Практичне заняття № 2.
Паралельні алгоритми розв'язання одновимірної задачі цифрової фільтрації.

Формулювання задачі цифрової фільтрації.

Розглянемо одновимірну *задачу цифрової фільтрації* (ЗЦФ), яка полягає у виконанні k переобчислень масиву значень n змінних за формулою:

$$x_i = \sum_{s=-m}^{m} x_{i+s} \cdot f_s \ , \ i = \overline{1,n}. \tag{1}$$

Тут переобчислення згладжування здійснюються через рухоме вікно розміром $2m+1$. При цьому в (1) значення $x_{1-m}, x_{2-m}, \dots, x_0$; $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ та вагові коефіцієнти $f_{-m}, f_{-m+1}, \dots, f_m$ – задані константи. Вагові коефіцієнти вибираються такими, щоб їх сума дорівнювала 1. Зазвичай на практиці $n >> m$.

У більшості випадків ЗЦФ доводиться розв'язувати в *режимі реального часу* з метою

попереднього опрацювання сигналів, зображень;
великих масивів експериментальних, пошкоджених,
спотворених даних тощо,

тому для високошвидкісної фільтрації необхідно використовувати паралельні алгоритми, орієнтовані на реалізацію на високопродуктивних обчислювальних засобах.

Послідовний алгоритм розв'язання ЗЦФ.

Стандартний послідовний алгоритм розв'язання сформульованої ЗЦФ має вигляд:

$$\begin{array}{l} FOR\ \ j = \overline{1,k}\ \ DO \\ \{\ FOR\ i = \overline{1,n}\ \ DO \\ \ \ \{\ \ p = 0 \\ \ \ \ \ FOR\ s = \overline{-m,m}\ \ DO \\ \ \ \ \ \{\ \ p = p + x_{i+s} * f_s\ \ \} \\ \ \ \ \ \ \ x_i = p\ \ \}\ \}. \end{array} \tag{2}$$

З цього алгоритму випливає, що для послідовного переобчислення значень змінної x_i на j-му кроці беруться значення $x_{i-m}, x_{i-m+1}, \dots, x_{i-1}$, які є вже переобчисленими на цьому ж кроці. Схематично інформаційні зв'язки між ітераціями циклу (2) для $m = 1$, $n = 5$, $k = 4$ зображено на рис. 1.

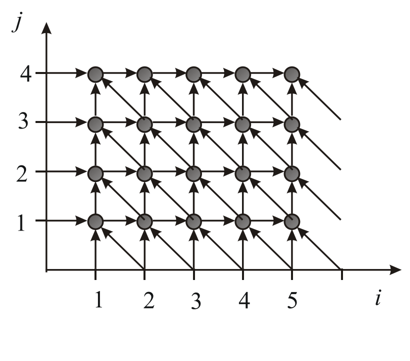

Рис. 1

Для реалізації послідовного алгоритму (2) необхідно виконати

$$2(2m+1)nk$$

операцій.

Паралельні алгоритми розв'язання ЗЦФ.

Природній перехід до *паралельної схеми* обчислень пов'язаний з одночасним переобчисленням значень всіх змінних $x_i \ (i=\overline{1,n})$, який можна здійснити з допомогою алгоритму:

$$\begin{array}{l} FOR \;\; j=1, k_1 \;\; DO \\ \{ \;\; FOR \; i=1, n \;\; DO \;\; PAR \\ \;\; \{ \;\; p_i = 0 \\ \;\;\; FOR \; s=-m, m \;\; DO \\ \;\;\; \{ \;\; p_i = p_i + x_{i+s} * f_s \;\; \} \\ \;\;\;\;\; x_i = p_i \;\; \} \; \}. \end{array} \tag{3}$$

Тут *PAR* – це тип паралельності.

Даний алгоритм задає паралельне переобчислення значень всіх змінних. Однак тут можливі *два варіанти* виконання обчислень. За ***першого варіанту***, коли *PAR* є *SIM* (simultaneous), для *j*-го переобчислення беруться значення змінних, одержані виключно під час (*j*-1)-го переобчислення. Для забезпечення цього необхідна синхронізація паралельних гілок після виконання чергового переобчислення. Тому таку схему обчислень будемо називати ***синхронною***, а відповідний алгоритм, який її реалізує, є зорієнтований на виконання на *SIMD*-обчислювальних системах. Інформаційні зв'язки між ітераціями (i, j) конструкції (3) для $m=1, \ n=5, \ k_1=5$ зображено на рис. 2.

