

**Автоматичне генерування та
еволюція алгоритмів на клітинному
автоматі з елементами стохастики**

Виконав: Міщенко Михайло Дмитрович

Науковий керівник: ас. Кухарєв Сергій Олександрович

Мета роботи:	дослідження виникнення алгоритмів у напівдетерміністичних системах
Об'єкт дослідження:	штучне життя
Предмет дослідження:	клітинний автомат "гра Життя" з випадковими збуреннями
Метод дослідження:	комп'ютерне моделювання

- Дослідження виникнення та еволюції алгоритмів у напівдетерміністичних системах, здатних до обчислень.
- Дослідження алгоритмів як штучного життя.
- Дослідження штучного життя в умовах, коли критерій відбору не задано дослідником безпосередньо, але він витікає з характеристик середовища, в якому воно існує.

Постановка задачі

4

Побудувати модель віртуального світу для дослідження алгоритмічного штучного життя, яка:

1. Придатна для моделювання на сучасних ЕОМ.
2. Містить абстракції простору та часу
3. Повна за Тюрінгом.
4. Будь-які можливі в рамках моделі об'єкти здатні еволюціонувати.
5. Має як детерміністичні, так і стохастичні характеристики

У якості такої моделі обрано клітинний автомат “гра Життя” зі стохастичною модифікацією

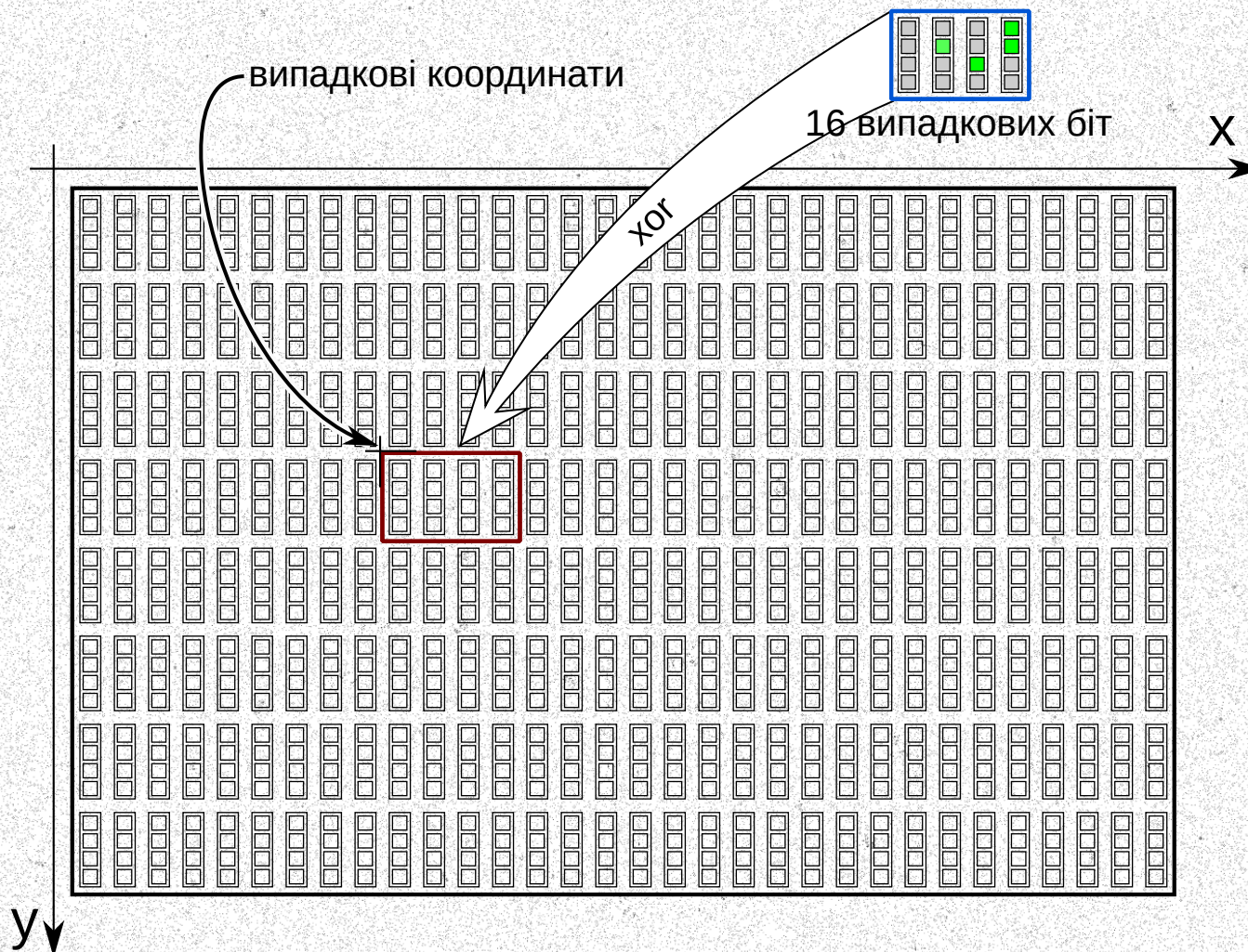
Програмний продукт повинен:

