

Министерство образования Республики Беларусь

Учреждение образования  
«Полоцкий государственный университет»

О. Е. ШЕСТОПАЛОВА

## МОДЕЛИРОВАНИЕ

Учебно-методический комплекс  
для студентов специальностей

1-40 01 01 «Программное обеспечение информационных технологий»  
и 1-40 02 01 «Вычислительные машины, системы и сети»

Новополоцк  
ПГУ  
2009

УДК 004.9(075.8)  
ББК 32.844-02я73  
Ш51

Рекомендован к изданию методической комиссией факультета  
информационных технологий (протокол № 1 от 01.10.2008)  
в качестве учебно-методического комплекса

**РЕЦЕНЗЕНТЫ:**

заместитель начальника отдела АСУ ОАО «Нафтан» В. В. ДЮЖЕВ;  
зав. кафедрой информационных технологий С. В. КУХТА

**Шестопалова, О. Е.**

Ш51 **Моделирование** : учеб.-метод. комплекс для студентов спец.  
1-40 01 01 «Программное обеспечение информационных технологий» и  
1-40 02 01 «Вычислительные машины, системы и сети» / О. Е. Шесто-  
палова. – Новополоцк : ПГУ, 2009. – 304 с.  
ISBN 978-985-418-936-9.

Приведены введения в дисциплину для студентов дневной и заочной форм обучения, регламентирующие состав и содержание всех видов учебных активностей, методы и формы контроля. Содержит конспект лекций; индивидуальные задания, методические указания и примеры выполнения лабораторных работ; методические указания к курсовому проектированию; примеры аттестационных заданий, вопросы к экзамену, типы экзаменационных задач. Дан перечень и характеристика дополнительных электронных материалов, выдаваемых в комплекте с УМК.

Предназначен для студентов специальностей 1-40 01 01 «Программное обеспечение информационных технологий» и 1-40 02 01 «Вычислительные машины, системы и сети».

**УДК 004.9(075.8)**  
**ББК 32.844-02я73**

**ISBN 978-985-418-936-9**

© Шестопалова О. Е., 2009  
© УО «Полоцкий государственный  
университет», 2009

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ В КУРС «МОДЕЛИРОВАНИЕ».....	6
Дневная форма обучения .....	7
Заочная форма обучения.....	11
КОНСПЕКТ ЛЕКЦИЙ.....	14
1. Основы моделирования .....	15
1.1. Общие понятия. Классификация моделей.....	15
1.2. Режимы функционирования и задачи анализа технических объектов моделирования .....	20
2. Методология моделирования .....	25
2.1. Системный подход к моделированию .....	25
2.2. Иерархия вычислительных систем. Математический аппарат моделирования на различных уровнях декомпозиции .....	27
2.3. Моделирование структурных примитивов .....	30
2.3.1. Задача управления.....	31
2.3.2. Задача идентификации.....	33
2.4. Вероятностное моделирование. Метод Монте-Карло .....	40
3. Моделирование вычислительных сетей и систем .....	44
3.1. Автоматная концепция функционального моделирования .....	45
3.1.1. Абстрактные конечные автоматы.....	45
3.1.2. Сети Петри .....	49
3.2. Модели массового обслуживания .....	59
3.3. Аналитическое моделирование систем массового обслуживания .....	62
3.3.1. Обслуживание с ожиданием без потерь.....	65
3.3.2. Обслуживание с ограниченным временем ожидания.....	67
3.3.3. Обслуживание с ограниченным временем пребывания.....	69
3.3.4. Модели приоритетного обслуживания .....	71
3.4. Имитационное моделирование систем массового обслуживания .....	72
3.4.1. Элементы имитационных моделей систем массового обслуживания .....	73
3.4.2. Способы управления модельным временем.....	74
3.4.3. Алгоритмы имитационного моделирования для событийного управления модельным временем .....	75
3.4.4. Алгоритмы имитационного моделирования для пошагового управления модельным временем .....	77
ЛАБОРАТОРНЫЙ ПРАКТИКУМ .....	83
Лабораторная работа №1. Функциональный анализ цифровых схем с использованием дискретных детерминированных моделей .....	84
Теория .....	84
Ход выполнения работы .....	87
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	87
Варианты индивидуальных заданий .....	95
Лабораторная работа №2. Моделирование структурных примитивов. Решение задач управления и идентификации .....	97
Теория .....	98
Ход выполнения работы .....	100

Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	101
Варианты индивидуальных заданий .....	114
Лабораторная работа №3 Статистическое моделирование одноканальных СМО безприоритетного обслуживания однородного потока заявок .....	118
Теория .....	119
Ход выполнения работы .....	120
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	121
Варианты индивидуальных заданий .....	130
Лабораторная работа №4. Статистическое моделирование многоканальных СМО безприоритетного обслуживания неоднородного потока заявок с ограничением очередей обслуживания .....	133
Теория .....	133
Ход выполнения работы .....	133
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	134
Пример 1. Двухфазная одноканальная разомкнутая СМО с экспоненциальным распределением времени поступления и обслуживания .....	137
Пример 2. Трехканальная СМО с общей неограниченной очередью .....	139
Пример 3. Трехканальная СМО с независимыми очередями для обслуживания потока заявок трех типов .....	141
Пример 4. Двухканальная СМО с ограниченной очередью .....	143
Варианты индивидуальных заданий .....	145
Лабораторная работа №5. Статистическое моделирование многоканальных СМО приоритетного обслуживания неоднородного потока заявок с ограничением времени ожидания .....	156
Теория .....	156
Ход выполнения работы .....	156
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	157
Пример 1. Одноканальная СМО с ограничением времени ожидания обслуживания .....	161
Пример 2. Одноканальная СМО с абсолютным приоритетом обслуживания заявок определенного типа .....	163
Варианты индивидуальных заданий .....	165
Лабораторная работа №6. Статистическое моделирование отказов и ошибок обслуживания в многоканальных СМО с ограничением коэффициентов использования устройств обслуживания .....	169
Теория .....	169
Ход выполнения работы .....	169
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	170
Пример 1. Трехканальная СМО с распределением обслуживания в зависимости от коэффициентов использования устройств. ....	177
Пример 2. Двухканальная СМО с резервным каналом для обслуживания во время отказов основного канала .....	180

Пример 3. Двухканальная СМО с резервным каналом для обслуживания во время отказов доступа к ограниченной очереди основного канала.....	181
Пример 4. Одноканальная СМО с ошибками обслуживания .....	184
Варианты индивидуальных заданий.....	186
Лабораторная работа №7. Моделирование систем массового обслуживания сетями Петри.....	205
Ход выполнения работы.....	205
Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования .....	205
Пример 1. Сеть Петри для моделирования процессов отказов и устранения неисправностей .....	208
Пример 2. Сеть Петри для моделирования процесса пакетирования заявок с переменным размером пакета и параллельного обслуживания ...	213
Пример 3. Сеть Петри для моделирования магистрального канала передачи данных .....	215
Варианты индивидуальных заданий.....	216
КОНТРОЛЬ УСПЕВАЕМОСТИ.....	223
Первая аттестация.....	224
Вторая аттестация.....	226
Экзамен.....	228
Вопросы к экзамену.....	228
Типы экзаменационных задач .....	229
КУРСОВОЕ ПРОЕКТИРОВАНИЕ .....	233
Методические указания .....	234
Содержание пояснительной записки .....	234
Организация проектирования, защита и оценка.....	239
Основы моделирования в GPSS World.....	241
Объекты языка имитационного моделирования GPSS .....	242
Операторы модификации атрибутов и параметров заявок.....	253
Операторы изменения последовательности движения заявок.....	262
Операторы объектов аппаратной категории .....	265
Операторы получения статистических результатов.....	266
Операторы организации списка пользователя.....	268
Специальные операторы .....	276
ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ ПО КУРСУ.....	279
ЛИТЕРАТУРА.....	282
ПРИЛОЖЕНИЕ 1. Элементы стандартного отчета GPSS .....	284
ПРИЛОЖЕНИЕ 2. Основные операторы GPSS .....	289
ПРИЛОЖЕНИЕ 3. Стандартные числовые атрибуты GPSS.....	294
ПРИЛОЖЕНИЕ 4. Пакетирование заявок .....	298

# МОДЕЛИРОВАНИЕ

**В В е Д е Н и е**

**В К у р с**



✓ для студентов дневной формы обучения (+ правила контроля)  
✓ для студентов заочной формы обучения  
рейтинговой системы

## • структура курса

**объемы**



**сроки**



• лекционная часть

• лабораторный практикум

• курсовой проект

✓ возможность выбора сложности задания и базовой оценки

• аттестации

• экзамен

✓ возможность получения экзаменационного бонуса и оценки «автоматом»



**рейтинг**

## ВВЕДЕНИЕ В КУРС «МОДЕЛИРОВАНИЕ»

Целью изучения курса «Моделирование» является освоение общей методологии моделирования и приобретение практических навыков системного анализа, аналитического и имитационного моделирования информационных систем и вычислительных сетей с учетом их специфических особенностей как объектов моделирования.

### Дневная форма обучения

*Структура курса.* Курс «Моделирование», согласно учебным планам специальностей 1-40 01 01 «Программное обеспечение информационных технологий» и 1-40 02 01 «Вычислительные машины, системы и сети», читается на 3-м курсе обучения, в весеннем семестре и включает:

- 32 часа лекционных занятий;
- 32 часа лабораторного практикума в каждой подгруппе;
- курсовой проект.

Итоговой формой контроля по курсу является экзамен.

*Рейтинговая система контроля успеваемости.* Для оценивания аудиторной и самостоятельной работы в рамках курса используется рейтинговая система контроля успеваемости. Она является накопительной и предполагает суммирование балльных оценок и бонусов, выставяемых за все виды учебных активностей в течение прохождения курса. Итоговая сумма, набранная за время прохождения курса, является индивидуальным рейтингом студента (ИРС).

*Лекционная часть курса.* Лекции по курсу проводятся в традиционной форме. Основная часть теоретического материала по курсу приведена в *Конспекте лекций*. Помимо этого на лекциях будет рассматриваться дополнительный материал в виде пояснений и практических примеров. Ссылки на дополнительный материал в тексте *Конспекта лекций* оформлены специальным значком, например:

*ПРИМЕР* преобразования моделей технических объектов схематического вида к функционально-логическому.



Некоторые практические примеры подробно рассматриваются в лабораторном практикуме, и на них также имеются специальные ссылки в тексте *Конспекта лекций*, например:

*ПРИМЕРЫ* моделирования с использованием метода Монте-Карло других неравномерных распределений – см. *Лабораторную работу №4*



Промежуточный контроль освоения лекционной части курса проводится в виде теста, дважды в течение семестра, на аттестационных неделях. Тест состоит из 10 – 15 вопросов по пройденному материалу с ответами выборочного типа в каждом вопросе. Правильный ответ оценивается в 10 баллов рейтинга. Результат тестирования, составляющий *80 и более процентов от максимального результата тестирования* на потоке, поощряется бонусом в 50 баллов рейтинга. Кроме тестовых вопросов в аттестацию могут входить практические задания по пройденным на лабораторном практикуме методам моделирования.

На лекциях, посвященных конкретным методам моделирования, для индивидуального или группового выполнения могут быть предложены практические задания, правильное выполнение которых оценивается определенным числом баллов рейтинга. Оригинальные и первоочередные решения поощряются бонусами.

*Лабораторный практикум.* Целью лабораторного практикума является освоение практических навыков разработки и реализации моделей систем различной природы, а также анализа функционирования и определение характеристик систем на основании модельных экспериментов. По каждой лабораторной работе выдается индивидуальное задание.

Отчет по работе представляется в электронном виде. В отчет включается: название работы, вариант задания, анализ задания, основные и промежуточные результаты по каждому пункту хода выполнения работы, выводы по работе. Защита работ проводится индивидуально и оценивается в соответствии со следующими правилами:

- работа, защищенная непосредственно на занятии, отведенном на ее выполнение, оценивается в 150 баллов рейтинга (в этом случае оформление отчета не обязательно, достаточно представление основных и промежуточных результатов);

- работа, защищенная на занятии, следующем за занятием, отведенным на выполнение работы, оценивается в 100 баллов рейтинга;

- работа, защищенная не в срок, оценивается в 75 баллов рейтинга.

Набор 750 баллов рейтинга по лабораторному практикуму поощряется дополнительным бонусом в 100 баллов рейтинга. Суммарный рейтинг по лабораторному практикуму, равный 375 баллам позволяет претендовать только на итоговую экзаменационную оценку «5 баллов» и являются нижней границей допуска к экзамену.

Защита лабораторных работ проводится исключительно на занятиях по лабораторному практикуму в соответствии с расписанием. Защиты лабораторных работ принимаются только в течение семестра, до календар-

ной даты начала сессии (за исключением случаев недобора 375 баллов рейтинга по лабораторному практикуму). В случае если на лабораторном занятии возникает очередь из желающих защищать лабораторные работы, студенты, имеющие бóльший суммарный рейтинг по лабораторному практикуму пользуются преимущественным правом первоочередной защиты.

*Курсовое проектирование.* Целью курсового проекта является освоение практических навыков моделирования вычислительных и технологических систем массового обслуживания. Задание на курсовое проектирование разделено на три части по категориям сложности:

1. Имитационное моделирование системы массового обслуживания. Программное обеспечение моделирования: среда *GPSS World*;

2. Анализ и оптимизация системы массового обслуживания. Программное обеспечение моделирования: среда *GPSS World*;

3. Функциональное моделирование системы массового обслуживания сетью Петри. Программное обеспечение моделирования: любой графический редактор и/или ПО моделирования сетей Петри.

Первая категория сложности соответствует базовой оценке «6,5» и предполагает выполнение только части 1 задания на курсовое проектирование. Вторая категория сложности соответствует базовой оценке «8,5» и предполагает выполнение только частей 1 и 2 задания. Полная категория сложности соответствует базовой оценке «9,5» и предполагает выполнение всего задания целиком, т. е. частей 1, 2 и 3.

Выбор базовой оценки означает, что курсовой проект оценивается исходя из указанной оценки, что, разумеется, не гарантирует ее получения без соответствующего уровня работы.

Задание на курсовое проектирование в электронном виде выдается на первой лекции по курсу, а в твердой копии – не позднее 4-ой недели семестра. В твердой копии задания, кроме даты завершения работ над проектом, указывается дата и время защиты курсового проекта. В качестве даты защиты может быть установлена любая дата не ранее срока завершения работ над проектом, указанного в задании, и не позднее календарного начала сессии. Неявка на защиту в срок автоматически влечет за собой выставление «не явился» в ведомости по курсовому проектированию, недопуск к сессии и возможность сдачи проекта только по направлению деканата после календарного начала сессии.

Защита курсового досрочно (раньше даты завершения работ над проектом, указанной в задании) поощряется бонусом в 150 баллов рейтинга. Сумма баллов рейтинга, выставляемая за курсовое проектирование на момент проведения аттестаций, соответствует процентной доле выполненных

работ от общего объема работ по проекту. Курсовой проект представляется к защите в твердой копии и в электронном виде. Перечень материалов по каждой форме представления, обязательных к сдаче, указан в задании на курсовое проектирование.

*Аттестации (промежуточный контроль успеваемости).* Для положительной аттестации ИРС по всем учебным активностям на момент аттестации должен составлять не менее, чем  $2/3$  от среднего ИРС на потоке. Аттестационная ведомость (список потока с суммарным рейтингом каждого студента) выдается старостам групп.

*Экзамен (итоговый контроль успеваемости).* Экзамен представляет собой письменную работу, на выполнение которой отводится 2 часа. Билет включает в себя:

- один теоретический вопрос, оцениваемый из 100 баллов;
- тестовую часть, состоящую из 4-х вопросов выборочного типа, каждый из которых оценивается из 10 баллов;
- задачу, оцениваемую из 60-ти баллов.

Стандартные десятибалльные оценки определяются по результатам экзаменационной работы как показано в таблице 1.

Таблица 1

<b>Сумма баллов за экзаменационную работу</b>	0...25	26...45	46...74	75...85	86...105	106...125	126...145	146...165	166...189	190...200
<b>Оценка</b>	1	2	3	4	5	6	7	8	9	10

Экзаменационная оценка выставляется на основании письменной работы, дополнительные вопросы и обсуждения не практикуются. Работа выполняется на специальном бланке. Бланк и черновики выдаются преподавателем. При себе на экзамене следует иметь ручку, карандаш, стёрку, а также зачетную книжку. Пользование конспектом и консультации друг с другом запрещены. Экзамен начинается в 9.00. Время опоздания вычитается из времени выполнения работы. В течение первых 20-ти минут работы разрешаются консультации с преподавателем по формулировкам вопросов и задачи. По истечении 20-ти минут консультации прекращаются. Выходить во время работы запрещается.

К экзамену разрешается подготовить страницу справочного материала единого для группы образца, на которую вынести все формулы и схемы, запоминание которых представляет трудности, но без комментариев, обозначений и подписей. Допустимый объем материала – одна сторона фор-

мата А4, шрифт – не менее 10 pt. Материал утверждается и разрешается к использованию преподавателем на консультации к экзамену.

Студенты, имеющие суммарный рейтинг по результатам семестра *равный или больший 80% от максимального* на потоке, имеют право на два дополнительных балла к экзаменационной оценке (за исключением неудовлетворительных) или на автоматическое засчитывание максимально возможного результата за любую часть экзаменационного билета (теоретическую, тестовую или задачу). Выбор экзаменационного бонуса осуществляется студентом на экзамене, после выдачи билета.

Студенты, имеющие суммарный рейтинг, *на 80 или более процентов больше, чем средний* на потоке, получают 10 баллов за экзамен автоматически. Студенты, занимающие в конце семестра *три верхние строчки итогового рейтинга*, получают соответственно 9, 8 и 7 баллов за экзамен автоматически, если их результат не обеспечивает 10 баллов автоматом, и если они не претендуют на более высокую оценку – в этом случае они могут сдавать экзамен на общих основаниях.

### **Заочная форма обучения**

*Структура курса.* Курс «Моделирование», согласно учебному плану по специальности, читается на 4-м курсе заочной формы обучения по специальностям 1-40 01 01 «Программное обеспечение информационных технологий» и 1-40 02 01 «Вычислительные машины, системы и сети». Курс включает в себя:

- установочную лекцию;
- 8 часов лекционных занятий;
- 8 часов лабораторного практикума в каждой подгруппе;
- курсовой проект.

Итоговой формой контроля по курсу является экзамен.

*Лекционная часть курса.* Основная обязательная часть теоретического материала по курсу приведена в *Конспекте лекций*. Целью лекционных занятий является реферативное представление теоретического материала по курсу с рекомендациями по самостоятельной работе с конспектом лекций и дополнительными материалами.

*Лабораторный практикум.* Обязательному выполнению и защите подлежат лабораторные работы №1 и №2. Методы моделирования, используемые в остальных работах, подлежат самостоятельному освоению по методическим указаниям лабораторного практикума и используются в курсовом проектировании. Работы №1 и №2 выполняются по индивиду-

альному варианту – номер Вашего варианта соответствует номеру варианта задания на курсовой проект.

Если работа выполняется и защищается в срок, т. е. во время сессии на занятиях по лабораторному практикуму, то отчет по работе представляется в электронном виде. В отчет включается: название работы, вариант задания, анализ задания, основные и промежуточные результаты по каждому пункту хода выполнения работы, выводы по работе. В процессе защиты проверяется самостоятельность выполнения работы, навыки по использованию программного обеспечения моделирования, умение изменять характеристики разработанных моделей и режимов реализации модельных экспериментов по заданию преподавателя.

Если лабораторные работы не выполнены или не защищены в срок, т. е. на занятиях по лабораторному практикуму, то отчеты по ним предоставляются для проверки в распечатанном и сброшюрованном виде на кафедру вычислительных систем и сетей. К отчету прикладывается электронный носитель с реализациями моделей в программном обеспечении моделирования. Решение о времени и месте защиты принимаются преподавателем после проверки.

Рекомендуется выполнять лабораторные работы заранее, самостоятельно, присылать на проверку по электронной почте и защищать досрочно по договоренности с преподавателем.

Выполнение и защита лабораторных работ является обязательным условием допуска к экзамену по курсу «Моделирование».

*Курсовое проектирование.* Цели курсового проектирования и структурирование задания по категориям сложности – см. в разделе *Дневная форма обучения*.

Задание на курсовое проектирование выдается на установочной лекции. Курсовые проекты предоставляются для проверки на кафедру вычислительных систем и сетей не позднее календарной даты начала сессии. Перечень материалов курсового проектирования, предоставляемых распечатанными и сброшюрованными, и перечень материалов курсового проектирования, предоставляемых в электронном виде, приведены в задании на курсовое проектирование. Решение о времени и месте защиты принимаются преподавателем после проверки.

Защита проектов проводится индивидуально. В процессе защиты проверяется самостоятельность выполнения проекта, навыки по использованию программного обеспечения моделирования, умение изменять характеристики разработанных моделей и режимов реализации модельных экспериментов по заданию преподавателя.

*Экзамен (итоговый контроль успеваемости).* Экзамен представляет собой самостоятельную письменную работу. Билет разделен на две части по категориям сложности и включает в себя:

- 1) часть I – теоретический вопрос, оцениваемый из 100 баллов;
- 2) часть II – тест, состоящий из 4-х вопросов выборочного типа (каждый из которых оценивается из 10 баллов) и задача (оценивается в 60 баллов).

Если Вы не претендуете на оценку выше, чем «5 баллов», то на экзамене можете выполнять только часть I экзаменационной работы. Теоретические вопросы, включённые в часть I, относятся к основной обязательной части теоретического материала по курсу, и в перечне экзаменационных вопросов курса отмечены *звездочкой*.

Полная категория сложности позволяет претендовать на любую оценку свыше 5 баллов и предполагает выполнение частей I и II билета.

Правила организации и проведения экзамена, а также таблицу перевода экзаменационных баллов в оценки десятибалльной системы – см. в разделе *Дневная форма обучения*.

# МОДЕЛИРОВАНИЕ

## К О Н С П Е К Т

## Л Е К Ц И Й



### 1. Основы моделирования

- общие понятия, классификация моделей
- режимы функционирования и задачи анализа технических объектов моделирования

### 2. Методология моделирования

- системный подход
- иерархия ВС и мат. аппарат разных уровней декомпозиции
- моделирование структурных примитивов
- вероятностное моделирование, метод Монте-Карло

### 3. Моделирование ВС и систем

- автоматная концепция ФМ
- модели массового обслуживания
- аналитическое моделирование СМО
- имитационное моделирование СМО



- ✓ ссылки на дополнительные примеры
- ✓ ссылки на примеры в лабораторной практике

# КОНСПЕКТ ЛЕКЦИЙ

## 1. Основы моделирования

### 1.1. Общие понятия. Классификация моделей

*Моделью* реального объекта называют его представление в некоторой форме, отличной от реального воплощения. Для естественных материальных объектов модель вторична, т. е. появляется как следствие изучения и описания этого объекта (например, модель солнечной системы). Для объектов, создаваемых человеком или техникой модель первична, так как предшествует появлению самого объекта (например, модель самолета, модель триггера).

Степень соответствия модели реальному объекту определяется разработчиком. Для одного объекта можно разработать целый ряд моделей, отражающих его поведение или свойства с разных сторон или с разной степенью детальности. В идеале модель должна позволять предсказывать неизвестные состояния моделируемого объекта – возможность интерполяции и прогнозирования важные качества модели.

*ПРИМЕР* неоднозначности интерпретации результатов натурального эксперимента с точки зрения интерполяции и прогнозирования.



*Моделирование* есть процесс создания модели реального объекта и постановка экспериментов на этой модели для исследования и оптимизации характеристик объекта в соответствии с заданными ограничениями.

Процесс моделирования есть процесс перехода из реальной области в виртуальную (модельную) посредством формализации, далее происходит собственно моделирование и, наконец, интерпретация результатов как обратный переход из виртуальной области в реальную. Этот путь заменяет прямое натурное исследование реального объекта. Итак, в самом простом случае технология моделирования включает 3 этапа: *формализацию*, собственно *моделирование* и *интерпретацию* результатов. Если требуется уточнение, эти этапы повторяются в цикле.

*ПРИМЕР* – схема процесса моделирования с детализацией этапов.



Объектами технического моделирования являются технические процессы и системы. Для инженеров специальностей 1-40 01 01 «Программное обеспечение информационных технологий» и 1-40 02 01 «Вычислительные машины, системы и сети» объект моделирования это чаще всего информационная система или вычислительная сеть, представляемая как множество взаимосвязанных реальных элементов, объединенных общей целью функционирования.

Целями моделирования технических процессов и систем являются:

1) получение средств решения задач синтеза и анализа при проектировании объекта моделирования;

2) исследование и анализ поведения реально существующего объекта в режимах и условиях, реализовать которые в натурном эксперименте невозможно либо нецелесообразно;

3) получение средств решения задач управления реальным объектом.

Рассмотрим классификацию моделей, в основу которой положено различие моделей по критериям подобия модели и объекта, а также соотношение точности и абстрактности моделей (см. рис. 1).

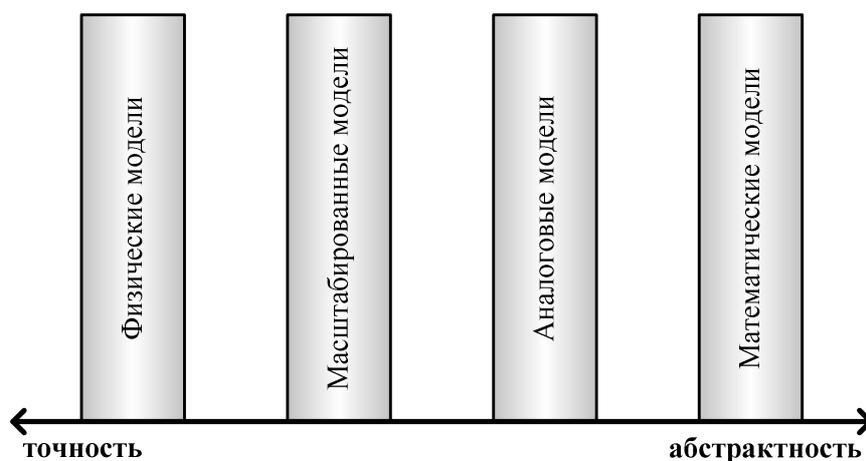


Рис. 1

Модели, находящиеся в начале спектра (см. рис. 1) называют *физическими* или *натурными*. Критерием подобия физического моделирования является общая природа взаимодействий в модели и в реальном объекте. При создании *масштабированных* моделей критерием подобия является математический критерий, представляющий собой безразмерную величину – соотношение определенных параметров процесса или системы. Сочетание в реальных объектах моделирования различных по своей природе процессов (например, физических и химических) затрудняет выбор общего критерия подобия и в пределе делает масштабирование невозможным.

*Аналоговыми* являются модели, в которых свойство реального объекта представляется некоторым другим свойством аналогичного по поведению объекта. Критерием подобия является аналогия поведения или свойств. Примером аналоговой модели является любой график: величины длин отрезков, отложенных по координатным осям, отображают взаимосвязанное изменение определенных характеристик объекта. Аналоговыми моделями также являются различного рода схемы. Критерием подобия в *математических* моделях является подобность математических уравнений.

*ПРИМЕР систем различной природы, подобных с математической точки зрения.*



В общем случае, уравнения математической модели связывают величины, которые характеризуют состояние системы или объекта, с входными и внутренними переменными. Величины, характеризующие состояние объекта моделирования в процессе его функционирования, называются *фазовыми переменными (фазовыми координатами)*. Вектор фазовых переменных  $\vec{V}_\phi = (\overrightarrow{v_1, v_2 \dots v_n})$  задает точку в пространстве, называемом фазовым пространством. Фазовое пространство, в отличие от геометрического, многомерно. Его размерность определяется количеством используемых фазовых координат.

*ПРИМЕР представления векторов фазовых координат в фазовых пространствах различной размерности.*



Обычно в уравнениях модели фигурируют не все фазовые переменные, а только часть из них, достаточная для однозначной идентификации состояния объекта моделирования. Такие фазовые переменные называют *базисными координатами*:  $\vec{V} = (\overrightarrow{v_i, v_l \dots v_k}) \subset \vec{V}_\phi$ . Через базисные координаты могут быть вычислены значения всех остальных фазовых переменных.

Математические модели делятся на *аналитические* и *имитационные*. В свою очередь, аналитические модели делятся на *теоретические* и *экспериментальные* (см. рис. 2).

Имитационные модели – это модели типа «черного ящика», которые рассматривают поведение объекта моделирования только с точки зрения входных воздействий и выходных параметров (вектора базисных координат). Они представляет собой алгоритм, которым описывается поведение объекта моделирования или способ вычисления вектора его базисных ко-

ординат при заданных входных воздействиях. Имитационные модели не способны формировать решение в том виде, который имеет место в аналитических моделях, а служат средством анализа поведения системы в условиях, определяемых экспериментатором.

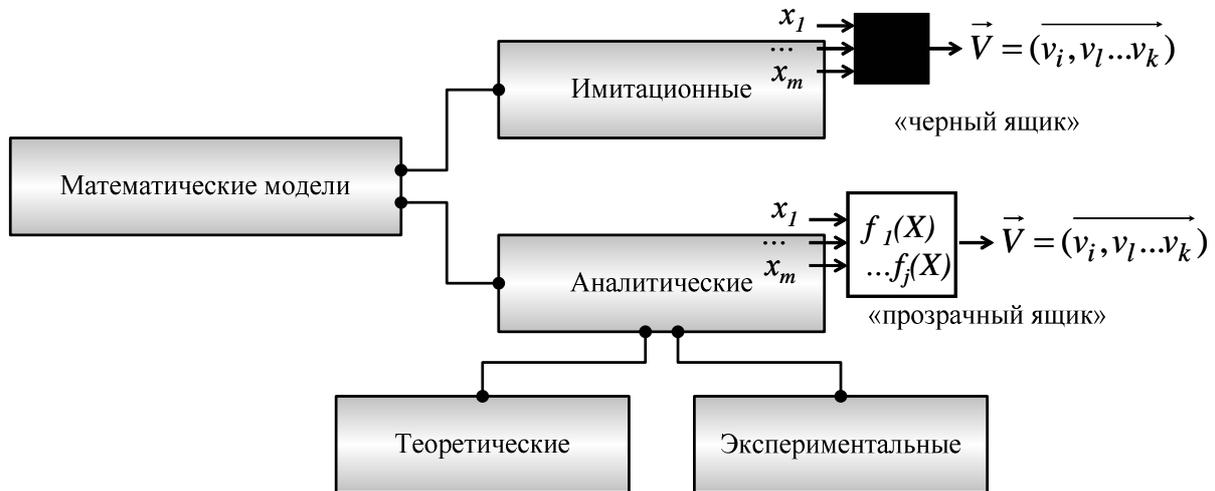


Рис. 2

Применение имитационного моделирования целесообразно в следующих условиях:

- 1) аналитической модели исследуемой системы не существует, не разработаны методы ее решения либо существующие методы не алгоритмируются и не формализуются;
- 2) аналитическая модель сложна и трудоемка, и имитационное моделирование дает более простой способ решения задачи;
- 3) кроме оценки определенных параметров необходимо осуществить наблюдение за ходом процессов в системе в течение определенного интервала времени – требуется сжатие временной шкалы.

Теоретические модели получают на основе известных теоретических описаний процессов функционирования объекта. Экспериментальные – на основе изучения поведения объекта моделирования во внешней среде. Экспериментальные модели являются формальными. Они не учитывают всего комплекса свойств моделируемого объекта, а лишь устанавливают обнаруживаемую в процессе эксперимента связь между отдельными параметрами объекта моделирования, которые удастся варьировать и/или осуществлять их измерение. Варьируемые параметры называют *факторами*. Такие модели дают адекватное описание исследуемых процессов только в ограниченной области факторного пространства, в которой осуществля-

лось варьирование факторов в эксперименте, т. е., как правило, не обладают свойством экстраполяции. Поэтому экспериментальные математические модели носят частный характер, в то время как теоретические отражают общие универсальные закономерности. Вместе с тем, применяемые для построения экспериментальных моделей методы (регрессионный и корреляционный анализ, планирования эксперимента) широко используются при проверке гипотез.

При моделировании любого объекта могут быть использованы разные модели или их совокупности из числа разновидностей, рассмотренных выше. Представления объекта моделью в рамках одной разновидности могут различаться по сложности и детализации.

*ПРИМЕРЫ различных моделей логического элемента И-НЕ.*



Еще одним важным классификационным признаком моделей является учет свойств реальных взаимодействий в объекте. По этому признаку модели делятся на *детерминированные* и *вероятностные*.

При разработке детерминированных моделей предполагается, что свойства входных воздействий и объекта моделирования неизменны во времени, т. е. предполагается, что соответствия между внешним воздействием на объект и его реакцией на это воздействие взаимно однозначны. Детерминированные модели отражают поведение объекта моделирования с позиций полной определенности в настоящем и будущем, они просты и экономичны, но, как правило, недостаточно точны. Эти модели применяются на начальных стадиях проектирования и анализа систем, для приближенной проверки гипотез и предварительных оценок.

Простота детерминированных моделей достигается за счет того, что в них игнорируются или моделируются весьма примитивно многие свойства, присущие реальным объектам (например, задержка и нагрузочная способность логических элементов). В них не учитываются и случайные факторы, такие как технологический разброс параметров, температурные и временные изменения, а также случайные характеристики процессов, таких как поступление заявок на обслуживание или изменение количества пользователей сети.

Вероятностные модели учитывают случайные факторы и, следовательно, оценивает изменение состояния объекта моделирования с позиций вероятности тех или иных событий. Случайные значения параметров модели обычно задаются с помощью генераторов псевдослучайных чисел по

заданному закону распределения. Один «прогон» модели дает одну реализацию случайного процесса. Поэтому для получения достоверных оценок требуется представительная выборка, т. е. большое число экспериментов на модели. Процесс моделирования с использованием вероятностных моделей называется *статистическим моделированием* и является разновидностью имитационного моделирования. Достоинство вероятностных моделей – высокая точность и надежность получаемых результатов. Недостаток – значительные затраты машинного времени на проведение статистического моделирования.

Детерминированное моделирование широко используется на начальных стадиях исследования, проектирования или решения задач управления техническими системами. Заключительные стадии выполняют на вероятностных моделях, когда уже приняты решения относительно структуры объекта и определены ориентировочно (на детерминированных моделях) значения его параметров.

Модель должна удовлетворять требованию *точности*. Для оценки точности используются различные статистические критерии, например, среднеквадратическое отклонение результатов расчетов на модели от реальных значений вектора базисных координат, полученных в результате эксперимента:

$$J = \frac{\sum_{i=1}^n (\vec{V}_{\text{реальное}_i} - \vec{V}_{\text{экспериментальное}_i})^2}{n}.$$

Если результат проверки не удовлетворяет требованию  $J \leq J_{\text{допустимое}}$ , то модель чрезмерно упрощена или исходная гипотеза неверна, и следует выбрать другую модель [1-4].

## **1.2. Режимы функционирования и задачи анализа технических объектов моделирования**

В зависимости от характера внешних воздействий технический объект (ТО) моделирования может находиться в установившемся или неустановившемся состоянии. Изменение его состояния выявляется анализом поведения фазовых переменных.

*Установившееся* состояние технического объекта достигается при неизменных характеристиках внешних воздействий. Режим функционирования ТО в установившемся состоянии называют *статическим* или *равновесным*. Статичность состояния определяется неизменностью базисных

координат всех элементов технической системы при постоянных внешних воздействиях. Если внешние воздействия изменяются, то состояние ТО будет *неустановившимся*. Режим работы ТО при этом называют *динамическим*. Он сопровождается непрерывным изменением фазовых координат.

Если при моделировании учитываются инерционные свойства объекта моделирования и изменение во времени параметров объекта или внешней среды, то модель называют *динамической*. В противном случае модель *статическая*.

Математическое представление динамической модели в общем случае может быть выражено системой дифференциальных уравнений, а статической – системой алгебраических уравнений. Динамическая модель может также представлять собой интегральные уравнения, передаточные функции, а в аналитической форме – явные зависимости фазовых координат или выходных параметров технического объекта от времени.

В модельном эксперименте детерминированного анализа поведения ТО обычно задают некоторые стандартные типовые воздействия на объект: ступенчатые, импульсные, гармонические, кусочно-линейные, экспоненциальные и др. Их называют *тестовыми воздействиями*. Предположим, что на технический объект, находящийся в установившемся состоянии, в некоторый момент времени  $t_0$  действует ступенчатое воздействие вида:

$$x(t) = \begin{cases} x_0, t \geq t_0, \\ 0, t < t_0, \end{cases}$$

где  $x_0 = const$  – модуль ступенчатого воздействия. Изменение состояния ТО будет определяться его внутренними свойствами и внешним воздействием. Пусть состояние ТО характеризуется фазовой координатой  $v(t)$ . Изменение ее после приложения ступенчатого воздействия можно представить в виде суммы двух составляющих: *переходной*  $v_n(t)$  и *вынужденной*  $v_g(t)$ . Переходная составляющая устойчивого ТО с течением времени затухает (стремится к нулю) и ТО приходит в новое установившееся состояние равновесия, характеризуемое новым постоянным значением вынужденной составляющей  $v_g(t) = v_k = const$ .

*ПРИМЕРЫ характеристик технического объекта при подаче на вход ступенчатого тестового воздействия.*



Следовательно, при приложении ступенчатого воздействия ТО осуществляет переход из одного установившегося состояния в другое, находясь при этом в течение некоторого времени в динамическом режиме (см. рис. 3-а).

Такой динамический режим называют *переходным процессом*, а графики изменения фазовых координат ТО – *переходными характеристиками*.

Если внешние воздействия на ТО переменны во времени, то они вызывают в нем непрерывный ряд переходных процессов и состояние системы в течение всего времени наблюдения будет неустановившимся. Переходные процессы возникают также при изменении структуры или параметров ТО в процессе ее функционирования.

Если внешнее воздействие  $x(t)$  – периодическая функция, для которой  $x_0=A \cdot \sin(\omega_0 t + \varphi)$ , то после затухания свободных колебаний (переходной составляющей) в устойчивом ТО установятся вынужденные колебания с частотой  $\omega_0$  и некоторыми постоянными амплитудами  $A_{vi}=k_i A$ , где  $k_i$  – постоянный коэффициент,  $i$  – номер фазовой координаты системы. Такое состояние ТО также относится к установившемуся, а режим называют *стационарным режимом колебаний* (см. рис. 3-б).

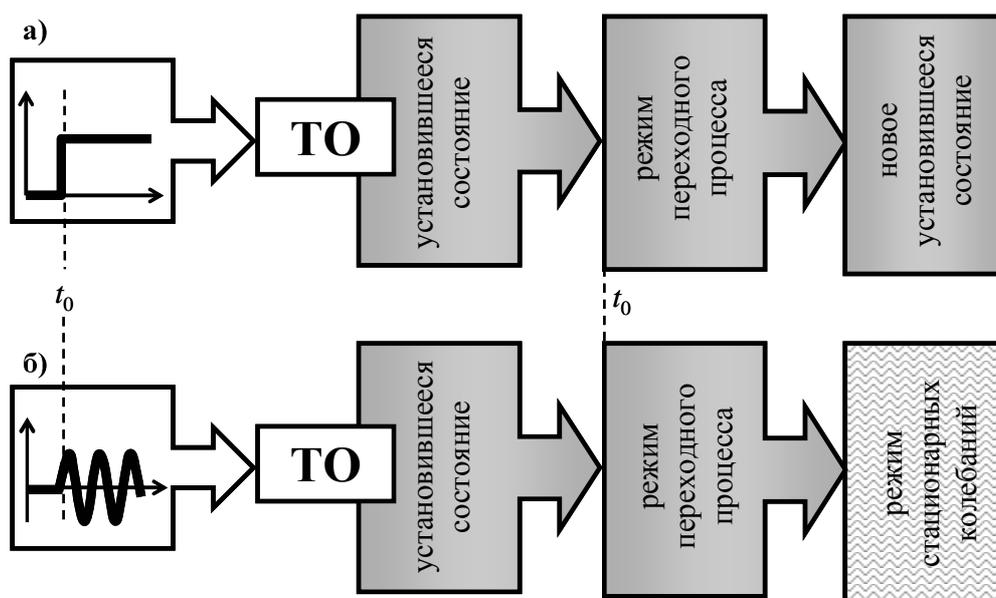


Рис. 3

*ПРИМЕР характеристик технического объекта при подаче на вход гармонического тестового воздействия.*



Рассмотренные тестовые входные воздействия – идеальная ступенька и гармонический сигнал с постоянными параметрами – являются модельными и при эксплуатации реальных ТО в чистом виде практически не встречаются. Однако модельное исследование поведения ТО при подаче на вход этих воздействий позволяют значительно облегчить и ускорить

решение проектных задач, так как детерминированные модели гораздо проще вероятностных. Получаемая при этом информация об объекте хотя и не претендует на полноту, но оказывается практически полезной.

Реальные внешние воздействия среды на технический объект описываются случайными функциями, а изменения фазовых координат системы представляют собой случайные процессы. Техническая система в этом случае все время находится в динамическом режиме.

Рассмотрим основные *задачи анализа*, решаемые при проектировании технических систем. В зависимости от модельного режима, положенного в основу решения конкретной проектной задачи, различают следующие виды анализа:

- статических состояний;
- переходных процессов;
- устойчивости;
- стационарных режимов колебаний;
- частотных характеристик;
- чувствительности;
- статистический.

Анализ статических состояний относится к задачам *статики*, а остальные виды анализа – к задачам *динамики*. Исходная математическая модель объекта представляет собой систему нелинейных дифференциальных уравнений, которая в нормальной форме Коши имеет вид:

$$\frac{d\vec{V}}{dt} = F(\vec{V}, t),$$

где  $V$  – вектор базисных координат;  $t$  – независимая переменная (время).

Система уравнений описывает динамические режимы функционирования технического объекта. Анализ этих режимов заключается в решении системы уравнений и последующем определении выходных параметров объекта. Задавая начальные условия  $V(0)=V_0$ , находят решения  $V(t)$ , а затем вычисляют значения *выходных параметров*  $Y$  (целевых функций). Большинство выходных параметров имеют смысл функционалов зависимостей базисных координат. Функционал представляет собой отображение класса функций в класс чисел. Примеры функционалов: определенные интегралы, экстремумы функций, значения функций при заданных значениях аргументов и т. п. Например, коэффициент использования (утилизации) сервера является выходным параметром, значение которого в первую очередь определяется интенсивностью поступления запросов (входная переменная) и ин-

тенсивностью их обработки (базисная переменная), а также рядом других факторов – дисциплинами обслуживания, приоритетами запросов и т. д.

Математическая модель рассмотренного вида непосредственно используется при анализе переходных процессов, устойчивости, стационарных режимов колебаний. Эта же модель позволяет решать и задачи анализа статических состояний: при  $dV/dt=0$  система дифференциальных уравнений переходит в систему алгебраических уравнений  $F(V)=0$ .

Частотный анализ проводится для определения резонансных режимов, для исследования передачи или преобразования информационных сигналов, представленных в частотной области. Если математическая модель линейна, используют преобразование Фурье и система дифференциальных уравнений преобразуется в систему линейных алгебраических уравнений с комплексными переменными, которая затем используется для определения частотных характеристик объекта. Процедура преобразования дифференциальных уравнений в алгебраические называется алгебраизацией исходной дифференциальной задачи, но полученная при этом модель, тем не менее, описывает динамические свойства объекта. Частотными методами можно также решать задачи анализа устойчивости и стационарных режимов колебаний.

Анализ чувствительности выполняется для оценки влияния вариации параметров объекта на изменение выходных функции. Выходные параметры объекта  $Y$  непосредственно не фигурируют в исходной системе ДУ. Они определяются по результатам решения  $V(t)$  системы уравнений.

Сложный технический объект обычно имеет множество внутренних параметров. Решение задачи оптимизации в этом случае вызывает значительные трудности. Вместе с тем не все параметры эффективно изменяют целевую функцию. Поэтому целесообразно их классифицировать и отобрать для оптимизации лишь те параметры, которые оказывают наибольшее влияние на целевую функцию. Эта задача решается методами *факторного анализа*.

Статистические методы анализа требуют использования вероятностных моделей. Большинство показателей качества, характеризующие процесс функционирования информационных систем и вычислительных сетей, являются вероятностными (например, вероятность надежного представления информации при выполнении функциональной задачи, вероятность предотвращения несанкционированного доступа и т. п.) [1, 5-7].

## 2. Методология моделирования

### 2.1. Системный подход к моделированию

Основой современной методологии моделирования являются принципы *системного подхода*. Он является альтернативой детерминистскому (Ньютоновскому) подходу, согласно которому поведение объекта моделирования описывается некоторой функцией и сопровождается определенным набором параметров. При этом предполагается, что внешняя среда не оказывает влияния на работу объекта. Это также означает отсутствие взаимного влияния таких объектов друг на друга, т. е. предполагается, что во взаимодействии они будут работать точно так же, как и в изоляции. Такой подход значительно упрощает изучение технических объектов, но приводит к необходимости множества допущений, порой необоснованных. Например, в электронике приходится предполагать, что входные сопротивления неизмеримо выше выходных, а мощности любых источников питания и выходных сигналов бесконечно велики. Главное достоинство этого подхода – простота исследования объекта – нередко вступает в противоречие с точностью и надежностью получаемых результатов.

При системном подходе объект моделирования рассматривается как сложная система, состоящая из взаимосвязанных, целенаправленно функционирующих частей и находящаяся во взаимодействии с внешней средой. Системный подход основан на предположении, что целое – *система* – обладает качествами, не свойственными ее частям. При этом части системы могут, в свою очередь, представлять системы. В этом случае их называют *подсистемами*. Подсистемы обладают свойством функциональной полноты, т. е. им присущи все свойства систем.

Основой системного подхода являются структурная декомпозиция объекта моделирования и иерархический подход к моделированию.

Системный подход к моделированию заключается в разбиении всей системы на подсистемы – *декомпозиция системы* – и учете при разработке ее модели не только свойств конкретных подсистем, но и связей между ними. В соответствии с требованиями системного подхода, при разработке модели идут от общего к частному. Общей задаче, сформулированной для системы в целом, подчиняются задачи, решаемые ее составляющими.

Достоинства системного подхода:

- позволяет представить одну сложную задачу совокупностью более простых, которые можно решать быстрее и легче, а главное – параллельно;

– моделирование на любом иерархическом уровне ведется с учетом воздействий внешней среды (это исключает ситуации, когда готовый проект при реализации отказывается работать из-за не идеальности источников питания, внешних помех и наводок, изменения температуры, неучтенной нагрузки и т. п.);

– универсальность: применим как для сложных, так и для простых объектов, а при многоуровневом описании сложных систем он «работает» на любом иерархическом уровне, т. е. применим не только к системе, но и к любой ее части.

Описание объекта моделирования как системы является многокомпонентным (см. рис. 4).

Процедуры системного подхода делятся на процедуры анализа и синтеза. *Процедуры анализа* заключаются в исследовании объекта моделирования или его описания, направленном на получение определенной информации о свойствах объекта.



Рис. 4

Процедуры анализа применяются в том случае, когда объект исследования реально существует и задача сводится к его изучению. Чаще приходится сталкиваться с противоположной задачей: объекта нет, и его предстоит создать – это задача синтеза. *Процедуры синтеза* заключаются в создании описаний объекта (см. рис. 5).

Рассуждая с позиции системного подхода для решения задачи синтеза необходимо: выявить функцию системы (абстрактный синтез), разработать структуру системы (структурный синтез) и определить параметры системы (параметрический синтез) так, чтобы получить желаемое качество проектируемой системы или объекта.

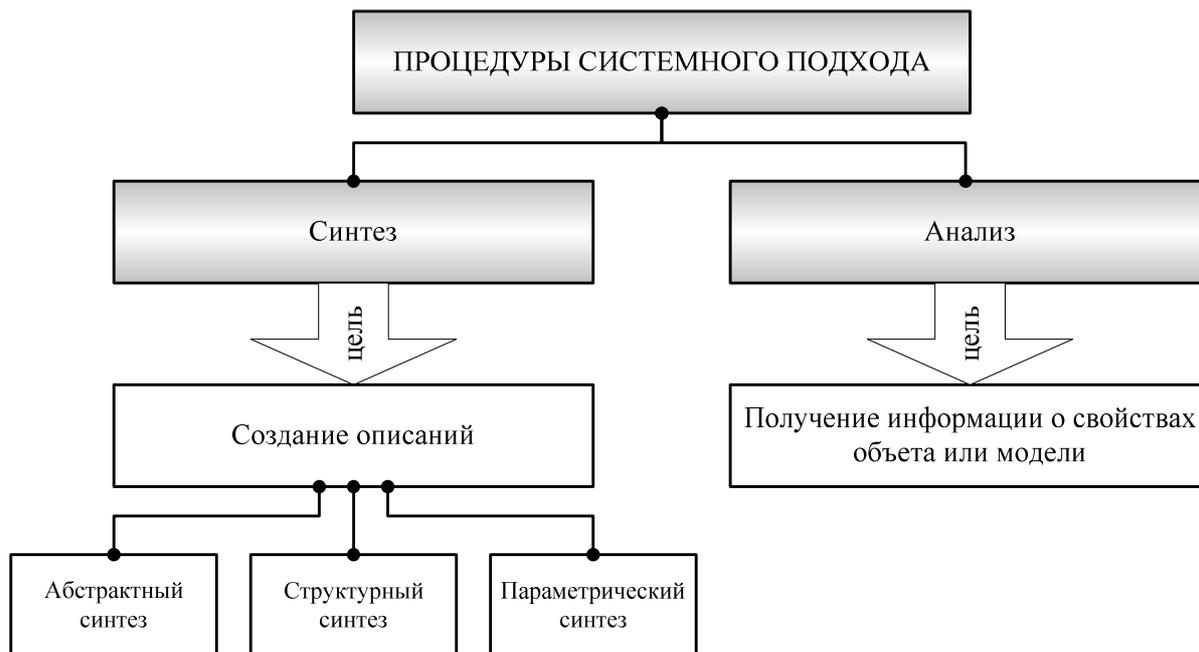


Рис. 5

Процедуры синтеза как правило достаточно сложны. Лишь для узкого класса технических систем разработаны и используются на практике методы формального синтеза. К ним можно отнести пассивные электрические цепи (в частности, фильтры), типовые системы автоматического управления, цифровые автоматы, комбинационные и некоторые несложные регистровые схемы.

В остальных случаях нет альтернативы методу проб и ошибок, реализованному в процедуре итерационного проектирования: сначала создается базовый вариант системы, а затем он улучшается до тех пор, пока не будет достигнуто желаемое качество работы [1, 3, 8].

*ПРИМЕР – совокупность процедур синтеза и анализа в алгоритме итерационного проектирования.*



## 2.2. Иерархия вычислительных систем. Математический аппарат моделирования на различных уровнях декомпозиции

В соответствии с принципами системного подхода каждому иерархическому уровню моделирования соответствует своя степень детализации и набор *структурных примитивов*. Структурным примитивом уровня называется наименьший элемент декомпозиции, который на данном уровне рассматривается как неделимый. Графическую иерархию объектов пред-

ставляют в виде усеченной пирамиды, расширение которой книзу означает увеличение степени детализации и количества примитивов, которые должны рассматриваться при моделировании.

Рассмотрим иерархию уровней моделирования вычислительных систем в соответствии с выделяемыми структурными примитивами и видами используемого моделирования (см. рис. 6).

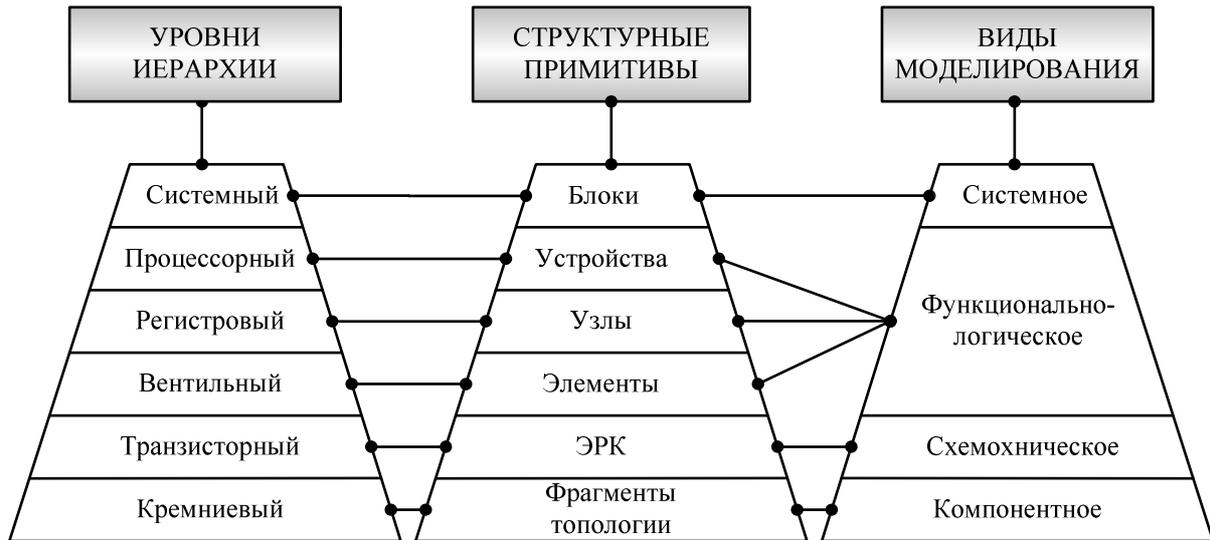


Рис. 6

Для процессорного уровня иерархии вычислительных систем структурными примитивами являются отдельные устройства: память, порты, микропроцессоры, т. е. микросхемы большой степени интеграции (БИС). Для регистрового уровня в качестве структурных примитивов выступают узлы – регистры, счетчики, мультиплексоры, АЛУ, дешифраторы. Для вентильного уровня это логические элементы и триггеры. На транзисторном уровне – электрорадиокомпоненты: транзисторы, диоды, резисторы, конденсаторы и проч. На кремниевом уровне эти же компоненты предстают уже не как черные ящики, а как некоторое множество топологических фрагментов в объеме полупроводникового кристалла (области диффузии, металлизации и т. п.).

На различных уровнях декомпозиции используется различный математический аппарат моделирования. В *компонентном моделировании* исследуются процессы, протекающие в трехмерной среде и в непрерывном времени. Для описания этих процессов используются дифференциальные уравнения в частных производных, в которых в качестве независимых переменных фигурируют время  $t$  и пространственные координаты  $x$ ,  $y$  и  $z$ .

Для описания электрических процессов в полупроводниковых компонентах электронных схем такими уравнениями являются уравнения непрерывности, переноса и Пуассона; для описания тепловых процессов – уравнение теплопроводности.

*ПРИМЕР преобразования модели технического объекта компонентного вида к схематехническому.*



В *схематехническом моделировании* рассматриваются более сложные системы – совокупности компонентов, функционирующих в составе электронной схемы. Для моделирования на приемлемом уровне здесь осуществляется переход от непрерывного к дискретному пространству при сохранении непрерывного представления времени. Поэтому математическим аппаратом моделирования и анализа электрических процессов в электронных схемах является аппарат численного решения обыкновенных дифференциальных уравнений, в статических случаях вырождающихся в алгебраические. В этих уравнениях в качестве единственной независимой переменной фигурирует время.

*ПРИМЕР преобразования моделей технических объектов схематехнического вида к функционально-логическому.*



На уровне *функционально-логического моделирования* в моделях отражаются процессы преобразования информации. Вместо фазовых переменных, описывающих электрические, магнитные или тепловые процессы, используются переменные, отражающие информационное состояние объектов. Для цифровой аппаратуры этими объектами являются логические операторы, состояние которых характеризуется дискретными, чаще всего булевыми, величинами. Поэтому используемый здесь математический аппарат – математическая логика, в том числе булева алгебра и теория конечных автоматов. На этом уровне в модели отображаются действия, которые выполняются моделируемым объектом в соответствии с алгоритмом функционирования. Здесь еще можно отождествить информационные переменные с сигналами: поставить в соответствие сигналу некоторую физическую величину – напряжение или ток на выходе конкретного элемента, однако целью моделирования такое представление уже не является.

На *системном уровне моделирования* происходит окончательное абстрагирование от физической сущности информационных процессов. Состояние некоторого устройства системы характеризуется только тем, заня-

то устройство обработкой информации на данном отрезке времени или нет. Обработываемая информация делится на составные части – задачи (заявки, запросы, транзакты). Задачи в отличие от сигналов не привязаны жестко к какому-либо устройству, а обслуживаются в системе, перемещаясь от устройства к устройству в соответствии с алгоритмом функционирования системы и своими параметрами. При проектировании системы необходимо обеспечить обслуживание заданного потока заявок при рациональном использовании оборудования. Математическим аппаратом анализа на системном уровне является теория массового обслуживания и алгоритмические модели.

Способ описания взаимодействия структурных примитивов в рамках функциональной структуры также зависят от уровня декомпозиции и цели моделирования. *Иерархический подход* или *концепция слойной структуры*. В основе этой концепции лежат понятия уровня (слоя) и отношение иерархии. Слои представляют собой объекты структуры, а иерархия – вид отношения между объектами. Достоинством этой структуры является высокий уровень абстракции описания, что позволяет ее использовать на начальных этапах моделирования.

*Теоретико-автоматная концепция* базируется на понятиях «состояние» и «переход». Достоинством концепции является высокая точность описания системы, которая требует использования большого количества исходных данных. Примерами таких структур являются: конечный автомат, сеть Петри, формальная грамматика, таблица решений и т. д.

В основе *теоретико-графовой концепции* лежат понятия «узлы» и «дуги». Достоинство – большая наглядность. Примером реализации являются логические и структурные схемы.

*Концепция массового обслуживания* базируется на понятиях «ресурс» и «заявка». Реализацией концепции являются структуры систем и сетей массового обслуживания [1-3, 9, 10].

### **2.3. Моделирование структурных примитивов**

При моделировании структурных примитивов на любом уровне декомпозиции рассматривают функциональный блок вида, представленного на рис. 7. Чтобы моделировать такой блок, необходимо знать или предположить известными два из трех объектов описания.

Случай, когда известны уравнения, описывающие структурный примитив, предполагается заданным входное воздействие и требуется найти отклик системы (см. рис. 7-а), относится к классу задач *управления*. Более

сложная задача возникает, когда заданы множество векторов входных воздействий и соответствующее ему множество значений вектора базисных координат (см. рис. 7-б), а найти необходимо математическое описание структурного примитива. Это задачи *идентификации*.

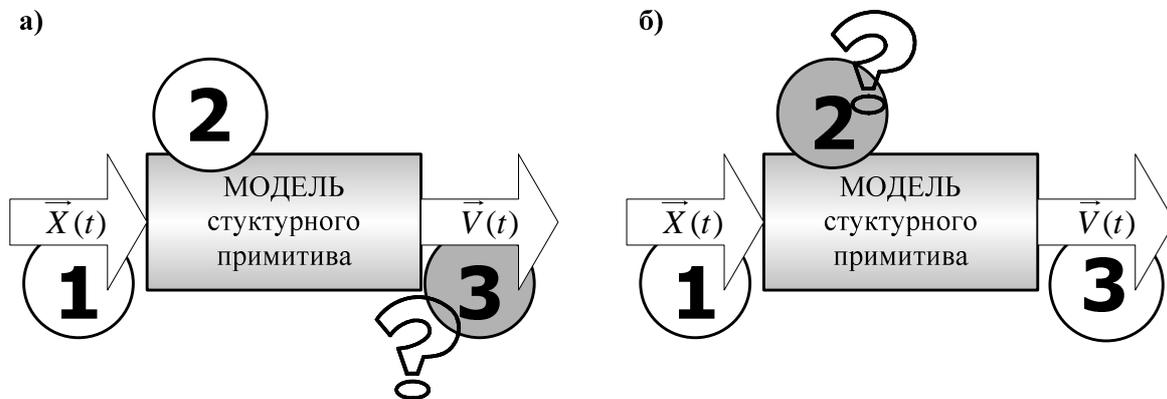


Рис. 7

### 2.3.1. Задача управления

Для решения задач управления на уровне схмотехнического и функционально-логического моделирования используют математические методы *теории автоматического управления*. Аппарат теории управления предполагает анализ систем на основании моделей управления, которые представляют собой совокупность дифференциальных уравнений связи между входными воздействиями и базисными переменными системы или ее элементов, которые называются *звеньями системы*.

Рассмотрим произвольное звено системы, описываемое входным воздействием  $x(t)$ , выходной фазовой координатой  $v(t)$  и внешним возмущением  $f(t)$ . В общем случае ДУ звена имеет нелинейный вид:

$$F(x, x' \dots x^{(n)}, v, v' \dots v^{(m)}) = \varphi(f, f' \dots f^{(l)}).$$

В задачах анализа чаще используют линейризованные модели звеньев. В основе линейризации нелинейных уравнений лежит предположение о том, что в исследуемом процессе переменные  $x$  и  $v$  изменяются так, что их отклонения от установившихся значений остаются все время достаточно малыми. Это условие обычно выполняется, поскольку этого требует сама идея целенаправленной устойчивой работы системы. Внешнее возмущение  $f$  как правило является случайным и несанкционированным, а следовательно в отношении него предположение о малости отклонений может оказаться несправедливым, поэтому правая часть ДУ звена линейризации не подлежит. Таким образом, линейризованные модели являются упрощен-

ными, так как описывают поведение звеньев в отсутствии помех и возмущений. Геометрически линеаризация является заменой реальной нелинейной характеристики системы или звена на линейную.

*ПРИМЕР геометрической линеаризации характеристики звена.*



Линеаризованные модели управления имеют вид:

$$a_n \frac{d^n v(t)}{dt^n} + a_{n-1} \frac{d^{n-1} v(t)}{dt^{n-1}} + \dots + a_0 v(t) = b_m \frac{d^m x(t)}{dt^m} + b_{m-1} \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_0 x(t).$$

Решением данного уравнения для заданного  $x(t)$  является переходная характеристика звена.

*ПРИМЕРЫ получения моделей с использованием экспериментального и теоретического подходов.*



Для упрощения операций преобразования моделей звеньев и систем используют *передаточные функции*. Передаточной функцией называется отношение изображения по Лапласу выходной величины системы (звена) к изображению по Лапласу входной величины при нулевых начальных условиях. Изображением по Лапласу функции  $f(t)$  является функция  $F(p)$ :

$$F(p) = \int_0^{\infty} e^{-pt} f(t) dt,$$

где  $t$  – независимая переменная;  $p=d/dt$ .

Основные свойства изображений имеют вид:

$$f(t) = \frac{d\varphi}{dt} \Rightarrow F(p) = p\varphi(p) - p\varphi(0);$$

$$f(t) = \frac{d^2\varphi}{dt^2} \Rightarrow F(p) = p^2\varphi(p) - p^2\varphi(0) - p\varphi'(0);$$

$$f(t) = \frac{d^n\varphi}{dt^n} \Rightarrow F(p) = p^n\varphi(p) - p^n\varphi(0) - p^{n-1}\varphi'(0) - \dots - p\varphi^{(n-1)}(0);$$

$$f(t) = \int_0^1 \varphi(t) dt \Rightarrow F(p) = \frac{1}{p}\varphi(p).$$

Таким образом, при нулевых начальных условиях, т. е. при  $f(0)=0$ , переход от оригинала функции к ее изображению по Лапласу может быть осуществлен формальной заменой дифференцирования на символическое умножение на  $p$ , а символа интегрирования – на умножение на  $1/p$ . Передаточная функция, полученная для линеаризованного ДУ имеет вид:

$$W(p) = v(p) / x(p) = \frac{\sum_{i=0}^m b_i p^i}{\sum_{j=0}^n a_j p^j}.$$

Переход от передаточной функции к ДУ звена осуществляется обратным преобразованием Лапласа [6, 11].

*ПРИМЕРЫ взаимного преобразования моделей звеньев.*



### 2.3.2. Задача идентификации

Результатом решения задачи идентификации является имитационная или аналитическая модель структурного примитива. Как уже отмечалось, аналитические модели делятся на теоретические и экспериментальные. Теоретические модели получают на основе известных описаний процессов функционирования объекта. Экспериментальные – на основе изучения поведения объекта моделирования во внешней среде.

Для построения экспериментальных моделей используют:

- методы аппроксимация зависимостей;
- методы корреляционного и регрессионного анализа;
- методы планирования эксперимента.

Пусть экспериментальная статистика функционирования структурного примитива задана таблично. В этом случае значения функции базисной координаты известны только для дискретных значений входной переменной. Для того чтобы вычислять значение базисной координаты в любой произвольной точке, необходимо восстановить непрерывную функцию  $v=f(x)$ . Такое приближение называют *аппроксимацией*.

Аппроксимация характеристик структурных примитивов применяется в следующих случаях:

- если аналитическое описание характеристики неизвестно и она задана набором экспериментальных данных;
- аналитическое описание  $v=f(x)$  сложное и затрудняет расчеты.

Постановка задачи аппроксимации имеет два варианта. В первом случае осуществляется поиск аппроксимирующей функции, наилучшим образом описывающей экспериментальную статистику, при условии обязательного прохождения графика аппроксимирующей функции через определенные заданные точки экспериментальной статистики (см. рис. 8-а). Эти точки называются *узлами*. Второй вариант постановки задачи аппроксимации не имеет ограничивающего условия обязательного прохождения функции через узлы (см. рис. 8-б).

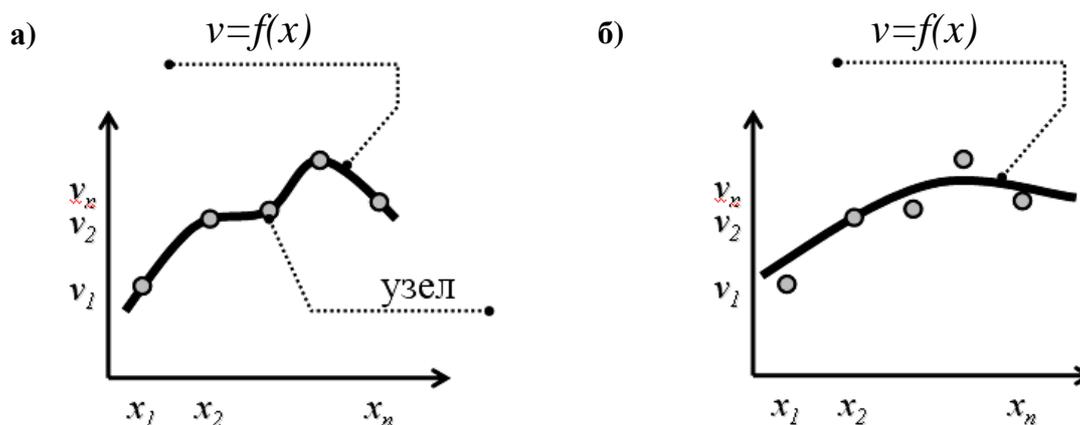


Рис. 8

Для решения *первой задачи* используются методы *кусочно-линейной аппроксимации* и *аппроксимации сплайнами*.

Кусочно-линейная аппроксимация получается соединением узлов отрезками прямых линий. Узлы располагаются так, чтобы обеспечить наименьшую ошибку между аппроксимирующей и точной функцией.

Главный недостаток кусочно-линейного приближения заключается в том, что, хотя аппроксимирующая функция непрерывна в узлах, ее производные в этих точках имеют разрывы. Другой недостаток – требование большого числа узлов для достаточно точного описания характеристик.

*ПРИМЕР* использования кусочно-линейной аппроксимации.



Для преодоления указанного недостатка можно в качестве аппроксимирующей функции использовать полином высокой степени. С математической точки зрения для  $n$  заданных узлов экспериментальной статистики всегда найдется полином степени  $n$ , график которого строго пройдет через все узлы. Однако на практике при большом числе пар значений такая

аппроксимация часто бывает бессмысленной или требует неоправданных затрат ресурсов. Чаще используют аппроксимацию *сплайнами*. В отличие от интерполяции полиномом, которым описывается вся область данных, при интерполяции сплайнами строится отдельный полином, описывающий интервал от узла  $x_{i-1}$  до узла  $x_i$  (см. рис. 9).

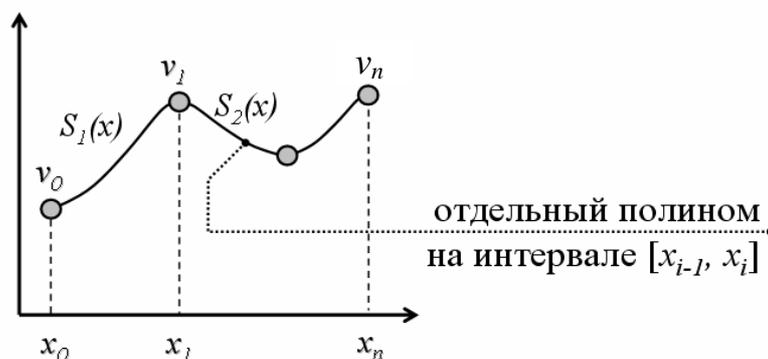


Рис. 9

Наиболее часто используют полиномы третьей степени – кубические сплайны. На каждом отрезке  $[x_{i-1}, x_i]$  кубический сплайн  $S_i(x)$  является многочленом третьей степени:

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i.$$

В узлах  $x_i$  сплайн  $S_i(x)$  принимает заданные значения  $v_i$ ,  $i = \overline{0, n}$ :

$$S_i(x_{i-1}) = v_{i-1}, i = \overline{1, n}, \quad (1)$$

$$S_i(x_i) = v_i, i = \overline{1, n}. \quad (2)$$

Условия (1) и (2) требуют, чтобы сплайны соприкасались в заданных точках. Количество условий таких условий равно  $2n$ . Во внутренних узлах  $x_i$ ,  $i = \overline{1, n-1}$  сплайн имеет непрерывную первую и вторую производные:

$$S'_i(x_i) = S'_{i+1}(x_i), i = \overline{1, n-1}. \quad (3)$$

$$S''_i(x_i) = S''_{i+1}(x_i), i = \overline{1, n-1}. \quad (4)$$

Условия (3) и (4) означают, что в местах соприкосновения сплайнов их первые и вторые производные должны быть равны. Таких условий  $2n - 2$ . Для отыскания искомого сплайна требуется найти коэффициенты  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$  многочленов  $S_i(x)$ ,  $i = \overline{1, n-1}$ , т. е.  $4n$  неизвестных. Однако количество уравнений, записанных по условиям (1) – (4) равно  $4n - 2$ . Чтобы система алгебраических уравнений имела решение, необходимо, чтобы число уравнений равнялось числу неизвестных. Следовательно, для

разрешимости задачи нужны еще два дополнительных условия. Их обычно получают из естественного предположения о нулевой кривизне графика сплайна на концах:

$$S''(x_0) = 0, S''(x_n) = 0.$$

Полученный таким образом сплайн называют естественным. Если есть дополнительные сведения о поведении функции на концах интервала интерполяции, то можно записать другие краевые условия.

Для решения *второй задачи* аппроксимации используют методы *регрессионного и корреляционного анализа*. При помощи регрессионного анализа на основании экспериментальных данных строится уравнение, вид которого задает исследователь. Например, линейный парный регрессионный анализ заключается в определении параметров эмпирической линейной зависимости  $v(x) = b_1x + b_0$ , где для определения неизвестных коэффициентов  $b_1$  и  $b_0$  используют систему линейных уравнений

$$\begin{aligned} b_0N + b_1 \sum_{i=1}^N x_i &= \sum_{i=1}^N v_i, \\ b_0 \sum_{i=1}^N x_i + b_1 \sum_{i=1}^N x_i^2 &= \sum_{i=1}^N x_i v_i. \end{aligned}$$

Решение этой системы имеет вид:

$$\begin{aligned} b_1 &= \frac{\sum_{i=1}^N x_i \sum_{i=1}^N v_i - N \sum_{i=1}^N x_i v_i}{\left( \sum_{i=1}^N x_i \right)^2 - N \sum_{i=1}^N x_i^2}, \\ b_0 &= \frac{1}{N} \left( \sum_{i=1}^N v_i - b_1 \sum_{i=1}^N x_i \right). \end{aligned}$$

Найденные коэффициенты регрессионного уравнения обеспечивают минимум среднеквадратических отклонений расчетных значений  $v$  от экспериментальных. Параболическая регрессия обеспечивает получение трех параметров  $b_0$ ,  $b_1$  и  $b_2$  приближения экспериментальной зависимости параболической функцией  $v(x) = b_0 + b_1x + b_2x^2$ .

