

АЛГОРИТМ ПОБУДОВИ ПРОГРАМНИХ РЕАЛІЗАЦІЙ ІМІТАЦІЙНИХ МОДЕЛЕЙ ЧИСЛО-ІМПУЛЬСНИХ ФУНКЦІОНАЛЬНИХ ПЕРЕТВОРЮВАЧІВ

Стаття присвячена розробленню алгоритму побудови програмних реалізацій імітаційних моделей число-імпульсних функціональних перетворювачів.

The given article is dedicated to the development of the algorithm for constructing software implementations of simulation models of pulse-numerical functional convertors.

1. ВСТУП

В роботі [1] показано можливість та переваги використання методів імітаційного моделювання апроксимуючих число-імпульсних функціональних перетворювачів (ЧІФП) в процесі їх параметричного синтезу на ЕОМ. Проведені автором дослідження виявили відсутність інженерних методик розробки програмних реалізацій імітаційних моделей ЧІФП. Тому задача розробки алгоритму побудови таких реалізацій є актуальною.

2. РОЗРОБЛЕННЯ АЛГОРИТМУ

Вихідними даними для алгоритму побудови програмної реалізації імітаційної моделі ЧІФП будемо вважати проміжне відображення структури ЧІФП суперпозицію графів поширення число-імпульсного коду (ЧІК) та двійкового позиційного коду (ДПК) [1]. Представимо алгоритм імітаційного моделювання ЧІФП у вигляді блок-схеми, кожний блок якої адекватно моделює реакцію певного структурного елемента на поодинокий вхідний імпульс. Послідовність процедур моделювання, прийняту в імітаційній моделі, будемо позначати напрямленими міжблоковими зв'язками.

В теорії структурного програмування така блок-схема розглядається як різновид орієнтованого графа [2]. Порівнюючи цей граф з графом поширення (ЧІК), можна простежити певні функціональні аналогії між їхніми компонентами. Наприклад, вершини графа поширення ЧІК відображають операції, здійснювані структурними елементами, а блоки у блок-схемі моделюють роботу цих структурних елементів. Ця

¹ Національний університет „Львівська політехніка”

обставина використана при розробленні алгоритму імітаційного моделювання ЧФП.

Алгоритм, блок-схема якого зображена на рис.1, призначено для побудови імітаційних моделей ЧФП, для яких граф поширення ЧК є орієнтованим деревом. Як показав аналіз, клас таких пристроїв охоплює значну частину існуючих практичних розробок.

В основу роботи алгоритму покладено метод перегляду дерева у прямому порядку відомий, також, як перегляд з відходом назад. Під час просування в напрямі орієнтації ребер для усіх переглянутих вершин здійснюється побудова відповідних функціональних блоків блок-схеми та частини зв'язків між ними. Решта міжблокових зв'язків будується при відході назад.

Для побудови блок-схеми застосовуються блоки чотирьох типів. Блок першого типу позначає початок блок-схеми і має один вихід. Блок другого типу позначає завершення блок-схеми і має лише вхід. Блоки третього типу будуються для вершин з нульовим додатним степенем. Такі блоки мають один вхід і один вихід. Блоки четвертого типу будуються для вершин з ненульовим додатним степенем. Їх особливістю є наявність одного входу і двох виходів. Розгалуження по першому виходу відповідає появі імпульса на імпульсному виході відповідного структурного елемента, розгалуження по другому виходу – його відсутності.

Надалі, для зручності, блок, збудований для деякої вершини v_i будемо позначати $B(v_i)$, а k -тий вихід такого блоку - $B(v_i)_k$, де $k = 1$ для виходів блоків першого та третього типів і першого виходу блоків четвертого типу, $k = 2$ для другого виходу блоків четвертого типу. Відповідно до цього, направлений зв'язок від k -го виходу деякого блоку $B(v_i)$ до виходу деякого іншого блоку $B(v_j)$ будемо позначати як $B(v_i)_k \rightarrow B(v_j)$.

Розглянемо роботу алгоритму побудови блок-схеми імітаційної моделі ЧФП детальніше.

У блоці 1 (див. рис.1) виконується початкова ініціалізація змінних v_1 і v_2 , які мають зміст вершин графа, після чого здійснюється побудова відповідних блоків $B(v_1)$ та $B(v_2)$.