- Забезпечити високу швидкодію при значному різноманітті вмісту поля
- Забезпечити можливість спостереження за ходом експерименту
- Надати інструментарій для дослідження “цікавих” досліднику об’єктів

Стохастична модифікація

6

Кожні n кроків до випадкової області 4×4 застосовується операція xor з випадковими бітами

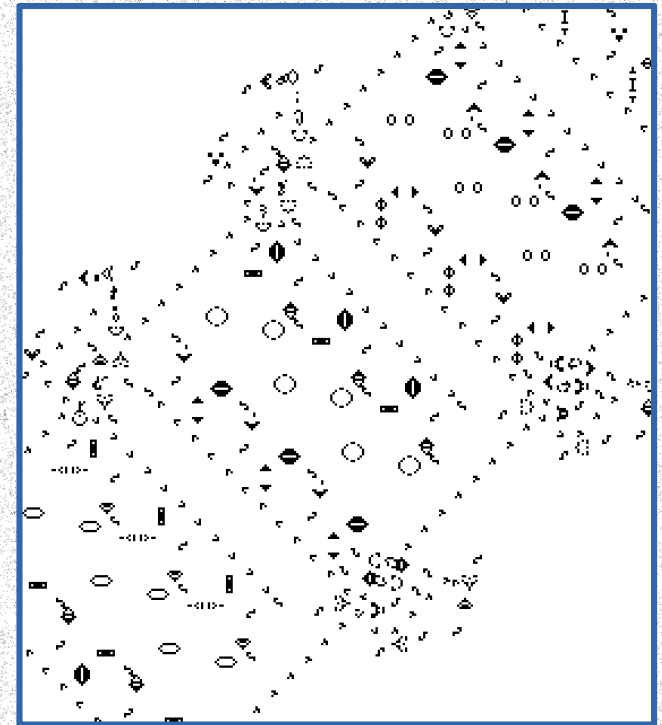


Висока швидкодія при значному різноманітті вмісту поля

7

Основні міркування

- Відомі алгоритми швидкісного обчислення клітинних автоматів використовують високу регулярність їх вмісту.
- На полях з високим різноманіттям вмісту вони дають погані результати.
- Обчислення на GPU оптимальні для виконання великої кількості однотипних обчислень.