Для многофакторных моделей (две и более переменных  $x$ ) наиболее распространенными являются модели в виде полинома степени  $d$ :

$$\begin{aligned} v &= a_0 + \sum_{i=1}^n a_i x_i + \sum_{i \leq j \leq n} \sum a_{ij} x_i x_j + \sum_{i_1 i_2 \dots i_n} a_{i_1 i_2 \dots i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}; \\ &\sum i_n \leq d. \end{aligned}$$

В частном случае это может быть линейный полином  $a_0 + a_1x_1 + a_2x_2$ , или неполный квадратичный полином  $a_0 + a_1x_1 + a_2x_2 + a_{12}x_1x_2$ .

Корреляционный анализ позволяет оценить степень взаимосвязи двух переменных по набору их экспериментальных значений. Применительно к задачам аппроксимации корреляционный анализ используют для оценки согласованности выбранной аппроксимации с экспериментальными данными. Для оценки рассчитывают *коэффициент корреляции* между расчетными  $y_p$  и экспериментальными  $y_{э}$  значениями базисной переменной:

$$R = \frac{\sum_{i=1}^N y_{p_i} y_{э_i} - \frac{\sum_{i=1}^N y_{p_i} \sum_{i=1}^N y_{э_i}}{N}}{\sqrt{\sum_{i=1}^N y_{p_i}^2 - \frac{\left(\sum_{i=1}^N y_{p_i}\right)^2}{N}} \sqrt{\sum_{i=1}^N y_{э_i}^2 - \frac{\left(\sum_{i=1}^N y_{э_i}\right)^2}{N}}}$$

Значение  $R$  лежит в пределах от -1 до +1. Коэффициент -1 соответствует максимальной отрицательной корреляции, когда с ростом параметра, откладываемого по оси  $x$ , параметр откладываемый по оси  $y$  уменьшается, а все точки лежат точно на прямой. Значение 0 соответствует отсутствию корреляции; +1 – максимальной положительной корреляции. Чем ближе значение  $R$  по модулю к 1, тем бóльшую точность имеет аппроксимация. Квадрат коэффициента корреляции называется *коэффициентом детерминации* или *коэффициентом точности аппроксимации*.

*ПРИМЕРЫ* использования корреляционного анализа для оценки взаимосвязи переменных и точности аппроксимации.



Получение экспериментальной статистики функционирования объекта моделирования (структурного примитива на  $n$ -ом уровне иерархии или системы в целом) для последующей идентификации методами регрессионного и корреляционного анализа является самостоятельной задачей, которая решается методами планирования эксперимента. Различают три типа экспериментов:

- активный: экспериментатор преднамеренно параметры системы с целью получения ее модели;
- пассивный: связь между параметрами устанавливается по результатам наблюдений за самопроизвольными изменениями в системе;
- смешанный.

Под *планированием эксперимента* понимается процедура выбора числа опытов и условий их проведения, необходимых для идентификации с требуемой точностью. Все переменные, определяющие объект моделирования, изменяются одновременно по специальным правилам. Результаты эксперимента представляются в виде математической модели, обладающей определенными статистическими свойствами, например минимальной дисперсией оценок параметров модели.

Варьирование одного фактора на нескольких уровнях при фиксированном постоянном значении всех остальных факторов называется *однофакторным экспериментом*. В этом случае можно получить количественную оценку эффекта только одного фактора. Влияние других факторов оценить нельзя. Выводы о влиянии изучаемого фактора могут существенно различаться в зависимости от уровня фиксирования прочих факторов. Это может привести к неадекватности моделей идентификации. Только в тех случаях, когда отклик является функцией одного фактора, однофакторный эксперимент вполне закономерен.

Однако на практике приходится иметь дело с многофакторными объектами, где однофакторный эксперимент неэффективен. В *многофакторных* планах эксперимента *одновременно* варьируется несколько факторов, а не каждый в отдельности. В планировании эксперимента сам эксперимент рассматривается как объект исследования и оптимизации. Планирование многофакторных экспериментов – основной подход к организации и проведению экстремальных исследований сложных систем. Цель планирования эксперимента – извлечение максимума информации при заданных затратах на эксперимент либо минимизация затрат при получении информации, достаточной для решения задач. Планирование эксперимента позволяет соразмерить число опытов поставленной задаче.

*ПРИМЕР* логических оснований планирования эксперимента.



Табличная форма плана эксперимента называется *матрицей планирования*. Построение матрицы начинается с определения числа рассматриваемых факторов и уровней варьирования факторов. Уровни варьирования кодируются. Если при проведении эксперимента реализуются все возможные сочетания уровней факторов, такой эксперимент называется *полным факторным экспериментом*.

Матрица полного трехфакторного двухуровневого эксперимента имеет вид, приведенный в таблице 2. Данный план обладает следующими свойствами:

1) симметричностью относительно центра эксперимента – сумма элементов столбца матрицы для каждого фактора равна нулю;

2) попарной ортогональностью столбцов – сумма почленных произведений их элементов равна нулю:  $\sum_{i=1}^N x_{ji}x_{ui} = 0, \quad j \neq u, \quad u = 1, 2, \dots, k$ ;

3) нормировкой – сумма квадратов элементов каждого столбца равна числу опытов:  $\sum_{i=1}^N x_{ij}^2 = N$ .

Таблица 2

Номер опыта	$x_1$	$x_2$	$x_3$
1	+	+	+
2	+	-	-
3	-	+	-
4	-	-	+
5	+	+	-
6	-	+	+
7	+	-	+
8	-	-	-

Свойства плана полного трехфакторного двухуровневого эксперимента обеспечивают независимость оценок коэффициентов регрессионной функции. Коэффициенты регрессии оцениваются по всем  $N$  опытам и, соответственно, в  $N$  раз уменьшается дисперсия их оценки по сравнению с дисперсией единичного опыта. Если опыты выполняются произвольным образом, то оценки закоррелированы.

Ортогональность обеспечивает хорошие статистические свойства оценок и приводит к простой формуле расчета  $b$ -коэффициентов регрессионной модели:

$$b_j = \sum_{i=1}^n \frac{x_{ji}y_i}{N}.$$

При этом важно помнить, что планирование эксперимента само по себе не может улучшить физический смысл модели, оно только улучшает ее статистические свойства [9, 12 – 15].

## 2.4. Вероятностное моделирование. Метод Монте-Карло

Если некоторые входные или базисные переменные объекта моделирования носят случайный характер, то помимо стандартных задач анализа, целью моделирования является определение статистических характеристик выходных характеристик объекта: плотности распределения параметров, математических ожиданий, среднеквадратических отклонений (дисперсий) и т. п., а также вероятности тех или иных событий.

*ПРИМЕРЫ вероятностных характеристик случайных процессов.*



Наибольшее распространение получили вероятностные методы статистического анализа – аналитический и численный, основанный на применение *метода Монте-Карло (метод статистических испытаний)*

Аналитический метод состоит в поиске аппроксимирующей функции для функции распределения:  $f(x)=F_i(v_i,t)$ . Этот метод обладает сравнительно невысокой точностью и значительной трудоемкостью.

В методе Монте-Карло данные о моделируемых событиях вырабатываются искусственно путем использования генератора равномерно распределенной в интервале  $[0,1]$  случайной величины  $x^R$  в сочетании с интегральной функцией распределения вероятностей для исследуемого процесса. В результате использования метода получается серия частных значений случайных величин, статистическая обработка которых позволяет получить сведения о характеристиках. Чем больше реализаций случайного процесса (прогонов), тем точнее результат анализа. Так, при оценке, например, математического ожидания (МО) случайной величины, каждая новая реализация случайного процесса дает новое значение, изменяющее значение МО. После первой реализации имеется единственное значение случайной величины  $x_1$ , и МО равно этому значению  $MO_1 = x_1$ . После второй реализации МО рассчитывается как среднее двух случайных величин:

$$MO_2 = \frac{x_1 + x_2}{2} = \frac{MO_1}{2} + \frac{x_2}{2}.$$

После третьей реализации:

$$MO_3 = \frac{x_1 + x_2 + x_3}{3} = \frac{2MO_2}{3} + \frac{x_3}{3},$$

и так далее. В пределе для произвольного числа реализаций  $n$  случайного процесса имеем:

$$MO_n = \frac{(n-1)MO_{n-1} + x_n}{N}. \quad (5)$$

Из формулы (5) следует, что каждое новое значение случайной величины оказывает все меньшее влияние на МО. Следовательно, с увеличением числа реализаций случайного процесса точность моделирования по МО возрастает, и в пределе, при числе реализации, стремящихся к бесконечности, МО, рассчитанное по экспериментальным модельным реализациям случайного процесса, будет стремиться к реальному МО.

Для оценки вероятности наступления некоторого события в моделируемой системе, используется известное из теории вероятностей положение, согласно которому, если некоторое случайное событие  $A$  наступает как следствие какого-либо из  $n_A$  событий при общем числе  $n$  возможных событий (несовместимых и равновероятных), то частота наступления события  $A$  при увеличении числа испытаний стремиться к вероятности:

$$P_A = \lim_{n \rightarrow \infty} \frac{n_A}{n},$$

при условии существования данного предела.

В методе Монте-Карло для реализации генератора с заданной функцией распределения используют источник случайных чисел с равномерным распределением в интервале  $(0,1)$ . *Равномерным* называют такое распределение, при котором каждое из чисел избранного интервала имеет одинаковую вероятность появления. Искусственную случайную выборку из совокупности величин, описываемой некоторой функцией распределения вероятностей, получают в следующем порядке:

- 1) строят график или таблицу функции распределения на основе ряда чисел, отражающего исследуемые события или процессы, откладывая по оси  $x$  значения случайной переменной, а по оси  $y$  – значения вероятностей;
- 2) выбирают случайное число с помощью генератора случайных чисел в пределах от 0 до 1 (с требуемым числом разрядов);
- 3) строят линию, параллельную оси  $x$ , от значения выбранного случайного числа до пересечения с кривой распределения вероятностей и, затем, от полученной точки линию, параллельную оси  $y$ , до пересечения с осью  $x$ , получая выборочное значение случайной величины по заданному типу распределения.

Рассмотрим пример использования метода Монте-Карло для дискретного распределения вероятностей. Пусть вероятность одновременного

обращения к серверу определенного количества удаленных клиентов в каждый 10-ти минутный интервал времени соответствует распределению, заданному в таблице 3.

Таблица 3

<b>Число клиентов</b>	0	1	2	3	4
<b>Вероятность</b>	0,2	0,25	0,3	0,15	0,1
<b>Суммарная вероятность</b>	0,2	0,45	0,75	0,9	1

Построим график распределения вероятностей (см. рис. 10-а). Проведем мысленный эксперимент для 5 интервалов времени. Выберем пять случайных чисел, каждое из которых используем для определения числа клиентов, одновременно обращающихся к серверу, в каждый из 10-ти минутных интервалов. Так, если первое случайное число равно 0,38, то в результате «прогона» получаем значение числа клиентов, равное 1 (см. рис. 10-а). Результаты эксперимента приведены в таблице 4.

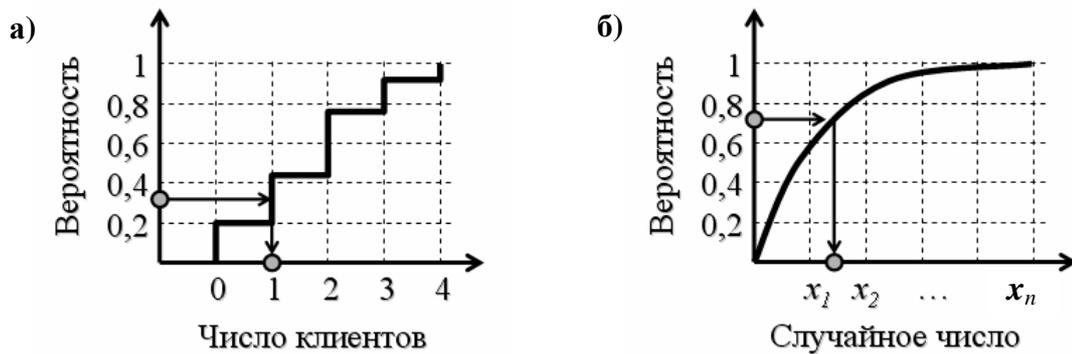


Рис. 10

Таблица 4

<b>Интервал времени</b>	1	2	3	4	5
<b>Случайное число</b>	0,38	0,01	0,41	0,81	0,32
<b>Число клиентов</b>	1	0	1	3	1

Рассмотрев большое количество выборок, нетрудно убедиться, что каждое из значений числа клиентов в процессе данного эксперимента будет появляться с относительной частотой, равной заданной вероятности.

В примере использованы дискретные значения вероятностей. Если распределение вероятностей непрерывное, способ применения метода остаётся без изменений, но кривая распределения будет не ступенчатой, а плавной (см. рис. 10-б). В любом случае, если случайная величина  $x$  может

иметь только положительные значения, и задана плотностью распределения  $f(x)$ , то:

$$\int_0^x f(x)dx = x^R. \quad (6)$$

Если возможно аналитическое вычисление интеграла в формуле (6), то для определения  $x$  можно получить уравнение. Например, для экспоненциального распределения  $f(x) = \lambda e^{-\lambda x}$ , следовательно:

$$\begin{aligned} \int_0^x \lambda e^{-\lambda x} dx &= x^R, \\ 1 - e^{-\lambda x} &= x^R, \\ x^E &= -\frac{1}{\lambda} \ln(1 - x^R). \end{aligned}$$

Однако, величина  $(1 - x^R)$ , распределена точно так же, как  $x^R$ , поэтому можно для розыгрыша случайных чисел  $x^E$ , распределенных по экспоненциальному закону, использовать формулу [2, 6, 16-18]:

$$x^E = -\frac{1}{\lambda} \ln(x^R).$$

*ПРИМЕРЫ моделирования с использованием метода Монте-Карло других неравномерных распределений – см. Лабораторную работу №4*



Метод Монте-Карло может быть использован и для решения некоторых детерминированных задач: например, приближенного вычисления интеграла вида:

$$v = \int_{x_1}^{x_2} f(x)dx.$$

Геометрически такой интеграл равен площади фигуры, ограниченной графиком функции  $f(x)$ , приведенным на рис. 11.

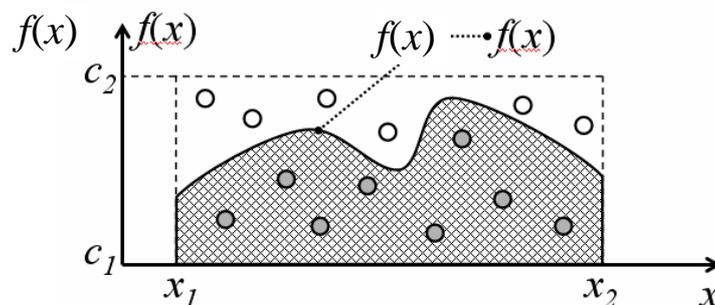


Рис. 11

Распределим случайным образом точки в прямоугольнике поиска (см. рис. 11). Обозначим через  $N_1$  количество точек, попавших в прямоугольник, и через  $N_2$  – количество точек под кривой, т. е. попавших в заштрихованную площадь под функцией (эти точки изображены на рис. 11 закрашенными). Количество точек, попавших под кривую по отношению к общему числу точек пропорционально площади под кривой (величине интеграла)  $S$  по отношению к площади испытываемого прямоугольника:

$$\frac{N_2}{N_1} = \frac{S}{(x_2 - x_1)(c_2 - c_1)},$$
$$S = \frac{N_2}{N_1} (x_2 - x_1)(c_2 - c_1).$$

Вычисление  $S$  будет тем более точным, чем большее число точек будет использовано.

### 3. Моделирование вычислительных сетей и систем

Эффективность построения и использования корпоративных информационных систем в условиях ограниченного финансирования является актуальной задачей. Критериями оценки эффективности могут служить снижение стоимости реализации информационной системы, соответствие текущим требованиям и требованиям ближайшего времени, возможность и стоимость дальнейшего развития и перехода к новым технологиям.

Основу информационной системы составляет вычислительная система, включающая такие компоненты, как кабельная сеть и активное сетевое оборудование, компьютерное и периферийное оборудование, оборудование хранения данных (библиотеки), системное программное обеспечение (операционные системы, системы управления базами данных), специальное ПО (системы мониторинга и управления сетями) и в некоторых случаях прикладное ПО.

Задача моделирования всех функций вычислительной системы (ВС) в полном объеме в силу структурной и функциональной сложности ВС трудно разрешима. Однако возможно моделирование работы подсистем ВС в динамике, при этом его результаты позволяют по косвенным показателям судить о функционировании всей системы. Основу моделирования ВС составляют модели оборудования и процессов (технологий, программного обеспечения) разных уровней.

### 3.1. Автоматная концепция функционального моделирования

В зависимости от назначения модели разработчика может интересовать либо специфические характеристики работы структурных примитивов вычислительной сети, либо только время, которое необходимо примитиву для осуществления того или иного процесса и/или смена состояния системы в целом в результате завершения очередного процесса. Вторым вариантом реализуем в рамках теоретико-автоматной концепции, предполагающей получение абстрактного описания динамического поведения распределенной системы путем задания множества процессов и его отображения на множество состояний. Реализацией этой концепции являются абстрактные конечные автоматы и сети Петри.

#### 3.1.1. Абстрактные конечные автоматы

*Абстрактный автомат* задается как совокупность шести объектов:

- множества входных сигналов  $X$  (*входной алфавит автомата*);
- множества выходных сигналов  $Y$  (*выходной алфавит автомата*);
- множества состояний автомата  $A$ ;
- элемента  $a_0 \in A$ , называемого начальным состоянием автомата;
- функций переходов  $\varphi(a, x)$  и выходов  $\psi(a, x)$ , задающих однозначные отображения множества  $(a, x)$ , где  $a \in A$  и  $x \in X$ , в множества  $A$  и  $Y$ .

Абстрактный автомат функционирует в *дискретном времени*, в каждый момент этого времени имеет определенное состояние  $a(t)$  из множества  $A$  состояний автомата. В каждый момент времени, отличный от начального, автомат способен воспринимать входной сигнал  $x(t)$  – произвольную букву входного алфавита  $X$  и выдавать соответствующий выходной сигнал  $y(t)$  – определенную букву выходного алфавита.

Закон функционирования автомата *первого рода* задается уравнениями вида:

$$\begin{aligned} a(t) &= \varphi[a(t-1), x(t)]; \\ y(t) &= \psi[a(t-1), x(t)]; \\ t &= 1, 2, \dots; \end{aligned} \tag{7}$$

Закон функционирования автомата *второго рода*:

$$\begin{aligned} a(t) &= \varphi[a(t-1), x(t)]; \\ y(t) &= \psi[a(t), x(t)]; \\ t &= 1, 2, \dots \end{aligned} \tag{8}$$

В практике используют:

- *автомат Мили*: произвольный конечный автомат первого рода;
- *автомат Мура*: частный случай конечных автоматов второго рода, у которого функция выходов  $\psi(a,x)$  не зависит от переменной  $x$ .

Автомат называется *конечным* если конечно число его состояний. Автоматы задают табличным способом или направленным графом. В первом случае строят матрицы *переходов* и *выходов*. Строки обеих этих таблиц обозначаются входными сигналами автомата, а столбцы – его состояниями. На пересечении строки и столбца таблицы переходов ставится соответствующее значение функции переходов  $\varphi(a,x)$ , а в таблице выходов – значение  $\psi(a,x)$ . Для автомата Мура сдвинутая таблица выходов сводится к одной строке, поэтому часто в таблице переходов над каждым состоянием  $a_i$  автомата, обозначающим тот или иной столбец таблицы, ставят соответствующий этому состоянию выходной сигнал  $\varphi(a_i,x) = \psi(a_i)$ .

При задании автомата с использованием направленного графа вершины графа отождествляются с состояниями автомата, а стрелки – с выходными сигналами. Если входной сигнал  $x_i$  вызывает переход автомата из состояния  $a_j$  в состояние  $a_k$ , то на графе автомата этому сигналу соответствует помеченная буквой  $x_i$  стрелка, соединяющая вершину, соответствующую состоянию  $a_j$ , с вершиной, соответствующей состоянию  $a_k$ . Для задания функции выхода ребра графа также помечаются соответствующими выходными сигналами. Если обозначенная входным сигналом  $x_i$  стрелка соединяет вершину  $a_j$  с  $a_k$ , то в случае автоматов первого рода ей предписывается выходной сигнал  $\psi(a_j, x_i)$ , а в случае автоматов второго рода – выходной сигнал  $\psi(a_k, x_i)$  (см. рис. 12).

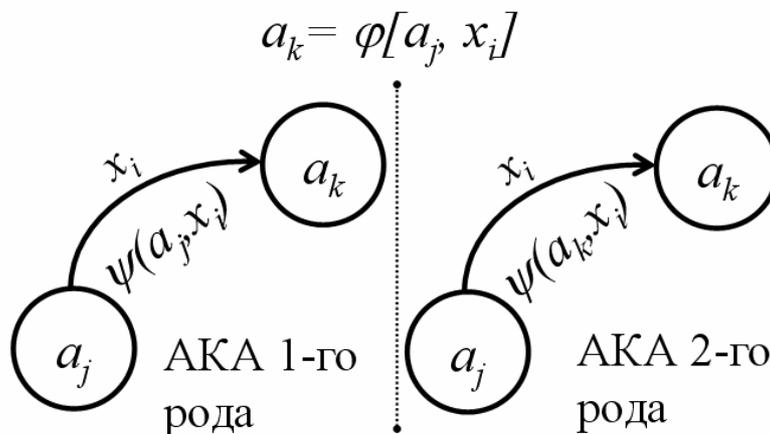


Рис. 12

Пример графа АКА первого рода представлен на рис. 13, а соответствующие ему матрицы переходов и выходов – в таблицах 5 и 6.

Таблица 5

**Матрица переходов**

	A	B	C	D	E
x <sub>1</sub>	B	C	D	E	A
x <sub>2</sub>	E	C	D	D	E
x <sub>3</sub>	A	B	D	B	A

Таблица 6

**Матрица выходов**

	A	B	C	D	E
x <sub>1</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>
x <sub>2</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>2</sub>	y <sub>4</sub>	y <sub>5</sub>
x <sub>3</sub>	y <sub>5</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>5</sub>	y <sub>1</sub>

Пример графа АКА второго рода представлен на рис. 14, а соответствующие ему матрицы переходов и выходов – в таблицах 7 и 8.

Таблица 7

**Матрица переходов**

	a	b	c	d	e
x <sub>1</sub>	b	c	e	a	d
x <sub>2</sub>	c	d	b	e	a

Таблица 8

**Матрица выходов**

	a	b	c	d	e
x <sub>1</sub>	y <sub>5</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>3</sub>
x <sub>2</sub>	y <sub>5</sub>	y <sub>4</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>4</sub>

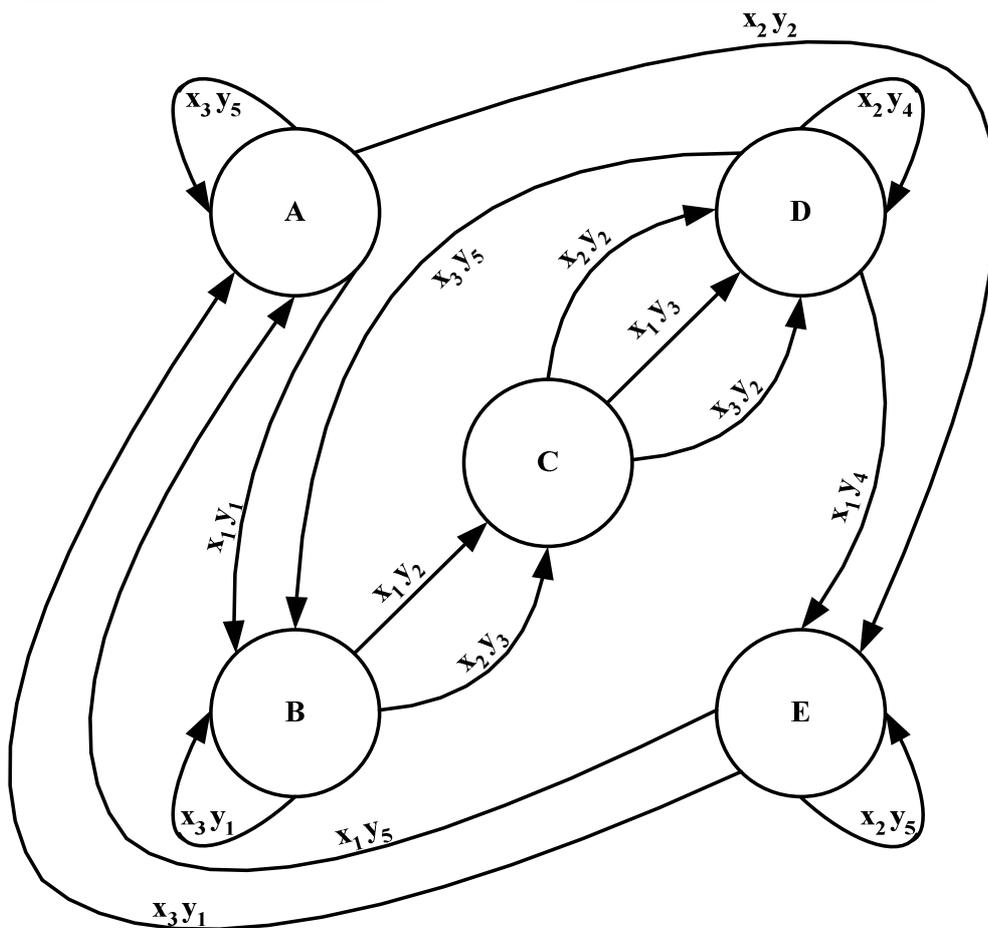


Рис. 13

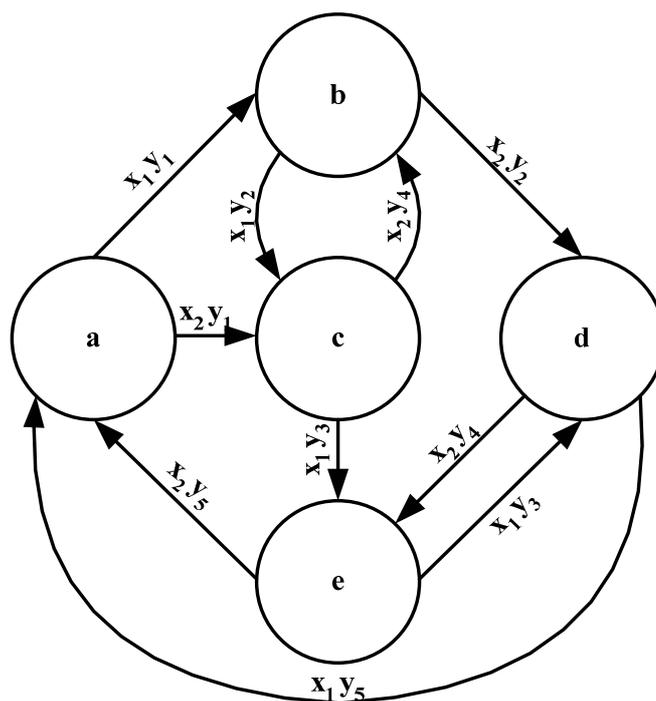


Рис. 14

В случае автомата Мура все стрелки, входящие в одну и ту же вершину  $a_k$ , должны быть обозначены одним и тем же выходным сигналом. Поэтому принято обозначать выходными сигналами не стрелки, а вершины, в которые эти ребра входят, т. е. на графе автомата Мура каждая вершина имеет два обозначения – одно, определяющее состояние автомата, и другое, обозначающее выходной сигнал.

Пример графа АКА Мура представлен на рис. 15, а соответствующая ему совмещенная матрица переходов и выходов – в таблице 9.

Таблица 9

Совмещенная матрица переходов и выходов

y	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>
a	1	2	3	4
x <sub>1</sub>	2	3	3	2
x <sub>2</sub>	1	4	4	1

ПРИМЕРЫ автоматных моделей дискретных устройств.



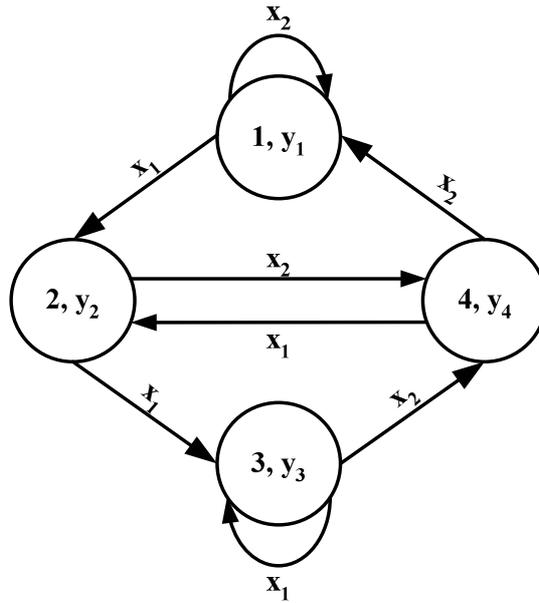


Рис. 15

Для моделирования элементов вычислительных систем и сетей, проявляющих статистически закономерное случайное поведение, можно использовать *вероятностные автоматы*. Вероятностный автомат определяется, в дополнение с семейству множеств  $X, Y, A$  конечного автомата, семейством матриц  $\{M(y/x)\}$ . Если  $X(x_1...x_l)$  – входной алфавит,  $Y(y_1...y_m)$  – выходной алфавит,  $A(a_1...a_n)$  – множество состояний, то  $M(y/x)$  – семейство  $l \times m$  матриц размерностью  $n \times n$ . Элемент  $\mu_{ij}(y_p/x_r)$  матрицы  $M(y_p/x_r)$  есть вероятность  $\mu_{ij}(y_p/x_r) = P(y_p, a_j/a_i, x_r)$ ,  $i, j = 1...n$ , того, что, находясь в состоянии  $a_i$  и получив входной сигнал  $x_i$ , автомат перейдет в состояние  $a_j$ , а выходной сигнал будет  $y_p$ . Если  $\mu_{ij}(y_p/x_r)$  принимает только значения единицы или нуля, имеем частный случай вероятностного автомата – обычный детерминистический конечный автомат [3, 6, 20].

### 3.1.2. Сети Петри

Понятие конечного автомата является удобной математической схемой для описания функционирования тех объектов и систем, для которых характерно наличие дискретных состояний и дискретный характер работы во времени. Однако для моделирования ряда объектов, например средств вычислительной техники и вычислительных сетей, использование абстрактно-автоматной математической схемы встречает трудности, несмотря на то, что данные объекты могут быть представлены как дискретные переходные системы. Это объясняется тем, что конечный автомат является *последовательной* алгоритмической схемой, а названные объекты могут вы-

полнять свои функции параллельно и асинхронно, независимо друг от друга, за исключением некоторых заданных точек, называемых *точками взаимодействия параллельных процессов*. Описание функционирования с использованием конечного автомата в таких случаях теоретически возможно, однако требует рассмотрения большого числа состояний и увеличения сложности модели. Данного недостатка лишены сети Петри.

*Сеть Петри* это ориентированный граф, содержащий позиции (вершины), определяющие условия, имеющиеся в системе, и переходы, отображающие связанные с этими условиями действия. В позициях проставляются метки, если соответствующее условие выполнено. Передвижение меток по сети определяет последовательность изменения состояний моделируемого объекта. Позиции изображаются кружками, переходы – планками. Позиции соединяют дугой с переходом, если выполнение заданного условия является необходимым для запуска связанного с данным переходом действия. Переход соединяют дугой с позицией, если связанное с ним действие порождает выполнение условия, представленного данной позицией. Соединение дугой позиции с позицией или перехода с переходом *не допускается*.

Сети Петри функционируют в непрерывном времени. Динамика функционирования определяется правилами срабатывания переходов. Изменение состояния сети связано с механизмом изменения маркировок позиций. В случае *простой временной* сети Петри:

- срабатывает только активный переход, т. е. такой, во всех входных позициях которого имеются метки;
- срабатывание перехода наступает через заданный конечный промежуток времени после его активизации, причем если возникает конфликт – одновременная активизация нескольких переходов, имеющих общие входные вершины, то срабатывает равновероятно только один из конфликтных переходов;
- в результате срабатывания перехода число меток в каждой входной позиции уменьшаются на единицу, а число меток во всех выходных позициях увеличиваются на единицу.

*ПРИМЕР смены маркировок простой сети Петри при разрешении конфликта.*



Кроме графического используют табличное описание сети Петри. Выделяют два типа таблиц: первая для каждой вершины сети задает список ее последователей, вторая определяет веса дуг сети. Число колонок таблицы определяет степень параллелизма модели.

Элементарный цикл обслуживания моделируется простой временной сетью Петри, представленной на рис. 16.

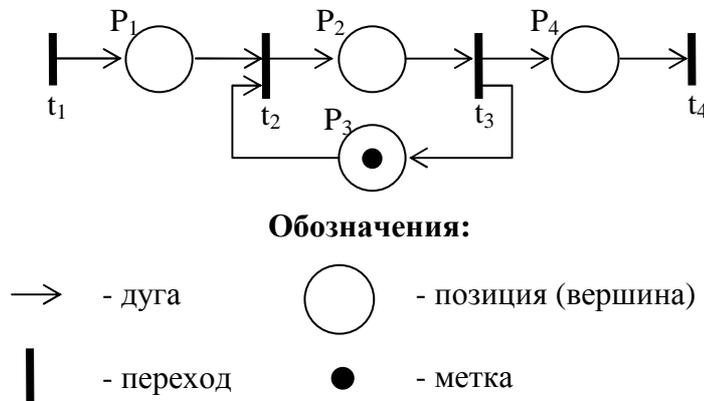


Рис. 16

С переходами на рис. 16 связаны времена выполнения следующих действий:  $t_1$  – поступление заявки на обслуживание во входную очередь;  $t_2$  – начало обслуживания;  $t_3$  – конец обслуживания;  $t_4$  – выход заявки из цикла обслуживания. Позиции этой сети Петри соответствуют условиям:  $P_1$  – наличие заявки, ожидающей обслуживания, во входной очереди;  $P_2$  – наличие заявки на обслуживании в процессоре;  $P_3$  – процессор свободен;  $P_4$  – наличие обслуженной заявки в выходной очереди. Маркировка сети Петри, показанной на рис. 16, соответствует начальному состоянию системы обслуживания: заявок, ожидающих обслуживание, во входной очереди нет, и процессор свободен (вершина  $P_3$  содержит метку).

Другая маркировка сети Петри, моделирующей элементарный цикл обслуживания, показана на рис. 17.

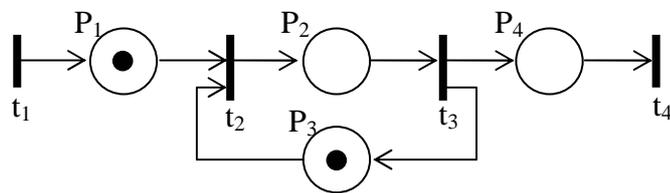


Рис. 17

Маркировка, показанная на рис. 17, соответствует следующему состоянию: заявка ожидает обслуживания и процессор свободен: вершины  $P_1$  и  $P_3$  содержат метки, и, следовательно, переход  $t_2$  активизирован. Моделирование дальнейших событий в пошаговом режиме показано на рис. 18.

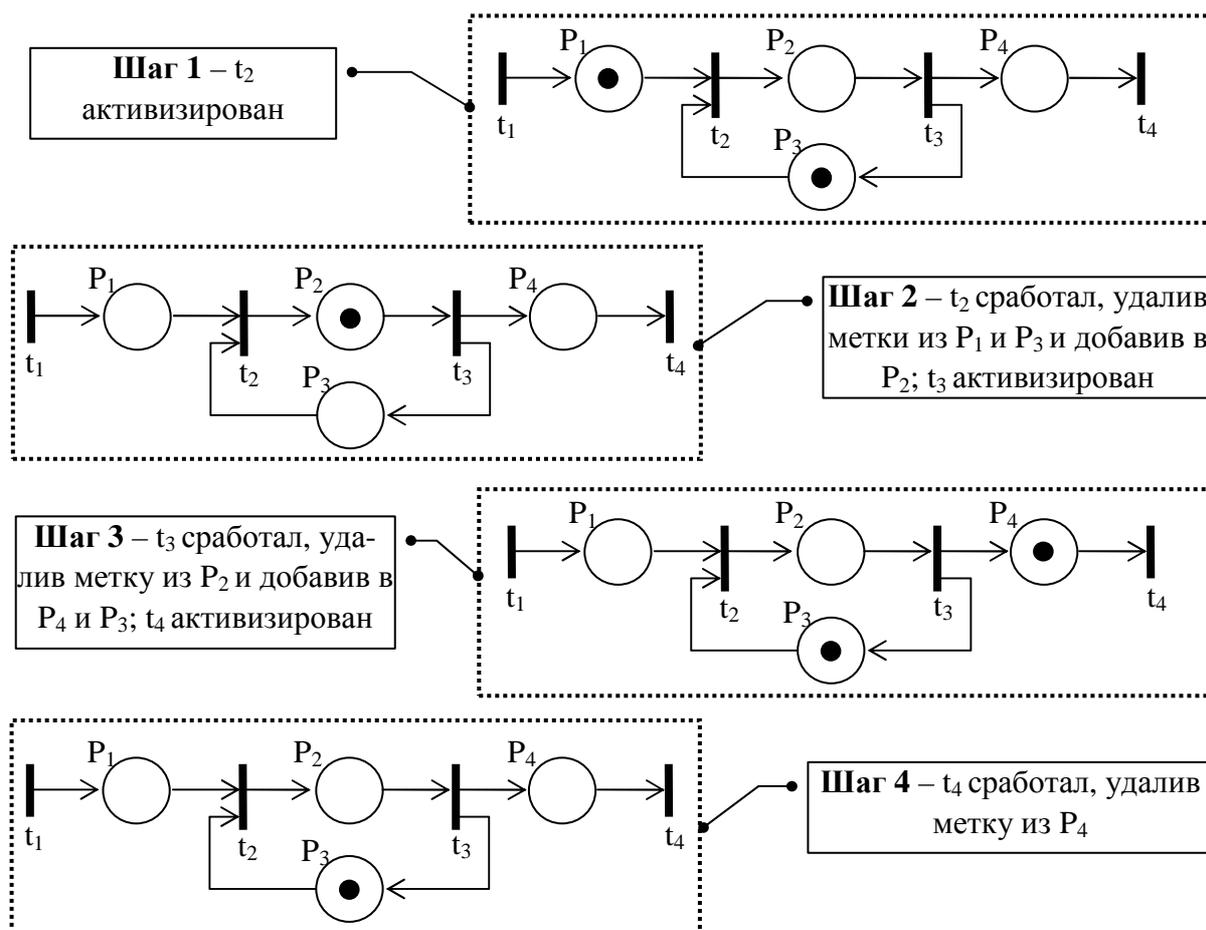


Рис. 18

После срабатывания перехода  $t_2$  метки из его входных вершин  $P_1$  и  $P_3$  будут изъяты, а в выходной вершине  $P_2$  метка появится, что соответствует наличию заявки на обслуживание в процессоре. Метка в вершине  $P_2$  активизирует переход  $t_3$ , который сработает по истечению времени обслуживания. В результате срабатывания перехода  $t_3$  метка изымается из вершины  $P_2$ , а в вершинах  $P_4$  и  $P_3$  метки появляются. Срабатывание перехода  $t_4$ , соответствующее удалению обслуженной заявки из цикла, изымет метку из вершины  $P_4$ , и система вернется к начальной маркировке (см. рис. 16).

В общем случае в позиции может быть более одной метки. Тогда, для срабатывания перехода вида  $n/m$  требуется наличие во входных позициях суммарного количества меток не менее  $n$ . При срабатывании перехода из всех входных позиций по числу исходящих дуг удаляются в сумме  $n$  меток, а во всех выходных позициях по числу выходных дуг появляются  $m$  меток (см. рис. 19).

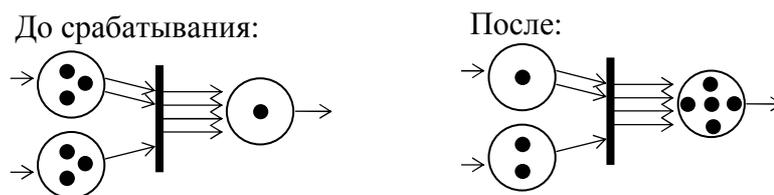


Рис. 19

Для удобства в графическом представлении сетей Петри и в программном обеспечении моделирования, как правило, при задании переходов типа  $n/m$  вместо множественных дуг используют одну, задавая ее вес. Также из соображений удобства количество меток в вершине более 2-х можно указывать числом вместо множества меток. В таком представлении пример, приведенный на рис. 19, будет иметь вид, показанный на рис. 20.

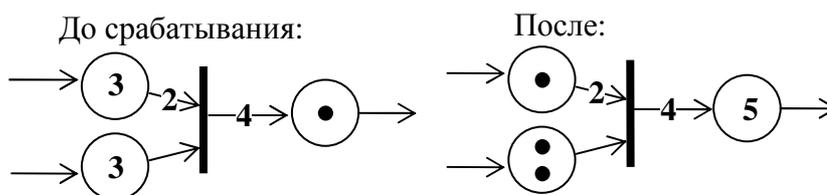


Рис. 20

Переходы типа  $n/m$  используются для моделирования процессов, сопровождаемых изменением типа моделируемого ресурса. Например, для моделирования процессов сборки в технологической системе, процессов пакетирования заявок в вычислительной сети (в этом случае  $n > m$ ) или процесса разделения материальных потоков, процесса создания копий заявок (в этом случае  $n < m$ ).

*ПРИМЕРЫ сетей Петри для моделирования процесса пакетирования заявок и синхронного моделирования элемента И-НЕ.*



Для моделирования средств вычислительной техники и процессов обработки информации используются разновидности сетей Петри, различающиеся способами разрешения конфликтов. В *стохастических* сетях Петри дополнительно вводятся случайные задержки или вероятности срабатывания активных переходов. В примере на рис. 21-а сработает либо переход  $t_1$  (с вероятностью  $p_1$ ), либо  $t_2$  (с вероятностью  $1-p_1$ ). В *приоритетных* сетях конфликтные ситуации разрешаются введением различных при-

оритетов для ветвей. Конфликт, показанный в примере на рис. 21-б, всегда будет разрешаться в пользу перехода  $t_1$ , так как он имеет приоритет, а переход  $t_2$  сможет сработать только в том случае, если, при наличии меток в вершинах  $P_2$  и  $P_3$ , метки в вершине  $P_1$  не окажется.

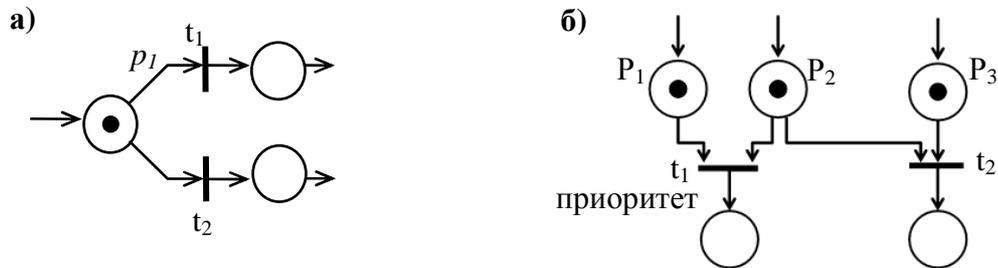


Рис. 21

Особой разновидностью сетей Петри являются *ингибиторные* сети, которые в дополнение к обычным дугам (ветвям) графа сети содержат запрещающие, так называемые ингибиторные ветви. Такая ветвь запрещает активацию перехода при наличии достаточного количества меток во входных вершинах обычных дуг до тех пор, пока в ее входной вершине имеются метки. Во фрагменте сети Петри, приведенном на рис.22-а, ветвь  $a$  запрещает запуск перехода  $t_1$  при наличии метки в позиции  $P_1$ . Пример реализации простейшего цикла обслуживания с использованием ингибиторной сети Петри представлен на рис.22-б. Здесь переход  $t_2$  при наличии метки в позиции  $P_2$  будет «заперт» не смотря на наличия метки в вершине  $P_1$  до тех пор, пока метка не покинет  $P_2$  через переход  $t_3$ , что эквивалентно завершению очередного обслуживания.

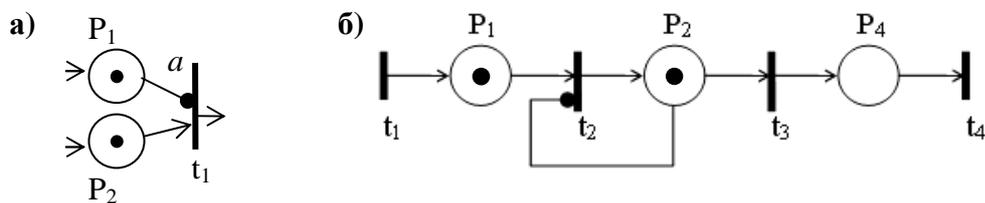


Рис. 22

Рассмотрим примеры моделирования сетями Петри различных процессов и систем.

*Сеть Петри для моделирования магистрального канала передачи данных.* Пусть к общему каналу связи подключены  $N$  абонентов и возмож-

на связь любых абонентов друг с другом. Абонент-отправитель осуществляет попытку связи в случайный момент времени  $T_1$ . Если канал занят передачей информации от другого абонента, это обнаруживается по наличию сигналов несущей частоты в канале связи. Абонент задерживает передачу на время  $\tau_1$ , являющееся реализацией равномерно распределенной в заданном диапазоне случайной величины  $\tau$ . Если в момент времени  $(T_1 + \tau_1)$  канал связи опять занят, то передача задерживается по тому же правилу. Если два абонента или более пытаются начать передачу одновременно, возможны конфликты. Одновременность описывается условием  $\Delta T < \varepsilon$ , где  $\Delta T$  – промежуток времени между моментами начала передачи данных различными абонентами,  $\varepsilon > 0$ . При конфликте передача начинается, но передаются искаженные данные. Ликвидация конфликта заключается в том, что все абоненты, начавшие одновременно передачу данных, прекращают ее и пытаются начать работу через промежуток времени, индивидуальный для каждого абонента и являющийся функцией  $\tau$ .

В модельной реализации (см. рис. 23) источник (открытый переход  $t_2$ ) имитирует поток заявок на передачу от всех абонентов. Если канал свободен и конфликта нет, заявка проходит через  $t_3, t_6, t_7, t_{10}, t_{11}$  и выходит из системы обслуженной, причем в  $t_6$  происходит задержка на время  $\varepsilon$ , а в  $t_{10}$  – на время  $(T_n - \varepsilon)$ , где  $T_n$  – время передачи пакета.

Если канал занят (заявка задержана в  $t_{10}$ ), то попытка другого абонента начать передачу приводит к прохождению заявки по маршруту  $t_3, t_6, t_9$ , и далее в один из переходов  $t_{12}..t_n$ . Срабатывание перехода  $t_9$ , а не  $t_7$ , происходит потому, что предыдущая заявка, прошедшая через  $t_7$  и еще не вышедшая из  $t_{10}$ , изъяла метку из позиции  $p_9$ . Тем самым переход  $t_7$  оказался запрещенным, а  $t_9$  разрешенным.

Переходы  $t_{12}..t_n$  моделируют задержку пакетов на время  $\tau_i$ . Через время  $\tau_i$  заявка переходит к  $p_3$ , т. е. предпринимается новая попытка передачи сообщения. Конфликты возникают, если новая заявка приходит в позицию  $p_3$ , когда предыдущая еще не покинула переход  $t_6$ . Поэтому метка не может пройти переход  $t_3$ , но может пройти через переход  $t_4$  в позицию  $p_6$ . Теперь вышедшая из  $t_6$  заявка сможет пройти через  $t_8$  на переходы  $t_{12}..t_n$ , где обе заявки будут задержаны на случайные отрезки времени перед повторными попытками передачи. Чтобы метка из  $p_8$  перешла в  $t_8$ , а не в  $t_9$ , ветви, ведущей в  $t_8$ , присваивается более высокий приоритет. Переход  $t_5$  срабатывает в случае, если в конфликт вошло более двух заявок [3, 20 – 23].

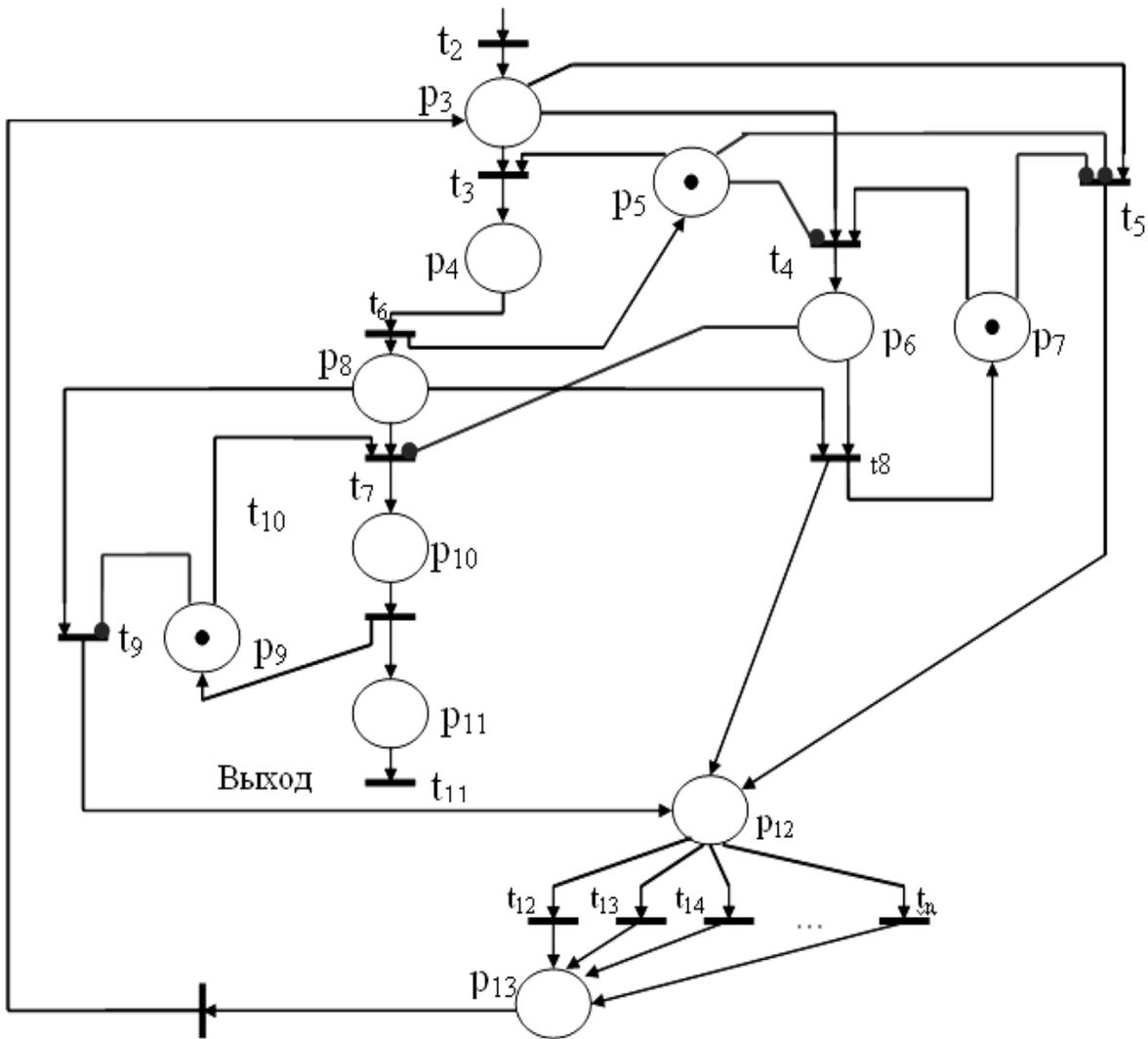


Рис. 23

*ПРИМЕРЫ сетей Петри различных систем и процессов и их реализации в программе IngProject – см. Лабораторную работу №7*



*Сеть Петри для моделирования процессов возникновения отказов и устранения неисправностей в технологической системе с запасным блоком и дополнительным резервом. Пусть в технологической системе эксплуатируются три рабочих блока. С некоторой периодичностью в любом из рабочих блоков может наступить отказ. Для замены отказавшего блока изначально имеется один запасной. При наступлении отказа неисправный блок заменяется запасным, после чего неисправный блок подвергают восстановлению (ремонту) и он становится запасным. В системе кроме запас-*

ного блока имеется еще один, резервный. Его использование разрешено только в случае отказа всех трех основных рабочих блоков. Если резервный блок использован – он должен быть восстановлен в первую очередь. Все временные характеристики системы (время отказа, время замены, время восстановления) являются случайными величинами.

При функционировании данной системы могут иметь место следующие ситуации:

- все три рабочих блока исправны, запасной и резервный блоки готовы к использованию;
- один из рабочих блоков отказал и запасной готов к использованию (в этом случае будет произведена замена и отказавший блок будет отправлен на восстановление для последующей передачи в запас);
- если запасной блок не готов к использованию, т. е. находится на восстановлении, возможны ситуации последовательного наступления отказов одного и второго рабочих блоков без замены;
- если запасной блок не готов к использованию, т. е. находится на восстановлении, а отказывают последовательно все три рабочих блока, то для замены будет использован резервный блок;
- если резервный блок уже использован, то после завершения восстановления очередного отказавшего блока он будет передан не в запас, а в резерв, так как восстановление резерва выполняется в первую очередь;
- если резервный блок уже восстановлен, то очередной отказавший блок будет отправлен на восстановление и, далее, в запас.

Заданную систему моделирует ингибиторная приоритетная сеть Петри, представленная на рис. 24.

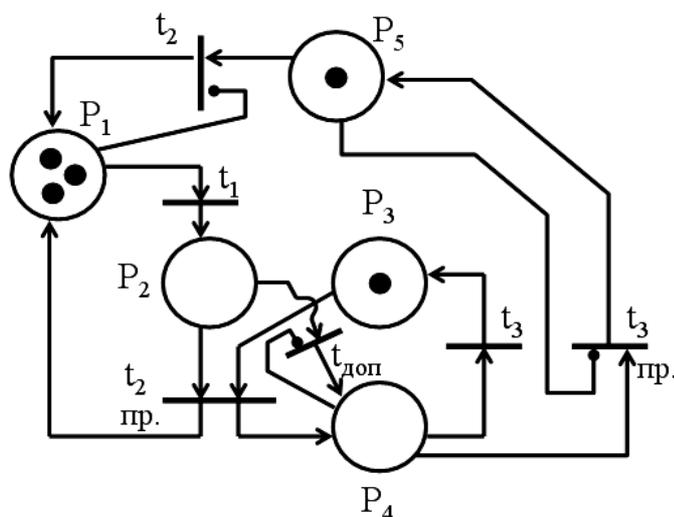


Рис. 24

В таблице 10 представлено описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны, т. е. выполняются при наличии в вершине метки-маркера.

Таблица 10

Вершина	Описание
$P_1$	Наличие в системе исправных рабочих блоков (количество меток соответствует количеству блоков).
$P_2$	Наличие в системе отказавших рабочих блоков (количество меток соответствует количеству блоков).
$P_3$	Наличие в системе исправного, готового к использованию для замены запасного блока.
$P_4$	Наличие в системе неисправного, нуждающегося в восстановлении блока
$P_5$	Наличие в системе исправного, готового к использованию для замены резервного блока.

Начальная маркировка сети, представленная на рис. 24, соответствует наличию в системе трех исправных рабочих блоков; исправного, готового к использованию для замены запасного блока и исправного, готового к использованию для замены резервного блока. Описание всех переходов с точки зрения связанных с ними событий представлено в таблице 11.

*ПРИМЕР сети Петри для моделирования процессов возникновения отказов и устранения неисправностей в технологической системе с дополнительным резервом (второй вариант).*



Таблица 11

Переход	Описание
$t_1$	Наступление отказа (время случайно).
$t_2$	Выполнение замены неисправного блока на запасной (резервный) (время случайно).
$t_3$	Выполнение восстановления неисправного блока (время случайно). Два идентичных перехода реализованы для моделирования восстановления запасного и резервного блоков. Так как при необходимости восстановление резервного блока должно осуществляться в первую очередь, соответствующий переход имеет больший приоритет (помечен на рис. 24 «пр.») и в случае конфликта сработает именно он.
$t_{доп}$	Отправление на восстановление одного из отказавших блоков без замены. Данный переход необходим, так как в случае первоочередного восстановления резерва метка в вершине $P_3$ отсутствует и переход $t_2$ сработать не может. В нормальном режиме работы, когда в системе имеется запасной блок (метка в вершине $P_3$ ) для разрешения конфликта переходу $t_2$ присвоен больший приоритет (помечен на рис. 24 «пр.»).

### 3.2. Модели массового обслуживания

В теории массового обслуживания изучаются системы, на вход которых поступает случайный поток заявок (требований), приходящихся в общем случае на случайные моменты времени. Поступившая заявка обслуживается в системе путем предоставления ей некоторых ресурсов на какое-то время и, будучи в той или иной мере обслуженной, покидает систему.

Помимо случайного появления заявок на обслуживание и случайной длительности обслуживания каждой заявки для систем массового обслуживания характерным является наличие очередей, в которых заявки ждут момента освобождения ресурсов, занятых обслуживанием других заявок.

Поскольку события, происходящие в ВС, носят случайный характер, то для их изучения наиболее подходящими являются вероятностные математические модели теории массового обслуживания. Так, используемые в настоящее время в локальных сетях протоколы канального уровня используют методы доступа к среде, основанные на ее совместном использовании несколькими узлами за счет разделения во времени. В этом случае, как и во всех случаях разделения ресурсов со случайным потоком запросов, могут возникать очереди.

Объектами исследования в теории массового обслуживания являются системы и сети массового обслуживания (СМО). В системах, моделируемых в виде СМО, различают *статические* и *динамические* объекты. Статические объекты – обслуживающие аппараты (ОА) или ресурсы, моделируют средства обработки информации (аппаратные и программные). Динамические объекты – заявки (запросы, требования) моделируют решаемые в ВС задачи. Поток заявок физически представляет собой явления одной природы, например попытки модемного соединения, запросы к базе данных и т. д. С математической точки зрения поток заявок на обслуживание характеризуется законом распределения случайной величины – времени между появлением соседних заявок.

Функционирование СМО представляется как процесс прохождения заявок через систему. Правило, по которому заявки поступают из очередей на обслуживание в ОА, называется *дисциплиной обслуживания*, а величина преимущественного права на обслуживания – *приоритетом*. Для каждого приоритета на входе ОА образуется своя очередь. Если заявка поступает на вход ОА, занятого обслуживанием заявки с более низким приоритетом, то возможно прерывание ранее начатого обслуживания – такой приоритет на-

зывается абсолютным. Если прерывания ранее начатого обслуживания не происходит – приоритет относительный.

СМО бывают одно- и многоканальными в зависимости от числа ОА, параллельно обрабатывающих входной поток заявок; одно- и многофазными в зависимости от числа последовательно включенных ОА.

Классификационное обозначение СМО имеет вид  $A/B/C/D/E$ , где позиции, обозначенные буквами, означают следующие характеристики:

$A$  – обозначение закона распределения времени поступления заявок входного потока (обозначение  $M$  соответствует экспоненциальному закону распределения,  $\Gamma$  – гамма-распределению,  $E$  – распределению Эрланга,  $H$  – гиперэкспоненциальному распределению,  $N$  – нормальному распределению,  $R$  – равномерному распределению,  $D$  – постоянному времени обслуживания,  $G$  – произвольному или неизвестному закону распределения,  $Gr$  – групповому (пакетному) поступлению заявок на обслуживание);

$B$  – обозначение закона распределения времени обслуживания в устройствах (используются те же обозначения, что и для распределения времени поступления заявок);

$C$  – число ОА;

$D$  – число мест в очереди (для неограниченных опускается);

$E$  – дисциплина обслуживания: для дисциплины FIFO данное обозначение опускается; также используются обозначения LIFO, RANDOM, SF (Short Forward – «короткие вперед» – в первую очередь обслуживаются те заявки из очереди, которые требуют меньшего времени обслуживания).

Примеры обозначений:

–  $M/M/1$ : СМО с одним ОА, бесконечной очередью, экспоненциальными законами распределения интервалов времени между поступлениями заявок и времени обслуживания, дисциплиной обслуживания FIFO;

–  $E/H/m/r/LIFO$ : СМО с  $m$  обслуживающими аппаратами, очередью, ограниченной  $r$  местами, эрланговским законом распределения интервалов между поступлениями заявок, гиперэкспоненциальным распределением времени обслуживания в ОА, дисциплиной обслуживания LIFO.

Если СМО в дополнение к перечисленным характеристикам обладает какими-либо особенностями, последние добавляются к обозначению в качестве комментария (например, СМО типа  $G/G/1$  с ненадежным ОА и временем ожидания в очереди, ограниченным 3,5 секундами).

Для моделирования ВС наиболее часто используются комбинации типов СМО, приведенные в таблице 12.

Таблица 12

Наименование	Обозначение	Графическое обозначение	Описание
Одноканальная СМО с ожиданием	$G/G/1$		Один ОА с бесконечной очередью. С той или иной долей приближения моделирует любой узел или процесс ВС, например механизм разделения среды протокола Ethernet.
Одноканальная СМО с потерями	$G/G/1/r$		Один ОА с конечным числом мест в очереди. Если число заявок превышает число мест в очереди, то лишние заявки теряются. Используется при моделировании каналов передачи в ВС.
Многоканальная СМО с ожиданием	$G/G/m$		Несколько параллельно работающих ОА с общей бесконечной очередью. Используется при моделировании групп абонентских терминалов ВС, работающих в диалоговом режиме.
Многоканальная СМО с потерями	$G/G/m/r$		Несколько параллельно работающих ОА с общей очередью, число мест в которой ограничено. Используются для моделирования каналов связи в ВС.
Одноканальная СМО с групповым поступлением заявок	$Gr/G/1$		Один ОА с бесконечной очередью. Перед обслуживанием заявки группируются в пакеты по определенному правилу. Используется для моделирования узлов коммутации.
Одноканальная СМО с групповым обслуживанием заявок	$G/Gr/1$		Один ОА с бесконечной очередью. Заявки обслуживаются пакетами, составляемыми по определенному правилу. Используется для моделирования узлов коммутации.

Целью моделирования СМО является определение *статистических* и *операционных* характеристик, определяющих поведение систем в процессе функционирования. Вероятность потери заявки (вероятность отказа) одна из основных статистических характеристик СМО. Помимо этого по результатам функционирования СМО определяют следующие характеристики: средняя длина очереди, коэффициент загрузки ОА (доля времени, в течение которого ОА занят обслуживанием), среднее время ожидания заявки в очередях СМО, среднее время обслуживания заявки в ОА СМО, среднее время пребывания заявки в СМО и т. д.

К основным операционным характеристикам относятся:

$Q(t)$  – длина очереди в момент времени  $t$ , т. е. число заявок, ожидающих обслуживания с учетом или без тех заявок, обслуживание которых уже началось;

$Q_n$  – длина очереди на  $n$ -й стадии, при этом предполагается, что стадии реализуются в дискретном режиме и определяются теми или иными событиями (например, появлением запроса на обслуживание, или выбытием заявки из системы);

$W(t)$  – виртуальная продолжительность ожидания относительно момента времени  $t$ , т. е. время ожидания обслуживания для заявки, которое поступит в систему в момент времени  $t$ ;

$W_n$  – продолжительность периода, в течение которого  $n$ -я заявка ожидает обслуживания;

$T_i$  – продолжительность периода занятости системы, начало которого соответствует  $Q(0)=i$ , т. е. длина периода занятости системы, начинающегося при наличии в системе  $i$  заявок;

$I_n$  – продолжительность  $n$ -го периода простоя системы, т. е. длина интервала, в течении которого система в  $n$ -й раз оказывается незанятой.

Наряду с указанными характеристиками используются их различные модификации – полная продолжительность пребывания запроса в системе, операционный цикл – сумма продолжительности периода занятости и непосредственно следующего за ним периода простоя, суммарное полезное время (доля времени с полной загрузкой), и т. д.

Многие системы массового обслуживания обладают тем свойством, что по истечении определенного времени их поведение в некотором смысле стабилизируется. Формально это выражается в появлении стационарных (периодических) свойств процессов  $Q(t)$  и  $W(t)$  при  $t \rightarrow \infty$ , и  $Q_n$ ,  $W_n$  при  $n \rightarrow \infty$ . Условия, при которых системы переходят в стационарное состояние, представляют отдельный интерес для исследования.

В соответствии с общей классификацией моделей различают аналитические и имитационные модели СМО [5, 24-27].

### 3.3. Аналитическое моделирование систем массового обслуживания

*Аналитические теоретические модели СМО* представляют собой совокупность явных зависимостей параметров, образующих вектор фазовых переменных СМО  $V$ , от векторов внутренних  $Q$  и внешних  $X$  параметров:

$$\overline{V}(t) = F(\overline{X}(t), \overline{Q}(t)).$$

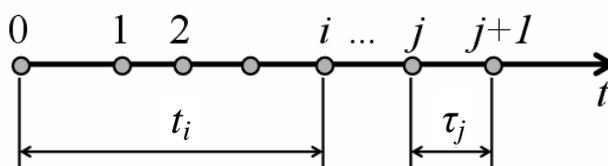
Вектор  $Q$  составляют параметры ОА, вектор  $X$  – параметры входных потоков заявок. Аналитические модели СМО можно получить только в частных случаях со следующими ограничениями:

1. Входные потоки заявок должны обладать свойствами стационарности, ординарности и отсутствия последействия – такие потоки называ-

ются *простейшими*. В подавляющем большинстве работ по теории МО рассматривается простейший поток, для которого вероятность поступления в промежуток времени  $t$  ровно  $k$  заявок задается формулой:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad (9)$$

где  $\lambda$  – *интенсивность* поступления заявок (параметр экспоненциального распределения), положительная постоянная величина, обратная МО времени поступления. На оси времени поток поступления заявок можно изобразить, как показано на рис. 25.



$\tau_j$  – интервал между поступлениями заявок (случайная величина);  
 $t_i$  – момент поступления  $i$ -той заявки (отсчитывается от  $t=0$ ).

Рис. 25

Свойство *стационарности* заключается в том, что вероятность поступления определенного числа заявок в интервале времени  $\Delta t$  зависит только от длительности этого интервала и не зависит от положения этого интервала на оси времени (см. рис. 26). Иначе говоря, вероятностные характеристики и интенсивность такого потока со временем не изменяются.

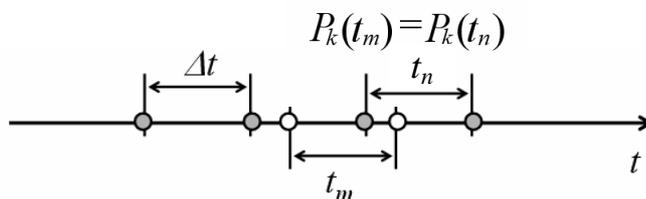


Рис. 26

*Ординарность* означает невозможность одновременного поступления двух и более заявок на вход системы. *Отсутствие последовательности* выражается в том, что вероятности разных непересекающихся интервалов не зависят друг от друга. Иногда это свойство формулируют следующим об-

разом: распределение времени до ближайшего события не зависит от времени наблюдения, т. е. от того, сколько времени прошло после последнего события. Отсутствие последствия в потоке означает, что события, образующие поток появляются в последовательные моменты времени независимо друг от друга.

2. Интервалы времени между поступлениями заявок и времена обслуживания заявок в ОА СМО распределены по экспоненциальному закону, т. е. функция распределения вероятностей и функция плотности вероятностей для этих интервалов времени имеют вид:

$$F(t) = 1 - e^{-\lambda t},$$

$$f(t) = \lambda e^{-\lambda t}.$$

3. Приоритетность обслуживания не рассматривается, используются дисциплины обслуживания типа FIFO.

Как уже отмечалось, вероятность поступления в промежуток времени  $t$  ровно  $k$  заявок для простейшего потока определяется формулой (9). Вероятность того, что в течение интервала времени  $t$  не поступит ни одной заявки (эквивалентно вероятности того, что следующая после последней заявки поступит в СМО по истечении времени  $t$ ) составляет:

$$P_0(t) = e^{-\lambda t}.$$

Вероятность, что за то же время поступит хотя бы одна заявка:

$$P_{\geq 1}(t) = 1 - e^{-\lambda t}.$$

При построении и анализе непрерывно – стохастических моделей исследуются вероятности появления или не появления событий в течении элементарного интервала времени  $\Delta t \rightarrow 0$ . Произведя замену  $t$  на  $\Delta t$ , и, разлагая  $e^{-\lambda \Delta t}$  в степенной ряд, получим, пренебрегая величинами высшего порядка малости:

$$P_0(\Delta t) = e^{-\lambda \Delta t} = 1 - \lambda \Delta t + \frac{1}{2!} (\lambda \Delta t)^2 - \dots = 1 - \lambda \Delta t + 0 \Delta t \cong 1 - \lambda \Delta t.$$

Тогда, вероятность появления хотя бы одной заявки:

$$P_{\geq 1}(\Delta t) = 1 - P_0(\Delta t) = \lambda \Delta t.$$

Учитывая ординарность простейшего потока, можно утверждать, что последнее выражение представляет собой вероятность появления ровно одной заявки.

Простейший поток обладает устойчивостью, состоящей в том, что при суммировании независимых простейших потоков получается снова простейший поток, причем интенсивности складываемых потоков суммируются. Кроме этого для простейшего потока характерно, что поступление заявок через короткие промежутки времени более вероятно, чем через длинные – 63% промежутков времени между заявками имеют длину, меньшую математического ожидания времени поступления ( $1/\lambda$ ). Следовательно, моделирование процесса поступления или обслуживания заявок с использованием экспоненциального распределения является предположением о том, что значительная доля заявок поступает или обслуживается в системе в течение кратковременного интервала, меньшего МО. Однако в реальной системе реальные характеристики обслуживания могут не совпадать с характеристиками простейшего потока – так, например, часто выполнение некоторой определенной операции в ВС не может занимать время менее некоторого интервала.

С учетом рассмотренных ограничений множество практических задач анализа ВС не удастся решать в полном объеме с использованием аналитических моделей СМО. Их применяют для ориентировочных оценок свойств проектируемых систем на начальных стадиях проектирования и в качестве макромоделей отдельных фрагментов ВС при их имитационном моделировании [4, 5, 24-26]. Рассмотрим постановку, ограничения и подходы к решению типовых задач аналитического моделирования СМО.

### **3.3.1. Обслуживание с ожиданием без потерь**

В классической постановке задача формулируется следующим образом: в СМО типа  $M/M/t$  на  $t$  одинаковых ОА поступает простейший поток заявок с интенсивностью  $\lambda$ . Если в момент поступления заявки имеется хотя бы один свободный ОА, она немедленно начинает обслуживаться, если нет – становится в очередь. Длительность обслуживания также распределена по экспоненциальному закону, т. е. представляет собой случайную величину с одним и тем же распределением вероятностей  $F(x)=1-e^{-\mu x}$ , где  $\mu$  – интенсивность обслуживания (величина, обратная МО времени обслуживания).

Выбор экспоненциального распределения в аналитическом моделировании СМО для описания длительности процесса не случаен. В этом предположении задача допускает простое решение, которое с удовлетворительной точностью описывает ход рассматриваемого процесса. Данное распределение играет в теории массового обслуживания важную роль, которая в значительной мере вызвана следующим свойством: при показа-

тельном распределении длительности обслуживания распределение длительности оставшейся части работы по обслуживанию не зависит от того, сколько оно уже продолжалось.

Действительно, пусть  $P_a(t)$  означает вероятность того, что обслуживание, которое уже продолжалось время  $a$ , продлится еще не менее чем  $t$ .

*ПРИМЕР – исследование свойств экспоненциального распределения.*



В предположении, что длительность обслуживания распределена показательно, вероятность того, что обслуживание продлится от нулевого момента времени до момента времени  $t$  составляет  $P_0(t)=e^{-\mu t}$ . Далее ясно, что  $P_0(a)=e^{-\mu a}$  и  $P_0(a+t)=e^{-\mu(a+t)}$ . А так как всегда  $P_0(a+t)=P_0(a)P_a(t)$ , то  $e^{-\mu(a+t)}=e^{-\mu a}P_a(t)$  и, следовательно,  $P_a(t)=e^{-\mu t}=P_0(t)$ .

Как уже отмечалось, в реальной обстановке показательное время обслуживания является, как правило, приближением, так как предположение, что  $F(x)=1-e^{-\mu x}$  приводит к тому, что значительная доля заявок нуждается лишь в кратковременной операции, близкой к 0. Необходимость снятия этого ограничения решается *распределением Эрланга (гамма-распределение с целочисленным параметром)*, плотность распределения которого задается формулой:

$$\varphi_k(t) = \mu \frac{(\mu t)^{b-1}}{(b-1)!} e^{-\mu t},$$

где  $\mu > 0$ , а  $b$  – целое положительное число. Распределение Эрланга представляет собой распределение суммы  $b$  независимых слагаемых, каждое из которых имеет распределение  $F(x)=1-e^{-\mu x}$ .

