Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

№ кода и наименование направления подготовки

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 22

Выполнил:

студент гр. ИП-411 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Выдрин Андрей Э./

подпись

Проверил:

доцент кафедры ВС

к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /А.В. Ефимов /

ОЦЕНКА, подпись

Новосибирск 2016

ОГЛАВЛЕНИЕ

[1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС](#_Toc469228175)

[1.1. ЗАДАНИЕ 3](#_Toc469228176)

[1.2. ОТВЕТ 3](#_Toc469228177)

[1.2.1 Аппаратная часть 3](#_Toc469228178)

[1.2.2 Программная часть 4](#_Toc469228179)

[1.2.3 Вывод 5](#_Toc469228180)

[2. ОТВЕТ НА ВТОРОЙ ВОПРОС](#_Toc469228181)

[2.1 ЗАДАНИЕ 6](#_Toc469228182)

[2.2 ОТВЕТ 6](#_Toc469228183)

[2.2.1 P-алгоритм умножения двух матриц 6](#_Toc469228184)

[2.2.2 Коэффициент накладных расходов 9](#_Toc469228185)

# 1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС

## 1.1. ЗАДАНИЕ

Обосновать необходимость использования парадигмы мультиархитектуры в суперВС.

## 1.2. ОТВЕТ

Основной целью при создании вычислительных суперсистем является достижение максимальной эффективности при решении прикладных задач. Эта цель состоит из двух взаимосвязанных компонентов – эффективности использования аппаратных средств и эффективности программного обеспечения. Рассмотрим, как использование парадигмы мультиархитектуры влияет на эти компоненты.

### **1.2.1 Аппаратная часть**

Технологические возможности и конструктивные особенности определяют два основных параметра, влияющих на производительность системы, – тактовую частоту и объем оборудования, от которого зависит степень параллелизма. При создании архитектуры и аппаратной реализации суперсистем основным вопросом является выбор такой формы параллелизма, которая при соответствующей оптимизации программ обеспечит рост производительности пропорционально росту объема аппаратуры. На аппаратном уровне основные формы параллелизма – это совокупность параллельно функционирующих устройств, одинаковых или разнородных (параллельные структуры) и объединение в конвейерную структуру различных функционально специализированных устройств.

В суперсистемах параллельные структуры объединяют функциональные устройства, процессоры или машины. Для повышения эффективности желательна высокая степень локализации обработки данных в каждом из таких устройств и сокращение потоков данных между ними.

В конвейерных структурах степень параллелизма обработки данных определяется числом станций конвейера. Оно соответствует числу подфункций, на которые может быть разбита функция. Такие структуры эффективны, когда массивы данных для обработки достаточно велики и имеются средства их доставки на вход конвейера. Следует подчеркнуть, что эффективность конвейера пропорциональна числу станций (при сохранении их загруженности), т.е. его длине, поскольку скорость передачи данных зависит не от числа станций, а только от быстродействия логических схем и расстояния между соседними станциями.

Очевидно, что в суперсистемах должны быть использованы обе формы параллелизма на аппаратном уровне, и только от их оптимального взаимодополнения зависит конечная эффективность системы.

### **1.2.2 Программная часть**

Большинство алгоритмов и программ ориентировано на последовательные вычисления. Однако зачастую их можно подвергать параллельной обработке. Например, при наличии больших массивов данных возникает так называемый параллелизм на уровне данных. Наиболее эффективно он используется в векторных машинах. Если программу можно разбить на множество независимых или слабо связанных подзадач, то в ней присутствует параллелизм на уровне задач.

Доминирующая в задаче форма параллелизма должна соответствовать форме параллелизма на уровне аппаратуры, преобладающей в вычислительной системе. Если это условие не выполняется, необходима специальная подготовка программ. Например, задачи с параллелизмом на уровне данных (содержащие

большие массивы) при решении на системе с чисто параллельной структурой разбиваются на отдельные части (фрагменты массива). При этом необходима специальная программа для стыковки и объединения результатов фрагментарных вычислений. Для эффективного решения на векторной машине программ с параллелизмом на уровне задач необходимо эти задачи объединить для лучшей векторизации программы. При переносе задач с одной системы на другую возникают большие трудности, особенно если в этих системах преобладают разные формы параллелизма.

Следовательно, наиболее эффективна вычислительная система,

содержащая подсистемы с различной архитектурой и формами параллелизма, – мультиархитектурная система.

### **1.2.3 Вывод**

Архитектура мультиархитектурной вычислительной суперсистемы может

повысить эффективность использования аппаратных и программных средств.

Однако в ней возникают новые проблемы, связанные с выявлением формы параллелизма того или иного участка программы и дальнейшей его оптимизацией. Также должны быть решены проблемы распределения ресурсов, планирования выполнения программы и собственно управления ее выполнением.

# 2. ОТВЕТ НА ВТОРОЙ ВОПРОС

## 2.1 ЗАДАНИЕ

Построить блок-схему p -алгоритма умножения матриц:

W[1:M; 1:N] , Y[1:L; 1:M],

обеспечивающего распределение элементов результирующей матрицы по горизонтальным полосам в элементарных машинах ВС.

Отыскать максимум коэффициента ε накладных расходов при реализации p -алгоритма на вычислительной системе, имеющей следующие параметры:

– разрядность *l* = 32;

– полосу пропускания канала между машинами ν = 100 Мегабод;

– время выполнения операции сложения t­c = 10 нс;

– время выполнения операции умножения tу = 100 нс.