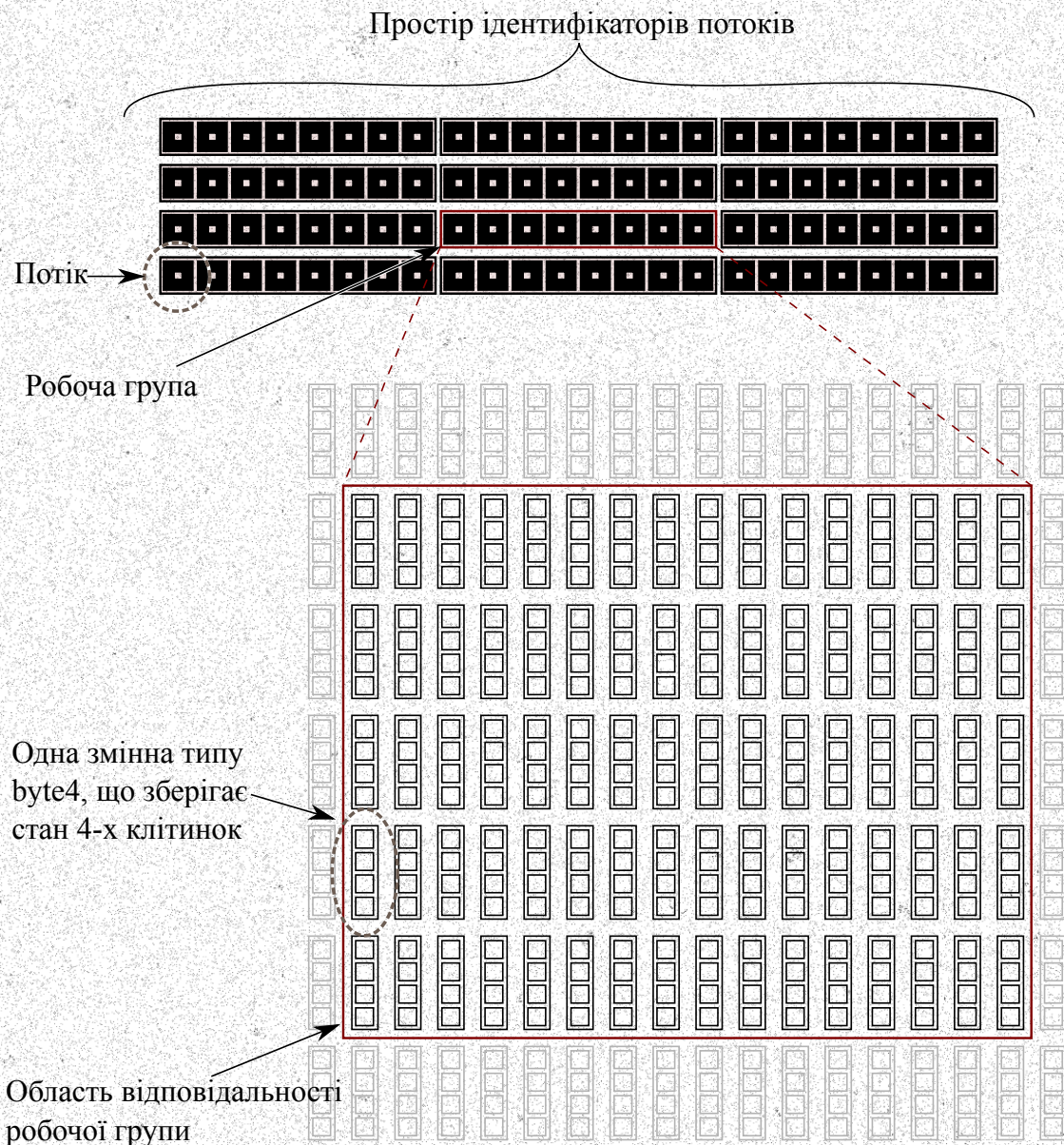


Приклад регулярного вмісту

Машина Тьюрінга
(фрагмент, авт. Paul Rendell)

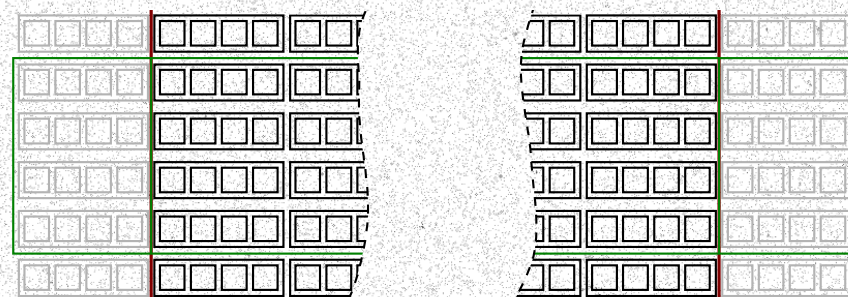
Висока швидкодія при значному різноманітті вмісту поля

8



Співвідношення простору потоків із полем

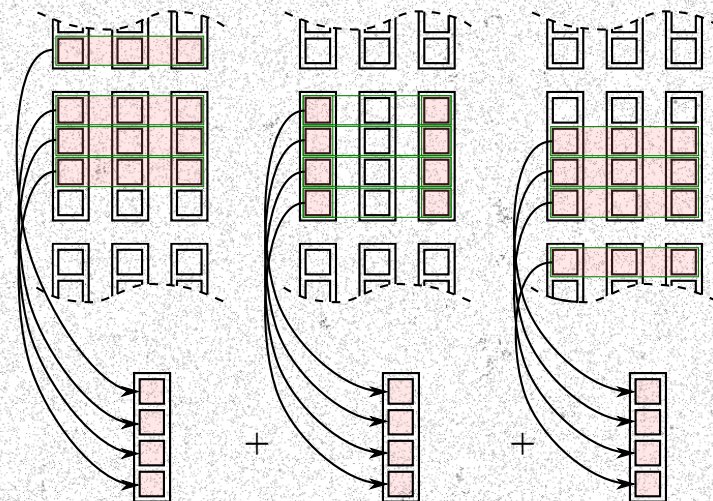
- Потоки об'єднуються в робочі групи.
- Синхронізація роботи потоків можлива лише в рамках однієї робочої групи.