В каждый момент многоканальная система обслуживания с ожиданием  $M/M/m$  может находиться в одном из следующих состояний: в момент времени  $t$  в системе находится  $k$  заявок. Если  $k \leq m$ , то в системе находятся и обслуживаются  $k$  заявок, а  $m-k$  ОА свободны. Если  $k > m$ , то  $m$  заявок обслуживаются, а  $k-m$  находятся в очереди и ожидают обслуживания. Если  $E_k$  – состояние, когда в системе находятся  $k$  заявок, система может находиться в состояниях  $E_0, E_1, E_2, \dots$  и т. д. Пусть в некоторый момент времени  $t_0$  система находилась в состоянии  $E_i$ . Дальнейшее течение обслуживания полностью определяется тремя следующими факторами:

- моментами завершения обслуживаний, производящихся в момент времени  $t_0$ , т. е. начатых ранее  $t_0$ ;

- моментами поступления новых заявок;
- длительностью обслуживания заявок, поступивших после  $t_0$ .

В силу рассмотренных особенностей показательного распределения длительность остающейся части обслуживания не зависит от того, как долго уже продолжалось обслуживание до момента времени  $t_0$ . Поток заявок является простейшим, и длительность обслуживания заявок, поступивших после  $t_0$ , никак не зависит от того, что и как обслуживалось до момента  $t_0$ . Таким образом, последующее течение процесса обслуживания не зависит в вероятностном смысле от того, что происходило до момента времени  $t_0$ . Система с ожиданием в случае простейшего потока и показательного времени обслуживания представляет случайный процесс Маркова или процесс без последствия – случайный процесс, для которого будущее развитие зависит только от достигнутого в данный момент состояния и не зависит от того, как происходило развитие в прошлом. Аналитическое моделирование СМО применимо только к Марковским процессам и системам.

В задаче обслуживания с ожиданием очереди многоканальной СМО представляются неограниченными, и рассматривается обслуживание без потерь. В реальных ВС потери могут иметь место. Аналитическими методами теории массового обслуживания решаются задачи моделирования СМО с потерями, которые могут иметь место по причине:

- ограничения количества мест в очереди;
- ограничения времени ожидания;
- ограничения времени пребывания;
- приоритетного обслуживания [4, 5, 27-28].

### 3.3.2. Обслуживание с ограниченным временем ожидания

Постановка задачи обслуживания с ограничением времени ожидания совпадает с постановкой задачи обслуживания с ожиданием без потерь с тем лишь отличием, что время ожидания обслуживания заявок в очередях  $m$ -канальной системы ожидание ограничено определенным временем  $\tau$ . Если заявка за время  $\tau$  со времени его поступления не начало обслуживаться, то оно теряется.

В случае  $\tau = \text{const}$  аналитически описывать состояние этой системы через число заявок, находящихся в системе в данный момент времени уже невозможно – эта характеристика при такой постановке задачи теряет Марковские свойства. Если известно, что в момент времени  $t$  в системе обслуживания находится  $k$  заявок, то состояние в момент времени  $t+h$ , при любом  $h>0$ , зависит не только от  $k$  и  $t$ , но и от того, как долго ждут заявки,

поступившие до момента  $t$ . Следовательно, для обеспечения возможности использовать для моделирования аналитический аппарат моделирования теории массового обслуживания, необходимо ввести в рассмотрение некую характеристику состояния заданной СМО, которая обладала бы Марковскими свойствами.

Рассмотрим  $m$ -мерный случайный процесс  $\xi(t) = \{\xi_1(t), \xi_2(t) \dots \xi_m(t)\}$ , где  $\xi_i(t)$  – время, которое должно протечь от момента  $t$  до освобождения ОА с номером  $i$  от обслуживания заявок, поступивших ранее  $t$ . Если в момент времени  $t$  ОА с номером  $i$  свободен и в системе нет заявок, ожидающих обслуживания, то  $\xi_i(t) = 0$ .

Вектор  $\xi(t)$  со временем изменяется следующим образом: все компоненты, отличные от нуля, уменьшаются на длину промежутка времени, протекшего с момента  $t$ , если только не появилось новая заявка или же какая-либо из указанных разностей не стала меньше нуля. Если же в момент времени  $t_1 > t$  какая-нибудь разность обратилась в 0, то  $\xi(t_1) = 0$ .

Заявка, поступившее в момент  $t$ , выбирает ОА с номером  $i$  только тогда, когда  $\xi_i(t - 0) = \min_k \xi_k(t - 0)$ .

Пусть на  $i$ -й ОА заявки поступают в моменты времени  $t_{i1}, t_{i2} \dots$ , а необходимые для их обслуживания длительности времени равны соответственно  $\eta_{i1}, \eta_{i2} \dots$ . Пусть в момент  $t=0$  ОА с номером  $i$  был свободен. Функция  $\xi_i(t)$  до момента  $t_{i1}$  равна 0, в момент  $t_{i1}$  совершает скачек, равный  $\eta_{i1}$ , далее, она убывает на длительность протекшего промежутка времени до тех пор, пока разность положительна, или же до очередного момента поступления заявки. Если в момент поступления новой заявки  $\xi_i(t)$ , была меньше чем  $\tau$ , то в этот момент она возрастает скачком на соответствующую величину  $\eta$ . Если же  $\xi_i(t) > \tau$ , то поступившая в этот момент заявка теряет, получает отказ (см. рис. 27).

На рис. 27 в моменты времени  $t_i$  в СМО с ограниченным временем ожидания поступали заявки на обслуживание. Заявка, поступившая в момент времени  $t_{i2}$ , была потеряна, поскольку в момент ее поступления  $\xi_i(t) > \tau$ .

Из приведенного описания видно, что состояние процесса  $\xi(t)$  в момент времени  $t+h$ , при любом  $h > 0$  полностью определяется его состоянием в момент времени  $t$ , т. е. процесс  $\xi(t)$  является Марковским, и, следовательно, аналитическое моделирование СМО с ограниченным временем ожидания возможно, только если в качестве операционной характеристики состояния рассматривается  $\xi(t)$ .

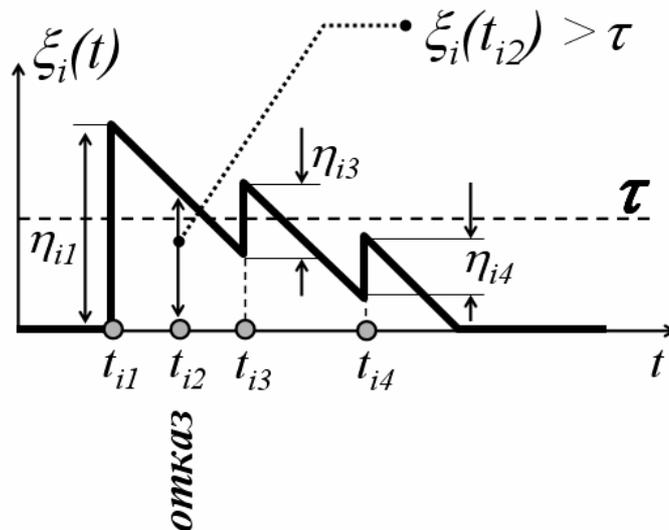


Рис. 27

### 3.3.3. Обслуживание с ограниченным временем пребывания

Постановка задачи обслуживания с ограничением времени пребывания совпадает с постановкой задачи обслуживания с ожиданием без потерь с тем лишь отличием, что каждая заявка может находиться в системе не более чем время  $\tau$ . Для заявки, поступившей в систему, может иметь место один из трех вариантов завершения пребывания в СМО:

- время ожидания и период обслуживания заявки оказалось меньше, чем  $\tau$  (заявка обслужена полностью);
- время ожидания оказалось меньше, чем  $\tau$ , но оставшегося до истечения  $\tau$  времени не хватило, чтобы полностью завершить обслуживание (заявка была потеряна, не будучи полностью обслуженной);
- время ожидания оказалось большим, чем  $\tau$ , и произошла «чистая» потеря заявки без затраты времени на обслуживание.

Если заявки обслуживаются в порядке очередности их поступления, то чистых потерь быть не может. Все три случая возможны, если обслуживание происходит не в порядке очередности, а в порядке случайного выбора из очереди.

*ПРИМЕР – доказательство невозможности «чистых» потерь для ординарного потока заявок.*



Рассмотрим случай упорядоченного обслуживания простейшего потока  $m$  одинаковыми ОА. Если  $\tau = \text{const}$ , то знания того, сколько заявок находится в системе в данный момент, совершенно не достаточно для заклю-

чения о том состоянии, в котором СМО будет находиться в последующие моменты времени. Судьба каждой заявки в значительной мере определяется моментом ее поступления.

Рассмотрим случайную функцию  $\xi_i(t)$ , определенную следующим образом:  $\xi_i(t)=0$ , если в момент времени  $t$   $i$ -й ОА свободен, в остальных случаях  $\xi_i(t)$  равно времени, которое должно пройти с момента  $t$  до того момента, когда ОА с номером  $i$  освободится от обслуживания заявок, поступивших до момента времени  $t$ . Из постановки задачи следует, что при любых  $t$  выполняется  $\xi_i(t) \leq \tau$ . Отложим по оси абсцисс время, а по оси ординат – функцию  $\xi_i(t)$ . Отметим моменты, в которые поступают заявки на обслуживание (см. рис. 28). Пусть до момента времени  $t_{i1}$  ОА был свободен. Это значит, что до этого момента функция  $\xi_i(t)=0$ , а в момент  $t_{i1}$  совершает скачек. Если необходимая длительность обслуживания  $\eta_{i1}$  равна или меньше  $\tau$ , то  $\xi_i(t_{i1}+0) = \eta_{i1}$ , а если  $\eta_{i1} > \tau$ , то  $\xi_i(t_{i1}+0) = \tau$ . При возрастании времени  $\xi_i(t)$  убывает на величину протекшего периода до тех пор, пока не обратится в 0 или же до ближайшего момента поступления новой заявки.

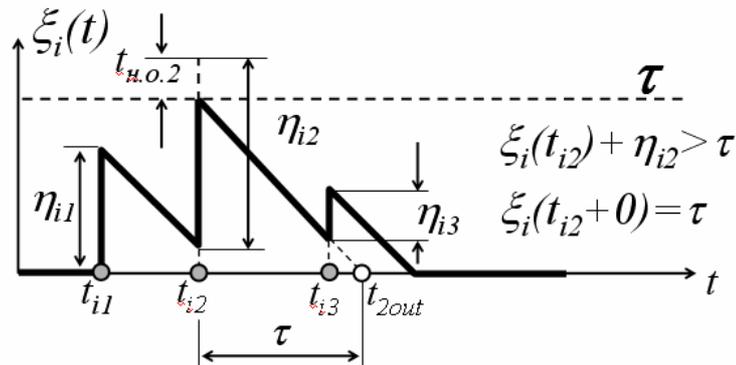


Рис. 28

Как видно на рис. 28, вторая заявка поступила в  $i$ -ый ОА на обслуживание, когда первая заявка еще не была обслужена полностью. Сумма остатка незавершенного обслуживания первой заявки и требуемой длительности обслуживания второй заявки  $\eta_{i2}$  превысила заданное ограничение  $\tau$ . В результате вторая заявка покинула систему в момент времени  $t_{2out} = t_{i2} + \tau$ , а остаток незавершенного обслуживания составил  $t_{n.o.2}$ .

Если рассматривать  $m$ -мерный случайный процесс  $\xi(t) = \{\xi_1(t), \dots, \xi_m(t)\}$ , то для заявки, поступающей в систему в момент времени  $t$ , время ожидания начала обслуживания равно  $\min_{i=1 \dots m} \xi_i(t)$ .

Т. о. процесс  $\xi(t)$  дает необходимые сведения о том, что ожидает поступившую в систему заявку. Если имеется несколько ОА, для которых  $\xi_i(t-0)$  достигают минимума, то заявка выбирает любой из них.

Процессы  $\xi(t)$ , определенные для двух рассмотренных случаев – обслуживание с ограниченным временем ожидания и ограниченным временем пребывания, описаны одинаково. Однако, особенности постановок задач на них отразились – в первом случае  $\xi_i(t)$  может принимать любые неотрицательные значения, тогда как при обслуживании с ограниченным временем пребывания она ограничена сверху величиной  $\tau$ .

### 3.3.4. Модели приоритетного обслуживания

Если длительность обслуживания имеет произвольное распределение, задача о приоритетном обслуживании может быть сформулирована по одному из трех вариантов:

1) при поступлении заявки первого типа обслуживание заявки второго типа прерывается: после того, как все имеющиеся заявки первого типа обслужены, ОА возобновляет прерванное обслуживание заявки второго типа, причем оставшееся время обслуживания уменьшается на то время, на протяжении которого эта заявка обслуживалась до момента поступления заявки первого типа (абсолютный приоритет с завершением незавершенного обслуживания);

2) то же, с тем лишь отличием, что при возобновлении обслуживания заявки второго типа время, ранее потраченное на его обслуживание, не учитывается, и обслуживание начинается заново (абсолютный приоритет с возобновлением незавершенного обслуживания).

3) при поступлении заявки первого типа обслуживание заявки второго типа полностью прекращается и эта заявка теряется (абсолютный приоритет с потерей незавершенного обслуживания).

Заявки первого типа обслуживаются совершенно независимо от заявок второго типа и время ожидания окончания обслуживания равно времени начала обслуживания плюс длительность обслуживания, поэтому для моделирования обслуживания заявок первого типа используется аналитическая модель *обслуживания с ожиданием*. Более сложно исследовать соответствующие характеристики по отношению к заявкам второго типа (в случае потери незавершенного обслуживания интерес представляет еще и величина вероятности того, что произвольная заявка будет потеряна). Для изучения характеристик обслуживания заявок второго типа используется математический аппарат теории массового *обслуживания с ненадежным*

*ОА*. Обслуживание заявок первого типа по отношению к заявкам второго типа эквивалентно отказу *ОА*. Таким образом, вместо того, чтобы рассматривать обслуживание заявок двух типов, можно рассмотреть схему обслуживания заявок только второго типа, а обслуживание поток заявок первого типа интерпретировать как поток отказов *ОА*.

### **3.4. Имитационное моделирование систем массового обслуживания**

При использовании математического аппарата ТМО часто удается быстро получить аналитические модели для решения достаточно широкого круга задач исследования ВС. В то же время аналитические модели имеют ряд существенных недостатков:

- значительные упрощения, свойственные большинству аналитических моделей (представление потоков заявок как простейших, предположение об экспоненциальном распределении длительностей обслуживания заявок, невозможность моделирования обслуживания заявки одновременно, например процессором и оперативной памятью, и др.);
- громоздкость вычислений;
- сложность учета в показателях эффективности характеристик процессов функционирования ПО.

В отличие от аналитического, имитационное моделирование снимает большинство ограничений, связанных с возможностью отражения в моделях реального процесса функционирования исследуемой ВС. Хотя имитационные модели во многих случаях более трудоемки чем аналитические, они могут быть сколь угодно близки к моделируемой системе. Имитационные модели ВС воспроизводят процессы генерации сообщений приложениями, разбиение сообщений на пакеты и кадры определенных протоколов, задержки, связанные с обработкой сообщений, пакетов и кадров внутри операционной системы, процесс получения доступа компьютером к разделяемой сетевой среде, процесс обработки поступающих пакетов маршрутизатором и т. д. Дополнительным преимуществом имитационных моделей является возможность подмены процесса смены событий в исследуемой системе в реальном масштабе времени на ускоренный процесс смены событий в модельном времени. Результатом работы имитационной модели являются собранные в ходе наблюдения за протекающими событиями статистические данные о наиболее важных характеристиках сети: временах реакции, коэффициентах использования каналов и узлов, вероятности потерь пакетов и т. п.

Существуют специальные языки имитационного моделирования (например, GPSS World), которые существенно облегчают процесс создания программной модели ВС по сравнению с использованием универсальных языков программирования.

### **3.4.1. Элементы имитационных моделей систем массового обслуживания**

Описания компонентов реальной ВС в имитационной модели носят определенный логико-математический характер и представляют собой совокупность алгоритмов, имитирующих функционирование исследуемой ВС. Моделирующая программа, построенная на основе этих алгоритмов, позволяет свести имитационное моделирование к проведению экспериментов путем их «прогона» на некотором множестве входных данных, имитирующих первичные события, которые происходят в системе. Статистика, фиксируемая в процессе моделирования, позволяет определить показатели, характеризующие качества ВС.

Имитационные модели состоят из элементов, в качестве которых фигурируют источники входных потоков заявок, устройства, памяти и узлы.

*Модель источника* входного потока заявок представляет собой алгоритм, по которому вычисляются моменты поступления заявок. Источники могут быть независимыми – реализует алгоритм выработки значений случайной величины, распределенной по заданному закону, и зависимыми – заявка вырабатывается при поступлении на некоторый вход синхронизирующей заявки. Каждый источник вырабатывает заявки одного типа и определенного приоритета.

Ресурсы ВС делятся на устройства (ОА) и памяти, причем устройство может обслуживать в каждый момент времени только одну заявку, а память – несколько.

*Модель устройства* представляет собой алгоритм выработки значений интервалов обслуживания. Для каждого типа заявок могут быть установлены свои распределения. Модель устройства также отражает заданную дисциплину обслуживания, так как в нее входит алгоритм, управляющий очередями в соответствии с дисциплиной обслуживания и приоритетами поступивших заявок.

*Модель памяти* представляет собой алгоритм определения объема памяти, требуемой для обслуживания заявки. Объем определяется как реализация случайной величины, причем закон распределения и его параметры зависят от типа заявки. Параметры памяти – емкость и дисциплина обслуживания. Заявка, поступившая в память, занимает вычисленный объем

и продолжает движение по СМО до встречи со специальным элементом освобождения памяти.

*Модель узла* используется для маршрутизации движения заявок в СМО и связей между элементами этой модели. Узлы могут быть нескольких типов и применяться для направления заявок по определенному или случайно выбираемому маршруту в зависимости от типа заявки или выполнения определенных условий; разделения потока заявок на части; объединения заявок друг с другом; изменения параметров заявок [4, 5, 25, 28].

### **3.4.2. Способы управления модельным временем**

Модельным временем в имитационной модели называется воспроизведение физического времени в модели. Соотношение физического и модельного времени определяется спецификой модели и задается диапазоном физического времени, принимаемого за единицу модельного времени. Сущностью имитационного моделирования является продвижение модельного времени и выполнение событий, связанных с определенными значениями модельного времени. Событие в модели это программный модельный образ значимого, с точки зрения разработчика модели, изменения в моделируемой системе.

Основной задачей имитационного моделирования является правильное отображение порядка и временных отношений между изменениями в моделируемой системе на порядок выполнения событий в модели.

Сжатие временной шкалы в непрерывном времени невозможно, поэтому в имитационном моделировании время изменяется скачкообразно. На оси модельного времени моменты наступления событий составляют дискретное множество. В дискретном имитационном моделировании используются два способа управления продвижением модельного времени:

- *событийный* (Event Driven), при котором в качестве следующего значения модельного времени выбирается минимальное время наступления события из списка будущих событий;
- *пошаговый* (Time Stepped), при котором значение модельного времени увеличивается на постоянную фиксированную величину – шаг модельного времени.

*ПРИМЕРЫ* соотношения физического и модельного времени при пошаговом и событийном управлении.



Пошаговый подход удобен при наличии условных событий, т. е. событий, для выполнения которых требуется истинность некоторого логиче-

ского условия. В этом случае на каждом шаге требуется вычислять логические условия и выполнять соответствующие события. При событийном способе управления временем можно «проскочить» момент времени, при котором условие стало истинным.

Однако пошаговое управление значительно менее эффективно – для обеспечения удовлетворительной точности шаг приращения времени должен быть минимальным, а в этом случае  $\approx 95\%$  обращений к элементам имитационной модели – лишние и много процессорного времени тратится на обработку «пустых» интервалов. При этом пошаговое управление не позволяет указать истинное положение событий внутри шага модельного времени (такта). Поэтому принимается соглашение переносить их на начало (или на конец) того такта, в пределах которого они в действительности произошли. При этом искажается реальная картина событий, теряются причинно-следственные связи, последовательно протекающие события становятся параллельными, задержки распространения сигналов в структурных элементах не отображаются. Этим недостатком лишено событийное управление модельным временем [2, 29].

### **3.4.3. Алгоритмы имитационного моделирования для событийного управления модельным временем**

Отличие имитационного моделирования от объектно-ориентированного программирования заключается в том, что объект имитационной модели может не только выполнить некоторое событие в момент своей активности, но и запланировать выполнение своего события или события другого объекта в будущем, т. е. на момент модельного времени, больший или равный текущему значению модельного времени. Для реализации выполнения будущих событий и упорядочивания их в хронологическом порядке необходимо использование управляющей программы – планировщика. Алгоритм работы планировщика состоит в выполнении следующих действий:

1) активизация объектов для выполнения событий, запланированных на текущее значение модельного времени и удаление выполненных событий из списка;

2) включение в список новых событий, запланированных активными объектами, вместе со значениями моментов модельного времени, в которые каждое из этих событий должно быть выполнено;

3) увеличение значения модельного времени, если на текущий момент времени невыполненных событий не осталось, и переход на п. 1).

Понятно, что при такой организации моделирования события в модели выполняются последовательно, даже если они относятся к одному моменту модельного времени. Однако, поскольку продвижение модельного времени на время моделирования параллельных событий приостанавливается, говорят, что процессы выполняются *квазипараллельно*.

В алгоритме имитационного моделирования СМО при событийном управлении модельным временем используется несколько информационных массивов: списки текущих (СТС) и будущих (СБС) событий, массив заявок (МЗ) – хранит имя, тип, приоритет, время, местонахождение заявки в системе, и очередей (МО) – хранит информацию о заявках в очереди, упорядочен по ОА. Моделирование начинается с просмотра операторов генерирования заявок, т. е. с обращений к моделям источников заявок. Для каждого независимого источника такое обращение позволяет рассчитать момент генерации первой заявки. Этот момент вместе с ссылкой на заявку заносится в СБС, а сведения о заявке – в МЗ (имя, значения параметров, место появления или нахождения заявки в системе). В СБС события упорядочиваются по возрастанию времен совершения. Далее из СБС выбирается совокупность событий, относящихся к наиболее раннему моменту модельного времени. Эта совокупность переносится в СТС и начинается моделирование событий, отмеченных в СТС (см. рис. 29).

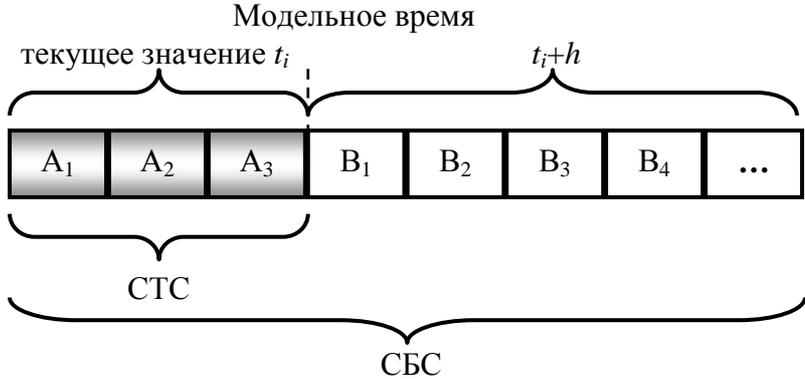


Рис. 29

Выбирается ссылка на событие, по ней в МЗ определяется соответствующая заявка и ее место в системе, моделируется продвижение по системе по маршруту, определяемому программой моделирования, до тех пор, пока заявка не придет на вход некоторого ОА. Тогда обращение к модели ОА позволяет определить длительность задержки на обслуживание и момент наступления события, связанного в выходом заявки из ОА. Корректируется местонахождение заявки в МЗ. Ссылка на новое предвидимое со-

бытие заносится в СБС так, чтобы сохранилась упорядоченность списка по моментам наступления событий. Программа моделирования приступает к выбору очередной ссылки на СТС.

После имитации всех событий из СТС в него переносится очередная совокупность событий из СБС, относящихся к ближайшему моменту модельного времени  $t_{i+1}$ , текущее модельное время принимает это значение.

Если при моделировании движения заявки она придет на вход занятого ОА (устройства или накопителя), то вместо расчета длительности обслуживания имя заявки заносится в МО. При моделировании события, связанного с освобождением ОА заявкой  $a$ , проверяется состояние очереди к освобождающемуся ОА. Если имеется очередь, то в соответствии с дисциплиной обслуживания из очереди выбирается заявка  $b$  и входит на обслуживание ОА. Обращение к модели обслуживающего аппарата дает значение освобождения ОА заявкой  $b$  и соответствующая ссылка заносится в СБС. Затем программа производит моделирование продвижения заявки  $a$  до того момента, когда произойдет ее выход из системы или задержка в очереди к новому ОА [1, 28, 29].

#### **3.4.4. Алгоритмы имитационного моделирования для пошагового управления модельным временем**

При пошаговом управлении модельным временем трудоемкость анализа имитационной модели определяется количеством уравнений в итоговой системе и количеством тактов, на которое разделен моделируемый интервал времени. Для моделирования в этом случае используются итерационные алгоритмы функционального моделирования дискретных систем.

Для решения систем уравнений вида  $V = \psi(V, X)$ , где  $V$  – вектор базисных переменных;  $X$  – вектор входных переменных модели;  $\psi$  – оператор преобразования дискретных переменных, применяются итерационные алгоритмы. Анализ начинается с задания вектора входных воздействий  $X$  и вектора начального приближения  $V_0$  для искомого вектора  $V$ .

*ПРИМЕРЫ синхронного анализа логической схемы с использованием итерационных алгоритмов.*



*Алгоритм простой итерации* состоит в выполнении итераций по следующей формуле:

$$V_i = \psi(V_{i-1}, X), \quad (10)$$

где  $V_i$  – значение вектора  $V$  на  $i$ -й итерации. Если  $V_i = V_{i-1}$ , то решение найдено; если  $V_i \neq V_{i-1}$ , то выполняется новая итерация; если итерационный

процесс не сходится, то это свидетельствует об ошибках моделирования или реального объекта, вызывающих неустойчивость состояния. Практически считается, что процесс не сходится, если условие  $V_i = V_{i-1}$  не достигается на заранее заданном количестве итераций.

Рассмотрим постановку задачи моделирования СМО с ОА типа G/G/1 (см. рис. 30) для пошагового управления модельным временем.

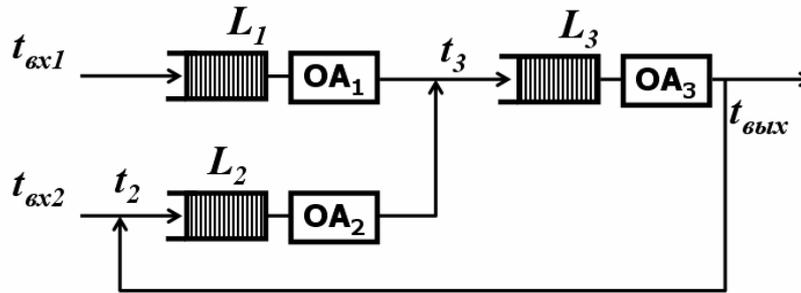


Рис. 30

Вектор входных переменных модели для СМО, представленной на рис. 30, имеет вид:

$$\vec{X} = \{t_{ex1}, t_{ex2}\},$$

где  $t_{ex1}$  и  $t_{ex2}$  – время поступления заявок входных потоков на обслуживание в очередь  $OA_1$  и  $OA_2$  соответственно. Вектор базисных координат может быть определен как:

$$\vec{V} = \{t_2, t_3, t_{вых}, L_1, L_2, L_3\},$$

где  $t_2$  – время поступления заявки из рецикла на повторное обслуживание в очередь  $OA_2$ ;  $t_3$  – время поступления заявки на обслуживание в очередь  $OA_3$ ;  $t_{вых}$  – время выхода заявки из системы;  $L_1, L_2, L_3$  – длины очередей соответствующих ОА.

Пусть на  $k$ -тый момент модельного времени значение вектора базисных координат известно и может быть использовано в качестве начального приближения для расчетов в следующем такте модельного времени:

$$\vec{V}_0 = \{t_{20}, t_{30}, t_{вых0}, L_{10}, L_{20}, L_{30}\}$$

Пусть в  $k+1$  момент модельного времени значение вектора входных переменных составило:

$$\vec{X}_{k+1} = \{t_{ex1}, t_{ex2}\}$$

Требуется определить значение вектора базисных координат  $\vec{V}$  в  $k+1$  такте модельного времени. В отсутствии обратных связей (рециклов) в

схеме СМО такой расчет потребовал бы однократного пересчета всех базисных координат с учетом нового значения вектора входных переменных, т. е. с учетом значений времен поступления очередных заявок входного потока, округленных до целого числа тактов модельного времени, и времен обслуживания в ОА СМО. При наличии обратных связей, как в заданной СМО, вычисление окончательного значения вектора базисных координат потребует нескольких итераций, так как переменные  $t_2$  и  $t_{вых}$  взаимосвязаны. Система уравнений модели заданной СМО имеет вид:

$$\begin{cases} t_2 = \Psi_1(t_{ex2}, t_{вых}); \\ t_3 = \Psi_2(t_2, L_2, L_1); \\ t_{вых} = \Psi_3(t_3, L_3); \\ L_1 = \Psi_4(t_{ex1}, L_1); \\ L_2 = \Psi_5(t_2, L_2); \\ L_3 = \Psi_6(t_3, L_3), \end{cases}$$

где  $\Psi_i$  – функции, зависящие от закона распределения времени обслуживания в АО и дисциплины очереди.

Согласно формуле (10) в правые части уравнений модели на каждой итерации в этом алгоритме подставляются значения базисных координат, полученных на предыдущей итерации. На первой итерации такими значениями являются значения базисных координат из начального приближения вектора, следовательно:

$$\begin{cases} t_{2_1} = \Psi_1(t_{ex2}, t_{вых_0}); \\ t_{3_1} = \Psi_2(t_{2_0}, L_{2_0}, L_{1_0}); \\ t_{вых_1} = \Psi_3(t_{3_0}, L_{3_0}); \\ \dots, \end{cases}$$

и так далее. Следует обратить внимание, что, например, для расчета базисной координаты  $t_{3_1}$  в правую часть соответствующего уравнения подставляется значение  $t_{2_0}$  несмотря на то, что к моменту расчета  $t_{3_1}$  уже известно и значение  $t_{2_1}$ . Такая организация расчетов потребует, очевидно, большего количества итераций, но, при этом, в алгоритме нет необходимости выполнять проверку: определено ли значение очередной базисной координаты уже на этой итерации, или еще нет.

Признаком того, что решение найдено, в этом итерационном метода, как и во всех прочих, является совпадение результата последней итерации с предыдущей.

Уменьшить количество вычислений удастся при построении итерационного процесса с использованием *алгоритма Зейделя*, в котором при вычислении очередного из элементов вектора  $V_i$  в правую часть уравнений системы там, где это возможно, подставляются не элементы вектора  $V_{i-1}$ , а те элементы вектора  $V_i$ , которые уже вычислены к данному моменту, т. е. итерации выполняются по формуле:

$$V_i = \psi(V_i, V_{i-1}, X).$$

Количество итераций в алгоритме Зейделя существенно зависит от порядка, в котором реализуются уравнения модели. В алгоритме *Зейделя без ранжирования* уравнения модели перечисляются в произвольном порядке. В алгоритме *Зейделя с ранжированием* уравнения располагаются в том порядке, в каком соответствующие уравнениям элементы схемы образуют путь прохождения заявок. Тогда для анализа схем без обратных связей потребуется всего одна итерация. В схемах с обратной связью метод Зейделя с ранжированием уравнений порождает несколько итераций, но их количество существенно меньше, чем в методе простой итерации.

*Ранжирование уравнений* производится следующим образом: уравнение модели (элемент СМО) получает ранг  $j$ , если все аргументы этого уравнения (входы элемента) ранжированы и максимальный среди рангов аргументов (входов) равен  $j-1$ . Переменная модели получает ранг  $j$ , если она является левой частью уравнения (является выходом элемента), имеющего ранг  $j$ . Выполнение алгоритма начинается с того, что всем входным переменным присваивается ранг  $j=0$ . Если в схеме имеются контуры ОС, одна из цепей каждого контура должна быть предварительно разорвана и части разорванных цепей, подключенные к входам элементов, получают ранг  $j=0$ . Затем определяют уравнения первого ранга, переменные первого ранга, элементы второго ранга и т. д. В итоге уравнения располагаются в порядке возрастания рангов.

Выполним ранжирование уравнений системы, описывающей СМО, приведенную на рис. 30 и рассмотренную выше, в примере моделирования по методу простой итерации. Присвоим нулевой ранг входным переменным:

$$\left\{ \begin{array}{l} 1) t_2 = \psi_1(t_{ex2}^0, t_{вых}); \\ 2) t_3 = \psi_2(t_2, L_2, L_1); \\ 3) t_{вых} = \psi_3(t_3, L_3); \\ 4) L_1 = \psi_4(t_{ex1}^0, L_1); \\ 5) L_2 = \psi_5(t_2, L_2); \\ 6) L_3 = \psi_6(t_3, L_3). \end{array} \right.$$

В четвертом уравнении теперь фигурирует единственная непроранжированная переменная:  $L_1$ . Ей присваивается ранг, на 1 больший минимального в уравнении, т. е. в данном случае равный 1. Этим же рангом помечаются все вхождения переменной  $L_1$  в другие уравнения:

$$\left\{ \begin{array}{l} 1) t_2 = \psi_1(t_{ex2}^0, t_{быx}); \\ 2) t_3 = \psi_2(t_2, L_2, L_1^1); \\ 3) t_{быx} = \psi_3(t_3, L_3); \\ 4) L_1^1 = \psi_4(t_{ex1}^0, L_1^1); \\ 5) L_2 = \psi_5(t_2, L_2); \\ 6) L_3 = \psi_6(t_3, L_3). \end{array} \right.$$

После выполнения последнего шага дальнейшее ранжирование становится невозможным без условного разрыва обратной связи. Он позволяет отнести переменную  $t_{быx}$  к нулевому рангу. В первом уравнении все переменные в правой части оказываются проранжированными,  $t_2$  в этом уравнении, а также в правых частях второго и пятого уравнений, получает ранг 1:

$$\left\{ \begin{array}{l} 1) t_2^1 = \psi_1(t_{ex2}^0, t_{быx}^0); \\ 2) t_3 = \psi_2(t_2^1, L_2, L_1^1); \\ 3) t_{быx} = \psi_3(t_3, L_3); \\ 4) L_1^1 = \psi_4(t_{ex1}^0, L_1^1); \\ 5) L_2 = \psi_5(t_2^1, L_2); \\ 6) L_3 = \psi_6(t_3, L_3). \end{array} \right.$$

В пятом уравнении теперь фигурирует единственная непроранжированная переменная:  $L_2$ , которая фигурирует и в правой, и в левой части. Ей присваивается ранг, на 1 больший минимального в уравнении, т. е. в данном случае равный 2. Этим же рангом помечаются все вхождения переменной  $L_2$  в другие уравнения, что позволяет определить ранг переменной  $t_3$  равным 3. С учетом вхождений этой переменной получим:

$$\left\{ \begin{array}{l} 1) t_2^1 = \psi_1(t_{ex2}^0, t_{быx}^0); \\ 2) t_3^3 = \psi_2(t_2^1, L_2^2, L_1^1); \\ 3) t_{быx} = \psi_3(t_3^3, L_3); \\ 4) L_1^1 = \psi_4(t_{ex1}^0, L_1^1); \\ 5) L_2^2 = \psi_5(t_2^1, L_2^2); \\ 6) L_3 = \psi_6(t_3^3, L_3). \end{array} \right.$$

В шестом уравнении теперь фигурирует единственная непроранжированная переменная:  $L_3$ , которая фигурирует и в правой, и в левой части. Ей присваивается ранг, на 1 больший минимального в уравнении, т. е. в данном случае равный 4. Этим же рангом помечаются вхождение переменной  $L_3$  в третье уравнение, что позволяет определить действительный (без условного разрыва обратной связи) ранг переменной  $t_{бых}$  равным 5. Окончательный результат ранжирования имеет вид:

$$\left\{ \begin{array}{l} 1) t_2^1 = \psi_1(t_{ex2}^0, t_{бых}^0); \\ 2) t_3^3 = \psi_2(t_2^1, L_2^2, L_1^1); \\ 3) t_{бых}^5 = \psi_3(t_3^3, L_3^4); \\ 4) L_1^1 = \psi_4(t_{ex1}^0, L_1^1); \\ 5) L_2^2 = \psi_5(t_2^1, L_2^2); \\ 6) L_3^4 = \psi_6(t_3^3, L_3^4). \end{array} \right.$$

Расположим уравнения по возрастанию рангов:

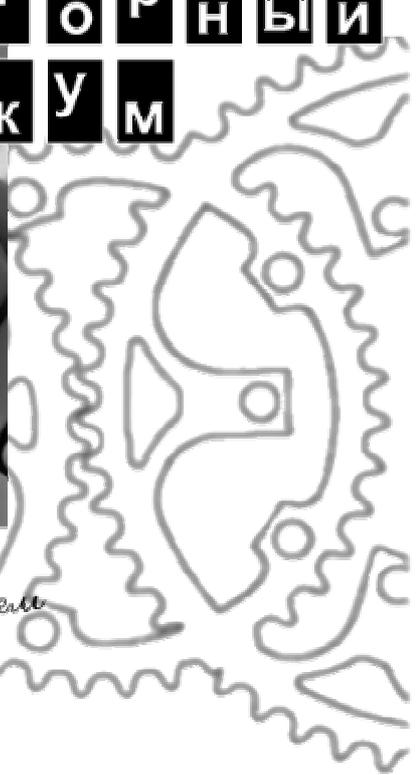
$$\left\{ \begin{array}{l} t_2^1 = \psi_1(t_{ex2}^0, t_{бых}^0); \\ L_1^1 = \psi_4(t_{ex1}^0, L_1^1); \\ L_2^2 = \psi_5(t_2^1, L_2^2); \\ t_3^3 = \psi_2(t_2^1, L_2^2, L_1^1); \\ L_3^4 = \psi_6(t_3^3, L_3^4); \\ t_{бых}^5 = \psi_3(t_3^3, L_3^4). \end{array} \right.$$

Если сопоставить ранжированный порядок уравнений со схемой на рис. 30, легко убедиться, что порядок вычислений в этом случае соответствует логике прохождения заявок по СМО и порядку изменения переменных модели. Понятно, что для такой простой схемы расположить уравнения в правильном порядке можно и без процедуры ранжирования. Однако в сложных схемах с большим количеством ОА и обратных связей эта задача не тривиальна, и приходится прибегать к ранжированию.

Наименьший объем вычислений обеспечивает *событийный алгоритм*. Основная идея событийного метода заключается в выполнении вычислений по уравнениям только *активизированных* элементов, т. е. элементов, у которых хотя бы на одном входе произошло событие (изменилась входная переменная). В алгоритме событийного метода на каждой итерации имеется своя группа активизированных элементов. Использование метода позволяет существенно сократить затраты машинного времени при анализе имитационных моделей СМО [3, 10, 29].

# МОДЕЛИРОВАНИЕ

## ЛАБОРАТОРНЫЙ ПРАКТИКУМ



- работа №1
- работа №2

*моделирование систем  
различной природы*

*моделирование  
систем  
массового  
обслуживания в  
GPSS World*

- работа №3

`*GENERATE*QUEUE*SEIZE*DEPART*ADVANCE*  
*RELEASE*TERMINATE*START*QTABLE*`

- работа №4

`*STORAGE*ENTER*LEAVE*TRANSFER*GATE*  
*FUNCTION*`

- работа №5

`*ASSIGN*MARK*PRIORITY*TEST*PREEMPT*  
*RETURN*`

- работа №6

`*VARIABLE*BVARIABLE*INITIAL*SAVEVALUE*  
*FUNAVAIL*FAVAIL*SUNAVAIL*SAVAIL*DISPLACE*`

- работа №7

*моделирование систем массового  
обслуживания сетями Петри*



Из-за задержек в прохождении сигналов в схемах могут возникать сбои: появление ложных сигналов. Для оценки рисков сбоя используют трех- и пятизначные модели. Риск сбоя это *возможность* появления ложных сигналов на выходах схемы из-за взаимного влияния задержек распространения сигналов. Различают статический и динамический риск сбоя. *Статический риск сбоя* это возможность такого изменения переменной на выходе какого-либо элемента, которого при нормальном функционировании быть не должно. *Динамический риск сбоя* представляет собой опасность многократных изменений выходной переменной вместо правильного однократного изменения.

Ситуация, связанная с статистическим риском сбоя, иллюстрируется на примере двухвходового элемента И-НЕ временной диаграммой. Если в некотором такте переключение сигналов  $a$  и  $b$  происходит одновременно, то выходной сигнал  $c=1$  остается неизменным в течение такта  $T$  (см. рис. 31-а). Но сигналы, вызывающие изменения  $a$  и  $b$ , могут проходить через предшествующие цепочки элементов с задержками. Поэтому возможно неодновременное, с задержкой на некоторую величину  $\tau < T$  переключение  $a$  и  $b$ , и в ситуации, показанной на рис. 31-б, появляется ложный сигнал  $c=0$ . Наличие сбоя в работе схемы в середине такта установить с использованием двоичной модели невозможно, так как из-за характерного для этих моделей дискретного представления времени вычисление переменных модели производится только на начало и конец такта  $T$ , а значит в данном случае расчет даст следующий результат: сигнал  $c$  в течение такта не изменяется и равен 1.

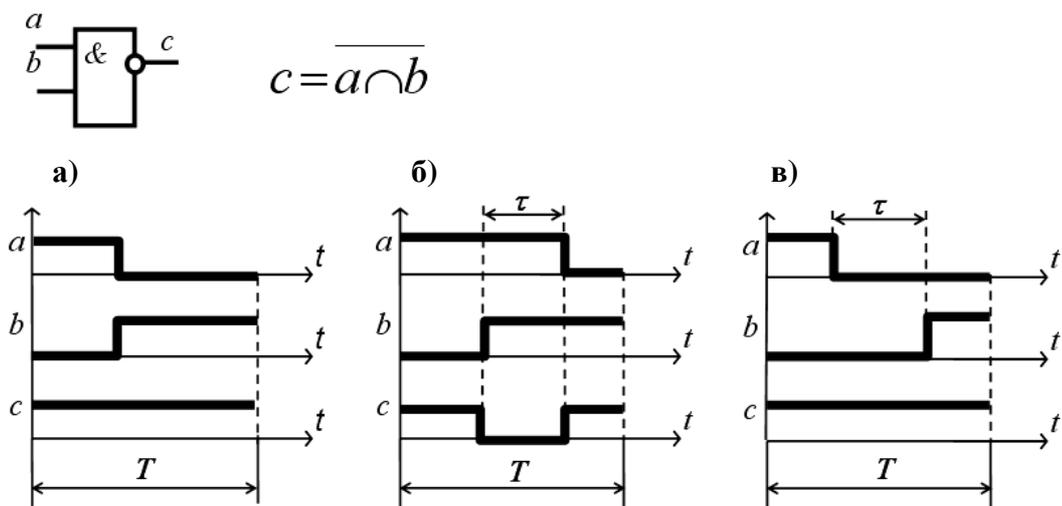


Рис. 31

Вместе с тем, сам факт прохождения входных сигналов с задержками, вызывающими их несинхронное переключение, еще не означает, что сбой обязательно будет. Так, в ситуации, показанной на рис. 31-в, сигналы  $a$  и  $b$  переключаются не синхронно, но сбоем тем не менее не возникает, поскольку на время задержки  $\tau$  оба входных сигнала имеют значение 0, и сигнал  $c$  остается неизменно равным 1.

В трехзначной модели значение равное 2 задает неопределенное состояние, соответствующего выхода или входа элемента. Неопределенные состояния возникают во время переходных процессов как промежуточные при переключениях из состояния 1 в состояние 0 или наоборот. Правила выполнения основных логических операций И, ИЛИ, НЕ приведены в таблице 13.

Таблица 13

$a$	0	1	2	0	1	2	0	1	2
$b$	0	0	0	1	1	1	2	2	2
$a \cap b$	0	0	0	0	1	2	0	2	2
$a \cup b$	0	1	2	1	1	1	2	1	2
$\bar{a}$	1	0	2	1	0	2	1	0	2

При анализе трехзначных моделей в течение одного такта значения переменных вычисляются дважды. Сначала определяются промежуточные значения, затем окончательные. Промежуточные значения *входных сигналов* определяются по следующему правилу: если исходное и окончательное значение сигнала не совпадают, то промежуточное равно 2, а если исходное и окончательное совпадают, то промежуточное равно им. Промежуточные значения *выходных сигналов* рассчитываются по правилам трехзначной логики. Статический риск сбоя по выходному сигналу имеет место в случае, если сочетание значений в исходном, промежуточном и окончательном состоянии имеет вид 0-2-0 или 1-2-1 [3, 20].

Рассмотрим пример оценки статического риска сбоя для схемы И-ИЛИ-НЕ (см. рис. 32) с использованием трехзначного моделирования. Пусть вектор входных воздействий  $X=(a,b,c)$  изменяется в некотором такте с (0,1,0) на (1,0,0). Результат расчета показывает наличие статического риска сбоя по выходным переменным  $g$  и  $e$  (см. таблицу 14).

Обнаружение статического риска сбоя при помощи трехзначных моделей рассмотренного типа, называемых синхронными, еще не означает, что сбой действительно будет. Результаты синхронного анализа – это результаты наихудшего случая; они лишь подтверждают, что для данной схемы в принципе существует такое сочетание задержек прохождения сигналов, при котором возникнет сбой.

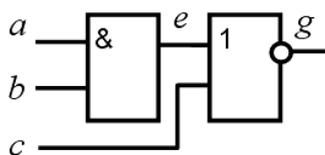


Рис. 32

Значения переменных	a	b	c	e	g
Исходное	0	1	0	0	1
Промежуточное	2	2	0	2	2
Окончательное	1	0	0	0	1

### Ход выполнения работы

1. Для заданного варианта схемы записать систему уравнений математической модели.

2. В *MS Excel* разработать рабочий лист трехзначного анализа схемы (расчета промежуточных значений входных переменных; исходных, промежуточных и окончательных значений выходных переменных). Для 3-х заданных вариантов изменения вектора входных переменных осуществить трехзначный анализ схемы на наличие статического риска сбоя. Сделать выводы по результатам анализа.

3. Подтвердить результаты анализа схемы построением диаграмм выходных сигналов схемы в *CircuitMaker*.

4. Результаты выполнения работы по пунктам 1-3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

### Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования *MS Excel*.

Рабочий лист *MS Excel* представляет собой электронную таблицу, в ячейках которой могут содержаться числа, текст и формулы. Формула *MS Excel* начинается со знака «=» и представляет собой выражение, содержащее ссылки на ячейки, содержащие числа; операнды функций и знаки арифметических действий. Например, формула вида:

$$= A21^2 + \text{COS}(B21)$$

осуществит сложение второй степени числа, содержащегося в ячейке A21 и косинуса числа, содержащегося в ячейке B21. Ввод формул можно осуществлять вручную, при этом вводимая формула отображается в строке состояния в верхней части таблицы. При вводе формул ссылки на ячейки таблицы следует набирать на латинском регистре. При нажатии клавиши Enter в ячейке ввода отобразится результат расчета по формуле.

Формулы также можно вводить с помощью Мастера Функций, запуск которого осуществляется кнопкой со значком  $f_x$ . В открывшемся окне

диалога следует выбрать категорию требуемой функции, конкретную функцию и далее следовать диалогу Мастера Функций.

При копировании фрагментов таблицы вставка текста и чисел в новом месте осуществляется без изменений, а в формулах автоматически изменяются ссылки. Так, если формулу предыдущего примера из C21 скопировать в D25, то она изменится следующим образом:

$$=B25^2+COS(C25)$$

Если при копировании формулы необходимо защитить ссылку от автоматического изменения, перед буквой, и/или номером ячейки в зависимости от того, что требуется оставлять при копировании без изменений, следует поставить знак \$.

Для решения задачи анализа цифровых схем необходимо реализовывать расчеты в двоичной и трехзначной логике. В *MS Excel* имеются встроенные стандартные логические функции двоичной логики. Так, например, логическая функция ИЛИ возвращает результат ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА; возвращает ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ. Синтаксис функции:

$$=ИЛИ(\text{логическое\_значение1}; \text{логическое\_значение2}; \dots)$$

Так, формула =ИЛИ(1;0) даст результат ИСТИНА, а формула =ИЛИ(0;0;0) результат ЛОЖЬ. Для преобразования результата расчета логической функции в цифровой двоичный вид ее следует умножить на 1, т. е., например, формула =1\*ИЛИ(0;0;0) даст результат 0.

Встроенных функций трехзначной логики в *MS Excel* нет, однако их можно реализовать комбинациями логических и арифметических действий. Так, правила выполнения операции И в трехзначной логике имеют вид, приведенный в таблице 15.

Таблица 15

<i>a</i>	0	1	2	0	1	2	0	1	2
<i>b</i>	0	0	0	1	1	1	2	2	2
$a \cap b$	0	0	0	0	1	2	0	2	2

Обобщенные правила выполнения операции И могут быть сформулированы следующим образом:

- если арифметическое произведение *a* и *b* равно нулю, то результат выполнения операции И равен нулю;
- если *a* и *b* равны друг другу и равны единице, то результат выполнения операции И равен единице;
- во всех остальных случаях результат равен 2.

В *MS Excel* эти правила могут быть реализованы формулой:

$$=ЕСЛИ(a*b=0;0;ЕСЛИ(a*b=1;1;2))$$

где  $a$  и  $b$  – ссылки на ячейки, в которых содержатся значения входных переменных. Здесь логическая функция ЕСЛИ имеет следующий синтаксис:

$$=ЕСЛИ(\text{лог\_выражение};\text{значение\_если\_истина};\text{значение\_если\_ложь})$$

и возвращает одно значение, если заданное условие при вычислении дает значение ИСТИНА, и другое значение, если ЛОЖЬ.

Аналогично правила выполнения операции ИЛИ в трехзначной логике имеют вид, приведенный в таблице 16:

Таблица 16

$a$	0	1	2	0	1	2	0	1	2
$b$	0	0	0	1	1	1	2	2	2
$a \cup b$	0	1	2	1	1	1	2	1	2

Эти правила можно обобщить следующим образом:

- если  $a$  и  $b$  равны друг другу, то результат выполнения операции ИЛИ равен их общему значению;
- если арифметическая сумма  $a$  и  $b$  равна 1 или 3, то результат выполнения операции ИЛИ равен единице;
- во всех остальных случаях результат равен 2.

В *MS Excel* сформулированные правила могут быть реализованы формулой:

$$=ЕСЛИ(a=b;b;ЕСЛИ(ИЛИ(a+b=1;a+b=3);1;2))$$

Правила выполнения операции НЕ в трехзначной логике можно сформулировать следующим образом:

- если подоперационное выражение равно 2, то результат выполнения операции равен 2;
- в остальных случаях выполняется двоичная операция НЕ.

В *MS Excel* это реализуется как:

$$=ЕСЛИ(a=2; 2;1*НЕ(a))$$

где  $a$  – ссылка на ячейку или формула вычисления подоперационного выражения.

Пример разработки рабочего листа синхронного анализа цифровой схемы И-ИЛИ-НЕ на наличие статического риска сбоя в *MS Excel* приведен в файле *Пример (ЛР№1).xls*.

### *Circuit Maker PRO.*

*CircuitMaker* это программный продукт моделирования цифровых, аналоговых и аналоговые/цифровых электрических схем. В панели инструментов *CircuitMaker* расположены следующие кнопки:

- инструмент  используется для выбора элементов цепи, их перемещения, выбора элементов на панели инструментов и т. д.;
- инструмент для рисования проводников  используется для размещения проводников в рабочей области и соединения элементов цепи;
- инструмент  используется для удаления элемента из схемы;
- клавиша  служит для инициализации схем и отказа от предыдущих результатов анализа (например, от построенных диаграмм);
- клавиша  служит для запуска анализа электрической цепи.

В режиме редактирования схем в левой части рабочего окна *CircuitMaker* располагается дерево базы данных элементов схем (вывод его на экран осуществляется клавишей  в панели инструментов).

Для просмотра диаграмм цифровых сигналов в соответствующую точку схемы следует присоединить устройство SCOPE из библиотеки Instruments/Digital/SCOPE. Для вывода/остановки вывода диаграммы цифрового сигнала используйте клавишу /, для пошагового прогона – клавишу .

Для моделирования входных сигналов цифровых схем используется импульсный генератор Instruments/Digital/Pulser. Диалоговое окно свойств имеет вид, приведенный на рис. 33. В свойствах данного инструмента можно задавать длительность сигналов высокого и низкого уровней, нормальный или инвертированный тип импульсного сигнала. При установленном флаге External Trigger генератор сработает только один раз.

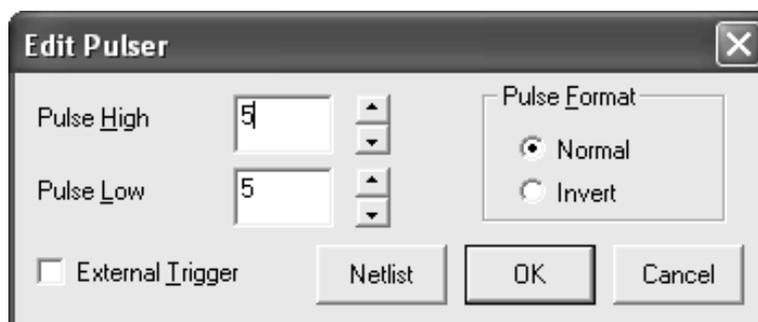


Рис. 33

Для задания и редактирования опций цифрового моделирования требуется вызвать диалоговое окно через меню Simulation>Digital Options... Диалоговое окно имеет вид, приведенный на рис. 34. В нем можно определить размер шага (Step Size) в тактах (Ticks) или циклах (Cycles). Также можно задать масштабирование сигнала по оси  $x$  на диаграммах сигналов (X Magnification). Цикл состоит из 10 тактов.

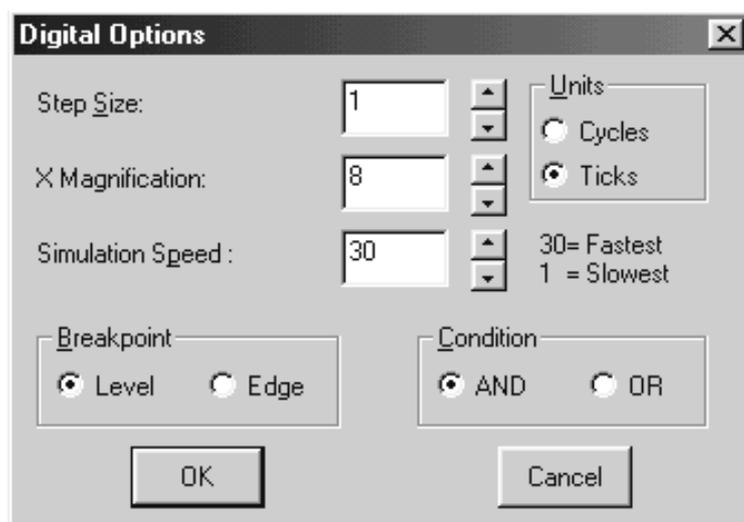


Рис. 34

Для копирования и вставки в отчет схем и графиков, построенных в *Circuit Maker*, следует использовать меню Edit/Copy to Clipboard.

Пример схемы И-ИЛИ-НЕ, разработанной в *CircuitMaker* для построения диаграмм выходных сигналов по результатам статического анализа статических рисков сбоя, имеет вид, приведенный на рис. 35 (см. также файл *Пример (ЛР№1).ckt*). Пусть для данной схемы вектор входных сигналов в первом случае изменяет в течение такта значение с исходного  $(a,b,c)=(1,0,0)$  на окончательное  $(0,1,0)$ . Расчет на трехзначной модели для этого случая показывает наличие статического риска сбоя для выходных переменных  $e$  и  $g$  (см. файл *Пример (ЛР№1).ckt*).

Результат построения диаграмм выходных сигналов для первого случая изменения вектора входных переменных при прохождении входных сигналов без задержек, имеет вид, представленный на рис. 36.

Результат построения диаграмм выходных сигналов для первого случая изменения вектора входных переменных при прохождении входных сигналов с задержками, имеет вид, представленный на рис. 37. *Обратите внимание*, что параметры входных сигналов подобраны таким образом, чтобы исследуемое с задержкой переключение  $a$  из 1 в 0 и  $b$  из 0 в 1 про-

исходило в середине такта, а обратные переключения совпадали с границей такта и были синхронными. В этом случае легко идентифицировать наличие/отсутствие сбоя. Видно, что именно при не одновременном переключении сигналов на входах *a* и *b* из 1 в 0 и из 0 в 1 соответственно, при сохранении значения 0 на входе *c*, на выходах *e* и *g* появляются ложные сигналы (1 и 0 соответственно), т. е. имеет место статический сбой, что совпадает с результатами анализа на трехзначной модели. Сдвиг фронтов выходных сигналов объясняется условиями моделирования – как уже отмечалось, все функциональные элементы в *CircuitMaker* по умолчанию имеют задержку, равную 1 такту.

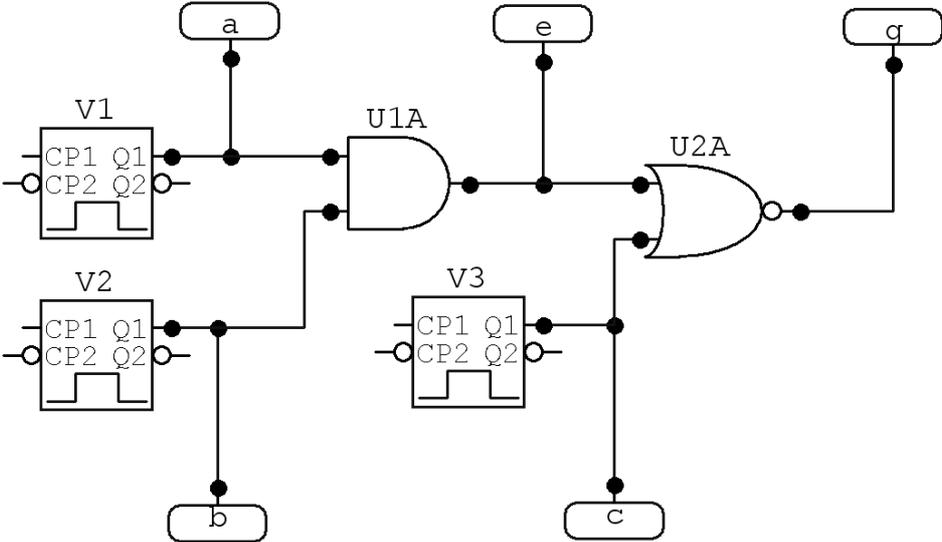


Рис. 35



Рис. 36

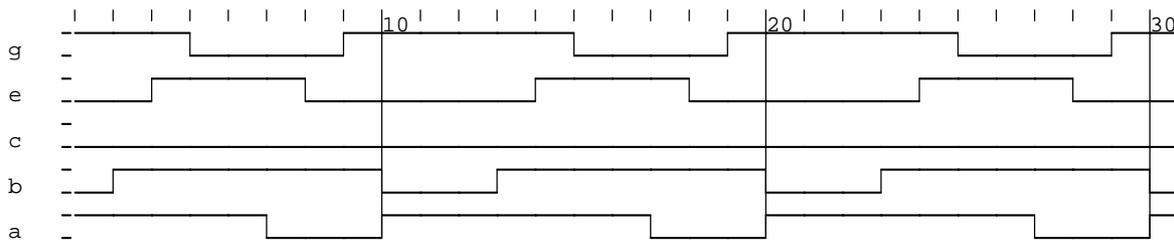


Рис. 37

Во втором случае для данной схемы вектор входных сигналов изменяет в течение такта значение с исходного  $(a, b, c) = (1,0,0)$  на окончательное  $(0,0,1)$ . Расчет на трехзначной модели для этого случая показывает отсутствие статического риска сбоя (см. файл *Пример (ЛР№1).ckt*).

Результат построения диаграмм выходных сигналов для второго случая изменения вектора входных переменных при прохождении входных сигналов без задержек приведен на рис. 38.

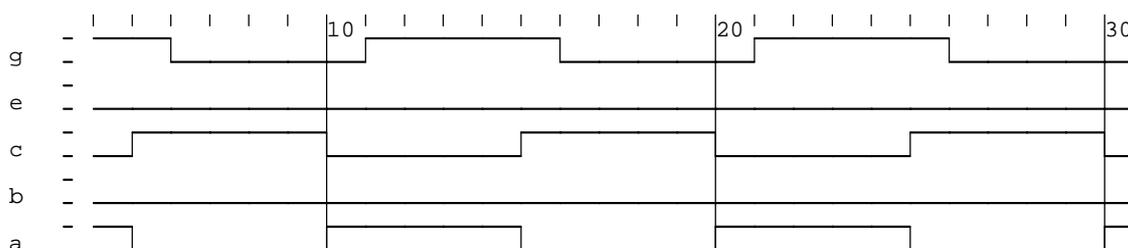


Рис. 38

Результат построения диаграмм выходных сигналов для второго случая изменения вектора входных переменных при прохождении входных сигналов с задержками приведен на рис. 39. Видно, что при не одновременном переключении сигналов на входах  $a$  и  $c$  форма сигналов на выходах  $e$  и  $g$  не изменяется, т. е. отсутствие статического риска сбоя, установленное с использованием трехзначной модели, подтверждается результатами асинхронного анализа в *CircuitMaker*. Для полного подтверждения результатов анализа на трехзначной модели следует построить еще одну диаграмму: для случая, когда в результате задержек сигнал  $c$  переключается раньше, чем сигнал  $a$ .

В третьем случае для данной схемы вектор входных сигналов изменяет в течение такта значение с исходного  $(a, b, c) = (1,0,1)$  на окончательное  $(0,1,1)$ . Расчет на трехзначной модели показывает для этого случая наличие статического риска сбоя только по выходу  $e$  (см. файл *Пример (ЛР№1).ckt*).

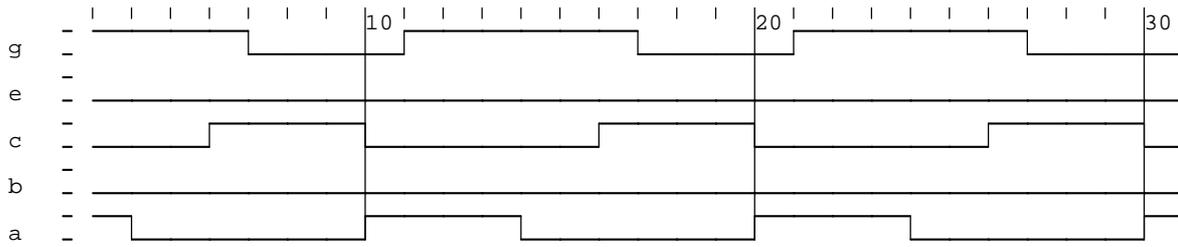


Рис. 39

Результат построения диаграмм выходных сигналов для третьего случая изменения вектора входных переменных (см. файл *Пример (ЛР№1).ckt*) при прохождении входных сигналов без задержек, имеет вид, представленный на рис. 40.

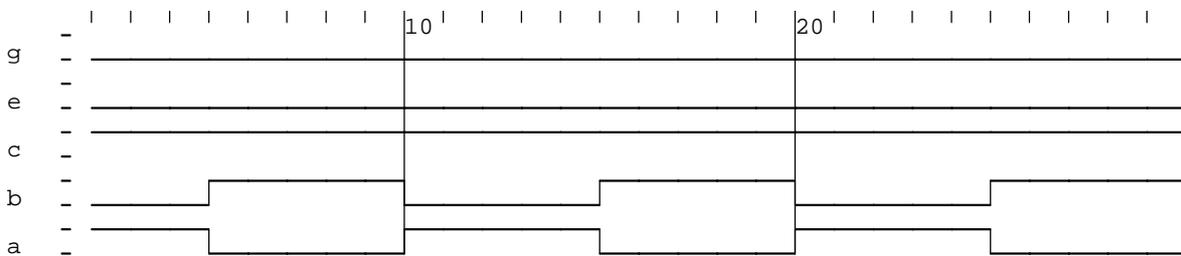


Рис. 40

Результат построения диаграмм выходных сигналов для третьего случая изменения вектора входных переменных при прохождении входных сигналов с задержками показан на рис. 41. Видно, что при не одновременном переключении сигналов на входах *a* и *b* из 1 в 0 и из 0 в 1 соответственно, при сохранении значения 1 на входе *c*, ложный сигнал появляется только на выходе *e*, т. е. статический сбой имеет место только по этому выходу, что совпадает с результатами анализа на трехзначной модели.

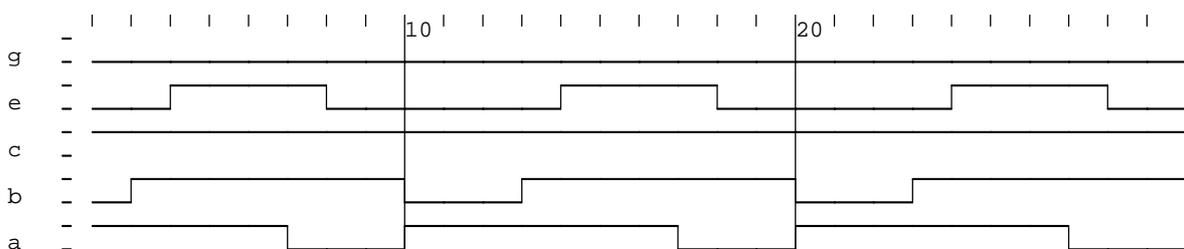


Рис. 41

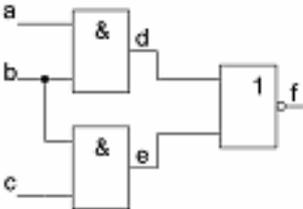
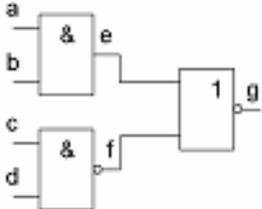
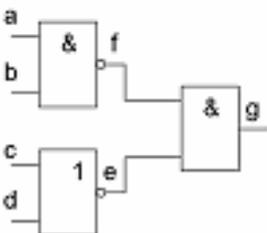
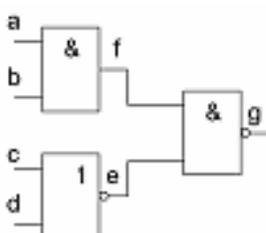
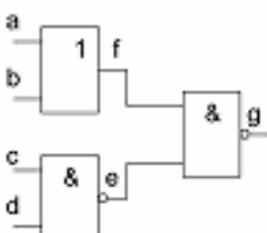
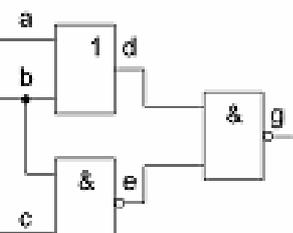
### Варианты индивидуальных заданий

Варианты индивидуальных заданий приведены в таблице 17. Значения входных переменных заданы в столбцах таблицы алфавитном порядке: первый столбец соответствует входному сигналу  $a$ , второй –  $b$  и т. д.

Три случая изменения вектора входных переменных рассматриваются в соответствии с очередностью перечисления векторов входных сигналов в строчках, а именно:

- в первом случае исследуется переключение вектора входных сигналов со значения, указанного в первой строчке таблицы, на значение, указанное во второй строчке;
- во втором случае исследуется переключение вектора входных сигналов со значения, указанного во второй строчке, на значение, указанное в третьей строчке;
- в третьем случае исследуется переключение вектора входных сигналов со значения, указанного в третьей строчке, на значение, указанное в четвертой строчке.

Таблица 17

Номер варианта и схема	Вектор входа			Номер варианта и схема	Вектор входа				
1. 	1	0	0	2. 	0	0	0	1	
	0	1	0		0	0	1	0	0
	1	1	1		1	1	0	0	0
	0	0	1		1	0	1	1	0
3. 	1	0	0	1	4. 	0	1	0	0
	0	1	0	0		1	0	0	0
	0	0	1	0		1	1	1	1
	1	0	1	1		1	0	1	0
5. 	0	1	0	1	6. 	1	0	0	
	0	0	1	0		1	1	1	
	1	0	1	1		0	1	0	
	0	1	1	0		0	0	1	

<p>7.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> </table>	1	0	0	0	0	1	1	1	0	0	1	1	<p>8.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0				
1	0	0																																	
0	0	1																																	
1	1	0																																	
0	1	1																																	
1	0	0	0																																
0	1	0	0																																
0	0	1	1																																
1	0	0	0																																
<p>9.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	1	0	1	1	0	0	0	1	0	1	<p>10.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	0	0	1	0	0	0	1	1	1	1								
1	0	1																																	
0	1	1																																	
0	0	0																																	
1	0	1																																	
1	0	0																																	
0	1	0																																	
0	0	1																																	
1	1	1																																	
<p>11.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	0	1	1	0	1	0	1	0	0	0	0	1	1	0	1	1	<p>12.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	0
0	1	1	0																																
1	0	1	0																																
0	0	0	1																																
1	0	1	1																																
0	1	0	0																																
1	0	1	0																																
1	0	0	1																																
1	1	1	0																																
<p>13.</p>	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	0	1	1	0	1	0	1	1	0	0	1	1	0	1	0	<p>14.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table>	0	1	0	1	0	0	1	1	1	1	0	0	0	1	1	1
0	0	1	1																																
0	1	0	1																																
1	0	0	1																																
1	0	1	0																																
0	1	0	1																																
0	0	1	1																																
1	1	0	0																																
0	1	1	1																																
<p>15.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	<p>16.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> </table>	1	0	1	1	1	0	0	0	0	0	1	1				
1	0	1	0																																
1	1	0	0																																
1	0	0	1																																
0	0	0	0																																
1	0	1																																	
1	1	0																																	
0	0	0																																	
0	1	1																																	
<p>17.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	0	1	0	1	1	0	0	1	0	1	0	0	0	0	1	0	<p>18.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	1	0	0	0	0	1	1	1	0	1				
0	1	0	1																																
1	0	0	1																																
0	1	0	0																																
0	0	1	0																																
1	0	1																																	
0	0	0																																	
0	1	1																																	
1	0	1																																	

<p>19.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	0	0	0	1	0	1	0	1	1	1	<p>20.</p>	<table border="1"> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	1	1	0	1	0	1	1	1	0	0	0	1	0	1	0	0				
1	0	0																																	
0	0	1																																	
0	1	0																																	
1	1	1																																	
1	1	0	1																																
0	1	1	1																																
0	0	0	1																																
0	1	0	0																																
<p>21.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> </table>	1	0	0	0	0	1	1	0	0	0	1	1	<p>22.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	1	0	1	1	1	1	0	0								
1	0	0																																	
0	0	1																																	
1	0	0																																	
0	1	1																																	
1	0	0																																	
0	1	0																																	
1	1	1																																	
1	0	0																																	
<p>23.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	0	1	1	0	1	1	0	0	0	1	0	1	0	1	1	0	<p>24.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	1	0	1	1	0	0	1	1	0	1	0	1	1	0	0
0	1	1	0																																
1	1	0	0																																
0	1	0	1																																
0	1	1	0																																
0	1	0	1																																
1	0	0	1																																
1	0	1	0																																
1	1	0	0																																
<p>25.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	1	0	1	0	1	0	0	0	<p>26.</p>	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </table>	1	0	0	0	1	0	1	0	0	0	0	1								
1	0	0																																	
0	1	0																																	
1	0	1																																	
0	0	0																																	
1	0	0																																	
0	1	0																																	
1	0	0																																	
0	0	1																																	
<p>27.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	0	1	1	0	1	1	0	0	0	1	0	1	0	1	1	0	<p>28.</p>	<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	1	0	1	1	0	0	1	1	0	1	0	1	1	0	0
0	1	1	0																																
1	1	0	0																																
0	1	0	1																																
0	1	1	0																																
0	1	0	1																																
1	0	0	1																																
1	0	1	0																																
1	1	0	0																																

## Лабораторная работа №2. Моделирование структурных примитивов. Решение задач управления и идентификации

*Цели:* Часть 1 – решение задачи управления: моделирование и анализ поведения структурного примитива в динамическом режиме. Часть 2 – решение задачи идентификации: разработка многофакторной регрессионной модели по статистике функционирования структурного примитива.

