# Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01 Информатика и вычислительная техника

№ кода и наименование направления подготовки

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 15

Выполнил:

студент гр. ИП-411 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Беляев Р.А./

подпись

Проверил:

доцент кафедры ВС

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

ОЦЕНКА, подпись

Новосибирск 2017

ОГЛАВЛЕНИЕ

1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС..……………………………………………3

1.1 ЗАДАНИЕ…………………………………………………………………3

1.2 ОТВЕТ……………………………………………………………………..3

1.2.1 Анализ возможностей вычислительных систем …………………...3

1.2.2 Пример функциональной структуры промышленной ВС…………4

2 ОТВЕТ НА ВТОРОЙ ВОПРОС..…………………………………………….6

2.1 ЗАДАНИЕ…………………………………………………………………6

2.2 ОТВЕТ……………………………………………………………………..6

2.2.1 P-алгоритм умножения двух матриц………………………………..6

2.2.2 Коэффициент накладных расходов………………………………….9

# 1 ОТВЕТ НА ПЕРВЫЙ ВОПРОС

## 1.1 ЗАДАНИЕ

Осуществить анализ возможностей вычислительных систем с MISD-архитектурой.

Привести пример функциональной структуры промышленной (современной) ВС.

## 1.2 ОТВЕТ

**1.2.1 Анализ возможностей вычислительных систем**

MISD-архитектура (англ. Multiple Instruction stream, Single Data stream, множественный поток команд, одиночный поток данных, МКОД) — вычислительная система, принадлежащая к классификации Флинна. В MISD-архитектуре над одинаковыми данными производятся различные операции в множестве процессоров.

Систолический массив волновых процессоров, который был впервые описан Чарльзом Лейзерсоном и Цзун Сян Кунгом можно отнести к MISD-архитектуре. В систолическом массиве параллельные входные данные проходят через сеть процессорных узлов. Узлы могу сортировать, обрабатывать и сливать входные данные.

Преимущество систолических массивов заключается в том, что все входные данные и результаты промежуточных вычислений находятся в процессорах. Из-за этого не нужно использовать внутренний кэш, внешние шины или основную память в каждой операции, как в SISD-архитектуре. Систолические массивы отлично себя показывают в разработке искусственного интеллекта, обработке изображений, распознавании образов и других задач, которые легко решаются человеческим мозгом.

Так же к MISD-архитектуре относятся конвейерные вычислительные системы. Средства обработки информации, базирующиеся на конвейерных процессорах можно расположить между средствами, основанных на модели вычислителя, и между средствами, основанными на коллективе вычислителей.

Конвейерные вычислительные системы постоянно совершенствуются, стремясь к модели коллектива вычислителей.

Большое разнообразие конвейерных вычислительных систем обеспечивается тремя принципами, присущими всем конвейерным системам:

1. Параллельность выполнения операций. Её можно достичь параллельной работой различного числа процессоров, параллельной работой десятка конвейеров на уровне процессоров, или организацией конвейеров из десятков элементарных блоков обработки информации.
2. Программируемость структуры. Например, в ранних вычислительных системах используя специальный булевский вектор можно было добиться избирательности обработки компонентов векторов. В более поздних системах удалось добиться программирование межпроцессорных взаимодействий.
3. Конструктивная однородность. Даже в простых конвейерных системах имеется несколько идентичных конвейеров. Более поздние конвейерные системы полностью опираются на данный принцип.

Возможности повышения быстродействия при помощи конвейерной обработки информации сильно ограничены. Как пример может послужить тот факт, что элементарные блоки обработки информации имеют ограниченную надёжность, поэтому нельзя достичь произвольно длинной последовательности операций при передаче результатов по конвейеру. Из-за этого сейчас всё больше внимания уделяется более перспективным системам с SIMD и MIMD архитектурами.

**1.2.2 Пример функциональной структуры промышленной ВС**

В качестве примера ВС с MISD-архитектурой можно привести суперкомпьютер iWarp. Эта система разрабатывалась совместно университетом Карнеги — Меллон и Intel.

Каждый процессор iWarp имел 32-битный АЛУ и 64-битный математический сопроцессор. Процессор работал с тактовой частотой в 20МГц. Взаимодействие между процессорами осуществлялось отдельным модулем, имеющим четыре канала со скоростью 40Мб/с, и поддерживающим до 20 виртуальных каналов. В iWarp была использована более быстрая, но дорогая, статическая оперативная память.

Также отличительной особенностью этой ВС было то, что вычислители соединялись в торе, а не более часто используемом гиперкубе. Система имела 64 процессора, которые были соединены в 8х8 торе, при этом достигалась производительность в 1,2 GFLOPS.

