Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №1**

по дисциплине

«Модели решения задач в интеллектуальных системах»

Вариант 13

Выполнили:

студенты гр. 121702 Витковская С. И.

Проверил: Ивашенко В. П.

Минск, 2024

**Тема**: Реализация модели решения задачи на конвейерной архитектуре.

**Цель**: Реализовать и исследовать модель решения на конвейерной архитектуре задачи вычисления попарного произведения компонентов двух векторов чисел.

**Задание:** Реализовать алгоритм вычисления произведения пары 8-разрядных чисел умножением с младших разрядов со сдвигом множимого влево.

**Дополнительные теоретические сведения:**

*Конвейер –* архитектура, состоящая из нескольких параллельно решающих обобщенную задачу этапов, последовательно обрабатывающих сущности одного потока.

*Сбалансированный конвейер –* конвейер, все этапы в котором имеют одинаковое время работы.

*Коэффициент ускорения:* , где:

–время решения задачи на первой(последовательной) архитектуре;

– время решения задачи на другой (параллельной архитектуре);

– количество процессорных элементов в системе (совпадает с количеством этапов конвейера);

– ранг задачи (количество объектов, которые в процессе решения задачи могли бы обрабатываться параллельно).

*Эффективность вычислительных систем:*  , где:

– количество процессорных элементов в системе (совпадает с количеством этапов конвейера);

– ранг задачи (количество объектов, которые в процессе решения задачи могли бы обрабатываться параллельно).

**Алгоритм вычисления произведения:**

1. Исходное значение частичной суммы принимается равным нулю.
2. Анализируется очередной бит множителя (анализ начинается с младших разрядов).
3. Получаем частичное произведение умножением цифры текущего бита множителя на множимое.
4. Сдвигаем полученное произведение на количество битов, равное номеру бита, рассматриваемого на данной итерации (отсчет ведется с младших битов с 0).
5. Прибавляем к частичной сумме частичное произведение.
6. Пункты 3-5 повторяются для всех разрядов множителя.

**Пример:**

Произведение пары чисел 6 и 120. , .

| Этап | Смещение на | Цифра бита | Итог этапа | Пояснение |
| --- | --- | --- | --- | --- |
|  |  |  | 0000000000000000 | Изначальная нулевая сумма |
| **1** | 0 | 0 | 0000000000000000  0000000000000000 | Частичное произведение  Частичная сумма |
| **2** | 1 | 0 | 0000000000000000  0000000000000000 | Частичное произведение  Частичная сумма |
| **3** | 2 | 0 | 0000000000000000  0000000000000000 | Частичное произведение  Частичная сумма |
| **4** | 3 | 1 | 0000000000110000  0000000000110000 | Частичное произведение  Частичная сумма |
| **5** | 4 | 1 | 0000000001100000  0000000010010000 | Частичное произведение  Частичная сумма |
| **6** | 5 | 1 | 0000000011000000  0000000101010000 | Частичное произведение  Частичная сумма |
| **7** | 6 | 1 | 0000000110000000  0000001011010000 | Частичное произведение  Частичная сумма |
| **8** | 7 | 0 | 0000000000000000  0000001011010000 | Частичное произведение  Частичная сумма |

Ответ:

**Описание программы:**

Программа включает в себя классы Utils, Pair и Conveyor.

Utils отвечает за вспомогательные функции и включает методы convert\_direct и check.

1. convert\_direct переводит число из десятичной системы в двоичную;
2. check проверяет, является ли введенный пользователем элемент той или иной пары целым неотрицательным числом в промежутке от 0 до 255 включительно.

Класс Pair описывает пару чисел. Содержит поля, хранящие номер пары, сами числа в двоичной системе, текущую частичную сумму и последнее частичное произведение, индекс, с которого начинается очередная итерация и результат перемножения. Методы:

1. sum производит сложение двух бинарных чисел;
2. partial\_multiplication производит частичное произведение с учетом индекса, передаваемого в качестве аргумента;
3. part\_sum выполняет частичной суммирование: вызывает функцию partial\_multiplication, затем sum для сложения нового частичного произведения и последней частичной суммы.

Класс Conveyor хранит список пар, представленных объектами класса Pair, длину очереди и длину очереди.