- Робочі групи можуть виконуватись як всі одночасно, так і пакетами по n робочих груп.
- За загальним правилом, заборонено запис у змінні розміром не кратним 4-м.
- На архітектурі CUDA 4-байтні транзакції сусідніх потоків до сусідніх комірок пам'яті об'єднуються в одну.



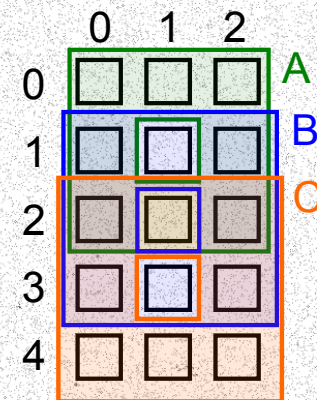
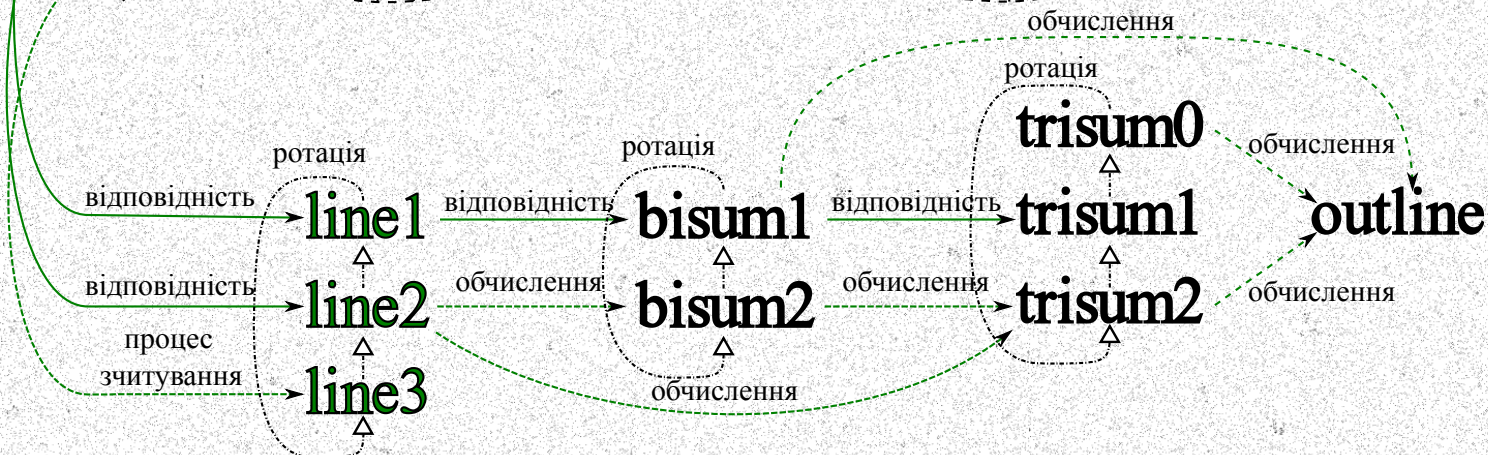
Варіант із горизонтальним розташуванням клітинок вектора

