Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

[09.03.01.Б1.Б.14 - АВС](https://eios.sibsutis.ru/course/view.php?id=41)

№ кода и наименование направления подготовки

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 14

Выполнил:

студент гр. ИА-431 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Линёв К.С./

подпись

Проверил:

доцент кафедры ВС

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

ОЦЕНКА, подпись

Новосибирск 2016

Оглавление

1. Ответ на первый вопрос………………………………………………….3
   1. Задание………………………………………………………………...3
   2. Ответ…………………………………………………………………..3

1.21 Сравнительный анализ…...…………………………………….3

1.22 Пример функциональной схемы...…..………………………...6

2. Ответ на второй вопрос…………………………………………………..7

2.1 Задание………………………………………………………………...7

2.2 Ответ…………………………………………………………………...7

2.21 Решение…………………………………………………………7

3. Список литературы…………………………………………………........11

# 

# 1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС

### 1.1. ЗАДАНИЕ

Выполнить сравнительный анализ вычислительных систем с архитектурами MISD и MIMD. Привести примеры функциональных структур промышленных ВС.

### 1.2. ОТВЕТ

**1.21 Сравнительный анализ вычислительных систем с архитектурами MISD и MIMD.**

В архитектурном плане выделяют четыре типа архитектур вычислительных средств: SISD, MISD, SIMD, MIMD. Архитектуры MISD и MIMD относятся к вычислительным системам.

Архитектура MISD (Multiple Instruction stream, Single Data stream) или МКОД (Множественный поток Команд, Одиночный поток Данных) позволяет нескольким потокам команд обрабатывать один поток данных.

Архитектура MIMD (Multiple Instruction stream, Multiple Data stream) или МКМД (Множественный поток Команд, Множественный поток Данных) позволяет нескольким потокам команд обрабатывать несколько потоков данных.

Из этого следует, что в этих архитектурах имеет место “множественность” потоков или команд, или данных. Множественность характеризуется количеством одновременно реализуемых потоков команд или данных.

MIMD-системы могут параллельно выполнять множество подзадач, чтобы сократить время выполнения основной задачи, что недоступно MISD-системам.

Архитектура MIMD дает большую гибкость: может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей.

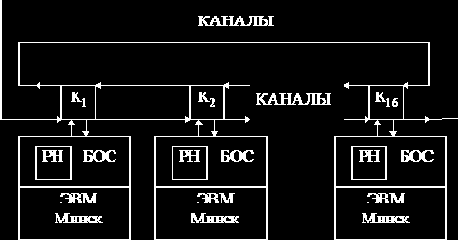
* MIMD архитектуры классифицируются в зависимости от физической организации памяти, то есть имеет ли процессор свою собственную локальную память и обращается к другим блокам памяти, используя коммутирующую сеть, или коммутирующая сеть подсоединяет все процессоры к общедоступной памяти. Исходя из организации памяти, различают следующие типы параллельных архитектур:
* **Компьютеры с распределенной памятью** (Distributed memory)  
  Процессор может обращаться к локальной памяти, может посылать и получать сообщения, передаваемые по сети, соединяющей процессоры. Сообщения используются для осуществления связи между процессорами или, что эквивалентно, для чтения и записи удаленных блоков памяти. В идеализированной сети стоимость посылки сообщения между двумя узлами сети не зависит как от расположения обоих узлов, так и от трафика сети, но зависит от длины сообщения.
* **Компьютеры с общей (разделяемой) памятью**(True shared memory)   
  Все процессоры совместно обращаются к общей памяти, обычно, через шину или иерархию шин. В идеализированной PRAM (Parallel Random Access Machine - параллельная машина с произвольным доступом) модели, часто используемой в теоретических исследованиях параллельных алгоритмов, любой процессор может обращаться к любой ячейке памяти за одно и то же время. На практике масштабируемость этой архитектуры обычно приводит к некоторой форме иерархии памяти. Частота обращений к общей памяти может быть уменьшена за счет сохранения копий часто используемых данных в кэш-памяти, связанной с каждым процессором. Доступ к этому кэш-памяти намного быстрее, чем непосредственно доступ к общей памяти.
* **Компьютеры с виртуальной общей (разделяемой) памятью** (Virtual shared memory)   
  Общая память как таковая отсутствует. Каждый процессор имеет собственную локальную память и может обращаться к локальной памяти других процессоров, используя "глобальный адрес". Если "глобальный адрес" указывает не на локальную память, то доступ к памяти реализуется с помощью сообщений, пересылаемых по коммуникационной сети.
* В MISD-системе реализуется принцип конвейерной (магистральной) обработки, который основан на разбиении всего процесса на последовательно выполняемые этапы, причем каждый этап выполняется на отдельном процессоре. Одинарный поток исходных данных для решения задачи поступает на вход процессорного конвейера. Каждый процессор решает свою часть задачи, и результаты решения в качестве исходных данных передает на вход последующего процессора.

