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 3

Выполнила:

студент гр. 221703 Демидовец Д.В.

Проверил: Ивашенко В. П.

Минск, 2025

**Тема:** Программирование операций обработки данных и знаний с конвейеризированной обработкой потока данных.

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

**Постановка задачи:** реализовать модель сбалансированного конвейера для вычисления произведения пары 4-разрядных чисел умножением со старших разрядов со сдвигом множимого вправо.

**Описание лабораторной работы:** требуется реализовать алгоритм вычисления произведения пары n-разрядных чисел умножением со старших разрядов со сдвигом частичной суммы вправо. Входом программы является количество пар чисел, вводимых пользователем с клавиатуры и каждая пара, представленая в десятичной форме, которая затем будет переведена в двоичную систему, выходом программы является текстовое отображение, представляющее процесс работы арифметического конвейера в виде таблицы.

**Использованные структуры данных:**

1. Массивы - структура данных, хранящая набор элементов, идентифицируемых по индексу.

3. Conveyer – класс, который моделирует работу арифметического конвейера для поэтапного умножения пар четырёхразрядных двоичных чисел. Основная задача класса — обеспечивать последовательную передачу данных между стадиями, выполнение арифметических действий и формирование итоговых результатов умножения.

Для реализации программы был выбран язык программирования Python.

**Описание программы:**

Программа реализует следующие ключевые компоненты:

1. Арифметические методы:

* to\_binary(value) переводит десятичное число в двоичную систему в виде списка битов.
* to\_binary\_direct(value) аналогична предыдущей, но дополняет результат до 4 битов.
* to\_decimal(binary\_list) преобразует список битов обратно в десятичное число.
* summ(x1, x2) выполняет сложение двух двоичных чисел в виде списков и возвращает их сумму с учётом переноса.

1. Утилитарные методы:

* input\_number(length) обрабатывает ввод пар чисел пользователем, преобразует в двоичную форму и валидирует ввод.
* print\_v(binary\_list) форматирует вывод двоичных чисел для отображения.
* print\_list\_v(list\_of\_binary\_lists) объединяет форматированные значения в строку для отображения всех чисел пары.

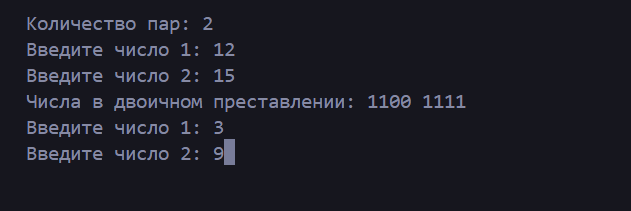
1. Класс Conveyer

Отвечает за реализацию конвейерной архитектуры и содержит следующие методы:

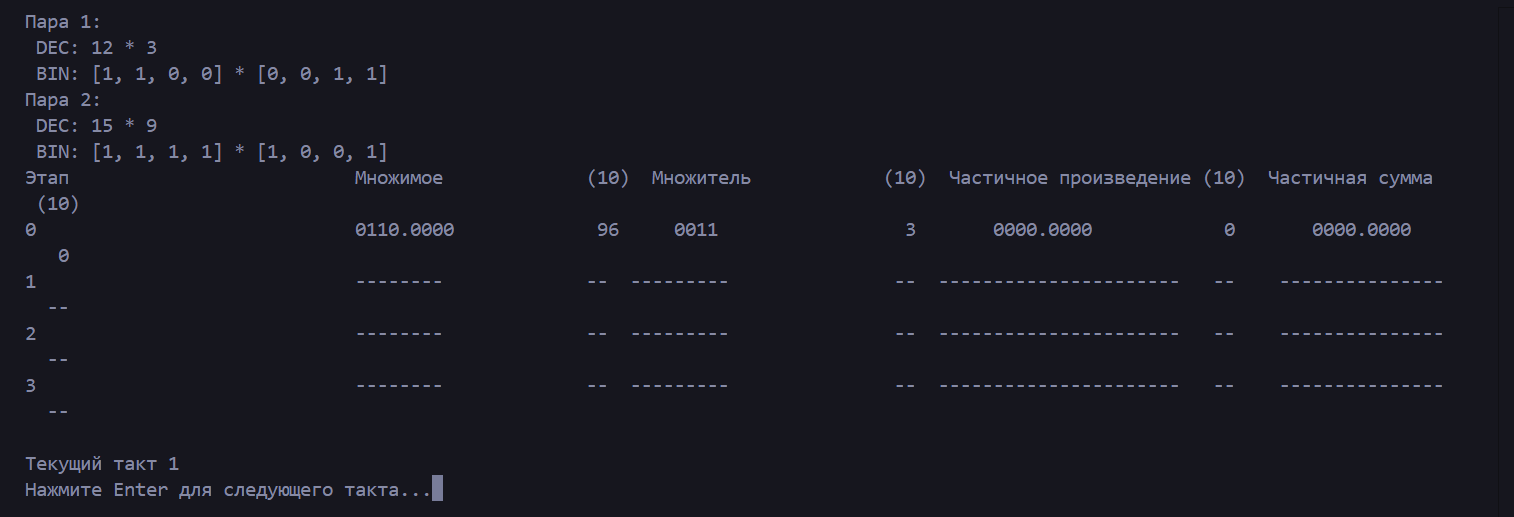
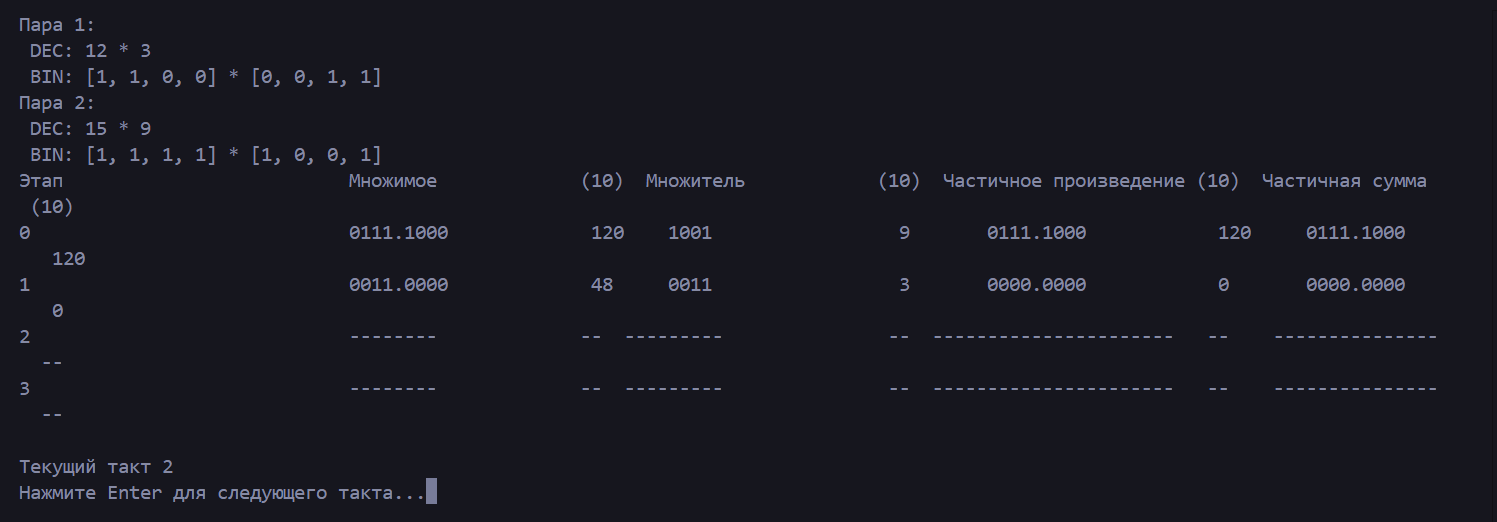
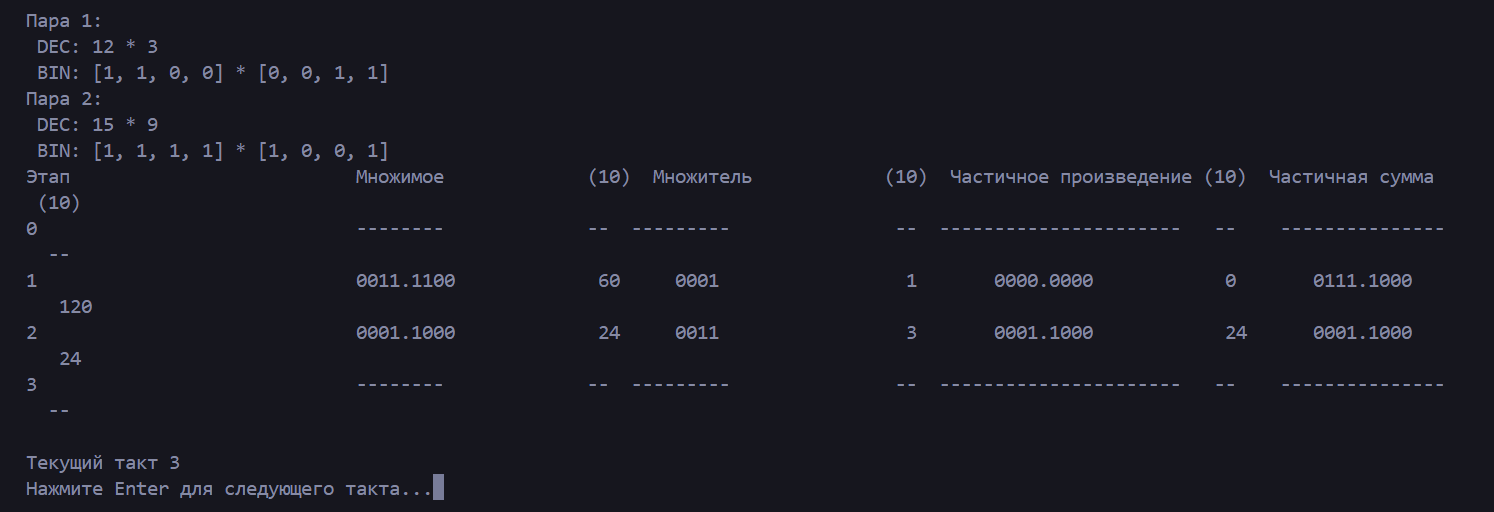
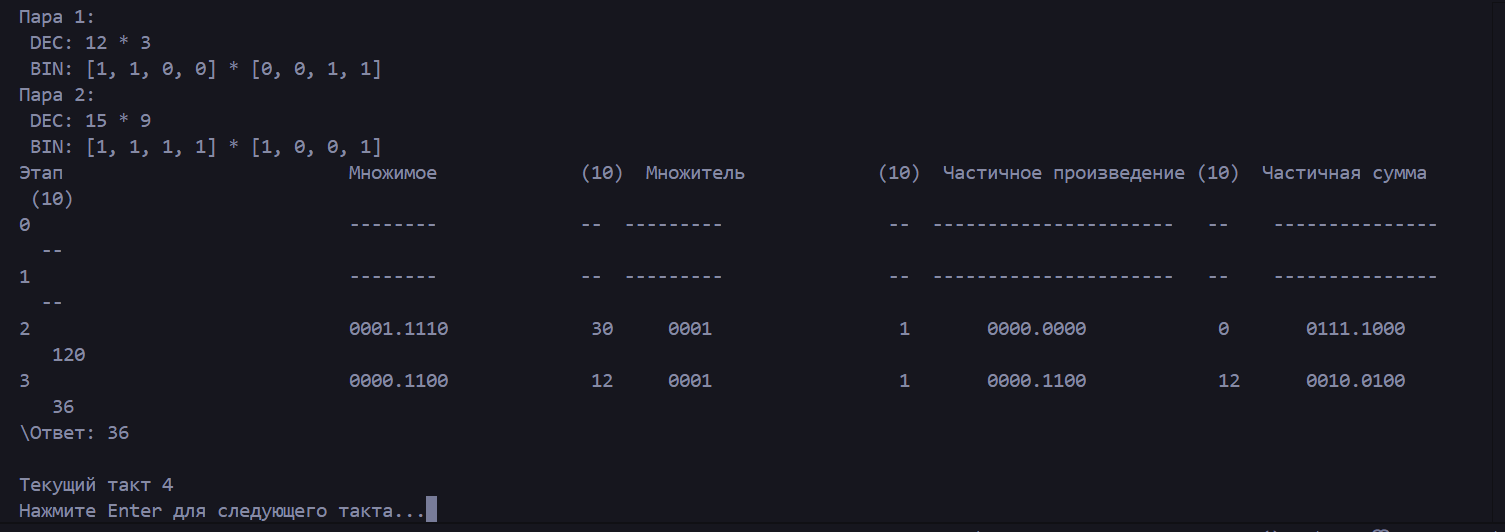
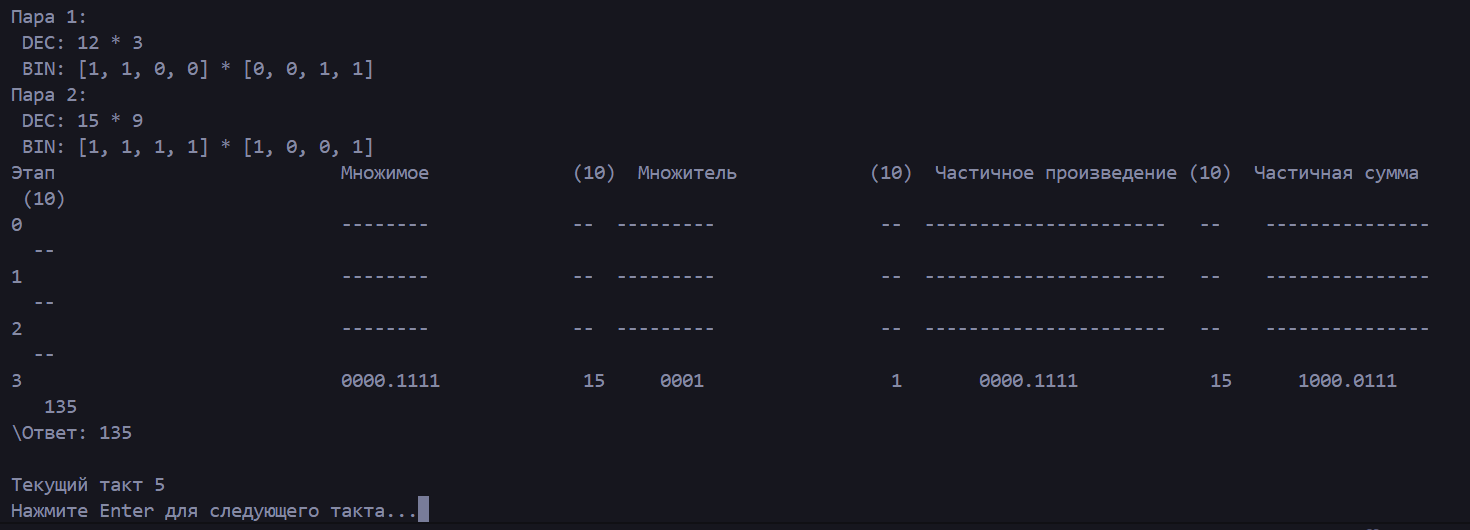
* init() инициализирует все данные, включая состояния конвейера, входные пары, промежуточные и конечные значения.
* init(index) подготавливает новую пару чисел для загрузки в конвейер (множимое расширяется до 8 бит, множитель остаётся 4-битным).
* step(prev\_state) выполняет один шаг на одной стадии конвейера:
  + сдвигает множимое вправо;
  + проверяет текущий бит множителя;
  + если бит равен 1 – множимое прибавляется к частичной сумме как частичное произведение;
  + возвращает обновлённое состояние стадии.
* tact() управляет тактами работы конвейера:
  + передаёт состояния между стадиями;
  + вызывает step() для обработки текущего состояния;
  + запускает вывод промежуточных результатов.
* output(state\_stage, clock\_cycle) отображает данные для всех стадий текущего такта, включая двоичные и десятичные значения множимого, множителя, частичного произведения и накопленной суммы.
* get\_results() возвращает финальные результаты произведений для всех пар.

