ПЛІС, на увагу заслуговує мікропрограмний мікропроцесор KCPSM (Constant(k) Coded Programmable State Machine [5]).

Програмне забезпечення для даного мікропроцесора створюється на мові Асемблер. За допомогою програми KCPSMBLE.EXE, яка вільно поставляється, здійснюється аналіз тексту програми і його трансляція у машинні коди, які переносяться до пам'яті команд на етапі проектування топології ПЛІС.

Даний RISC-мікропроцесор існує у вигляді бібліотечного елемента, але недоліком даного варіанту є “закритість” вихідних текстів мікропроцесора і крос-засобів розробки програм для нього. Однак, враховуючи малі апаратні витрати і його високу продуктивність, доцільно провести роботу з створення відкритого VHDL-опису аналогічного RISC-процесора і проектуванню крос-засобів для нього.

3. Використання “однокристалної” мікроЕОМ

Однокристалні мікроЕОМ типу 8051 користуються заслуженою популярністю серед розробників цифрових схем, тому зрозумілою є поява бібліотечних елементів ПЛІС, які є функціональними аналогами таких мікроЕОМ. Реалізація таких “однокристалних” мікроЕОМ на ПЛІС дозволяє використовувати існуючі засоби проектування програмного забезпечення і відповідних доробок. Однак при цьому необхідні великі апаратні витрати – велика частка ресурсів кристалу ПЛІС, яка займається даною мікроЕОМ. Для дослідженого варіанту мікроЕОМ займає 42 % ресурсів площі ПЛІС і 93 % її контактів [7]. Тактова частота може перевищувати 30 МГц. Технічні характеристики ЕОМ для максимальної конфігурації наведені нижче. Оскільки дана мікроЕОМ вимагає пам'ять великого об'єму, в складі локального процесора необхідно передбачити додаткові вузли пам'яті поза ПЛІС.

4. Використання багаторозрядного RISC-процесора.

В Європейському космічному агентстві [13] розробляється ядро 32-розрядного RISC-процесора LEON [8]. Ядро LEON є SPARC* [14] сумісним цілочисельним пристроєм, розробленим для майбутніх космічних потреб (*SPARC є зареєстрованою торговою маркою SPARC International. SPARC [scalalable processor architecture]— нарощувана архітектура процесора (архітектура процесорів з змінною продуктивністю).* Він розроблявся у вигляді висококонфігурованої, синтезованої VHDL-моделі. Для просування стандарту SPARC і аби дозволити розвиток пристроїв system-on-a-chip (SoC), використовуючи ядра SPARC, Європейська космічна агенція зробила легкодоступним повний програмний код за ліцензією LGPL GNU [15]. Ядро LEON було значним чином протестовано згідно документації архітектури SPARC V8 і стандарту IEEE-P1754 (SPARC), проте не було формально протестовано і сертифіковано SPARC international як SPARC-V8-сумісне.

LEON також існує у відмовостійкій версії, яка може знаходити та виправляти помилки single-event upset (SEU) в довільному регістрі чи елементі RAM. Це зроблено виключно на рівні апаратури без програмного втручання. Видатки по площі для відмовостійких функцій складають приблизно 30 %, в той час як часова затримка становить біля 5 %. Функції відмовостійкості дають можливість використовувати LEON в жорстких умовах без потреби розробляти специфічні SEU-hardend елементні бібліотеки. Відмовостійка VHDL-модель LEON може бути ліцензована ESA. Оскільки даний RISC-процесор вимагає пам'ять великого об'єму, в складі ЦП необхідно передбачити додаткові вузли пам'яті, які повинні знаходитися поза ПЛІС.

