Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 Супервычисление

№ кода и наименование направления подготовки

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 23

Выполнил:

студент гр. ИС-441 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Мазанов А. С./

подпись

Проверил:

доцент кафедры ВС

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

ОЦЕНКА, подпись

Новосибирск 2016

Оглавление

[1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС 2](#_Toc470378069)

[1.1. ЗАДАНИЕ 2](#_Toc470378070)

[1.2. ОТВЕТ 2](#_Toc470378071)

[1.2.1. Выполнить сравнительный анализ вычислительных систем с архитектурами MISD и SIMD 2](#_Toc470378072)

[1.2.2 Примеры функциональной структуры промышленных ВС 6](#_Toc470378079)

[2. ОТВЕТ НА ВТОРОЙ ВОПРОС 9](#_Toc470378081)

[2.1 ЗАДАНИЕ 9](#_Toc470378082)

[2.2 ОТВЕТ 9](#_Toc470378083)

[2.2.1 P-алгоритм умножения двух матриц 9](#_Toc470378084)

[2.2.2 Коэффициент накладных расходов 12](#_Toc470378085)

# ОТВЕТ НА ПЕРВЫЙ ВОПРОС

## ЗАДАНИЕ

Выполнить сравнительный анализ вычислительных систем с архитектурами MISD и SIMD. Привести примеры функциональной структуры промышленных ВС.

## 1.2. ОТВЕТ

### 1.2.1. Выполнить сравнительный анализ вычислительных систем с архитектурами MISD и SIMD

Модель коллектива вычислителей есть диалектическое обобщение модели вычислителя, следовательно, вычислительные системы в сравнении с ЭВМ Дж. фон Неймана являются принципиально новыми средствами техники обработки информации, средствами с качественно новыми архитектурными возможностями. В самом деле, в архитектурном плане выделяют четыре типа архитектур вычислительных средств: SISD, MISD, SIMD, MIMD или, при использовании русской аббревиатуры, ОКОД, МКОД, ОКМД, МКМД.

Архитектуры MISD и SIMD относятся к вычислительным системам. В этих архитектурах имеет место “множественность” потоков или команд, или данных. Множественность характеризуется количеством одновременно реализуемых потоков команд или данных. Архитектура MISD (Multiple Instruction stream / Single Data stream) или МКОД (Множественный поток Команд и Одиночный поток Данных) позволяет нескольким потокам команд обрабатывать один поток данных. Архитектура SIMD (Single Instruction stream / Multiple Data stream) или ОКМД (Одиночный поток Команд и Множественный поток Данных) предоставляет возможность одному потоку команд обрабатывать несколько потоков данных.

Приведенная классификация архитектур средств обработки информации была предложена профессором Стенфордского университета США М. Дж. Флинном (M.J. Flynn) в 1966 г. и получила широкое распространение.

В архитектурах типа MISD, SIMD (рис.1) допустимо построение нескольких классов вычислительных систем, среди которых наибольший интерес представляют: 1)конвейерные ВС, 2)матричные ВС, 3)ВС с программируемой структурой.

Конвейерные ВС – это класс систем, архитектура которых является предельным вариантом эволюционного развития последовательной ЭВМ и простейшей версией модели коллектива вычислителей. В основе таких систем лежит конвейерный (или цепочечный) способ обработки информации, а их функциональная структура представляется в виде “последовательности” связанных элементарных блоков обработки (ЭБО) информации. Все блоки работают параллельно, но каждый из них реализует лишь свою операцию над данными одного и того же потока. Сказанное позволяет относить конвейерные ВС к MISD-системам (рис. 1). Реальные промышленные высокопроизводительные ВС являются, как правило, мультиконвейерными. В них единое управляющее устройство (управляющая ЭВМ или подсистема, контроллер и т.п.) формирует один поток команд и несколько параллельных потоков данных на подсистемы–конвейеры. Последнее обстоятельство позволяет относить такие мультиконвейерные ВС к системам с архитектурой SIMD.

Матричные ВС основываются на принципе массового параллелизма, в них обеспечивается возможность одновременной реализации большого числа операций на элементарных процессорах (ЭП), “объединенных” в матрицу. Каждый ЭП – композиция из арифметико-логического устройства (АЛУ) и локальной памяти (ЛП); последняя предназначается для хранения части данных (но не части программы или параллельной ветви!). Поток команд на матрицу ЭП формируется устройством управления (следовательно, оно имеет в своём составе память для хранения программ обработки данных). Такие ВС рассчитаны, в частности, на решение задач матричной алгебры. Они имеют SIMD-архитектуру в классическом виде.