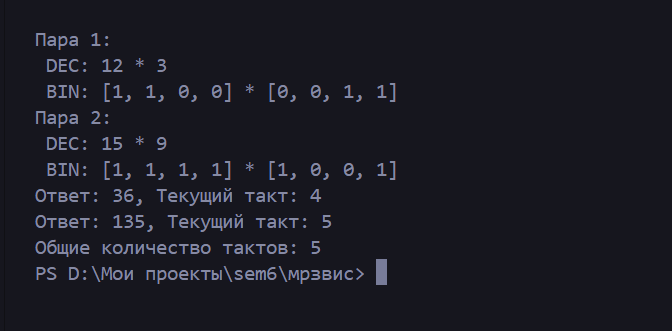
**Демонстрация результатов работы программы:**

Входные данные:



Работа программы:



**Графики:**

Обозначения:

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

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

T1(n, r) = r ∗ n – время, затрачиваемое на вычисления в однопроцессорной вычислительной системе.

Tn(n, r) = n + r − 1, при условии, что r > 0 – время, затрачиваемое на вычисления в параллельной вычислительной системе.

Ky = T1/Tn – коэффициент ускорения.

e = Ky/n – эффективность.

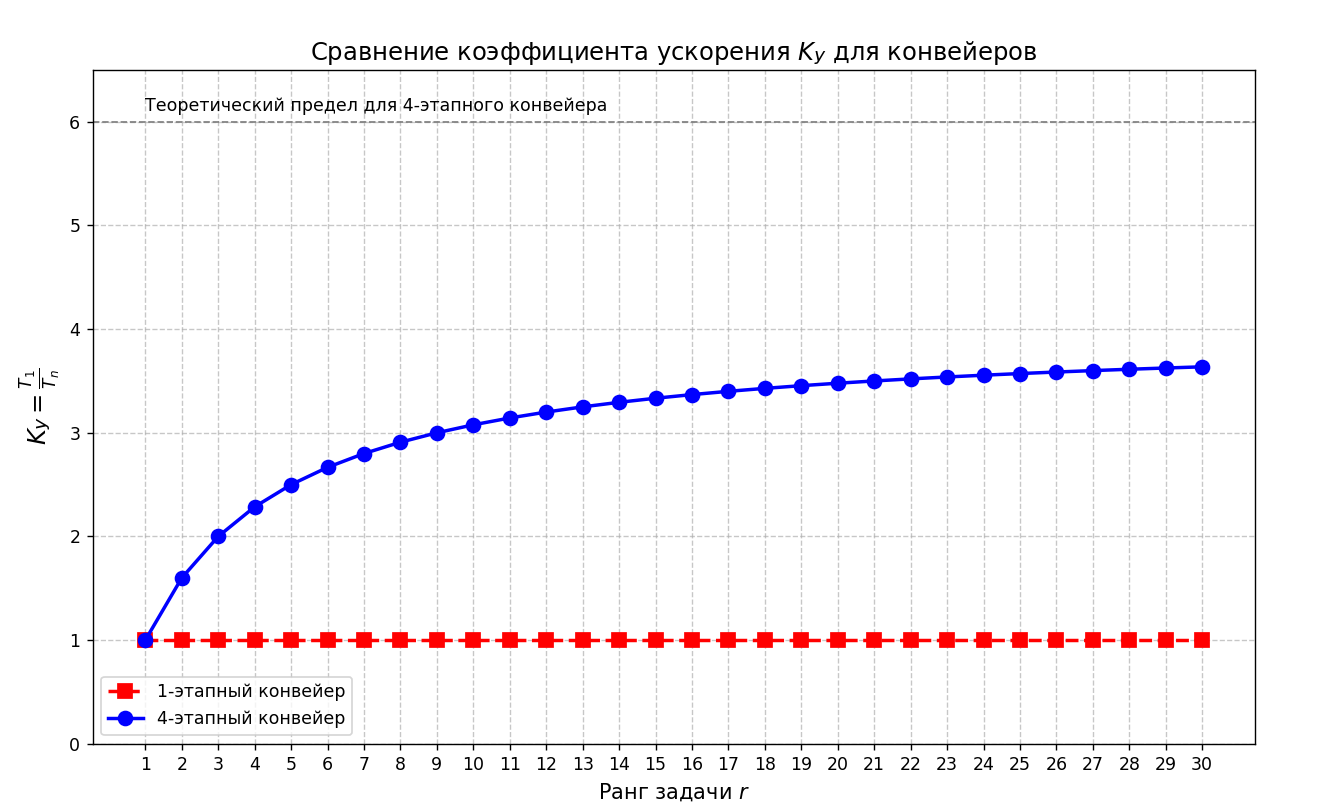
****

График зависимости коэффициента ускорения Ky от ранга задачи r

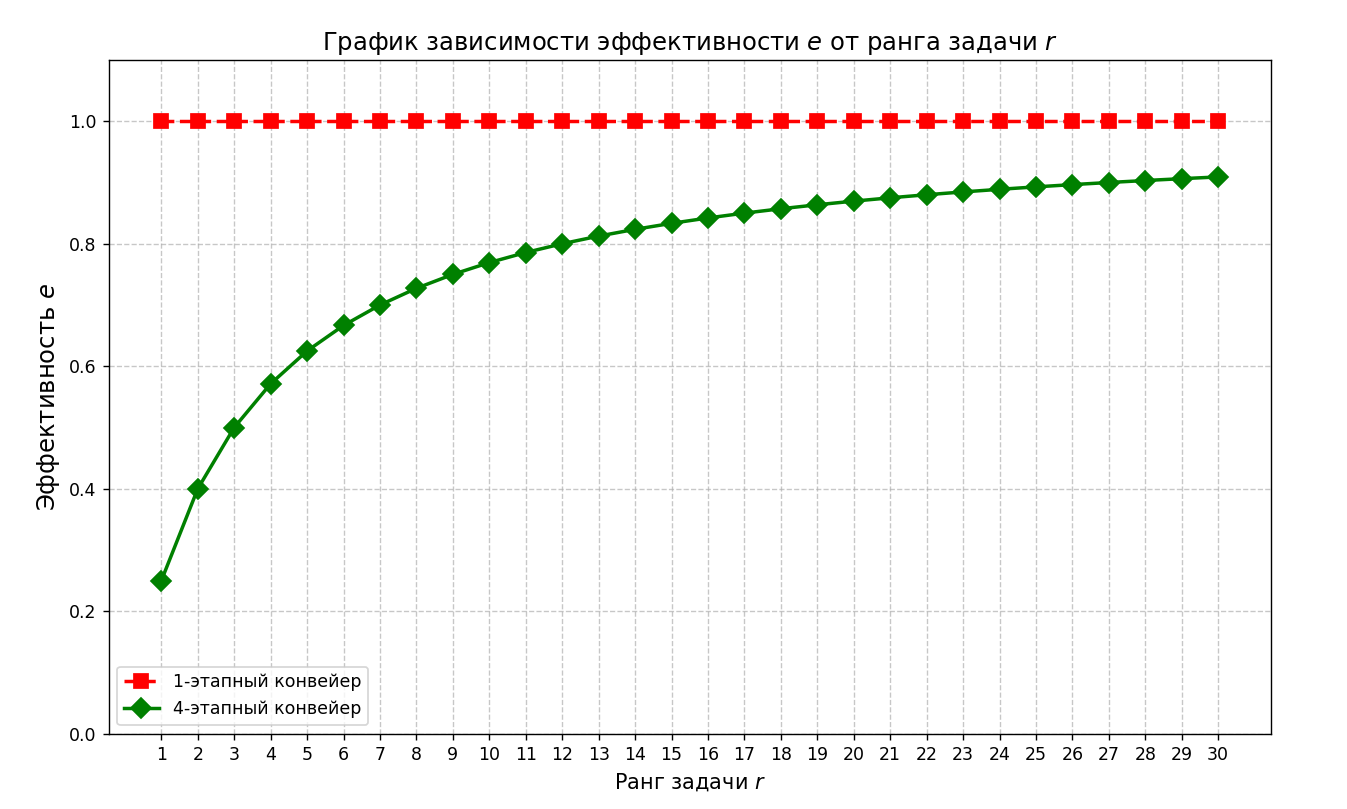
****

График зависимости эффективности e от ранга задачи r

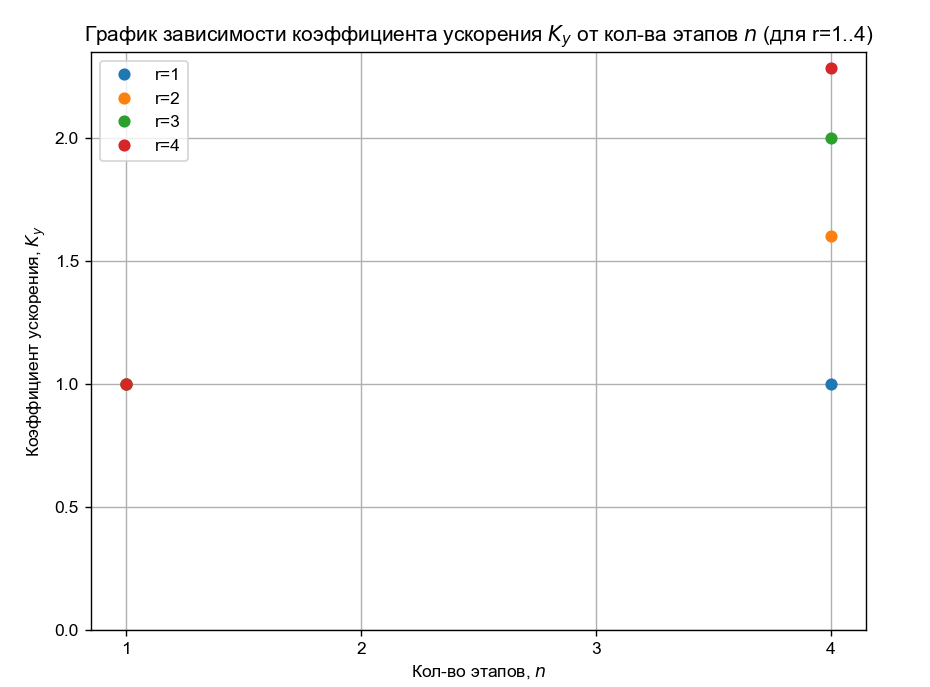
****

График зависимости коэффициента ускорения Ky от количества процессорных элементов n

