

УДК: 004.383.3

РОЗРОБКА І ДОСЛІДЖЕННЯ СТРУКТУРИ ЛОГІЧНОГО АНАЛІЗАТОРА ЦИФРОВИХ СИГНАЛІВ

Блінов Ю.В. , Зінченко Ю.Є.

Донецький національний технічний університет

Логічний аналізатор являє собою електронний інструмент, який відображає сигнали в цифрових схемах, які занадто швидкі, щоб спостерігати і представляє їх користувачеві, так, що користувач може легко перевірити роботу цифровою системи з великою точністю. Вони зазвичай використовуються для збору даних в системах, які мають занадто багато каналів, щоб бути дослідженими за допомогою осцилографа [3].

В даний час є дві категорії логічних аналізаторів:

- Перша категорія пристроїв являє собою автономну одиницю, яка має дисплей, органи керування, керуючий комп'ютер, і кілька слотів для підключення досліджуваної апаратури [4].
- До другої категорії відносяться пристрої на базі ПК. Цей вид логічних аналізаторів збирає данні, та передає їх до клієнтської програми на ПК. Ці пристрої є менш дорогими, ніж перша категорія пристроїв. Вони зазвичай набагато менші, тому що не мають потреби в дисплеях та органах керування, таких як клавіатура [4].

В статті розглянуто логічний аналізатор другого виду, що є приставкою до ПК на базі FPGA плати Spartan-3E.

З рис. 1 видно, що аналізатор сигналів є комплексним пристроєм, що складається з двох частин.

1. Персональний комп'ютер з розробленим програмним забезпеченням слугує для завдання параметрів захвату сигналів з досліджуваного пристрою, для прийому та обробки значень каналів.
2. Приставка у виді FPGA плати слугує для відстеження умов початку опитування та збереження захоплених значень з досліджуваного пристрою.

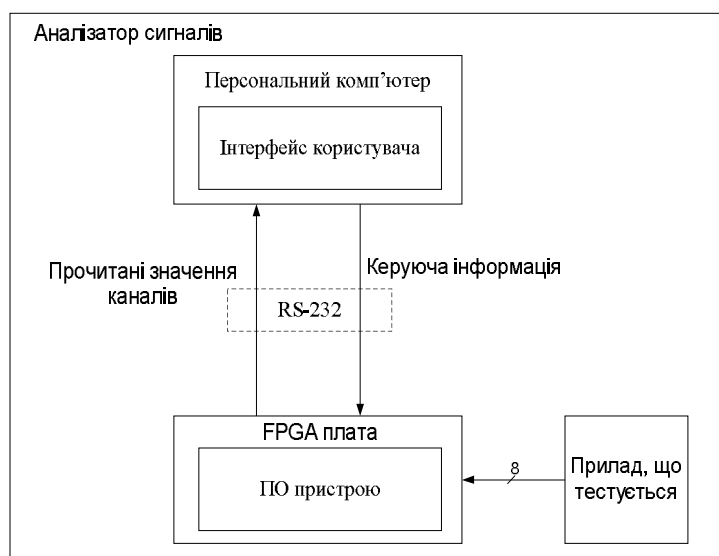


Рисунок 1 — Загальна структура проекту

Структурна схема FPGA-пристрою аналізатора сигналів наведена на рис. 2.

До складу структурної схеми увійшли наступні компоненти:

Керуюче ядро — блок працюючий за принципом машини з кінцевою кількістю станів (FSM, Finite State Machine). Керуюче ядро призначене для організації коректної взаємодії між іншими складовими частинами пристрою, та взаємодією логічного аналізатора с клієнтською програмою на ПК;

Блок синхронізації — генерування синхросигналу заданої частоти на базі сигналу 50MHz;

UART — організація обміну інформацією між ПК та пристроєм ;

Дешифратор команд — в функції цього блоку входить аналіз команд з ПК та формування відповідних керуючих сигналів.

3. Черга семплів — черга заданої довжини, призначення якої полягає в збереженні передісторії до появи потрібної комбінації на виходах приладу, що тестується, це дозволяє побачити стан приладу в попередніх тактах до того, як задана подія трапилася;

4. Блок пошуку початкового семплу — в залежності від встановленої маски, встановленого значення та поточного стану виходів приладу формує сигнал «НІТ», що сигналізує про появлення потрібної комбінації на виходах об'єкту діагностики (ОД);

5. Пам'ять — призначена виключно для збереження семплів отриманих на виходах ОД, при виявленні на них потрібної комбінації, до пам'яті вони записуються не на пряму, а проходячи через чергу семплів;