## 2.2 ОТВЕТ

### **2.2.1 P-алгоритм умножения двух матриц**

W[1:M; 1:N] \* Y[1:L; 1:M] = C[1:L; 1:N]

Пусть требуется построить параллельный алгоритм, вычисляющий произведение двух прямоугольных матриц:

Элементы матрицы-произведения C[1:L; 1:N] вычисляются по формуле. (1)

Минимум емкости памяти будет достигнут, если каждая из исходных матриц будет разбита на n равных частей, и в каждый вычислитель будет размещено по одной такой части матриц W и Y. Каждую из матриц W и Y можно разрезать на n равных соответственно горизонтальных и вертикальных полос.

Причем в n-м вычислителе строки

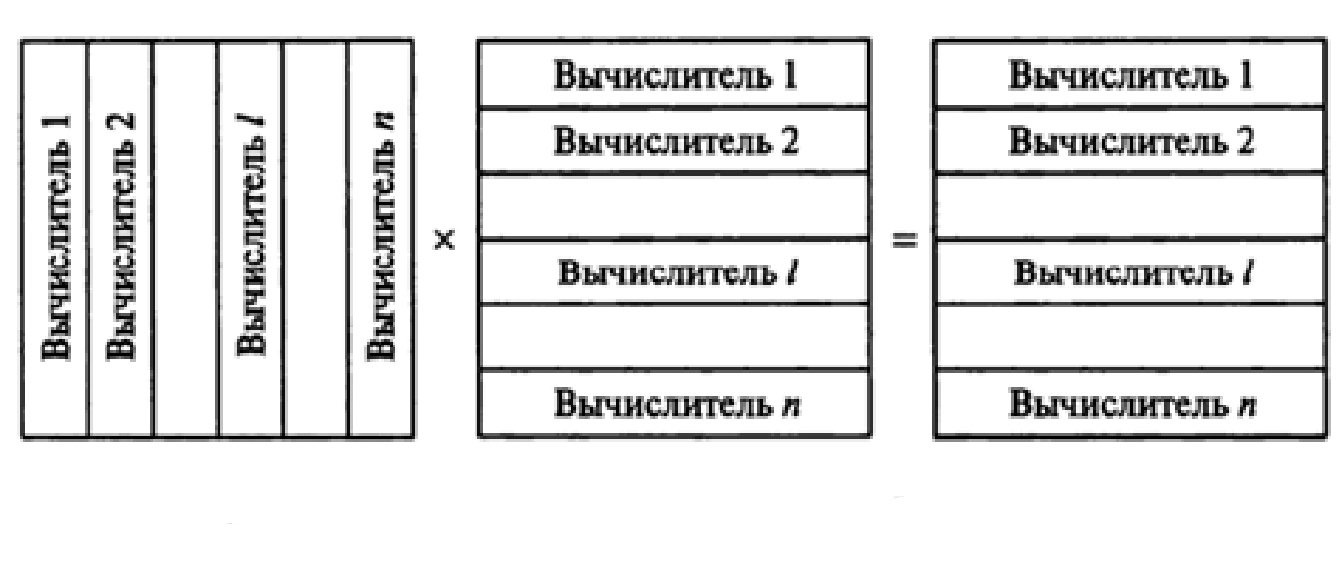
столбцы

матриц W и Y соответственно.

Через ]x[ обозначено такое ближайшее к x целое число, для которого справедливо неравенство .

Параллельный вычислительный процесс можно организовать следующим способом. Сначала первый вычислитель передает остальным вычислителям первый столбец матрицы W. После этого каждый из вычислителей по формуле **(1)** рассчитывает ]M/n[ элементов первой строки своей полосы для результирующей матрицы C. Затем первый вычислитель рассылает во все остальные вычислители второй столбец матрицы W и производятся вычисления элементов второй строки матрицы C и так до тех пор, пока первый вычислитель не перешлет все строки.

После этого пересылками будут заниматься последовательно второй вычислитель, третий вычислитель и далее до n-го вычислителя. Матрица C получается распределенной по вычислителям, причем в каждом будет своя горизонтальная полоса.



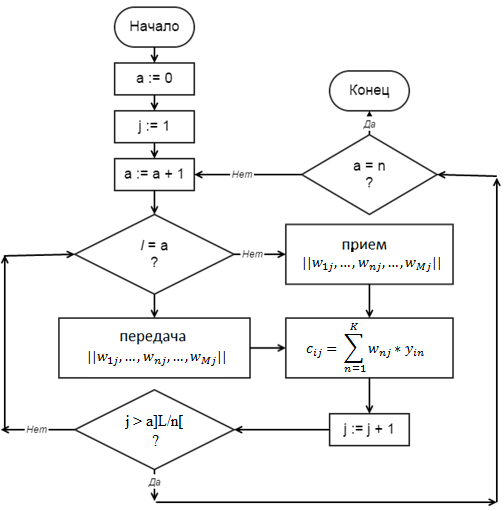


Рис. Схема ветви параллельного алгоритма умножения матриц

а - номер передающего вычислителя;

*l* – ветвь выполняющая вычисления

n - число вычислителей в системе;

### **2.2.2 Коэффициент накладных расходов**

Эффективность параллельного алгоритма умножения матриц большого размера можно характеризовать показателями:



Очевидно, что максимум накладных расходов будет при , или, что то же самое, равенство  достигается при  Таким образом, максимум коэффициента ε накладных расходов определяется формулой:

ε = tn / (ty + tc)

tn – время пересылки

tу – время умножения

tс  - время сложения

tn = *l* / ν = 32 / 100 \* 106 = 320 нс, тогда

ε = 320 / (100 + 10) = 2,909090