*Программное обеспечение моделирования:* CircuitMaker, программа моделирования систем автоматического регулирования ТАУ, MS Excel.

*Примеры выполнения работы:* Часть 1 – Пример (ЛР№2).zvp; Пример (ЛР№2).ckt; Часть 2 – Пример (ЛР2).xls

### Теория

Основные сведения о постановке и методах решения задачи управления – см. раздел 2.3.1. конспекта лекций.

Элементы систем, описываемые дифференциальными уравнениями и, соответственно, передаточными функциями не выше второго порядка, называются типовыми элементарными звеньями. Введение элементарных звеньев дает возможность свести многообразие различных технических устройств к небольшому количеству типовых звеньев, что позволяет использовать единые методы анализа для различных систем. Основные типы элементарных звеньев и их передаточные функции приведены в таблице 18. Передаточная функция последовательного соединения звеньев определяется как произведение передаточных функций звеньев.

Таблица 18

Тип звена	Передаточная функция W(p)
Усилительное безинерционное	$k$
Апериодическое первого порядка (устойчивое)	$\frac{k}{(Tp + 1)}$
Апериодическое второго порядка	$\frac{k}{(T_2^2 p^2 + T_1 p + 1)} = \frac{k}{(T_3 p + 1)(T_4 p + 1)}$ , $T_{3,4} = \frac{T_1}{2} \pm \sqrt{\frac{T_1^2}{4} - T_2^2}$
Колебательное устойчивое	$\frac{k}{(T_2^2 p^2 + T_1 p + 1)}$ или $\frac{k}{(T^2 p^2 + 2\xi Tp + 1)}$
Колебательное гармоническое	$\frac{k}{(T_2^2 p^2 + 1)}$
Идеальное дифференцирующее	$p$
Идеальное интегрирующее	$\frac{k}{p}$

Тип звена	Передаточная функция W(p)
Дифференцирующее первого порядка	$Tp + 1$
Дифференцирующее второго порядка	$T_2^2 p + T_1 p + 1$ или $T^2 p^2 + 2\xi Tp + 1$

Основные сведения о постановке и методах решения задачи идентификации – см. раздел 2.3.2. конспекта лекций.

Пусть экспериментальная статистика функционирования структурного примитива задана таблично. Если статистические данные позволяют предположить наличие многофакторной зависимости  $y(x_1, x_2, \dots, x_n)$ , важным является предварительный анализ исходных данных с целью выработки гипотезы о форме регрессионной функции. Если взаимосвязь целевой функции со всеми факторами носит линейный характер, то регрессионное уравнение может иметь линейную форму:

$$y(x) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n,$$

где определению подлежат коэффициенты  $b_0 \dots b_n$ . В случае нелинейного характера зависимостей целевой функции от факторов в регрессионное уравнение добавляются слагаемые со степенями факторов. Если предполагается наличие только квадратичной зависимости, то уравнение имеет вид:

$$y(x) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n + b_{11} x_1^2 + b_{22} x_2^2 + \dots + b_{nn} x_n^2.$$

Если же предполагаемая зависимость целевой функции от одного, нескольких или всех факторов имеет более высокую степень, то в регрессионное уравнение добавляются соответствующие слагаемые.

Наиболее сложный для анализа случай представляет задача, в которой помимо зависимости между целевой функцией и заданным набором переменных наблюдается взаимосвязь между факторами (попарно или в более сложных комбинациях). Для учета подобных влияний в регрессионную функцию вводят слагаемые вида: для парного влияния  $b_{ij} x_i x_j$ ; для взаимного влияния трех факторов  $b_{ijk} x_i x_j x_k$ , и так далее.

В качестве критерия точности расчета коэффициентов регрессионного уравнения, и, соответственно, точности искомой регрессионной функции, можно использовать минимум среднеквадратичного отклонения расчетных значений целевой функции от фактических. В случае фиксирован-

ного размера исходной статистической выборки также используют минимум дисперсии адекватности:

$$\bar{S}_{ad}^2 = \frac{\sum_{i=1}^n (y_i - y_{i_{расч}})^2}{n - p - 1},$$

где  $y_i$  – фактическое значение;  $y_{i_{расч}}$  – расчетное значение;  $n$  – размер выборки;  $p$  – число факторов.

Исходными данными к моделированию является параметрическая статистика функционирования устройства, приведенная в файле *Варианты индивидуальных заданий (часть 2).xls*. Ограничения, учитываемые при разработке факторной регрессионной модели:

– точность модели, определяемая заданным значением  $K$  квадрата коэффициента корреляции ( $R^2 \geq K$ ) между фактическими значениями выходных (зависимых) параметров и их значениями, рассчитанными на модели;

– заданная форма зависимости выходных параметров функционирования от входных (независимых) факторов:

$$y_k = b_0 + \sum_{i=1}^n b_i x_i + \sum_{i=1}^n b_{ii} x_i^2 + \sum_{\substack{i,j=1, \\ i \neq j}}^n b_{ij} x_i x_j, \quad (11)$$

где  $y_k$  –  $k$ -ый выходной параметр системы;  $b_0, b_i, b_{ii}, b_{ij}$  – коэффициенты модели, подлежащие определению на основании параметрической статистики;  $n$  – число входных факторов системы;  $x_i, x_j$  – входные факторы;

– требование обязательной минимизации числа слагаемых в формуле (11). Минимизация выполняется до выполнения условия  $R^2 \approx K$ .

## **Ход выполнения работы**

### *Часть 1:*

**1.** Для заданного варианта схемы подобрать номиналы элементов таким образом, чтобы постоянные времени передаточной функции были порядка 0,01...4,2 сек. По виду передаточной функции определить, последовательным соединением каких типовых элементарных звеньев можно моделировать заданный вариант схемы. Вывести дифференциальное уравнение схемы.

**2.** Построить переходной процесс схемы в программе моделирования систем автоматического регулирования ТАУ.

**3.** Смоделировать заданную схему в *CircuitMaker*. Построить переходной процесс. Сравнить результат с моделированием в ТАУ.

4. Результаты выполнения работы по пунктам 1-3 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

Часть 2:

5. Осуществить анализ исходных данных и выдвинуть гипотезу, обосновывающую конкретный вид регрессионного уравнения модели по общему виду (11). Рассчитать коэффициенты регрессионного уравнения модели.

6. Рассчитать корреляцию между исходными и рассчитанными на модели значениями целевой функции; в случае неудовлетворительной точности ( $R^2 < K$ ) скорректировать исходную гипотезу.

7. Обосновать число слагаемых итогового выражения модели по общему виду (11). По возможности минимизировать число слагаемых до выполнения условия  $R^2 \approx K$ .

8. Результаты выполнения работы по пунктам 5-7 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

### Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования

Часть 1.

Рассмотрим пример выполнения работы для исходных данных, приведенных на рис. 42.

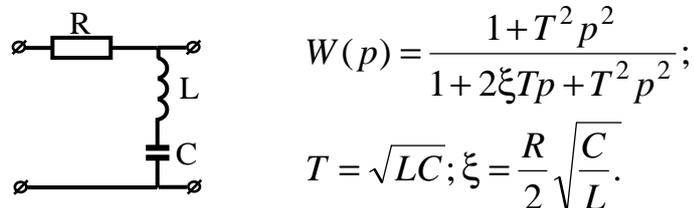


Рис. 42

Передаточная функция заданной схемы может быть представлена в виде:

$$W(p) = \frac{1 + T^2 p^2}{1 + 2\xi T p + T^2 p^2} = \frac{1}{1 + 2\xi T p + T^2 p^2} \cdot (1 + T^2 p^2),$$

следовательно, в терминах моделей управления данная схема может быть представлена последовательным соединением типовых элементарных звеньев: колебательного устойчивого и дифференцирующего второго порядка. Зададим номиналы элементов  $R=2$  Ом;  $L=0,1$  Н;  $C=0,025$  Ф. Постоянные передаточной функции, таким образом, будут равны:  $T=0,05$ ;  $\xi=0,5$ . Построим переходной процесс звена с использованием программного обеспечения моделирования.

### *Программа моделирования TAU.*

Запуск программы осуществляется файлом *tau.exe*. Выбор типовых элементарных звеньев моделируемой системы, источника входного сигнала и устройства вывода переходных процессов (экран) осуществляется в режиме графического редактора схем – пункт меню *Редактировать/Создать*. Для задания параметров элементов и звеньев используйте двойной щелчок. Двойной щелчок по устройству вывода – экрану – разворачивает его для просмотра результатов моделирования в отдельное окно.

Для соединения элементов схемы выделите элемент-источник щелчком мыши, затем – выход элемента-источника, элемент-приемник и вход элемента приемника. Для перемещения элемента его следует выделить однократным щелчком, а затем повторно нажать левую кнопку мыши, и, не отпуская ее, «перетащить» элемент.

Для задания интервала времени, на котором будет осуществляться моделирование переходного процесса, используйте пункт меню *Конфигурация*. Время моделирования следует выбирать в несколько раз большим, чем наибольшая постоянная времени схемы. Критерием корректности выбора времени моделирования служит вид графика переходного процесса – он должен полностью затухать (асимптотически приближаться к установившемуся значению). Шаг детализации следует оставлять по умолчанию.

Запуск моделирования осуществляется из меню *Операции/Старт*. После завершения режима расчета результаты моделирования можно просмотреть, развернув устройство вывода – экран – в отдельное окно.

*Внимание!* В звеньях второго порядка ограничение параметра  $\xi < 1$  является рекомендательным и может не выполняться.

Для рассматриваемого примера модель заданной схемы, реализованная в виде последовательного соединения типовых звеньев, в программе TAU имеет вид, приведенный на рис. 43 (см. также файл *Пример (ЛРН№2).zvn*), а задание параметров звеньев и результат моделирования – на рис. 44. Как видно, время переходного процесса до достижения значения, отличающегося от установившегося не более чем на 5% составляет 0,303 сек, а время полного затухания переходного процесса составляет от 0,6 до 0,7сек.

### *Circuit Maker PRO.*

Основные сведения о работе – см. *Лабораторную работу №1*.



Рис. 43

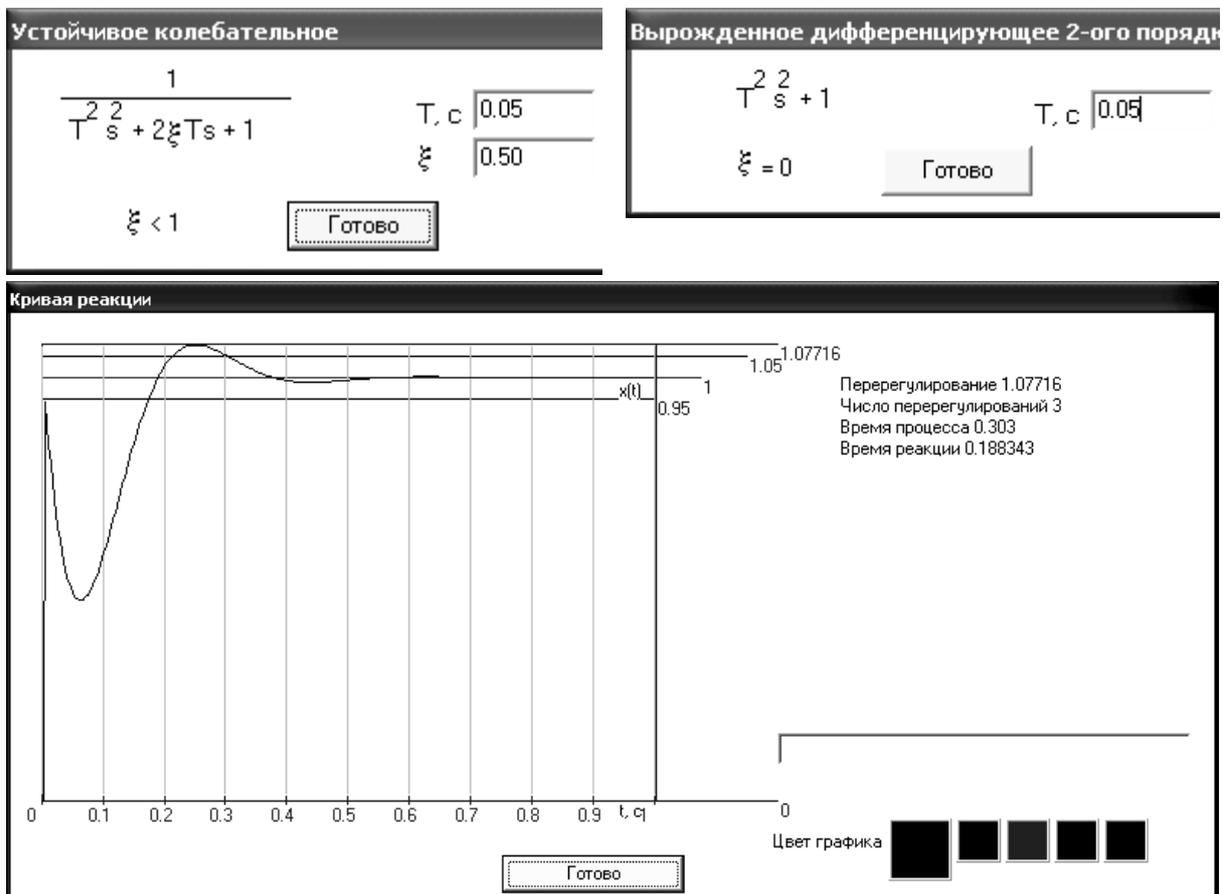


Рис. 44

Схема для моделирования переходного процесса для заданных исходных данных в *CircuitMaker* показана на рис. 45 (см. также файл *Пример (LP№2).ckt*). В качестве источника входного напряжения выбираем

генератор гармонического сигнала в базе данных элементов Analog/Instruments/Signal Gen и задаем прямоугольную форму сигнала, включив двойным щелчком по источнику режим редактирования и выбрав параметры Wave.../Pulse.

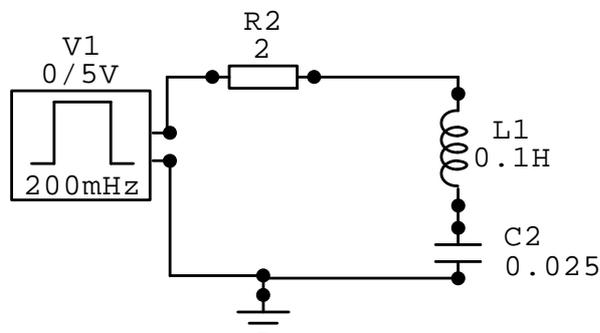


Рис. 45

При задании временных параметров входного сигнала следует учитывать временные характеристики схемы, а также то, что условия моделирования в *tau.exe* и *CircuitMaker* должны быть абсолютно идентичны. Как следует из результатов моделирования для заданного варианта в программе *tau.exe*, для того, чтобы схема работала устойчиво (переходной процесс завершался полностью до очередного изменения входного сигнала), необходимо, чтобы длительность прямоугольного импульса была равна 1 сек. Зададим параметры генератора как показано на рис. 46, где Pulse Width – ширина (длительность) импульса в значении 1, в секундах – должна составлять 0,5 периода. Параметры Rise Time/Fall Time задают длительность переключения периодического сигнала из 0 в 1 и наоборот. Для моделирования сигнала прямоугольной формы их следует оставить по умолчанию в значениях 1 наносекунда.

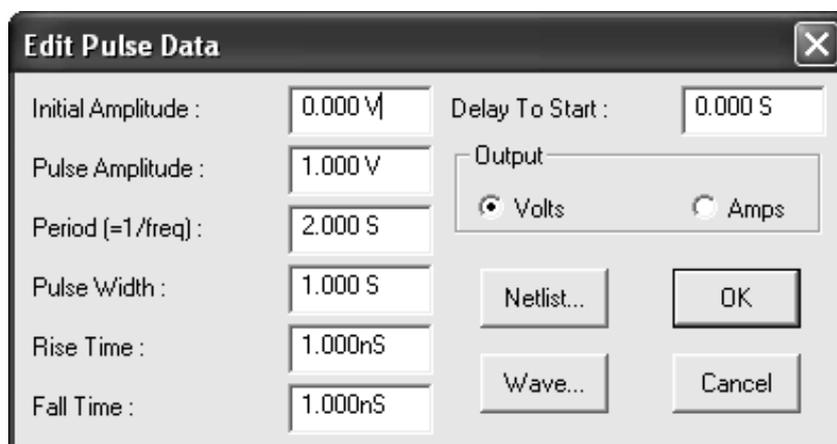


Рис. 46

Выбор режимов моделирования осуществляется в диалоговом меню **Analyses Setup**. Для включения режима анализа переходных характеристик следует выбрать режим **Transient/Fourier** (см. рис. 47).

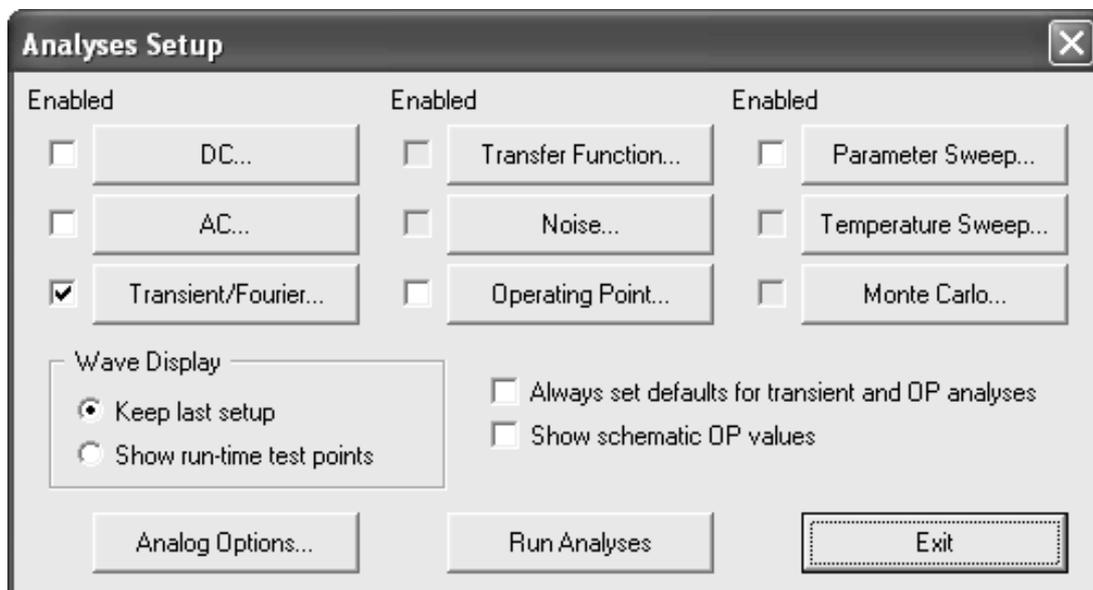


Рис. 47

Далее, нажатием кнопки **Transient/Fourier** задаем параметры моделирования как показано на рис. 48.

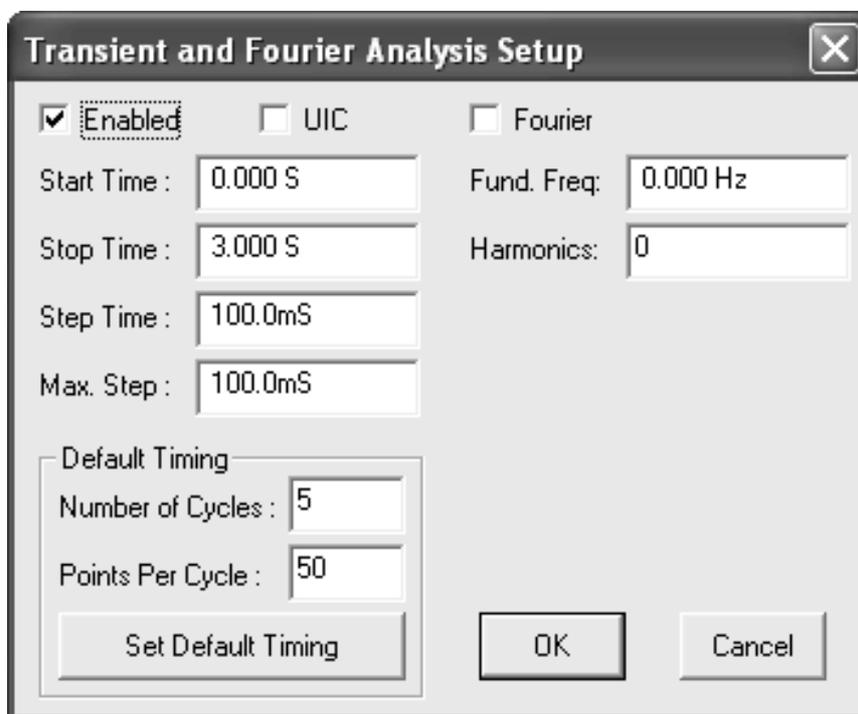


Рис. 48

Параметры Start/Stop Time (см. рис. 48) задают начальное/конечное значение общего времени моделирования. В нашем случае 3-х секунд достаточно, т.к. длительность прямоугольно импульса для гарантии завершения переходного процесса полностью выбрана равной 1 секунде. Параметр Step Time задает шаг приращения времени – его оптимальное значение можно задать автоматически, нажатием кнопки Set Defaults.

Для запуска расчета характеристик воспользуемся кнопкой Run Analyses в меню Analyses Setup или кнопкой  панели инструментов. Выбор на схеме точки снятия переходной характеристики осуществляется инструментом . Переходную характеристику следует снимать с выхода схемы. Результат анализа для заданного варианта имеет вид, показанный на рис. 49.

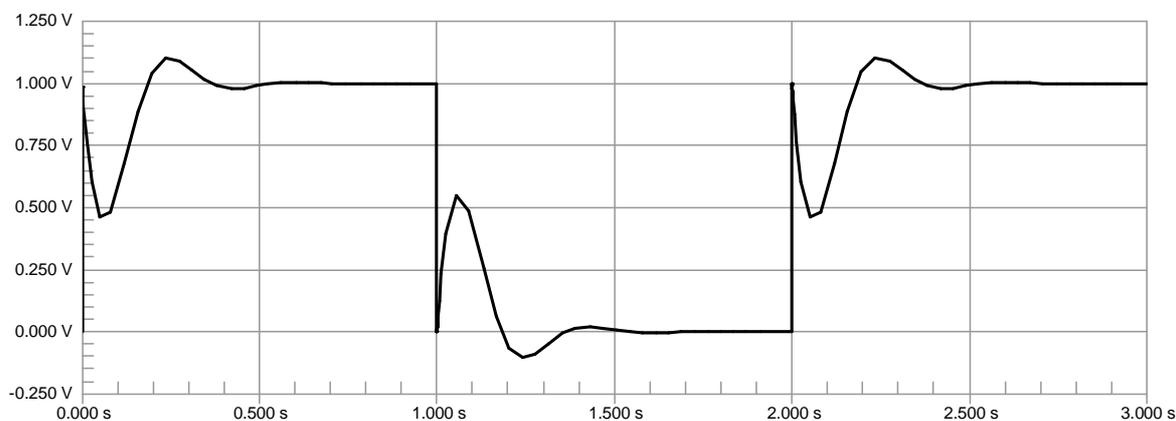


Рис. 49

Как видно из результатов, представленных на рис. 49, полученный график по переднему фронту прямоугольного входного сигнала, который соответствует ступеньке, в точности совпадает с результатам моделирования переходных характеристик корректора в программе *tau.exe*. Таким образом, результаты моделирования схемы в *Circuit Maker*, имитирующие натуральный эксперимент, идентичны результатам функционального моделирования с использованием модели корректора в виде передаточной функции. Для точной проверки соответствия результатов моделирования в *Circuit Maker*, например для определения длительности переходного процесса или времени достижения максимума, следует использовать измерительные курсоры (Measurement Cursors), режим использования которых включается в левой части рабочего окна в режиме вывода графиков (см. рис. 50). Как видно, координаты позиционирования измерительных курсоров позволяют точно определить время достижения переходной характе-

ристикой максимума:  $\sim 0,247$  секунд, и время завершения переходного процесса:  $\sim 0,690$  секунд, что удобно для сопоставительного сравнения с результатами моделирования в *tau.exe*.

Для копирования и вставки в отчет схем и графиков, построенных в *Circuit Maker*, следует использовать меню Edit/Copy to Clipboard.

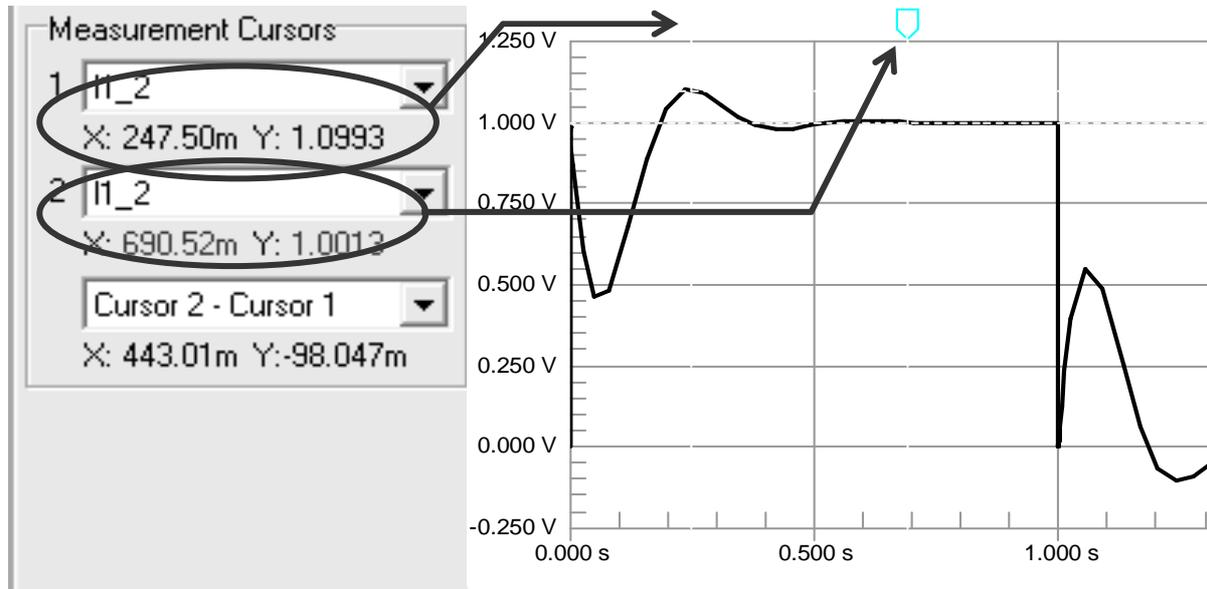


Рис. 50

*Наиболее часто возникающие ошибки моделирования и способы их исправления*

Если текст сообщения в специальном окне *Circuit Maker XSPICE For Windows* выглядит, как показано на рис. 51, это значит, что получена сингулярная (некорректная) матрица коэффициентов системы уравнений схемы, получаемой по законам токов и напряжений Кирхгофа, и используемой для моделирования всех характеристик схемы. Чаще всего ошибка возникает в схемах, критичных к сопротивлению в цепи заземления.

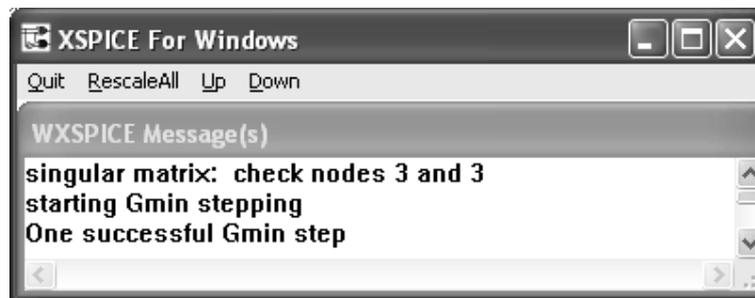


Рис. 51

*Способ исправления:* установить в параметрах моделирования Analyses Setup/Analog Options очень большое значение сопротивления в цепи заземления (самый последний пункт списка, в окне Analog Options выделен цветом). Значение вводится в окне ввода Option Value, после чего для подтверждения изменений нужно нажать кнопку Enter в правом верхнем углу окна Analog Options (см. рис. 52).

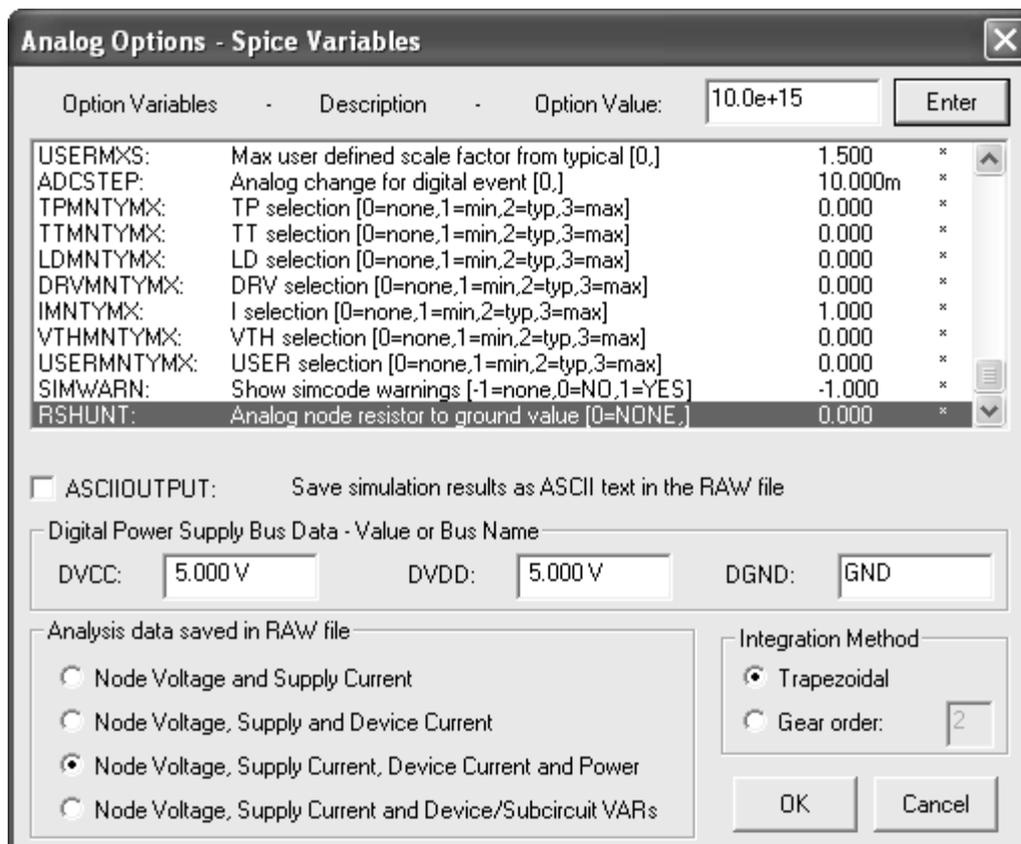


Рис. 52

Если текст сообщения в специальном окне *Circuit Maker XSPICE For Windows* выглядит, как показано на рис. 53, это значит, что моделирование переходного процесса прервано, так как шаг приращения времени слишком мал.



Рис. 53

*Способ исправления:* увеличить шаг приращения времени, изменив значения в полях Step Time/Max.Step в параметрах моделирования Analyses Setup/ TransientFourier.

Если текст сообщения в специальном окне tau.exe выглядит, как показано на рис. 54, это означает, что при редактировании схемы в таблице звеньев не осуществлен выбор звена – ни одна позиция не выделена.



Рис. 54

*Способ исправления:* перезапустить программу и при выборе звеньев в режиме редактирования обязательно выделять выбираемую позицию в списке щелчком мыши как показано на рис. 55.

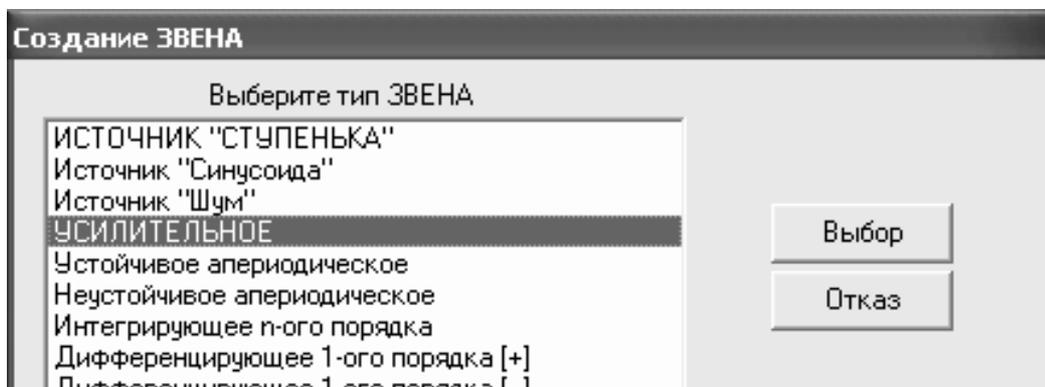


Рис. 55

### *Часть 2 – MS Excel.*

Основные начальные сведения по работе в *MS Excel* приведены в методических указаниях к *Лабораторной работе №1*.

Расчеты по работе осуществляются с использованием встроенных средств поиска решений *MS Excel* (меню СЕРВИС/ПОИСК РЕШЕНИЯ). Пусть исходные данные к моделированию имеют вид, приведенный в таблице 19 (см. также файл *Пример (ЛР№2).xls*). Необходимо рассчитать регрессию  $y(x_1, x_2, x_3)$ , которая обеспечивала бы точность аппроксимации не менее 0,95.

Как следует из заданного вида регрессионной функции (11), кроме первых степеней всех 3-х факторов, целевая функция  $y$  также может зависеть от вторых степеней факторов и их попарного взаимного влияния. Дополним таблицу исходных данных соответствующими расчетами (см. файл *Пример (ЛРН№2).xls*) и оценим корреляцию целевой функции  $y$  со всеми факторами ( $x_i$ ), вторыми степенями факторов ( $x_j$ ) и их попарными взаимными влияниями ( $x_i \cdot x_j, i \neq j$ ).

Таблица 19

№ пп	Фактор			Целевая функция
	$x_1$	$x_2$	$x_3$	$y$
1	10	4	3	1
2	12	5	3,1	2
3	25	6	3,2	2
4	13	6	3,3	3
5	24	7	3,3	3
6	14	7	3,3	4
7	15	3	3,4	5
8	21	3	3,4	6
9	23	7,5	3,5	6
10	16	8	3,5	7
11	22	2	3,55	7
12	17	2	3,6	7,5
13	19	8,5	3,61	8
14	20	1	3,62	8,5
15	18	9	3,65	9

Для оценки воспользуемся функцией *MS Excel*. КОРРЕЛ. Эта функция имеет формат:

$$=КОРРЕЛ(\text{массив1};\text{массив2})$$

где *массив1* и *массив2* – интервалы значений параметров, для которых рассчитывается коэффициент корреляции. Интервалы значений в *MS Excel* задаются через двоеточие, т. е. если рассматривается столбец значений, начинающийся в ячейке A11 и заканчивающийся в ячейке A23, то ссылка на него в формуле будет иметь вид A11:A23. Результат корреляционного анализа по заданному примеру имеет вид, приведенный на рис. 56.

Как видно из полученных результатов, целевая функция  $y$  в значительной степени зависит от третьего фактора ( $R(x_3, y) = 0,97$ ), второй степени третьего фактора ( $R(x_3^2, y) = 0,98$ ) и, в меньшей степени, от взаимного

влияния первого и третьего факторов ( $R(x_1, x_3, y) = 0,47$ ). Следовательно, с большой вероятностью можно предложить следующую гипотезу в отношении вида зависимости целевой функции от заданных факторов:

$$y = b_0 + b_3 x_3 + b_{33} x_3^2 + b_{13} x_1 x_3,$$

где  $b_0$ ,  $b_3$ ,  $b_{33}$  и  $b_{13}$  – коэффициенты регрессионного уравнения, которые требуется определить.

№ пп	$x_1$	$x_2$	$x_3$	$x_1^2$	$x_2^2$	$x_3^2$	$x_1 * x_2$	$x_2 * x_3$	$x_1 * x_3$	$y$
1	10	4	3	100	16	9	40	12	30	1
2	12	5	3,1	144	25	9,61	60	15,5	37,2	2
3	25	6	3,2	625	36	10,24	150	19,2	80	2
4	13	6	3,3	169	36	10,89	78	19,8	42,9	3
5	24	7	3,3	576	49	10,89	168	23,1	79,2	3
6	14	7	3,3	196	49	10,89	98	23,1	46,2	4
7	15	3	3,4	225	9	11,56	45	10,2	51	5
8	21	3	3,4	441	9	11,56	63	10,2	71,4	6
9	23	7,5	3,5	529	56,25	12,25	172,5	26,25	80,5	6
10	16	8	3,5	256	64	12,25	128	28	56	7
11	22	2	3,55	484	4	12,6025	44	7,1	78,1	7
12	17	2	3,6	289	4	12,96	34	7,2	61,2	7,5
13	19	8,5	3,61	361	72,25	13,0321	161,5	30,685	68,59	8
14	20	1	3,62	400	1	13,1044	20	3,62	72,4	8,5
15	18	9	3,65	324	81	13,3225	162	32,85	65,7	9
<b>Коррел.:</b>	<b>0,30</b>	<b>-0,03</b>	<b>0,97</b>	<b>0,20</b>	<b>0,16</b>	<b>0,98</b>	<b>0,05</b>	<b>0,08</b>	<b>0,47</b>	

Рис. 56

Дополним рабочий лист расчета таблицей значений коэффициентов регрессионной функции  $b_0$ ,  $b_3$ ,  $b_{33}$  и  $b_{13}$ . Зададим начальные значения коэффициентов произвольно (например, все равные 1). Добавим к таблице исходных значений столбец расчетных значений  $y$  и столбец значений квадрата отклонения реальных значений целевой функции от расчетных  $(y - y_{расч})^2$ . Рассчитаем значение дисперсии адекватности – для произвольных начальных значений коэффициентов эта величина, как и квадраты отклонений, будут иметь, очевидно, неудовлетворительно большие значения.

Осуществим с использованием встроенных средств поиска решений *MS Excel* (меню СЕРВИС/ПОИСК РЕШЕНИЯ) подбор значений коэффициентов регрессионной функции  $b_0$ ,  $b_3$ ,  $b_{33}$  и  $b_{13}$ , обеспечивающих минимум дисперсии адекватности, и, соответственно, максимум точности регрессионной функции. Для этого зададим параметры поиска решения как показано на рис. 57. Целевой ячейкой является ячейка с результатом расчета дисперсии, значение которой подлежит минимизации. Изменяемыми ячейками являются ячейки с начальными значениями коэффициентов. Для осуществления поиска решения необходимо, чтобы формула в целевой ячейке была связана с изменяемыми ячейками. Результат расчета для заданного примера имеет вид, показанный на рис. 58 (см. файл *Пример (ЛР№2).xls*).

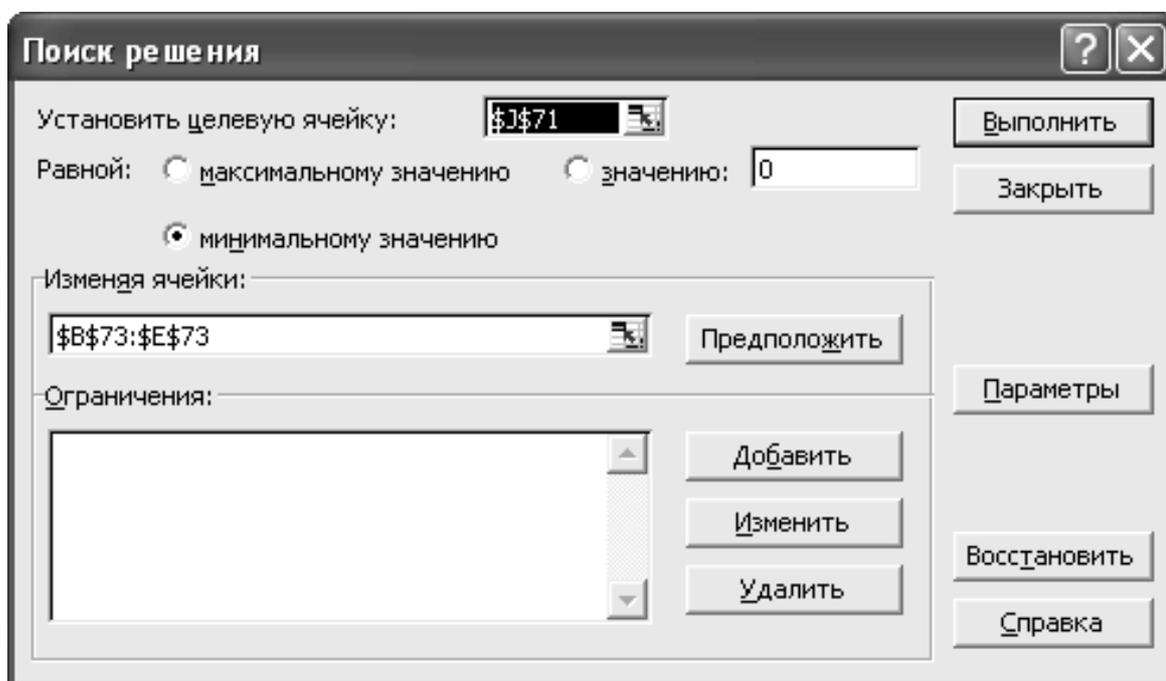


Рис. 57

Из результатов, представленных на рис. 58, видно, что итоговые значения коэффициентов регрессионной функции обеспечивают удовлетворительное совпадение расчетных и реальных значений целевой функции и незначительную дисперсию. Однако для количественной оценки точности полученной регрессионной функции необходимо определить точность аппроксимации.

Microsoft Excel - Пример (ЛРН№2)

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

Arial Cyr 10 Ж К Ч А А

N69 fx

49												
50	<b>2. Расчет коэффициентов регрессии, проверка точности аппроксимации, оптимизация.</b>											
51	2.1 Зададим в ячейках B73:E73 начальные значения коэффициентов (например, все равные 1).											
52	Рассчитаем значения выходной функции в столбце H, квадрат отклонения расчетных значений от											
53	фактических - в столбце I. В качестве целевой ячейки расчета будем использовать J71 (дисперсия).											
54			1	2	3	Целевая функция		Расчетное значение	Квадрат отклонения			
55		№ пп	x1	x2	x3	y		Урасч	(y-Урасч) <sup>2</sup>			
56		1	10	4	3	1		1,000	0,000			
57		2	12	5	3,1	2		1,738	0,069			
58		3	25	6	3,2	2		2,260	0,067			
59		4	13	6	3,3	3		3,808	0,653			
60		5	24	7	3,3	3		3,418	0,175			
61		6	14	7	3,3	4		3,773	0,052			
62		7	15	3	3,4	5		5,038	0,001			
63		8	21	3	3,4	6		4,819	1,396			
64		9	23	7,5	3,5	6		6,205	0,042			
65		10	16	8	3,5	7		6,468	0,283			
66		11	22	2	3,55	7		7,036	0,001			
67		12	17	2	3,6	7,5		8,064	0,318			
68		13	19	8,5	3,61	8		8,159	0,025			
69		14	20	1	3,62	8,5		8,294	0,042			
70		15	18	9	3,65	9		8,904	0,009			
71												Дисперсия
72												0,285
73		b0	b3	b33	b13							
74		54,713	-42,904	8,369	-0,011							

Рис. 58

Построим корреляцию расчетных значений целевой функции с фактическими; задав в параметрах линии тренда вывод коэффициента точности аппроксимации. Для этого следует построить точечную диаграмму зависимости расчетных значений  $y$  от фактических (заданных) значений  $x$ , выделив ряд данных в поле графика, нажать правую кнопку мыши. В появившемся меню нужно выбрать пункт *Добавить линию тренда...* и задать параметры следующим образом: в закладке *Тип* выбрать линейную аппроксимацию; в закладке *Параметры* – выбрать пункт *Поместить на диаграмму величину достоверности аппроксимации*. Результат построения корреляционного графика показан на рис. 59. Как видно, коэффициент точности аппроксимации соответствует заданному ограничению ( $K = R^2 > 0,95$ ).

Для оптимизации полученной регрессионной функции следует последовательно по очереди исключать из полученного выражения слагаемые, начиная со слагаемых с наименьшими по абсолютному значению коэффициентами и повторять расчет для каждого нового варианта. Если после исключения слагаемых точность аппроксимации по-прежнему удовлетворяет заданному ограничению, то новый вариант регрессионной функции принимается.

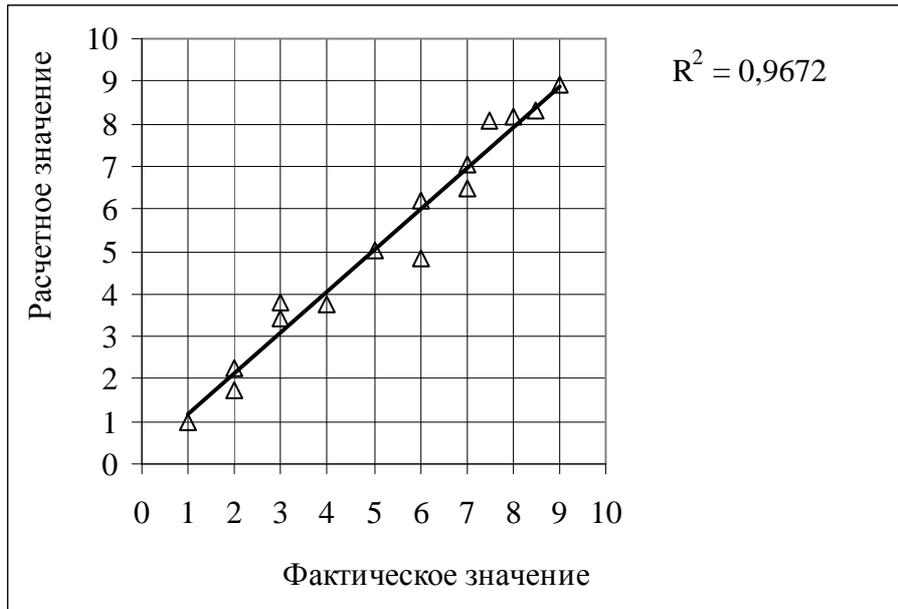


Рис. 59

### Варианты индивидуальных заданий

В таблице 20 приведены индивидуальные варианты по *Части 1* лабораторной работы. Варианты по *Части 2* – см. файл *Варианты индивидуальных заданий (часть 2).xls*.

Таблица 20

№	Схема	Передаточная функция и параметры
1		$\frac{Kp}{1+T_2p} ; K=R_1C_1 ; T_1= R_1C_1 ;$ $T_2=T_1(1+R_2/R_1)$
2		$\frac{Kp}{1+T_2p} ; K=R_1C_2 ; T_1= R_1C_2 ; T_2=R_1(C_2+C_1)$
3		$\frac{(1+T_1p)(1+T_2p)}{(1+T_3p)(1+T_4p)} ;$ $T_{3,4} = \frac{T_1 + aT_2}{2} \left( 1 \pm \sqrt{1 - \frac{4T_1T_2}{(T_1 + aT_2)^2}} \right) ;$ $a = \frac{R_1 + R_2}{R_2} ; T_1= R_1C_1 ; T_2=R_2C_2$

№	Схема	Передаточная функция и параметры
4		$K \frac{(1+T_1 p)}{(1+T_2 p)}; T_1 = R_1 C_2; T_2 = \frac{T_1}{1+C_2/C_1};$ $K = \frac{1}{1+C_2/C_1}$
5		$\frac{K p^2}{1+2\xi T p + T^2 p^2}; T = K = \sqrt{LC}; 2\xi T = RC$
6		$\frac{T_1 p + 1}{T_2 p + 1}; T_1 = R_1 C_2; T_2 = R_1 (C_1 + C_2)$
7		$\frac{K p^2}{1+2\xi T p + T^2 p^2}; T_1 = R_1 C_1; T_2 = R_2 C_2;$ $K = T_1 T_2; T = \sqrt{T_1 T_2}; 2\xi T = T_1 (1 + R_1/R_2) + T_2$
8		$\frac{(1+T_1 p)(1+T_2 p)}{T_1 T_2 p^2 + [T_1(1 + R_2/R_1) + T_2] p + 1};$ $T_1 = R_1 C_1; T_2 = R_2 C_2$
9		$K \frac{(1+T_1 p)}{(1+T_2 p)}; T_1 = R_2 C_1; T_2 = \frac{T_1}{1+R_2/R_1};$ $K = \frac{1}{1+R_2/R_1}$
10		$\frac{(1+T_1 p)(1+T_2 p)}{T_1 T_2 p^2 + [T_1(1 + R_2/R_1) + T_2] p + 1};$ $T_1 = \frac{R_1 R_2}{R_1 + R_2} C_1; T_2 = (R_1 + R_2) C_2$

№	Схема	Передаточная функция и параметры
11		$\frac{(1+T_1p)(1+T_2p)}{T_1T_2p^2 + [T_1(1+R_3/R_1)+T_2]p + 1/K};$ $T_1 = \frac{R_1R_2}{R_1+R_2}C_1; \quad T_2 = R_3C_2;$ $K = \left(1 + \frac{R_3}{R_1+R_2}\right)^{-1}$
12		$K \frac{(1+T_1p)}{(1+T_2p)}; \quad T_2 = T_1(1+R_2/R_1)$ $K = \left(1 + \frac{C_1}{C_2}\right)^{-1}; \quad T_1 = R_1C_1K;$
13		$\frac{K}{1+pT_1};$ $K = \left(1 + \frac{R_2}{R_1}\right)^{-1}; \quad T_1 = R_2C_1K;$
14		$K \frac{(1+T_1p)}{(1+T_2p)};$ $K = 1; \quad T_1 = R_1C_1; \quad T_2 = T_1 \left(1 + \frac{R_2}{R_1}\right)$
15		$\frac{1}{T_1T_2p^2 + [T_1(1+R_2/R_1)+T_2]p + 1};$ $T_1=R_1C_1; \quad T_2=R_2C_2$
16		$\frac{1}{T_1T_2p^2 + T_1p + 1}; \quad T_1=R_1C_1; \quad T_2=L_1/R_1$
17		$K \frac{(1+T_1p)}{(1+T_2p)};$ $T_1 = L_1/R_1; \quad T_2 = \frac{T_1}{1+R_2/R_1}; \quad K = \frac{1}{1+R_2/R_1}$

№	Схема	Передаточная функция и параметры
18		$\frac{K}{T_1 T_2 p^2 + \left[ T_1 + T_2 \frac{R_2}{R_2 + R_1} \right] p + 1};$ $T_2 = L_1 / R_2; T_1 = \frac{R_1 R_2}{R_1 + R_2} C_1; K = \frac{1}{1 + R_2 / R_1}$
19		$\frac{T_1 T_2 p^2 + (T_1 + a) p + 1}{T_1 T_2 p^2 + (T_1 + a + T_2) p + 1};$ $T_1 = R_1 C_1; T_2 = R_2 C_2; a = R_1 C_2$
20		$\frac{(1 + T^2 p^2)}{(1 + T_1 p)(1 + T_2 p)};$ $T = RC; T_1 = T(2 + \sqrt{3}); T_2 = T(2 - \sqrt{3})$
21		$\frac{K}{1 + T_1 p}; K = (1 + R_2 / R_1)^{-1}; T_1 = \frac{L_1}{R_1 + R_2}$
22		$\frac{Kp}{1 + T_2 p};$ $K = T_1 = \frac{R_1 R_2}{R_1 + R_3} C_1; T_2 = \left( R_1 + \frac{R_2 R_3}{R_2 + R_3} \right) C_1$
23		$K \frac{(1 + T_1 p)}{(1 + T_2 p)}; T_2 = \frac{R_1 + R_3}{R_1 + R_2 + R_3} T_1;$ $K = (1 + R_2 / R_1 + R_3 / R_1)^{-1}; T_1 = R_2 C_1$
24		$\frac{(1 + T_1 p)(1 + T_2 p)}{T_1 T_2 \left( 1 + \frac{R_3}{R_1} \right) p^2 + T p + 1};$ $T = \left[ T_2 + T_1 \left( 1 + \frac{R_2}{R_1} + \frac{R_3}{R_1} \right) \right]$ $T_1 = R_1 C_1; T_2 = R_2 C_2.$

№	Схема	Передаточная функция и параметры
25		$\frac{(1 + T_1 p)(1 + T_2 p)}{T_1 T_2 \left( 1 + \frac{R_2 R_3}{R_1 (R_2 + R_3)} \right) p^2 + T p + 1};$ $T = \left[ T_2 + T_1 \left( 1 + \frac{R_2}{R_1} \right) \right]$ $T_1 = R_1 C_1; T_2 = (R_2 + R_3) C_2.$
26		$\frac{T_1 p}{T_1 T_2 p^2 + T_1 \left[ \left( 1 + \frac{R_3}{R_2} \right) \left( 1 + \frac{C_1}{C_2} \right) + \frac{R_3}{R_1} \right] p + 1};$ $T_1 = \frac{R_1 R_2 C_2}{R_1 + R_3}; T_2 = R_3 C_1$
27		$\frac{a T_1 T_2 p^2 + (T_2 + T_1) p + 1}{b T_1 T_2 p^2 + \left[ T_2 + T_1 \left( 1 + \frac{R_3}{R_1} \right) \right] p + 1};$ $a = \left[ 1 + \frac{R_2 R_3}{R_1 (R_2 + R_3 + R_4)} \right];$ $b = \left( 1 + \frac{(R_4 + R_2) R_3}{R_1 (R_2 + R_3 + R_4)} \right)$ $T_1 = R_1 C_1; T_2 = (R_2 + R_3 + R_4) C_2$
28		$K \frac{(1 + T_1 p)}{(1 + T_2 p)}; T_2 = \frac{R_1 + R_3 + \frac{R_2 R_4}{R_2 + R_4}}{R_1 + R_2 + R_3}$ $K = \frac{1}{1 + R_2/R_1 + R_3/R_1}; T_1 = (R_2 + R_4) C_1;$

**Лабораторная работа №3. Статистическое моделирование  
одноканальных СМО безприоритетного  
обслуживания однородного потока заявок**

*Цель:* Экспериментальное определение числа прогонов вероятностной модели, обеспечивающее заданную точность статистического модели-

рования по математическому ожиданию, с использованием метода Монте-Карло. Детерминированный анализ и имитационное статистическое моделирование одноканальной СМО безприоритетного обслуживания.

*Программное обеспечение моделирования:* MS Excel, GPSS World.

### Теория

Основные сведения о *методе Монте-Карло* – см. *раздел 2.4. конспекта лекций*. Основные сведения об *имитационных моделях СМО* – см. *раздел 3.4. конспекта лекций*.

*Моделирование неравномерных распределений.* Для генерирования случайных чисел  $x^{\Gamma}$ , соответствующих гамма-распределению, используют формулу розыгрыша:

$$x^{\Gamma} = -\frac{1}{\lambda} \sum_{i=1}^b \ln(x_i^R),$$

где  $\lambda$  – интенсивность гамма-распределения;  $b$  – целочисленный параметр гамма-распределения.

При моделировании нормально распределенной случайной величины используют центральную предельную теорему [27], согласно которой распределение суммы  $n$  одинаково распределенных независимых случайных величин  $x_1, x_2 \dots x_n$  при неограниченном возрастании  $n$  неограниченно стремится к нормальному распределению. При  $n \geq 8$  распределение этой суммы может считаться нормальным с вероятностью  $P > 0,95$ . Используя реализации случайной равномерно распределенной величины  $x^R$ , можно составить выражение для определения случайной величины  $x^{N_0}$ , имеющей нормальное распределение с параметрами  $m_x=0$  и  $\sigma_x=1$ :

$$x^{N_0} = \sum_{i=1}^{12} x_i^R - 6. \quad (12)$$

На основании формулы (12) из 12-ти случайных равномерно распределенных чисел  $x_i^R$  получается одно случайное число  $x^{N_0}$ , принадлежащее новой совокупности, которая представляет собой случайные числа с нормальным распределением.

Для получения  $x^{N_0}$  также можно использовать формулу:

$$x^{N_0} = 0,774596 \left( \sum_{i=1}^{20} x_i^R - 10 \right).$$

Случайные нормально распределенные числа с заданными параметрами  $m_x$  и  $\sigma_x$  получают из  $x^{N_0}$  по формуле:

$$x^N = m_x + \sigma_x x^{N_0}.$$

### **Ход выполнения работы**

Каждым индивидуальным вариантом задания для исследования задана одноканальная разомкнутая СМО с бесконечной очередью.

**1.** Для заданного варианта СМО определить лимитирующий процесс – процесс поступления или обслуживания заявок с наименьшей интенсивностью. Для лимитирующего процесса разработать алгоритм розыгрыша случайной последовательности по методу Монте-Карло для заданного типа распределения времени поступления заявок входного потока или времени обслуживания в обслуживаемом устройстве. *Если процесс поступления/обслуживания описывается равномерным законом распределения, лимитирующим считать второй процесс, описываемый неравномерным законом распределения.*

**2.** Осуществить реализацию генератора случайных чисел в *MS Excel* для заданного типа распределения. Экспериментально оценить число прогонов, необходимое для обеспечения 95%-ной точности по математическому ожиданию.

**3.** С использованием детерминированного анализа на основании выбранного числа прогонов и математических ожиданий времени поступления и обслуживания заявок приближенно оценить следующие характеристики СМО:

- число заявок, поступивших в СМО;
- число заявок, обслуженных в СМО;
- коэффициент использования (КИ) устройства обслуживания;
- среднюю длину очереди (для СМО с КИ устройства меньше 1);
- среднее время ожидания заявки в очереди (для СМО с КИ устройства меньше 1).

**4.** Реализовать имитационную модель заданного варианта СМО в *GPSS World*. В программной реализации обязательны построчный комментарий.

**5.** Осуществить имитационное моделирование СМО. По результатам моделирования оценить совпадение результатов моделирования с результатами выполнения п.3, а также определить:

- максимальную длину очереди;
- коэффициент использования очереди.

**6.** Построить гистограмму времени ожидания обслуживания.

**7.** Решить задачу оптимизации (модификации) СМО согласно индивидуальному варианту. По результатам моделирования оценить характеристики, перечисленные в п.3 и п.5, для оптимизированного (модифицированного) варианта СМО.

8. Результаты выполнения работы по пунктам 1-7 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

**Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования**

*MS Excel.*

Пусть задана одноканальная разомкнутая СМО с обслуживающим устройством и бесконечной очередью, для которой время поступления заявок распределено по экспоненциальному закону с интенсивностью  $\lambda_n=6,66$ , а время обслуживания – по экспоненциальному закону с интенсивностью  $\lambda_o=10$ . Математическое ожидание для этого типа распределения есть величина обратная интенсивности, следовательно заявки в СМО поступают в среднем через  $MO_n=0,15$  секунд, а на обслуживание тратится в среднем  $MO_o=0,1$  секунда, т. е. заявки обслуживаются быстрее, чем поступают. Следовательно, в данной СМО лимитирующим является процесс поступления заявок.

Требуется экспериментально установить число реализаций (прогнозов) случайного процесса генерации времени обслуживания в устройстве, гарантирующее имитационной модели 95%-ю точность по математическому ожиданию. В нашем случае при  $MO=0,15$  это означает, что нужно установить с какой реализации случайного процесса  $MO$  случайной величины гарантированно не выходит за пределы диапазона  $0,1425\dots 0,1575$ , т. е. отклонение рассчитываемого по экспериментальным реализациям случайного процесса значения  $MO$  от заданного значения не превышает 5%.

Для произвольного числа реализаций  $N$  случайного процесса  $MO$  вычисляется по формуле:

$$MO_N = \frac{(N-1)MO_{N-1}}{N} + \frac{x_N}{N}.$$

При программной реализации следует с использованием метода Монте-Карло реализовать генератор случайных чисел по заданному закону распределения с расчетом  $MO$  после каждой генерации случайного числа и определением порядкового номера генерации, начиная с которой расчетное значение  $MO$  больше не выходит за пределы 5%-го отклонения. Реализация расчета с использованием других математических пакетов выполняется аналогично.

Для заданного примера фрагмент рабочего листа с реализованными формулами в *MS Excel* имеет вид, представленный на рис. 60-а (в ячейке В3 содержится значение интенсивности), а результат расчета для 1000 реализаций случайного процесса показан на рис. 60-б.

Графическое представление двух экспериментов по 3000 прогонов случайной величины приведено на рис. 61 и 62 (нижний и верхний предел 5%-го отклонения МО от заданного значения показан жирными линиями). Видно, что в первом эксперименте экспериментальное значение числа прогонов  $N_9$ , начиная с которого МО больше не выходило за пределы диапазона допустимого отклонения, составило порядка 1050 генераций, а во втором эксперименте – ровно 1000. Для гарантированного обеспечения 95%-ной точности по МО при испытаниях имитационной модели число прогонов следует выбрать бóльшим, например 1500.

а)

	А	В	С	В
5	№	Рандом	Экспериментальные значения случайной величины	Экспериментальное МО
6	1	=СЛЧИС()	=-LN(B6)/\$B\$3	=C6
7	=A6+1	=СЛЧИС()	=-LN(B7)/\$B\$3	=D6*A6/A7+C7/A7
8	=A7+1	=СЛЧИС()	=-LN(B8)/\$B\$3	=D7*A7/A8+C8/A8
9	=A8+1	=СЛЧИС()	=-LN(B9)/\$B\$3	=D8*A8/A9+C9/A9
...				
1004	=A1003+1	=СЛЧИС()	=-LN(B1004)/\$B\$3	=D1003*A1003/A1004+C1004/A1004
1005	=A1004+1	=СЛЧИС()	=-LN(B1005)/\$B\$3	=D1004*A1004/A1005+C1005/A1005

б)

№	Рандом	Экспериментальные значения случайной величины	Экспериментальное МО
1	0,2813747	0,190210196	0,190210196
2	0,5924539	0,07852232	0,134366258
3	0,9276752	0,011261034	0,093331183
4	0,4386575	0,123605445	0,100899749
...			
999	0,9803922	0,002970382	0,157943337
1000	0,6434532	0,066135888	0,15785153

Рис. 60

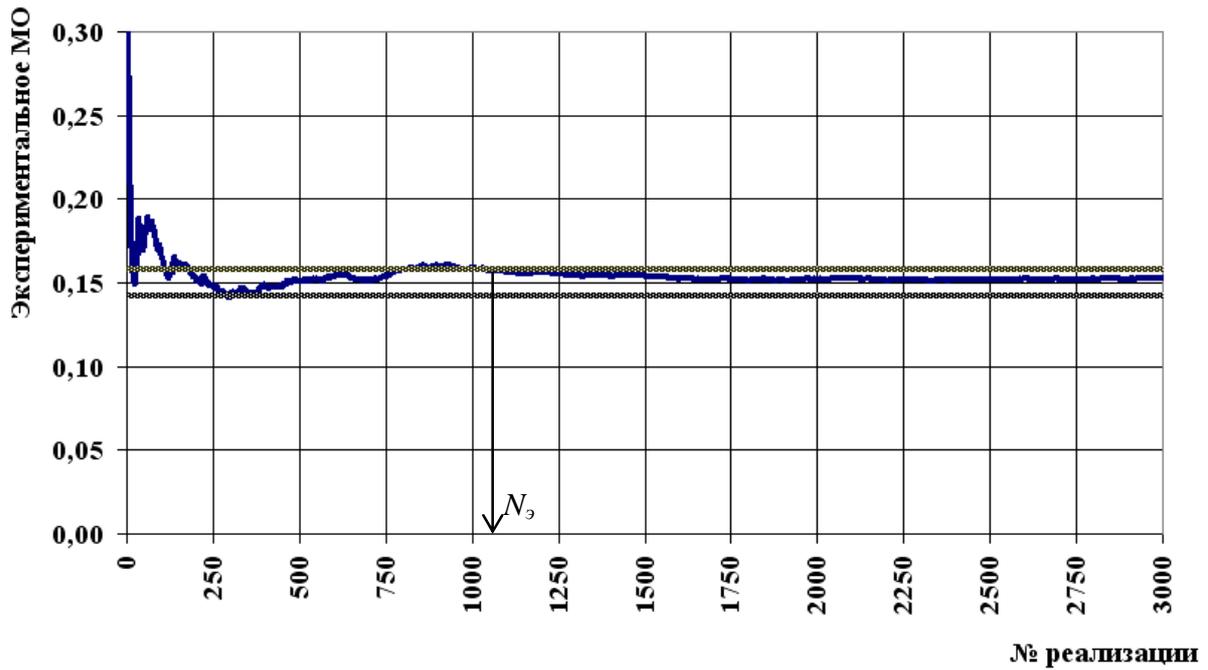


Рис. 61

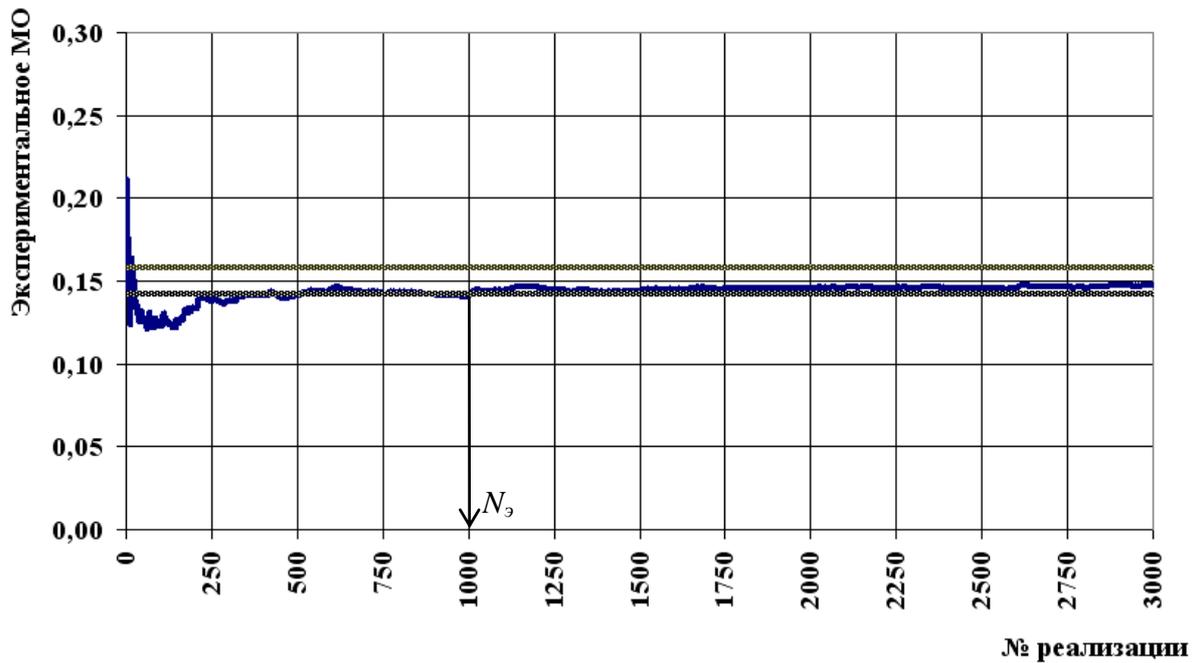


Рис. 62

Выполним приближенный детерминированный анализ заданной СМО. Так как в данном случае лимитирующим является процесс поступления и для него выбрано количество прогонов, равное 1500, в СМО по-

ступит 1500 заявок, а время моделирования составит  $t_m = 1500 \times MO_n = = 1500 \times 0,15 = 225$  секунд. За это время устройство обслуживания смогло бы обслужить  $t_m/MO_o = 225/0,1 = 2250$  заявок, т. е. все заявки, поступившие в СМО за время моделирования будут обслужены. Коэффициент использования устройства составит  $KI = MO_o/MO_n = 0,66$  [6]. Средняя длина очереди рассчитывается по формуле [27]:

$$L_{оч} = \frac{KI^2}{1 - KI} = \frac{0,66^2}{1 - 0,66} = 1,28 \text{ заявки.}$$

Среднее время ожидания обслуживания в очереди составит  $t_{ож} = L_{оч} \times MO_n = 1,28 \times 0,15 = 0,192$  секунды.

### *GPSS World.*

*Базовые понятия GPSS.* Модель GPSS состоит из блоков-операторов, которые задают логику функционирования системы и определяют пути движения заявок между элементами системы. В операторах могут происходить события следующих типов:

- создание или уничтожение заявок;
- изменение числовых атрибутов объекта (заявки, устройства, очереди, списка и т. д.);
- задержка заявки на определенный период времени;
- изменение маршрута движения заявки.

Заявки перемещаются в системном времени и пространстве, переходя от одного оператора модели к другому и воздействуя на них. Входя в оператор, заявка вызывает определяемую типом оператора подпрограмму, которая обрабатывает соответствующее событие. Далее заявка в общем случае пытается войти в следующий оператор. Продвижение продолжается до тех пор, пока не окажется, что очередной оператор должен выполнить одну из следующих функций:

- удалить заявку из модели;
- задержать заявку в предыдущем операторе до выполнения некоторых условий;
- задержать заявку на определяемое моделью время.

Тогда начинается продвижение другой заявки и т. д. – до завершения моделирования. За один шаг работы управляющей программы производит-

ся поочередный просмотр всех имеющихся процессов и имитация тех, которые могут быть запущены. При этом может освободиться ранее занятый ресурс или оказаться порожденным процесс более высокого приоритета, и просмотр списка процессов можно начать сначала. При описании этих процессов и в сборе итоговой статистики используются таймер модельного времени, стандартные числовые атрибуты (СЧА) и параметры заявок, а также определяемые пользователем переменные, выражения и таблично задаваемые функции. Наиболее сложные аспекты модели – планирование предстоящих событий и очередности их обработки – автоматически реализует планировщик событий (интерпретатор) *GPSS*.

Работа начинается с составления *GPSS*-программы. Компиляция организуется по команде `Command/Create Simulation`. Синтаксический контроль достаточно развит (в частности, контролируется парность скобок). При обнаружении ошибок создается их список. Каждый выбор элемента списка вызывает сообщение о типе ошибки в статусную строку и помещает точку вставки точно перед ошибочным элементом, например «Line 10, Col 20, Invalid Delimiter. Expecting a *GPSS* Verb».

*Операторы языка GPSS*. Оператор – это специальное обозначение для определенного действия над операндами (данными). Оператор модели *GPSS* может быть собственно оператором *GPSS*, либо PLUS-оператором. В отличие от PLUS-операторов, которые могут содержать несколько строк, операторы *GPSS* записываются одной строкой и состоят из полей, разделенных пробелами или ограничителями. В общем случае оператор состоит из следующих полей:

- метка (необязательно);
- глагол (обязательно);
- операнды (в зависимости от глагола);
- комментарий (обязательно).

Для вставки оператора *GPSS* в модельную реализацию можно использовать меню `Edit/Insert GPSS Blocks`.

Реализация потока заявок с заданным распределением интервалов времени между поступлениями и задержки заявки в обслуживающем устройстве на время обслуживания с заданным законом распределения осуществляется в *GPSS World* с использованием операторов `GENERATE` и `ADVANCE` соответственно. Способы задания типа и параметров распределения приведены в таблице 21.

Синтаксис операторов	Операнды
<i>Равномерное распределение (типа <math>a \pm b</math>)</i>	
GENERATE p1,p2 ADVANCE p1,p2	p1=a, p2= b
<i>Экспоненциальное распределение с интенсивностью <math>\lambda</math></i>	
GENERATE (Exponential(p1,p2,p3)) ADVANCE (Exponential(p1,p2,p3))	p1 – параметр равномерного распределения, обычно p1=1; p2 – смещение распределения, обычно p2=0; p3 - МО (величина обратная интенсивности $\lambda$ )
<i>Гамма-распределение с интенсивностью <math>\lambda</math> и целочисленным параметром <math>b</math></i>	
GENERATE (Gamma(p1,p2,p3,p4)) ADVANCE (Gamma(p1,p2,p3,p4))	p1 – параметр равномерного распределения, обычно p1=1; p2 – смещение распределения, обычно p2=0; p3 – величина равная $1/\lambda$ ; p4 – целочисленный параметр $b$ . МО для этого типа распределения равно $b/\lambda$
<i>Нормальное распределение с параметрами <math>m</math> и <math>\sigma</math></i>	
GENERATE (Normal(p1,p2,p3)) ADVANCE (Normal (p1,p2,p3))	p1 – параметр равномерного распределения, обычно p1=1; p2 – МО ( $m$ ); p3 – стандартное отклонение $\sigma$

Рассмотрим моделирование в *GPSS World* заданной одноканальной разомкнутой СМО. Моделирование потока требований выполняется с помощью оператора GENERATE:

GENERATE (Exponential(1,0,0.15))

Очередь задается с помощью операторов QUEUE и DEPART. Оператор QUEUE записывается в виде (в поле операнда дается символьное или числовое имя очереди):

QUEUE OCH1

Заявка будет находиться в очереди до тех пор, пока не поступит сообщение об освобождении устройства обслуживания. Для этого используется оператор SEIZE, который определяет занятость устройства обслуживания, и при освобождении последнего отправляет заявку на обслуживание (в поле операнда дается символьное или числовое имя устройства обслуживания):

SEIZE U1

Выход заявки из очереди фиксируется оператором DEPART (в поле операнда дается символьное или числовое имя очереди):

DEPART OCH1

Далее должно быть задано время пребывания заявки в устройстве обслуживания. Для этого используется оператор ADVANCE:

ADVANCE (Exponential(1,0,0.1))

После обслуживания заявка должна освободить устройство:

RELEASE U1

Парные операторы QUEUE и DEPART для каждой очереди должны иметь одно и то же уникальное имя или номер. Это же относится и к операторам SEIZE и RELEASE.

