

## ДОДАТОК А

### Лістинг коду програми сортування на GPU

```
#include<thrust/device_vector.h>
#include<thrust/tuple.h>
#include<thrust/reduce.h>
#include<thrust/fill.h>
#include<thrust/generate.h>
#include<thrust/sort.h>
#include<thrust/sequence.h>
#include<thrust/copy.h>
#include<cstdlib>
#include<ctime>
#include<time.h>

usingnamespace thrust;

// return the biggest of two tuples
template<class T>
struct bigger_tuple {
    __device__ __host__
    tuple<T,int>operator()(const tuple<T,int>&a, const tuple<T,int>&b)
    {
        if (a > b) return a;
        elsereturn b;
    }
};

template<class T>
int max_index(device_vector<T>& vec) {

// create implicit index sequence [0, 1, 2, ... )
```

```

    counting_iterator<int>begin(0); counting_iterator<int> end(vec.size());
tuple<T,int> init(vec[0],0);
tuple<T,int> smallest;

smallest = reduce(make_zip_iterator(make_tuple(vec.begin(), begin)),
make_zip_iterator(make_tuple(vec.end(), end)),
init, bigger_tuple<T>());
return get<1>(smallest);
}
// return the biggest of two tuples
template<class T>
struct smaller_tuple {
    __device__ __host__
tuple<T,int>operator()(const tuple<T,int>&a, const tuple<T,int>&b)
    {
if (a < b) return a;
elsereturn b;
    }
};

template<class T>
int min_index(device_vector<T>& vec) {

// create implicit index sequence [0, 1, 2, ... )
    counting_iterator<int>begin(0); counting_iterator<int> end(vec.size());
tuple<T,int> init(vec[0],0);
tuple<T,int> smallest;

smallest = reduce(make_zip_iterator(make_tuple(vec.begin(), begin)),
make_zip_iterator(make_tuple(vec.end(), end)),

```

```

init, smaller_tuple<T>());
return get<1>(smallest);
}

int main(){

    thrust::host_vector<int> h_vec(1000000);
    thrust::sequence(h_vec.begin(), h_vec.end()); // values = indices

// transfer data to the device
    thrust::device_vector<int> d_vec = h_vec;
        clock_t begin = clock();
int index = max_index(d_vec);
        int min_i = min_index(d_vec);
        clock_t end = clock();

    std::cout <<"MAX Value is: "<< h_vec[index] <<std::endl;
    std::cout <<"MIN Value is: "<< h_vec[min_i] <<std::endl;
        std::cout <<"Time: "<<double(end - begin) / CLOCKS_PER_SEC <<
std::endl;
        std::system("Pause");
return 0;
}

```

## ДОДАТОК Б

### Лістинг коду програми сортування на CPU

```
#include<iostream>
#include<ctime>

usingnamespace std;
int main()
{
setlocale(0,"Rus");
srand((unsigned) time(NULL));
int* ptr;
int n;
unsignedlongint i;
int min, max;
int temp;
    time_t start, stop;

cout<<"Введитеразмермассива ";
clock_t begin = clock();
    cin >> n;
    ptr = newint[n]; // динамически выделяем память
for ( i = 0; i < n; i++)
ptr[i] = rand() % 100; // заполняемслучайнымичислами

/*for ( i = 0; i < n; i++)
cout<< ptr[i] << " ";
cout<< endl; */

min = ptr[0];
```

```
for ( i = 1; i < n; i++)
if(min > ptr[i])
    {
temp = ptr[i];
ptr[i] = min;
min = temp;
    }
```

```
cout <<"Минимальный элемент " << min << endl;
```

```
max = ptr[0];
for ( i = 1; i < n; i++)
    if( max < ptr[i])
        {
            temp = ptr[i];
            ptr[i] = max;
            max = temp;
        }
clock_t end = clock();
cout <<"Максимальный элемент " << max << endl;
cout <<"Час выполнения " << double(end - begin) / CLOCKS_PER_SEC << endl;
    system("Pause");
}
```

## АПАРАТНА РЕАЛІЗАЦІЯ ОБЧИСЛЕННЯ МАКСИМАЛЬНОГО І МІНІМАЛЬНОГО ЧИСЕЛ В МАСИВІ ДАНИХ

Цмоць І.Г.<sup>1)</sup>, Ігнатів І.В.<sup>2)</sup>, Данілов П.О.<sup>3)</sup>

Тернопільський національний економічний університет

<sup>1)д.т.н., професор;</sup> <sup>2) викладач;</sup> <sup>3) магістрант</sup>

### I. Постановка задачі

Навчання мережі на «сирому» наборі, як правило, не дає високих показників продуктивності. Існує ряд способів покращити «сприйняття» мережі, а саме: нормалізація, квантування та фільтрація. Існує велика кількість способів нормалізації вхідних даних. У більшості випадків, якщо вхідні дані є більш-менш рівномірними, для апаратної реалізації найкраще використовувати лінійну нормалізацію. Для апаратної реалізації такої нормалізації потрібно розробити методи та структури для обчислення таких базових операцій:

- визначення максимального числа з групи чисел;
- ділення.