В качестве **вывода** хотелось бы сказать, что и MIMD, и MISD архитектуры имеют свои преимущества и недостатки.

MIMD-системы характеризуются невысокой функциональной сложностью, дают больше возможностей, в системе легко осуществляется реконфигурация структуры путем добавления или удаления функциональных модулей, *но* при этом ограниченность производительности системы пропускной способностью общей шины и ухудшение общей производительности системы при ее расширении путем добавления модулей, а также отказ общей шины приводит к выходу из строя всей системы.

MISD-системы позволяют сравнивать результаты вычислений в целях обнаружения отказов, что гарантирует отказоустойчивость; они могут стать полезными для разработки новых концепций в теории и практике построения вычислительных систем, *но* при этом вычислительных машин такого класса практически нет и трудно привести пример их успешной реализации; также они дорогие и низко производительные по отношению к MIMD-системам и имеют более высокое время выполнения задач.

**1.22 Пример функциональной схемы промышленной ВС.**



***МК "Минск-222"***

**"Минск-222"** – это система с множественным потоком команд и данных, с низкой степенью связанности и однородной структурой. Межмашинные связи осуществляются с помощью каналов, причем эти связи линейные – между каждой парой соседних ЭВМ.

 Связь между машинами осуществляется через каналы и систему коммутации, образованную из коммутаторов (К), входящих в отдельные машины. Коммутаторы реализуют некоторый набор системных операций, которые позволяют осуществить обмен информацией между данной ЭВМ и ее ближайшим соседом. Все выполняемые системные операции координируются содержимым регистров настройки (РН). Коммутатор ЭВМ и блок системных операций (БОС) составляют элементарную машину. Системные операции позволяют изменять характер взаимодействия ЭВМ за счет изменения содержимого РН, осуществлять обмен информацией и другие действия. Системные устройства занимают примерно 1 % всего МК.

## 2. ОТВЕТ НА ВТОРОЙ ВОПРОС

## 2.1. ЗАДАНИЕ

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

* средним временем безотказной работы =105 ч.
* интенсивностью восстановления *µ*=10 1/ч.

## 2.2. ОТВЕТ

**2.21 Решение**

Функция (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(-1/\**t*)

*r*(*t*)=exp(-*t*/105);

Рассчитаем значения функции и построим график:

|  |  |
| --- | --- |
| *t,*ч. | *r*(*t*) |
| 0 | 1 |
| 1 | 0,999999 |
| 5 | 0,999950 |
| 10 | 0,999900 |
| 100 | 0,999000 |
| 1000 | 0,990050 |
| 10000 | 0,904837 |
| 20000 | 0,818731 |
| 30000 | 0,740818 |
| 40000 | 0,670320 |
| 50000 | 0,606531 |
| 60000 | 0,548812 |
| 70000 | 0,496585 |
| 80000 | 0,449329 |
| 90000 | 0,406569 |
| 100000 | 0,367879 |
| 150000 | 0,223130 |
| 200000 | 0,135335 |
| 250000 | 0,082085 |
| 300000 | 0,049787 |
| 350000 | 0,030197 |
| 400000 | 0,018316 |
| 500000 | 0,006738 |
| 1000000 | 0,000045 |

**Функция надежности r(t)**

0

0,2

0,4

0,6

0,8

1

1,2

0

5

100

10000

30000

50000

70000

90000

150000

250000

350000

500000

**t, ч**

**r(t)**

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*)=10 / (10 + 1/105) – 10 / (10 + 1/105)\*exp((-*t)* \* (10 + 1/105))=

=0,999999 – 0,999999 \* exp((-t) \* 10,00001);

*s* (1, *t*)=10 / (10 + 1/105) + 0,000001 / (10 + 1/105)\*exp((-*t)* \* (10 + 1/105))=

=0,999999 + 0,00000099 \* exp((-t) \* 10,00001).

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 |



3. СПИСОК ЛИТЕРАТУРЫ

1) Хорошевский В.Г. Архитектура вычислительных систем. – М.: МГТУ им. Н.Э. Баумана, 2008. – 315 с.

2) Введение в архитектуру компьютеров [электронный ресурс] // Вычислительные системы и многомашинные комплексы на базе однопроцессорных ЭВМ. URL: <http://www.safe-inform.ru/700b/index-vychislitelnye-sistemy-i-mnogomashinnye-kompleksy.htm> (дата обращения 10.12.16)

3) StudFiles [электронный ресурс] // Классификации архитектур ВС. URL:

<http://www.studfiles.ru/preview/1511376/> (дата обращения 12.12.16)