Выход заявки из системы моделируется следующим оператором (в поле операнда задается число удаляемых из системы заявок):

TERMINATE 1

В реализации модели можно жестко задать число прогонов с использованием оператора START (в поле операнда задается количество заявок, после удаления которых из системы процесс моделирования будет прекращен):

START 1500

Таким образом, имитационная модель заданной одноканальной разомкнутой СМО в среде *GPSS World* имеет вид, показанный на рис. 63.

Перед началом моделирования нужно установить вывод в стандартном отчете *GPSS World* тех параметров, которые необходимо получить в процессе моделирования, в меню Edit/Settings. В нашем случае выбору подлежат опции Facilities (устройства обслуживания) и Queues (очереди).

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого:

- щелкните по пункту Command главного меню системы или нажмите комбинацию клавиш Alt+C;
- щелкните по пункту Create Simulation (создать выполняемую модель) выпадающего меню.

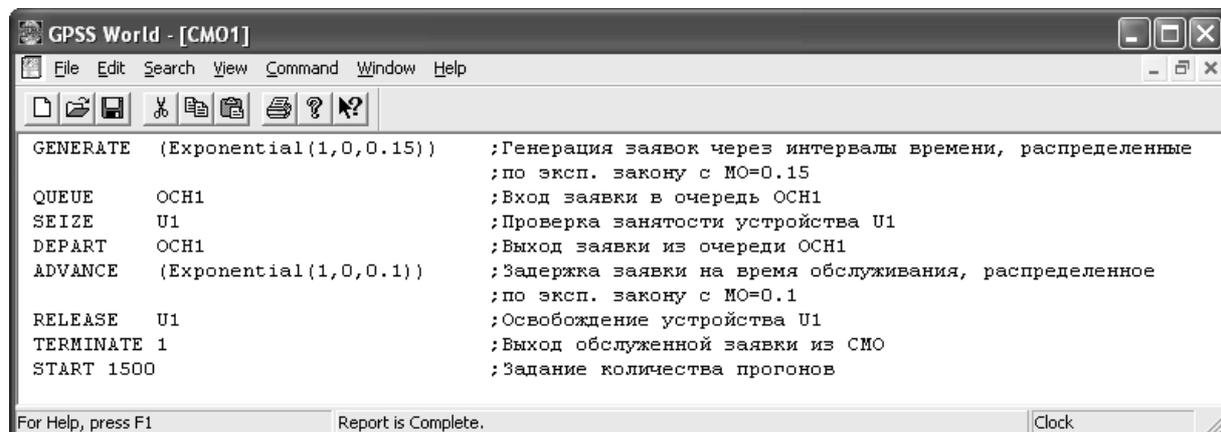


Рис. 63

Так как в модели имеется управляющая команда START, то исходная имитационная модель будет выполняться после транслирования, если в ней нет ошибок, а после завершения моделирования будет автоматически сформирован *стандартный отчет GPSS World*, показанный на рис. 64.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	225.539	7	1	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
U1	1501	0.684	0.103	1	1501	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
OCH1	13	1	1501	516	1.312	0.197	0

Рис. 64

Сравнение результатов имитационного моделирования СМО, выведенных в параметрах отчета, с результатами приближенного детерминированного анализа характеристик СМО, приведено в таблице 22 (расшифровку полей полного стандартного отчета GPSS – см. Приложение 1).

Удовлетворительное совпадение результатов приближенного анализа характеристик СМО с характеристиками, полученными по результатам статистического эксперимента в *GPSS World*, подтверждает корректность реализации имитационной модели.

Таблица 22

Параметр итогового отчета	Значение	Приближенная оценка
<i>Общие характеристики моделирования</i>		
START TIME (Начальное время)	0,000	–
END TIME (Время окончания)	225,539	225
BLOCKS (Число операторов)	7	–
FACILITIES (Число устройств обслуживания)	1	–
STORAGES (Число накопителей многоканальных устройств)	0	–
<i>Результаты моделирования для устройства обслуживания (FACILITY)</i>		
ENTRIES (Число входов)	1501	1500
UTIL. (Коэффициент использования)	0,684	0,66[6]
AVE. TIME (Среднее время обслуживания)	0,103	0,1

Параметр итогового отчета	Значение	Приближенная оценка
<i>Результаты моделирования очередей (QUEUE)</i>		
MAX (Максимальное длина очереди)	13	–
CONT. (Текущее содержание)	1	–
ENTRY (Число входов)	1501	1500
ENTRY(0) (Число входов без ожидания в очереди)	516	–
AVE.CONT. (Средняя длина очереди)	1,312	1,28
AVE.TIME (Среднее время ожидания заявки в очереди)	0,197	0,192
AVE. (-0) (Среднее время ожидания заявки в очереди, рассчитанное для заявок, обслуженных с ожиданием)	0,3	–

КИ очереди устройства СМО равен доле заявок, ожидавших обслуживания в очереди, от общего числа заявок, поступивших в очередь:

$$K_{Qj} = \frac{N_{Qj} - N_{0Qj}}{N_{Qj}},$$

где  $N_{Qj}$  – общее количество заявок, поступивших в очередь  $j$ ;  $N_{0Qj}$  – количество заявок с нулевым временем ожидания в очереди  $j$ . Для заданной СМО:

$$K_{Qочн1} = \frac{1501 - 516}{1501} = 0,656.$$

Для построения гистограммы времени ожидания заявок в очереди устройства обслуживания в программную реализацию модели следует добавить команду QTABLE формата:

метка QTABLE p1,p2,p3,p4

Метка при построении выводится как заголовок гистограммы; значения параметров: p1 – имя очереди; p2, p3 и p4 – начало координат (левая граница первого интервала), шаг (ширина интервала) и максимальное значение по оси  $x$  (количество интервалов).

Гистограмма времени ожидания заявок в очереди устройства, полученная путем табулирования данных с использованием статистической Q-таблицы GIS имеет следующий вид, представленный на рис. 65.

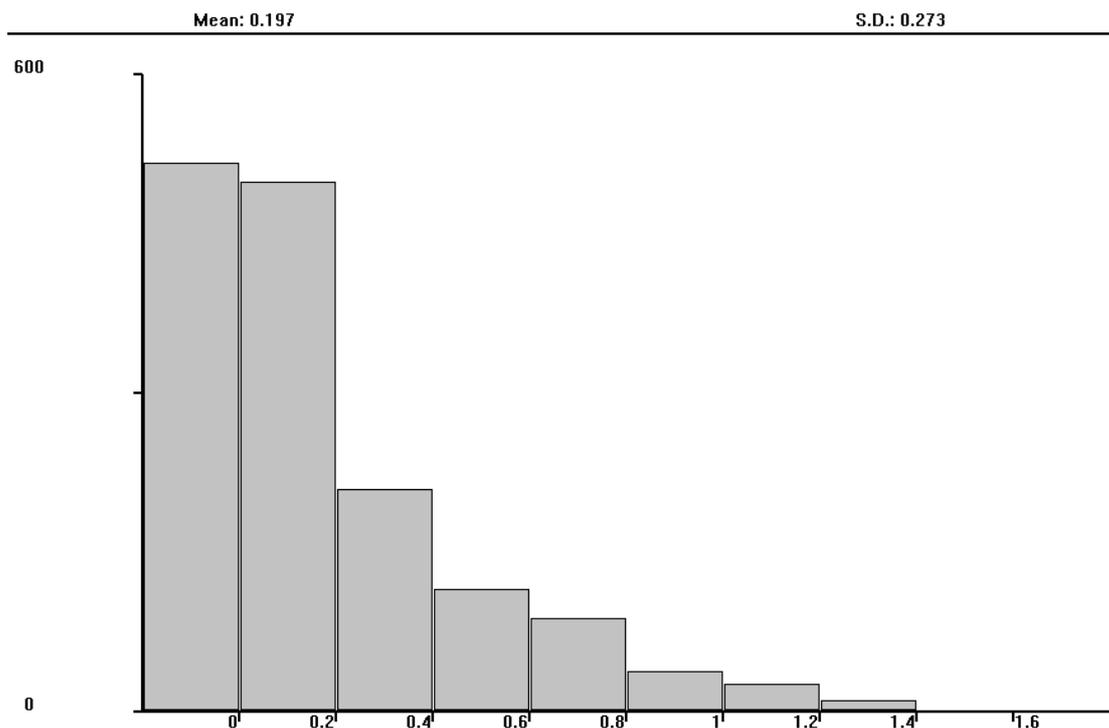


Рис. 65

Для вывода табличной формы в стандартном отчете необходимо в меню Edit/Settings выбрать дополнительно опцию Tables (таблицы). Фрагмент стандартного отчета *GPSS World* – раздел информации о статистических таблицах – для построенной гистограммы имеет вид, показанный на рис. 66.

TABLE	MEAN	STD. DEV.	RANGE	RETRY	FREQUENCY	CUM. %
GIS	0.197	0.273		0		
			0.000 - 0.200		516	34.40
			0.200 - 0.400		498	67.60
			0.400 - 0.600		209	81.53
			0.600 - 0.800		115	89.20
			0.800 - 1.000		88	95.07
			1.000 - 1.200		38	97.60
			1.200 - 1.400		26	99.33
					10	100.00

For Help, press F1      Report is Complete.      Clock

Рис. 66

## Варианты индивидуальных заданий

Варианты индивидуальных заданий представлены в таблице 23.

Таблица 23

№	Входной поток заявок		Обслуживание в устройстве		Задача оптимизации (модификации)
	закон распределения	параметры	закон распределения	параметры	
1	эсп.	$\lambda=0,01$	равн.	$10\pm 2$	Определить максимально возможную интенсивность поступления, при которой коэффициент использования устройства составит 65%.
2	равн.	$10\pm 2$	эсп.	$\lambda=0,33$	Определить максимально возможное значение параметра поступления $A$ для интервала поступления $A\pm 0,2\cdot A$ , при котором коэффициент использования устройства не превышает 65%.
3	эсп.	$\lambda=6,2$	эсп.	$\lambda=7,1$	Определить максимально возможную интенсивность входного потока, обеспечивающую безочередное обслуживание для 28% заявок.
4	эсп.	$\lambda=0,35$	эсп.	$\lambda=80$	Определить предельную интенсивность, при которой становится невозможным безочередное обслуживание.
5	эсп.	$\lambda=0,035$	гамма	$\lambda=0,033$ $b=3$	Определить максимальную интенсивность входного потока, при которой очередь можно ограничить 4 заявками.
6	эсп.	$\lambda=0,15$	равн.	$2\pm 0,2$	Определить максимально возможную интенсивность входного потока, обеспечивающую безочередное обслуживание для 20% заявок.
7	эсп.	$\lambda=0,72$	эсп.	$\lambda=5$	Определить интенсивность входного потока, при которой коэффициент использования очереди будет меньше начального значения 10%.
8	эсп.	$\lambda=0,27$	гамма	$\lambda=0,93$ $b=3$	Определить максимально возможную интенсивность входного потока, обеспечивающую безочередное обслуживание.
9	гамма	$\lambda=0,014$ $b=2$	эсп.	$\lambda=0,27$	Определить минимально возможную интенсивность обслуживания, при которой возможны потери при условии ограничения очереди 15-ю заявками.
10	эсп.	$\lambda=0,05$	эсп.	$\lambda=0,045$	Определить коэффициент использования очереди, являющийся необходимым условием для обеспечения коэффициента использования устройства равного 80%.
11	эсп.	$\lambda=0,2$	норм.	$m=4,8$ $\sigma=0,5$	Определить минимально возможное значение интервала времени между поступлениями заявок для обеспечения коэффициента использования очереди равного 12%.
12	норм.	$m=2,2$ , $\sigma=0,1$	эсп.	$\lambda=1,1$	Определить интенсивность обслуживания, при которой коэффициент использования очереди будет в два раза меньше начального значения.
13	гамма	$\lambda=0,5$ ; $b=2$	гамма	$\lambda=0,5$ $b=4$	Определить максимально возможную интенсивность поступления, при которой процент заявок, ожидающих обслуживания в очереди, составит не более 25%

14	гамма	$\lambda=0,5;$ $b=2$	норм.	$m=2,2$ $\sigma=0,1$	Определить максимально возможную интенсивность поступления, при которой коэффициент использования устройства составит 45%.
15	норм.	$m=0,5$ $\sigma=0,05$	гамма	$\lambda=4,5$ $b=3$	Определить минимально возможное значение интервала времени между поступлениями заявок для обеспечения коэффициента использования устройства равного 30%.
16	норм.	$m=4$ $\sigma=0,21$	равн.	$3,5\pm 0,5$	Определить значение параметра обслуживания $A$ для распределения времени обслуживания $A\pm 0,2A$ , при котором суммарное время использования очереди в процессе обслуживания будет составлять не более 25% от общего времени эксплуатации системы.
17	равн.	$3,5\pm 0,5$	гамма	$\lambda=0,5$ $b=2$	Определить, при какой интенсивности обслуживания коэффициент использования очереди составит 50%.
18	гамма	$\lambda=2,5$ $b=2$	равн.	$0,5\pm 0,2$	Определить, при какой интенсивности поступления очередь обслуживания можно ограничить 4-мя заявками.
19	норм.	$m=3$ $\sigma=0,09$	норм.	$m=2$ $\sigma=0,04$	Определить при каком минимальном среднем интервале между поступлениями заявок обслуживание будет безочередным.
20	гамма	$\lambda=0,2$ $b=2$	гамма	$\lambda=0,25$ $b=3$	Определить, при каком коэффициенте использования устройства очередь можно ограничить 3-мя заявками.
21	гамма	$\lambda=0,25;$ $b=2$	эксп.	$\lambda=0,25$	Определить, при каком коэффициенте использования очереди коэффициент использования устройства составит 25%.
22	равн.	$3\pm 1$	норм.	$m=3$ $\sigma=1$	Определить, при каком параметре $\sigma$ распределения времени обслуживания коэффициент использования очереди не будет превышать 10%.
23	эксп.	$\lambda=0,8$	гамма	$\lambda=0,62$ $b=2$	Определить, при какой интенсивности обслуживания коэффициент использования очереди будет в два раза меньше начального значения.
24	эксп.	$\lambda=0,72$	эксп.	$\lambda=5$	Определить максимально возможную интенсивность поступления, при которой коэффициент использования устройства составит 15%.
25	эксп.	$\lambda=0,5$	норм.	$m=20,8$ $\sigma=2$	Определить, при какой интенсивности входного потока средняя длина очереди в процессе обслуживания не будет превышать 25 заявок.
26	эксп.	$\lambda=0,25$	гамма	$\lambda=0,5$ $b=2$	Определить, при какой интенсивности входного потока среднее время ожидания обслуживания составит 1 секунду.
27	эксп.	$\lambda=0,1$	эксп.	$\lambda=0,25$	Определить, при какой интенсивности входного потока среднее время пребывания заявки в системе составит 15 секунд.
28	гамма	$\lambda=0,5$ $b=3$	равн.	$4\pm 0,5$	Определить значение параметра обслуживания $A$ для распределения времени обслуживания $A\pm 0,1A$ , при котором только 50% заявок от числа поступивших в систему будут обслужены с ожиданием.

## **Лабораторная работа №4. Статистическое моделирование многоканальных СМО безприоритетного обслуживания неоднородного потока заявок с ограничением очередей обслуживания**

*Цель:* Моделирование многоканальной многофазной СМО безприоритетного обслуживания с ограничением очередей обслуживания. Модификация СМО заданным способом по условию гарантии безочередного обслуживания для заданного процента заявок входного потока.

*Программное обеспечение моделирования:* GPSS World.

### **Теория**

СМО бывают одно- и многоканальными в зависимости от числа устройств, параллельно обрабатывающих входной поток заявок; одно- и многофазными в зависимости от числа последовательно включенных устройств. Параметры поступления и обслуживания заявок в СМО, как правило, являются случайными величинами, распределенными по заданному закону.

Основные сведения об имитационных моделях СМО – см. раздел 3.4. конспекта лекций.

### **Ход выполнения работы**

**1.** Реализовать имитационную модель заданного варианта СМО в *GPSS World*. В программной реализации обязательны построчный комментарий.

**2.** Осуществить имитационное моделирование СМО. По результатам моделирования оценить следующие характеристики СМО:

- коэффициент использования устройств обслуживания;
- среднее время обслуживания;
- число заявок, поступивших в устройства обслуживания;
- среднее содержимое очередей (накопителей);
- среднее время пребывания заявки в очередях (накопителях);
- максимальное содержимое очередей (накопителей);
- коэффициент использования очередей (накопителей);
- процент потерь заявок (если есть).

**3.** Проанализировать результаты моделирования. Оценить эффективность СМО с точки зрения загрузки устройств, задержек заявок в очередях и накопителях, потерь заявок из-за переполнения очередей (если есть).

**4.** Пункт выполняется в вариантах с четным номером! Предложить и реализовать вариант структурной оптимизации СМО, гарантирующей безочередное обслуживание без потерь для 95-98% заявок. По результатам моделирования оценить характеристики СМО.

**5.** *Пункт выполняется в вариантах с нечетным номером!* Предложить и реализовать вариант параметрической модификации СМО, обеспечивающей безочередное обслуживание для 50% заявок в лимитирующей очереди. По результатам моделирования оценить характеристики СМО.

**6.** Результаты выполнения работы по пунктам 1-4(5) и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

### **Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования**

*Моделирование многофазных систем.* В отсутствие условных и безусловных переходов операторы программной реализации имитационной модели GPSS World выполняются последовательно, что позволяет реализовать последовательное обслуживание входного потока заданным количеством устройств (см. *Пример 1*).

*Моделирование ограниченных очередей обслуживания.* Для моделирования объектов, обладающих определенной емкостью – очередей или памяти, используется оператор STORAGE, задающий накопитель:

имя STORAGE A

где A – емкость (количество единиц) накопителя. Для помещения заявки в накопитель используется оператор ENTER следующего формата:

ENTER имя,В

где В – число занимаемых единиц накопителя (необязательный операнд).

Для освобождения заявкой накопителя используется оператор LEAVE следующего формата:

LEAVE имя,В

где В – число освобождаемых единиц накопителя (необязательный операнд).

*Условные или вероятностные переходы в процессе обслуживания.* Для задания в программной реализации модели СМО условных или вероятностных переходов заявок используется оператор TRANSFER:

TRANSFER A,B,C,D

где A – режим (BOTH, ALL, PICK, FN, P, SBR, SIM); B, C – номера или метки операторов; D – приращение номера оператора (для режима ALL).

Возможные форматы использования:

– TRANSFER ,MET – безусловная передача заявки оператору с меткой (номером) MET (см. *Пример 3*);

– TRANSFER BOTH,LAB1,UNN – переход к оператору с меткой LAB1, если он невозможен, то к оператору с меткой UNN, если и он невозможен, то заявка задерживается до следующего момента модельного времени, в который повторяются указанные попытки перехода;

– TRANSFER .4,AAA,LAB – заявка с вероятностью 0.4 переходит к оператору с меткой LAB и с вероятностью 0.6 к оператору с меткой AAA.

– TRANSFER .4,,LAB – заявка с вероятностью 0.4 переходит к оператору с меткой LAB и с вероятностью 0.6 к оператору, следующему сразу за оператором TRANSFER;

– TRANSFER ALL,STK7,STK21,N переход к оператору с меткой STK7, если он невозможен, то заявка направляется к следующему оператору, пропуская N операторов, и так далее, до оператора с меткой STK21. Если ни один из переходов невозможен, то заявка задерживается до следующего момента дискретного модельного времени, в который повторяются указанные попытки переходов (см. *Пример 2*);

– TRANSFER PICK,STK7,STK21 – равновероятный случайный переход к операторам с метками STK7, STK7+1, STK7+2, . . . , STK21. Заявка пытается перейти *только* к выбранному для нее оператору. Если переход сразу невозможен, то она задерживается в операторе TRANSFER до тех пор, пока не будет снято блокирующее условие;

– TRANSFER P,4,41 – переход к оператору, метка которого равна сумме значения параметра №4 заявки и числа 41;

– TRANSFER SBR,PRC,7 – переход к оператору PRC с записью в параметр №7 заявки метки данного оператора. Если такого параметра нет, он будет создан.

Для изменения маршрута движения заявки в зависимости от состояния некоторого объекта может быть использован оператор GATE:

GATE O A,B

где O – условие в виде стандартного логического атрибута; A – номер или имя проверяемого объекта (устройства, накопителя, логического ключа); B – номер или метка перемещения заявки в случае невыполнения условия.

В качестве условия чаще всего используются стандартные логические атрибуты, представленные в таблице 24.

Оператор GATE работает в двух режимах: отказа во входе или разрешения во входе и альтернативном выходе.

Таблица 24

Оператор	Значение
FV	Устройство доступно.
FNV	Устройство недоступно.
SE	Накопитель пуст.
SF	Накопитель заполнен.
SNE	Накопитель не пуст.
SNF	Накопитель не заполнен.
NU	Устройство свободно от обслуживания.
U	Устройство занято обслуживанием.

Варианты использования:

- GATE SNF NAK,RESERVE – если накопитель с именем NAK не заполнен, то выполняется оператор, следующий за GATE; в противном случае заявка переходит к оператору с меткой RESERVE (см. *Пример 4*);
- GATE U Zrk – оператор не пропустит заявку, если устройство с именем Zrk не занято;
- GATE NU Vat,Oper – если устройство с именем Vat занято, то заявка будет направлена к оператору с меткой Oper.

*Моделирование многоканальных систем.* Для организации параллельного обслуживания в имитационной модели задаются условия передачи заявок на обслуживание в устройства многоканальной системы с использованием операторов TRASFER или GATE (см. *Пример 2*)

*Задание функций дискретного распределения вероятностей.* Описание функции, позволяющей получить дискретно распределенные случайные числа по методу Монте-Карло имеет вид:

имя FUNCTION RN1,Dn

$$P_{1,x_1}/P_{2,x_2}/\dots/P_{n,x_n}$$

Здесь параметр RN1 указывает на то, что для реализации генератора псевдослучайных чисел используется равномерно распределенная в интервале от 0 до 1 случайная величина; константа n параметра D задает число интервалов дискретного распределения;  $P_1 \dots P_n$  – суммарные вероятности того, что значение функции будет равно  $x_1 \dots x_n$  соответственно. Ссылка на значение функции задается в виде:

FN\$имя

Пусть на вход СМО поступает поток заявок 3-х типов. Вероятности поступления заявок каждого типа равны, соответственно: 0,33; 0,33 и 0,34.

Для определения типа заявки следует, таким образом, использовать дискретную функцию распределения, приведенную на рис. 67.

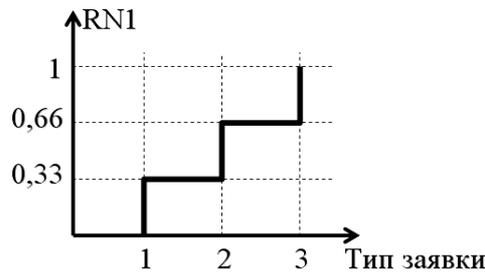


Рис. 67

В программной реализации имитационной модели данная функция может быть реализована следующим образом (см. *Пример 3*):

```
type FUNCTION RN1,D3
0.33,T1/0.66,T2/1,T3
```

***Пример 1. Двухфазная одноканальная разомкнутая СМО с экспоненциальным распределением времени поступления и обслуживания***

Пусть дана двухфазная одноканальная разомкнутая СМО с обслуживающими устройствами S1, S2 и бесконечными очередями. Окно имитационной модели в среде *GPSS World* для данного примера имеет вид, показанный на рис. 68.

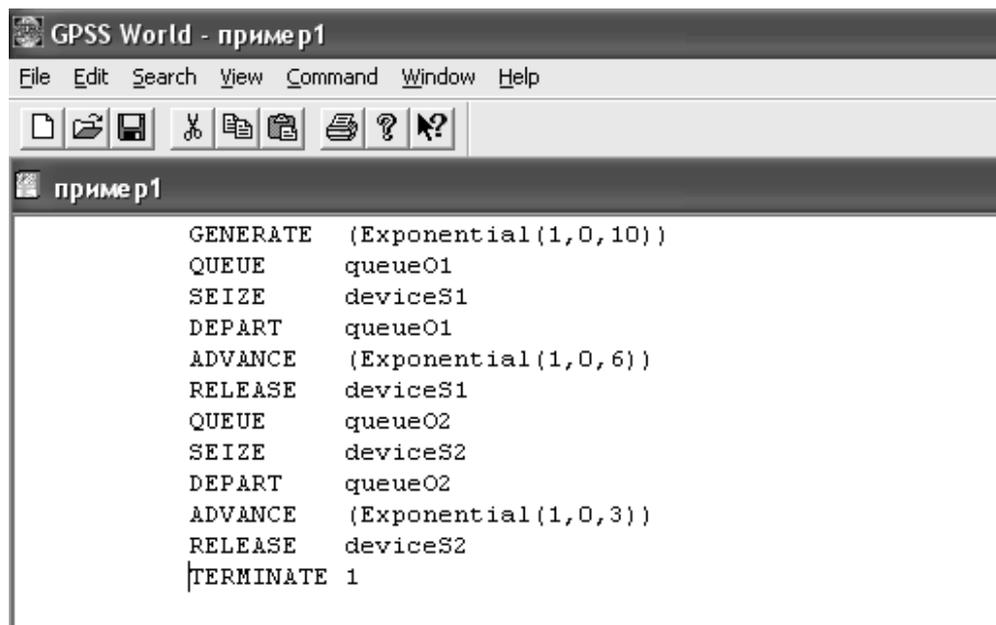


Рис. 68

В модели отсутствует команда START, поэтому для запуска моделирования после транслирования необходимо выбрать Command/Start и в появившемся окне указать количество прогонов модели. Окно Report с результатами моделирования СМО имеет вид, представленный на рис. 69.

Параметры итогового отчета:

- START TIME (Начальное время) – 0.000;
- END TIME (Время окончания) – 10446.825;
- BLOCKS (Число операторов-блоков) – 12;
- FACILITIES (Число устройств обслуживания) – 2;
- STORAGES (Число накопителей) – 0.

Результаты моделирования для устройств обслуживания (FACILITY) deviceS1 и deviceS2:

- ENTRIES (Число входов) – 1002 и 1001 соответственно;
- UTIL. (Коэффициент использования) – 0,563 и 0,294;
- AVE. TIME (Среднее время обслуживания в ОУ) – 5,867 и 3,073.

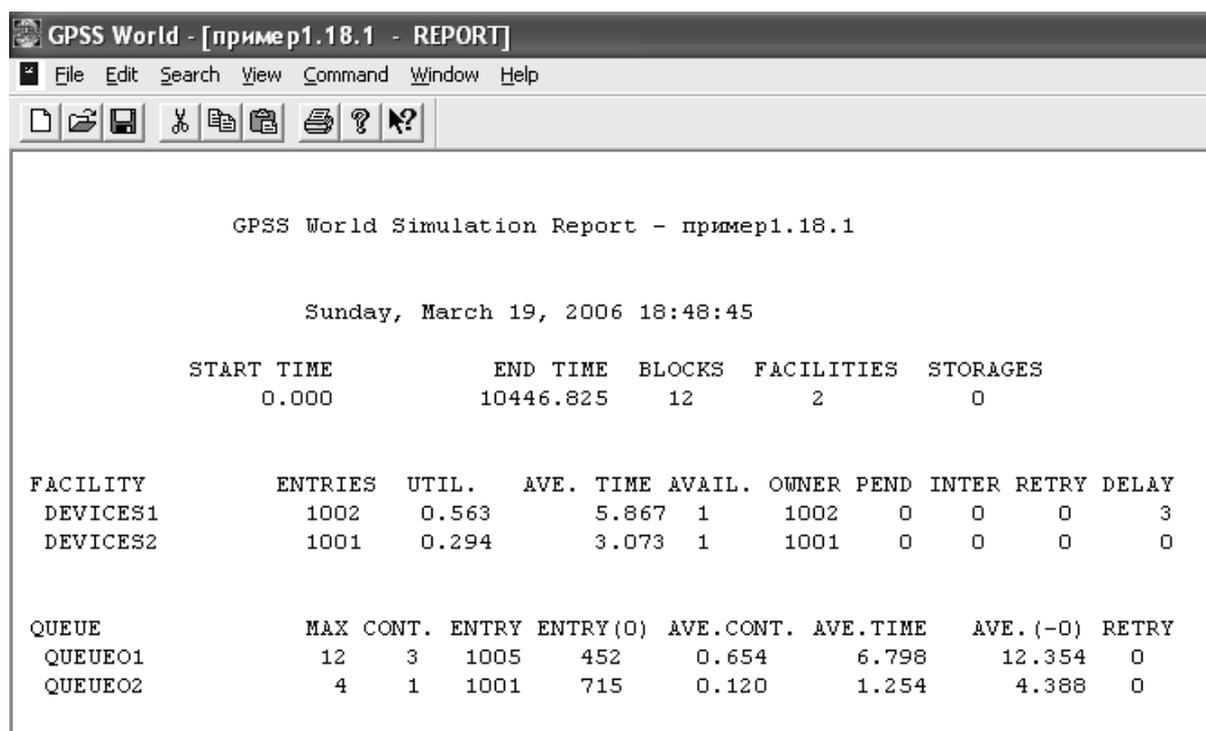


Рис. 69

Результаты функционирования очередей queueO1 и queueO2:

- MAX (Максимальная длина очереди) – 12 и 4 соответственно;
- CONT. (Текущее содержание) – 3 и 1;
- ENTRY (Число входов) – 1005 и 1001;
- ENTRY(0) (Число входов без ожидания в очереди) – 452 и 715;
- AVE.CONT. (Средняя длина очереди) – 0,654 и 0,12;

- AVE.TIME (Среднее время ожидания) – 6,798 и 1.254;
- AVE. (-0) (Среднее время ожидания, рассчитанное для заявок, обслуженных с ожиданием) – 12,354 и 4,388 соответственно.

**Пример 2. Трехканальная СМО с общей неограниченной очередью**

Пусть дана трехканальная СМО с устройствами обслуживания S1-S3 и общей неограниченной очередью (см. рис. 70). Заявки входного потока поступают на обслуживание в свободное устройство с наименьшим номером. Время поступления и обслуживания распределено по экспоненциальному закону с различными интенсивностями.

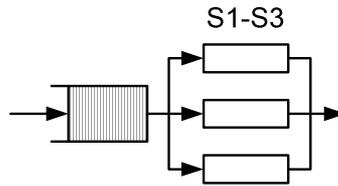


Рис. 70

Окно имитационной модели в среде *GPSS World* имеет вид, представленный на рис. 71.

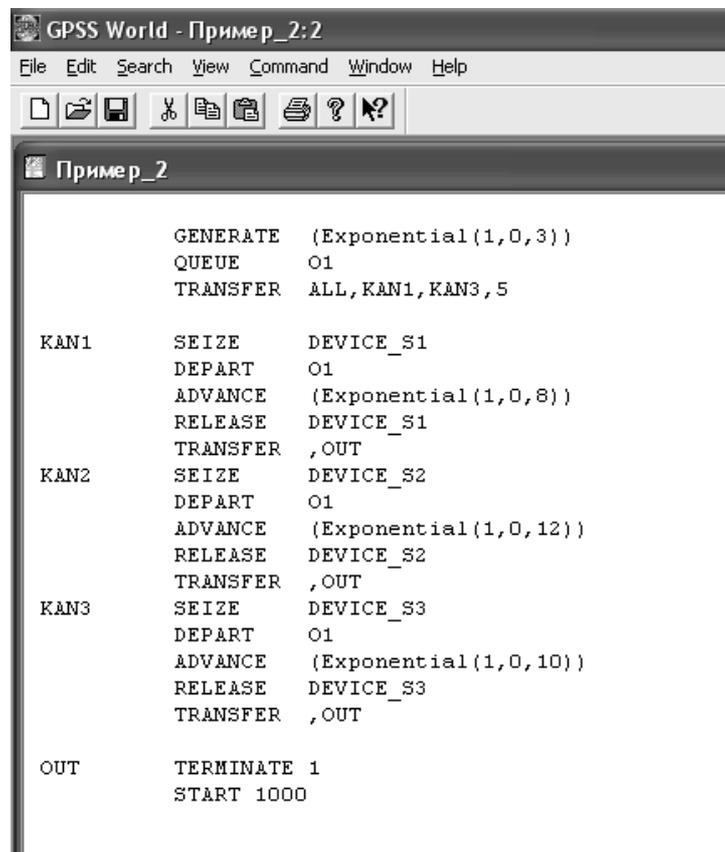


Рис. 71

Окно Report с результатами моделирования СМО имеет вид, показанный на рис. 72 (обратите внимание: в стандартный отчет добавлен в меню Edit/Settings раздел информации об операторах модели Blocks). Как видно из отчета, из 1000 заявок, покинувших систему (оператор №19 TERMINATE), 423 были обслужены в первом устройстве (количество входов заявок в оператор №7 RELEASE), 268 – во втором (количество входов в оператор №12 RELEASE), и 309 – в третьем (количество входов в оператор №17 RELEASE). На момент завершения моделирования 79 заявок из числа поступивших 1081 находились в общей входной очереди O1 (см. количество заявок, задержанных в операторе №3 TRANSFER), одна заявка не завершила обслуживание в S2 и одна – в S3 (см. количество заявок, задержанных в операторах ADVANCE №11 и №16 соответственно).

	START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
	0.000	3274.339	19	3	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1081	0	0
	2	QUEUE	1081	0	0
	3	TRANSFER	1081	79	0
KAN1	4	SEIZE	423	0	0
	5	DEPART	423	0	0
	6	ADVANCE	423	0	0
	7	RELEASE	423	0	0
	8	TRANSFER	423	0	0
KAN2	9	SEIZE	269	0	0
	10	DEPART	269	0	0
	11	ADVANCE	269	1	0
	12	RELEASE	268	0	0
	13	TRANSFER	268	0	0
KAN3	14	SEIZE	310	0	0
	15	DEPART	310	0	0
	16	ADVANCE	310	1	0
	17	RELEASE	309	0	0
	18	TRANSFER	309	0	0
OUT	19	TERMINATE	1000	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
DEVICE_S1	423	0.999	7.731	1	0	0	0	79	0
DEVICE_S2	269	0.999	12.159	1	1002	0	0	79	0
DEVICE_S3	310	0.996	10.521	1	1000	0	0	79	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
O1	85	79	1081	5	45.504	137.831	138.472

Рис. 72

### Пример 3. Трехканальная СМО с независимыми очередями для обслуживания потока заявок трех типов

Дана трехканальная разомкнутая СМО с обслуживающими устройствами S1, S2, S3 и бесконечными очередями (см. рис. 73).

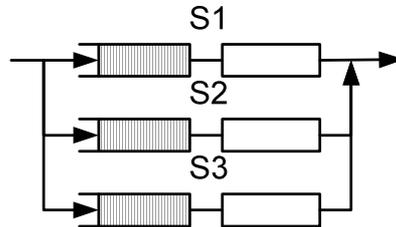


Рис. 73

На вход СМО поступает поток заявок 3-х типов. Вероятности поступления заявок каждого типа равны, соответственно: 0,33; 0,33 и 0,34. Время поступления заявок и время обслуживания являются постоянными (3 и 1 секунда соответственно); заявки каждого типа обслуживаются в устройстве соответствующего номера с независимыми очередями.

Окно имитационной модели в среде *GPSS World* для данного примера имеет вид, показанный на рис. 74.

```
GPSS World - diskr_potok
File Edit Search View Command Window Help
[Icons]
diskr_potok
type FUNCTION RN1,D3 ; дискретная функция распределения заявок 3-х типов
0.33,T1/0.66,T2/1,T3
GENERATE 3,0
TRANSFER ,FN$type ; переход по типу
T1 QUEUE O1
SEIZE DEVICE_S1
DEPART O1
ADVANCE 1,0
RELEASE DEVICE_S1
TRANSFER ,OUT
T2 QUEUE O2
SEIZE DEVICE_S2
DEPART O2
ADVANCE 1,0
RELEASE DEVICE_S2
TRANSFER ,OUT
T3 QUEUE O3
SEIZE DEVICE_S3
DEPART O3
ADVANCE 1,0
RELEASE DEVICE_S3
OUT TERMINATE 1
START 100 ; генерация 100 заявок
For Help, press F1 Report is Complete.
```

Рис. 74

Окно Report с результатами моделирования СМО имеет вид, представленный на рис. 75. Как видно из результатов моделирования из 100 заявок входного потока:

- 33 оказались заявками первого типа и были обслужены в устройстве S1 (количество входов в оператор №7 RELEASE);
- 29 оказались заявками второго типа и были обслужены в устройстве S2 (количество входов в оператор №13 RELEASE);
- 38 оказались заявками третьего типа и были обслужены в устройстве S3 (количество входов в оператор №19 RELEASE).

GPSS World - [diskr\_potok.15.1 - REPORT]

File Edit Search View Command Window Help

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	301.000	20	3	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	100	0	0
	2	TRANSFER	100	0	0
T1	3	QUEUE	33	0	0
	4	SEIZE	33	0	0
	5	DEPART	33	0	0
	6	ADVANCE	33	0	0
	7	RELEASE	33	0	0
	8	TRANSFER	33	0	0
T2	9	QUEUE	29	0	0
	10	SEIZE	29	0	0
	11	DEPART	29	0	0
	12	ADVANCE	29	0	0
	13	RELEASE	29	0	0
	14	TRANSFER	29	0	0
T3	15	QUEUE	38	0	0
	16	SEIZE	38	0	0
	17	DEPART	38	0	0
	18	ADVANCE	38	0	0
	19	RELEASE	38	0	0
OUT	20	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
DEVICE_S2	29	0.096	1.000	1	0	0	0	0	0
DEVICE_S1	33	0.110	1.000	1	0	0	0	0	0
DEVICE_S3	38	0.126	1.000	1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
O2	1	0	29	29	0.000	0.000	0.000	0
O1	1	0	33	33	0.000	0.000	0.000	0
O3	1	0	38	38	0.000	0.000	0.000	0

For Help, press F1      Report is Complete.

Рис. 75

Таким образом, вероятности поступления заявок каждого типа, полученные по результатам имитационного статистического моделирования, приблизительно соответствуют заданным. Для увеличения точности моделирования необходимо увеличить число прогонов. Из-за того, что интенсивность обслуживания в данном примере значительно превышает интенсивность поступления, все заявки были обслужены без ожидания в очереди, а коэффициенты использования устройств составили: для S1 – 0,11; S2 – 0,096; S3 – 0,126.

#### **Пример 4. Двухканальная СМО с ограниченной очередью**

Дана двухканальная разомкнутая СМО, очередь основного обслуживания которой ограничена 3-мя заявками. Если заявка входного потока поступает в систему в момент времени, когда в очереди основного обслуживания (устройство S1) уже находятся 3 заявки, то она отправляется на обслуживание в резервное устройство S2 с неограниченной очередью.

Окно имитационной модели в среде *GPSS World* для данного примера имеет вид, показанный на рис. 76.

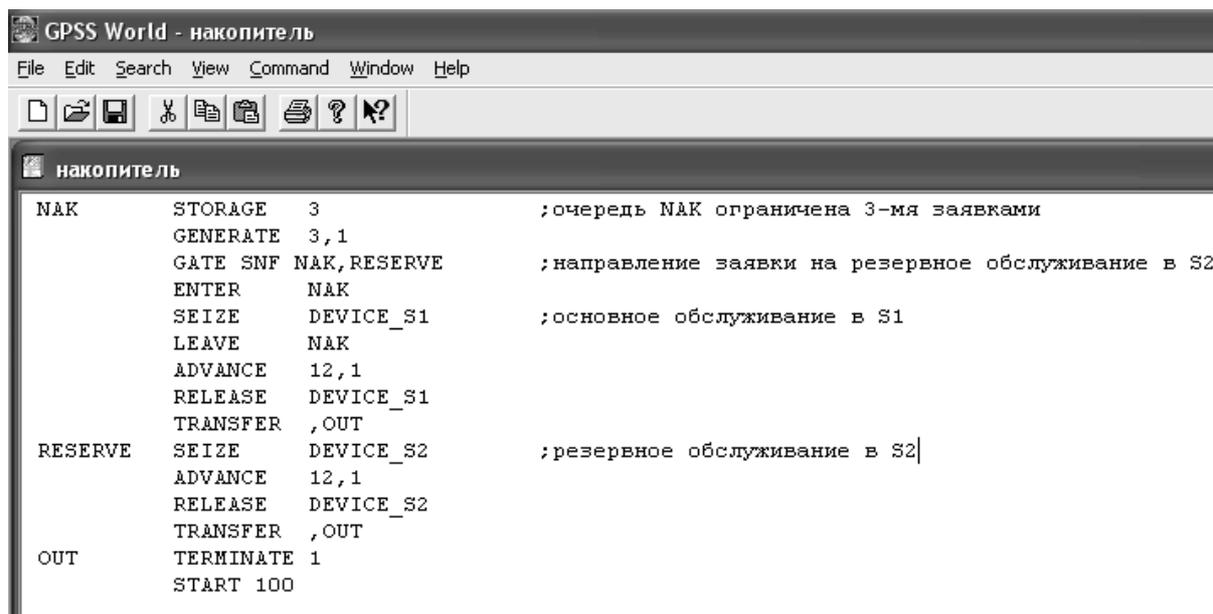


Рис. 76

Окно Report с результатами моделирования СМО имеет вид, представленный на рис. 77.

Результаты функционирования ограниченной очереди – накопителя NAK, полученные из стандартного отчета, имеют вид:

- CAP (Вместимость) – 3;
- REM (Количество удаленных заявок) – 0;

- MIN (Минимальное число заявок в накопителе) – 0;
- MAX (Максимальное число заявок в накопителе) – 3;
- ENTRIES (Число входов) – 54;
- AVL (Доступность) – 1;
- AVE.C (Среднее количество заявок за время моделирования) – 2.846;
- RETRY (Повтор) – 0;
- DELAY (Отказ) – 0.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	614.842	13	2	1

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	206	0	0
	2	GATE	206	102	0
	3	ENTER	54	2	0
	4	SEIZE	52	1	0
	5	LEAVE	51	0	0
	6	ADVANCE	51	0	0
	7	RELEASE	51	0	0
	8	TRANSFER	51	0	0
RESERVE	9	SEIZE	50	0	0
	10	ADVANCE	50	1	0
	11	RELEASE	49	0	0
	12	TRANSFER	49	0	0
OUT	13	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
DEVICE_S1	52	0.994	11.753	1	194	0	0	2
DEVICE_S2	50	0.976	12.000	1	71	0	0	102

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NAK	3	0	0	3	54	1	2.846	0.949	0	0

Рис. 77

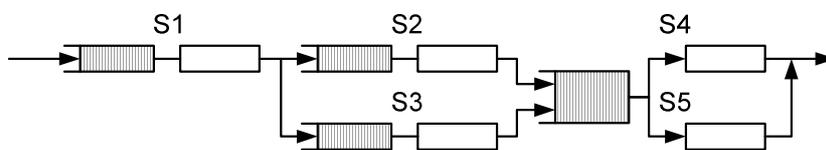
Результаты моделирования, представленные в отчете (см. рис. 77), показывают, что на момент завершения моделирования (когда 100-я заявка покинула систему после обслуживания):

– 54 заявки поступили на основное обслуживание в S1 (количество входов в оператор №3 ENTER), из них 3 не успели пройти обслуживание на момент завершения моделирования, причем 2 не приступили к обслуживанию и остались в очереди (количество заявок, задержанных на момент завершения моделирования в операторе №3 ENTER).

- 50 заявок поступили на резервное обслуживание в S2 из-за переполнения очереди основного обслуживания (количество входов в оператор №9 SEIZE), из них 1 не успела пройти обслуживание на момент завершения моделирования (задержана в операторе №10 ADVANCE). Из-за низкой интенсивности обслуживания (в сравнении с интенсивностью поступления) в неограниченной очереди резервного обслуживания на момент завершения моделирования находилось 102 заявки (в данном случае моделируется оператором №2 GATE).

### Варианты индивидуальных заданий

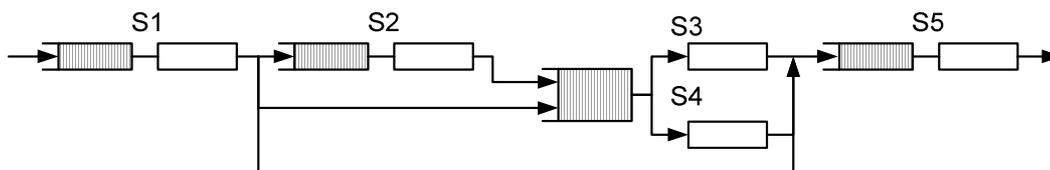
1.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. После обслуживания в S1, заявки первого типа обслуживаются в S2, а второго – в S3. Общая очередь S4-S5 ограничена 3-мя заявками, при переполнении заявки теряются. Вероятности обслуживания в S4 – 0,3; в S5 – 0,7.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	равн.
параметры	$\lambda=2$	$1\pm 0,5$	$2\pm 0,5$	$2\pm 1,5$	$\lambda=0,13$	$6\pm 1$

2.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,3 и 0,7 соответственно. Заявки первого типа обслуживаются в S1 и S2; второго – после обслуживания в S1 с вероятностью 0,7 поступают в общую очередь устройств S3-S4, а с вероятностью 0,3 – сразу в очередь устройства S5.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	эсп.	равн.	равн.	равн.
параметры	$0.5\pm 0.2$	$\lambda=0,2$	$\lambda=0,25$	$3\pm 1$	$2\pm 0,5$	$6\pm 2$

3.

Схема – см. вариант 2. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,6 и 0,4 соответственно. Очередь S2 ограничена 3-мя заявками. При переполнении очереди заявки второго типа теряются, первого с вероятностью 0,2 в общую очередь устройств S3-S4, а с вероятностью 0,8 – сразу в очередь устройства S5.

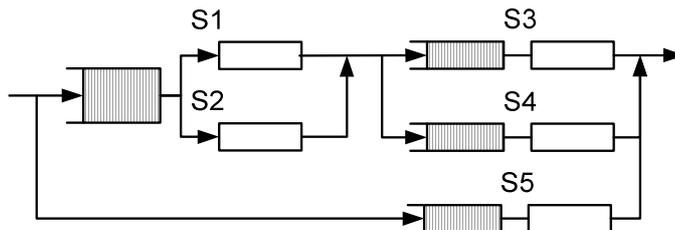
объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	гамма	равн.	равн.
параметры	$1 \pm 0,5$	$1.2 \pm 0,5$	$\lambda=0,2$	$\lambda=0,5; b=3$	$2 \pm 0,5$	$2 \pm 0,5$

4.

Схема – см. вариант 1. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,55 и 0,45 соответственно. Очередь S2 ограничена 2-мя заявками; заявки покинувшие очередь из-за ограничения, поступают на обслуживание в S3. Общая очередь S4-S5 ограничена 3-мя заявками, при переполнении заявки теряются. Заявки первого типа обслуживаются в S4, второго – в S5.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	равн.
параметры	$\lambda=3$	$10 \pm 2$	$9 \pm 1$	$12 \pm 4$	$\lambda=0,07$	$8 \pm 1$

5.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,35 и 0,65 соответственно. Общая очередь S1-S2, где заявки обслуживаются согласно типу, ограничена 3-мя заявками; заявки всех типов покинувшие очередь из-за ограничения, поступают на обслуживание в S5. Обслуживание в S3-S4 заявок обоих типов равновероятно.

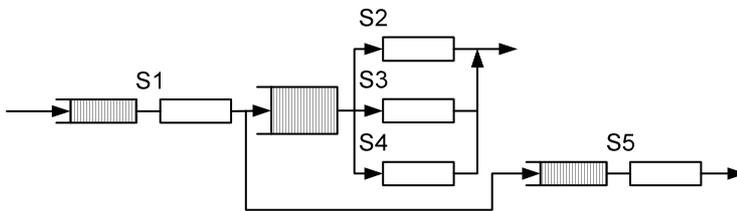
объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	равн.	эсп.
параметры	$\lambda=1$	$8 \pm 1$	$4 \pm 1$	$4 \pm 1$	$2 \pm 0,5$	$\lambda=0,2$

6.

Схема – см. вариант 5. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,7 и 0,3 соответственно. Заявки второго типа обслуживаются в S5, очередь которого ограничена 2-мя заявками; при переполнении очереди заявки теряются. Заявки первого типа в S1-S2 обслуживаются равновероятно, и далее – в S3, очередь которого ограничена 3-мя заявками. Заявки, покинувшие очередь из-за переполнения, завершают обслуживание в S4.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эсп.	равн.	равн.	равн.	равн.
параметры	$\lambda=0,5; b=3$	$\lambda=0,5$	$1\pm 0,5$	$15\pm 5$	$20\pm 5$	$16\pm 2$

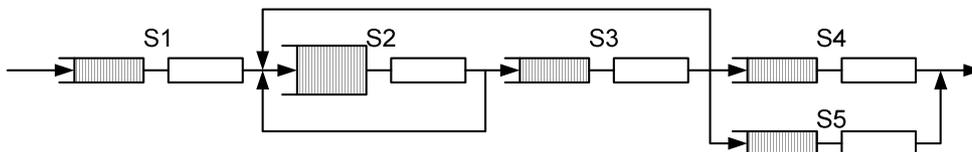
7.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,2; 0,3 и 0,5 соответственно. Общая очередь S2-S4, где заявки обслуживаются согласно типу, ограничена 3-мя заявками. Заявки первого и второго типов при переполнении очереди теряются; третьего типа – завершают обслуживание в S5.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	эсп.	эсп.
параметры	$1\pm 0,5$	$1\pm 0,5$	$12\pm 2$	$10\pm 5$	$\lambda=0,2$	$\lambda=0,2$

8.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,3 и 0,7 соответственно. После обслуживания в S2 заявки второго типа с вероятностью 0,1 возвращаются на повторное обслуживание. После обслуживания в S3 заявки всех типов с вероятностью 0,1 возвращаются на повторное обслуживание. В S4-S5 заявки обслуживаются согласно типу.

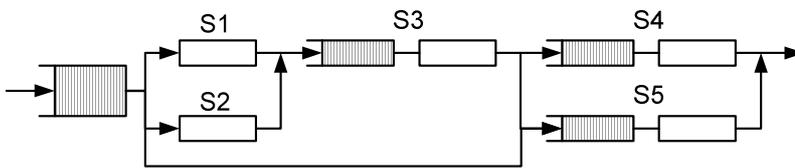
объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	эксп.	равн.	равн.	гамма	равн.	равн.
закон распределения	эксп.	равн.	равн.	гамма	равн.	равн.
параметры	$\lambda=0,2$	$5\pm 2$	$2\pm 0,5$	$\lambda=0,5; b=3$	$15\pm 5$	$10\pm 2$

9.

Схема – см. вариант 8. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,2; 0,3 и 0,5 соответственно. После обслуживания в S2 заявки первого типа с вероятностью 0,05 возвращаются на повторное обслуживание. После обслуживания в S3 заявки второго типа с вероятностью 0,1 возвращаются на повторное обслуживание. Очередь S4 ограничена 2-мя заявками. В S5 обслуживаются все заявки, покинувшие очередь S4 из-за переполнения, и половина всех заявок третьего типа.

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	гамма	равн.	равн.	эксп.	эксп.	равн.
закон распределения	гамма	равн.	равн.	эксп.	эксп.	равн.
параметры	$\lambda=1; b=3$	$1\pm 0,5$	$3\pm 1$	$\lambda=0,5$	$\lambda=0,15$	$2\pm 0,5$

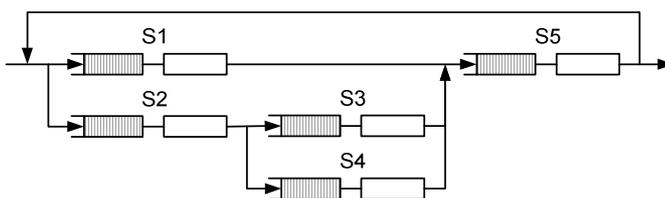
10.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно. Заявки первого и второго типов обслуживаются в S1 и S2 соответственно, заявки третьего типа направляются в S5, где также обслуживаются заявки, покинувшие очередь S4 из-за переполнения (ограничена 3-мя заявками).

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	эксп.	равн.	эксп.	равн.	эксп.	равн.
закон распределения	эксп.	равн.	эксп.	равн.	эксп.	равн.
параметры	$\lambda=0,5$	$5\pm 1$	$\lambda=0,25$	$2\pm 1$	$\lambda=0,125$	$1\pm 0,5$

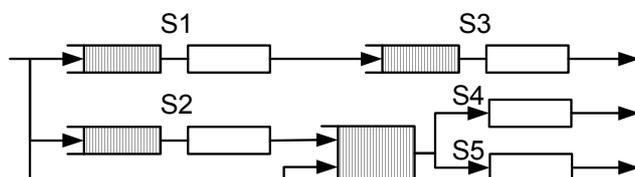
11.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,1; 0,4 и 0,5 соответственно. Заявки первого типа обслуживаются в S1, второго и третьего – в S2, и, затем, соответственно типу, в S3-S4. Очередь S4 ограничена 2-мя заявками, заявки, покинувшие очередь из-за переполнения, теряются. С вероятностью 0,1 заявки всех типов с выхода системы возвращаются на вход.

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	эксп.	равн.	эксп.	равн.	равн.	равн.
параметры	$\lambda=0,5$	$5\pm 1$	$\lambda=0,5$	$8\pm 2$	$8\pm 2$	$8\pm 2$

12.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Заявки первого типа обслуживаются в S1, очередь которого ограничена 3-мя заявками. Заявки, покинувшие очередь из-за переполнения, отправляются в общую очередь S4-S5. Заявки второго типа обслуживаются в S2 и, далее, в S5.

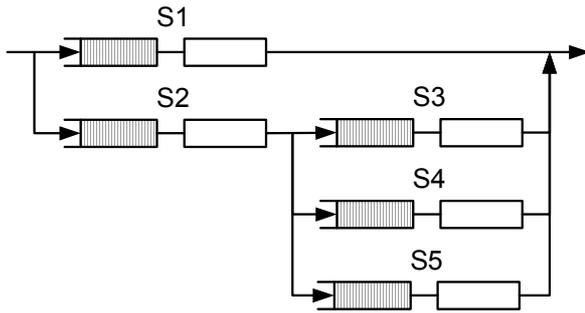
объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	равн.	эксп.	равн.	равн.	равн.	эксп.
параметры	$1\pm 0,5$	$\lambda=0,25$	$10\pm 2$	$10\pm 2$	$10\pm 2$	$\lambda=0,5$

13.

Схема – см. вариант 12. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,3 и 0,4 соответственно. Заявки первого типа обслуживаются в S1, второго – равновероятно в S1 и S2, третьего – в S2. Вероятность обслуживания заявок всех типов в S4 равна 0,7; в S5 – 0,2, с остаточной вероятностью заявки теряются.

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика	гамма	равн.	эксп.	эксп.	эксп.	эксп.
параметры	$\lambda=1; b=4$	$1\pm 0,5$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$

14.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно, и обслуживаются с вероятностью 0,4 в S1, с вероятностью 0,6 – в S2. Кроме того, очередь S1 ограничена 3-мя заявками; заявки, покинувшие очередь из-за переполнения, также направляются в S2. В S3-S5 обслуживание осуществляется соответственно типу.

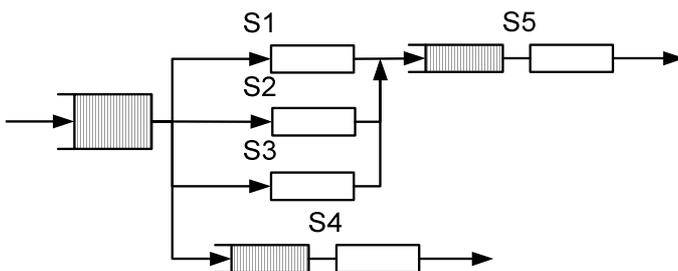
объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	эсп.	эсп.
параметры	$1 \pm 0,5$	$9 \pm 1$	$5 \pm 1$	$5 \pm 1$	$\lambda = 0,5$	$\lambda = 0,1$

15.

Схема – см. вариант 14. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Очередь S1 ограничена 2-мя заявками. Заявки, покинувшие очередь из-за переполнения, направляются в S2, и далее – в S3, очередь которого ограничена 3-мя заявками. Заявки первого типа, покинувшие очередь из-за переполнения, обслуживаются в S4, второго типа – в S5.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	равн.	эсп.
параметры	$\lambda = 1$	$4 \pm 0,5$	$2 \pm 0,5$	$15 \pm 0,5$	$2 \pm 0,5$	$\lambda = 0,8$

16.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,25; 0,5 и 0,25 соответственно. Очередь системы S1-S3, в которой заявки обслуживаются соответственно типу, ограничена 5-ю заявками. Заявки, покинувшие очередь из-за переполнения, направляются в S4 с вероятностью 0,87; остальные – теряются.

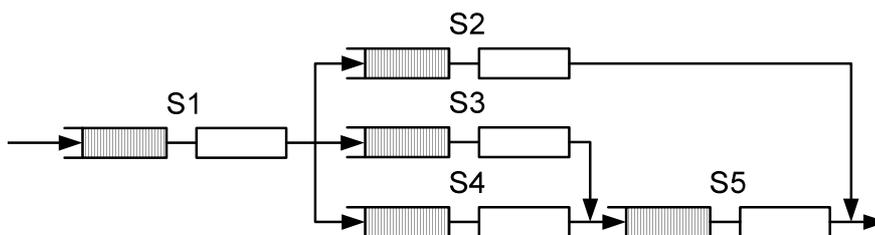
объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	равн.	равн.	эсп.	эсп.	равн.
параметры	$\lambda=1; b=3$	$20\pm 5$	$13\pm 2$	$\lambda=0,028$	$\lambda=0,15$	$2\pm 0,5$

17.

Схема – см. вариант 16. На вход СМО поступает поток заявок 4-х типов, с вероятностями поступления 0,1; 0,2; 0,3 и 0,4 соответственно. Заявки первого типа с вероятностью 0,8 обслуживаются в S1, и с вероятностью 0,2 – в S2. Заявки второго типа равновероятно обслуживаются в S2 и S3. Заявки третьего типа – в S3; четвертого – в S4, очередь которого ограничена 2-мя заявками, при переполнении очереди заявки теряются.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda=2; b=3$	$\lambda=0,2$	$\lambda=1$	$12\pm 1$	$16\pm 1$	$7\pm 1$

18.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,15; 0,25 и 0,6 соответственно. Заявки первого типа обслуживаются в S2, заявки второго типа равновероятно в S2 и S3, заявки третьего типа – в S3. Очередь S3 ограничена 3-мя заявками; при переполнении очереди заявки отправляются на обслуживание в S4.

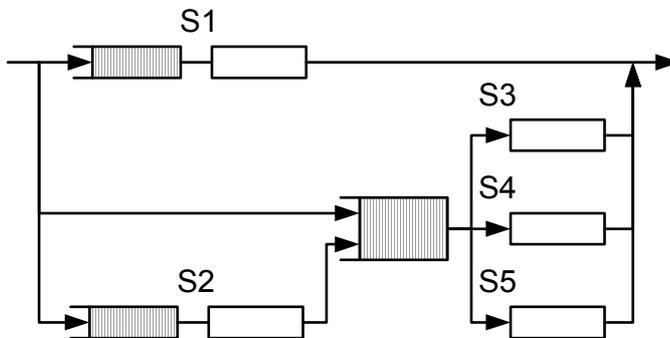
объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.	эсп.
параметры	$\lambda=1$	$1\pm 0,5$	$15\pm 2$	$25\pm 5$	$\lambda=0,2$	$\lambda=0,2$

19.

Схема – см. вариант 18. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,25 и 0,75 соответственно. Заявки всех типов обслуживаются в S1 и, далее, в S2, очередь которого ограничена 5-ю заявками. При переполнении очереди заявки первого типа отправляются в S3, второго – в S4. Очередь S5 ограничена 3-мя заявками, при переполнении очереди заявки теряются.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	эсп.	равн.	равн.	равн.
параметры	$1 \pm 0,5$	$\lambda=2$	$\lambda=0,12$	$3 \pm 0,5$	$1,2 \pm 0,2$	$4,2 \pm 1,5$

20.



На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,42; 0,48 и 0,1 соответственно. Заявки первого типа обслуживаются в S1, заявки второго типа сразу поступают на вход многоканальной системы с общей очередью, третьего типа – предварительно обслуживаются в S2. В многоканальной системе заявки направляются на обслуживание в свободное устройство с наименьшим номером.

объект / характеристика	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,05$	$12 \pm 2$	$12 \pm 2$	$12 \pm 2$

21.

Схема – см. вариант 20. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,32; 0,3 и 0,38 соответственно. Заявки всех типов с вероятностью 0,3 обслуживаются в S1, с вероятностью 0,5 сразу поступают на вход многоканальной системы с общей очередью, с остаточной вероятностью предварительно обслуживаются в S2. В многоканальной системе обслуживание заявок выполняется согласно типу.

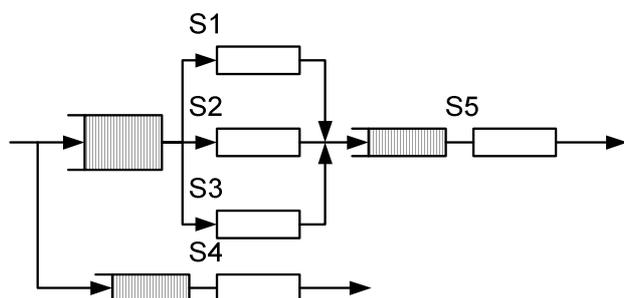
объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика						
закон распределения	равн.	равн.	гамма	эсп.	эсп.	эсп.
параметры	$1 \pm 0.2$	$3 \pm 1$	$\lambda=0,4; b=3$	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,5$

22.

Схема – см. вариант 20. На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,52 и 0,48 соответственно. Заявки второго типа обслуживаются в S2, заявки первого типа – в S1, очередь которого ограничена 3-мя заявками. При переполнении очереди заявки первого типа поступают на вход многоканальной системы с общей очередью. В многоканальной системе заявки с вероятностями 0,4 обслуживаются в S3 и S4, с остаточной – в S5.

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика						
закон распределения	равн.	эсп.	гамма	равн.	равн.	равн.
параметры	$1 \pm 0.5$	$\lambda=0,1$	$\lambda=0,35; b=2$	$2 \pm 1$	$2 \pm 1$	$2 \pm 1$

23.



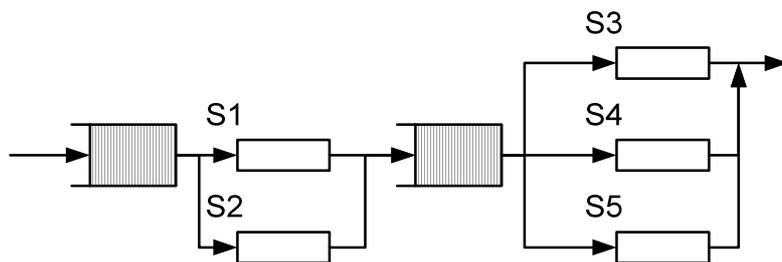
На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,22; 0,38 и 0,4 соответственно. Заявки второго типа обслуживаются в S4, первого типа – в S1, третьего типа – равновероятно в S2 и S3. Очередь S5 ограничена 5-ю заявками, при переполнении очереди заявки теряются.

объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
характеристика						
закон распределения	равн.	равн.	равн.	эсп.	равн.	эсп.
параметры	$2 \pm 1$	$4 \pm 1$	$2 \pm 0,5$	$\lambda=0,25$	$5 \pm 0,5$	$\lambda=0,045$

24.

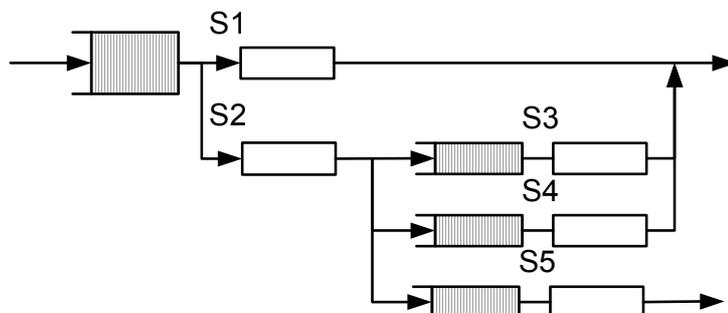
На вход СМО поступает поток заявок 4-х типов, с вероятностями поступления 0,25; 0,3; 0,22 и 0,23 соответственно. Заявки с нечетными типами обслуживаются в устройствах с нечетными номерами, во второй мно-

гоканальной системе – равновероятно. Заявки с четными типами обслуживаются равновероятно в любых устройствах.



характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эксп.	эксп.	эксп.
параметры	$1 \pm 0,5$	$2 \pm 1$	$2 \pm 1$	$\lambda = 0,25$	$\lambda = 0,5$	$\lambda = 0,75$

25.



На вход СМО поступает поток заявок 2-х типов, с вероятностями поступления 0,4 и 0,6 соответственно. Общая очередь ограничена 5-ю заявками, при переполнении очереди заявки теряются. В S1-S2 заявки обслуживаются равновероятно; в S3-S4 согласно типу. Очереди S3-S4 ограничены 3-я заявками, при переполнении очереди заявки обслуживаются в S5.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эксп.	эксп.	эксп.	равн.	равн.	гамма
параметры	$\lambda = 1$	$\lambda = 0,12$	$\lambda = 0,1$	$27 \pm 5$	$22 \pm 5$	$\lambda = 1,2; b = 3$

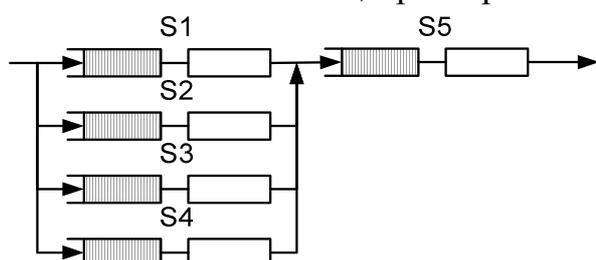
26.

Схема – см. вариант 25. На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,2; 0,4 и 0,4 соответственно. Заявки первого типа обслуживаются в S1, второго и третьего – в S2. Далее заявки второго типа с вероятностями, соответственно, 0,8 и 0,2 направляются в S3 и S4; заявки третьего типа с вероятностями, соответственно, 0,2 и 0,8 – в S4 и S5.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	равн.	равн.	равн.	эсп.
параметры	$\lambda=1$	$\lambda=0,16$	$1.2\pm 0,5$	$6\pm 2$	$4\pm 2$	$\lambda=0,2$

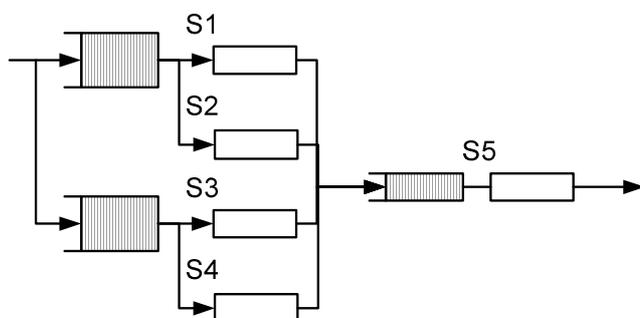
27.

На вход СМО поступает поток заявок 3-х типов, с вероятностями поступления 0,3; 0,45 и 0,25 соответственно. Заявки первого типа обслуживаются в S1 и с вероятностью 0,5 – в S2; второго типа – в S2 и с вероятностью 0,5 – в S3; третьего типа – в S3 и с вероятностью 0,5 – в S4. Очередь S5 ограничена 3-мя заявками, при переполнении очереди заявки теряются.



характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	эсп.	эсп.	эсп.	эсп.	равн.
параметры	$\lambda=1; b=3$	$\lambda=0,4$	$\lambda=0,5$	$\lambda=0,5$	$\lambda=0,1$	$10\pm 2$

28.



На вход СМО поступает поток заявок 4-х типов, с вероятностями поступления 0,2; 0,3; 0,3 и 0,2 соответственно. Заявки первого и второго типа обслуживаются в системе S1-S2, с вероятностью 0,8 в «своем» устройстве, с вероятностью 0,2 – не в «своем». Заявки третьего и четвертого типа – аналогично в системе S3-S4. Очередь S5 ограничена 5-ю заявками. При переполнении очереди заявки теряются.

характеристика \ объект	входной поток	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	равн.	эсп.
параметры	$\lambda=2$	$2\pm 0,5$	$2\pm 0,5$	$2\pm 0,5$	$2\pm 0,5$	$\lambda=0,31$

## **Лабораторная работа №5. Статистическое моделирование многоканальных СМО приоритетного обслуживания неоднородного потока заявок с ограничением времени ожидания**

*Цель:* Моделирование многоканальных многофазных СМО приоритетного обслуживания с ограничением времени ожидания. Расчет статистических характеристик по результатам моделирования. Структурная оптимизация СМО заданным способом по условию обслуживания без потерь входного потока повышенной интенсивности.

*Программное обеспечение моделирования:* GPSS World.

### **Теория**

Основные сведения по моделированию СМО в *GPSS World* – см. *Лабораторные работы №3 и №4*.

В СМО причинами потерь заявок могут являться:

- ограничение мест в очереди на обслуживание;
- приоритетность обслуживания заявок определенного типа;
- ограничение времени ожидания обслуживания или пребывания заявки в системе.

*Приоритетом* называется величина преимущественного права на обслуживание. Для каждого приоритета на входе обслуживающего устройства образуется своя очередь. Если заявка поступает на вход устройства, занятого обслуживанием заявки с более низким приоритетом, то возможно прерывание ранее начатого обслуживания – такой приоритет называется *абсолютным*. Если прерывания ранее начатого обслуживания не происходит – приоритет *относительный*.

При ограничении времени ожидания в очереди величиной  $\tau$  заявка теряется или перенаправляется в альтернативные каналы обслуживания, если она не начала обслуживаться за время за время, меньшее или равное  $\tau$ . Время пребывания – это сумма времен ожидания и времен обслуживания заявки в некотором узле СМО или в целом в системе. При ограничении времени пребывания заявка теряется или перенаправляется в альтернативные каналы обслуживания, если время ее превысило  $\tau$ .

### **Ход выполнения работы**

**1.** Модифицировать исходную СМО из задания к *Лабораторной работе №4* в соответствии с дополнительными условиями согласно варианту индивидуального задания. Реализовать модификацию. *В программной реализации обязательен построчный комментарий.*

2. Осуществить имитационное моделирование СМО. По результатам моделирования оценить стандартные характеристики СМО, а также количество и процент потерь заявок каждого типа с определением причин потерь (превышение времени ожидания или прерывание обслуживания при поступлении заявки с более высоким приоритетом).

3. Проанализировать результаты моделирования. Оценить эффективность СМО с точки зрения загрузки устройств, задержек заявок в очередях и накопителях, потерь заявок. Определить максимально возможную интенсивность входного потока для обслуживания без потерь.

4. Предложить и реализовать вариант структурной оптимизации СМО, гарантирующей безочередное обслуживание без потерь для входного потока с интенсивностью, на 25% превышающей значение, определенное в п.3 хода выполнения работы.

5. Результаты выполнения работы по пунктам 1-4 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс*.

### **Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования**

С каждой заявкой в имитационной модели GPSS связан упорядоченный набор данных. Он включает номер заявки; номер оператора, в котором в данный момент находится заявка; номер следующего оператора; время перехода в следующий оператор; приоритет, характеризующий очередность обработки заявок в определенных случаях; а также набор параметров, которые можно создавать и которым необходимо присваивать значения до того, как они будут использоваться.