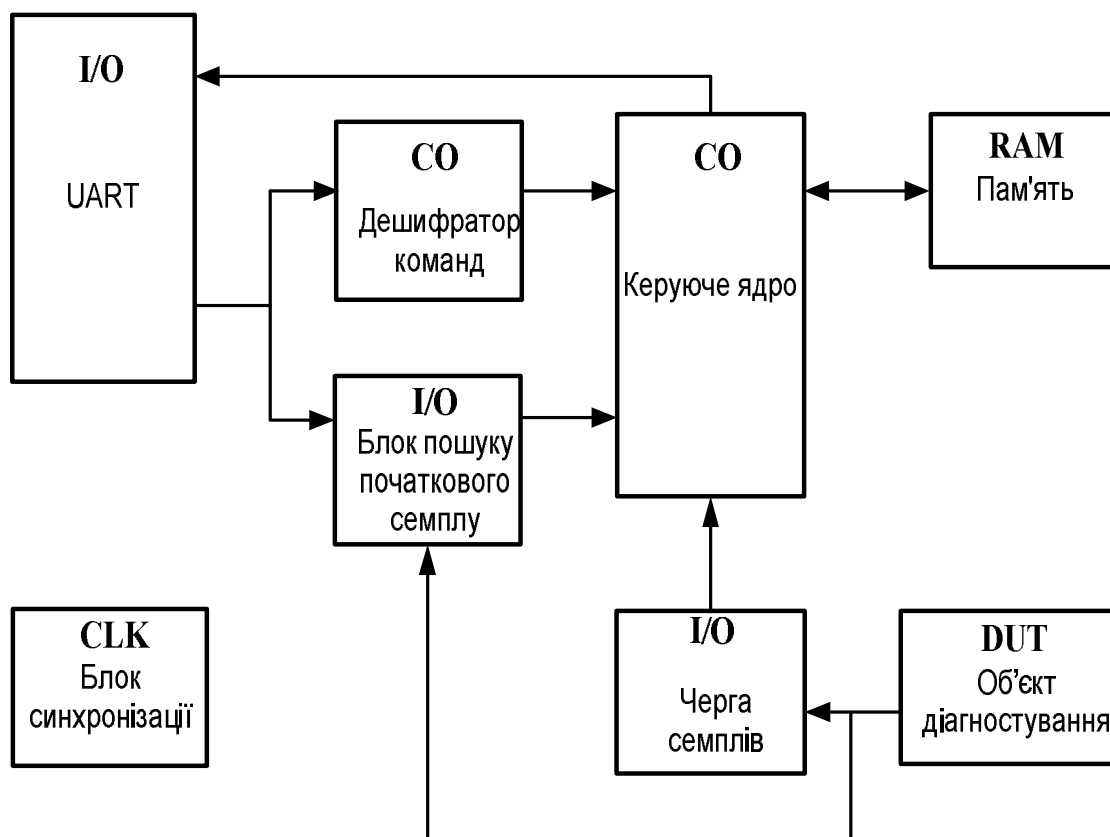


Рисунок 2 — Структурна схема аналізатора сигналів на базі FPGA плати

Розглянемо алгоритм функціонування блоку контролю (рис. 3). Згідно з алгоритмом виконується наступне:

1. стан RESET — відбувається встановлення початкових значень сигналів потрібних для коректної роботи;
2. стан INIT — в цьому стані відбувається встановлення маски, значення, що треба знайти та кількості семплів, що треба зберегти до пам'яті. Коли всі необхідні параметри успішно записано, то контролер переходить до стану WAIT_HIT. Також, находячись в стані INIT, контролер здатен виконати команду відповіді на запит спеціальним кодом з 4 байт;
3. стан WAIT_HIT — в цьому стані контролер перебуває поки сигнал HIT не буде встановлено в 1, це трапиться коли на виходах ОД з'явиться потрібна комбінація. Після цього контролер переходить до стану SAMPLE;
4. стан SAMPLE — в цьому стані контролер виконує зберігання семплів з виходу черги семплів до блочної пам'яті (порт А), з початку зберігається 16 значень передісторії, тобто значень до появи потрібної комбінації. Коли збережено задану кількість семплів, то контролер переходить до стану SEND;
5. стан SEND — контролер читає дані з порту Б блочної пам'яті (4 байта за 1 раз) та відправляє їх до ПК користувача. Виконавши це пристрій переходить в стан очікування INIT.