Висока швидкодія при значному різноманітті вмісту поля

9



Обчислення околів за допомогою векторів



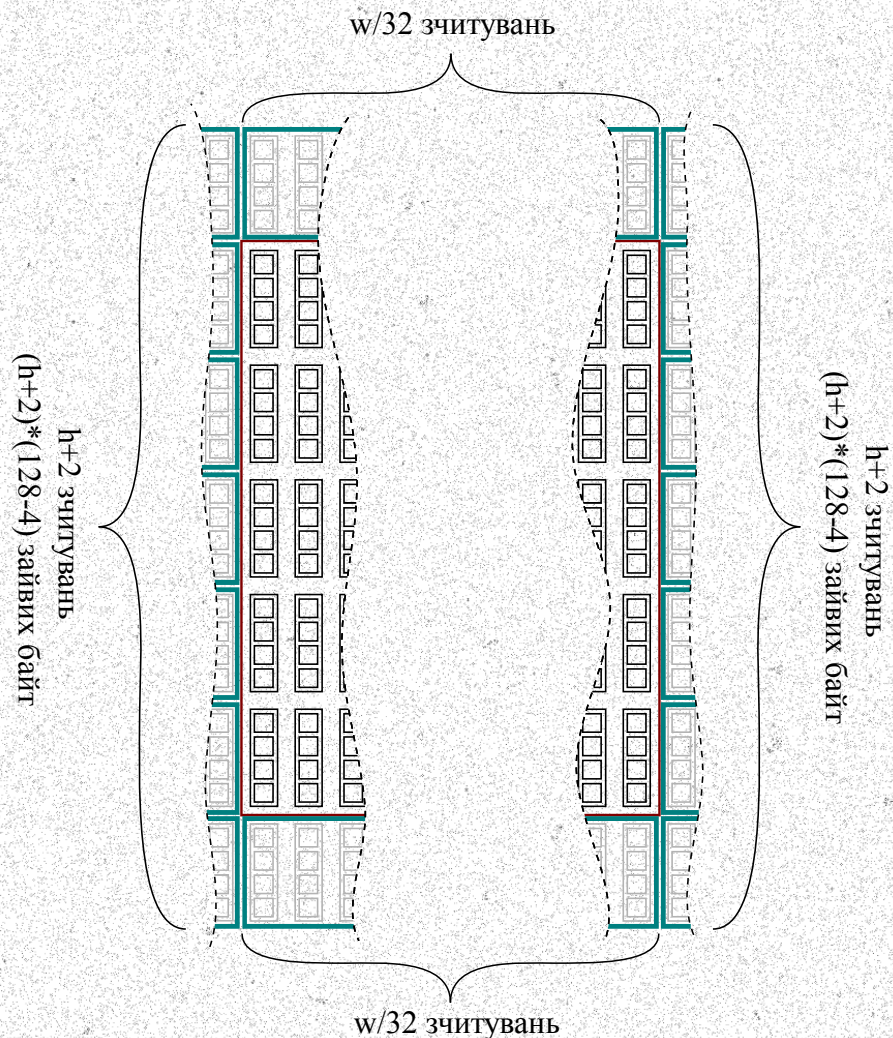
Ротация буферів локальної пам'яті GPU

Перетини околів

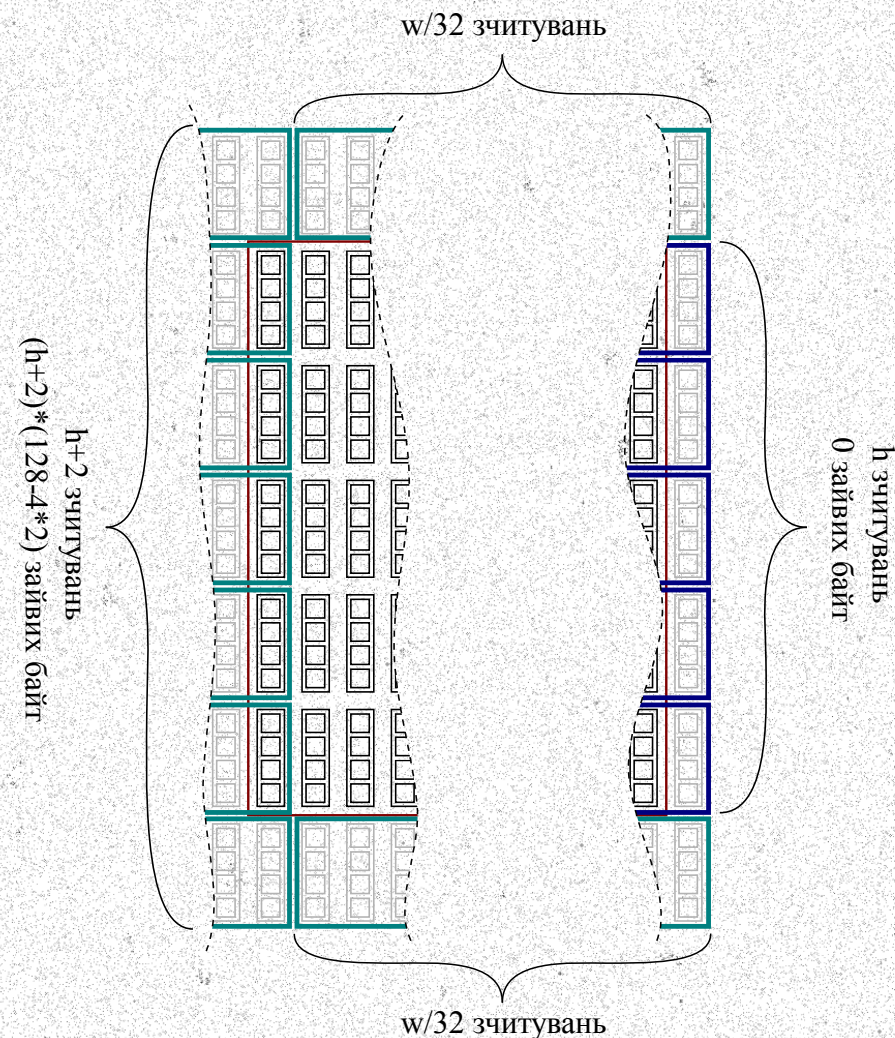
Висока швидкодія при значному різноманітті вмісту поля