Для реалізації перелічених вище процесорних вузлів як основний елемент реконфігурованого процесора обрана ПЛІС сімейства „Virtex”. Кристали цього сімейства відзначаються великою логічною ємністю, забезпечують системну частоту до 200 МГц. Архітектура характеризується широкою різноманітністю високошвидкісних трасувальних ресурсів, наявністю виділеного блочного ОЗП та розвиненою логікою прискореного переносу. Технологія „Virtex” підтримує значну кількість стандартів вводу-виводу. Важливою особливістю цього сімейства є радіаційно стійкий варіант його виконання.

Результати реалізації на ПЛІС XCV300 центральних процесорних елементів різної складності і продуктивності наведені у таблиці.

Використання ПЛІС для побудови СЗНІ дозволяє реалізувати динамічне самотестування і реконфігурування у випадку виявлення відмов.

В основу динамічного (в процесі роботи) самотестування покладено такі властивості ПЛІС:

- використання системи граничного сканування JTAG IEEE 1149.1 – 1990;
- можливість багаторазового завантаження конфігурації ПЛІС;
- можливість зворотного читання і перевірки конфігурації ПЛІС;
- можливість зміни завдання під час проектування топології ПЛІС, які CLB (конфігуровані логічні блоки) можна використовувати, а які не можна.

Таким чином алгоритм динамічного самотестування набуває такого вигляду:

- завантаження у ПЛІС тестової топології (структури, схеми), призначеної лише для визначення працездатності CLB і комутаторів;
- перепроєктування робочої топології ПЛІС з врахуванням непрацездатних CLB і комутаторів;
- завантаження робочої топології ПЛІС.

Даний алгоритм нагадує процедуру тестування поверхні гнучких дисків для виявлення і маркування непрацездатних блоків. Основною задачею при реалізації даного алгоритму є розробка спеціальних тестових структур ПЛІС. Можливі такі варіанти реалізації даного алгоритму:

- тестування вузлів ПЛІС для варіанту резервування на рівні декількох ПЛІС;
- тестування незайнятої у даній реалізації частини ПЛІС (резервування в самій ПЛІС).

Таким чином, одним із основних підходів до реалізації СЗНІ є її репрограмованість, яка забезпечується широким використанням ПЛІС. Очевидно, що алгоритм динамічного самотестування і реконфігурації вимагає двостороннього зв'язку з КА.

1. Lessons Learned from FPGA Developments. EUROPEAN SPACE AGENCY CONTRACT REPORT.//Technical Report FPGA-001-01,Version 0.0, April 2002.
2. Q-Pro-for the Aerospace & Defense market by Horward Bogrow, Marketing Manager, Aerospace & Defense Products, Xilinx, Inc., [//howard@xilinx.com](mailto:howard@xilinx.com)// Xcell No.4/1999.
3. Adaptive Instrument Module - A Reconfigurable Processor for Spacecraft Applications. Richard F. Conde, Ann Garrison Darrin, F. Charles Dumont, Phil Luers, Steve Jurczyk, Dr. Neil Bergmann and Dr. Anwar Dawood. Xilinx Application Note rp004e5.pdf, v1.0, 9/99.
4. Integrated Circuits for Aerospace, Dual-use Radiation Hard.@@Atmel Nantes S.A.2001.
5. Application Note: Virtex Series and Spartan-II family. 8-Bit Microcontroller for Virtex Devices. Ken Chapman. XAPP213 (v1.1) October 4, 2000. www.xilinx.com
6. Flip805x-PR Core. Product Specification. Dolphin Integration. January 10, 2000. www.dolphin.fr
7. C8051 Synthesizable HDL Core Specification. Version 3.01. Copyright 1999-2001 Evatronix SA. www.evatronix.pl
8. European Space Agency. European Space Research and Technology Center. TOS-ESD/JG/508 Issue 1.0. November 2000. LEON/AMBA VHDL model description (leon-2.2). Prepared by J. Gaisler.
9. AMBA™ Specification (Rev 2.0) © Copyright ARM Limited 1999. All rights reserved. ARM IHI 0011A. <http://www.arm.com>
10. REPORT CONCERNING SPACE. DATA SYSTEM STANDARDS. LOSSLESS DATA COMPRESSION. TELEMETRY CHANNEL CODING. CCSDS 120.0-G-1. GREEN BOOK. May 1997.
11. RECOMMENDATION FOR SPACE. DATA SYSTEM STANDARDS. PACKET TELEMETRY. CCSDS 102.0-B-5. BLUE BOOK. November 2000.
12. RECOMMENDATION FOR SPACE. DATA SYSTEM STANDARDS. TELEMETRY CHANNEL CODING. CCSDS 101.0-B-5. BLUE BOOK. June 2001.
13. <http://www.esa.int>
14. <http://www.sparc.org>
15. <http://www.gnu.org>
16. <http://www.estec.esa.nl/wsmwww/leon/leonccs.pdf>
17. <http://www.estec.esa.nl/wsmwww/leon/dhry.c>