1. Метод entry выполняет проверки введенных пользователем чисел, генерирует пары, присваивает им номера и добавляет их в соответствующий список;
2. Методы write\_pairs, write\_tacts и write\_results отвечают за вывод;
3. Метод algorithm создает основной цикл, регулирующий количество и порядок пар на конвейере, вызов методов подсчета частичной суммы и вывода результатов.

**Графики:**

График зависимости коэффициента ускорения от количества объектов, которые в процессе решения задачи могли бы обрабатываться параллельно (количества пар) r при фиксированном количестве процессорных элементов, соответственно варианту, равному 8.

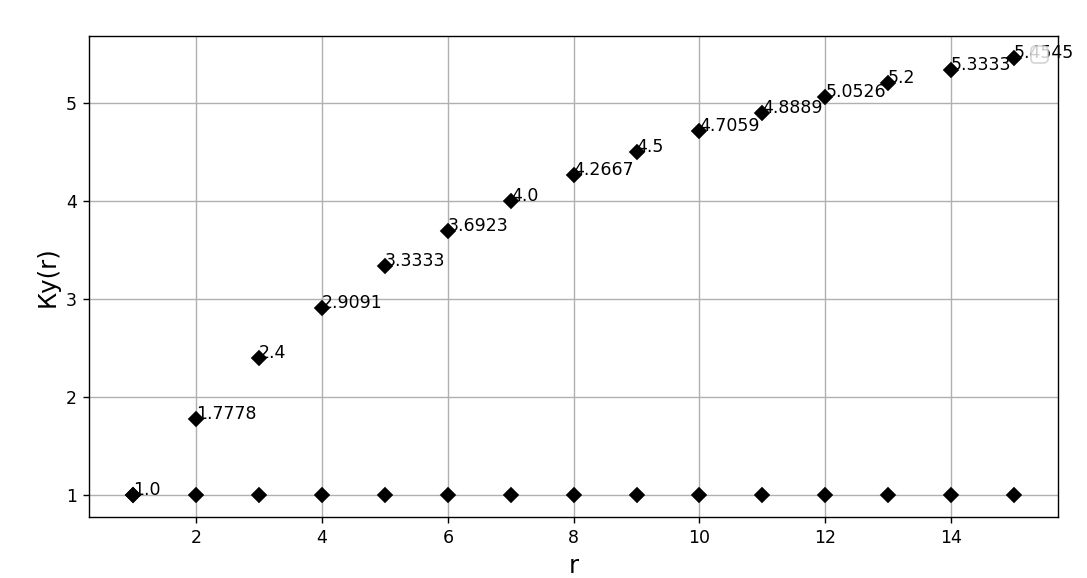
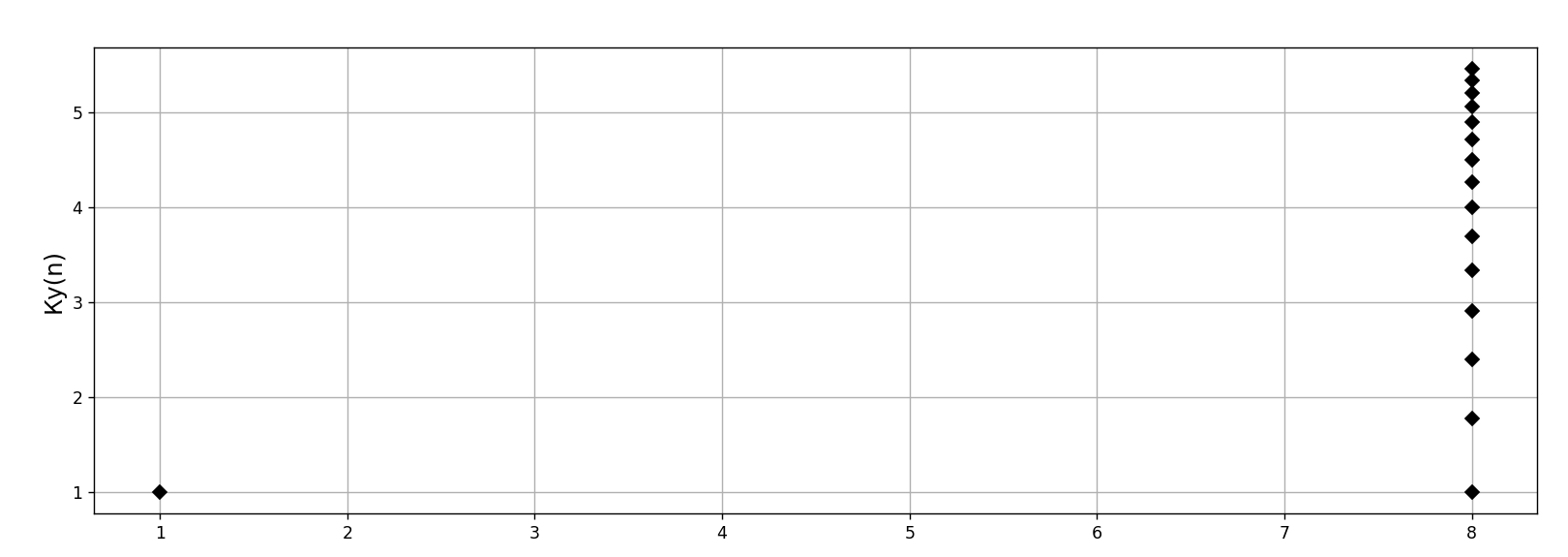
График зависимости Ку от n:

График зависимости эффективности e от количества этапов конвейера n и количества объектов, которые в процессе решения задачи могли бы обрабатываться параллельно (количества пар) r:

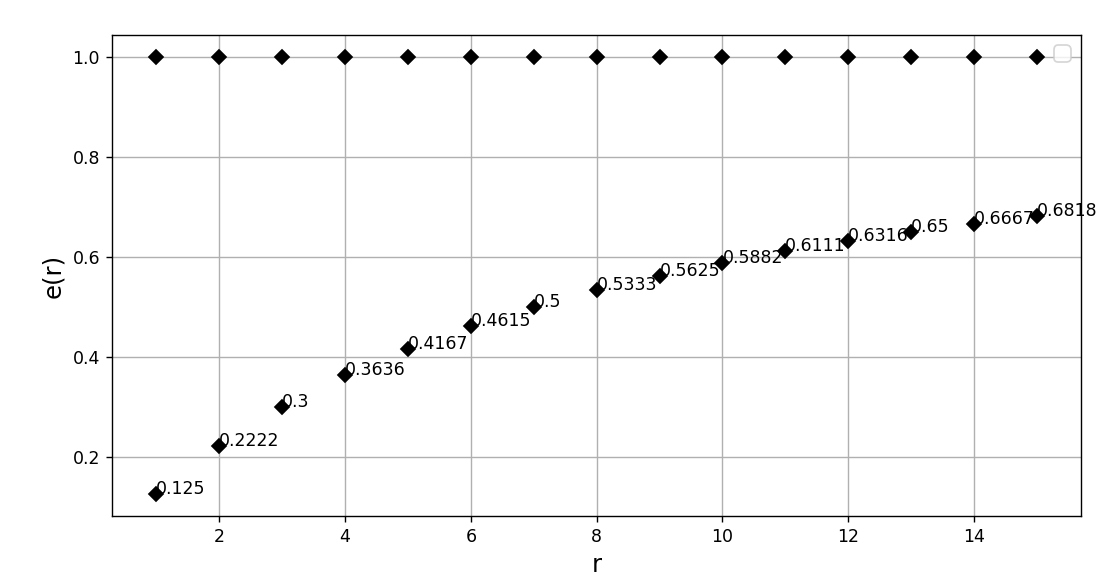
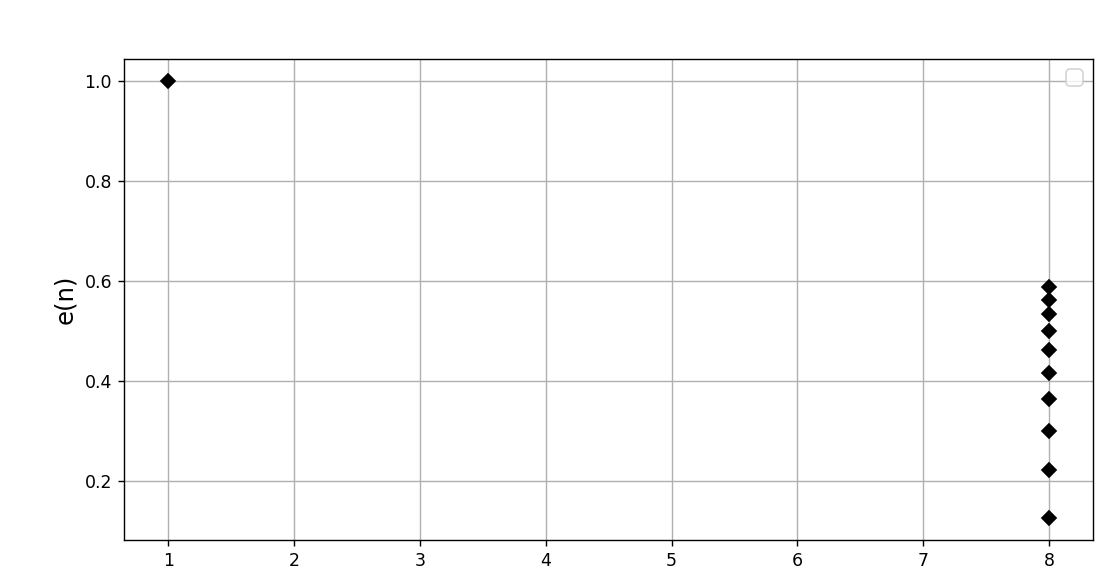
****