Вычислительные системы с программируемой структурой полностью основываются на модели коллектива вычислителей и являются композицией взаимосвязанных элементарных машин (ЭМ). Каждая ЭМ в своем составе обязательно имеет локальный коммутатор (ЛК), процессор и память; может иметь также внешние устройства. Локальная память ЭМ предназначается для хранения и части данных, и, главное, ветви параллельной программы. Архитектура ВС с программируемой структурой относится к типу MIMD. Такие ВС по своим потенциальным архитектурным возможностям не уступают ни одному из перечисленных выше классов систем. Они прежде всего ориентированы на распределенную обработку информации; эффективны и при конвейерной, и при матричной обработке. При распределенном способе обработки данных на ВС полностью используются возможности MIMD-архитектуры. При конвейерном и матричном способах обработки данных архитектура MIMD виртуально трансформируется соответственно в архитектуру MISD и SIMD. Системы с программируемой структурой рассчитываются на работу во всех основных режимах: решения сложной задачи, обработки наборов задач, обслуживания потоков задач, реализации функций вычислительной сети.

MISD-архитектура;

Конвейерные ВС







**…**

**…**

# Конвейер элементарных блоков обработки

Поток данных



Память

**Поток результатов**

**. . .**

**. . .**

Рисунок 1.1.

SIMD-архитектура;

## Матричные ВС

поток команд,

поток данных

Устройство управления

****

****

****

****

****

****

****

****







****

****





**…**

**…**

**…**

**…**

**…**

**…**

**…**

**АЛУ**

ЛП

**ЭП – элементарный процессор**

Рисунок 1.2.

**АЛУ** – арифметико-логическое устройство, **ЛП** – локальная память,

**ЛК** – локальный коммутатор,

### 1.2.2 Примеры функциональной структуры промышленных ВС

1. Вычислительная система DAP

Разработку матричной ВС DAP (Distribut ед Аггау Processor распре- деленный матричный процессор) осуществляла английская фирма ICL (International Computers Ltd.). Работы были начаты в 1972 г., опытные образцы системы DAP (на дискретных элементах) из1024 и 4096 элементарных процессоров были построены в 1976 г. и 1977 г.

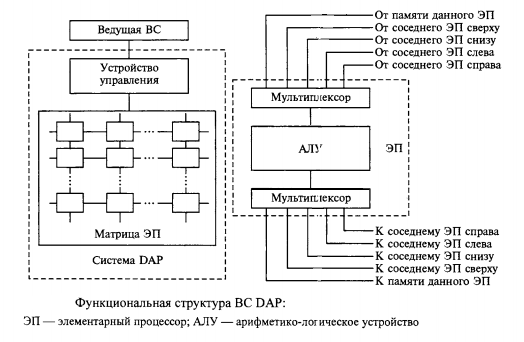


Рисунок 1.3.

1. Вычислительная система СМ-1

Вычислительная система СМ-1 первая модель семейства Connection Machine была спроектирована в Thinking Machines Corp. в течение 1983 г. и первой половины 1984 г. Прототип модели СМ-1 из 16 K процессоров был построен к концу 1984 г. при финансовой поддержке Агентства по перспективному планированию исследований в области обороны США (Defense Advanced Research Projects Agency). Демонстрация возможностей прототипа СМ-1 была осуществлена в мае 1985г. Полная 65 536-процессор- ная конфигурация СМ-1 была собрана и успешно продемонстрирована в ноябре 1985 г.

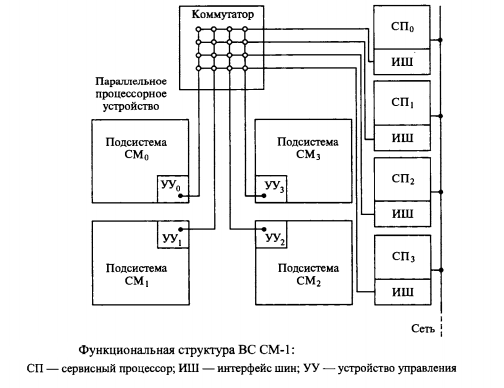


Рисунок 1.4

# Вычислительная система Cray-1

Создание ВС Cray-1 было завершено фирмой Cray Research в 1976 г. Быстродействие системы составляло 160 MFL ОРД (16 • 10' опер./c при вы- полнении операций c плавающей запятой над векторами данных и 37 млн опер./с над скалярами), емкость оперативной памяти 8...64 Мбайт, длина слова данных 64 двоичных разряда, цена 5...9 млн долл. Система Cray-1 успешно конкурировала на рынке супер ЭВМ вплоть до 1982 г. Вычислительнaя система Cray-1 предназначалась для векторной и скaлярной обработки данных. Эта система сс стояла из четырех функцио- нaльных подсистем: управления программой, конвейеров, регистров, памяти и ввода-вывода (Рисунок 1.5)