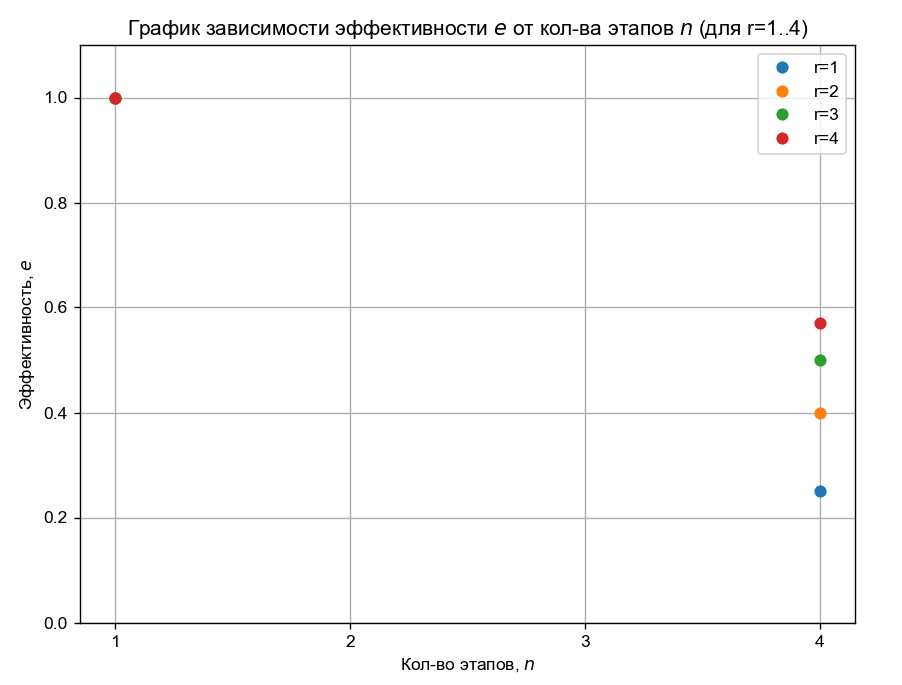
****

График зависимости эффективности e от количества процессорных элементов n

**Контрольные вопросы:**

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

Ответ: В разделе демонстрации работы конвейера, на каждом этапе работы конвейера были получены верные вычисления, результат работы программы является правильным.

2) Объясните на графиках точки перегиба и асимптоты.

Ответ:

Асимптоты при r стремящемся к ∞

Для коэффициента ускорения Ky = (n\*r)/(n+r-1):

1. Исходный предел:
2. Разделим числитель и знаменатель на r:
3. Уберём слагаемые в знаменателе, которые стремятся к нулю:
4. Найденный предел равен n

Таким образом, асимптотическое значение коэффициента ускорения приближается к n, что подтверждается графиком.

Для эффективности e = r/(n+r-1):

1. Исходный предел:

1. Разделим числитель и знаменатель на r:
2. Уберём слагаемые в знаменателе, которые стремятся к нулю:

Найденный предел равен 1.

Таким образом, асимптотическое значение эффективности приближается к 1, что подтверждается графиком.

Точки перегиба. Чтобы найти точки перегиба, нужно решить уравнение, где вторая производная функции от r равна 0.

Для коэффициента ускорения:

1. Найдём первую производную функции (n\*r)/(n+r-1):
2. Найдём вторую производную функции:
3. , при n = 0 и n = 1

Для эффективности:

1. Найдём первую производную функции r/(n+r-1):
2. Найдём вторую производную:
3. 1-n = 0, при n = 1.

Точка перегиба при n = 1.

3) Спрогнозируйте, как изменится вид графиков при изменении параметров модели.

Ответ: при увеличении n, Ky растёт, e уменьшается. При увеличении r, Ky растёт, е растёт.

4) Каково соотношение между параметрами n, r, m, p модели сбалансированного конвейера?