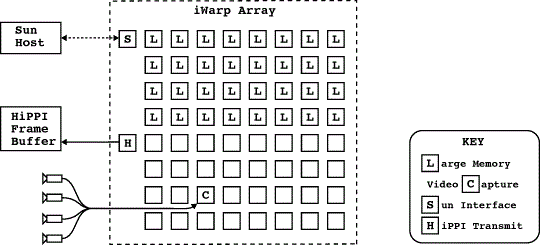


Рисунок 1.1 Функциональная структура системы iWarp

# 2. ОТВЕТ НА ВТОРОЙ ВОПРОС

**2.1 ЗАДАНИЕ**

Построить блок-схему p -алгоритма умножения двух матриц:

*,*

применив методику крупноблочного распараллеливания.

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

– разрядность ;

– полосу пропускания канала между машинами Гигабод;

– время выполнения операции сложения = 0,1 нс;

– время выполнения операции умножения = 1нс.

**2.2 ОТВЕТ**

**2.2.1 P-алгоритм умножения двух матриц**

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

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

(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 и . Каждую из матриц и можно разрезать на n равных соответственно горизонтальных и вертикальных полос.

Причем в n-м вычислителе строки

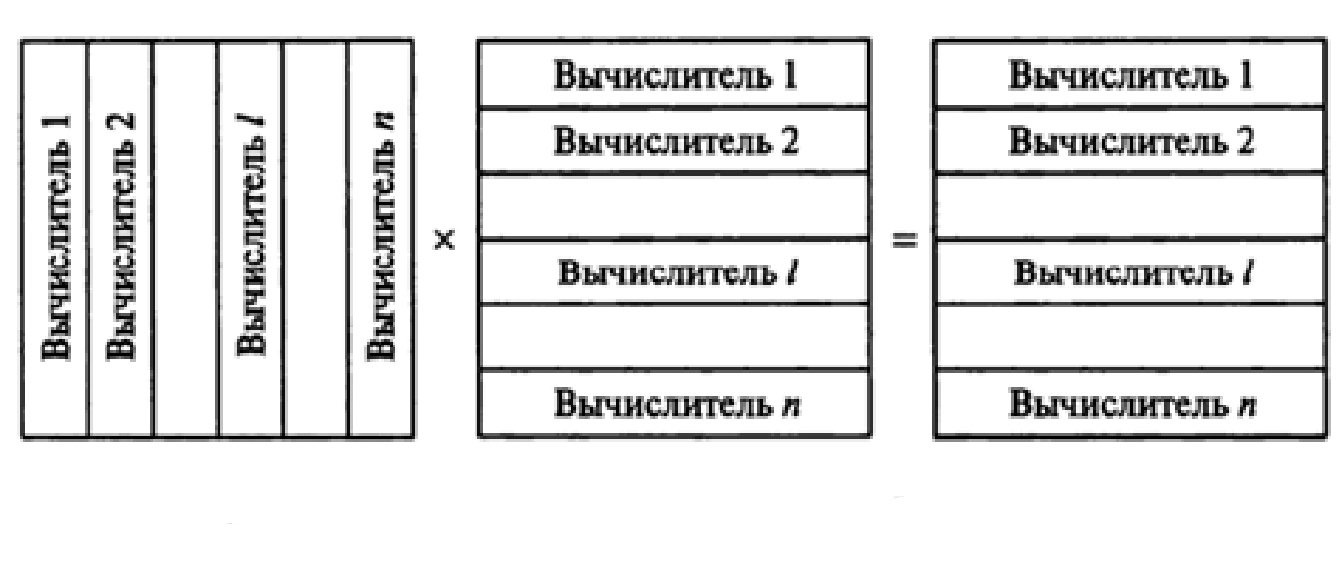
столбцы

матриц и соответственно.

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

Параллельный вычислительный процесс можно организовать следующим способом. Сначала первый вычислитель передает остальным вычислителям первый столбец матрицы . После этого каждый из вычислителей по формуле (1) рассчитывает ]N/n[ элементов первой строки своей полосы для результирующей матрицы C. Затем первый вычислитель рассылает во все остальные вычислители второй столбец матрицы 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.



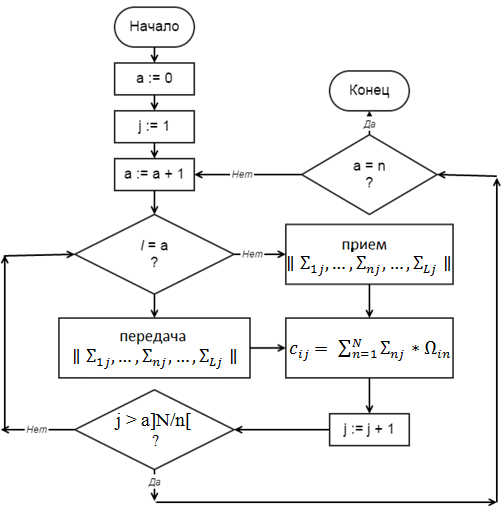


Рисунок 2.1 Схема ветви параллельного алгоритма умножения матриц

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 / 10 \* 109 = 3,2 нс, тогда

ε = 3,2 / (1 + 0,1) = 2,909090