*Операторы, изменяющие значения параметров заявок.* Для присвоения или изменения значения параметра используется оператор ASSIGN:

ASSIGN A,B,C

где А – номер присваиваемого или изменяемого параметра; В – новое значение параметра; С – номер или имя модификатора-функции (не обязательный операнд).

Возможные форматы использования:

- ASSIGN 2,NAP – второй параметр заявки получит значение NAP;
- ASSIGN 3+,V4 – изменится значение третьего параметра заявки; к нему прибавится значение переменной V4;

- ASSIGN 3-,5.8,7 – изменится значение третьего параметра заявки (из него вычитается произведение 5.8 на текущее значение функции 7);
- ASSIGN Name,«Plan» – параметру с именем Name присваивается текстовое значение Plan.

Стандартный числовой атрибут, связанный с параметром заявки может быть использован в условных переходах (P<номер> – возвращает значение параметра активной заявки).

Для установок меток времени используется оператор MARK. при прохождении этого оператора в параметр активной заявки записывается значение абсолютного модельного времени:

### MARK A

где A – номер параметра (необязательно). Возможные форматы:

- MARK – если оператор используется без операнда A, то метка времени активной заявки становится равной абсолютному времени моделирования (см. *Пример 1*, вариант 1);
- MARK 20 – параметр с номер 20 активной заявки принимает значение абсолютного времени моделирования (см. *Пример 1*, вариант 2 и 3).

Оператор MARK можно использовать, если необходимо, например, определить время суммарное время обработки на нескольких этапах. Для этого с помощью оператора MARK нужно записать абсолютное модельное время начала и конца обработки, а затем найти разность, которая и будет искомым временем.

Стандартные числовые атрибуты, связанные с метками времени (могут быть использованы в условных переходах):

- MP<номер> – возвращает время прохождения заявки, определяемое как разница между абсолютным временем моделирования и временем, значение которого записано в параметр активной заявки с указанным номером (см. *Пример 1*, вариант 3);
- M1 – возвращает время прохождения заявки, определяемое как разница между абсолютным временем моделирования и меткой времени активной заявки (см. *Пример 1*, вариант 1).

Для присвоения или изменения значения приоритета активной заявки используется оператор PRIORITY:

### PRIORITY A

где A – значение присваиваемого или изменяемого приоритета. Стандартный числовой атрибут, связанный с приоритетом, может быть использован

в условных переходах: СЧА PR возвращает приоритет активной заявки (см. *Пример 2*).

*Условные или вероятностные переходы в процессе обслуживания.* Для задания в программной реализации модели СМО условных или вероятностных переходов заявок используется операторы TRANSFER, GATE (см. *Лабораторную работу №4*), а также оператор еще один оператор условной передачи управления – TEST:

TEST O A,B,C

где O – условие в виде стандартного оператора отношения; A – проверяемое значение; B – контрольное значение; C – номер или метка перемещения заявки в случае невыполнения условия (необязательно). Если операнд C не указан, заявка при невыполнении условия не сможет войти в оператор TEST (режим отказа), и будет помещена в список повторных попыток всех объектов, участвующих в проверке условия. Когда состояние любого из этих объектов изменяется, заявка активизируется и проверка осуществляется снова. Если условие выполняется, то заявке разрешается войти в оператор TEST.

В качестве условия используются стандартные операторы отношения, представленные в таблице 25.

Таблица 25

Оператор	Значение
E	равно
G	больше
GE	больше или равно
L	меньше
LE	меньше или равно
NE	неравно

Возможные форматы использования оператора TEST:

– TEST E V7,K256,LAB – если переменная V7=256 выполняется оператор, следующий сразу за данным; иначе переход к оператору с меткой LAB (см. *Пример 2*);

– TEST LE M1,2,OUT – если время прохождения заявки, определяемое как разница между абсолютным временем моделирования и меткой времени активной заявки, меньше или равно 2, выполняется оператор, сле-

дующий сразу за данным; иначе переход к оператору с меткой OUT (см. *Пример 1*, вариант 1).

*Реализация абсолютного приоритетного обслуживания.* Для безусловного прерывания раннее начатого обслуживания или прерывания при поступлении заявки с более высоким абсолютным приоритетом используется оператор PREEMPT:

PREEMPT A,B,C,D,E

где А – номер или имя устройства обслуживания; В – в зависимости от режима использования может быть PR – приоритетный; или пропускается – режим безусловного прерывания; С – номер или метка оператора нового назначения (не обязательно); D – номер параметра заявки, в котором сохраняется остаточное время обслуживания (не обязательно); E – в зависимости от формата использования может быть RE или пропускается.

Возможные форматы использования:

– PREEMPT D1,,OUT,100,RE – использование оператора в режиме безусловного прерывания (операнд В пропущен). Новая заявка прерывает ранее начатое обслуживание в устройстве D1 в независимости от приоритета; остаточное время начатого обслуживания записывается в параметр 100 заявки, обслуживание которой было прервано; заявка с прерванным обслуживанием удаляется из устройства D1 и перенаправляется в оператор с меткой OUT;

– PREEMPT D1,PR,OUT – использование оператора в режиме абсолютного приоритета. Новая заявка прерывает ранее начатое обслуживание в устройстве D1, если приоритет начатого обслуживания меньше; заявка с прерванным обслуживанием удаляется из устройства D1 и перенаправляется в оператор с меткой OUT (см. *Пример 2*).

Если обслуживающее устройство, указанное в операнде А оператора PREEMPT, на момент поступления заявки свободно от обслуживания, то действие оператора эквивалентно оператору SEIZE вне зависимости от формата использования.

Заявка, занявшая устройство обслуживания по прерыванию, освобождает его вхождением в оператор RETURN:

RETURN A

где А – номер или имя освобождаемого устройства обслуживания.

### **Пример 1. Одноканальная СМО с ограничением времени ожидания обслуживания**

Пусть заявки на обслуживание поступают в СМО через  $2 \pm 0,5$  секунды и обслуживаются  $2 \pm 1$  секунды. Время ожидания обслуживания в очереди ограничено 2 секунды. При превышении времени ожидания заявки покидают систему.

Окно имитационной модели в среде *GPSS World* для первого варианта реализации имитационной модели (с использованием абсолютной метки времени) имеет вид, представленный на рис. 78.

Во втором и третьем варианте реализации имитационной модели реализовано сохранение времени ожидания в параметр заявки с последующим использованием в условном операторе. Окна этих моделей в среде *GPSS World* представлены на рис. 79 и 80.

Окно Report с результатами моделирования СМО (стандартный отчет *GPSS World*) для третьего варианта реализации имитационной модели имеет вид, представлено на рис. 81.

```
GPSS World - [limit t_1]
File Edit Search View Command Window Help
[Icons]
GENERATE 2,0.5
MARK ;начало отсчета времени
;ожидания (метка времени)
QUEUE queue_1
SEIZE device_1
DEPART queue_1
TEST LE M1,2,OUT ;выход заявки из системы
;при превышении ограничения
ADVANCE 2,1
RELEASE device_1
TRANSFER ,OUT_OK
OUT RELEASE device_1
TERMINATE 1
OUT_OK TERMINATE 1 ;выход заявки из системы
;при корректном завершении
;обслуживания
START 1000
For Help, press F1 Report is Complete. Clock
```

Рис. 78

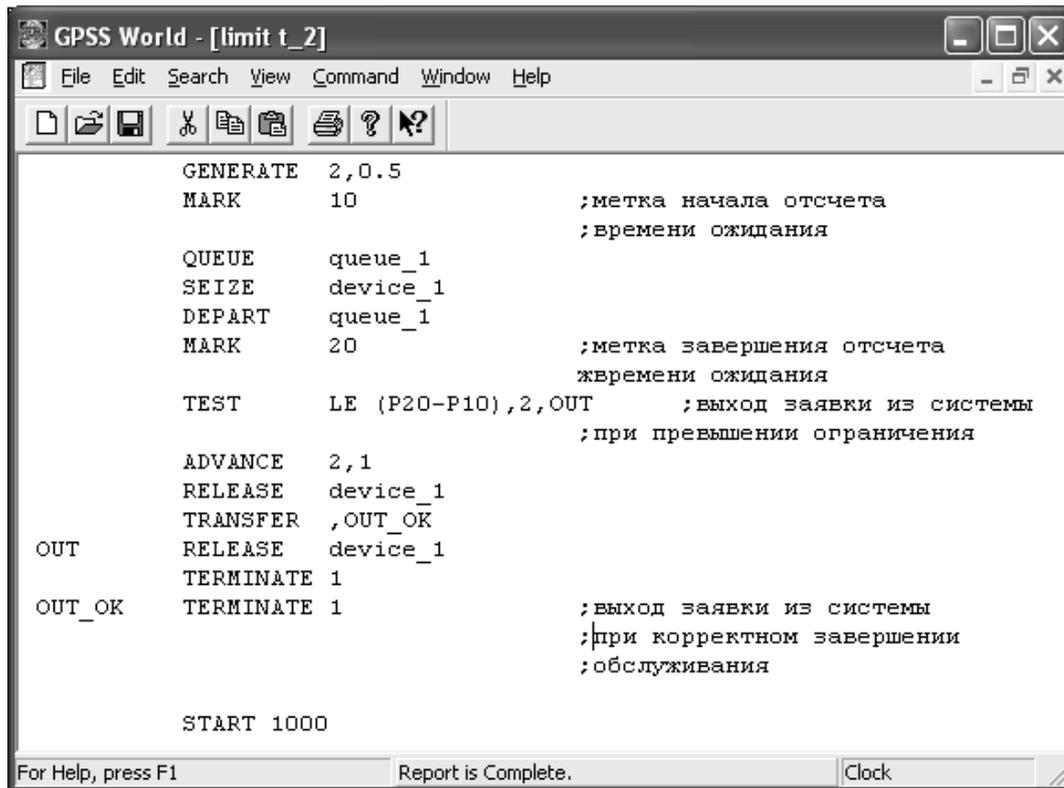


Рис. 79

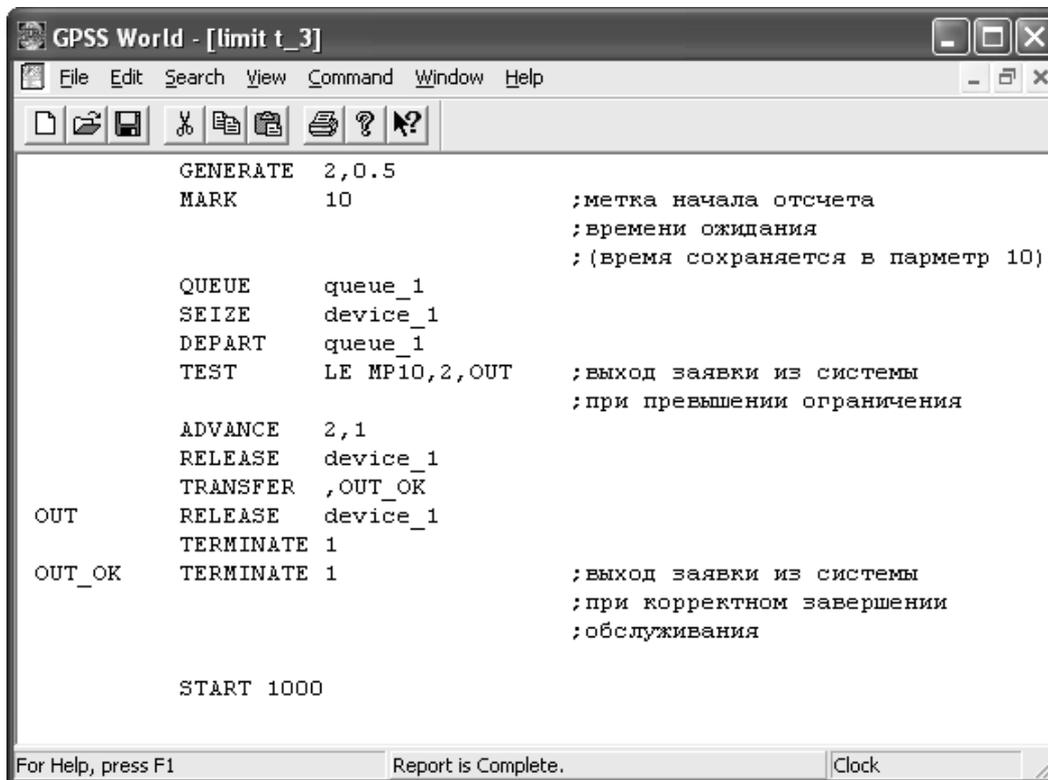


Рис. 80

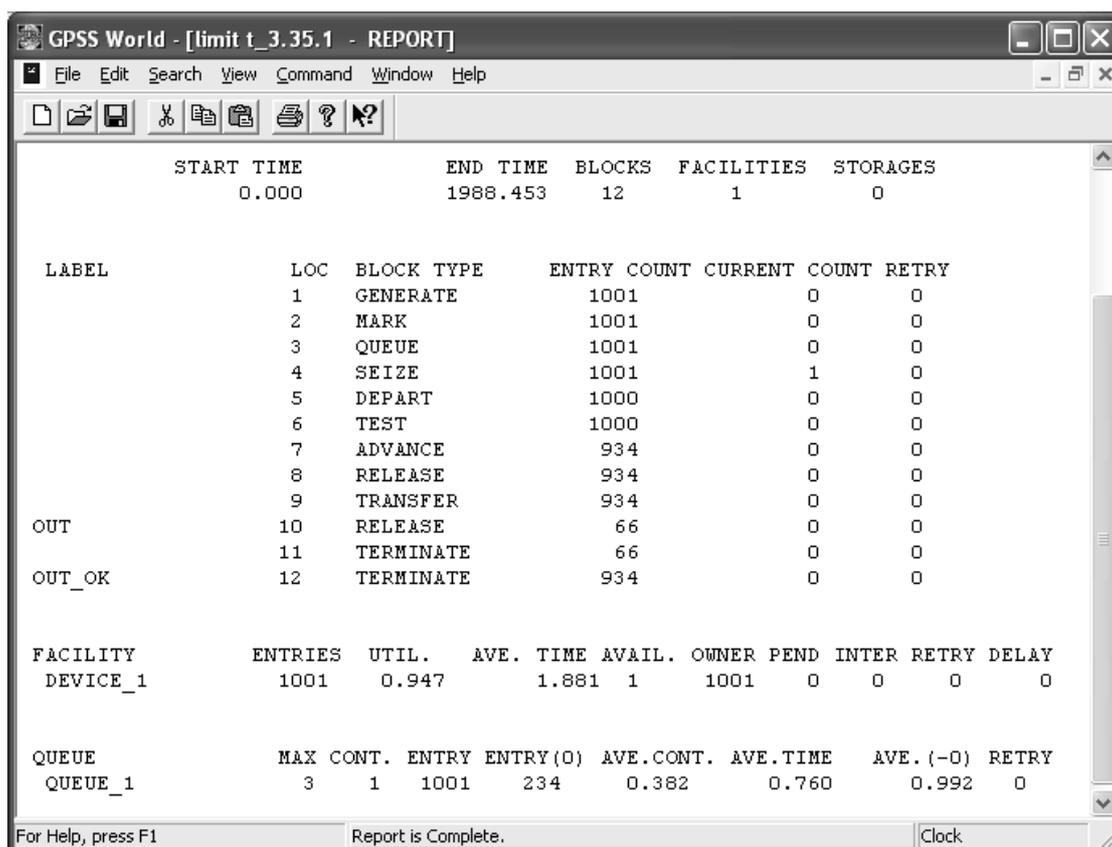


Рис. 81

Как видно из результатов моделирования, представленных на рис. 81, из 1000 заявок, покинувших очередь обслуживания (количество входов в оператор №6 TEST), корректно завершили обслуживание 934 (количество заявок, покинувших систему через оператор №12 TERMINATE с меткой OUT\_OK). Для 66 заявок время ожидания было превышено, и они покинули систему без обслуживания (количество заявок, покинувших систему через оператор №11 TERMINATE с меткой OUT).

***Пример 2. Одноканальная СМО с абсолютным приоритетом обслуживания заявок определенного типа***

На вход СМО поступает поток заявок с экспоненциальным распределением времени поступления. Заявки входного потока с вероятностями 0,4 имеют первый или второй тип, и с вероятностью 0,2 – третий. Заявки третьего типа обладают абсолютным приоритетом: если канал обслуживания на момент поступления заявки третьего типа занят заявкой первого или второго типа, обслуживание прерывается и заявка покидает систему с признаком незавершенного обслуживания.

Окно имитационной модели в среде *GPSS World* для данного примера имеет вид, показанный на рис. 82.

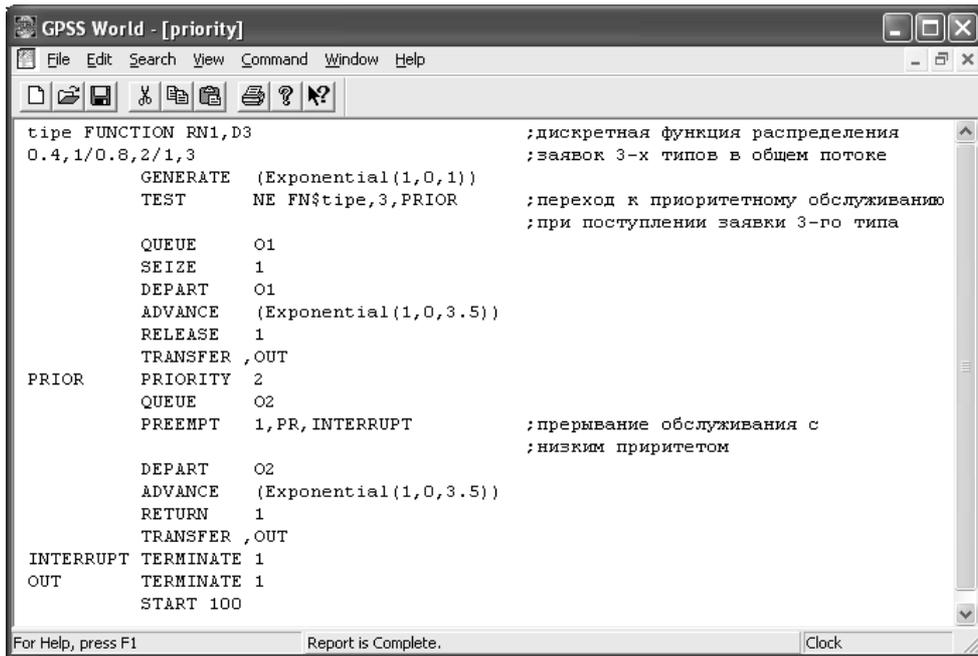


Рис. 82

Окно Report с результатами моделирования СМО имеет вид, показанный на рис. 83.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	292	0	0
	2	TEST	292	0	0
	3	QUEUE	231	189	0
	4	SEIZE	42	0	0
	5	DEPART	42	0	0
	6	ADVANCE	42	0	0
	7	RELEASE	29	0	0
	8	TRANSFER	29	0	0
PRIOR	9	PRIORITY	61	0	0
	10	QUEUE	61	2	0
	11	PREEMPT	59	1	0
	12	DEPART	58	0	0
	13	ADVANCE	58	0	0
	14	RETURN	58	0	0
	15	TRANSFER	58	0	0
INTERRUPT	16	TERMINATE	13	0	0
OUT	17	TERMINATE	87	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	101	0.999	2.932	1	282	0	0	0	191

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
O1	189	189	231	1	94.303	120.982	121.508
O2	5	3	61	13	1.080	5.246	6.666

Рис. 83

Как видно из результатов моделирования (см. рис. 83), из 292 заявок входного потока 231 оказались заявками первого и второго типа (количество входов в оператор №3 QUEUE); 61 – заявками третьего типа (количество входов в оператор №9 PRIORITY). Из 42-х заявок с низким приоритетом только 29 завершили обслуживание: количество заявок, приступивших к обслуживанию, определяется по числу входов в оператор №6 ADVANCE, а количество корректно его завершивших – по числу входов в оператор №7 RELEASE. Обслуживание 13-ти заявок было прервано поступлением заявки третьего типа, и они покинули систему через оператор №16 TERMINATE.

За время моделирования только 13 заявок третьего типа были обслужены без очереди (см. параметр ENTRY(0) для второй очереди); остальные поступали в систему в момент обслуживания заявки того же типа (с равным приоритетом), поэтому ожидали обслуживания в приоритетной очереди. Максимальное количество заявок в приоритетной очереди за время моделирования составило 5, а длина очереди на момент завершения моделирования была равна 2-м заявкам.

### **Варианты индивидуальных заданий**

*Внимание!* Индивидуальным вариантом для данной работы является модификация СМО, заданной в варианте задания к *Лабораторной работе №4*. В описании варианта указаны те изменения, которые должны быть внесены в СМО по сравнению с реализацией в работе № 4.

1. Заявки второго типа обладают абсолютным приоритетом в обслуживании. Общая очередь S4-S5 не ограничена по содержимому, но время ожидания обслуживания в S4-S5 не может превышать 3 секунды. Заявки, время ожидания обслуживания которых в S4-S5 превысило лимит, покидают систему без обслуживания.

2. Заявки обоих типов обслуживаются в S2, очередь которого не ограничена по содержимому, но время ожидания обслуживания не может превышать 5,2 секунды. Заявки, время ожидания обслуживания которых превысило лимит, с вероятностью 0,7 поступают в общую очередь устройств S3-S4, а с вероятностью 0,3 – сразу в очередь устройства S5. В S5 заявки второго типа обладают абсолютным приоритетом.

3. Очередь S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 2,6 секунды. Заявки обоих типов, время ожидания обслуживания которых превысило лимит, с вероятностью 0,7 поступают в общую очередь устройств S3-S4, а с вероятностью 0,3 – сразу в очередь устройства S5. В S1 заявки второго типа обладают абсолютным приоритетом.

4. Общая очередь S4-S5 не ограничена по содержимому, но время ожидания обслуживания не может превышать 15 секунд. Заявки, время ожидания обслуживания которых в S4-S5 превысило лимит, покидают систему без обслуживания. В устройствах S1-S3 заявки второго типа пользуются абсолютным приоритетом.

5. Общая очередь S1-S2 не ограничена по содержимому, но время ожидания обслуживания в S1-S2 не может превышать 1,05 секунды. Заявки, время ожидания обслуживания которых в S1-S2 превысило лимит, поступают на обслуживание в S5. При обслуживании в S3-S4 заявки второго типа имеют абсолютный приоритет.

6. Заявки входного потока обоих типов с равной вероятностью отправляются на обслуживание в S5 или в общую очередь S1-S2. Очередь S5 не ограничена по содержимому, но время ожидания обслуживания не может превышать 10 секунд. Заявки, время ожидания обслуживания которых превысило лимит, теряются. Заявки второго типа имеют абсолютный приоритет во всей системе, за исключением S5.

7. Общая очередь S2-S4 не ограничена по содержимому, но время ожидания обслуживания не может превышать 7,2 секунды. Заявки всех типов, время ожидания обслуживания которых превысило лимит, поступают на обслуживание в S5. В S5 заявки третьего типа обладают абсолютным приоритетом.

8. Очередь S1 не ограничена по содержимому, но время ожидания обслуживания не может превышать 2 секунды. Заявки обоих типов, время ожидания обслуживания которых превысило заданное ограничение, теряются. В дальнейшем обслуживании заявки первого типа пользуются абсолютным приоритетом.

9. Заявки второго типа имеют абсолютный приоритет. Очередь S4 не ограничена по содержимому, но время ожидания обслуживания не может превышать 5 секунд. В S5 обслуживаются все заявки, покинувшие очередь S4 из-за превышения заданного ограничения времени ожидания, и половина всех заявок третьего типа.

10. Заявки второго типа имеют абсолютный приоритет при обслуживании в S3. Очередь S4 не ограничена по содержимому, но время ожидания обслуживания не может превышать 12 секунд. Заявки, время ожидания обслуживания которых в S4 превысило лимит, поступают на обслуживание в S5.

11. Очередь S4 не ограничена по содержимому, но время ожидания обслуживания не может превышать 13 секунд. Заявки, время ожидания обслуживания которых в S4 превысило лимит, теряются. Заявки третьего типа имеют абсолютный приоритет при обслуживании в S5.

12. Заявки обоих типов обслуживаются в S1, заявки первого типа при обслуживании в S1 пользуются абсолютным приоритетом. Очередь S1 не ограничена по содержимому, но время ожидания обслуживания не может превышать 6 секунд. Заявки, время ожидания обслуживания которых в S1 превысило лимит, поступают на обслуживание в общую очередь S4-S5.

13. Заявки второго типа имеют абсолютный приоритет в обслуживании. Общая очередь S4-S5 не ограничена по содержимому, но время ожидания обслуживания не может превышать 1 секунды. Заявки, время ожидания обслуживания которых в S4-S5 превысило лимит, покидают систему без обслуживания.

14. Очередь S1 не ограничена по содержимому, но время ожидания обслуживания не может превышать 4 секунды. Заявки второго и третьего типов, время ожидания обслуживания которых превысило лимит, направляются в S2, где заявки второго типа пользуются абсолютным приоритетом. Заявки первого типа, время ожидания которых превысило заданное ограничение, теряются.

15. Заявки второго типа имеют абсолютный приоритет при обслуживании в S3. Очередь S3 не ограничена по содержимому, но время ожидания обслуживания не может превышать 25 секунд. Заявки первого типа, покинувшие очередь из-за превышения лимита ожидания, обслуживаются в S4, второго типа – в S5.

16. Общая очередь S1-S3 не ограничена по содержимому, но время ожидания обслуживания в S1-S3 не может превышать 50 секунд. Заявки, время ожидания обслуживания которых в S1-S2 превысило лимит, направляются в S4 с вероятностью 0,87; остальные – теряются. При обслуживании в S5 заявки первого типа имеют абсолютный приоритет.

17. При обслуживании в S5 заявки первого и второго типа имеют абсолютный приоритет. Очередь S4 не ограничена по содержимому, но время ожидания обслуживания не может превышать 40 секунд. Заявки, время ожидания обслуживания которых в S4 превысило лимит, покидают систему без обслуживания.

18. При обслуживании в S3 заявки второго типа имеют абсолютный приоритет. Очередь S3 не ограничена по содержимому, но время ожидания обслуживания не может превышать 1 секунды. Заявки, время ожидания обслуживания которых в S3 превысило заданное ограничение, поступают на обслуживание в S4.

19. Заявки первого типа имеют абсолютный приоритет. Очередь S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 50 секунд. Заявки первого типа, время ожидания обслуживания которых в S2 превысило лимит, отправляются в S3, второго – в S4.

20. Очередь S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 2 секунд. Заявки, время ожидания обслуживания которых в S2 превысило лимит, теряются. Заявки второго типа при обслуживании в S3-S5 имеют абсолютный приоритет.

21. Заявки третьего типа имеют абсолютный приоритет. Заявки всех типов с вероятностью 0,3 обслуживаются в S1, с вероятностью 0,5 сразу поступают на вход многоканальной системы с общей очередью, с остаточной вероятностью предварительно обслуживаются в S2. Очередь S1 не ограничена по содержимому, но время ожидания обслуживания не может превышать 5 секунд. Заявки, время ожидания обслуживания которых в S1 превысило лимит, отправляются на обслуживание в S2.

22. Очередь S1 не ограничена по содержимому, но время ожидания обслуживания не может превышать 20 секунд. Заявки, время ожидания обслуживания которых в S1 превысило лимит, поступают на вход многоканальной системы с общей очередью, при обслуживании в которой заявки первого типа пользуются абсолютным приоритетом.

23. При обслуживании в S5 заявки первого типа имеют абсолютный приоритет. Очередь S5 не ограничена по содержимому, но время ожидания обслуживания не может превышать 16 секунд. Заявки, время ожидания обслуживания которых в S5 превысило лимит, теряются.

24. Заявки с четными номерами при обслуживании в S3-S5 имеют абсолютный приоритет. Общая очередь S1-S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 8 секунд. Заявки, время ожидания обслуживания которых в S1-S2 превысило заданное ограничение, теряются.

25. Заявки первого типа пользуются абсолютным приоритетом. Общая очередь S1-S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 18 секунд. Заявки, время ожидания обслуживания которых в S1-S2 превысило лимит, теряются.

26. При обслуживании в S2 и S4 заявки третьего типа пользуются абсолютным приоритетом. Общая очередь S1-S2 не ограничена по содержимому, но время ожидания обслуживания не может превышать 36 секунд. Заявки, время ожидания обслуживания которых в S1-S2 превысило заданное ограничение, теряются.

27. Заявки второго типа в S2 и третьего типа в S3 пользуются абсолютным приоритетом. Очередь S5 не ограничена по содержимому, но время ожидания обслуживания не может превышать 25 секунд. Заявки, время ожидания обслуживания которых в S5 превысило лимит, теряются.

28. Заявки всех типов при обслуживании в «своем» устройстве пользуются абсолютным приоритетом. Очередь S5 не ограничена по содержанию, но время ожидания обслуживания не может превышать 12 секунд. Заявки, время ожидания обслуживания которых в S5 превысило заданное ограничение, теряются.

### **Лабораторная работа №6. Статистическое моделирование отказов и ошибок обслуживания в многоканальных СМО с ограничением коэффициентов использования устройств обслуживания**

*Цель:* Моделирование СМО с ограничениями коэффициентов использования устройств обслуживания. Модификация СМО по условиям периодического нарушения и восстановления работоспособности СМО в процессе обслуживания и/или возникновения ошибок обслуживания.

*Программное обеспечение моделирования:* GPSS World.

#### **Теория**

Основные сведения по моделированию СМО в *GPSS World* – см. *Лабораторные работы №3-5*.

Обслуживающее устройство (ОУ) в составе СМО в процессе эксплуатации может находиться в одном из двух режимов работы: в режиме обслуживания (ОУ занято обслуживанием очередной заявки) и в режиме простоя (ОУ свободно от обслуживания и в очереди ОУ отсутствуют заявки на обслуживание). Коэффициентом использования ОУ называется отношение суммы периодов времени, в течение которых устройство было занято обслуживанием, к общему времени функционирования СМО. Коэффициент использования выражается в долях.

В теории надежности отказом называется событие, заключающееся в нарушении работоспособности устройства. Одна из основных характеристик надежности устройства – интенсивность отказов – численно равна вероятности отказа устройства в единицу времени начиная с некоторого момента при условии, что до этого отказа не было. Среднее время безотказной работы является статистической величиной, и называется наработкой на отказ. В теории надежности чаще всего рассматривают экспоненциальное или равномерное распределение отказов.

#### **Ход выполнения работы**

**1.** Реализовать имитационную модель заданного варианта СМО с ограничениями коэффициентов использования в *GPSS World*. *В программной реализации обязателен построчный комментарий.*

**2.** Осуществить имитационное моделирование СМО в режиме увеличения (уменьшения) интенсивности экспоненциального входного потока.

Привести результаты моделирования для интенсивности, при которой подтверждаются ограничения по коэффициентам использования. По результатам моделирования оценить стандартные характеристики СМО, а также вероятность потери заявок из-за нарушений ограничений по коэффициентам использования (если есть).

3. Модифицировать исходную СМО в соответствии с дополнительными условиями по нарушениям надежности функционирования согласно варианту индивидуального задания. Реализовать модификацию. *В программной реализации обязательен построчный комментарий.*

4. Осуществить имитационное моделирование модифицированной СМО. Проанализировать результаты моделирования. Рассчитать вероятность потерь (если есть), завершения или повторного обслуживания (если есть) из-за отказов устройств и/или ошибок обслуживания.

5. Результаты выполнения работы по пунктам 1-4 и выводы занести в отчет. Правила оформления отчета – см. *Введение в курс.*

### **Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования**

*Коэффициенты использования.* Коэффициенты использования накопителей и устройств являются стандартными числовыми атрибутами *GPSS World*, и могут быть использованы в операторах условных или вероятностных переходов заявок в процессе обслуживания. Характеристика СЧА, связанных с коэффициентами использования, приведена в таблице 26.

Таблица 26

Код	Значение	Примечание
SRj	Коэффициент использования накопителя j.	Выражается в тысячных долях, но значение СЧА – целое, т. е. если коэффициент равен 0.65, то SRj равно 650.
FRj	Коэффициент использования устройства j.	Выражается в тысячных долях, но значение СЧА – целое, т. е. если коэффициент равен 0.88, то FRj равно 880.

*Арифметические и булевы переменные GPSS.* Арифметическая переменная в GPSS определяется командой:

имя VARIABLE A

где A – задаваемое пользователем выражение. Например, команда:

Pas VARIABLE P5+Q\$Bu1/5

задает переменную Pas, которая равна сумме параметра P5 заявки и текущего содержимого очереди Bu1, деленного на 5.

Значение арифметического выражения вычисляется, когда заявка входит в оператор, у которого операнд содержит одну или более ссылок на арифметические переменные. Ссылкой на арифметическую переменную является V\$имя. Имя может быть заменено номером – положительным целым числом. Для этого используется команда EDU:

имя EDU номер

Ссылка на номерную переменную имеет вид Vj, где j – номер переменной. Булева переменная определяется командой BVARIABLE:

имя BVARIABLE A

где A – задаваемое пользователем логическое выражение. В выражении используются арифметические и логические операторы. Конечный результат преобразуется в целое значение 0, если равен нулю, или в целое значение 1, если отличен от нуля. Логические операторы связаны с устройствами СМО, накопителями и логическими ключами, и используются для идентификации их состояния (см. таблицу 27).

Таблица 27

Оператор	Значение
FV	1 – если устройство доступно, иначе – 0
FI	1 – если устройство обслуживает прерывание, иначе – 0
SV	1 – если накопитель доступен, иначе – 0
SE	1 – если накопитель пуст, иначе – 0
SF	1 – если накопитель заполнен, иначе – 0
LS	1 – если логический ключ включен, иначе – 0

Примеры использования:

- Oky BVARIABLE FV\$Rem1 – булева переменная Oky равна 1, если устройство Rem1 доступно, и – 0, если не доступно;
- Can3 BVARIABLE SF\$Use1 – булева переменная Can3 равна 1, если накопитель Use1 заполнено полностью;
- Prov2 BVARIABLE LS2 – булева переменная Prov2 равна 1, если логический ключ номер 2 включен.

Операторы отношения, используемые для алгебраического сравнения операндов, представлены в таблице 28.

Примеры использования:

- Prov1 BVARIABLE V\$VseAvt'G'16 – булева переменная Prov1 равна 1, если переменная VseAvt больше 16, иначе равна 0;

– Prov2 BVARIABLE Q5\$'LE'P3 – булева переменная Prov2 равна 1, если текущая длина очереди номер 5 меньше или равна значению третьего параметра заявки;

– Prov3 BVARIABLE MX\$Plan(Stroka,Stolbez)'GE'P2 – булева переменная Prov3 равна 1, если значение определенного элемента матрицы Plan больше или равно значению параметра 2.

Таблица 28

Оператор	Значение
E	равно
G	больше
GE	больше или равно
L	меньше
LE	меньше или равно
NE	неравно

Логических операторов два: OR – ИЛИ и AND – И. Они могут быть использованы следующим образом:

– Con1 BVARIABLE FI\$Rem'OR'SF4 – булева переменная Con1 равна 1, если выполняется одно из условий: устройство Rem обслуживает прерывание или накопитель 4 заполнен;

– Con2 BVARIABLE FI\$Rem'AND'SF\$Pogr – булева переменная Con2 равна 1, если выполняются оба условия: устройство Rem обслуживает прерывание и накопитель с именем Pogr заполнен;

– Con3 BVARIABLE (V3'G'7)'AND'(FN\$Rav'OR'LS7) – булева переменная Con3 равна 1, если выполняются оба условия: значение переменной номер 3 больше 7 и логический ключ номер 7 включен;

– PROV1 BVARIABLE F1'AND'F2'AND'(FR3>900) – булева переменная PROV1 равна 1 если первое и второе устройства заняты обслуживанием и коэффициент использования третьего больше 0,9 (см. *Пример 1*).

Если булева переменная задается следующим образом:

имя1 BVARIABLE V\$имя2

то вычисляется значение арифметической переменной с именем имя2, и если оно отлично от нуля – значение булевой переменной имя1 будет равно 1, в противном случае – 0.

*Сохраняемые ячейки.* В GPSS используются ячейки памяти, начальные значения которых могут быть заданы перед моделированием и к которым можно обратиться из любого места модели во время моделирования.

Эти ячейки являются СЧА. Значения сохраняемых ячеек изменяются только при прямом указании пользователя. По желанию пользователя отдельным сохраняемым ячейкам могут быть присвоены ненулевые начальные значения с помощью команды INITIAL.

Команда INITIAL применяется при отсутствии активной заявки или же когда нельзя использовать значения параметров заявки. Она имеет следующий формат:

INITIAL A,B

где А – номер или имя сохраняемой ячейки в (формате Xположительное целое или X\$имя); В – присваиваемое первоначальное значение (не обязательно). Если не определено, то значение ячейки становится равным 1. Например:

– INITIAL X32,3895.56 – в ячейку с номером 32 заносится число 3895,56;

– INITIAL X\$Def,Q\$Pod – в ячейку с именем Def заносится текущая длина очереди с именем Pod;

– INITIAL X\$Cost,Zen1 – в ячейку с именем Cost заносится значение переменной пользователя Zen1;

– INITIAL X\$Time, "Result number ..." – в ячейку с именем Time заносится строковая константа;

– INITIAL X1 – операнд В не используется, поэтому сохраняемой ячейке номер 1 присваивается значение 1.

В процессе моделирования значение сохраняемой ячейки изменяется при входе заявки в оператор SAVEVALUE. Формат оператора:

SAVEVALUE A,B

где А – номер или имя ячейки; В – новое значение.

Оператор SAVEVALUE может быть использован как в режиме замещения, так и в режиме накопления или уменьшения. В режиме накопления предыдущее значение сохраняемой ячейки увеличивается на значение, стоящее в операнде В, а в режиме уменьшения – уменьшается. Режимы накопления или уменьшения определяются введением соответственно знака + или – перед запятой, разделяющей операнды А и В. Например:

– SAVEVALUE 5+,X2 – значение сохраняемой ячейки номер 5 при входе заявки в оператор SAVEVALUE увеличивается на значение сохраняемой ячейки номер 2;

– SAVEVALUE Vad-,V\$Hdl – значение сохраняемой ячейки с именем Vad уменьшается на вычисленное значение арифметической переменной Hdl;

– SAVEVALUE Ais,-659 – значение сохраняемой ячейки с именем Ais замещается числом -659;

– SAVEVALUE 3,(25.75#X\$Num1+Q4) – вычисляется выражение в скобках и присваивается ячейке номер 3 (в GPSS по умолчанию в качестве оператора умножения используется «#»). Для использования «\*» нужно выбрать Edit/Settings... и установить флажок в закладке Simulation).

– SAVEVALUE NUMBER,XN1 – значение сохраняемой ячейки с именем NUMBER замещается порядковым номером заявки, вошедшей в оператор SAVEVALUE (см. *Пример 4*).

Перевод в состояние недоступности объектов аппаратной категории используется для моделирования процессов возникновения и устранения отказов оборудования в процессе обслуживания.

Для моделирования неисправностей обслуживаемых устройств и других ситуаций предусмотрены операторы, реализующие недоступность и доступность устройств. При использовании этих операторов статистика функционирования устройства не искажается. Для моделирования неисправностей можно использовать и режим прерывания (оператор PREEMPT). Однако при этом заявки, вызывающие прерывания (имитирующие отказы устройства), учитываются в статистике как и заявки, обслуженные при исправном функционировании.

Недоступность устройства моделируется оператором FUNAVAIL. При использовании этого оператора статистика ОКУ не искажается. Формат оператора:

FUNAVAIL A,B,C,D,E,F,G,H

где А – номер или имя устройства обслуживания. Все заявки, обрабатываемые оператором FUNAVAIL, разделяются на три класса, которые и определяют назначение операндов:

– заявка, занимающая ОУ (по SEIZE или PREEMPT) в момент перевода его в недоступное состояние (операнды В, С, D);

– заявки, обслуживание которых было прервано, находящиеся в списке прерываний (операнды Е, F);

– заявки, находящиеся в списке отложенных прерываний и в списке задержки ОУ (операнды G, H).

Значения операндов:

– В – режим обработки заявки, занимающей ОУ в момент перевода его в недоступное состояние: СО – режим продолжения (продолжить обработку занимающей ОУ заявки во время недоступности), RE – режим удаления (удалить и направить занимающую ОУ заявку к оператору, метка

которого должна быть указана операндом С), по умолчанию – прервать обработку и поместить в список прерываний ОУ (после восстановления доступности эта заявка может занять ОУ и «дообслужиться»);

- С – метка оператора нового назначения, в который будет направлена в режиме удаления заявка, занимавшая ОУ в момент перевода его в недоступное состояние.

- D – номер или имя параметра заявки, в котором сохраняется остаточное время обслуживания для режима удаления(режим RE).

- E – режим обработки заявок, находящихся к моменту перевода ОУ в недоступное состояние в списке прерываний, т. е. тех заявок, обслуживание которых на данном ОУ было ранее прервано: CO – режим продолжения: продолжить работу ОУ во время недоступности (обслуживать заявки из списка прерываний), RE – режим удаления (удалить и направить заявки из списка прерываний к новому оператору, метка которого должна быть указана операндом F), по умолчанию – оставить ранее прерванные заявки в списке прерываний ОУ и запретить им занимать его во время недоступности;

- F – метка оператора нового назначения, в который будут направлены заявки из списка прерываний.

- G – режим обработки заявок, находящихся к моменту перевода ОУ в недоступное состояние в списке отложенных прерываний, т. е. ожидающих выполнения с прерыванием, и в списке задержки: CO – режим продолжения: продолжить работу ОУ во время недоступности (обслуживать заявки из списка отложенных прерываний и списка задержки), RE – режим удаления (удалить и направить заявки к новому оператору, метка которого должна быть указана операндом H); по умолчанию – оставить заявки в списке отложенных прерываний и списке задержки и запретить им занимать его во время недоступности;

- H – метка оператора нового назначения, в который будут направлены заявки из списка отложенных прерываний и списка задержки. Когда операнд G не используется, нельзя использовать и операнд H.

Недоступность ОУ сохраняется до тех пор, пока заявка, вызвавшая переход в недоступное состояние, не войдет в оператор:

#### FAVAIL A

где A – номер или имя устройства обслуживания. Все заявки, ожидающие доступного состояния ОУ, указанного операндом A, активизируются и могут попытаться занять его.

Перевод ОУ в недоступное состояние и разрешение продолжать обработку заявок из указанных трех классов дает возможность имитировать не только отказы, но и различные дисциплины обслуживания.

При перемещении прерванные в ОУ обслуживания не сбрасываются, т. е. заявка продолжает занимать устройство. Поэтому, если необходимо по логике модели освободить ОУ, заявка должна пройти через соответствующий оператор RELEASE.

Моделирования процессов возникновения и устранения отказов в СМО с использованием операторов FUNAVAIL/FAVAIL – см. *Пример 2*.

Недоступность объектов аппаратной категории, обладающих ограниченной емкостью (моделируются оператором STORAGE), задается оператором SUNAVAIL, формат которого:

SUNAVAIL A

где A – имя или номер накопителя. Когда заявка входит в такой оператор, накопитель, имя которого задано операндом A становится недоступным. Если при переводе в недоступное состояние в накопителе находились заявки, то их обслуживание продолжается до тех пор, пока текущее содержимое накопителя не станет равным нулю. Заявки, которые пытаются занять накопитель во время нахождения его в недоступном состоянии, задерживаются на входе в оператор ENTER. Нахождение в недоступном состоянии продолжается до тех пор, пока заявка не войдет в оператор SAVAIL. Формат оператора:

SAVAIL A

где A – имя или номер накопителя. Если в момент перевода накопителя в доступное состояние в его списке задержки были заявки, им предоставляется возможность занять накопитель с учетом его ограниченности.

Пример моделирования процессов возникновения и устранения отказов доступа ограниченной очереди обслуживания с использованием операторов SUNAVAIL/SAVAIL – см. *Пример 3*.

*Поиск и перемещение заявки по номеру.* Для нахождения любой заявки предназначен оператор DISPLACE:

DISPLACE A,B,C,D

где A – номер заявки, которую необходимо переместить; B – метка оператора, к которому перемещается заявка, указанная операндом A; C – номер параметра перемещаемой заявки, в который записывается оставшееся до конца обслуживания время, если она находилась в списке будущих событий (не обязательно); D – метка альтернативного оператора для активной заявки (не обязательно). Если операнд D не указан, активная заявка переходит к следующему оператору.

Например: DISPLACE (P2+32),Term3,Ostatok,Met2 – операнд A указан выражением в скобках. Это выражение вычисляется и округляется до

целого. Полученный результат является номером заявки, которую следует переместить. При этом возможны случаи:

- заявка есть в модели и не находится в списке будущих событий;
- заявка есть в модели и находится в списке будущих событий;
- заявки с нужным номером нет в модели.

В первом случае заявка перемещается к оператору с меткой Term3. Во втором случае определяется время, оставшееся до ее повторного ввода в процесс моделирования, и записывается в параметр с именем Ostatok. Если параметра с таким именем нет, он создается. Заявка также перемещается к оператору с меткой Term3. В третьем случае, т. е. когда в модели нет заявки с нужным номером, активная заявка, вошедшая в оператор DISPLACE, направляется к оператору с меткой Met2. Аналогично, например: DISPLACE X\$NUMBER,OUT – заявка, номер которой сохранен в ячейке с именем NUMBER, перемещается к оператору с меткой OUT (см. *Пример 4*). При перемещении прерванные в ОУ обслуживания не сбрасываются, т. е. заявка продолжает занимать устройство. Поэтому, если необходимо по логике модели освободить ОУ, заявка должна пройти через соответствующий оператор RELEASE.

***Пример 1. Трехканальная СМО с распределением обслуживания в зависимости от коэффициентов использования устройств.***

Пусть в заданной трехканальной СМО с параллельными устройствами обслуживания 1, 2 и 3 распределение заявок осуществляется в зависимости от коэффициентов использования устройств: для устройства 1 он не должен превышать 0,5, а для устройства 2 – 0,6. Вся остальная нагрузка приходится на устройство 3, причем если коэффициент его использования составит 0,9, то заявки теряются без обслуживания.

Окно первой части имитационной модели в среде *GPSS World* имеет вид, показанный на рис. 84. В этой части осуществлена реализация блока задания ограничений коэффициентов использования устройств обслуживания, а также блока генерации заявок входного потока и блока проверки ограничений для определения пути перенаправления заявки. Окно второй части имитационной модели в среде *GPSS World* имеет вид, приведенный на рис. 85. В этой части осуществлена реализация обслуживания заявок в устройствах 1-3 с проверкой условий перенаправления после завершения очередного обслуживания. Временные характеристики обслуживания подобраны таким образом, чтобы в СМО срабатывали все ограничения по КИ устройств.

```

GPSS World - [ЛР7_пример1]
File Edit Search View Command Window Help
; Задание булевых переменных
PROV1    BVARIABLE F1'AND'F2'AND' (FR3>900) ;1 если первое и второе устройства
;заняты и КИ третьего больше 0,9
PROV2    BVARIABLE F1'OR' (FR1>500)
;1 если первое устройство занято
;обслуживанием, или его КИ>0,5
PROV3    BVARIABLE F2'OR' (FR2>600)
;1 если второе устройство занято
;обслуживанием, или его КИ>0,6
PROV4    BVARIABLE FR3>900
;1 если КИ третьего устройства
;больше 0,9

; Генерация заявок входного потока
GENERATE (Exponential(1,0,6.3))
TEST E   BV$PROV1,0,LOSS
;если PROV1=1 переход к метке LOSS,
;иначе - к следующему блоку
TEST E   BV$PROV2,0,MET2
;если устройство 1 занято или его
;КИ>0,5 заявка пытается занять
;устройство 2

For Help, press F1
Report is Complete.
Clock

```

Рис. 84

```

GPSS World - [ЛР7_пример1]
File Edit Search View Command Window Help
; Устройство 1
QUEUE    1
SEIZE    1
DEPART   1
ADVANCE  38,7.3
RELEASE  1
TERMINATE 1
;выход заявок, обслуженных
;устройством 1

; Устройство 2
MET2     TEST E   BV$PROV3,0,MET3
;если устройство 2 занято или его
;КИ>0,6 заявка пытается занять
;устройство 3
QUEUE    2
SEIZE    2
DEPART   2
ADVANCE  38,9.35
RELEASE  2
TERMINATE 1
;выход заявок, обслуженных
;устройством 2

; Устройство 3
MET3     TEST E   BV$PROV4,0,LOSS
;если КИ устройства 3 больше 0,9
;завка теряется
QUEUE    3
SEIZE    3
DEPART   3
ADVANCE  (Exponential(1,0,12.3))
RELEASE  3
TERMINATE 1
;выход заявок, обслуженных
;устройством 3

LOSS     TERMINATE 1
;потерянные заявки
START    5000

For Help, press F1
Report is Complete.
Clock

```

Рис. 85

Окно Report с результатами моделирования СМО имеет вид, приведенный на рис. 86. Как видно, коэффициенты использования устройств оказались такими, как и было указано в задании. Из 5003 заявок, поступивших в СМО, 1784 были потеряны без обслуживания, 417 – обслужены первым устройством; 498 – вторым устройством и 2301 – третьим (количество входов заявок в операторы TERMINATE №9, 16 и 24 соответственно). На момент завершения моделирования одна заявка не завершила обслуживание в третьем устройстве, и две находились в очереди на обслуживание.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	5003	0	0	
	2	TEST	5003	0	0	
	3	TEST	4367	0	0	
	4	QUEUE	417	0	0	
	5	SEIZE	417	0	0	
	6	DEPART	417	0	0	
	7	ADVANCE	417	0	0	
	8	RELEASE	417	0	0	
	9	TERMINATE	417	0	0	
MET2	10	TEST	3950	0	0	
	11	QUEUE	498	0	0	
	12	SEIZE	498	0	0	
	13	DEPART	498	0	0	
	14	ADVANCE	498	0	0	
	15	RELEASE	498	0	0	
	16	TERMINATE	498	0	0	
MET3	17	TEST	3452	0	0	
	18	QUEUE	2304	2	0	
	19	SEIZE	2302	1	0	
	20	DEPART	2301	0	0	
	21	ADVANCE	2301	0	0	
	22	RELEASE	2301	0	0	
	23	TERMINATE	2301	0	0	
LOSS	24	TERMINATE	1784	0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
1	417	0.500	37.838	1	0	0	0	0	0
2	498	0.600	37.988	1	0	0	0	0	0
3	2302	0.900	12.323	1	5001	0	0	0	2

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
1	1	0	417	417	0.000	0.000	0.000	0
2	1	0	498	498	0.000	0.000	0.000	0
3	48	3	2304	208	8.133	111.301	122.346	0

For Help, press F1      Report is Complete.      Clock

Рис. 86

## **Пример 2. Двухканальная СМО с резервным каналом для обслуживания во время отказов основного канала**

Пусть время поступления заявок в СМО распределено по экспоненциальному закону с МО 3 секунды. Заявки обслуживаются в основном канале за время, распределенное по экспоненциальному закону, в среднем за 2 секунды. Отказы основного канала происходят через интервалы времени, распределенные по экспоненциальному закону, с МО 15 секунд. Если отказ возникает во время передачи, то резервный канал продолжает обслуживание прерванной заявки. Время восстановления основного канала распределено по экспоненциальному закону с МО 5 секунд. После восстановления резервный канал отключается и основной продолжает работу с очередной заявки. Окно первой части имитационной модели (блок генерации заявок и блок обслуживания в каналах) для данного примера в среде GPSS World имеет вид, приведенный на рис. 87.

```
; Генерация заявок входного потока
GENERATE (Exponential(1,0,3))
GATE FV OsnK,Rez ; проверка доступности основного
; канала, если недоступен - переход
; к резервному

; Основной канал
QUEUE QOsnK
SEIZE OsnK
DEPART QOsnK
ADVANCE (Exponential(1,0,2))
RELEASE OsnK
TERMINATE 1 ; выход заявок, обслуженных в
; основном канале

; Резервный канал
Rez QUEUE QRezK
SEIZE RezK
DEPART QRezK
ADVANCE (Exponential(1,0,2))
RELEASE RezK
TERMINATE 1 ; выход заявок, обслуженных в
; резервном канале

RezPR RELEASE OsnK ; освобождение основного канала от
; прерванного обслуживания

QUEUE QRezK
SEIZE RezK
DEPART QRezK
ADVANCE P10
RELEASE RezK
TERMINATE 1 ; выход завершеного обслуживания
; после прерывания в момент отказа
; осн. канала
```

Рис. 87

Окно второй части имитационной модели (блок моделирования отказов и восстановления основного канала) для данного примера в среде GPSS World имеет вид, приведенный на рис. 88.

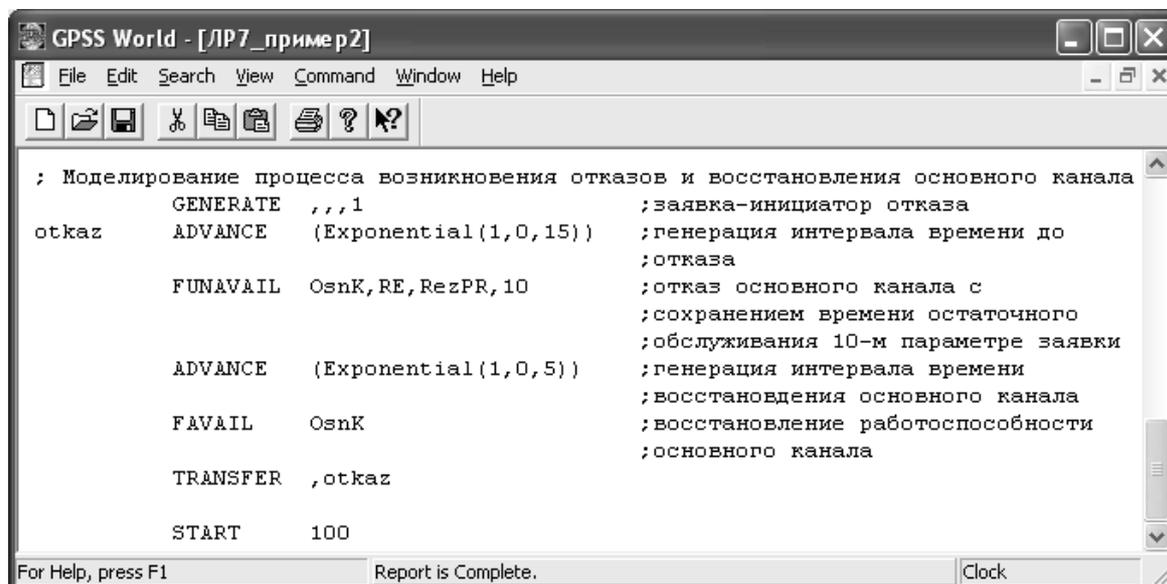


Рис. 88

Окно Report с результатами моделирования заданной СМО имеет следующий вид, приведенный на рис. 89. Как видно, из 100 заявок, завершивших обслуживание в системе, 55 начали и завершили обслуживание в основном канале (количество входов в оператор №8 TERMINATE); 34 заявки вошли в систему во время отказов основного канала, поэтому были обслужены в резервном канале (количество входов в оператор №14 TERMINATE); а 11 заявок начали обслуживание в основном канале, но оно было прервано из-за возникновения отказа, поэтому завершение остаточного обслуживания было осуществлено в резервном канале (количество входов в оператор №21 TERMINATE). Всего за время моделирования отказ основного канала возникал 17 раз (количество входов в оператор №25 ADVANCE).

***Пример 3. Двухканальная СМО с резервным каналом для обслуживания во время отказов доступа к ограниченной очереди основного канала***

Пусть в двухканальной СМО очередь основного канала ограничена 5-ю заявками, очередь резервного канала – неограниченна. Заявки поступают в систему через  $3 \pm 1$  секунды и требуют обслуживания в течении  $6 \pm 1$  секунд. Резервный канал используется при переполнении основной очереди

ди, а также во время отказов доступа к основной очереди. Отказы наступают в среднем через  $12 \pm 1$  секунд; время восстановления составляет 1 секунду. Отказ не нарушает работоспособность очереди, он распространяется только на доступность. Если на момент наступления отказа в очереди имеются заявки, обслуживание продолжается основным каналом до освобождения очереди, и только после этого начинается восстановление.

GPSS World - [ЛР7\_пример2.28.1 - REPORT]

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	103	0	0	
	2	GATE	103	0	0	
	3	QUEUE	69	2	0	
	4	SEIZE	67	0	0	
	5	DEPART	67	0	0	
	6	ADVANCE	67	1	0	
	7	RELEASE	55	0	0	
	8	TERMINATE	55	0	0	
REZ	9	QUEUE	34	0	0	
	10	SEIZE	34	0	0	
	11	DEPART	34	0	0	
	12	ADVANCE	34	0	0	
	13	RELEASE	34	0	0	
	14	TERMINATE	34	0	0	
REZPR	15	RELEASE	11	0	0	
	16	QUEUE	11	0	0	
	17	SEIZE	11	0	0	
	18	DEPART	11	0	0	
	19	ADVANCE	11	0	0	
	20	RELEASE	11	0	0	
	21	TERMINATE	11	0	0	
OTKAZ	22	GENERATE	1	0	0	
	23	ADVANCE	18	1	0	
	24	FUNAVAIL	17	0	0	
	25	ADVANCE	17	0	0	
	26	FAVAIL	17	0	0	
	27	TRANSFER	17	0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
OSNK	67	0.358	1.590	1	98	0	0	0	2
REZK	45	0.358	2.370	1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
QOSNK	8	2	69	21	1.449	6.257	8.994	0
QREZK	3	0	45	23	0.187	1.240	2.536	0

For Help, press F1      Report is Complete.      Clock

Рис. 89

Окно имитационной модели для заданного *Примера №3* (моделирование двухканальной СМО с резервным каналом для обслуживания во время отказов доступа к ограниченной очереди основного канала) в среде *GPSS World* имеет вид, показанный на рис. 90.

```

GPSS World - [ЛР7_пример3]
File Edit Search View Command Window Help

;Задание емкости накопителя
NAK STORAGE 5 ;очередь NAK ограничена 3-мя заявками
;Задание булевой переменной
PROV1 BVARIABLE SF$NAK ;1 если накопитель заполнен
PROV2 BVARIABLE SV$NAK ;1 если накопитель доступен

GENERATE 3,1
TEST E BV$PROV1,0,RESERVE ;если PROV=1 переход к резервному
;каналу
TEST E BV$PROV2,1,RESERVE ;если PROV=0 переход к резервному
;каналу

ENTER NAK
SEIZE DEVICE_S1 ;основное обслуживание в S1
LEAVE NAK
ADVANCE 6,1
RELEASE DEVICE_S1
TERMINATE 1
RESERVE QUEUE O2
SEIZE DEVICE_S2 ;резервное обслуживание в S2
DEPART O2
ADVANCE 6,1
RELEASE DEVICE_S2
TERMINATE 1

;Моделирование процесса возникновения отказов и восстановления накопителя
GENERATE ,,1 ;заявка-инициатор отказа
MET ADVANCE 12,1 ;моделирование времени до наступления
;отказа
SUNAVAIL NAK ;моделирование (перевод в состояние
;недоступности)
GATE SE NAK ;накопитель пуст?
ADVANCE 1 ;моделирование времени
;восстановления
SAVAIL NAK ;перевод в состояние доступности
TRANSFER ,MET ;отправить на очередной цикл

START 100

For Help, press F1 Report is Complete. Clock

```

Рис. 90

Окно Report с результатами моделирования СМО имеет следующий вид, представленный на рис. 91. Как видно, из 100 заявок, завершивших обслуживание в системе, 53 были обслужены основным каналом (количество входов в оператор №9 TERMINATE), а 47 – резервным (количество входов в оператор №15 TERMINATE). За время моделирования отказы наступали 13 раз (количество входов в оператор №18 SANAVAIL). В результате, несмотря на достаточную емкость ограниченной очереди основного канала, коэффициент использования которой составил всего 0,271, на момент завершения моделирования 6 заявок ожидают обслуживания в очереди резервного канала (см. параметр CURRENT COUNT для оператора №10

QUEUE). На момент остановки моделирования начался отсчет времени до наступления очередного отказа, но этот интервал еще не истек, поэтому заявка-инициатор отказа задержана в операторе №17 ADVANCE.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	109	0	0
	2	TEST	109	0	0
	3	TEST	109	0	0
	4	ENTER	55	1	0
	5	SEIZE	54	1	0
	6	LEAVE	53	0	0
	7	ADVANCE	53	0	0
	8	RELEASE	53	0	0
	9	TERMINATE	53	0	0
RESERVE	10	QUEUE	54	6	0
	11	SEIZE	48	0	0
	12	DEPART	48	0	0
	13	ADVANCE	48	1	0
	14	RELEASE	47	0	0
	15	TERMINATE	47	0	0
MET	16	GENERATE	1	0	0
	17	ADVANCE	14	1	0
	18	SUNAVAIL	13	0	0
	19	GATE	13	0	0
	20	ADVANCE	13	0	0
	21	SAVAIL	13	0	0
	22	TRANSFER	13	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
DEVICE_S1	54	0.989	5.920	1	109	0	0	0	1
DEVICE_S2	48	0.902	6.078	1	98	0	0	0	6

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
O2	7	6	54	4	3.056	18.302	19.766

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
NAK	5	3	0	4	55	1	1.354	0.271	0	0

For Help, press F1      Report is Complete.      Clock

Рис. 91

**Пример 4. Одноканальная СМО с ошибками обслуживания**

Поток заявок с экспоненциальным распределением времени поступления обслуживается в одноканальной СМО. Время обслуживания заявки составляет  $2 \pm 0,5$  секунды. Периодически в системе возникают ошибки обслуживания. Время возникновения ошибок распределено равномерно в интервале  $10 \pm 1$  секунд. При возникновении ошибки обслуживание прерывается, а заявка теряется.

Окно имитационной модели для заданного примера в среде *GPSS World* имеет вид, представленный на рис. 92.

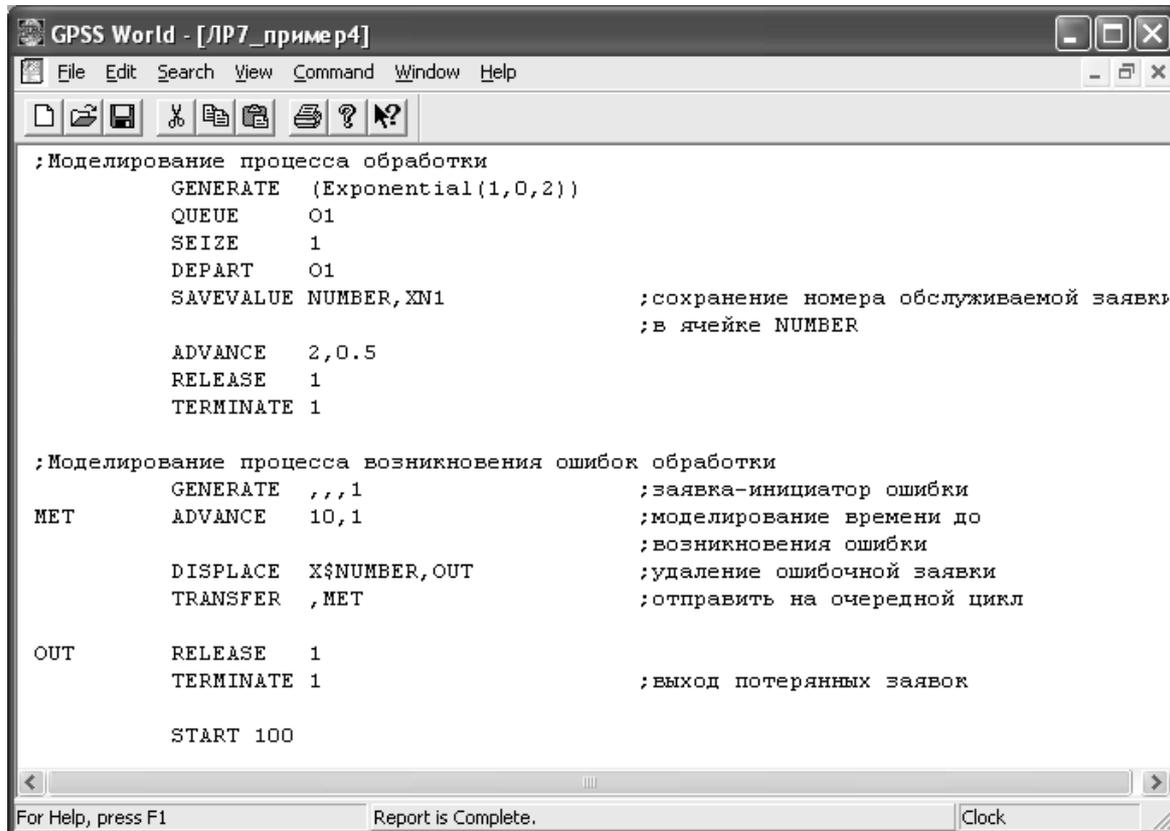


Рис. 92

Окно Report с результатами моделирования СМО имеет следующий вид, представленный на рис. 93. Как видно, на момент завершения моделирования 109 заявок входного потока поступили в систему (количество входов в оператор №1 GENERATE). Из 100 заявок, завершивших обслуживание в системе, 82 были обслужены безошибочно (количество заявок, покинувших систему через оператор №8 TERMINATE), а 18 были потеряны (количество заявок, покинувших систему через оператор №14 TERMINATE). При этом ошибки обслуживания в процессе моделирования возникали 20 раз (количество входов в оператор №20 DISPLACE), однако в двух случаях из 20-ти во время генерации ошибки, заявок на обслуживании в устройстве не было. Количество таких случаев оказалось невелико, так как устройство обслуживания загружено значительно: коэффициент его использования составил 0,867. На момент завершения моделирования начался уже отсчет времени до наступления очередной ошибки обслуживания, но время наступления не успело истечь, поэтому заявка-инициатор отказа задержана в операторе № 10 ADVANCE.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	109	0	0
	2	QUEUE	109	8	0
	3	SEIZE	101	1	0
	4	DEPART	100	0	0
	5	SAVEVALUE	100	0	0
	6	ADVANCE	100	0	0
	7	RELEASE	82	0	0
	8	TERMINATE	82	0	0
	9	GENERATE	1	0	0
MET	10	ADVANCE	21	1	0
	11	DISPLACE	20	0	0
	12	TRANSFER	20	0	0
OUT	13	RELEASE	18	0	0
	14	TERMINATE	18	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME AVAIL.	OWNER PEND	INTER	RETRY	DELAY
1	101	0.867	1.776	1	102	0	8

QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
01	10	9	109	19	2.595	4.927	5.967

SAVEVALUE	RETRY	VALUE
NUMBER	0	101.000

For Help, press F1      Report is Complete.      Clock

Рис. 93

### Варианты индивидуальных заданий

*Внимание!* Задание индивидуального варианта состоит из двух частей. Первая часть (первый абзац текста задания) – это задание на моделирования безотказной работы СМО с ограничениями КИ устройств (п.1 и п.2 хода выполнения работы). При реализации СМО интенсивность входного потока заявок подбирается таким образом, чтобы продемонстрировать корректность реализации в имитационной модели всех ограничений по КИ – в стандартном отчете GPSS World значения ограниченных КИ должны быть равны верхнему (нижнему) пределу ограничения.

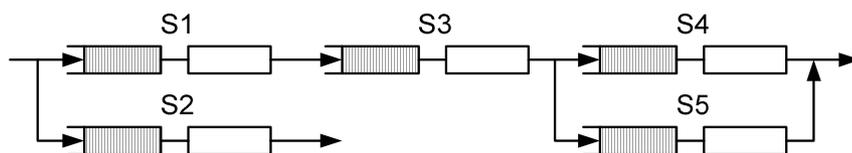
Вторая часть (второй абзац текста задания) – это задание на моделирования работы СМО с ограничениями КИ устройств в условиях действия отказов (п.3 и п.4 хода выполнения работы). Будьте внимательны при реализации! В условии определены отказы различных типов: отказ устройства обслуживания (устройство теряет работоспособность на время восстановления), ошибка в обработки (однократная ошибка обслуживания заявки, возникающая в исправном устройстве) и отказ доступа (очередь устройства теряет работоспособность и становится недоступной на время восста-

новления). Перечисленные типы отказов моделируются различными операторами *GPSS World*.

Результаты выполнения первой и второй части в отчете должны быть представлены отдельно, в соответствии с ходом выполнения работы.

Единица измерения времени во всех вариантах – секунды.

1.

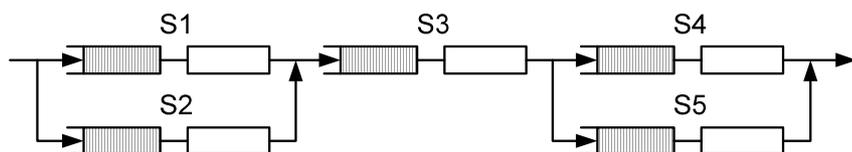


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	гамма	равн.	равн.
параметры	$2 \pm 0,5$	$\lambda = 0,8$	$\lambda = 0,5; b = 3$	$2 \pm 0,5$	$2 \pm 0,5$

КИ S1 ограничен 0,85. При превышении КИ заявки направляются на обслуживание в S2, КИ которого ограничен 0,55. При превышении КИ S2 заявки теряются. КИ S4 и S5 ограничены 0,5 и 0,6 соответственно. При превышении КИ S4 заявки направляются на обслуживание в S5, при превышении КИ S5 – теряются.

Время возникновения отказов S1 распределено по экспоненциальному закону с  $MO = 25$ . Время восстановления занимает  $2,5 \pm 0,5$ . Обслуживание, прерванное отказом, завершается в S2, там же происходит обслуживание всех заявок, поступивших в систему во время отказа и восстановления. С периодичностью  $15 \pm 2$  в S3 происходит ошибка обработки. В случае возникновения ошибки обслуживание заявки в S3 прерывается, заявка направляется в S4/S5, где, дополнительно к времени обслуживания, на эту заявку затрачивается время корректирования ошибки  $4 \pm 0,5$ .

2.

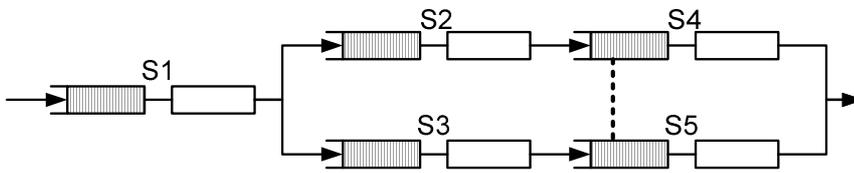


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эсп.	равн.
параметры	$10 \pm 2$	$9 \pm 1$	$12 \pm 4$	$\lambda = 0,06$	$8 \pm 1$

Первые 20 заявок, поступившие в систему, обслуживаются S1 без ограничения КИ, после чего КИ S1 должен составлять не менее 0,2 и не более 0,5. При нарушении ограничения заявки обслуживаются в S2. КИ S4 ограничен 0,6. При превышении КИ заявки направляются в S5.

В S3 с периодичностью  $55 \pm 5$  возникают отказы доступа к очереди, на восстановление доступа затрачивается  $2 \pm 0,5$ ; и ошибки обработки. Заявки, поступившие в S3 во время отказа доступа и восстановления, теряются. Заявка, обслуживание которой было прервано во время ошибки обработки, повторно с начала обслуживается в S3.

3.

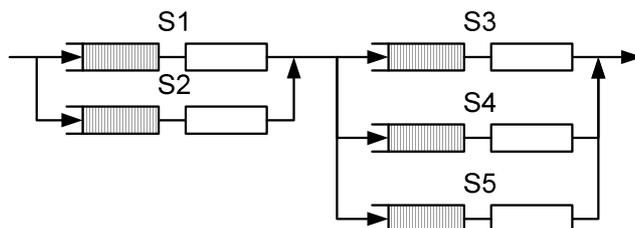


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	эсп.	эсп.
параметры	$\lambda=1$	$2 \pm 0,5$	$2 \pm 0,5$	$\lambda=0,5$	$\lambda=0,5$

При значениях КИ S1 от 0 до 0,5 заявки направляются на обслуживание в S2; при значениях от 0,5 до 0,95 – в S3, более 0,95 – теряются.

Устройства S4 и S5 работают в следующем режиме: если одно из них занято обслуживанием, другое недоступно. С периодичностью  $45 \pm 10$  в системе наступает отказ, который выражается в появлении возможности одновременного обслуживания заявок в S4 и S5. Это обслуживание прерывается через время, составляющее половину от полного времени обслуживания, причем одна из заявок (любая) – теряется, а другая возвращается в очередь устройства с признаком незавершенного обслуживания, и такие заявки после восстановления обслуживаются в первую очередь. Время восстановления распределено по экспоненциальному закону с  $MO=10$ .