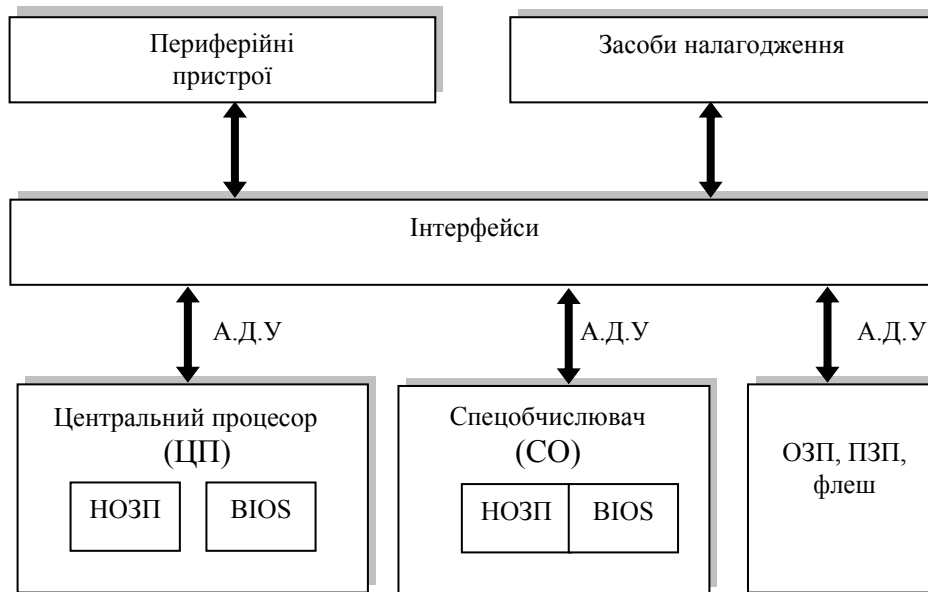


Рис. 1. Узагальнена структурна схема СЗН.