Призначенням змінної v_1 є зберігання останньої, знайденої на даний момент в процесі перегляду, вершини. Допоміжна змінна v_2 використовується, зокрема, при побудові міжблокових зв'язків вигляду $B(v_i)_2 \rightarrow B(v_j)$.

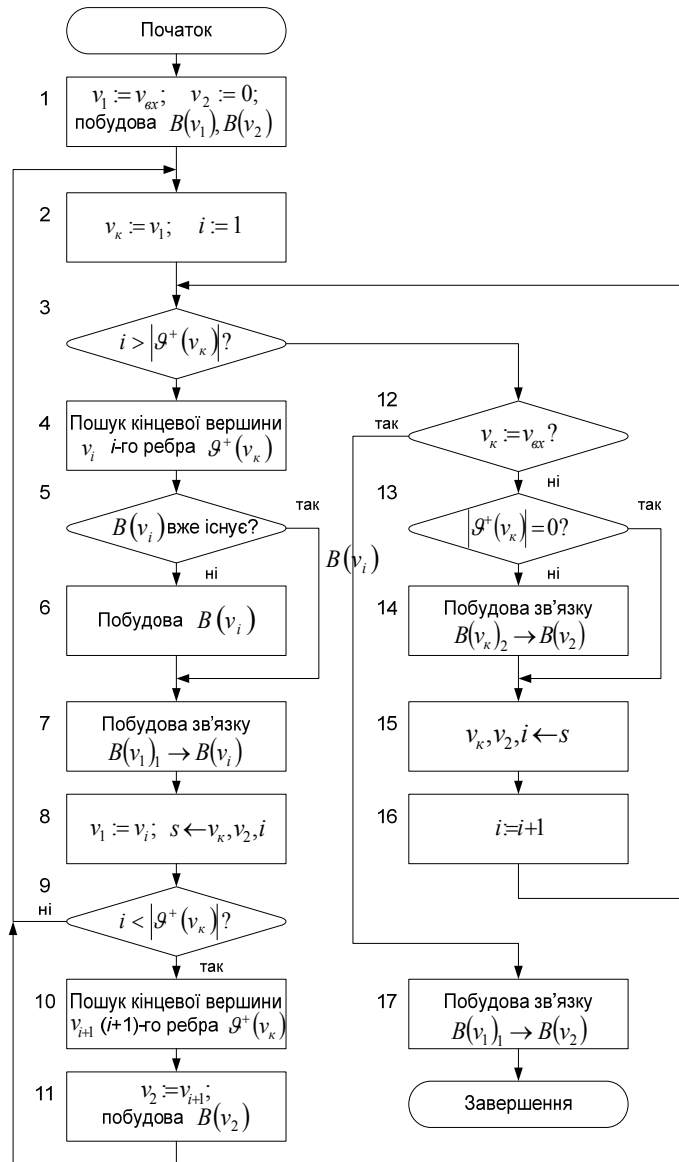


Рис. 1. Алгоритм побудови блок-схеми імітаційної моделі ЧІФП

Очевидно, що результатом виконання блоку 1 для заданих значень ініціалізаторів буде побудова початкового та кінцевого блоків блок-схеми імітаційної моделі.

Алгоритм перегляду дерева реалізовано за допомогою блоків 2, 3, 4, 8, 12, 15, 16. У блоці 2 корінь v_k , який визначає піддерево, що буде переглядатися на даному кроці, переноситься у вершину v_1 . У цьому ж блоці задається номер i ребра, з якого почнеться перегляд зірки $\mathcal{G}^+(v_k)$. Далі, у блоці 4 здійснюється пошук кінцевої вершини v_i i -го ребра зірки $\mathcal{G}^+(v_k)$. У блоці 8 у знайдену вершину переноситься вершина v_1 , а значення змінних v_k, v_2 та i заносяться на вершину стеку s для подальшого зберігання.

Легко побачити, що в результаті циклічного виконання блоків 2, 4, 8 вершини v_k і v_1 переміщуються від кореня дерева у напрямі орієнтації ребер. Паралельно, для кожної переглянутої вершини v_i в блоці 6 виконується побудова блоку $B(v_i)$, а в блоці 7 – міжблокового зв'язку вигляду $B(v_1)_1 \rightarrow B(v_i)$. Можливість повторної побудови одного і того ж блоку усувається за допомогою блоку 5.

