Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ВС

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 26

Выполнил:

студент гр. ИП-312 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /~~Золотарев А.П~~./

подпись

Проверил:

доцент кафедры ВС

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

ОЦЕНКА, подпись

Новосибирск 2015

ОГЛАВЛЕНИЕ

[ОТВЕТ НА ПЕРВЫЙ ВОПРОС 3](#_Toc432276047)

[ЗАДАНИЕ 3](#_Toc432276048)

[ОТВЕТ 3](#_Toc432276049)

[ОТВЕТ НА ВТОРОЙ ВОПРОС](#_Toc432276047) 8

[ЗАДАНИЕ](#_Toc432276048) 8

[ОТВЕТ](#_Toc432276049) 8

# 1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС

## 1.1. ЗАДАНИЕ

Осуществить анализ возможностей вычислительных систем с SIMD-архитектурой. Привести пример использования SIMD-архитектуры в суперВС.

## 1.2. ОТВЕТ

Архитектура SIMD (Single Instruction – Multiple Data Stream) представляет возможность одному потоку команд обрабатывать несколько потоков данных.

SIMD архитектура связана с параллельной обработкой информации. Существует 4 типа SIMD архитектуры:

* Матричная структура
* Векторно-конвейерная
* Технология MMX
* Потоковые SIMD расширения

**Рассмотрим данные типы**

Процессор с матричной структурой представляет собой «матрицу». Суть такого процессора состоит в том, что есть множество элементарных процессоров, объединённых коммутатором, выполняющих одну команду над различными элементами матрицы. Основная проблема в трудоемкости программирования обмена данными между элементарными процессорами через коммутатор.

Основу конвейерной обработки составляет раздельное выполнение некоторой операции в несколько этапов (за несколько ступеней) с передачей данных одного этапа следующему. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько операций. При этом исчезает необходимость в использовании коммутатора, что служит основной проблемой в матричных процессорах.

При выполнении векторной команды одна и та же операция применяется ко всему конвейеру. Для загрузки конвейера может потребоваться какое-то время, но после загрузки операнды начинают поступать в конвейер с максимальной скоростью, которая возможна для используемой памяти.

Таким образом главный принцип вычислений на машине с конвейерной структурой состоит в выполнении одной операции над большим блоком данных.

Еще одним примером реализации SIMD-архитектуры является **технология MMX**, которая существенно улучшила архитектуру микропроцессоров фирмы Intel (Pentium MMX). Она разработана для ускорения выполнения мультимедийных и коммуникационных программ. Команды ММХ выполняют одну и ту же функцию с различными частями данных, например: 8 байт графических данных передаются в процессор как одно упакованное 64-х разрядное число и обрабатываются одной командой.

Следующим шагом по пути использования SIMD-архитектуры в микропроцессорах фирмы Intel (Pentium III) явились **потоковые SIMD-расширения – Streaming SIMD Etension (SSE),** которые реализуют новые SIMD-инструкции, оперирующие со специальными 128-битными регистрами. Каждый из этих регистров может хранить несколько упакованных целочисленных или вещественных данных. Таким образом, выполняя операцию над содержимым двух регистров под управлением команды SSE, процессор может обработать несколько пар операндов одновременно.

Несколько раньше то же самое было сделано фирмой AMD – расширение 3DNow!, которое было реализовано уже в процессорах К6-2 с введением новых инструкций, оперирующих с 64-битными регистрами.

Данное направление получило развитие и в следующих поколениях процессоров корпораций Intel и AMD. Современные процессоры Intel поддерживают потоковые расширения SSE, SSE2, SSE3, SSSE3, SSE4.

**Анализ процессоров с матричной структурой**

Матричные вычислительные системы начиная с 60-х годов 20 столетия относятся к основным концепциям построения сверхмощных средств вычислительной техники. Матричный способ обработки информации, в отличие от конвейерного, в принципе позволяет осуществлять неограниченное количество вычислительных процессов, следовательно, достичь любого уровня быстродействия вычислительных средств.

Матричные ВС – вариант технической реализации модели коллектива вычислителей. В таких системах в высокой степени воплощены фундаментальные архитектурные принципы.

Параллельность выполнения операций в матричных ВС обеспечивается на нескольких функциональных уровнях. На макроуровне параллельность достигается за счет одновременной работы нескольких матричных процессоров. На микроуровне параллельность выражается в возможности одновременной работы большого количества элементарных процессоров.

В современных суперсистемах выделяются промежуточные функционально-структурные образования из элементарных процессоров (обусловленные, в частности, технологическими возможностями элементарной базы). Так, в моделях семейства СМ в качестве таких образований выступают процессорные кристаллы (вершины гиперкубической структуры). Эти кристаллы-вершины есть по сути матричные процессоры, в каждом из которых все ЭП (их 16 в СМ) взаимодействуют друг с другом через -мерную (двумерную в СМ-1) решетчатую структуру. На рассмотренном уровне параллелизм также поддерживается: осуществляется параллельная работа кристаллов-вершин и элементарных процессоров в пределах каждой из них.