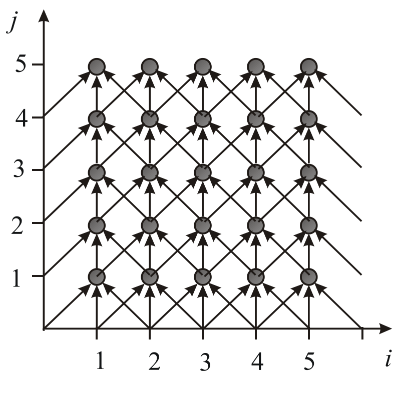

Рис. 2

Для реалізації паралельного алгоритму (3) (синхронна схема) необхідно виконати

$$2(2m+1)k_1$$

операцій.

За ***другого варіанту***, коли *PAR* є *CONC* (concurrent), переобчислення значень змінної x_i здійснюється незалежно від переобчислення значень інших змінних і при цьому як аргументи використовуються поточні значення $x_{i-m}, x_{i-m+1}, \dots, x_i, \dots, x_{i+m}$. Один із можливих порядків обчислень у даному разі для $m=1,\ n=5,\ k_1=4$ зображено на рис. 3. Таку схему обчислень називають ***асинхронною***.

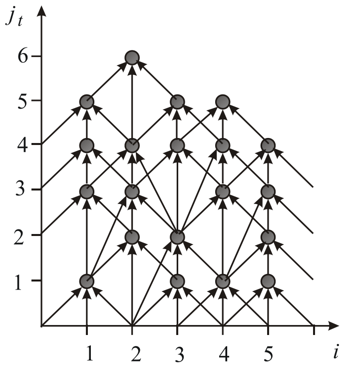

Рис. 3

Зауважимо, що графи інформаційної залежності для (2) і (3) є *різними*.

Прискорення паралельного алгоритму.

Еквівалентний послідовний аналог алгоритму (3) (синхронна схема) можна записати у вигляді:

$$
\begin{array}{l}
FOR\ \ j=1, k_1\ \ DO \\
\{\ FOR\ i=1, n\ \ DO \\
\quad \{\ p=0 \\
\qquad FOR\ s=-m, m\ \ DO \\
\qquad \{\ p=p+x_{i+s}^{j-1} * f_s\ \} \\
\qquad\quad x_i^j = p\ \}\ \}.
\end{array}
\qquad (4)
$$

Тут вважаємо, що $x_{i'}^0 = x_{i'}$ для всіх $i'=\overline{1-m,\ n+m}$. Тобто це є початкові значення змінних (до виконання першого переобчислення). Результатом роботи послідовного алгоритму (4) є $x_i^{k_1},\ i=\overline{1,\ n}$.

Графи інформаційної залежності алгоритму (3) (синхронна схема) та (4) співпадають. Це означає, що за однакових вхідних даних ці алгоритми дають однаковий результат.

Для реалізації послідовного алгоритму (4) необхідно виконати

$$2(2m+1)nk_1$$

операцій.

Оскільки графи інформаційної залежності між ітераціями для (3) (синхронна схема) та (4) співпадають, то ці алгоритми можна порівнювати за швидкодією. Тому прискорення алгоритму (3) (синхронна схема) порівняно з його послідовною реалізацією обчислюємо за формулою:

$$S = \frac{2nk_1(2m+1)}{2k_1(2m+1)} = n\,.$$

Ця оцінка приведена за умови, що синхронізація суттєво не впливає на час виконуваних обчислень. На практиці цього можна досягти завдяки апаратній реалізації процесу синхронізації або програмній унаслідок ефективного використання кеш-пам'яті.

Вправи для самостійної роботи.

1. Схематично зобразити інформаційні зв'язки між ітераціями для алгоритмів (2) у випадку $m = 2,\ n = 6,\ k = 3$ та (3) (синхронна схема) у випадку $m = 2,\ n = 7,\ k_1 = 4$.
2. Зобразити два можливі порядки обчислень під час розв'язання ЗЦФ за асинхронною схемою для $m = 1,\ n = 5,\ k_1 = 4$, відмінні від наведеного на рис. 3.