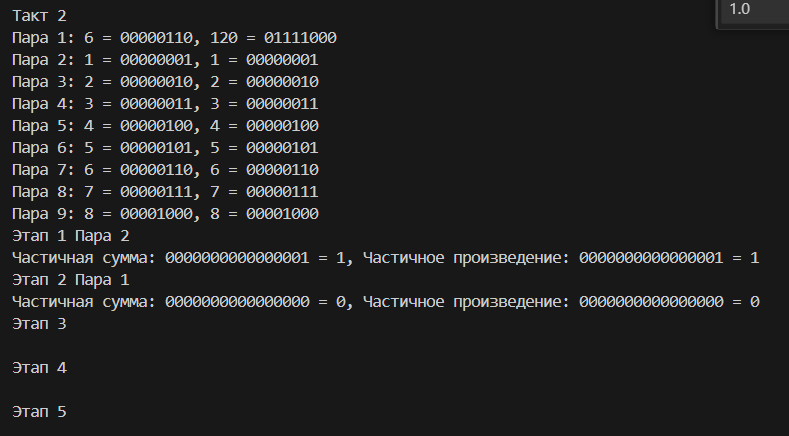
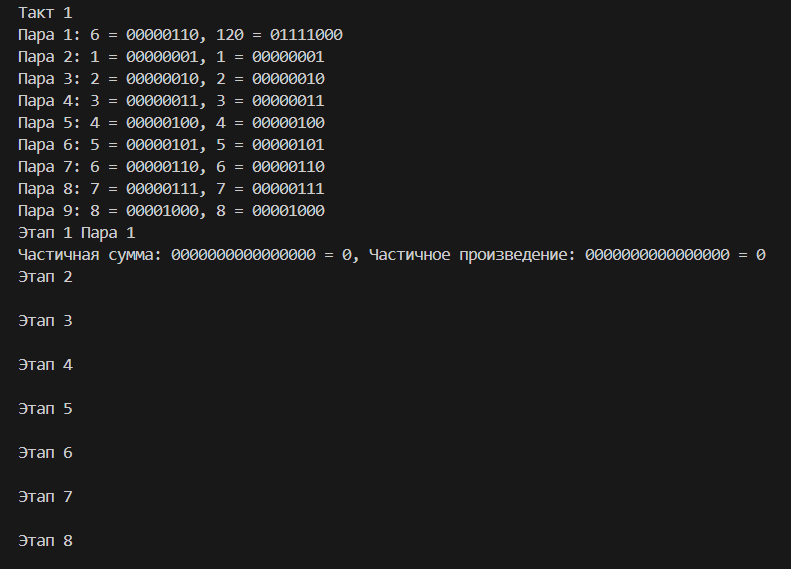
График зависимости е от n:

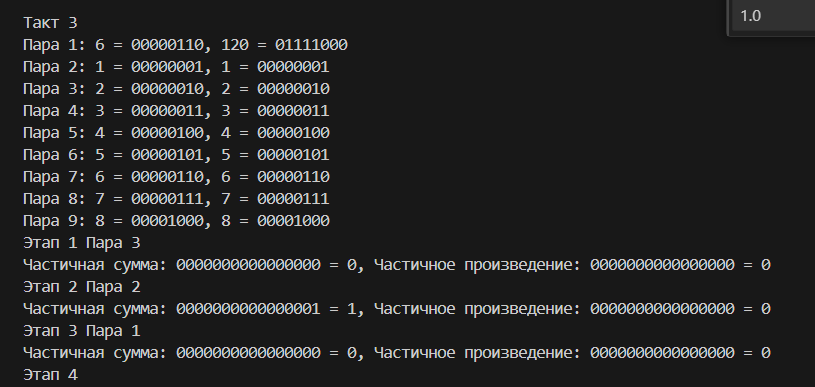
****

**Вопросы:**

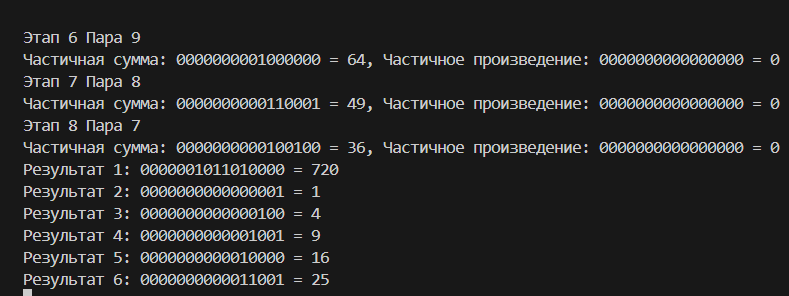
1. Проверить, что модель создана верно: программа работает правильно (на всех этапах конвейера).

Введем девять пар чисел, в том числе пару 6 и 120, как в примере выше.

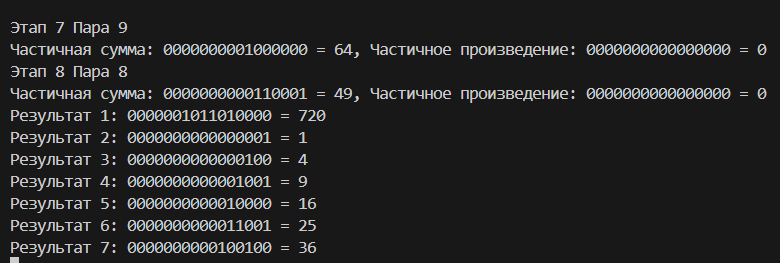


...

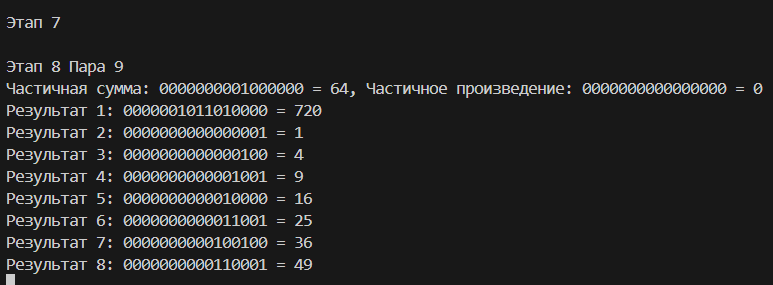
Такт 14:



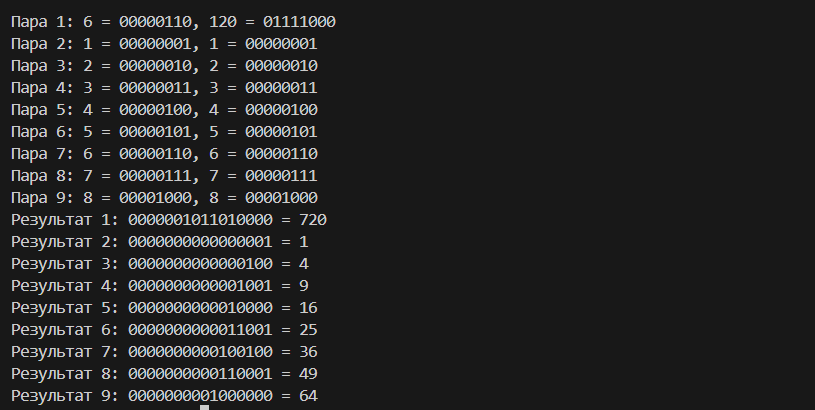
Такт 15:



Такт 16:



Результаты:

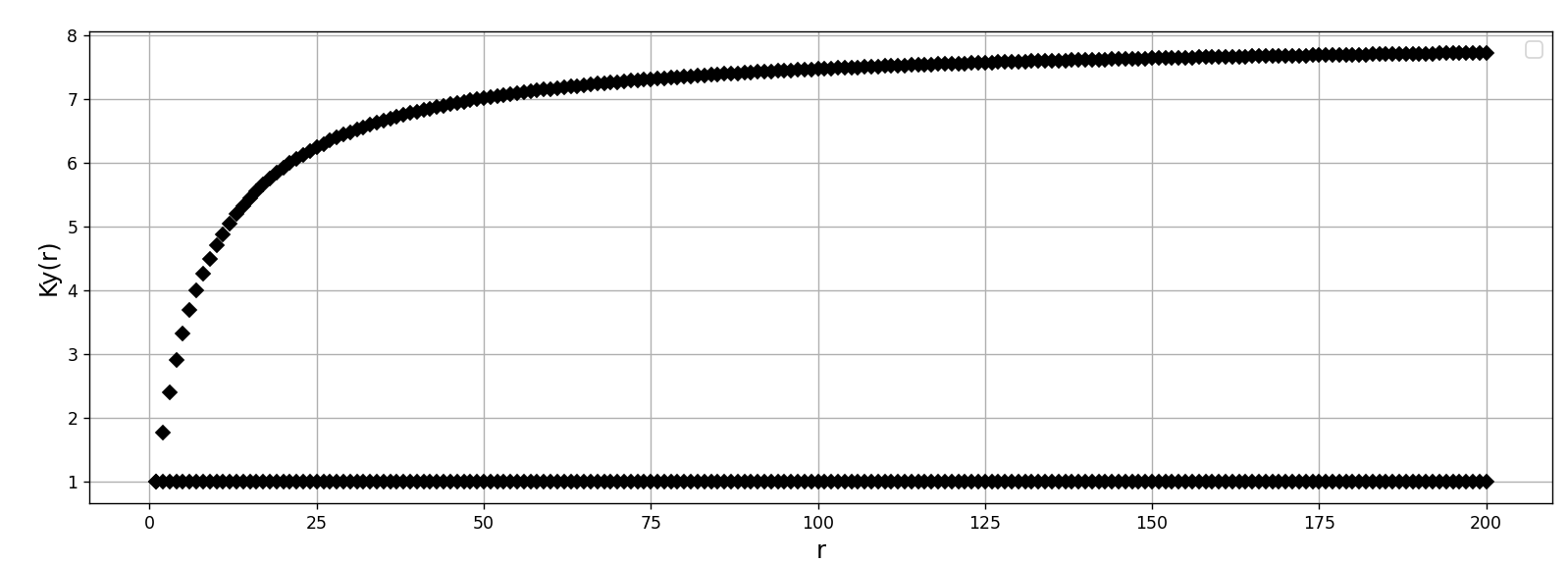


Результаты произведения верны. Для первой пары каждый этап модели совпал с соответствующим этапом в примере алгоритма.