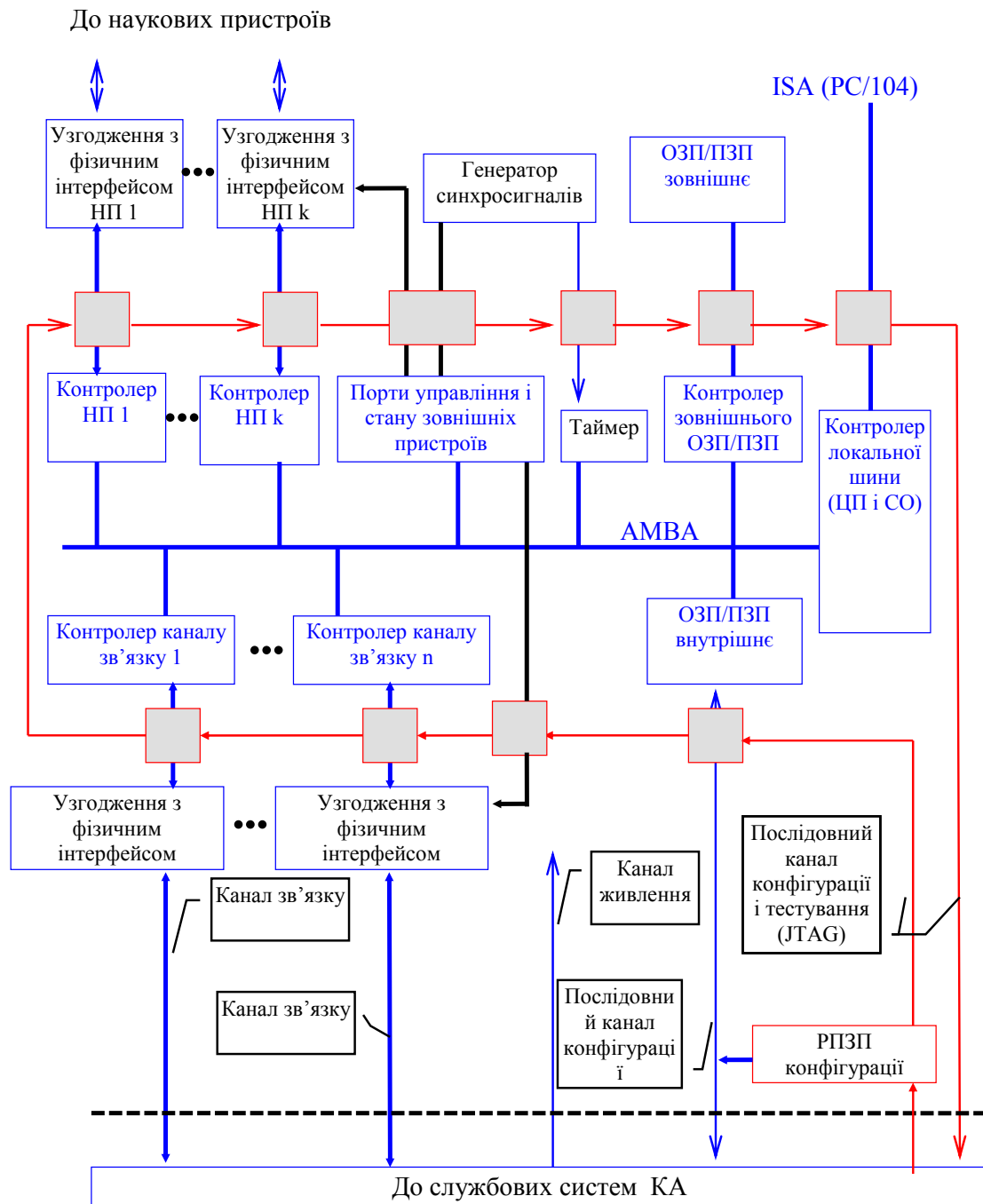


Рис. 2. Варіант структурної схеми реконфігурованого модуля СЗНІ.

Таблиця. Результати реалізації на ПЛІС XCV300 центральних процесорних елементів

Тип централь-ного процесорно-го елемента	Розрядність даних центрального процесорного елемента	Використання площі кристалу ПЛІС, %	Макс. тактова частота, МГц	Основні операції центрального процесорного елемента	Периферійні пристрої в складі центрального процесорного елемента
КСРС М	8	5	50	Арифметичні, логічні, зсуву, управління, вводу-виводу, переривання	По дві 8-розрядні шини даних і команд, пам'ять команд 256 байт, контролер переривань
8051	8	42	30	Арифметичні, логічні, операції з бітами, множення, ділення	Чотири 8-розрядних порти вводу-виводу, два 16-розрядних таймери, два інтерфейси SPI (UART), контролер переривань, внутрішня ОЗП 256 байт
LEON	32	75	45	Арифметичні, логічні, операції з бітами, множення, ділення	Два таймери, сторожовий таймер, два приймачі-передавачі UARTs, паралельний 16-розрядний порт вводу-виводу