10

Застосування зміщення



Без зміщення в буфері



Зі зміщенням на 4 байти

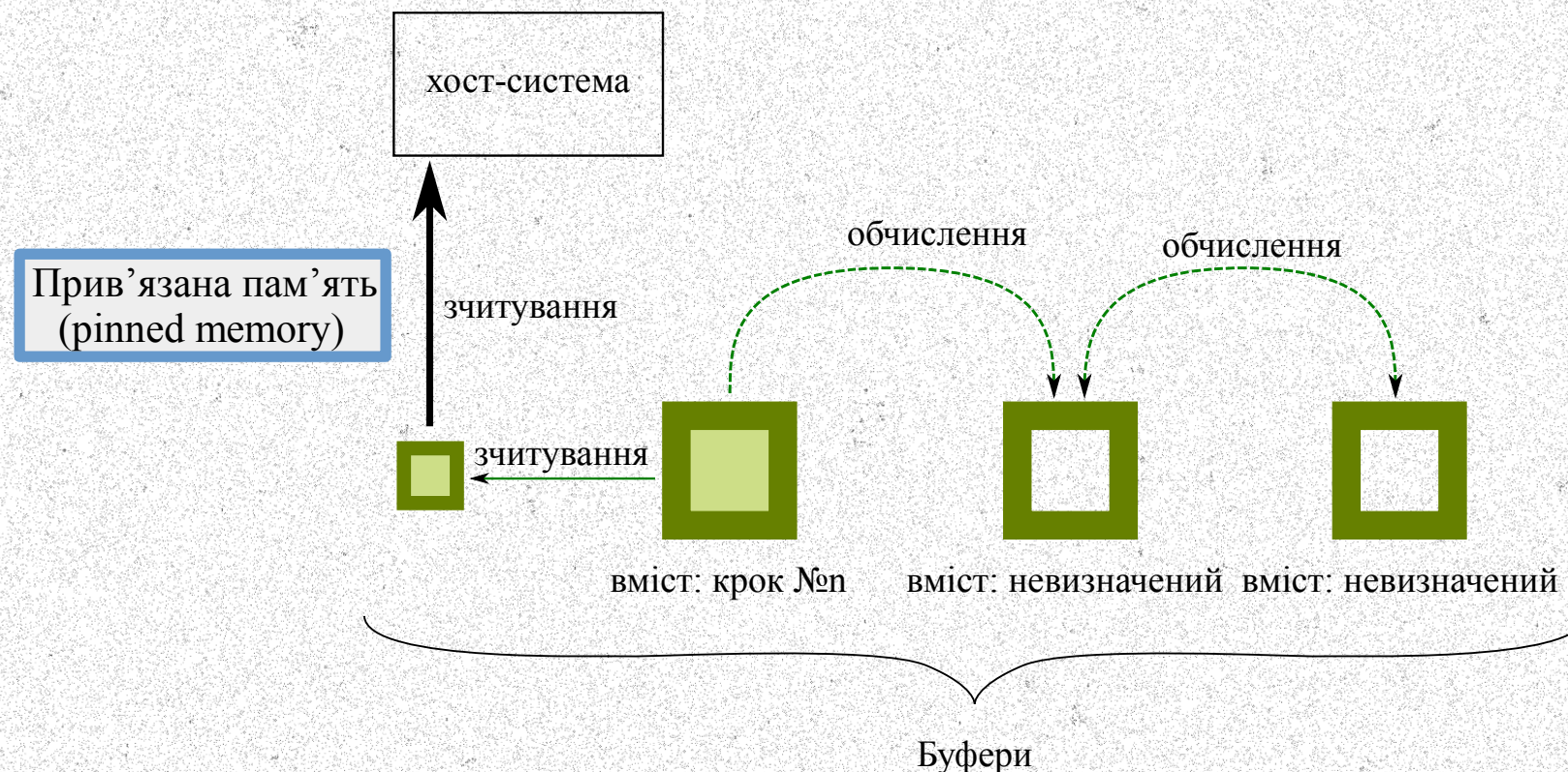


Схема використання буферів

Режими керування обчисленнями:

- “по натисканню кнопки”
- режим “трансляції”
- режим “перегляд офлайн”.

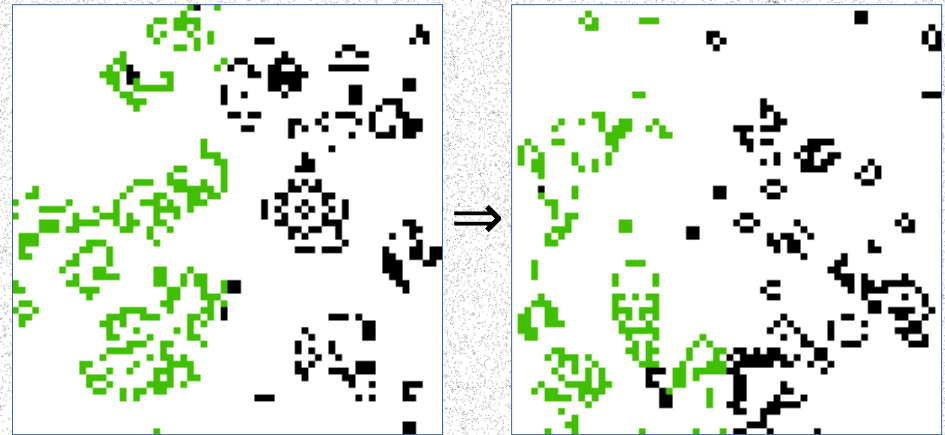
Інструментарій для дослідження “цікавих” досліднику об’єктів

12

Маркування об’єктів

У кожній клітинці:

- 7 комбінуємих флагів
- 1 біт “чистоти”



Оператор комбінування міток ($\oplus$)

Властивості:

Ідемпотентність	$a \oplus a = a$
Комутативність	$a \oplus b = b \oplus a$
Асоціативність	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$
Нейтральний елемент	$a \oplus 0 = 0 \oplus a = a$
Мітки	$(a \oplus b) \& 0xFE = (a \mid b) \& 0xFE$
Біт чистоти	$a \neq b, a \neq 0, b \neq 0 \Rightarrow (a \oplus b) \& 0x01 = 0$

Інструментарій для дослідження “цікавих” досліднику об’єктів

13

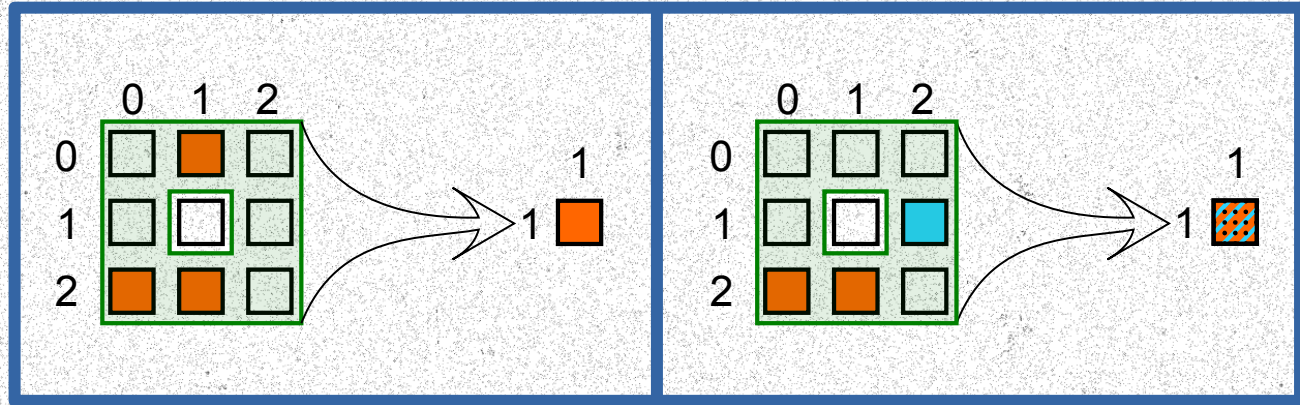
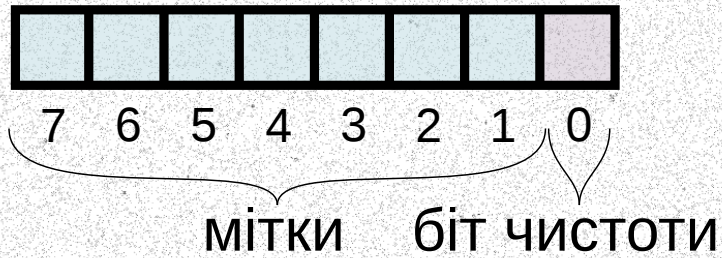