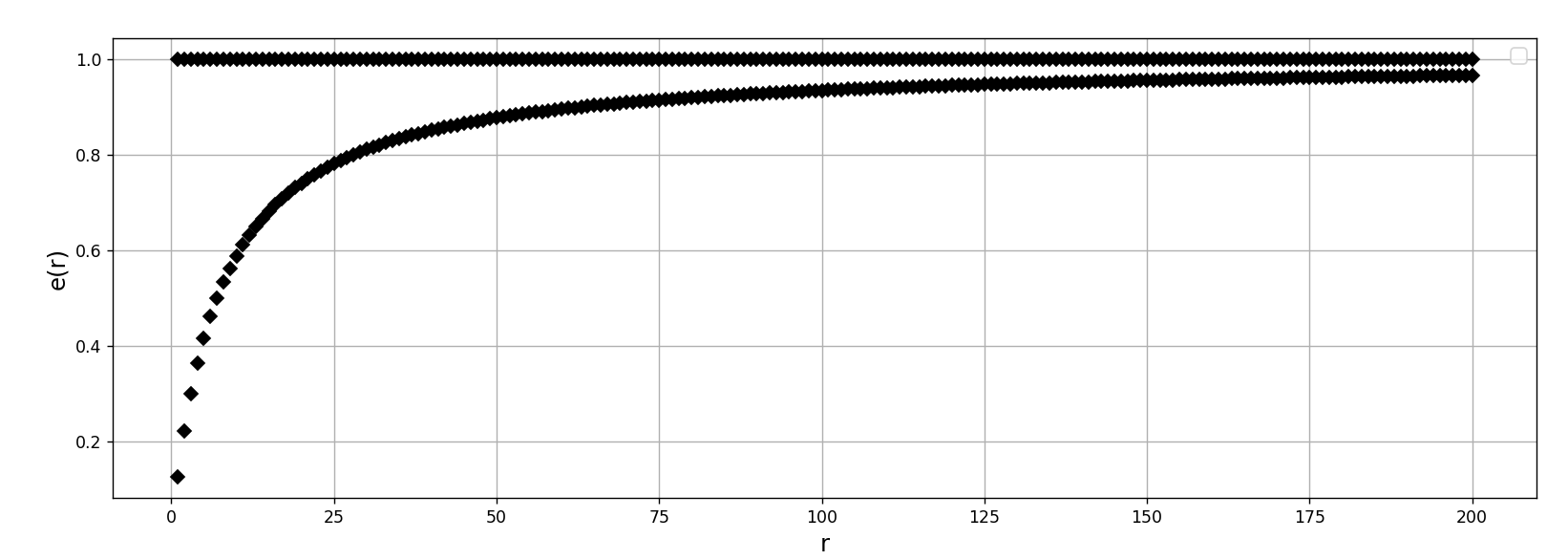
Можно сделать вывод, что модель работает верно.

1. Объяснить на графиках точки перегиба и асимптоты.

Возьмем значительно больший промежуток r (от 1 до 200). На графике становятся видна горизонтальная асимптота кривой, равная n, которое, в свою очередь, в соответствии с данным вариантом лабораторной работы, равно 8.



Аналогичным образом увеличив количество точек графика изменения эффективности, видим , что его горизонтальная асимптота равна 1.



Точек перегиба нет, так как графики монотонно возрастающие .

1. Спрогнозировать как изменится вид графиков при изменении параметров модели.

: при увеличении ранга задачи r и(или) увеличении количества процессорных элементов n, увеличивается и значение коэффициента, соответственно точки графика приближаются к асимптоте.

График : при увеличении ранга задачи r, значение эффективности увеличивается, соответственно точки графика приближаются к асимптоте 1, при увеличении количества процессорных элементов n – снижается.

1. Каково соотношение между параметрами **n**, **r**, **m**, **p** модели сбалансированного конвейера?m – количество пар - задается пользователем;  
   n = p = 8;

r = m.

1. Допустим: имеется некоторая характеристика h (эффективность e или ускорение ) и для неё выполняется: , . Каким будет соотношение между и ?

Допустим, =

=

=

=

=

Выразим и :

= + 1

= + 1

Подставим в неравенство, данное по условию :

+ 1 + 1

* при > 0:

> 1

<

<

* при < 0:

< 1

Случай с < 1 не имеет смысла, так как система с количеством процессорных элементов меньше 1 не имеет смысла.

* при = 0:

= 1

В случае с = 0 получаем противоречие, так как по условию , следовательно, не может быть равен 1, если строго больше , минимальное значение которого 1.

***При верно:***

* при > 0:

* при < 0:

Случай с < 0 не имеет смысла, так как система с отрицательным количеством объектов, которые в процессе решения задачи могли бы обрабатываться параллельно, не имеет смысла.