В промышленных матричных ВС число параллельно выполняемых операций порядка ;

Программируемость структуры в матричных системах проявляется более сильно, чем в конвейерных. В самом деле, матричная ВС может быть так настроена, что ее различные квадранты или подсистемы будут одновременно решать различные задачи. Кроме того, в пределах квадранта или подсистемы имеется возможность программировать направление передачи информации от каждого ЭП и, следовательно, настраивать канал связи между любыми ЭП. В матричных ВС заложены средства программного управления состоянием каждого элементарного процессора. Последнее позволяет матрицу или подсистему ЭП разбивать на группы, каждая из которых может реализовать свой режим обработки данных. Следовательно, на различных группах ЭП можно выполнять различные программы. Однако в силу того, что в квадранте или подсистеме имеется только одно устройство управления, группы ЭП должны работать последовательно.

Итак, если в мультипрограммном режиме в однопроцессорной ЭВМ имеет место разделение времени, то в матричной ВС – разделение “пространства” элементарных процессоров и системного времени. При этом различные задачи решаются на различных группах ЭП (состоящих из различного числа процессоров) и в различные (последовательные) отрезки времени. При такой организации мультипрограммного режима резко падает производительность всей системы, так как вместо параллельных вычислений производятся последовательно-параллельные и в пределе последовательные.

Матричные ВС располагают функционально гибкой структурой сети межпроцессорных связей. В современных системах (80-х и 90-х годов) двумерные решетки используются для организации сети межпроцессорных связей только в пределах одного кристалла, причем число элементарных процессоров в нем соответствует текущему уровню технологии БИС. Более того при эволюционном развитии архитектуры ВС такие двумерные решетки трансформируются в более совершенные -мерные (например, это имело место при переходе от модели СМ-1 к СМ-2). Процессорные кристаллы берутся в качестве функционально-структурных элементов, а сама система формируется как композиция множества кристаллов и сети связей между ними. Технология БИС и техника конструирования систем уже позволяют формировать из таких кристаллов гиперкубы.

Итак, программируемые структуры сетей межпроцессорных связей позволяют матричной ВС адаптироваться под область применения и структуру решаемой задачи, они делают локальную оперативную память любого ЭП общедоступной для других ЭП.

Однородность состава и структуры ВС видна на всех функциональных уровнях. На макроуровне однородность выражена тем, что все матричные процессоры (или квадранты в ILLIAC-IV, или подсистемы в моделях СМ) и устройства управления, входящие в них, одинаковы. На микроуровне однородность достигнута за счет применения множества идентичных элементарных процессоров.

Сети межпроцессорных связей в матричных ВС – однородные, это либо решетки либо гиперкубы.

Однородность проявляется и в конструкции ВС, они формируются из конструктивно однотипных элементарных процессоров или процессорных кристаллов.

Архитектура матричных ВС не лишена существенных недостатков. Так, единственное устройство управления в квадранте или подсистеме резко снижает надежность и живучесть, а также производительность при мультипрограммной работе и, следовательно, ограничивает сферу применения матричных ВС. Эти недостатки и сравнительно небольшие экономические преимущества, полученные за счет общего устройства управления в матричном процессоре (в квадранте или подсистеме), заставляют разработчиков постепенно отходить от принятой ими архитектурной концепции и переходить к сформулированной в Сибирском отделении РАН модели коллектива вычислителей. Так, по сравнению с системой SOLOMON, в которой были общими и устройство управления и память, в ILLIAC-IV осталось общим только устройство управления. В дальнейшем очевиден следующий шаг – ввести для каждого элементарного процессора свое устройство управления. Это и будет означать полный переход к модели коллектива вычислителей.

Таким образом, матричные и вычислительные системы – существенный шаг в развитии архитектуры средств переработки информации. Матричные ВС позволяют достичь производительности порядка  опер/с при решении широкого класса трудоемких задач.

Я считаю, что примером использования SIMD архитектуры в современных ВС являются современные видеокарты т.к. матричные процессоры дают возможность добавления множества элементарных процессоров для высокого быстродействия, и высокой скорости вычисления однотипных данных. Например, видеокарта **Sapphire Radeon HD 5450** в ней каждое ядро SIMD состоит из 16 суперскалярных потоковых процессоров, каждый из которых в свою очередь содержит по два исполнительных ядра, один блок выполнения специализированных функций, единицу ветвления и блок регистров общего назначения. При такой архитектуре каждый движок SIMD содержит 80 скалярных 32-битных потоковых процессора.