В блоках 9, 10, 11 значення змінної v_2 при необхідності оновлюється і для нової вершини будується відповідний блок. З аналізу умови блоку 9 видно, що ці дії виконуються, якщо додатний степінь кореня v_k піддерева, яке переглядається на даний момент, більший за одиницю, а в зірці $\mathcal{G}^+(v_k)$ лишилися не переглянуті ребра.

По досягненні вершини, що є листом, починається процес відходу назад. Момент зміни напрямку перегляду визначається у блоці 3, оскільки для вершини з нульовим додатним степенем, тобто, для листа дерева, умова блоку 3 завжди виконується.

Під час відходу назад, у блоках 12 і 13 для вершин з ненульовим додатним степенем здійснюється побудова зв'язків вигляду $B(v_k)_2 \rightarrow B(v_2)$, які відображають розгалуження по яких піде виконання алгоритму імітаційного моделювання при відсутності імпульсів на виходах структурних елементів. Така побудова блок-схеми дозволяє уникнути моделювання тих структурних елементів, на імпульсні виходи яких не надходять імпульси, що значно підвищує швидкість імітаційного моделювання.

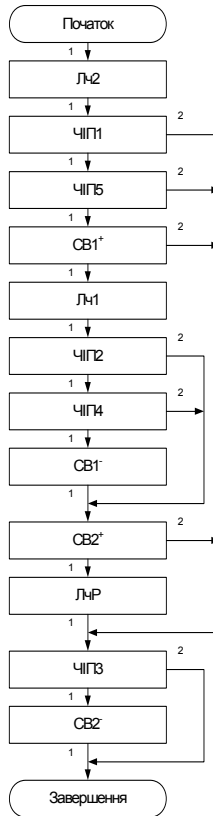


Рис. 2. Блок-схема імітаційної моделі

Власне відхід назад здійснюється наступним чином. У блоці 15 значення змінних v_k, v_2 та i замінюється значеннями, що знімаються з вершини стеку s , внаслідок чого вершина v_k переноситься на один рівень в напрямку до кореня дерева у вершину, яка вже переглянута. У блоці 16 визначається нове, ще не переглянуте ребро зірки $\mathcal{A}^+(v_k)$. Якщо таке ребро існує, а це визначається у блоці 3, то знову ініціюється процедура перегляду дерева. В протилежному випадку процес відходу назад повторюється.

Момент завершення алгоритму побудови імітаційних моделей ЧІФП визначається блоками 3 і 12. Критерієм завершення обрано факт переносу вершини v_k у вершину v_{Bx} при умові, що усі ребра зірки

$\mathcal{G}^+(v_{Bx})$ вже переглянуті. Кінцевою дією алгоритму є побудова у блоці 17 зв'язку вигляду $B(v_1)_1 \rightarrow B(v_2)$, а саме, зв'язку між виходом 1 блоку, збудованого для останньої знайденої вершини з нульовим додатним степенем, і входом блоку завершення блок-схеми.

На рис.2 зображено блок-схему імітаційної моделі, отриманої за допомогою розглянутого алгоритму для проміжного відображення структурної схеми кусково-апроксимуючого ЧФП, представленого в роботі [1]. Для зручності, блоки у складі блок-схеми позначені назвами структурних елементів, роботу яких вони моделюють: число-імпульсні помножувачі ЧП1-ЧП5, лічильники Лч1, Лч2, лічильник результату ЛчР, схеми віднімання імпульсних послідовностей СВ1, СВ2.

4. ВИСНОВКИ

Запропонований алгоритм дозволяє підвищити ефективність процесу створення адекватних програмних реалізацій імітаційних моделей ЧФП і може бути використаний у складі спеціалізованих систем автоматизованого проектування.

1. Отенко В. Математичне моделювання число-імпульсних функціональних перетворювачів // Комп'ютерні технології друкарства. Зб. наук. праць. – Львів: УАД, 2008. - №20. – С. 177-181. 2. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов: Пер. с англ. - М.: Мир, 1981. - 368 с.