Diagram illustrating the XOR operation for the 8th bit. Two 8-bit registers are shown. The first register has bits 7-1 as 0 and bit 0 as 1. The second register has bits 7-1 as 0 and bit 0 as 1. The result of the XOR operation is shown in a third register, where bits 7-1 are 0 and bit 0 is 1. The operation is labeled with a circled plus sign (XOR symbol).

Diagram illustrating the XOR operation for the 8th bit. The first row shows the input vectors: [0, 0, 0, 0, 0, 1, 0, 1] and [0, 0, 0, 0, 0, 0, 0, 1]. The second row shows the result vector: [0, 0, 0, 0, 0, 1, 0, 0]. The XOR operation is indicated by the symbol $\oplus$.

Diagram illustrating the XOR operation between two 8-bit numbers:

0	0	0	0	0	1	0	1
7	6	5	4	3	2	1	0

 $\oplus$

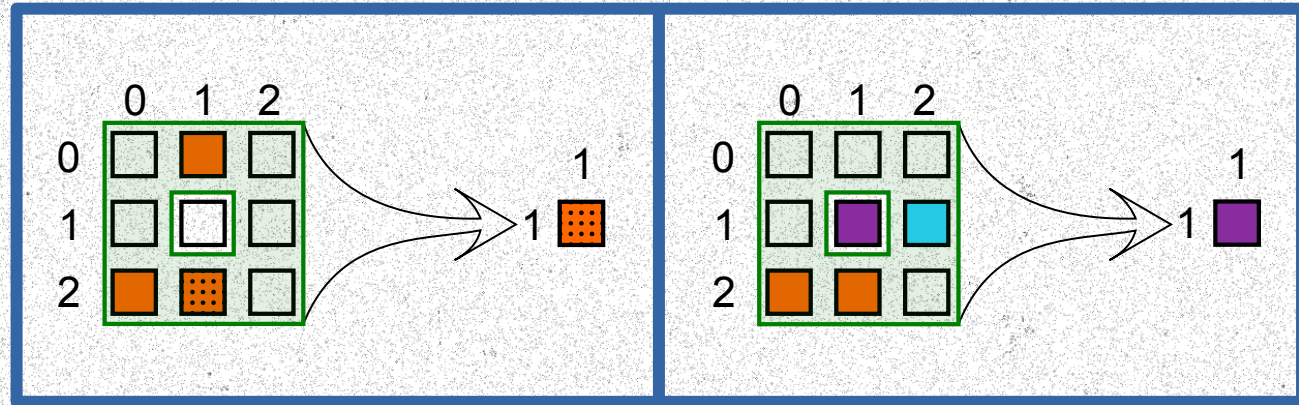
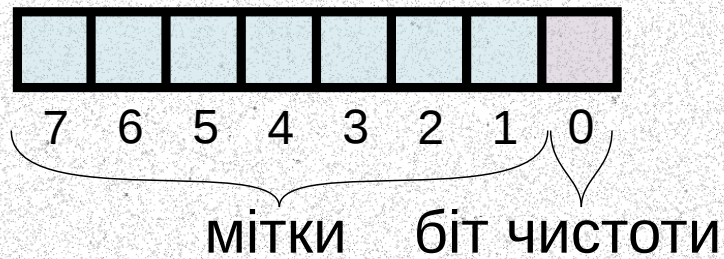
0	0	0	0	0	1	0	1
7	6	5	4	3	2	1	0

 $=$

0	0	0	0	0	1	0	1
7	6	5	4	3	2	1	0

Інструментарій для дослідження “цікавих” досліднику об’єктів

14



$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 \oplus
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 =
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 \oplus
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 =
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 \oplus
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 =
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 \oplus
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}
 =
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline \end{array}
 \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}$$

Інструментарій для дослідження “цікавих” досліднику об’єктів

15

Реалізація оператора:

- На C

```
#define COMPOSE_FLAGS(a,b) ((( (b) & ~!!(a)) | \
                                ((a) & ~!!(b)) \
                                ) & ~((a) == (b)) \
                                ) | \
                                (((a) == (b)) & (a) & (b)))
```