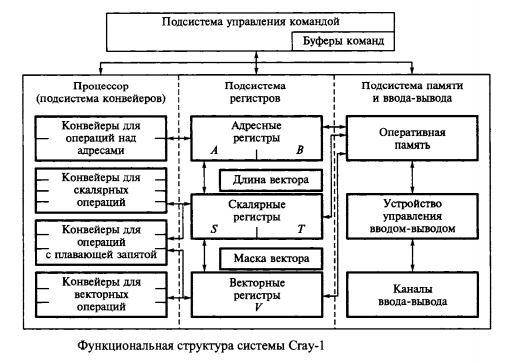


Рисунок 1.5

# 2. ОТВЕТ НА ВТОРОЙ ВОПРОС

## 2.1 ЗАДАНИЕ

Построить блок-схему p -алгоритма умножения матриц:

G[1:J; 1:K] , B[1:K; 1:H],

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* = 64;

– полосу пропускания канала между машинами ν = 5 Гигабод;

– время выполнения операции сложения t­c = 0,5 нс;

– время выполнения операции умножения tу = 1 нс.

## 2.2 ОТВЕТ

### 2.2.1 P-алгоритм умножения двух матриц

G[1:J; 1:K] \* B[1:K; 1:H] = C[1:J; 1:H

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

Элементы матрицы-произведения C[1:J; 1:H]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 G и B. Каждую из матриц G и B можно разрезать на n равных соответственно горизонтальных и вертикальных полос.

Причем в n-м вычислителе строки

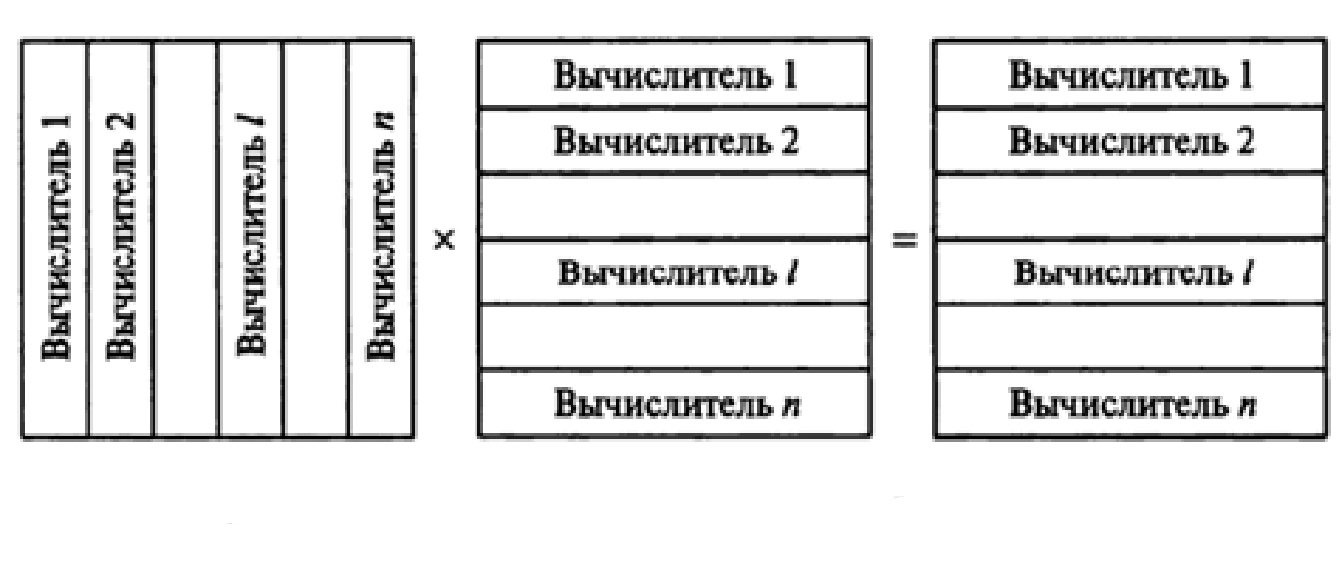
столбцы

матриц G и B 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 G. После этого каждый из вычислителей по формуле **(1)** рассчитывает ]J/n[ элементов первой строки своей полосы для результирующей матрицы C. Затем первый вычислитель рассылает во все остальные вычислители второй столбец матрицы G 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.



Началооо

a := 0

Конец

r := 1

нет

a = n

?

a := a + 1

l = a

?

прием

||,…,,…,||

нет

передача

||,…,,…,||

нет

r > a]J/n[

r := r + 1

да

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 = 64 / 5 \* 109 = 128 нс, тогда

ε = 128 / (1 + 0.5) = 85,3333