Після нормалізації даних, у залежності від типу мережі, можуть використовуватися інші процедури попередньої обробки даних. Тому актуальною проблемою є розробка алгоритмів визначення максимального та мінімального чисел з масиву.

### II. Мета роботи

Метою роботи є розробка алгоритмів та НВІС-структури пристроїв визначення максимального та мінімального значень з масиву даних.

### III. Виклад основного матеріалу

Аналіз методів і алгоритмів обчислення максимальних і мінімальних значень із масиву чисел показав, що для НВІС-реалізацій найефективнішими є алгоритмами, які ґрунтуються на методі порозрядного порівняння [4]. Обчислення максимального  $A_{max}$  і мінімального  $A_{min}$  чисел із групи чисел  $A_1, A_2, \dots, A_j, \dots, A_m$  за таким методом виконується послідовним порівнянням розрядів всіх чисел починаючи зі старшого. При кожному порівнянні отримуємо  $i$ -і розряди максимального і мінімального чисел, обчислення яких здійснюється за формулами:

$$\overline{A_{max}} = \bigwedge_{j=1}^m \overline{a_j} \wedge y_{ij}, y_{ij} = 1$$

$$\overline{A_{min}} = \bigwedge_{j=1}^m \overline{a_j} \wedge z_{ij}, z_{ij} = 1$$

де  $y_{ij}, z_{ij}$  -  $i$ -і розряди  $j$ -х слів управління;  $a_j$  -  $i$ -й розряд  $j$ -числа;  $m$  - кількість чисел у групі.

Формування  $(i+1)$ -х розрядів  $j$ -х слів управління виконується за формулами:

$$y_{(i+1)j} = (\overline{A_{max}} \vee x_j) \wedge y_{ij}$$

$$z_{(i+1)j} = (\overline{A_{min}} \vee x_j) \wedge z_{ij}$$

Процес синтезу паралельних НВІС-структур для обчислення максимальних і мінімальних чисел із групи чисел зводиться до виконання наступних етапів:

- виділення базової операції;
- просторово-часове відображення алгоритму;
- розробка схеми процесорного елемента (ПЕ), що реалізує базову операцію алгоритму;
- синтез НВІС-структур на базі ПЕ;
- організація інтерфейсу НВІС.

Виділена базова операція реалізується у вигляді процесорного елемента, схема процесорного елемента одноканального, конвексного пристрою та пристрою з вертикальним опрацюванням вхідних чисел зображена на рисунку 1.