Ответ: Ранг задачи (r) — это максимальное количество экземпляров данных одного типа, которые необходимо обработать одновременно, независимо от способа реализации. В данном случае экземпляры данных одного типа — это количество пар чисел (m), следовательно:

r = m

Разрядность умножаемых попарно чисел обозначается как p. В данном варианте:

p = 4

Количество процессорных элементов (n) совпадает с разрядностью чисел и с количеством этапов конвейера, поэтому:

n = p

5) Допустим: имеется некоторая характеристика h (эффективность e или ускорение Ky) и для неё выполняется:

a) h(n1, r1) = h(n2, r2)

б) n1 > n2

Каким будет соотношение между r1 и r2?

Ответ: Пусть h – эффективность, тогда e(n1, r1) = e(n2, r2), при n1 > n2.

1. e1 = T1/(Tn\*n1)=(r1\*n1)/((n1+r1-1)\*n1)= r1/(n1+r1-1)
2. e2 = T1/(Tn\*n2)=(r2\*n2)/((n2+r2-1)\*n2)= r2/(n2+r2-1)
3. e1 = e2
4. r1/(n1+r1-1)= r2/(n2+r2-1)
5. r1\*(n2+r2-1)=r2\*(n1+r1-1)
6. r1n2+r1r2-r1=r2n1+r2r1-r2
7. r1n2-r1=r2n1-r2

8. r1\*(n2-1)=r2\*(n1-1)

9. Т.к. n1 > n2 и n1∈N и n2∈N, r1∈N и r2∈N N – множество натуральных чисел, из пункта 8 r1\*(n2-1)=r2\*(n1-1), то r1 > r2

6) Дано:

а) Несбалансированный конвейер (заданы конкретные значения: n, {ti} – времена выполнения обработки на этапах конвейера);

б) e0 – некоторое фиксированное значение эффективности.

Определить значение r0, при котором выполняется e(n, r0) > e0?

Ответ:

1. ,
2. ,
3. Из пункта 3:

:

Подходит под условия, т.к. числитель больше знаменателя

:

Знаменатель не может быть равен нулю, поэтому такой случай нет смысла рассматривать

:

Числитель принимает значения больше 0, знаменатель меньше 0, следовательно результат выражения будет меньше 0, что не подходит под условие r0≥1

Итоговое выражение:

7) Для несбалансированного конвейера (использовать исходные данные предыдущего вопроса) определить: lim(e(n, r)), r -> 1/0.

Ответ:

1. ,
2. ,
3. Разделим числитель и знаменатель на r
4. Т.к и стремятся к 0, то

8) Дан несбалансированный конвейер (использовать исходные данные предыдущего вопроса). Каким образом можно перестроить данный конвейер, чтобы для заданного r0 выполнялось e(n, r0) > e0?

Ответ:

Мы можем попытаться изменить n в перестроенном конвейере.

1. ,
2. ,

, n

5. Т.к. n , то

1. e0 > 0, т.к конвейер с отрицательной или нулевой эффективностью не имеет смысла, то
2. 1

Таким образом, чтобы перестроить конвейер мы можем уменьшить n так, чтобы выполнялось условие из пункта 7, соединив соседние этапы.

9) Дан несбалансированный конвейер (использовать исходные данные предыдущего вопроса) и значение минимального кванта времени t0(условной временной единицы).

Каким образом нужно перестроить данный конвейер, чтобы получить максимально быстрый конвейер? Получить для него формулы Ky(n, r), e(n, r)

Ответ:

Необходимо, чтобы каждый этап конвейера выполнялся за t0 (минимальный квант времени), следовательно нужно разделить те этапы конвейера, которые выполняются за время большее t0 на несколько этапов. Выразим кол-во этапов такого конвейера:

Выразим Ky и e (учитывая, что конвейер теперь сбалансированный, т.к. каждый этап выполняется за t0):

**Вывод:** в ходе лабораторной работы была разработана модель арифметического конвейера для умножения пар четырёхразрядных чисел методом пошагового анализа битов множителя со сдвигом множимого вправо. Алгоритм реализован в виде четырёхстадийного конвейера, где на каждом такте вычисляется частичное произведение и накапливается сумма. Работа позволила на практике освоить принципы конвейерной обработки и структурировать алгоритм с помощью объектно-ориентированного подхода. Также были изучены такие числовые характеристики конвейерной архитектуры, как коэффициент ускорения Ky и эффективность e.

**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.