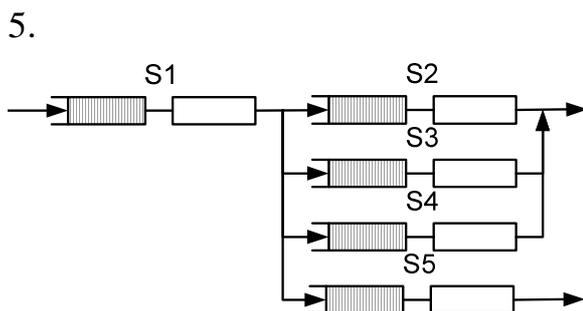
4.



характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	равн.	равн.	равн.
параметры	$5 \pm 1$	$\lambda=0,5$	$8 \pm 2$	$8 \pm 2$	$8 \pm 2$

КИ S1 ограничен 0,6. При превышении КИ заявки направляются на обслуживание в S2, КИ которого ограничен 0,75. При превышении КИ S2 заявки теряются. Распределение заявок на обслуживание в S3-S4 осуществляется равновероятно, КИ каждого из этих устройств ограничен 0,4. При превышении КИ заявки направляются на обслуживание в S5.

В S3-S4 с периодичностью  $35 \pm 2$  возникают отказы доступа к очереди, на восстановление доступа затрачивается  $5 \pm 1$ . Заявки, поступившие в S3 во время отказа доступа и восстановления, обслуживаются в S5. Время возникновения отказов S5 распределено по экспоненциальному закону с  $MO=78$ . Время восстановления занимает  $10 \pm 2$ . Обслуживание, прерванное отказом, завершается в S5 после восстановления. Заявки, поступившие во время отказа и восстановления, ожидают обслуживания в S5.



характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.
параметры	$\lambda=0,5$	$10 \pm 2$	$10 \pm 2$	$10 \pm 2$	$\lambda=0,5$

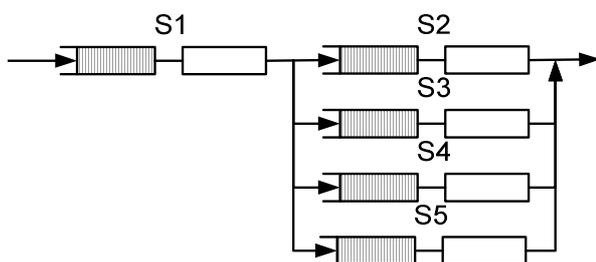
КИ S2-S5 ограничены соответственно 0,2; 0,4; 0,6 и 0,8. При превышении КИ устройства S2 заявка направляется на устройство S4; при превышении КИ устройства S3 – на устройство S5. При превышении КИ устройств S4-S5 заявки теряются.

В S1 с периодичностью  $85 \pm 10$  возникают отказы доступа к очереди, на восстановление доступа затрачивается  $5 \pm 1$ . Заявки, поступившие в S1 во время отказа доступа и восстановления, теряются. В системе S2-S4 периодически возникают ошибки обработки, время возникновения ошибок

распределено по экспоненциальному закону с  $MO=55$ , причем ошибка может произойти одновременно только в одном устройстве, номер которого определяется случайным образом. Заявки, во время обслуживания которых произошла ошибка обработки, теряются.

6.

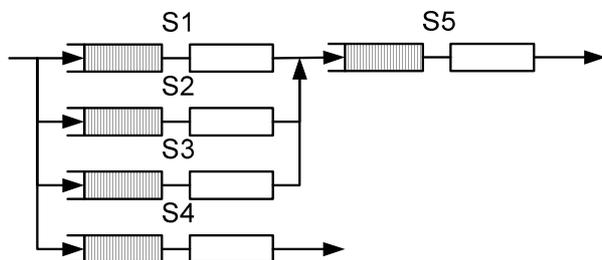
объект характеристика	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эсп.	эсп.	эсп.	эсп.
параметры	$1 \pm 0,5$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$	$\lambda=0,2$



КИ S2 ограничен 0,5; при нарушении ограничения заявка последовательно пытается пройти обслуживание в S3-S5, причем средний КИ S3 и S4 не должен превышать 0,6; а средний КИ S4 и S5 не должен превышать 0,7. При нарушении всех ограничений заявки теряются.

Время возникновения отказов S1 распределено по экспоненциальному закону с  $MO=25$ . Время восстановления занимает  $4 \pm 2$ . Обслуживание, прерванное отказом, начинается заново в S5 после восстановления. Заявки, поступившие во время отказа и восстановления, ожидают обслуживания в S5, но их число ограничено 4-мя заявками. При превышении ограничения заявки теряются.

7.

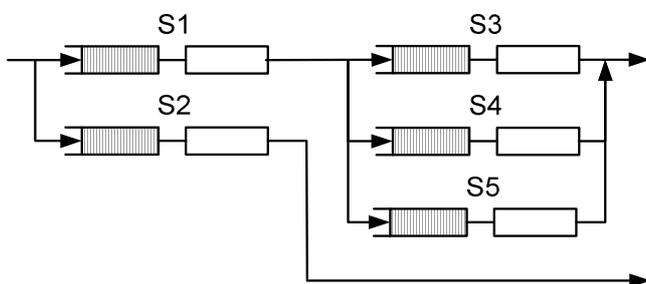


объект характеристика	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эсп.	эсп.
параметры	$5 \pm 1$	$5 \pm 1$	$5 \pm 1$	$\lambda=0,5$	$\lambda=0,1$

10% заявок входного потока направляются в S1, 30% – в S2, 60% – в S3, причем КИ устройств соответственно ограничены 0,1; 0,3; 0,6. При превышении заявки направляются в S4, КИ которого не должен превышать 0,8; иначе заявки теряются.

Время возникновения отказов S1-S3 распределено по экспоненциальному закону с MO = 65, причем отказ одновременно может наступить только в одном устройстве. Время восстановления занимает  $5 \pm 2$ . Обслуживание, прерванное отказом, завершается в S5, причем время завершения суммируется с временем собственного обслуживания в S5.

8.

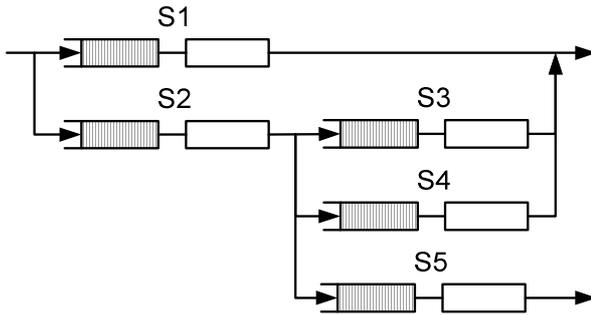


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda = 0,5$	$\lambda = 1$	$12 \pm 1$	$10 \pm 1$	$2 \pm 1$

В исходном варианте схемы S2 не используется. В процессе обслуживания заявки между S3 и S4 распределяются равномерно, причем суммарный КИ S3 и S4 не должен превышать 1. При нарушении ограничения заявки направляются в S5, КИ которого не может превышать 0,85; при нарушении заявки теряются.

Входная очередь S1 ограничена 3-мя заявками. По заполнению очереди, а также во время отказов доступа заявки направляются на обслуживание в S2. Отказы доступа происходят с периодичностью  $60 \pm 5$ , на восстановление доступа тратится время, распределенное по экспоненциальному закону с MO = 4. В S2 периодически возникают ошибки обслуживания, время возникновения ошибок распределено по экспоненциальному закону с MO = 55. Обслуживание заявок, во время которого произошла ошибка, завершается без прерывания, но затем заявка возвращается на повторное обслуживание в S2.

9.

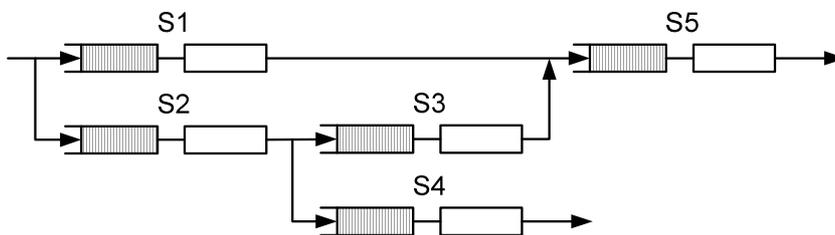


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.
параметры	$\lambda = 0,125$	$0,5 \pm 0,1$	$6 \pm 2$	$4 \pm 2$	$\lambda = 0,2$

КИ S1 ограничен 0,125; при нарушении ограничения заявки направляются в S2, КИ которого ограничен 0,9; при нарушении – заявки теряются. Поток заявок между S3 и S4 распределяется равномерно, КИ каждого из этих устройств не должен превышать 0,7. При нарушении ограничения заявки направляются в S5, КИ которого не ограничен.

Время возникновения отказов S2 распределено по экспоненциальному закону с  $MO = 15$ . Время восстановления занимает  $4 \pm 2$ . Обслуживание, прерванное отказом, завершается в S5. В S5 периодически возникают ошибки обслуживания, время возникновения ошибок распределено по экспоненциальному закону с  $MO = 30$ , причем ошибки возможны только для собственного обслуживания в S5, завершение обслуживания после отказа в S2 всегда происходит безошибочно.

10.



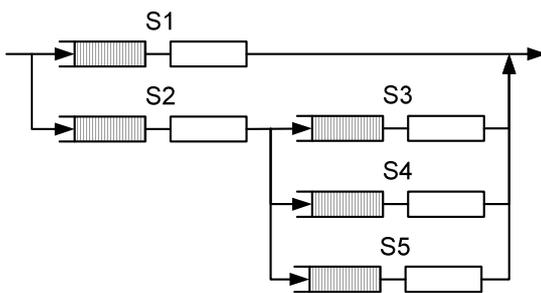
характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	гамма
параметры	$5 \pm 2$	$2 \pm 1$	$7 \pm 5$	$2 \pm 1$	$\lambda=1,2; b=3$

Поток заявок между S1 и S2 распределяется равномерно, но, если КИ S1 превышает 0,5, время обслуживания в S2 уменьшается в два раза. КИ S3 ограничен 0,8, при нарушении ограничения заявки направляются в S4, КИ которого не ограничен.

В S3 периодически возникают ошибки обслуживания, время возникновения ошибок  $25 \pm 5$ . Если заявка, во время обслуживания которой произошла ошибка, в S2 была обслужена с нормальным временем обслуживания, то ее обслуживание прерывается и завершается в S5. Если заявка, во время обслуживания которой произошла ошибка, в S2 была обслужена с уменьшенным временем, то она после прерывания теряется.

11.

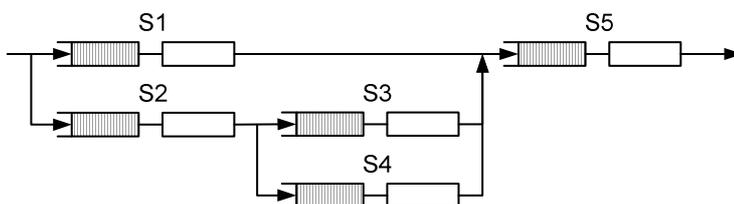
характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	эсп.	эсп.
параметры	$3 \pm 2$	$3 \pm 1$	$\lambda = 0,25$	$\lambda = 0,5$	$\lambda = 0,75$



В S1 обслуживается 10% заявок, остальные направляются в S2, КИ которого ограничен 0,5. При превышении ограничения заявки сразу, минуя S2, направляются в S3-S5, КИ которых ограничены, соответственно, 0,75; 0,5 и 0,25. При нарушении ограничения заявки теряются.

Время возникновения отказов S3 и S4 распределено по экспоненциальному закону с  $MO = 30$ , причем отказ одновременно может наступить только в одном устройстве: с вероятностью 55% в S3, с остаточной – в S4. Время восстановления занимает  $3 \pm 1$ . Обслуживание, прерванное отказом, завершается после восстановления. Заявки, поступившие на обслуживание в S3 (S4) во время отказа и восстановления, теряются.

12.

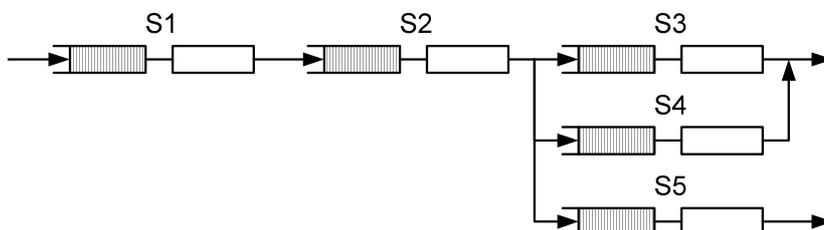


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	равн.	эсп.
параметры	$4 \pm 1$	$2 \pm 0,5$	$\lambda = 0,125$	$2 \pm 0,5$	$\lambda = 0,125$

Заявки входного потока обслуживаются в S1 и, далее, в S5. КИ S1 ограничен 0,6. При превышения ограничения заявки входного потока направляются в S2. КИ S3 ограничен 0,45 и, кроме того, средний КИ S3 и S4 не должен превышать 0,6. При нарушении ограничений заявки теряются.

В S1 и S2, а также в S3 и S4, с периодичность  $45 \pm 5$  наступают отказы, причем отказ может наступить либо в одном ОУ каждой пары, либо в двух устройствах одновременно. Время восстановления распределено по экспоненциальному закону с  $MO = 2$ . Если при отказе в паре есть работоспособное ОУ, оно завершает прерванное обслуживание и продолжает обслуживание заявок, поступивших во время отказа и восстановления. Если отказали оба устройства пары, то прерванные заявки ожидают восстановления и после него завершают начатое обслуживание в своих устройствах, а заявки, поступивших во время отказа и восстановления, теряются.

13.



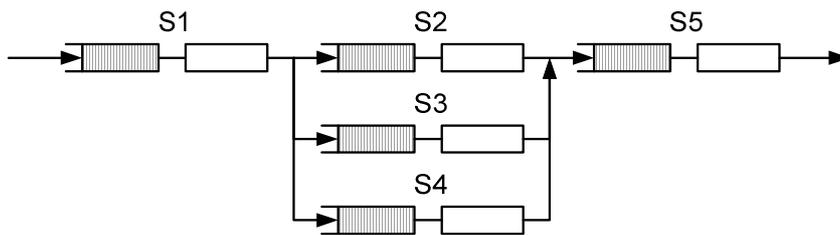
характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda = 0,45$	$\lambda = 0,5$	$1,2 \pm 0,5$	$2 \pm 0,5$	$2 \pm 0,2$

Первые 10 заявок входного потока обслуживаются в S3-S4 равномерно, остальные – направляются на устройство с меньшим текущим КИ, причем КИ обоих устройств ограничено 0,4. При превышении КИ заявки направляются в S5, КИ которого ограничено 0,2. При превышении КИ заявки теряются.

Входная очередь S1 ограничена 5-ю заявками. По заполнению очереди, а также во время отказов доступа заявки сразу направляются на обслуживание в S2. Время возникновения отказов распределено по экспоненциальному закону с  $MO = 2$ .

ненциальному закону с  $MO=25$ , на восстановление доступа тратится  $4\pm 1$ . В  $S2$  периодически возникают отказы, время возникновения отказов распределено по экспоненциальному закону с  $MO=45$ , на восстановление затрачивается время, распределенное по экспоненциальному закону с  $MO=3$ . Обслуживание заявок, во время которого произошел отказ, завершается без прерывания, но затем заявка сразу покидает систему. Заявки, поступившие во время отказа и восстановления, ожидают обслуживания в  $S2$ .

14.

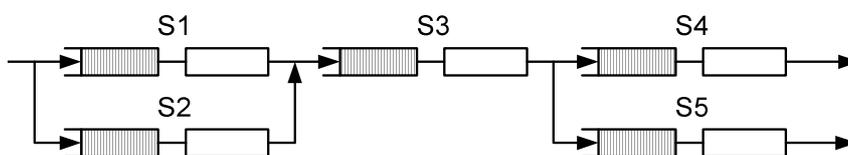


объект / характеристика	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.
параметры	$\lambda=0,5$	$2\pm 0,5$	$2\pm 0,5$	$2\pm 0,5$	$\lambda=0,125$

КИ  $S1$  не ограничен, но если он составляет от 0 до 0,3; заявки на дальнейшее обслуживание направляются в  $S2$ ; от 0,3 до 0,6 – в  $S3$ ; более 0,6 – в  $S4$ . Коэффициент использования  $S5$  ограничен 0,85, при превышении заявки теряются.

В  $S2$ - $S4$  периодически возникают отказы, причем отказ может наступить в одном устройстве, в двух одновременно или в трех одновременно. Если одно или два из устройств  $S2$ - $S4$  работоспособны, они завершают прерванное отказом обслуживание и обслуживают заявки, поступившие во время отказа и восстановления. Если отказывают все три устройства одновременно, прерванное обслуживание завершается в  $S5$ , причем время завершения прибавляется к времени собственного обслуживания в  $S5$ , а заявки, поступивших во время отказа и восстановления, теряются.

15.



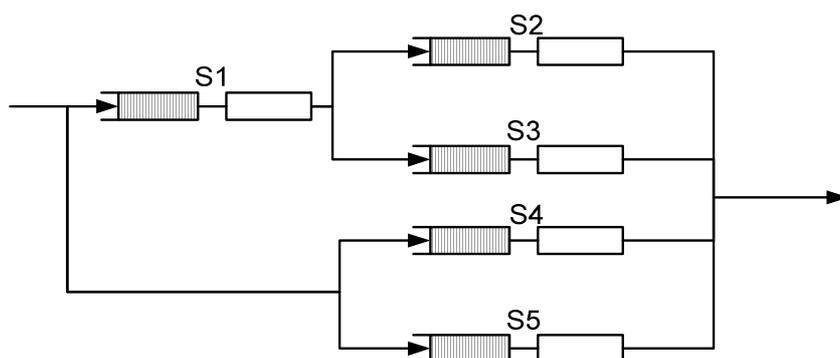
характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	равн.	равн.	равн.
параметры	$\lambda=0,2$	$\lambda=0,2$	$1\pm 0,5$	$2\pm 2$	$2\pm 1,5$

КИ S1 ограничен 0,4; при нарушении ограничения заявки направляются в S2, КИ которого ограничен 0,6; при нарушении – заявки теряются. Поток заявок между S4 и S5 распределяется равномерно, суммарный КИ этих устройств не должен превышать 1,2. При нарушении ограничения заявки задерживаются во входных очередях устройств.

Входная очередь S1 ограничена 5-ю заявками. По заполнению очереди, а также во время отказов доступа заявки направляются на обслуживание в S2. Время возникновения отказов распределено по экспоненциальному закону с  $MO = 15$ , на восстановление доступа тратится  $1\pm 0,2$ . В S3 периодически возникают ошибки обслуживания, время возникновения ошибок  $20\pm 2$ . Ошибочное обслуживание прерывается, а заявка возвращается на повторное обслуживание во входной поток.

16.

характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	эсп.	равн.	эсп.
параметры	$\lambda=4,5$	$1\pm 0,5$	$\lambda=2$	$4\pm 2$	$\lambda=0,2$

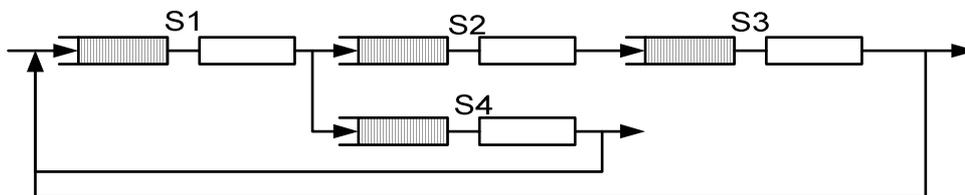


Если КИ S1 не превышает 0,5, заявки далее обслуживаются в S2-S3, если составляет от 0,5 до 0,8 – заявки, минуя S1, сразу направляются в S4-S5, при превышении 0,8 – теряются. Коэффициенты использования S2 и S4 ограничены 0,3, при превышении заявки обслуживаются в S3 и S5 соответственно, КИ которых не ограничены.

Время возникновения отказов S1 распределено по экспоненциальному закону с  $MO = 35$ . Время восстановления занимает  $2\pm 1$ . Обслуживание,

имевшее место в момент отказа, не прерывается и завершается в S1, но такая заявка в дальнейшем вызывает ошибку обслуживания в любом другом устройстве СМО и теряется. Заявки, поступившие в систему во время отказа и восстановления, направляются на обслуживание в S4-S5.

17.

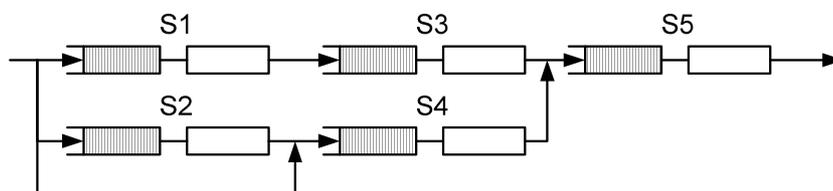


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4
закон распределения	равн.	эсп.	эсп.	эсп.
параметры	$1 \pm 0,5$	$\lambda = 0,2$	$\lambda = 0,125$	$\lambda = 0,5$

КИ S2 ограничен 0,5; при нарушении ограничения заявки обслуживаются в S4, КИ которого ограничен 0,8; при превышении этого ограничения заявки теряются.

С периодичностью  $20 \pm 4$  в S3 и S4 возникают ошибки обслуживания. Ошибочное обслуживание прерывается, а заявки возвращаются во входной поток. С периодичностью  $60 \pm 2$  в системе возникает частичный отказ, который на время, распределенное по экспоненциальному закону с  $MO = 10$ , уменьшает ограничение по КИ в два раза, а время обслуживания увеличивает в два раза. Этот же отказ повышает вероятность ошибочного обслуживания в S3 и S4 до 50% безотносительно к периодичности ошибочного обслуживания во время нормальной работоспособности.

18.

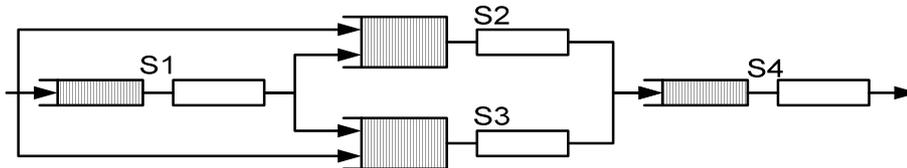


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	равн.	эсп.
параметры	$8 \pm 1$	$4 \pm 1$	$4 \pm 1$	$2 \pm 0,5$	$\lambda = 0,1$

КИ S1 ограничен 0,3. При превышении КИ заявки направляются на обслуживание в S2, КИ которого ограничен 0,65. При превышении КИ S2 заявки сразу направляются на обслуживание в S4 (КИ≤0,85). При превышении КИ S4 заявки теряются.

Время возникновения отказов S1 распределено по экспоненциальному закону с MO=65. Время восстановления занимает 8±2. Обслуживание, прерванное отказом, завершается в S2, там же происходит обслуживание всех заявок, поступивших в систему во время отказа и восстановления. В S5 с периодичностью 50±5 возникают ошибки обработки. Если заявка, обслуживание которой было прервано, попала в S5 после обслуживания в S3, она теряется, если после обслуживания в S4 – повторно сначала обслуживается в S5.

19.

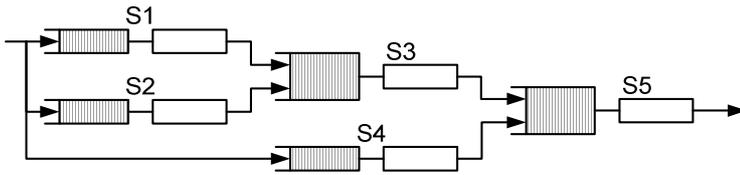


объект / характеристика	устройство S1	устройство S2	устройство S3	устройство S4
закон распределения	эсп.	эсп.	эсп.	равн.
параметры	$\lambda=1$	$\lambda=0,25$	$\lambda=0,25$	$2\pm0,5$

При значениях КИ S1 от 0 до 0,3 заявки далее направляются на обслуживание в S2; при значениях от 0,3 до 0,6 – в S3, более 0,9 – теряются.

Время возникновения отказов S1 распределено по экспоненциальному закону с MO = 60. Время восстановления занимает 20±2. Обслуживание, имевшее место в момент отказа, не прерывается и завершается в S1; заявки, поступившие в систему во время отказа и восстановления, распределяются на обслуживание в S2 и S3, причем на время отказа и восстановления КИ S2 ограничен 0,6. При нарушении ограничения заявки обслуживаются в S3, причем в этом случае среднее время обслуживания в S3 уменьшается в три раза. В S4 периодически возникают ошибки обслуживания, время возникновения ошибок 25±5. Если заявка, во время обслуживания которой произошла ошибка, была обслужена в S2 или в S3 с нормальным временем обслуживания, то ее обслуживание не прерывается. Если заявка, во время обслуживания которой произошла ошибка, была обслужена в S3 с уменьшенным временем обслуживания, то она после прерывания теряется.

20.

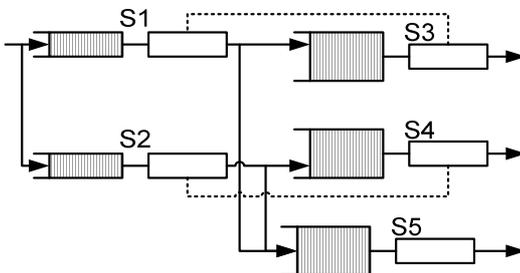


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	равн.	равн.	эсп.
параметры	$\lambda=1$	$\lambda=0,8$	$12\pm 2$	$2\pm 0,5$	$\lambda=0,5$

КИ S1 ограничен 0,4, при превышения заявки направляются в S2, КИ которого ограничен 0,6. КИ S3 ограничен 0,5, при превышении заявки из входного потока направляются на обслуживание в S4.

В S1-S2 периодически возникают отказы, причем отказ может наступить в одном устройстве, или в двух одновременно. Если одно из устройств S1-S2 работоспособны, они начинают сначала прерванное отказом обслуживание и обслуживают заявки, поступившие во время отказа и восстановления. Если отказывают два устройства одновременно, начатое до отказа обслуживание не прерывается и завершается, а заявки, поступивших во время отказа и восстановления, теряются. Отказы возникают с периодичностью  $40\pm 5$ , время восстановления составляет  $5\pm 1$ . Очередь S4 ограничена 3-мя заявками, и в ней периодически возникают ошибки доступа. По заполнению очереди, а также во время отказов доступа заявки теряются. Отказы доступа происходят с периодичностью  $80\pm 10$ , на восстановление доступа тратится время, распределенное по экспоненциальному закону с  $MO = 2$ .

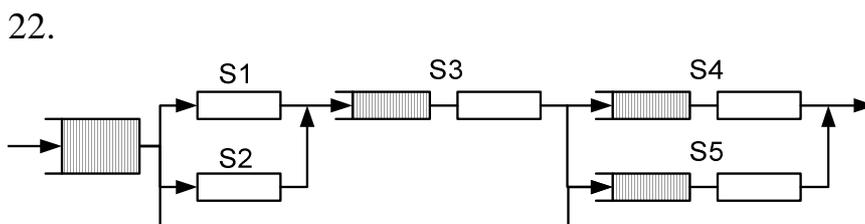
21.



характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	эсп.	эсп.
параметры	$1,6\pm 0,2$	$1,6\pm 0,2$	$\lambda=1$	$\lambda=1$	$\lambda=0,5$

Заявки между двумя основными каналами обслуживания распределяются равномерно. КИ S3-S4 ограничены 0,6. При превышении КИ среднее время обслуживания в первичном устройстве (S1-S2) увеличивается в два раза, при снижении КИ до значения менее 0,4 – возвращается к исходному значению. Если включение/отключение такого режима происходит три раза, в дальнейшем при превышении КИ заявки отправляются на обслуживание в S5, КИ которого не ограничен.

В S1-S2 периодически возникают ошибки обслуживания, время возникновения ошибок распределено по экспоненциальному закону с  $MO=60$ . Обслуживание заявок, во время которого произошла ошибка, завершается без прерывания, но в дальнейшем, если эта заявка обслуживается в S3-S4, то время ее обслуживания увеличивается на величину остаточного обслуживания на момент возникновения ошибки, а если в S5 – то на обслуживание затрачивается обычное время, но после завершения заявка теряется.

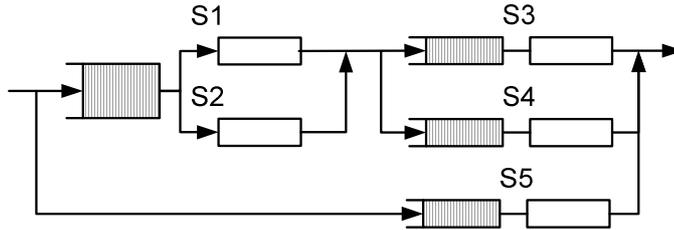


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	эсп.	эсп.	эсп.
параметры	$1 \pm 0,5$	$2 \pm 0,5$	$\lambda=1$	$\lambda=0,8$	$\lambda=1,5$

КИ S1 ограничен 0,2; при нарушении ограничения заявки направляются в S2, КИ которого ограничен 0,8; при нарушении – сразу в S4-S5, где распределяются на обслуживание равномерно. Средний КИ S4-S5 не должен превышать 0,7; при нарушении заявки теряются.

Входная очередь ограничена 5-ю заявками. По заполнению очереди, а также во время отказов доступа заявки входного потока теряются. Время возникновения отказов доступа распределено по экспоненциальному закону с  $MO = 20$ , на восстановление доступа тратится  $2 \pm 1$ . Заявки, которые имелись в очереди на момент возникновения отказа доступа, вне зависимости от прочих условий направляются в S4-S5, где в 12% случаев вызывают ошибку обслуживания и теряются.

23.

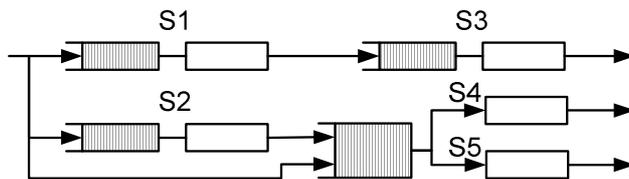


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эксп.	равн.	равн.	равн.	эксп.
параметры	$\lambda=0,5$	$2\pm 0,5$	$2,5\pm 0,5$	$2,5\pm 0,5$	$\lambda=0,1$

КИ S1 (S3) ограничен 0,4; при нарушении ограничения заявки направляются в S2 (S4), КИ которых ограничен 0,6; при нарушении ограничения заявки теряются. В исходном варианте S5 не используется.

В СМО возможны следующие виды отказов: отказ доступа к входной очереди – периодичность  $100\pm 1$ , время восстановления  $2\pm 1$ ; отказ одного из устройств S1-S2 – периодичность  $60\pm 2$ , время восстановления  $2\pm 1$ ; отказ одного из устройств S3-S4 – периодичность  $82\pm 2$ , время восстановления  $2\pm 1$ . В момент наступления любого из указанных отказов обслуживание заявок во всех устройствах СМО прерывается, заявки теряются; оставшиеся в системе заявки продолжают обслуживание (если возможно) или ожидают восстановления; а заявки входного потока, поступившие во время отказа и восстановления, отправляются на обслуживание в S5.

24.

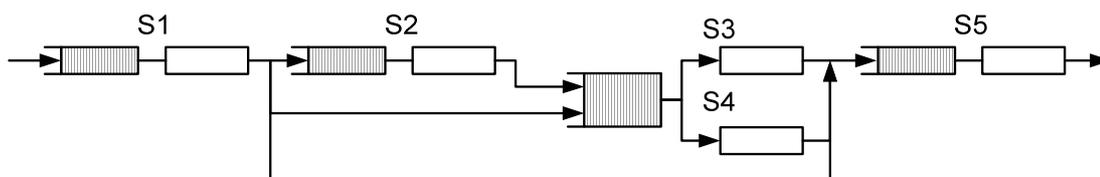


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	эксп.	гамма	равн.	равн.
параметры	$2\pm 0,5$	$\lambda=0,8$	$\lambda=0,5; b=3$	$2\pm 0,5$	$2\pm 0,5$

КИ S1 ограничен 0,3; при нарушении ограничения заявки направляются в S2, КИ которого ограничен 0,6; при нарушении – сразу в S4-S5, где распределяются на обслуживание на устройство с меньшим текущим КИ.

Общая очередь S4-S5 ограничена 5-ю заявками и с периодичностью  $40 \pm 5$  в ней случаются отказы доступа. Время восстановления доступа распределено по экспоненциальному закону с  $MO = 6$ . На время отказа восстановления доступа все заявки входного потока без учета ограничения по КИ обслуживаются в канале S1-S3. Время возникновения отказов S3 распределено по экспоненциальному закону с  $MO = 60$ . Время восстановления занимает  $8 \pm 2$ . Если время отказа-восстановления совпадает с отказом доступа к очереди S4-S5, то заявка с прерванным обслуживанием и поступившие на вход S3 заявки теряются. Если время отказа-восстановления не совпадает с отказом доступа к очереди S4-S5, то все заявки входного потока направляются на обслуживание в S2/S4-S5, а заявки в канале S1-S3, в том числе и заявка с прерванным обслуживанием, ожидают восстановления и продолжают обслуживание.

25.

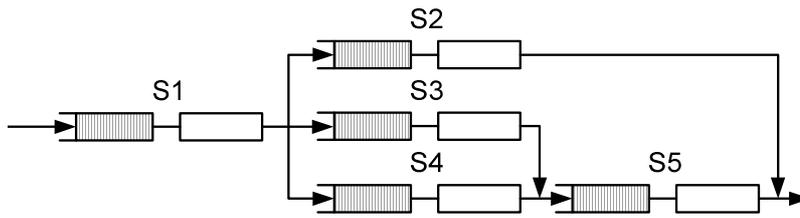


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	гамма	равн.	равн.	равн.	равн.
параметры	$\lambda=1; b=4$	$5 \pm 0,5$	$5 \pm 0,5$	$5 \pm 0,5$	$2 \pm 0,5$

Если КИ S1 в процессе обработки заявок входного потока составляет менее 0,5, то заявки на дальнейшее обслуживание направляются в S2, если от 0,5 до 0,8 – в общую очередь S3-S4, и если более 0,8 – в S5. КИ S3 ограничен 0,4; при превышении заявки направляются в S4.

В S2 периодически возникают ошибки обслуживания, время возникновения ошибок  $85 \pm 15$ . Ошибочное обслуживание прерывается, заявка на дальнейшее обслуживание направляется в S4, где обслуживается в порядке очереди, но на ее обслуживание затрачивается  $8 \pm 2$ . Время возникновения отказов S1 распределено по экспоненциальному закону с  $MO = 6.5$  Время восстановления занимает  $2 \pm 0.5$ . Обслуживание, прерванное отказом, завершается после восстановления; заявки, поступившие в систему за время отказа и восстановления, теряются.

26.

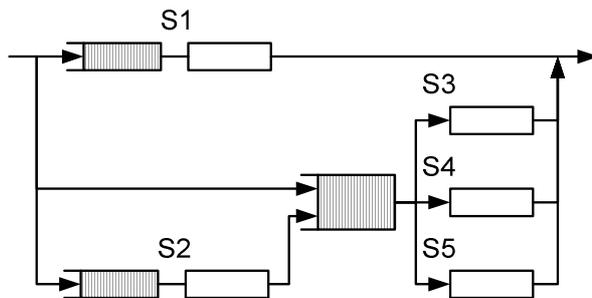


характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	равн.	равн.	равн.	эсп.
параметры	$\lambda=0,5$	$3\pm 1$	$3.2\pm 1,5$	$4\pm 1,5$	$\lambda=0,45$

КИ S2 ограничен 0,2; при превышении заявки направляются на обслуживание в S3, КИ которого ограничен 0,6; при превышении заявки направляются на обслуживание в S4, КИ которого ограничен 0,8; при превышении заявки теряются.

В S2 периодически возникают отказы, время возникновения  $85\pm 15$ с, время восстановления –  $5\pm 1$ . Если на момент возникновения отказа S2 занято обслуживанием – обслуживание прерывается, заявка направляется в очередь S5, где вызывает отказ доступа. Отказ продолжается до поступления очередной заявки на обслуживание из S4.

27.



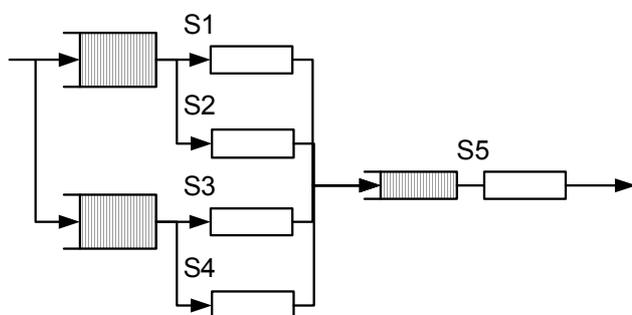
характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	эсп.	эсп.	эсп.	эсп.	эсп.
параметры	$\lambda=0,125$	$\lambda=0,5$	$\lambda=0,35$	$\lambda=0,5$	$\lambda=0,4$

Заявки входного потока с вероятностью 12% обслуживаются в S1, с вероятностью 35% сразу поступают в общую очередь многоканальной системы, с остаточной вероятностью – обслуживаются в S2. В многоканальной система заявка направляется на обслуживание на устройство с мень-

шим КИ с меньшим номером, при этом КИ каждого устройства ограничен 0,45. При превышении ограничения заявки теряются.

В СМО периодически, через  $55 \pm 5$ , начинаются периоды ошибочного обслуживания, которые в среднем длятся  $10 \pm 2$ . Во время таких периодов заявки входного потока, которые должны обслуживаться в S1, равновероятно направляются по двум другим направлениям. Время обслуживания таких заявок в многоканальной системе возрастает на время, равное обслуживанию в S1. В S2 периодически возникают отказы, время возникновения  $85 \pm 5$ , время восстановления распределено по экспоненциальному закону с  $MO = 3$ . Если на момент возникновения отказа S2 занято обслуживанием оно завершается без прерывания. Заявки, поступившие на обслуживание во время отказа и восстановления, ожидают восстановления.

28.



характеристика \ объект	устройство S1	устройство S2	устройство S3	устройство S4	устройство S5
закон распределения	равн.	равн.	равн.	эсп.	эсп.
параметры	$7 \pm 1$	$5 \pm 1$	$5 \pm 1$	$\lambda = 0,2$	$\lambda = 1$

10% заявок входного потока направляются в S1, 30% – в S2, 30% – в S3, причем КИ устройств S2 и S3 соответственно ограничены 0,3 и 0,4. Остальные заявки, а также при превышении КИ, направляются в S4, КИ которого не должен превышать 0,85; иначе заявки теряются.

Время возникновения отказов S3 и S4 распределено по экспоненциальному закону с  $MO = 100$ , причем отказ одновременно может наступить только в одном устройстве: с вероятностью 75% в S3, с остаточной – в S4. Время восстановления занимает  $12 \pm 1$ . Обслуживание, начатое к моменту отказа, завершается без прерывания. Заявки, поступившие на обслуживание в S3 (S4) во время отказа и восстановления, ожидают восстановления. Время возникновения отказов S1 и S2 распределено по экспоненциальному закону с  $MO = 75$ , причем отказ наступает одновременно в S1 и S2. Прерванное обслуживание завершается в S3; заявки, поступившие на обслуживание в S1 и S2 во время отказа и восстановления, обслуживаются в S4.

## **Лабораторная работа №7. Моделирование систем массового обслуживания сетями Петри**

*Цель:* Моделирование вычислительных и технологических систем сетями Петри.

*Программное обеспечение моделирования:* программа моделирования и анализа сетей Петри Ingproject.exe.

*Примеры выполнения работы:* пример1.pne, пример2.pne, пример3.pne

### **Ход выполнения работы**

Основные сведения о моделировании систем сетями Петри – см. *раздел 3.1.2. конспекта лекций.*

По варианту индивидуального задания разрабатывается сеть Петри. В отчете представляется:

- начальная (нулевая) разметка сети;
- описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны;
- описание всех переходов с точки зрения связанных с ними событий и времен наступления этих событий;
- описание всех значимых ситуаций функционирования заданной СМО и соответствующих им маркировок;
- схемы маркировок сети Петри (пошаговый прогон модели в *Ingproject.exe*) и их описание для всех значимых ситуаций функционирования СМО.

### **Практические рекомендации по выполнению работы и использованию программного обеспечения моделирования**

*Программа ingproject.exe*

Для задания графического представления модели сети Петри в программе используются кнопки графической панели инструментов, описание которых представлено в таблице 29.

Для задания свойств вершин и переходов используется кнопка «Свойства» . После нажатия кнопки выбор объекта для задания свойств осуществляется щелчком мыши в рабочем поле. Окно ввода свойств вершины имеет вид, представленный на рис. 94. Для вершины задаются:

- имя и начальное количество меток (для нулевой разметки сети);
- вес и стиль отображения исходящих дуг при выборе их из списка;

– при необходимости в окне редактирования свойств можно задать удаление отдельных дуг вершины.

*Внимание!* Изменение свойств выходных дуг необходимо подтверждать нажатием кнопки «Задать». Нажатие кнопки «ОК» в конце редактирования подтверждает принятие заданных изменений.

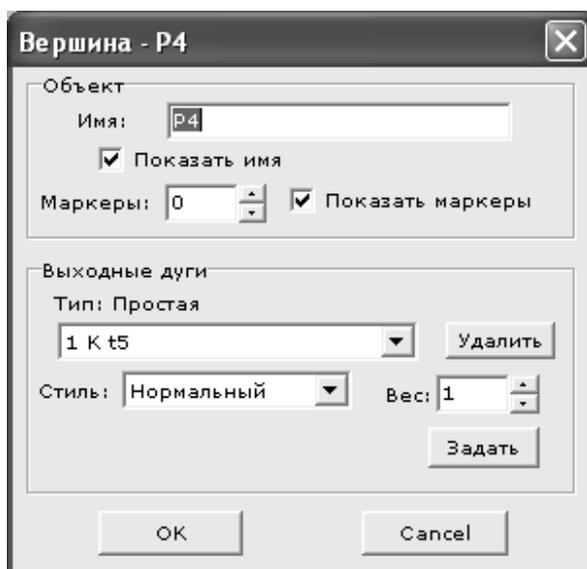


Рис. 94

Таблица 29

Обозначение	Описание
	<p>Кнопка для отображения позиции (вершины) – в рабочем поле размещение позиции задается щелчком мыши. При нажатой кнопке в панели инструментов левой клавишей мыши можно добавлять метки, правой – удалять. При наличии в вершине более двух меток их количество отображается числом.</p>
	<p>Кнопка для отображения перехода. При нажатой кнопке переключение горизонтальной/вертикальной ориентации перехода задается щелчком левой клавиши мыши.</p>
	<p>Кнопка задания дуги простого перехода. При нажатой кнопке в рабочем поле дуга единичного веса задается соединением вершины с переходом или перехода с вершиной при нажатой левой клавиши мыши. Соединение дугой двух позиций или двух переходов между собой невозможно.</p> <p>Если необходимо задать дугу с весом более 1 для задания перехода типа n/m, то можно прорисовать одинарную дугу столько раз, каков требуемый вес, или изменить вес дуги в свойствах связанных с ней вершин или переходов.</p>

Обозначение	Описание
	Кнопка задания дуги ингибиторного перехода. Разрешенное направление такой дуги – только от вершины к переходу.
	Кнопка перемещения – используется для перемещения элементов сети Петри – вершин и переходов.
	Кнопка удаления элементов сети. При нажатой кнопке объект – вершина или переход – удаляется щелчком левой клавиши мыши вместе со всеми входными и выходными дугами. Удаление отдельных возможно дуг при редактировании свойств элементов сети.
	Кнопка задания свойств вершин и переходов. При нажатой кнопке щелчок левой клавиши мыши по вершине или переходу открывает окно редактирования свойств.
	Запуск моделирования. Перед запуском необходимо задать начальную (нулевую) разметку сети.
	Остановка моделирования.
	Запуск моделирования в пошаговом режиме.

Окно ввода свойств перехода имеет вид, представленный на рис. 95.

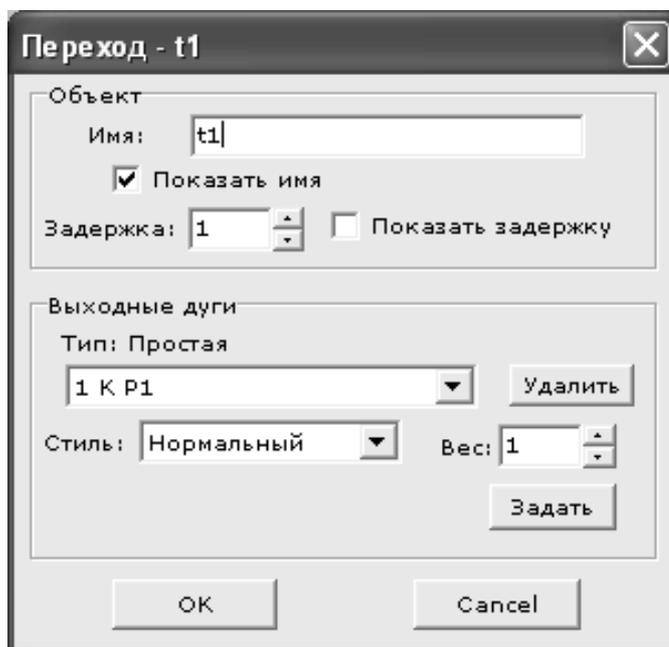


Рис. 95

Процедура заданий свойств и значений полей аналогична за исключением задержки: этот параметр задает время срабатывания перехода после активизации в секундах.

**Пример 1. Сеть Петри для моделирования процессов отказов и устранения неисправностей**

Пусть в некоторой системе одновременно эксплуатируются три одинаковых блока. С некоторой периодичностью блоки выходят из строя. Для устранения отказа в системе имеется запасной блок. Замена отказавшего блока на запасной занимает некоторое время, после чего отказавший блок направляется на восстановление и, после восстановления, становится запасным. Сеть Петри, моделирующая процесс функционирования этой системы в условиях возникновения отказов, использования запасного блока для замены и последующим устранением неисправностей (восстановления) отказавшего блока, имеет вид, приведенный на рис. 96.

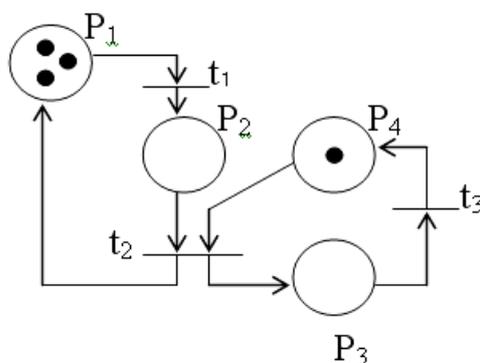


Рис. 96

В таблице 30 представлено описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны, т. е. выполняются при наличии в вершине метки-маркера.

Таблица 30

Вершина	Описание
$P_1$	Наличие в системе исправных рабочих блоков (количество меток соответствует количеству блоков)
$P_2$	Наличие в системе отказавших рабочих блоков (количество меток соответствует количеству блоков)
$P_4$	Наличие в системе исправного, готового к использованию для замены запасного блока
$P_3$	Наличие в системе неисправного, требующего в восстановлении блока

Описание всех переходов с точки зрения связанных с ними событий и времен их наступления представлено в таблице 31.

Начальная маркировка, приведенная на рис. 96, соответствует наличию в системе трех рабочих блоков и одного запасного. Переходам соответствуют временные задержки: 1 – время между отказами; 2 – время замены; 3 – время восстановления.

Таблица 31

Переход	Описание
$t_1$	Наступление отказа (время срабатывания перехода соответствует времени наступления отказа)
$t_2$	Выполнение замены неисправного блока на запасной или резервный (время срабатывания перехода соответствует времени замены)
$t_3$	Выполнение восстановления неисправного блока (время срабатывания перехода соответствует времени восстановления).

В реальной системе времена наступления событий случайны, и распределены по некоторому закону. В *Ingproject* возможно только детерминированное моделирование, т. е. величина времени задержки перехода является постоянной. Поэтому для исследования динамики работы системы необходимо назначать различные времена задержек, или реализовать равновероятное распределение времен путем использования нескольких переходов. Например, если время некоторого процесса составляет  $3 \pm 1$ , то это моделируется тремя равновероятными переходами с временем срабатывания 2; 3 и 4 соответственно.

Все возможные по условиям функционирования системы маркировки представлены в таблице 32.

Таблица 32

Маркировка	Количество меток в вершинах				Описание
	$P_1$	$P_2$	$P_3$	$P_4$	
№1	3	0	0	1	Начальная маркировка, соответствующая наличию в системе трех рабочих блоков и одного запасного.
№2	2	1	0	1	Наличие в системе двух рабочих блоков, одного отказавшего и одного запасного.
№3	1	2	0	1	Наличие в системе одного рабочего блока, двух отказавших и одного запасного (время замены больше чем суммарное время отказа двух блоков).
№4	0	3	0	1	Наличие в системе трех отказавших и одного запасного блока (время замены больше чем суммарное время трех отказов).

Маркировка	Количество меток в вершинах				Описание
	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	
№5	1	2	1	0	Наличие в системе одного рабочего блока, двух отказавших и одного, находящегося на восстановлении.
№6	0	3	1	0	Наличие в системе трех отказавших и одного, находящегося на восстановлении.
№7	2	1	1	0	Наличие в системе двух рабочих блоков, одного отказавшего и одного, находящегося на восстановлении.
№8	3	0	1	0	Наличие в системе трех рабочих блоков и одного, находящегося на восстановлении.

Исследование динамики работы для различных временных задержек переходов реализованной в *Ingproject.exe* модели (см. файл *пример1.pne*) приведено на рис. 97-102 в виде маркировок некоторых значимых состояний, полученных в режиме пошагового прогона. На всех рисунках активные переходы выделены цветом. Реализация, приведенная на рис. 97 соответствует начальной маркировке (№1 в таблице 32).

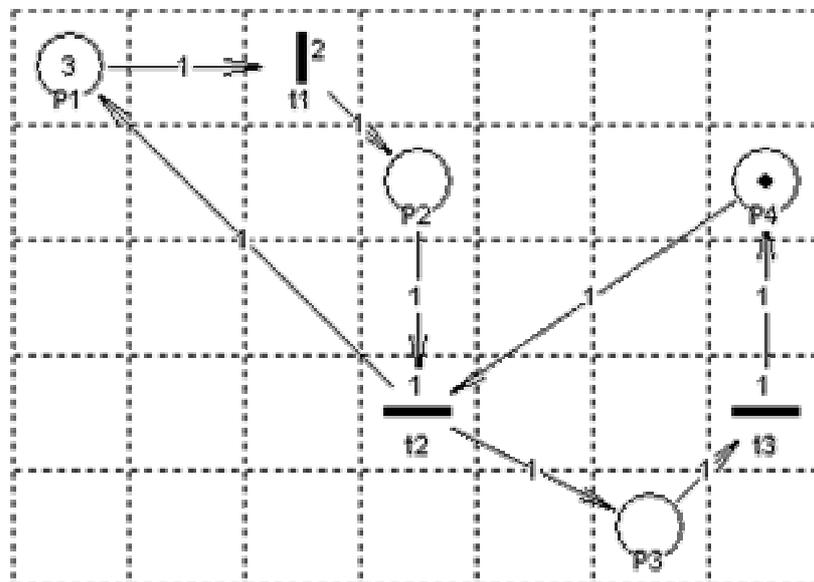


Рис. 97

На рис. 98 отображен запуск моделирования. Переход *t1* активен – для его активизации в *P1* должна быть хотя бы одна метка, но еще не сработал, поскольку еще не истекло время задержки *t1*, соответствующее наступлению отказа.

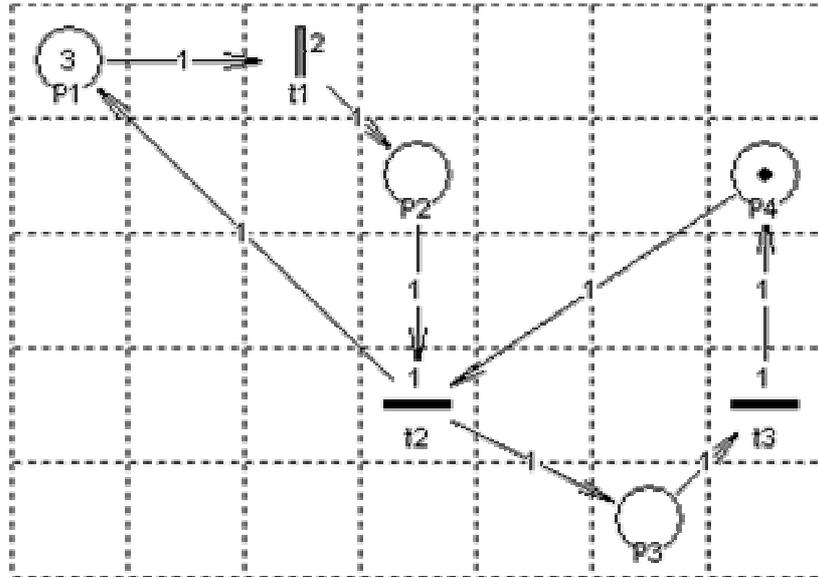


Рис. 98

На рис. 99 маркировка №2 (см. таблицу 32), соответствующая наличию одного отказавшего блока (метка в вершине P2), двух рабочих (две метки в вершине P1) и готового к использованию запасного (метка в вершине P4). Активны переходы t1 и t2.

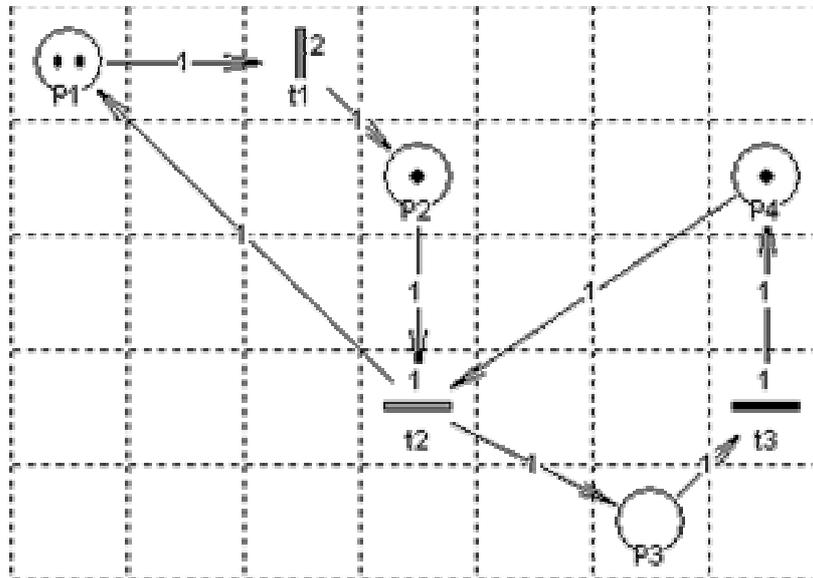


Рис. 99

На рис. 100 приведена маркировка №8 (см. таблицу 32), соответствующая наличию в системе трех рабочих блоков (три метки в вершине P1) и одного, находящегося на восстановлении (метка в вершине P3). Переход t1 активен – еще не истекло время до наступления очередного отказа. Переход t3 активен – еще не истекло время восстановления.

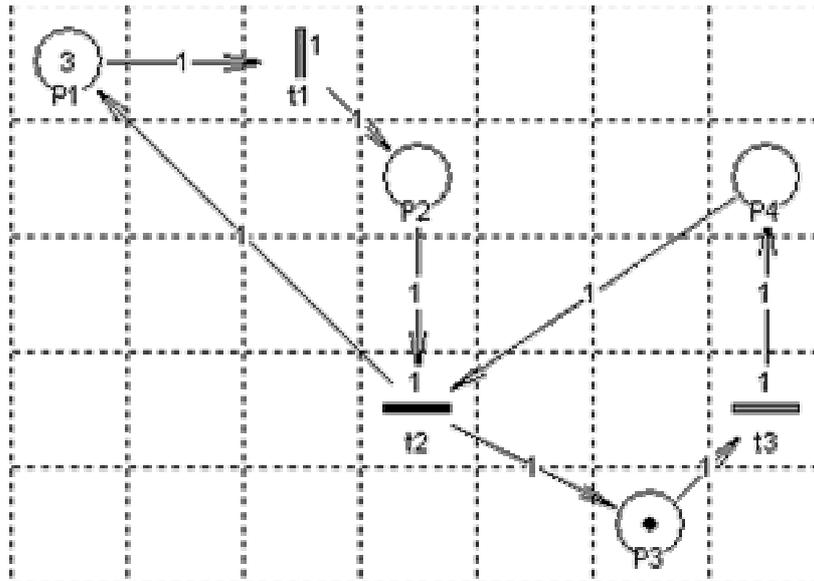


Рис. 100

На рис. 101 приведена маркировка №6 (см. таблицу 32), соответствующая наличию в системе трех отказавших блоков (три метки в вершине P2) и одного, находящегося на восстановлении (метка в вершине P3). Переход t3 активен – еще не истекло время восстановления.

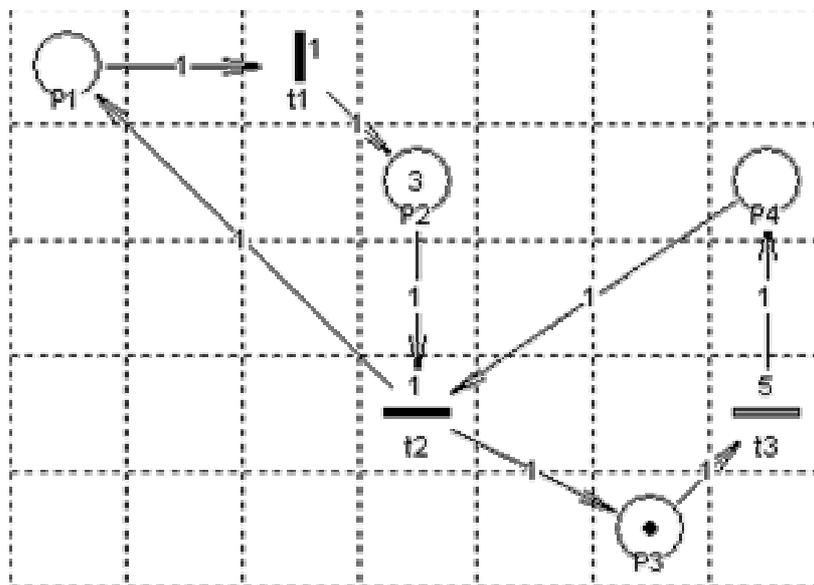


Рис. 101

На рис. 102 приведена маркировка №4 (см. таблицу 32), соответствующая наличию в системе трех отказавших блоков (три метки в вершине P2) и одного, готового к использованию исправного запасного блока (метка в вершине P4). Переход t2 активен – еще не истекло время замены одного из отказавших блоков на запасной.

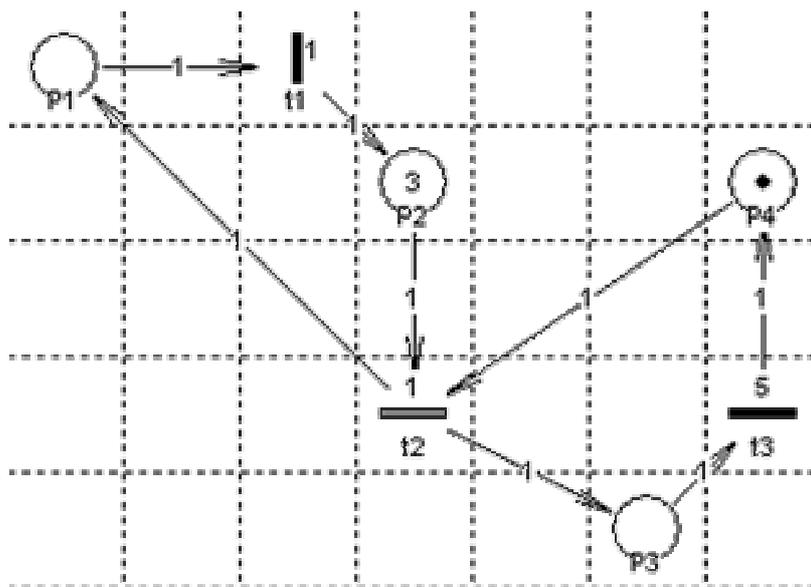


Рис. 102

В примере приведены скриншеты 6 маркировок (всего возможных – 8). При выполнении работы следует приводить скриншеты всех значимых маркировок.

**Пример 2. Сеть Петри для моделирования процесса пакетирования заявок с переменным размером пакета и параллельного обслуживания**

Пусть три независимых потока заявок поступают во входную очередь СМО и пакетируются по 2 при наличии достаточного количества для обслуживания в процессоре параллельного обслуживания, в котором обслуживание очередного пакета подгружается к уже начатому, но одновременно могут обслуживаться не более 3-х пакетов. Количество заявок во входной очереди не должно превышать 10. Если ограничение нарушается, то пакетирование по 2 прекращается, заявки пакетируются по 10 и обслуживаются во втором процессоре. Второй процессор обслуживает пакеты последовательно. В системе реализованы отдельные счетчики обслуженных пакетов по 2 и пакетов по 10.

Сеть Петри, реализующая заданную СМО, имеет вид, представленный на рис. 103.

С переходами  $t_1$ - $t_3$  связаны времена поступления заявок входных потоков. Вершина  $P_1$  моделирует входную очередь. Узлы  $t_4$ - $P_2$ - $t_5$  и  $t_6$ - $P_3$ - $t_7$  моделируют обслуживание пакетов в процессорах (моделирование цикла обслуживания с использованием ингибиторной ветви – см. рис. 22-б в разделе 3.1.2 конспекта лекций). Кроме того, переходами  $t_6$  и  $t_4$  моделируется

процесс пакетирования: для их активизации необходимо 2 или 10 меток соответственно во входной вершине  $P_1$ . При срабатывании перехода  $t_6$  из входной вершины  $P_1$  удаляются две метки, соответствующие заявкам, а в выходной вершине  $P_3$  появляется одна, соответствующая пакету. При появлении в вершине  $P_1$  десяти и более меток ингибиторная ветвь соответствующего веса «запирает» переход  $t_6$  и происходит пакетирование по 10, после чего возобновляется пакетирование по 2.

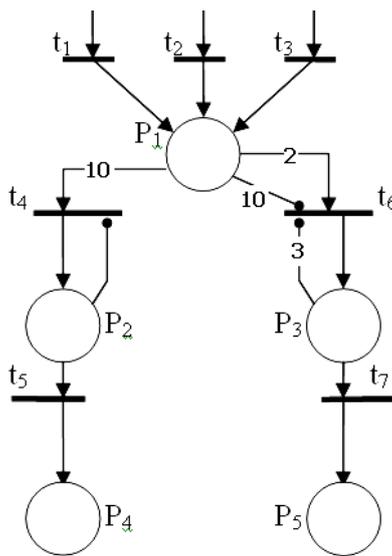


Рис. 103

В таблице 33 представлено описание всех позиций (вершин) сети с точки зрения условий, которые с ними связаны.

Таблица 33

Вершина	Описание
$P_1$	Наличие заявок во входной очереди (количество меток соответствует количеству заявок)
$P_2$	Наличие пакета по 10 на обслуживании во втором процессоре
$P_3$	Наличие пакетов по 2 на обслуживании в первом процессоре (количество меток соответствует количеству параллельно обслуживаемых пакетов)
$P_4$	Наличие в выходной очереди (счетчике) обслуженных пакетов по 10 (количество меток соответствует количеству пакетов)
$P_5$	Наличие в выходной очереди (счетчике) обслуженных пакетов по 2 (количество меток соответствует количеству пакетов)

Описание всех переходов с точки зрения связанных с ними событий и времен их наступления представлено в таблице 34.

Переход	Описание
$t_1-t_3$	Поступление заявок входных потоков 1-3 во входную очередь (время срабатывания перехода соответствует времени поступления очередной заявки потока)
$t_4$	Пакетирование заявок по 10 (время срабатывания перехода соответствует времени пакетирования)
$t_5$	Завершение обслуживания пакета по 10 (время срабатывания соответствует времени обслуживания)
$t_6$	Пакетирование заявок по 2 (время срабатывания перехода соответствует времени пакетирования)
$t_7$	Завершение обслуживания пакета по 2 (время срабатывания соответствует времени обслуживания)

Реализация модели в *Ingproject.exe* (см. файл *пример2.pne*) имеет вид, приведенный на рис. 104 (произвольно заданные времена срабатывания переходов обозначены над или рядом с планкой перехода).

### Пример 3. Сеть Петри для моделирования магистрального канала передачи данных

Описание данного примера приведено в разделе 3.1.2 конспекта лекций, а граф сети Петри с начальной разметкой – на рис. 23. Реализация модели в *Ingproject.exe* (файл *пример3.pne*), имеет вид, представленный на рис. 105.

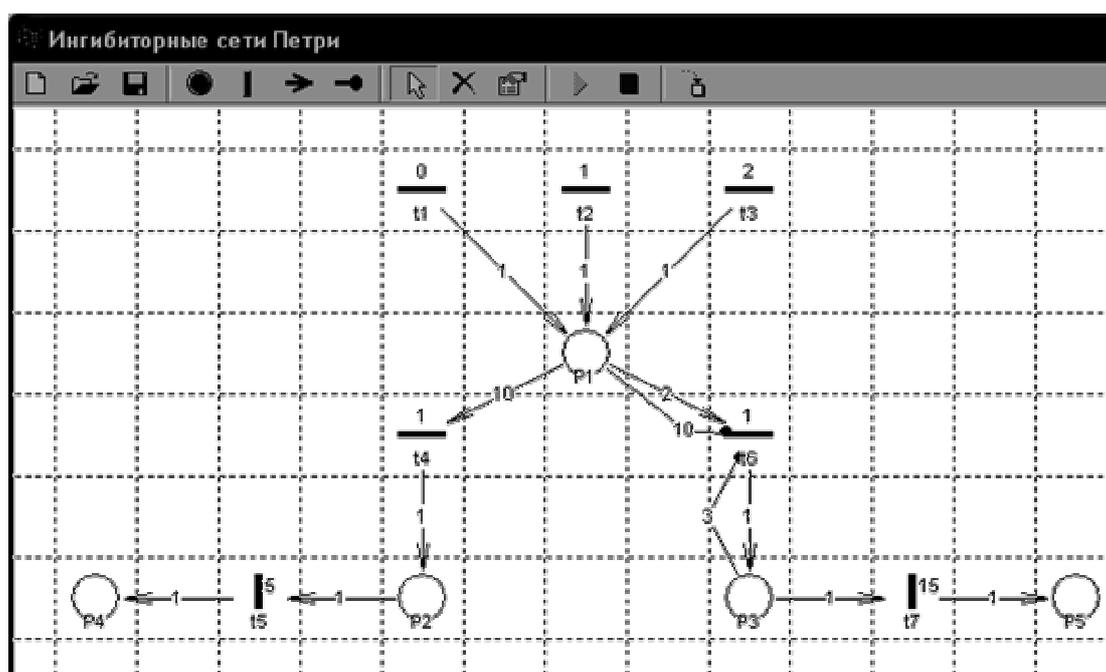


Рис. 104

## Варианты индивидуальных заданий

1. В систему на обслуживание от двух независимых источников поступают задачи 2-х типов. Задачи первого типа требуют 5 минут обслуживания, второго – 10 минут. Если две задачи разных типов приходят одновременно – сначала обслуживается задача первого типа. Если две задачи одного типа приходят одновременно – одна из них теряется.

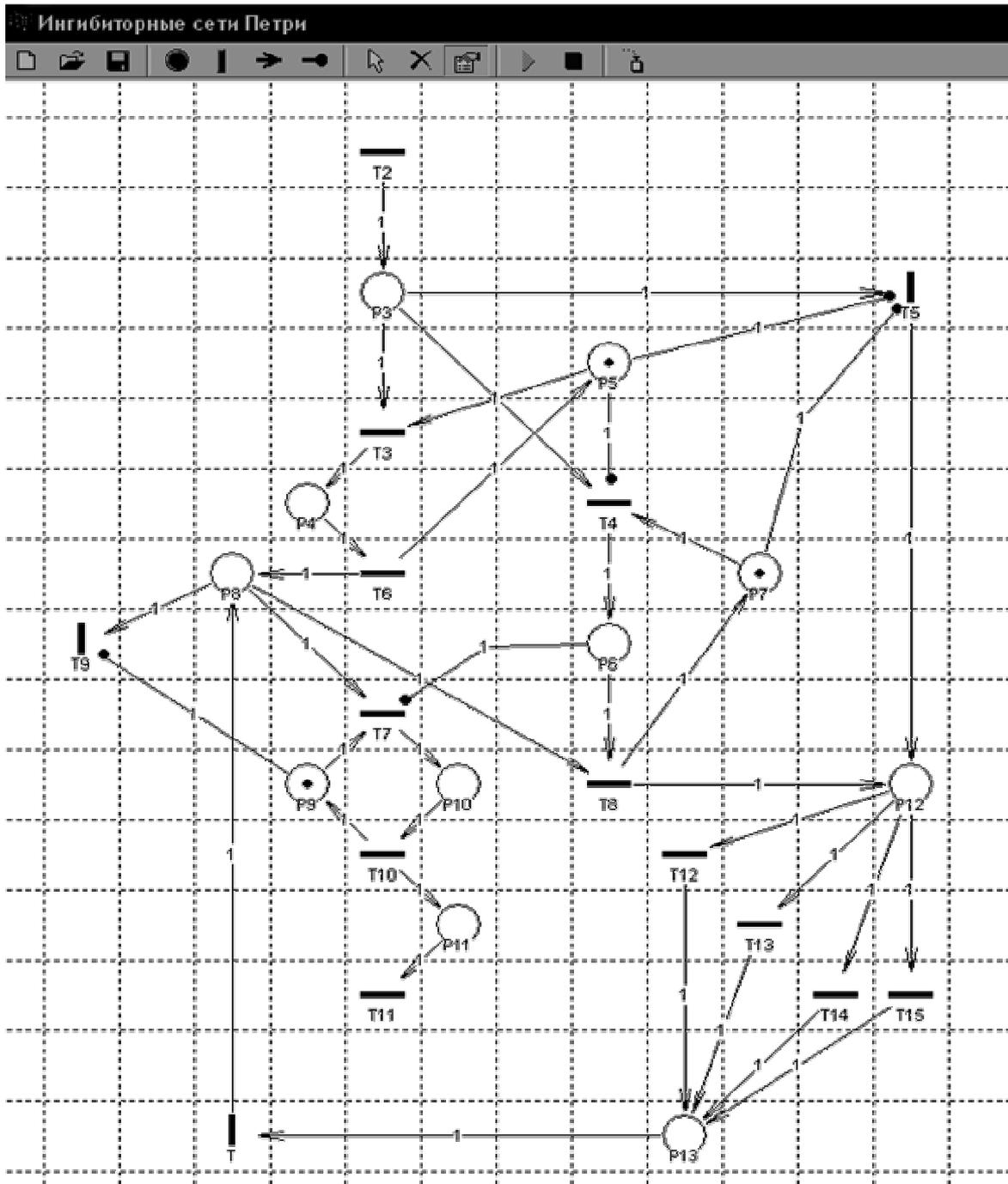


Рис. 105

2. Детали 2-х различных типов поступают по двум транспортерам на позицию сборки. Операция сборки требует наличия 3-х деталей с первого транспортера и одной со второго. Собранные изделия складываются по 10 штук на тележке и вывозятся из цеха. Если заполненную тележку не успевают вывезти, то изделия временно складываются на стеллаж с последующим первоочередным вывозом.

3. В процессе управления ТП непосредственно измеряются три технологических параметра (измерения осуществляются одновременно) и на основании результатов измерения рассчитывается показатель качества. Если хотя бы по одному из измерений происходит отказ (нет результата в требуемый момент времени, результат некорректен), результаты всех измерений аннулируются, а показатель качества оценивается аналитически на основании результатов оценки трех предыдущих значений.

4. Магистраль передачи данных состоит из двух каналов (основного и резервного) и общего накопителя. При нормальной работе данные в виде сообщений передаются по основному каналу. С некоторой периодичностью в основном канале происходит сбой. Если во время сбоя имела место передача сообщения, то передача прерывается и подключается запасной канал, передающий прерванное сообщение с начала. После восстановления основного запасной канал завершает начатую передачу, а основной продолжает работу с очередного сообщения.

