

## ЦИФРОВА ОБРОБКА СИГНАЛІВ НА ПЛІС

Руденок С.І,

Науковий керівник – к.т.н., доцент Зубков О.В.

Харківський національний університет радіоелектроніки

61166, Харків, просп. Науки, 14, каф. Мікропроцесорних технологій і

систем, тел. (057) 702-02-29

e-mail: serhii.rudenok@nure.ua.

It is now difficult to name a branch of science and technology that does not use Digital Signal Processing (DSP). The most active DSP is used in transport, radio engineering, communications, television, telecommunications, measuring and household appliances. The tasks of the DSP are characterized by high productivity of the means that solve them, as well as a number of features of the DSP algorithms, such as data streaming, a special mathematical apparatus, widespread.

Зараз важко назвати галузь науки і техніки, де не використовується цифрова обробка сигналів (ЦОС). Найбільш активно ЦОС застосовується у транспорті, радіотехніці, зв'язку, телебаченні, телекомунікаціях, вимірювальній та побутовій техніці. Задачі ЦОС відрізняються високою продуктивністю засобів, що їх вирішують, а також рядом особливостей алгоритмів ЦОС, таких як потокова обробка даних, особливий математичний апарат, широке розповсюдження.

Цифрова обробка сигналів має справу з перетвореннями сигналів, які є цифровими. Основні закономірності ЦОС визначені для дискретних сигналів і уточнюються для цифрових сигналів лише на етапі реалізації ЦОС у конкретному пристрої. Такі уточнення стосуються вибору розрядності даних, точності результатів, перевірки надійності обчислень, але істотно не впливають на властивості алгоритму.

Алгоритми ЦОС мають періодичну природу і полягають в обробці майже безперервних потоків даних. Найбільш природно такі алгоритми задавати на потоковій моделі. На цій моделі основана універсальна високорівнева мова VHDL. Досі ця мова використовувалась лише для моделювання та опису дискретних систем. Але вона має ряд неоціненних переваг у порівнянні з традиційними мовами, якими часто описують алгоритми ЦОС.

Часто алгоритм ЦОС представляють як сигнальний граф, рідше — як граф синхронних потоків даних (ГСПД). Два потоки у графовій моделі є синхронними, якщо є взаємна відповідність між усіма даними в одному та іншому потоках. Наприклад, у синхронних потоках можна перенумерувати дані й тому потоки  $y(n)$  та  $x(n)$  є синхронними так як є відповідність між  $n$ -ми відліками в них. Причому  $n$  розглядається як номер циклу дискретизації або тактового періоду, чи ітерації алгоритму.

Оскільки, у переважній більшості алгоритмів ЦОС сигнальні потоки

є синхронними, тому такі алгоритми можна представити ГСПД. Якщо у результуючому потоці наявність даних умовно залежить від вхідного потоку, то такі потоки можуть бути несинхронними. Це, наприклад, потоки в компресорі сигналів, який замінює ланцюжки нульових відліків кодами довжини цих ланцюжків.

Опис ЦОС мовою VHDL відбувається наступним чином: задаються алгоритми, що представлені на графовій моделі потоків даних загального виду. При цьому відсутність блокувань алгоритму перевіряється на етапі компіляції та під час виконання алгоритму у VHDL-симуляторі. ГСПД є різновидом потокової моделі, на яку накладено певні обмеження. Ці обмеження мають відображення в описі алгоритму мовою VHDL.

Основним обмеженням є те, що вершина графа повинна видавати результат при кожному своєму запуску. Тобто результат  $y(n)$  має такий самий номер або номер зі сталим зміщенням, що й вхідне дане  $x(n)$ . Тоді ці сигнали - вхідні дані й результат - будуть синхронними. Для цього елемент затримки повинен моделюватись оператором процесу, що запускається за синхросигналом, який є єдиним для усієї моделі. Для того, щоб елементи затримки, що стоять у замкнених циклах, мали певне початкове значення, оператори процесу повинні мати послідовні оператори початкового встановлення затриманих сигналів. Ці умови прийняті до уваги при формуванні відповідності між елементами ГСПД та операторами VHDL-програми (табл.1).

Таблиця 1 – Представлення елементів ГСПД у VHDL програмі

Елемент алгоритму	VHDL-програма
Сигнал $u(n)$	<b>signal</b> u: real;
Вхідний та вихідний порти, джерело та приймач сигналів $x(n), y(n)$	<b>port</b> (clk, rst: in std_logic; x: in real; y: out real);
Затримка $u(n)$ на 1 цикл: $v(n) = u(n-1)$	<b>process</b> (clk,rst) <b>begin</b> <b>if</b> rst = '1' <b>then</b> v<=0.0; <b>elsif</b> clk='1' <b>and</b> clk'event <b>then</b> v<=u; <b>end if</b> ; <b>end process</b> ;
Додавання сигналів $y(n) = a(n) + b(n)$	y<=a+b;
Множення сигналу на константу $y(n) = a x(n)$	або <b>generic</b> (a:real:=0,9876); <b>constant</b> a:real:=0,9876; ... y<=a*x;

#### Перелік джерел

1. А.М. Сергієнко, Ю.М. Виноградов, Т.М. Лесик, Цифрова обробка сигналів. Комп'ютерний практикум мовою VHDL. К.: НТУУ «КПІ», 2012.-104с.
2. Стешенко В. ПЛІС фірми ALTERA: Проектування пристроїв обробки сигналів. – М.: Додека, 2000. – 128 с