* при = 0:

< 0

Случай с = 0 не имеет смысла, так как система с нулевым количеством объектов, которые в процессе решения задачи могли бы обрабатываться параллельно, не имеет смысла.

Следовательно, соотношение между и при и 0 :

Ответ: при и 0

1. Дано:   
   1. несбалансированный конвейер (заданы конкретные значения: n, {} – времена выполнения обработки на этапах конвейера);   
   2**.** – некоторое фиксированное значение эффективности.

Определить значение , при котором выполняется ? (Получить формулу, затем подставить в нее значения параметров.)

В общем случае

;

;

;

Следовательно

;

>

***При верно:***

Для несбалансированного конвейера всегда верно: > 0

* при > 0: – удовлетворяет наложенным условиям > 0
* при < 0: – противоречит условию , так как при , , > 0 числитель > 0, a < 0

=> < 0 в то время, как по условию .

* при = 0:

– противоречит наложенным условиям , так как при , , > 0 числитель > 0

Следовательно, при

Ответ: при

1. Для несбалансированного конвейера (использовать исходные данные предыдущего вопроса) определить: .

= =

= = =

Ответ: =

1. Дан несбалансированный конвейер (использовать исходные данные предыдущего вопроса). Каким образом можно перестроить данный конвейер, чтобы для заданного выполнялось ?

Для того, чтобы выполнилось при известном необходимо наложить ограничение на :

Для несбалансированного конвейера: .

Следовательно, по условию дано

При :

Случай с не имеет смысла, так как конвейер с отрицательным или нулевым количеством процессорных элементов не имеет смысла.

При :

Случай с не имеет смысла, так как конвейер с отрицательной или нулевой эффективностью не имеет смысла, поэтому должны быть реализованы проверки на то, какие пользователь может задать.

Таким образом, конвейер можно перестроить путем изменения количества его процессорных элементов(или этапов) так, чтобы для было верно . Добиться этого можно путем объединения этапов конвейера.

Ответ: перестроить путем изменения количества его процессорных элементов(или этапов) так, чтобы для было верно (объединить этапы).

1. Дан несбалансированный конвейер (использовать исходные данные предыдущего вопроса) и значение минимального кванта времени (условной временной единицы). Каким образом нужно перестроить данный конвейер, чтобы получить максимально быстрый конвейер? Получить для него формулы ?

Для того, чтобы получить максимально быстрый конвейер, нужно перестроить так, чтобы он стал сбалансированным, и каждый этап выполнялся за минимальное время , следовательно, = . Добиться этого можно путем разделения этапов конвейера.

Ответ: изменить количество этапов так, чтобы они выполнялись за одинаковое минимальное время (разделить этапы)*,* , .

**Вклад:**

Витковская С. И. – реализация логики программы, тестирование и отладка системы, составление отчета.

Мойсевич А. В. – разработка консольного интерфейса, тестирование и отладка системы, составление отчета.

**Вывод:** В результате выполнения лабораторной работы была реализована модель решения задачи вычисления произведения пары 8-разрядных чисел умножением с младших разрядов со сдвигом множимого влево на конвейерной архитектуре. Была выполнена отладка и тестирование модели. Были созданы и проанализированы графики двух характеристик конвейерной архитектуры: коэффициента ускорения и эффективности.

**Использованные источники:**

1. Модели решения задач в интеллектуальных системах. В 2 ч. Ч.1: Формальные модели обработки информации и параллельные модели решения задач: учеб.-метод. пособие/ В. П. Ивашенко. – Минск : БГУИР, 2020. – 79 с.
2. Методы и алгоритмы обработки данных [Электронный ресурс]. Режим доступа: https://studref.com/636037/ekonomika/vychislitelnyy\_konveyer (дата обращения 05.03.2024).
3. Принцип векторной обработки. Векторно-конвейерные и матричные вычислительные системы. Факторы снижающие производительность векторных систем. [Электронный ресурс]. Режим доступа: https://www.icmm.ru/ob-institute/kontakty/35-stranitsy-sotrudnikov/502-printsip-vektornoj-obrabotki-vektorno-konvejernye-i-matrichnye-vychislitelnye-sistemy-faktory-snizhayushchie-proizvoditelnost-vektornykh-sistem (дата обращения 05.03.2024).