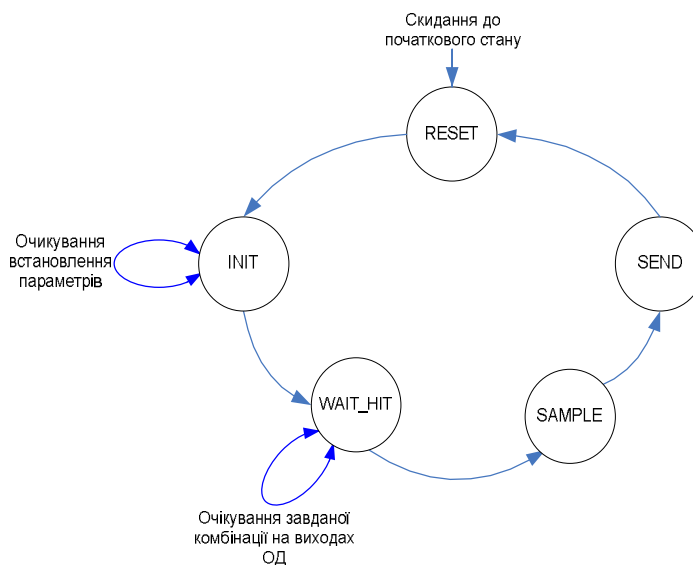


Рисунок 3 — Діаграма станів логічного аналізатору сигналів

Таблиця 1 – Технічні параметри розробленої структури

Параметр	Використано	Залишилось	Використано %
Кількість Slices	195	4656	4%
Кількість Slice Flip Flips	164	9212	1%
Кількість 4 входних LUTs	332	9312	3%
Кількість IOBs	70	223	30%
Кількість BRAMs	20	20	100%
Clock period: 6.799ns (frequency: 147.080MHz)			

Виходячи з таблиці 1 максимальна частота з якою може працювати пристрій 147.080МН, підвищити частоту семплування можна за допомогою швидкодіючої схеми регістру зсуву (рис. 4).

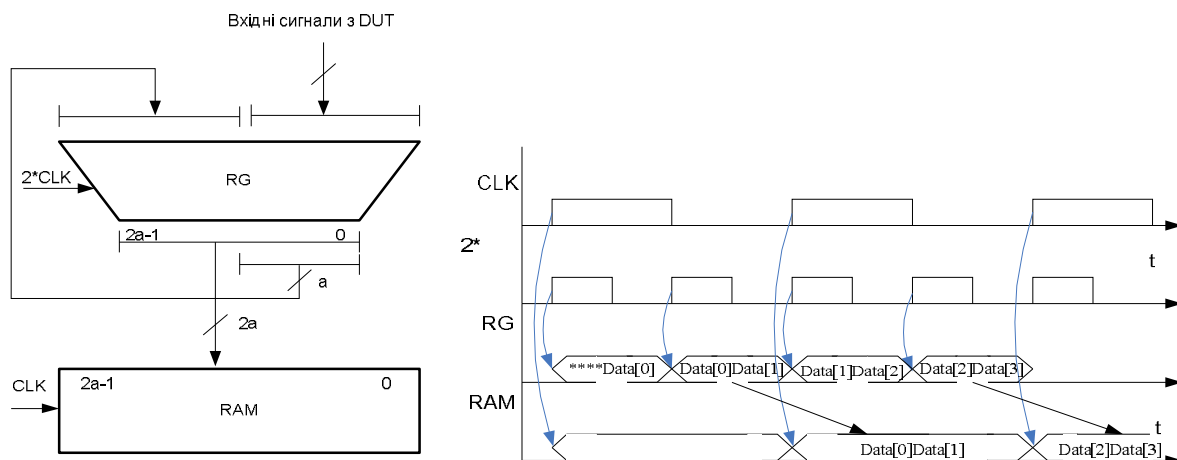


Рисунок 4 – Використання регістру зсуву для підвищення швидкодії

Результатом досліджень, проведених в рамках підготовки цієї статті, стала експериментальна поведінкова модель аналізатору сигналів на базі відлагоджувальної плати Xilinx Spartan-3E.

Виходячи з отриманих результатів, можна зробити висновок, що експериментальні структури, отримані під час розробки є задовільними. Також потрібно відмітити те, що вони є переносимими на інші FPGA плати. Це обумовлено тим, що для їх реалізації використано мову програмування VHDL.

Література

- [1] Spartan-3E FPGA Family Data Sheet [Електронний ресурс] — Режим доступу: <http://www.xilinx.com/support/documentation/spartan-3e.htm>
- [2] Agrawal V.D Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits / Agrawal V.D, Bushnell M. L. // Boston: Kluwer Academic Publishers, — 2000, 650 p
- [3] Рувинова Э. Логические анализаторы – цифровые детективы [Електронний ресурс] — Режим доступу: http://www.electronics.ru/pdf/1_2003/07.pdf
- [4] Перцовский М., Воробьев Е., Трифонов А. Применение логических анализаторов в тестировании цифровой техники [Електронний ресурс] — Режим доступу: <http://www.cta.ru/cms/f/?/366656.pdf>