	C	OpenCL C (для векторів)
true	1	-1
false	0	0

Представлення “істини”
в C та OpenCL C

- На OpenCL C для векторів

```
#define NZ(n) ((n) != (char4)(0))
#define ONE ((char4) (1))
#define COMPOSE_FLAGS(a,b) ((( (b) & ~(NZ(a) & ONE)) | \
                                ((a) & ~(NZ(b) & ONE)) \
                                ) & ~(((a) == (b)) & ONE) \
                                ) | \
                                (((a) == (b)) & (a) & (b) & ONE))
```

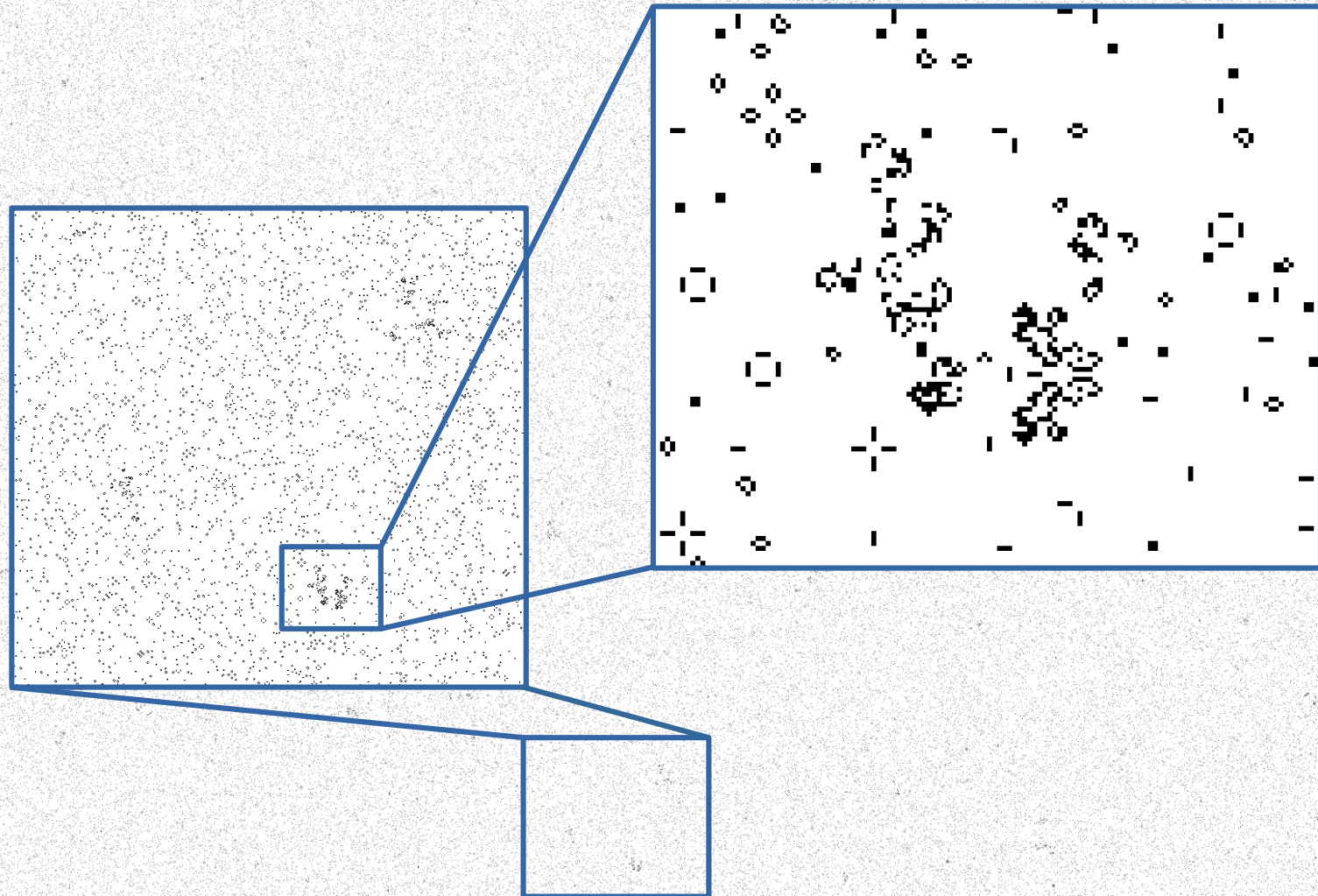

Приклад роботи програми

16

Результат
моделювання
впродовж 6 год.
реального часу

на полі
8192x6144

з випадковими
збуреннями
перед кожною
ітерацією.



- Створено експериментальну модель напівдетерміністичного середовища для дослідження виникнення і еволюції алгоритмів у ньому.
- Розроблено методологію додавання позначок до об'єктів клітинних автоматів і передачі їх на його наступні ітерації.
- Розроблено інструмент для дослідження еволюційних процесів в стохастичних клітинних автоматах, працюючий з високою швидкістю.

- Було розроблено алгоритм обчислення наступних станів клітинного автомату “гра Життя”, оптимізований під виконання на відеокарті.
- Було розроблено бінарний оператор комбінування міток клітинок клітинного автомату в певному околі.
- Було створено реалізацію цього алгоритму з застосуванням розробленого бінарного оператора у вигляді програмного продукту.

Дякую за увагу!