# 2. ОТВЕТ НА ВТОРОЙ ВОПРОС

## 2.1. ЗАДАНИЕ

Произвести численный расчет и построить графики для функций надежности *r*(*t*) и готовности *s*(*i*, *t*) ЭВМ, обладающей следующими техническими параметрами:

– средним временем безотказной работы ϑ =108 ч,

– интенсивностью восстановления μ =100 1/*ч* .

## 2.2. ОТВЕТ

Функция надежности (или вероятность безотказной работы) относится к основным показателям надежности ЭВМ. Характеризует производительность ЭВМ на промежутке времени, то есть эта функция обеспечивает потенциально возможную производительность. Функцией надежности ЭВМ называется



где запись  означает вероятность того, что для всякого , принадлежащего промежутку времени  производительность  ЭВМ равна единице, т.е равна потенциально возможной.

Функция  обладает следующими свойствами:

1.  Т.е. машина в момент начала функционирования находится в работоспособном состоянии.
2.  Событие, заключающееся в том, что ЭВМ работоспособна на конечном промежутке времени, является достоверным.
3.  для ;

# *Функцией ненадежности (или вероятностью отказа) ЭВМ называется*



Функция  позволяет определить среднее время безотказной работы (средняя наработка до отказа). По определению, *среднее время*  *безотказной работы* ЭВМ и оценка  соответственно равны:



где – время безотказной работы -й машины, 

*Интенсивностью отказов* *(лямбда-характеристикой*) ЭВМ называется функция



Практически установлено, что зависимость интенсивности отказов от времени имеет место на периоде приработки ЭВМ. После приработки ЭВМ интенсивность отказов остается постоянной (до вхождения в предельное состояние или, по крайней мере, в течение промежутка времени, перекрывающего время морального старения). Следовательно, в нормальных условиях эксплуатации ЭВМ  а функция надежности и математическое ожидание времени *безотказной работы* соответственно равны:

.

– *среднее число отказов, появляющихся в машине в единицу времени.*

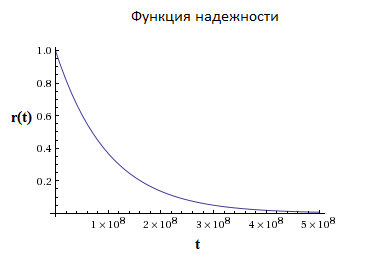
Подставляя известные нам данные получим следующую функцию для расчета надежности:

*r*(*t*)=exp(-*t*/)

*r*(*t*)=exp(-*t*/108);

Рассчитаем значения функции и построим график:

|  |  |
| --- | --- |
| *t,*ч. | *r*(*t*) |
| 0 | 1 |
| 100 | 0,999999 |
| 10000 | 0,999900 |
| 100000 | 0,999000 |
| 1000000 | 0,990049 |
| 10000000 | 0,984037 |
| 20000000 | 0,818730 |
| 30000000 | 0,740818 |
| 40000000 | 0,670320 |
| 50000000 | 0,606530 |
| 60000000 | 0,548811 |
| 70000000 | 0,496585 |
| 80000000 | 0,449328 |
| 90000000 | 0,406529 |
| 100000000 | 0,367879 |
| 200000000 | 0,135335 |
| 300000000 | 0,048797 |



Теперь рассчитаем значения функции готовности. Ф*ункция готовности ЭВМ*



 есть вероятность того, что (в условиях потока отказов и восстановлений) машина будет иметь в момент времени  производительность, равную единице, т.е. равную потенциально возможной.

Функция готовности ЭВМ обладает следующими свойствами:

1. 
2. 
3.   для 

Расчет будем производить по следующим формулам:

;

.

для начальных состояний ЭВМ , причем  соответствует состоянию отказа, а  – работоспособному состоянию машины, где *λ*=1/.

*s*(0, *t*) =100 / (100 + 1/108) – 100 / (100 + 1/108)\*exp((-*t)* \* (100 + 1/108))

= 0,99999999 – 0,99999999 \* exp((-t) \* 10,0000001);

*s*(1, *t*) =100 / (100 + 1/108) + 0,00000001 / (100 + 1/108) \* exp((-*t)* \* (100 + 1/108)) = 0,99999999 + 0,0000000099 \* exp((-t) \* 10,0000001)

Рассчитаем значения функции и построим график:

|  |  |
| --- | --- |
| *t*,ч. | *s*(0,*t*) |
| 0 | 0 |
| 0,001 | 0,00995 |
| 0,01 | 0,09516 |
| 0,08 | 0,55067 |
| 0,1 | 0,63212 |
| 0,2 | 0,86466 |
| 0,3 | 0,95021 |

|  |  |
| --- | --- |
| *t*,ч. | *s*(1,*t*) |
| 0 | 1 |
| 0,1 | 0,9999986 |
| 0,2 | 0,9999989 |
| 0,3 | 0,9999989 |