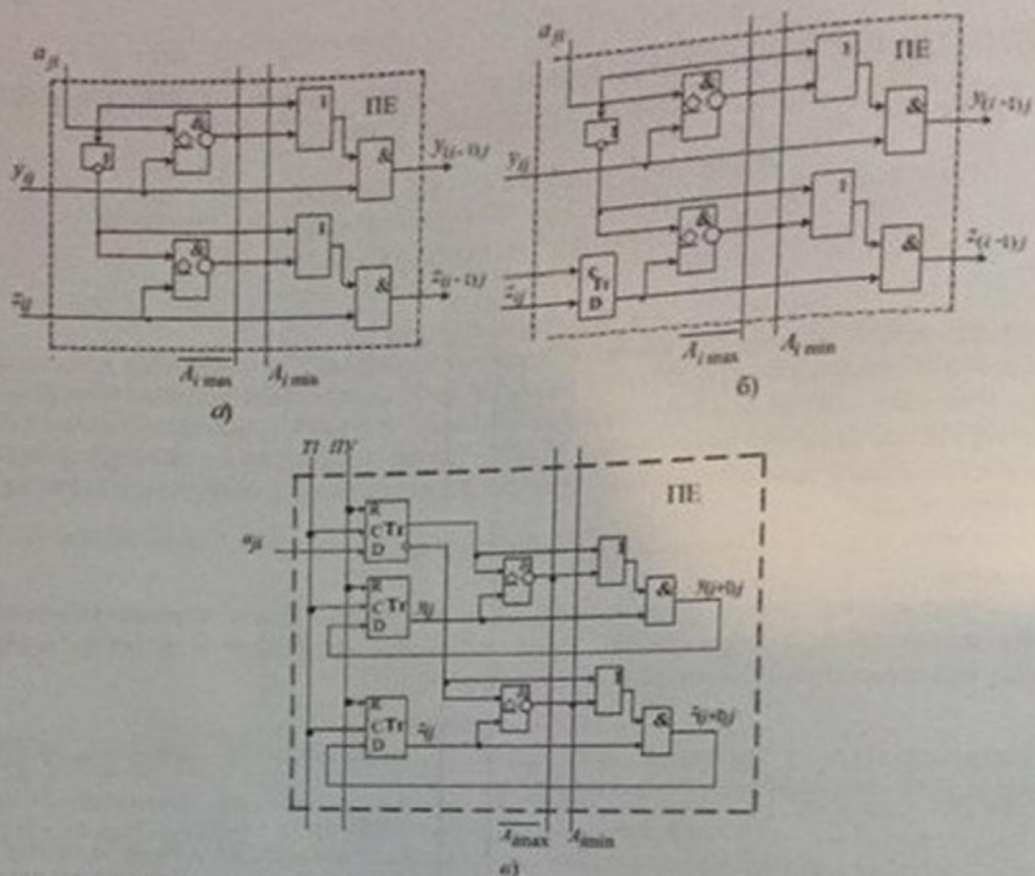


Рисунок 1 - Схеми процесорних елементів: а-однотактного пристрою; б – конвейсного пристрою; в – пристрою з вертикальним опрацюванням вхідних даних.

Вартість НВІС-пристроїв обчислення максимальних і мінімальних значень із групи чисел в основному залежить від розміру кристала, яка визначається як витрати на обладнання, кількість зовнішніх виводів, число яких технологічно обмежене. Орієнтація структур сортування на НВІС-реалізацію вимагає зменшення числа виводів інтерфейсу та кількості з'єднань між процесорними елементами. Забезпечити ці вимоги можна використанням паралельно-вертикального алгоритму обчислення максимальних і мінімальних значень із групи чисел, при якому надходження чисел і видача результатів здійснюється розрядами.

#### Висновок

Аналіз методів і алгоритмів обчислення максимальних і мінімальних значень із масиву чисел показав, що для реалізації обчислення мінімальних і максимальних чисел у масиві даних найкраще реалізувати способами, які ґрунтуються на методах порозрядного порівняння.

Час за який визначається мінімальні та максимальні числа з масиву даних за методом порозрядного порівняння визначається їх розрядністю а не кількістю.

Зменшено час обчислення максимального та мінімального числа з масиву даних за рахунок використання спільної магістралі.

#### Список використаних джерел

1. Нейроподібні методи, алгоритми та структури обробки сигналів і зображень у реальному часі: монографія / Ю.М. Рашкевич, Р.О. Ткаченко, І.Г. Цмоць, Д.Д. Пелешко. – Львів: Видавництво Львівської політехніки, 2014. – 256 с.
2. Проблемно-орієнтованые высокопроизводительные вычислительные системы: В.Ф. Гузик, В.Е. Золотовский. Учебное пособие. Тагапрог:Изд-во ТРТУ, 1998. 236 с.
3. Уоссермен Ф. Нейрокомпьютерная техника. – М.: Мир, 1992. – 259с.

## ДОДАТОК Г

ДОДАТОК —

Довідка про використання результатів дипломної роботи



Науково-дослідний інститут  
інтелектуальних комп'ютерних систем

Research Institute of  
Intelligent Computer Systems

Ternopil National Economic University  
Ministry of Education and Science of Ukraine  
Glushkov Institute of Cybernetics  
National Academy of Sciences of Ukraine

3 Peremoga Square, Ternopil, 46009, Ukraine  
Tel: +380 (352) 475050 ext.12234  
Fax: +380 (352) 47-5053 (24 hrs)  
[www.ics.tneu.edu.ua](http://www.ics.tneu.edu.ua) [ics@tneu.edu.ua](mailto:ics@tneu.edu.ua)

Тернопільський національний економічний  
університет Міністерство освіти і науки України  
Інститут кібернетики ім. В.М. Глушкова  
Національна академія наук України

пл. Перемоги 3, Тернопіль, 46009, Україна  
Тел: +380 (352) 475050 внутр.12234  
Факс: +380 (352) 47-5053 (24 год)  
[www.ics.tneu.edu.ua](http://www.ics.tneu.edu.ua) [ics@tneu.edu.ua](mailto:ics@tneu.edu.ua)

Завідувачу кафедри  
комп'ютерної інженерії  
д.т.н., проф. О.М. Березькому

### ДОВІДКА ПРО ВИКОРИСТАННЯ

Виконаний студентом групи КСМм-21 факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету Данілова П.О. дипломна робота на тему: «Апаратна реалізація обчислення максимальних і мінімальних значень чисел в масиві даних» відповідає профілю роботи науково-дослідного інституту, має практичну значимість і планується для використання.

Директор науково-дослідний інститут  
інтелектуальних комп'ютерних систем  
(підпис)

М.П.



Кочан В.В.