5. Готовые изделия по двум транспортерам поступают в ОТК. Если два изделия с различных транспортеров поступают на позицию контроля одновременно, то контроль выполняется поочередно в произвольном порядке. Если по одному из транспортеров изделие поступает в тот момент, когда на позиции контроля уже осуществляется контроль другого изделия, то изделие поступает на резервную позицию контроля. Если заняты обе позиции, изделия накапливаются в предварительном буфере и поступают на освободившуюся позицию. Если результат контроля четыре раза последовательно устанавливает наличие брака, то процесс останавливается до внешней команды на возобновление после устранения неполадок на ТП.

6. По трем транспортерам в промежуточные накопители поступают изделия 3-х типов. В соответствии с внешним заказом формируются комплекты, в каждом из которых должно быть 1 деталь первого типа и по 3 детали второго и третьего. Если какого-либо типа для полного комплекта по  $i$ -тому заказу недостаточно, недоукомплектованный заказ при последующем поступлении деталей доукомплектовывается в первую очередь.

7. На позиции сборки осуществляется сборка изделия из 3-х деталей, поступающих транспортерам. При наличии на момент завершения предыдущей сборки всех 3-х деталей, осуществляется сборка и изделие помещается на выходной транспортер. При отсутствии на момент завершения деталей 1 или 2, осуществляется сборка имеющейся детали с 3-ей и частично собранное изделие помещается в промежуточный накопитель, индивидуальный для каждого типа частичной сборки. Завершение сборки осуществляется в первоочередном порядке при поступлении недостающих деталей. При отсутствии на момент завершения предыдущей сборки детали 3 процесс приостанавливается до ее поступления.

8. По двум транспортерам с накопителями поступают детали двух типов, причем в каждом потоке порядок деталей случаен. Сортировочный автомат группирует детали разных типов из обоих потоков попарно и подает на выходной транспортер, который подает пары деталей на позицию сборки. Если на позиции сборки оказывается более 5-ти пар деталей, сортировка прекращается, а детали по входным транспортерам начинают поступать в накопители входных транспортеров. Сортировка возобновляется в момент, когда на позиции сборки остается только одна пара.

9. В централизованной АСУ ТП используется две ЭВМ. Первая ЭВМ выполняет обработку данных о технологическом процессе и выработку управляющих сигналов, а вторая находится в «горячем» резерве. Данные в ЭВМ поступают через фиксированный интервал времени и обрабатываются в течении некоторого времени. После этого посылается управляющий сигнал, поддерживающий заданную периодичность контроля. Если к моменту посылки следующего набора данных не получен управляющий сигнал, то периодичность контроля для первой ЭВМ уменьшается вдвое, а время обработки – увеличивается вдвое. После выдачи трех управляющих сигналов подряд в этом режиме первая ЭВМ переключается в исходный режим обработки. Режим работы ЭВМ «горячего» резерва постоянен, данные, поступающие до получения управляющего сигнала завершения предыдущей обработки, теряются.

10. На обработку в систему поступают три класса заданий – А, В и С. Задания А и В могут обрабатываться одновременно, задание типа С – монополизирует систему. Задание класса С принимается к обслуживанию, если система полностью свободна; задания типа А и В могут «подгружаться» к текущему обслуживанию. На выходе системы задания сортируются: А и С покидают систему по одному каналу, В – по другому.

11. К ЭВМ подключено 4 терминала. По команде с терминала выполняются операции редактирования, трансляции, планирования и реше-

ния. Если хотя бы один терминал выполняет планирование, остальные простаивают. Если два терминала выдают задания на решение, то оставшиеся два простаивают, и если работают три терминала, выдающие задания на трансляцию, то оставшийся простаивает.

12. На участке термической обработки выполняется цементация и закаливание шестерен, поступающих через  $10 \pm 5$  минут. Цементация занимает  $10 \pm 7$  минут, закаливание –  $10 \pm 6$  минут. Качество определяется суммарным временем обработки. Шестерни с временем обработки больше 25 минут покидают участок, с временем от 20 до 25 минут передаются на повторную закалку, при времени обработки меньше 20 минут должны пройти повторную полную обработку.

13. Детали, необходимые для работы цеха, находятся на центральном и цеховом складах. На цеховом складе хранится 10 комплектов деталей, потребность в которых возникает через  $60 \pm 10$  минут и составляет один комплект. В случае снижения запасов до трех комплектов формируется заявка на пополнение запаса до 20-ти, и посылается на центральный склад, где в течении  $30 \pm 20$  минут происходит комплектование и за  $30 \pm 10$  минут – доставка деталей в цех. Запасов центрального склада всегда достаточно для комплектации.

14. Вычислительная система включает три ЭВМ. В систему в среднем через 30 секунд поступают задания, которые попадают в очередь на обработку в первой ЭВМ. Обработка занимает в среднем 30 секунд. После этого задание поступает одновременно во вторую и третью ЭВМ. Вторая может обработать задание за  $14 \pm 5$  секунд, а третья – за  $16 \pm 1$  секунд. Окончание обработки задания на любой ЭВМ означает снятие ее с решения с той и другой машины.

15. Распределенная база данных системы сбора информации организован на базе ЭВМ, соединенных дуплексным каналом связи. Поступающий запрос обрабатывается на первой ЭВМ и с вероятностью 50% необходимая информация обнаруживается на месте. В противном случае необходима посылка запроса на вторую ЭВМ. Запросы поступают через  $10 \pm 3$  секунд, первичная обработка запроса требует 2 секунд, выдача ответа –  $15 \pm 3$  секунд, передача по каналу связи – 3 секунд. Временные параметры второй ЭВМ аналогичны первой.

16. ЭВМ обрабатывает сообщения, поступающие от датчиков и измерительных устройств через  $3 \pm 1$  секунд. До обработки сообщения хранятся в буфере. Продолжительность обработки сообщений в ЭВМ –  $5 \pm 2$  секунд. Динамика технологического процесса такова, что имеет смысл

обрабатывать сообщения, ожидавшие обслуживания не более 12 секунд. Остальные сообщения теряются. Если теряется 3 сообщения подряд – формируется сообщение об ошибке.

17. Задания на обработку поступают в трехканальную систему с общей очередью и направляются на обработку на свободное устройство с наименьшим номером. В 20% случаев в заданиях обнаруживается ошибка ввода. На время корректировки ввода задание освобождает устройство, после корректировки начинается его повторная обработка. Возможность ошибки при повторной обработке исключается.

18. ЭВМ обрабатывает запросы, поступающие с пяти терминалов. Если запросы с трёх любых терминалов приходят одновременно, то они объединяются в пакет и обрабатываются пакетом; в противном случае – индивидуально. Обслуживание – безочередное, запросы, поступившие во время обслуживания, теряются.

19. На склад готовой продукции предприятия поступают изделия типа А партиями по 5 штук, изделия типа В – партиями по 20 штук. На складе изделия комплектуются в комплект по 10 штук каждого типа. Комплектация начинается, если изделия обоих типов имеются в достаточном количестве. После формирования 10-ти комплектов осуществляется их отгрузка и вывоз со склада. Ежедневный план отгрузки 50 комплектов. После выполнения плана комплектация прекращается, оставшиеся неукомплектованными детали складываются в специальный накопитель.

20. Запросы 4-х типов поступают на обслуживание асинхронно, каждый – в собственную очередь, и обслуживаются попарно в обслуживающем устройстве – запрос 1-го типа с запросом 3-го; 2-го – с запросом 4-го. Обслуживание начинается в момент поступления запроса нужного типа для образования пары. В случае одновременного формирования пар «нечетная» обслуживается в первую очередь.

21. Пять обслуживающих терминалов соединены по кольцу. Терминалы 1 и 3 принимают внешние запросы на обслуживание. Если в момент поступления нового запроса принимающий терминал уже занят обслуживанием, то запрос направляется на обслуживание на ближайший по часовой стрелке свободный терминал со следующим ограничением – запросы, принятые терминалом 1, не могут быть обслужены терминалом 3, а запросы, принятые терминалом 3 не могут быть обслужены терминалом 1. Если в момент поступления запроса все терминалы заняты – он теряется, причем ведется учет потерянных запросов.

22. Запросы трех типов поступают в общий буфер, из которого запросы второго типа отправляются на обслуживание на устройство 1, а запросы первого и третьего типа – на устройство 2. Устройства по каждому типу запросов имеют отдельные выходные очереди. Из этих очередей обслуженные запросы пакетируются попарно в комбинациях 3-1 и 3-2. На выходе системы должны быть реализованы счетчики каждого из вариантов парных комбинаций.

23. Заявки поступают в систему на обслуживание и последовательно обслуживаются в устройствах 1 и 2. Очередь устройства 1 ограничена 3-мя заявками. В устройстве 2 могут одновременно обслуживаться две заявки, причем заявка, пришедшая позже, подгружается к текущему обслуживанию. 90% заявок покидают систему, 10% возвращаются на повторное обслуживание. Заявки, поступившие в систему с момента начала повторного обслуживания и до его завершения, теряются.

24. В систему поступают заявки двух типов – А и В, группируются в пакеты по две и обслуживаются первым устройством. Комбинации заявок АВ, ВА обслуживаются  $2 \pm 1$  секунды; АА, ВВ –  $3 \pm 1$  секунды. Затем пакеты поступают в очередь второго устройства, которое обрабатывает заявки по одной, и тратит на обслуживание заявки типа А  $1 \pm 0,5$  секунду, заявки типа В  $0,5 \pm 0,2$  секунды.

25. Компьютерная система бронирования принимает заказы на купейные билеты. По факту поступления единичных заявок отдельно комплектуются женские и мужские купе, в первую очередь бронируются нижние полки. В прочих случаях комплектация осуществляется по остаточному принципу с выдачей заказчику уведомления.

26. Система автоматического контроля измеряет 4 линейных размера детали. Если все размеры корректны, детали поступают на комплектацию, где формируются наборы из 10-ти штук. Если некорректны 1 или 2 размера – деталь возвращается на доработку, после чего снова поступает на контроль. Если некорректны более 2-х размеров – детали бракуются и поступают на комплектацию брака, где формируются наборы из 10-ти штук. Вывоз годных деталей осуществляется в первую очередь.

27. Заявки на обслуживание поступают от трех независимых источников. Заявки от первого источника обслуживаются первым устройством, заявки от второго – вторым, заявки третьего источника дублируются и поступают одновременно в оба устройства. Если заявка от третьего источника приходит одновременно с заявкой от первого или второго источ-

ника, то заявка первого или второго источника теряется. Если заявки от всех источников приходят одновременно – теряются все. Заявки покидают систему по одной, дубли заявок от третьего источника на выходе ликвидируются. Периодически происходит отказ одного из обслуживающих устройств. На время восстановления все заявки проходят обслуживание в работоспособном устройстве, потерь из-за одновременности и дублирования не происходит.

28. В систему на обработку поступают заявки двух типов; заявки первого типа поступают с вероятностью 30% и принимаются к обслуживанию в первую очередь. Время пребывания заявок в систему ограничено. Заявки каждого типа с каждым признаком завершения обслуживания (обслуживание не началась, обслуживание прервано, обслуживание завершено) покидают систему отдельно. На каждом выходе стоит счетчик заявок, который сбрасывается в ноль после каждого десятка.

# МОДЕЛИРОВАНИЕ

К О Н Т Р О Л Ь

У С П Е В А Е М О С Т И

Недели семестра

1 2 3

4 5 6

7 8 9

10 11 12

13 14 15

16

• аттестация №1

• аттестация №2

✓ тематика  
практических заданий

✓ примеры тестовых вопросов  
выборочного типа

✓ примеры тестовых  
вопросов оценочного типа

• экзамен

• перечень вопросов

✓ для заочников обязательные  
вопросы отмечены «звездочкой»

• типы задач

✓ примеры решения  
рассматриваются в течение  
семестра на лекциях и  
лабораторных работах



## КОНТРОЛЬ УСПЕВАЕМОСТИ

### Первая аттестация

Первая аттестация по курсу проводится на 6-ой учебной неделе для студентов дневной формы обучения. Решение об аттестации или не аттестации студента принимается на основании его индивидуального рейтинга, который складывается из полученных к моменту аттестации баллов рейтинга по всем видам выполняемых работ и результата аттестационного тестирования. Правила проведения и оценивания аттестационного тестирования, а также граничные условия аттестации – см. *Введение в курс* для студентов дневной формы обучения.

В аттестационный тест входят вопросы по всему пройденному к моменту аттестации теоретическому материалу и выполненным работам лабораторного практикума.

Тест состоит из двух частей. В первой части представлены выборочные вопросы следующего типа: приводится некоторое утверждение и предлагается оценить его достоверность путем выбора одного из 4-х вариантов оценки – «верно», «в общем случае верно», «в общем случае неверно» и «неверно». Оценка утверждения как строго верного (неверного) выбирается в том случае, если это утверждение является таковым всегда, и какие-либо исключения из справедливости этого утверждения отсутствуют. Например, утверждение «у треугольника четыре угла» является неверным, и ни при каких обстоятельствах верным стать не может. Утверждение же «у треугольника три угла» является строго верным.

Оценка утверждения как верного в общем случае выбирается тогда, когда утверждение является верным по умолчанию, при обычных, стандартных условиях, но Вам известны частные случаи, при которых оно может являться неверным. Например, утверждение «метод Монте-Карло применяется для моделирования систем случайной природы» является верным в общем случае, так как этот метод действительно, разработан для целей вероятностного моделирования, но также может быть использован и для приближенного решения некоторых детерминистических задач, например для определения площади сложной фигуры.

Аналогично, оценка утверждения как неверного в общем случае выбирается тогда, когда утверждение является неверным по умолчанию, при обычных, стандартных условиях, но Вам известны частные случаи, при которых оно может стать верным. Например, утверждение «если результат трехзначного моделирования показал наличие статического риска сбоя, то

в схеме произойдет сбой» в общем случае неверно, так как анализ рисков сбоя является анализом наихудшего случая, и сбой произойдет не обязательно, а только при определенном сочетании задержек прохождения сигналов, т. е. в некотором частном случае.

Примеры утверждений, используемых в первой части первого аттестационного теста:

- алгебраические уравнения описывают динамические состояния моделируемой системы;
- сочетание различных по природе процессов затрудняет выбор общего критерия подобия;
- передаточная функция является моделью статики;
- у устойчивой системы переходная составляющая фазовой координаты с течением времени стремится к нулю;
- двоичная модель позволяет установить вероятность возникновения статического риска сбоя;
- стационарный режим колебаний является установившимся;
- процедуры анализа заключаются в получении описаний моделируемой системы;
- модели, учитывающие непрерывность пространства и времени, разрабатываются на компонентном уровне иерархии моделей ВС;
- асинхронные модели цифровых схем обладают меньшей точностью в сравнении с моделями в трехзначной логике.

Во второй части теста представлены обычные выборочные вопросы, в которых требуется выбрать один или несколько верных ответов из предложенных вариантов. Если в тексте вопроса есть указание на возможную множественность правильных ответов, т. е. если вопрос начинается со слов «выберите все верные утверждения...» или «выберите из перечня характеристики (одну или несколько)...», то правильными ответами для него могут быть один, два и так далее, в пределах – все варианты, приведенные в вопросе. Примеры вопросов второй части первого аттестационного теста:

А. Выберите из перечня характеристики (одну или несколько), верные по отношению к методам корреляционного анализа:

1. С их помощью можно оценить взаимосвязь распределений двух параметров;
2. С их помощью можно оценить степень согласованности модели с экспериментальными данными;
3. В результате использования этих методов получают функцию зависимости  $y(x)$ ;

4. Значениям  $x=\{1,2,3\}$  соответствуют значения  $y=\{2,4,6\}$ . Коэффициент корреляции  $y(x)$  равен 2.

Б. Передаточная функция структурного примитива системы имеет вид:  $\frac{k}{(T \cdot p + 1)}$ . Укажите ДУ элемента, соответствующее данной передаточной функции:

1.  $T \frac{dy}{dt} + 1 = k \cdot x(t);$

2.  $T \frac{dy}{dt} + y(t) = k \cdot \frac{dx}{dt};$

3.  $T \frac{dx}{dt} + x(t) = k \cdot y(t);$

4.  $T \frac{dy}{dt} + y(t) = k \cdot x(t).$

Ответ на каждый из вопросов теста и в первой, и во второй части оценивается из 10 баллов, т. е. правильный ответ принесет Вам 10 баллов, частично правильный – от 0 до 10, неправильный – 0 баллов.

Кроме тестовых вопросов в первом аттестационном тесте предлагаются практические задания на оценку статического риска сбоя и построение соответствующих временных диаграмм. Вопросы и задания, аналогичные предлагаемым в аттестационных тестах, входят в тестовую часть экзаменационных билетов по курсу.

### **Вторая аттестация**

Вторая аттестация по курсу проводится на 12-ой учебной неделе для студентов дневной формы обучения. В аттестационный тест входят вопросы по всему пройденному с момента первой аттестации теоретическому материалу и выполненным работам лабораторного практикума.

Структура второго аттестационного теста, типы предлагаемых вопросов и правила оценки такие же, как и в первой аттестации.

Примеры утверждений, используемых в первой части второго аттестационного теста:

- отношение иерархии используется для описания структур в рамках теоретико-автоматного подхода;
- сеть Петри функционирует в дискретном времени;
- ингибиторная ветвь сети Петри запрещает запуск перехода до изъятия маркера из входной вершины;

- в моделях схемотехнического уровня пространственные координаты дискретны;
- анализ функционирования АКА не требует информации о начальных условиях.

Примеры вопросов второй части второго аттестационного теста:

А. Для некоторой системы создана модель управления. Выберите верные по отношению к этой модели утверждения (*одно или несколько*):

1. Если система находится в установившемся состоянии, то ДУ модели вырождаются в алгебраические;
2. Решением системы уравнений модели является переходной процесс системы;
3. В общем случае возможна линеаризация ДУ модели.

Б. На рис. 106 указана нулевая разметка простой сети Петри. Время срабатывания всех переходов (1-4) равно 1 секунде.

1. Верно ли, что при нулевой разметке активны все переходы (да, нет)?
2. Верно ли, что максимальное число меток в вершине Р, возможное при такой нулевой разметке, равно 4 (да, нет)?
3. Верно ли, что при заданной нулевой разметке переход 1 может вообще не сработать (да, нет)?

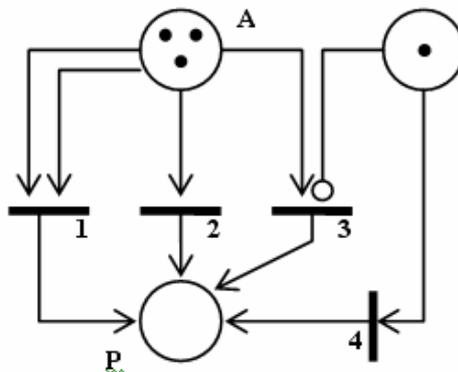


Рис. 106

В. При использовании кусочно-линейной аппроксимации аппроксимирующая функция:

- 1) разрывна в узлах, а ее производные непрерывны;
- 2) непрерывна в узлах, а ее производные разрывны;
- 3) разрывна в узлах;
- 4) между двумя соседними узлами имеет вид аналитической функции, описывающей реальную зависимость.

Кроме тестовых вопросов в тесте предлагаются практические задания по описаниям абстрактных конечных автоматов, анализу маркировок сетей Петри и решению задачи управления для структурного примитива, описываемого дифференциальным уравнением первого порядка.

## Экзамен

Правила организации, проведения и оценивания экзаменационной работы, а также условия допуска и учет семестрового рейтинга в итоговой оценке освоения курса – см. *Введение в курс* для соответствующей формы обучения.

### Вопросы к экзамену

Звездочкой отмечены вопросы, являющиеся обязательными для студентов заочной формы обучения (включены в *часть I* билетов по курсу – см. *Введение в курс* для заочной формы обучения).

Для студентов дневной формы обучения обязательным является полный перечень вопросов.

1. Определение модели, моделирования, свойств интерполяции и экстраполяции. Классификация моделей по критерию подобия и соотношению точности/абстрактности.

2. Математические модели – критерий подобия, фазовое пространство и координаты. Классификация и характеристика математических моделей.

3. Примеры использования и сравнительный анализ моделей различных типов по степени соответствия объекту моделирования.

4. \*Режимы функционирования технических объектов моделирования. Модельные тестовые воздействия.

5. \*Виды и модели анализа технических объектов моделирования.

6. Системный подход. Элементы описания объекта моделирования как системы.

7. Системный подход. Совокупность процедур синтеза и анализа в итерационном цикле проектирования.

8. \*Иерархические уровни моделирования ВС. Структурные примитивы уровней моделирования.

9. \*Математический аппарат моделирования ВС на различных уровнях декомпозиции.

10. Переход от компонентного моделирования к схмотехническому. Модели с распределенными и сосредоточенными параметрами.

11. Переход от схмотехнического моделирования к функционально-логическому. Модели с сосредоточенными параметрами и информационные модели.

12. Подходы к описанию функциональных структур. Типы элементов функциональных структур СМО, используемых для моделирования ВС.

13. Моделирование структурных примитивов. Постановка задачи управления. Линеаризация дифференциальных уравнений. Аппарат передаточных функций.

14. Способы получения моделей управления. Примеры получения моделей управления для различных структурных примитивов, относящихся к одному типу элементарных звеньев систем.
15. Моделирование структурных примитивов. Постановка задачи идентификации. Аппроксимация зависимостей.
16. Моделирование структурных примитивов. Постановка задачи идентификации. Методы корреляционного и регрессионного анализа.
17. Методы планирования эксперимента. Логические основания планирования эксперимента. Матрицы планирования. Типы экспериментов.
18. Вероятностное моделирование. Метод Монте-Карло для дискретного распределения вероятностей.
19. \*Использование метода Монте-Карло для реализации неравномерных распределений.
20. Абстрактные конечные автоматы 1-го и 2-го рода. Матрицы переходов и выходов. Представление графом.
21. \*Простые временные сети Петри. Способы задания. Моделирование элементарного цикла обслуживания простой временной сетью Петри.
22. \*Ингибиторные сети Петри. Моделирование элементарного цикла обслуживания ингибиторной сетью Петри. Пример моделирования системы или процесса ингибиторной сетью Петри.
23. \*Типы сетей Петри, используемые для моделирования ВС. Пример моделирования процесса параллельного обслуживания заявок с пакетированием сетью Петри.
24. \*Моделирование ВС с использованием теории массового обслуживания. Классификация СМО. Типы элементов функциональных структур СМО, используемых для моделирования ВС.
25. \*Аналитические модели массового обслуживания.
26. \*Обслуживание с ожиданием. Постановка задачи. Свойства экспоненциального распределения времени обслуживания. Обслуживание как Марковский процесс.
27. Обслуживание с потерями. Обслуживание с ограниченным временем ожидания. Постановка задачи. Обслуживание как Марковский процесс.
28. Обслуживание с потерями. Обслуживание с ограниченным временем пребывания. Постановка задачи. Обслуживание как Марковский процесс.
29. Обслуживание с потерями. Моделирование приоритетного обслуживания с использованием теории массового обслуживания.
30. \*Имитационные модели массового обслуживания. Элементы имитационных моделей.
31. \*Способы управления модельным временем.
32. Алгоритмы имитационного моделирования для событийного управления модельным временем.
33. Алгоритмы имитационного моделирования для пошагового управления модельным временем.

### **Типы экзаменационных задач**

#### *1. Анализ цифровой схемы на наличие риска сбоя.*

Для заданной схемы (см. рис. 107) подобрать такое изменение вектора входных сигналов ( $a, b, c$ ), при котором будет иметь место риск сбоя на выходе  $e$ . Подтвердить расчетом в трехзначной логике.

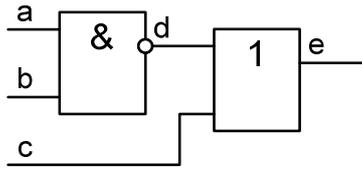


Рис. 107

Для выбранного изменения вектора входных сигналов построить следующие временные диаграммы:

- нормальной работы схемы для синхронного переключения входных сигналов ( $a, b, c$ );
- работы со сбоем для переключения входных сигналов ( $a, b, c$ ) с задержками;
- работы без сбоя для переключения входных сигналов ( $a, b, c$ ) с задержками.

2. Анализ переходных процессов. Решение прямой и обратной задачи управления.

Модель структурного примитива задана в виде дифференциального уравнения:  $T \cdot \frac{dy(t)}{dt} + y(t) = k \cdot x(t)$ . Решение задачи управления для данного структурного примитива при подаче на вход единичной ступеньки в момент времени  $t = 5$  с имеет вид, приведенный на рис. 108:

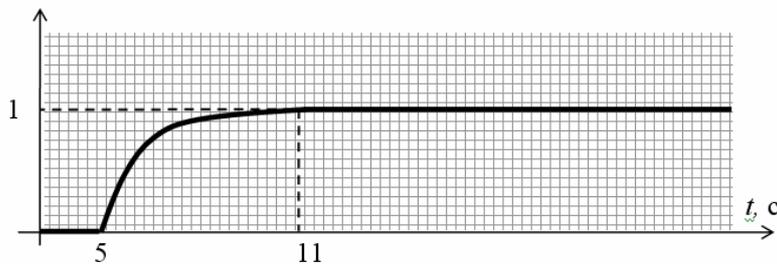


Рис. 108

Определить значение постоянных  $k$  и  $T$ . Решить задачу управления для входного воздействия, приведенного на рис. 109.

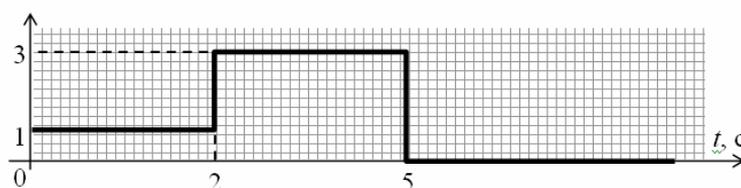


Рис. 109

3. Разработка заданной формы представления абстрактного конечного автомата.

Задан граф АКА второго рода (см., например, рис. 14). Построить матрицы переходов и выходов. Построить граф и матрицы переходов и выходов эквивалентного автомата первого рода.

4. Разработка модели цифровой комбинационной схемы в виде абстрактного конечного автомата.

По условиям функционирования заданной схемы (см. рис. 110) сигнал на входе  $c$  всегда равен 1, а входные сигналы  $a$  и  $b$  изменяются произвольно. Разработать модель схемы в виде автомата Мура, считая множеством состояний автомата значения выходного сигнала  $d$ , выходным алфавитом – множество значений сигнала  $e$ .

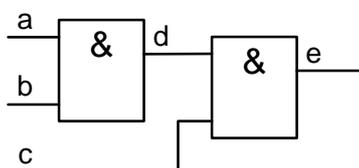


Рис. 110

5. Построение временной диаграммы изменения Марковской компоненты модели процесса обслуживания с ограничением времени ожидания или пребывания.

В многоканальной системе массового обслуживания время пребывания ограничено 6-ю секундами. Время поступления заявок на вход  $i$ -того устройства и длительности их обслуживания заданы в таблице 35. В нулевой момент времени устройство свободно от обслуживания.

Построить диаграмму изменения компоненты  $\xi_i(t)$  многомерного Марковского случайного процесса  $\xi(t) = \{\xi_1(t), \xi_2(t), \dots, \xi_m(t)\}$ , где  $\xi_i(t)$  – время, которое должно протечь от момента  $t$  до освобождения прибора с номером  $i$  от обслуживания требований, поступивших ранее  $t$ . Для каждой заявки указать признак завершения обслуживания (полностью обслужена без ожидания, полностью обслужена с ожиданием, потеряна с незавершенным обслуживанием из-за превышения времени ожидания).

Таблица 35

№ заявки	1	2	3	4
время поступления, сек.	2	6	10	15
длительность обслуживания, сек.	2	5	8	5

*6. Моделирование процесса или системы массового обслуживания сетью Петри.*

Задания поступают в трехканальную систему массового обслуживания с общей очередью и направляются на обработку на свободное устройство с наименьшим номером. В 20% случаев в заданиях в процессе обслуживания обнаруживается ошибка ввода. На время корректировки ввода задание освобождает устройство, после корректировки выполняется его повторная обработка. На время корректировки новые задания к обслуживанию не принимаются.

Разработать сеть Петри для моделирования описанной системы, привести ее начальную разметку. Описать все использованные переходы с точки зрения действий, которые с ними связаны. Описать все использованные вершины с точки зрения условий, которые с ними связаны.

*7. Анализ заданной сети Петри: построение возможных конечных маркировок или маркировок, которые будут иметь место через заданный интервал времени.*

Дана начальная маркировка сети Петри (см. рис. 111-а). Нарисуйте все возможные конечные маркировки этой сети. Нарисуйте начальную маркировку, которая может привести сеть к конечной маркировке, изображенной на рис. 111-б.

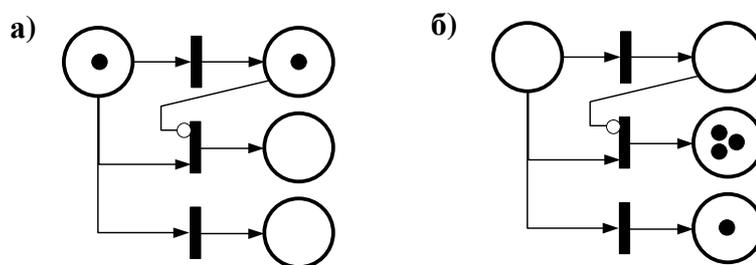


Рис. 111

# МОДЕЛИРОВАНИЕ

## Курсовое Проектирование

• методические указания

• содержание  
пояснительной записки

• защита и оценка

• GPSS World

• объекты языка

• операторы

✓ модификации параметров заявок

\*GENERATE\*SPLIT\*TERMINATE\*ASSEMBLE\*ADVANCE\*  
\*MATCH\*GATHER\*ASSIGN\*INDEX\*MARK\*PLUS\*PRIORITY\*

✓ маршрутизации движения заявок

\*DISPLACE\*TRANSFER\*LOOP\*TEST\*GATE\*

✓ аппаратных объектов

\*SEIZE\*RELEASE\*PRE-EMPT\*RETURN\*FUNAVAIL\*  
\*FAVAIL\*ENTER\*LEAVE\*SAVAIL\*SUNAVAIL\*LOGIC\*

✓ сбора статистики

\*QUEUE\*DEPART\*TABULATE\*

✓ организации списков пользователя

\*LINK\*UNLINK\*

✓ специальные

\*COUNT\*SELECT\*



# КУРСОВОЕ ПРОЕКТИРОВАНИЕ

## Методические указания

### Содержание пояснительной записки

Соответствие содержания пояснительной записки изложенному ниже аннотированному плану является обязательным. Рекомендуется рассматривать приведенный план совместно с примером пояснительной записки (см. файл *Пояснительная записка.doc*). При обоснованной необходимости допускается дополнять содержание другими подразделами, но исключать обязательные разделы и подразделы нельзя.

#### 1. *Имитационное моделирование системы массового обслуживания.*

##### 1.1 *Анализ динамики функционирования СМО.*

- Дается характеристика функционирования СМО, описание которой приведено в задании на курсовое проектирование, выполняется классификация СМО (см. *раздел 3.2 конспекта лекций*). Если СМО не является однородной (многофазной или многоканальной с одинаковыми устройствами обслуживания), то выполняется классификация отдельных узлов СМО. Для внутренних узлов особое внимание следует уделить идентификации типа потока, поступающего в очередь. Если он не является однородным, а, например, складывается из нескольких независимых потоков с разными типами распределений времен поступления заявок, то следует использовать классификационное обозначение для произвольного потока ( $G$ ). Разрабатывается функциональная схема СМО (если не задана в задании).

- Рассматриваются единовременность и последовательность протекания процессов, все возможные последовательности событий, приводящие к каждому из возможных состояний системы; процессы, вызывающие смену состояний; условия переходов из одного состояния в другое.

##### 1.2 *Формализация модели СМО средствами GPSS World.*

- Выполняется формализация модели. Нумерованным списком приводятся все характеристики функционирования СМО, заданные к определению в части 1 задания на курсовой проект. Все характеристики классифицируются как первичные и вторичные параметры моделирования в GPSS. К первичным параметрам относят характеристики, получаемые автоматически, встроенными средствами GPSS, т. е. являющиеся стандартными числовыми атрибутами (СЧА). Приводится табличная форма классификации первичных параметров.

- К вторичным параметрам относят все характеристики, которые необходимо получить с использованием первичных параметров программно или расчетно, по результатам статистического моделирования. Для вторичных параметров вводятся собственные обозначения и приводятся формулы расчета через первичные параметры и, при необходимости, дополнительные переменные.

### **1.3 Приближенный детерминированный анализ СМО.**

- С использованием детерминированной модели приближенно оцениваются все характеристики функционирования СМО, подлежащие определению в соответствии с заданием, за исключением среднеквадратических отклонений, максимальных длин неограниченных очередей и длин очередей с ограничениями, которые нельзя учесть в детерминированном анализе (см. пример пояснительной записки). В качестве исходных данных для расчета используются математические ожидания (средние значения) времен поступления и обслуживания, вероятности тех или иных событий в СМО и т. д. Результаты расчета сводятся в итоговую таблицу.

- Обосновывается целесообразность использования имитационного статистического моделирования СМО для точного анализа характеристик функционирования.

- Особый случай представляют СМО с управлением обслуживанием по отклонениям величин: например, с зависимостью времени вторичного обслуживания от того, больше или меньше среднего значения времени обслуживания составило первичное обслуживание. Для таких случаев детерминированный анализ либо вообще неприменим, либо дает заведомо совершенно не точный результат, поэтому его можно не выполнять, однако обязательно нужно привести доказательное обоснование такого решения.

### **1.4 Определение числа прогонов имитационной модели для обеспечения заданной точности моделирования по лимитирующему процессу.**

- На основании анализа исходных данных из числа процессов СМО, время которых распределено неравномерно, устанавливается лимитирующий процесс – процесс поступления или обслуживания, имеющий наименьшую интенсивность или частоту обращения. Для данного процесса программно или в любом математическом пакете с использованием метода Монте-Карло и встроенного генератора равномерного распределения реализуется генератор случайных чисел по заданному закону.

- Обязательно приводятся формулы расчета случайных величин по методу Монте-Карло, фрагмент реализации расчета в математическом пакете и графики зависимости МО от числа реализаций случайного процесса, полученные в нескольких экспериментах.

- По результатам экспериментов устанавливается число прогонов имитационной модели, гарантирующее заданную точность вероятностного моделирования по лимитирующему процессу (погрешность по математическому ожиданию времени поступления или обслуживания – не более 5%). Для обеспечения запаса по точности окончательное число прогонов выбирается в 1,5-2 раза больше, чем полученное.

- Выбранное число прогонов сравнивается с заданным временем моделирования (если задано). Проверяется, успеет ли за заданное время моделирования лимитирующий процесс осуществиться необходимое число раз. Если нет – время моделирования увеличивается до значения, гарантирующее требуемое количество осуществлений лимитирующего процесса для обеспечения заданной точности.

### **1.5 Имитационное моделирование СМО в среде GPSS World.**

- Приводится характеристика принципиальных особенностей реализации: используемых переменных, ячеек памяти, массивов, параметров заявок и т. д. с подробным описанием (имена, типы, соответствие обозначениям переменных в формулах из п.1.2).

- Приводится реализация имитационной модели в GPSS World. Программный код должен быть структурирован (разбит на поименованные блоки: например, блок генерации заявок входных потоков, блок моделирования первого устройства обслуживания и т. д.) и обязательно иметь построчный комментарий. Код допускается вставлять в записку только отформатированным текстом. Разрешается уменьшать шрифт до 8pt. Не допускается вставка фрагментов программного кода в графическом формате (экранные копии). Построчный комментарий должен быть информативным, т. е. описывать не операцию, выполняемую оператором GPSS, а изменение состояния моделируемой СМО в связи с выполнением операции, заданной оператором. Например, информативный комментарий к оператору формата TEST NE FN\$типе,3,PRIOR может иметь вид «проверка типа заявки и переход заявок 3-го типа к приоритетному обслуживанию». Не информативны комментарий для того же случая: «условный переход к метке PRIOR».

- Приводятся результаты модельного эксперимента в виде полного стандартного отчета GPSS World Simulation Report, с расшифровкой па-

раметров (полученные для числа прогонов модели, определенного в п. 1.4). Приводятся результаты программного определения характеристик функционирования с расшифровкой. Приводится расчет характеристик функционирования СМО, не получаемых программно. в расчете обязательно указывается источник данных для каждого из используемых значений (ссылка на номер оператора в стандартном отчете или на СЧА со ссылкой на графу стандартного отчета).

- Приводится табличная форма сравнения и производится анализ согласованности полученных результатов с результатами оценки характеристик на детерминированной модели, полученными в п.1.3, приводятся выводы по Части 1 курсового проектирования.

## **2. Анализ и оптимизация системы массового обслуживания.**

В каждом задании на проектирования могут быть даны условия одной или более задач оптимизации, или задач анализа моделируемой СМО.

Пример задач оптимизации: «1. Определить интенсивность входного потока  $\lambda$ , при которой коэффициент использования  $S_1$  будет составлять 0,63. 2. Определить среднее время обслуживания в  $S_2$ , при котором коэффициент использования  $S_3$  будет составлять не более 0,02.». Каждая из задач решается отдельно. Исходными данными для каждой задачи является первоначальный вариант реализации модели СМО (п.1.5).

### **2.1 Формализация модели СМО для задач анализа и оптимизации.**

- Для каждой задачи анализа, если это необходимо, выполняется дополнительная формализация модели. Нумерованным списком приводятся все характеристики функционирования СМО, заданные к определению в части 2 задания на курсовой проект. Выполняется их классификация и проч. (см. п.1.2).

- Для каждой задачи оптимизации, если это необходимо, выполняется дополнительная формализация модели. Приводятся начальное значение параметра оптимизации, полученное в модельном эксперименте п.1.5, и начальное значение изменяемого параметра СМО или входного потока. Выдвигается гипотеза об увеличении/уменьшении изменяемого параметра с некоторым шагом. Если изменяемых параметров 2 и более – строится матрица планирования эксперимента.

### **2.2 Модификация имитационной модели.**

- Приводится характеристика принципиальных особенностей модификации: используемых переменных, ячеек памяти, массивов, параметров заявок и т. д. с подробным описанием (имена, типы, соответствие обозначениям переменных в формулах из п.2.1).

- Приводятся фрагменты или, при необходимости, вся реализация модифицированной имитационной модели в GPSS World с построчным комментарием.

### **2.3 Реализация модельных экспериментов и анализ результатов.**

- Для каждой *задачи анализа* приводятся результаты модельного эксперимента (полученные для числа прогонов модели, определенного в п.1.4) в виде полного стандартного отчета GPSS World Simulation Report, с расшифровкой параметров. Приводятся результаты программного определения характеристик функционирования с расшифровкой. Приводится расчет характеристик функционирования СМО, не получаемых программно. Если необходимо, выполняются прочие виды расчетов – например, рассчитываются регрессионные зависимости целевых параметров от заданного набора факторов.

- Для каждой *задачи оптимизации* приводятся результаты экспериментов, осуществленных согласно плану п.2.1. в виде полного стандартного отчета GPSS World Simulation Report, с расшифровкой параметров. Приводятся результаты программного определения характеристик функционирования с расшифровкой. Приводится расчет характеристик функционирования СМО, не получаемых программно. Приводятся выводы о возможности/невозможности решения поставленной задачи оптимизации указанным способом. Если решение возможно – приводится окончательное значение изменяемого параметра. Если решение указанным способом невозможно – определяются пути возможного решения другими способами.

- Приводится анализ результатов решения задач анализа/оптимизации и выводы по Части 2 курсового проектирования.

## **3. Функциональное моделирование системы массового обслуживания сетью Петри.**

### **3.1 Формализация модели СМО средствами сетей Петри.**

- Дается характеристика сетей Петри как средства моделирования СМО; обосновывается возможность использования для моделирования заданной системы конкретного типа сетей Петри.

- Приводится граф сети Петри с нулевой (начальной) разметкой, разработанной по заданному варианту. Если для моделирования выбрано ПО – скриншот реализации модели с начальной разметкой.

- В табличной форме приводится описание всех позиций (вершин) сети с точки зрения условий, возникающих в процессе функционирования СМО; описание всех переходов с точки зрения связанных с ними

событий; описание временных, приоритетных, вероятностных и прочих характеристик сети Петри.

### **3.2 Имитационное моделирование СМО сетью Петри.**

- Приводятся графы (в случае реализации в ПО моделирования – скриншеты) всех возможных маркировок сети Петри, полученные для всех состояний системы, дается описание каждой маркировки.

- Приводится анализ результатов моделирования заданной СМО сетями Петри и выводы по Части 3 курсового проектирования.

*Выводы.*

- В выводах по курсовому проекту обобщаются результаты выполнения всех частей проекта. Нумерованным списком дается развернутая, предметная характеристика решенных задач и полученных результатов, приводятся основные качественные и численные характеристики СМО, полученные в результате моделирования.

- В табличной форме дается характеристика всех файлов реализации курсового проекта, предоставляемых к защите в электронном виде.

### **Организация проектирования, защита и оценка**

Цели и тематика курсового проектирования; сроки; аттестация; баллы рейтинга, начисляемые за промежуточные и итоговые результаты курсового проектирования; выбор категории сложности и базовой оценки – см. *Введение в курс* для соответствующей формы обучения.

Консультации по курсовому проектированию проводятся согласно расписанию, в течение всего срока проектирования (12 недель с момента выдачи задания), до даты завершения работ над проектом. На консультациях преподаватель отвечает на конкретные, четко сформулированные проблемные вопросы по тематике и практической реализации проекта, не отвечает на вопросы типа «что делать?» и не проверяет курсовые проекты, т. е. консультации не являются ни аналогом лекций по курсовому проектированию, ни репетицией защиты.

В электронном виде выдаются дополнительные материалы по курсовому проектированию: пример модельных реализаций по типовому заданию и пример пояснительной записки.

Выбор базовой оценки означает, что курсовой проект оценивается исходя из указанной оценки, что, разумеется, не гарантирует ее получения без соответствующего уровня работы. Базовая оценка на защите будет корректироваться в большую или меньшую сторону в зависимости от поощрений и штрафов, перечисленных в таблице 3б. В первую очередь озна-

комьтесь с перечнем штрафующих ошибок. Именно здесь студенты теряют больше всего баллов.

Курсовой проект представляется к защите в твердой копии и в электронном виде. Полный перечень материалов по каждому виду указан в задании на курсовое проектирование. В твердой копии материалы представляются сброшюрованными в папке в следующем порядке: титульный лист, задание на курсовое проектирование, пояснительная записка. В папку вкладывается (не подшивается!) распечатка бланка защиты курсового проекта (см. файл *Бланк защиты КП.doc*).

Защита проектов проводится индивидуально. В процессе защиты путем постановки *трех контрольных вопросов-заданий* проверяется самостоятельность выполнения проекта, навыки по использованию программного обеспечения моделирования, умение изменять характеристики разработанных моделей и режимов реализации модельных экспериментов.

Таблица 36

Условия начисления	Баллы
<i>Поощрения</i>	
1. Высококачественное, образцово-показательное оформление записки в соответствии с ГОСТ 2.105-95.	от 0,4 до 0,6
2. Повышенная сложность задания в рамках базовой оценки.	от 0,5 до 1,5
3. Досрочная сдача: – студент, сдавший проект первым; – студент, сдавший проект вторым; – студент, сдавший проект третьим.	1,5 1 0,8
4. Дополнительные виды и результаты анализа, не указанные в задании на проектирование, выявляющие принципиальные особенности функционирования моделируемой системы (например, определение условий, при которых случайные процессы проявляют признаки стационарности; наблюдаемая цикличность или периодичность в процессах функционирования, автоколебания и т. п.)	от 0,8 до 2
5. Нестандартные решения в GPSS (использование списков пользователя; PLUS-процедур; генерируемых экспериментов или экспериментов пользователя, осуществляемых в автоматическом режиме и т. п.)	от 0,5 до 2
6. Программная реализация модели сети Петри или реализация в любом пакете моделирования сетей Петри помимо IngProject.	от 0,5 до 2
7. Дополнительное представление (помимо реализации в GPSS) результатов моделирования системы или ее узлов в любом пакете, пригодном для статистического моделирования СМО – Simulink, MatLab и т. п.	от 0,5 до 2

Условия начисления	Баллы
8. Оптимизация задания на проектирование по результатам предварительного исследования статистических характеристик процессов, протекающих в СМО.	от 0,4 до 1
9. Интересные наблюдения, выводы.	от 0,4 до 1,5
10. Альтернативные варианты решения задач оптимизации, отличные от указанных в задании, и обеспечивающие решение более экономичным, более эффективным способом.	от 0,4 до 1
<i>Штрафы</i>	
1. Защита курсового проекта не в срок без уважительной причины.	1
2. Грубая ошибка проектирования (неправильная работа модели, неверная формула расчета параметров модели, неверный тип распределения случайных величин и т. п.)	от 0,5 до 2
3. Неполное соответствие реализации заданию на проектирование.	от 0,5 до 2
4. Заимствования из других курсовых проектов.	от 0,5 до 3
5. Плохое оформление пояснительной записки (грамматические ошибки, нарушения ГОСТа, отсутствие ссылок на рисунки, таблицы и литературу, отсутствие номеров страниц, оглавления, списка литературы и т. п.)	от 0,2 до 0,8
6. Нарушения содержания пояснительной записки – отсутствие обязательных разделов, не соответствие содержания и результатов разделов заголовкам, методическим указаниям.	от 0,2 до 2
7. Формальное или поверхностное выполнение анализа исходных данных и/или анализа и интерпретации результатов моделирования. Необоснованные выводы.	от 0,4 до 1
8. Отсутствуют или очень кратки и непонятны пояснения выполненных модельных экспериментов, не определена цель экспериментов, нет анализа результатов, эксперименты некорректно спланированы.	от 0,4 до 1
9. Неверный ответ на вопросы защиты или студент не в состоянии внести заданные изменения в модельную реализацию (указан штраф за каждый неверный ответ или невыполненное задание).	1
10. Отсутствие структурирования и форматирования в тексте программной реализации. Вставка фрагментов программного кода в графическом формате (экранные копии). Неинформативный построчный комментарий.	от 0,2 до 0,8

### Основы моделирования в GPSS World

В данном разделе подробно излагаются только те аспекты моделирования в GPSS World, которые *не рассматриваются* в лабораторном практикуме по курсу «Моделирование». Для удобства использования в тексте имеются ссылки на соответствующие разделы лабораторного практикума и другие материалы.

## Объекты языка имитационного моделирования GPSS

Система GPSS World предназначена для имитационного моделирования систем с дискретными и непрерывными процессами. Языком моделирования в ней является язык GPSS, который построен в предположении, что модель сложной системы можно представить совокупностью элементов и логических правил их взаимодействия в процессе функционирования моделируемой системы, причем набор логических правил ограничен и может быть описан небольшим числом стандартных операций. Комплекс программ, описывающих функционирование объектов и выполняющих логические операции, является основой для создания программной модели системы. Кроме этого комплекса в составе системы GPSS World имеется специальная программа-планировщик, которая выполняет следующие функции:

- продвижения по заданным маршрутам заявок;
- планирование событий, происходящих в модели, путем регистрации времени наступления каждого события и выполнения их в нарастающей временной последовательности;
- регистрация статистической информации;
- продвижение модельного времени в процессе моделирования.

Чтобы обеспечить правильную последовательность обработки событий во времени, имеются системные часы, хранящие значения абсолютно-го модельного времени.

Характеристика объектов GPSS приведена в таблице 37.

Заявки являются *динамическими объектами*, которые создаются в определенных точках модели, продвигаются планировщиком через операторы, а затем уничтожаются. С каждой заявкой связаны параметры, которые нумеруются или именовются. Номера параметров и имена используются для ссылок на значения, присвоенные параметрам. Заявкам может присваиваться приоритет. Приоритет определяет предпочтение, которое получает заявка, когда она и другие заявки претендуют на один и тот же ресурс.

Таблица 37

Категория	Типы объектов
Динамическая	Заявки
Операционная	Блоки (операторы)
Аппаратная	Обслуживающие устройства, накопители, логические ключи
Вычислительная	Переменные, функции, генераторы случайных чисел

Категория	Типы объектов
Статистическая	Очереди, таблицы
Запоминающая	Ячейки, матрицы ячеек
Группирующая	Числовые группы, группы заявок, списки

*Объекты аппаратной категории* – это абстрактные элементы, на которые может быть декомпозирована реальная система. Воздействуя на эти объекты, заявки могут изменять их состояние и влиять на движение других заявок. Ранее происходившие события в системе могут заблокировать, изменить движение заявок и наступление последующих событий. Например, один канал связи вышел из строя и все последующие заявки на передачу сообщений должны быть направлены на исправные каналы связи. Для моделирования таких ситуаций введены *логические ключи*. Заявка может устанавливать эти ключи в положение «включено» или «выключено». Впоследствии состояние ключей может быть проверено другими заявками для определения пути их дальнейшего следования.

*Операционные объекты*, т. е. операторы, задают логику функционирования модели системы и определяют пути движения заявок между объектами аппаратной категории. В операторах могут происходить события четырех основных типов:

- 1) создание или уничтожение заявок;
- 2) изменение числового атрибута объекта;
- 3) задержка заявки на определенный период времени;
- 4) изменение маршрута движения заявки в модели.

Версия GPSS, реализованная в системе GPSS World, содержит 53 оператора, которые можно объединить в следующие группы.

*Операторы, осуществляющие модификацию атрибутов заявок:*

- генерирование и уничтожение заявок GENERATE (см. ЛР№3), SPLIT, TERMINATE (см. ЛР№3), ASSEMBLE;
- временная задержка ADVANCE (см. ЛР№3);
- синхронизация движения заявок MATCH и GATHER;
- изменение параметров заявки ASSIGN (см. ЛР№5), INDEX, MARK (см. ЛР№5), PLUS;
- изменение приоритета заявки PRIORITY (см. ЛР№5).

*Операторы, изменяющие последовательность движения заявок (операторы передачи управления):* DISPLACE (см. ЛР№6), TRANSFER (см. ЛР№4), LOOP, TEST (см. ЛР№5), GATE (см. ЛР№4).

*Операторы, описывающие объекты аппаратной категории:*

– одноканальные устройства (технические средства) SEIZE (см. ЛРН<sup>№3</sup>), RELEASE (см. ЛРН<sup>№3</sup>), PREEMPT (см. ЛРН<sup>№5</sup>), RETURN (см. ЛРН<sup>№5</sup>), FUNAVAIL (см. ЛРН<sup>№6</sup>), FAVAIL (см. ЛРН<sup>№6</sup>);

– многоканальные устройства (памяти) ENTER (см. ЛРН<sup>№4</sup>), LEAVE (см. ЛРН<sup>№4</sup>), SAVAIL (см. ЛРН<sup>№6</sup>), SUNAVAIL (см. ЛРН<sup>№6</sup>);

– ключи (логические переключатели) LOGIC.

*Операторы, сохраняющие значения, необходимые для дальнейшего использования:* SAVEVALUE (см. ЛРН<sup>№6</sup>), MSAVEVALUE.

*Операторы для получения статистических результатов:*

– очереди QUEUE (см. ЛРН<sup>№4</sup>), DEPART (см. ЛРН<sup>№4</sup>);

– таблицы TABULATE.

*Операторы для организации списка пользователя:* LINK, UNLINK.

*Специальные операторы:* COUNT, SELECT.

Перечень, форматы и операнды основных операторов GPSS World приведены в *Приложении 2*.

*Вычислительная категория* служит для описания связей между компонентами моделируемой системы в виде математических (аналитических и логических) соотношений. В качестве объектов вычислительной категории введены арифметические и булевы переменные и функции.

*Переменные* представляют собой сложные выражения, которые включают константы, стандартные числовые атрибуты (СЧА), библиотечные арифметические функции, арифметические и логические операции. Выражения могут применяться в переменных и операторах GPSS. При применении в переменных выражения определяются командами GPSS. При применении в операторах GPSS выражения определяются как часть встроенного языка PLUS. Использование переменных – см. ЛРН<sup>№6</sup>.

*Стандартные числовые атрибуты.* Каждому объекту соответствуют атрибуты, описывающие его состояние в данный момент времени. Они доступны для использования в течение всего процесса моделирования и называются *стандартными числовыми атрибутами (СЧА)*. Например, объект вычислительной категории – генератор случайных чисел – имеет СЧА R<sub>Nn</sub>, который соответствует числу, вычисляемому генератором равномерно распределенных случайных чисел номер n. У объекта динамической категории – заявки – несколько СЧА: P<sub>R</sub> – приоритет обрабатываемой в данный момент заявки; P<sub>i</sub> – значение i-го параметра активной заявки и др. Всего в GPSS World имеется свыше 50 СЧА (см. *Приложение 3*).

Имена СЧА зарезервированы. По способу определения имен СЧА можно разбить на три группы. Имя СЧА в *первой группе* состоит из двух

частей. Первая часть указывает групповое имя, идентифицирующее тип объекта и тип информации об объекте. Вторая часть идентифицирует конкретного члена группы. Групповое имя состоит из одной-двух букв, фиксированных для информации об объектах определенного типа, например, Q – ссылка на текущее значение длины очереди, FR – коэффициент использования устройства, SR – коэффициент использования накопителя и т. д. Объекты GPSS могут быть идентифицированы с помощью числовых или символьных имен. Идентификация СЧА зависит от вида адресации, которых в GPSS предусмотрено две: прямая и косвенная.

Рассмотрим прямую адресацию. Если объект идентифицирован с помощью номера, то ссылка на его СЧА записывается как СЧА<sub>j</sub>, где j – номер объекта (положительное целое число). Например, Q<sub>3</sub> – текущее значение длины очереди номер 3, FR<sub>2</sub> – коэффициент использования устройства номер 2. При указании имени объекта ссылка на его СЧА записывается как СЧА\$имя, где имя – имя объекта. Например, Q\$Basa – текущее значение длины очереди с именем Basa, SR\$Rem – коэффициент использования накопителя с именем Rem.

При косвенной адресации СЧА определяются как СЧА\*параметр. Это означает следующее:

- СЧА\*j, где j – номер параметра активной заявки (положительное целое число), содержащего номер нужного оператора;
- СЧА\*имя, где имя – имя параметра активной заявки, содержащего номер нужного оператора.

Например, Q\*3 – текущее значение длины очереди, номер которой является значением параметра 3 активной заявки, SR\*Rem – коэффициент использования памяти, номер которой содержится в параметре с именем Rem активной заявки.

*Вторую группу* имен СЧА составляют матричные СЧА. Матричный СЧА MX может содержать до трех идентификаторов (косвенных адресов). Например MX\*Result(\*Stroka,\*Stolbez) является ссылкой на матрицу, номер которой содержится в параметре активной заявки с именем Result, а элемент матрицы – номер строки и номер столбца – определяется значениями параметров с именами Stroka и Stolbez соответственно той же активной заявки.

*Третью группу* составляют следующие СЧА: A1, AC1, C1, M1, PR, TG1, XN1 и Z1. Эти СЧА, в отличие от СЧА первой и второй групп, не требуют указания номера или имени объекта.

СЧА по признаку доступности к ним пользователя можно разделить на две категории. *Первую категорию* составляют атрибуты, которые в части, касающейся их формирования, доступны только GPSS World, однако они могут быть использованы также и пользователем в выражениях, но не могут им изменяться в процессе моделирования. Эта группа содержит наибольшее число СЧА. Во *вторую категорию* входят СЧА вычислительных объектов, которые могут изменяться разработчиком модели и использоваться им в выражениях. К ним относятся:

- FNj (FN\$Raspr) – вычисленное вещественное значение функции номер j (с именем Raspr);
- Vj (V\$Ver) – вычисленное значение вещественной или целочисленной переменной номер j (с именем Ver);
- BVj (BV\$Per) – вычисленное вещественное значение булевой переменной номер j (с именем Per).

*Булевы переменные* позволяют пользователю проверять в одном операторе GPSS одновременно несколько условий, исходя из состояния или значения других переменных, сохраняемых значений и/или атрибутов.

С помощью *функций* пользователь может производить вычисления непрерывных или дискретных функциональных зависимостей между аргументом функции (независимая величина) и зависимым значением функции. Все функции в GPSS задаются табличным способом с помощью команд описания функций. Как и переменные, функции не связаны с определенными операторами. Кроме библиотечных арифметических функций GPSS World имеет 24 встроенных генератора случайных чисел.

*Объекты запоминающей категории* обеспечивают обращения к сохраняемым значениям. *Ячейки* сохраняемых величин и *матрицы ячеек* сохраняемых величин используются для сохранения числовой информации. Например, значение, занесенное в ячейку, может представлять собой коэффициент использования устройства в какой-то момент времени. Любая активная заявка может инициировать запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считать любая другая заявка. Примеры использования ячеек – см. ЛР№7.

*Матрицы* могут иметь до шести измерений. Матрица должна быть предварительно описана. Команда описания матрицы имеет формат:

Name MATRIX A,B,C,[D],[E],[F],[G]

Метка Name определяет имя матрицы и должна быть именем (не может быть числом). Операнд A не используется, он оставлен для совместимости с более старыми версиями GPSS. Операнды B, C, D, E, F, G могут

быть только целыми положительными числами. Операндом В задается число строк, операндом С – число столбцов матрицы. Операнды D, E, F, G задают максимальное количество элементов в третьем, четвертом, пятом и шестом измерениях соответственно. Например, команда

TraSr MATRIX ,5,7

определяет матрицу с именем TraSr, содержащую 5 строк и 7 столбцов. Матрицы имеют единственный СЧА с названием MX, с помощью которого можно обращаться к любому элементу матрицы. В качестве значений строк и столбцов могут использоваться только имена, целые числа и параметры заявок. Например:

– MX6(13,5) – определяется элемент матрицы номер 6, находящийся на пересечении 13-й строки и 6-го столбца;

– MX\$Tab2(P4,P\$Nal) – для определения элемента матрицы с именем Tab2 используются значения параметра номер 4 и параметра заявки с именем Nal;

– MX\$Spis(Usor,P3) – для определения элемента матрицы с именем Spis используются имя переменной пользователя и параметр заявки номер три (переменной пользователя предварительно должно быть присвоено целое положительное значение командой EQU).

Следует обратить внимание на то, что в первом примере матрица имеет не имя, а номер, хотя в команде определения матрицы в качестве метки Name может быть использовано только имя. Для присвоения матрице номера вместо имени нужно до записи команды определения матрицы использовать команду EQU. Например:

Ski EQU 6

Ski MATRIX ,10,7

В данном примере имя Ski матрицы заменено на номер 6, по которому MX6 теперь и нужно обращаться к матрице. Если в команде MATRIX вместо имени Ski сразу записать число 6, то при трансляции возникнет останов по ошибке «Пропущена метка оператора».

Перед началом моделирования или выполнения команды CLEAR ON значения всех элементов матрицы устанавливаются равными нулю. Некоторым или всем элементам по желанию разработчика могут быть присвоены ненулевые значения или они могут быть переведены в неопределенное состояние командой INITIAL. Например:

– INITIAL MX5(2,6),-648.237 – элементу матрицы номер 5, находящемуся на пересечении второй строки и шестого столбца, присваивается начальное значение -648.237;

- INITIAL MX\$Ntf(4,7),48 – элементу (4,7) матрицы с именем Ntf присваивается начальное значение 48;
- INITIAL MX\$Avt(13,24),Koef – элементу (13,24) матрицы с именем Avt присваивается начальное значение переменной пользователя Koef;
- INITIAL Gros,48 – вместо операнда указано имя матрицы Gros, поэтому всем элементам этой матрицы присваивается значение 48;
- INITIAL Pult – вместо операнда А стоит имя матрицы Pult, но операнд В не используется, поэтому всем элементам матрицы присваивается значение 1;
- INITIAL MainResult,UNSPECIFIED – вместо операнда В записано ключевое слово UNSPECIFIED, поэтому все элементы предварительно созданной матрицы MainResult будут переведены в неопределенное состояние. Этим переводом матрица подготовлена для использования в эксперименте, в котором могут быть пропущенные, т. е. не полученные в ходе эксперимента данные. Неопределенное состояние элемента матрицы указывает на то, что данных нет. Если же элементы матрицы были бы переведены в нуль, то это бы воспринималось как получение данных эксперимента, равных нулю, что неверно.

Для записи в процессе моделирования значений в матрицы, а также для увеличения или уменьшения значений, записанных в матрицы, служит оператор MSAVEVALUE. Формат его записи следующий:

MSAVEVALUE A,B,C,D

В операнде А задается имя или номер матрицы. Крайним правым символом операнда может быть знак плюс (режим накопления) или знак минус (режим вычитания). В операнде В задается номер строки, в операнде С – номер столбца матрицы. Операнд D определяет значение, которое должно храниться, прибавляться или вычитаться. Все операнды являются обязательными. Они могут быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Например:

- MSAVEVALUE 2,13,P21,782.34 – при входе заявки в оператор MSAVEVALUE значение элемента матрицы номер 2, расположенного на пересечении строки 13 и столбца, номер которого содержится в параметре 21 вошедшей заявки, замещается числом 782.34;

- MSAVEVALUE X\$Bba-,8,23,V\$Hit – значение элемента на пересечении строки 8 и столбца 23 матрицы, номер которой содержится в сохраняемой ячейке с именем Bba, уменьшается на вычисленное значение арифметической переменной с именем Hit;

- MSAVEVALUE Rta+,11,(User#2+3),M1 – номер столбца находится как результат вычисления выражения в скобках (User#2+3), а затем значение определенного таким образом элемента матрицы Rta увеличивается на величину относительного времени M1 пребывания заявки в модели;

- MSAVEVALUE (Rra+Q\$Rem)+,(Rra-5),12,P\$Nht – номер матрицы и номер строки элемента определяются как результаты вычисления выражений в скобках (Rra+Q\$Rem) и (Rra-5) соответственно. После этого значение найденного элемента увеличивается на величину, содержащуюся в параметре с именем Nht активной заявки.

- MSAVEVALUE 4,Stroka,Stolbez,"Dlina Mars" – элементу матрицы номер 4, находящемуся на пересечении строки и столбца, определяемых переменными пользователя Stroka и Stolbez, присваивается строка "Dlina Mars". Если переменным пользователя предварительно командой EQU не будут присвоены соответствующие значения, то возникнет останов по ошибке.

Останов по ошибке возникает и в том случае, если при входе заявки в оператор SAVEVALUE не будет найдена матрица с нужным именем или номером, или не будет найден элемент с соответствующими номерами строки и столбца. СЧА MX в операторе MSAVEVALUE при указании операнда A, как и СЧА X в операторе SAVEVALUE при указании операнда A, не используется. Ссылка на этот СЧА применяется также в выражениях, функциях, операторе INITIAL.

Из рассмотренного выше формата команды описания матрицы следует, что матрица GPSS может иметь до шести измерений. Однако, как видно из формата оператора MSAVEVALUE, он предоставляет возможность использовать только двумерные матрицы, т. е. в нем доступны только первые два измерения. Остальные индексы равны 1. Если нужно использовать матрицы более чем двух измерений, для этого следует создать одну или более PLUS-процедур. PLUS-процедуры имеют доступ ко всем элементам всех матриц. Матрицы, определенные командой MATRIX, являются глобальными и доступны всем PLUS-процедурам. На время выполнения PLUS-процедуры могут быть созданы временные матрицы с локальной областью видимости.

*Статистические объекты.* К статистическим объектам относятся очереди и таблицы. В любой системе движение потока заявок может быть задержано из-за недоступности устройств. В этом случае задержанные заявки ставятся в. Учет этих очередей составляет одну из основных функций планировщика, который автоматически накапливает определенную статистику.

стику относительно устройств и очередей. Кроме этого пользователь может собирать дополнительную статистическую информацию, указав специальные точки в модели.

Для облегчения табулирования статистической информации в GPSS предусмотрен специальный объект – таблица. Таблицы используются для получения выборочных распределений некоторых случайных величин. Таблица состоит из частотных классов (диапазонов значений), куда заносится число попаданий конкретного числового атрибута в каждый, тот или иной, частотный класс. Для каждой таблицы вычисляется также математическое ожидание и среднеквадратическое отклонение. В конце эксперимента с моделью результаты, содержащиеся в таблицах, могут выводиться либо на дисплей, либо на печать.

*Объекты группирующей категории.* К группирующей категории относятся три типа объектов: числовая группа, группа заявок и списки.

Заявки представляют собой объекты, характеризующиеся общими СЧА, которые обычно представляются как параметры заявок и меняются при прохождении заявок через оператор ASSIGN. Это неудобно в том случае, когда необходимо одновременно менять некоторые атрибуты всех заявок некоторого множества. Объекты группирующей категории предоставляют возможность пользователю обращаться к атрибутам заявок заданной группы. Они работают в одном из режимов: в режиме заявки или числом. Режим определяется первым обращением к определенной группе. После того как он установлен, необходимо его поддерживать. В режиме заявки величины, поступающие в группу, представляются номерами по очереди заявок данной группы. В другом режиме поступающие величины представляются группой числовых значений.

При моделировании заявки хранятся в списках. Существует пять видов списков, только в одном из которых в любой момент времени может находиться заявка: 1) текущих событий; 2) будущих событий; 3) задержки обслуживающих устройств (ОУ); 4) отложенных прерываний ОУ; 5) пользователя.

В *списке текущих событий* (СТС) находятся заявки, соответствующие событиям, время наступления которых меньше (заявки, которые должны были начать продвижение по модели в некоторый момент в прошлом, но были заблокированы) или равно текущему времени. Заявки из СТС готовы к входу в операторы и должны войти в них до очередного изменения модельного времени. В СТС заявки расположены в порядке убы-

вания приоритета. Заявки с одинаковыми приоритетами располагаются в порядке поступления их в список.

*Список будущих событий* (СБС) содержит заявки, соответствующие событиям, время наступления которых больше текущего времени, т. е. событиям, которые должны произойти в будущем. Такие заявки размещаются в списке строго в порядке возрастания времени начала движения. Приоритеты не влияют на порядок движения в этом списке.

В *список повторных попыток* помещаются заявки, для которых не выполнены условия входа в следующий оператор. Эти условия проверяются при попытке входа заявки в операторы GATE (см. ЛР№4), TEST (см. ЛР№5), TRANSFER ALL и TRANSFER BOTH (см. ЛР№4). Заявки из списка повторных попыток ожидают изменения СЧА. Когда СЧА изменяется, заявка активируется, проверяются условия ее входа в следующий оператор. Если условия выполняются, заявка входит в следующий оператор и автоматически исключается из списка повторных попыток.

Обслуживающее устройство имеет следующие списки:

- список отложенных прерываний – список заявок, ожидающих занятия ОУ по приоритету;
- список прерываний – список заявок, обслуживание которых данным ОУ было прервано;
- список задержки – список заявок, ожидающих занятия ОУ в порядке приоритета;
- список повторных попыток – список заявок, ожидающих изменения состояния ОУ.

Накопитель, моделирующий ограниченную очередь или память, имеет следующие списки:

- список задержки – список заявок в порядке приоритета, ожидающих возможность занять освободившиеся места в накопителе;
- список повторных попыток – список заявок, ожидающих изменения состояния накопителя.

*Список синхронизируемых заявок* имеют операторы ASSEMBLE, GATHER, MATH. Он содержит заявки, находящиеся в данный момент времени в состоянии сравнения.

*Список пользователя* содержит заявки, удаленные пользователем из списка текущих событий и помещенные в список пользователя как временно неактивные. Списки пользователя используются для организации очередей с дисциплинами, отличными от FIFO.

Арифметические, условные и логические операторы, используемые в моделях GPSS World, представлены в таблице 38.

Таблица 38

Оператор	Пример использования	Описание
^	$A^B$	Возведение в степень.
#	$A\#B$	Умножение (оператор «#» использован для операции умножения, т. к. оператор «*» используется в косвенной адресации. Пользователь может менять значения операторов «#» и «*» в журнале настроек Settings на закладке Simulation).
/	$A/B$	Деление.
\	$A\B$	Целочисленное деление. Возвращает результат целочисленного деления A на B.
@	$A@B$	Целочисленный остаток. Возвращает целочисленный остаток от деления A на B.
-	$A-B$	Вычитание.
+	$A+B$	Сложение.
$\geq$ или 'GE'	$A\geq B$	Возвращается 1, если A в цифровой форме больше или равно B, в противном случае возвращается 0.
$\leq$ или '@'	$A\leq B$	Возвращается 1, если в цифровой форме A меньше или равно B, в противном случае возвращается 0.
$>$ или 'G'	$A>B$	Возвращается 1, если в цифровой форме A больше B, в противном случае возвращается 0.
$<$ или 'L'	$A<B$	Возвращается 1, если в цифровой форме A меньше B, в противном случае возвращается 0.
= или 'E'	$A=B$	Возвращается 1, если в цифровой форме A равно B, в противном случае возвращается 0.
$\neq$ или 'NE'	$A\neq B$	Возвращается 1, если в цифровой форме A отличается от B, в противном случае возвращается 0.
& или 'AND'	$A\&B$	Возвращается 1, если A и B не равны нулю, в противном случае возвращается 0.
или 'OR'	$A B$	Возвращается 1, если A, или B, или оба не равны нулю, в противном случае возвращается 0.
INT(A)	INT(A)	Возвращает целую часть числа A.

## **Операторы модификации атрибутов и параметров заявок**

*ADOPT* – определение или смена семейства активной заявки. При создании заявки оператором GENERATE параметру семейства присваивается исходное значение, равное *порядковому номеру заявки*. Если создаются копии заявки (оператор SPLIT), то все они принадлежат к семейству порождающей заявки.

При входе заявки в оператор ADOPT происходит присоединение заявки к семейству, номер или имя которого определено операндом A:

ADOPT A

Операнд A быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА\*параметр. Например, ADOPT 2000 – все активные заявки, прошедшие оператор, становятся членами семейства с номером 2000 (значение присваивается параметру Assembly Setvalue заявки).

*SPLIT* – создание копий заявок. Оператор GENERATE является основным средством создания заявок и ввода их в модель. Вход заявок в оператор GENERATE не допускается. Кроме оператора GENERATE для создания заявок используется оператор SPLIT (расщепить). Но, в отличие от GENERATE, оператор SPLIT не создает самостоятельных заявок, а лишь генерирует заданное число копий входящего в него заявки – порождающей. Оператор SPLIT имеет следующий формат:

SPLIT A,[B],[C]

Операнд A – число создаваемых копий одного семейства. Он может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА\*параметр. Все копии формируются в момент входа порождающей заявки в оператор SPLIT. Операнд B – номер оператора, к которому переходят копии порождающей заявки. Значение операнда B вычисляется для каждой копии отдельно.

После прохождения оператора SPLIT порождающая заявка направляется в следующий оператор, а все копии направляются по адресу, указанному операндом B (или к следующему оператору, если операнд B не указан – по умолчанию). Таким образом, если операнд A равен  $k$ , то из оператора выйдут  $k+1$  заявка. Далее порождающая заявка и копии являются равноправными и при необходимости могут проходить снова через любое число операторов SPLIT.

Операндом C может быть задан номер параметра, используемого для присвоения копиям последовательных номеров (нумерации). Если операнд C не используется, по умолчанию номеров выходящих из оператора SPLIT заявок в их параметрах нет.

Операнды В и С могут быть такими же, как и операнд А. Например:

– SPLIT 6 – из оператора SPLIT выйдет  $1 + 6 = 7$  заявок, шесть копий вслед за порождающей заявкой будут направлены к следующему оператору, так как операнд В не используется. Записи номеров в параметр порождающей и копий заявок также не будет – операнд С опущен. Копии имеют тот же приоритет, значения параметров и время входа в модель, что и порождающая заявка.

– SPLIT 9, Met11 – оператор SPLIT сгенерирует  $1 + 9 = 10$  заявок. Порождающая заявка перейдет к следующему оператору, а девять копий – к оператору с меткой Met11. Записи номеров в параметры заявок также нет.

– SPLIT 4, Met1,5 – оператор SPLIT при входе порождающей заявки генерирует четыре копии, которые направляются к оператору с меткой Met1. Каждая копия имеет тот же приоритет, время входа в модель и значения параметров, что и порождающая заявка, за исключением параметра номер 5. В параметр номер 5 каждой копии будет записан порядковый номер. Если параметр номер 5 порождающей заявки не определен заранее, он будет создан, и ему будет присвоено значение 0. Если параметр номер 5 порождающей заявки определен заранее и его значение равно  $n$ , то в параметре номер 5 порождающей заявки и копий после выхода из блока SPLIT будут записаны  $n+1$ ,  $n+2$ ,  $n+3$ ,  $n+4$  и  $n+5$  соответственно.

– SPLIT 3, P\$Adr, Adr – оператор, к которому должны направляться копии заявок, указывается операндом В, значение которого, т. е. номер оператора, вычисляется для каждой вновь создаваемой копии. Если параметр с именем Adr содержит номер оператора, например  $n$ , то первая копия будет направлена к оператору  $n+2$ , вторая – к  $n+3$ , третья – к  $n+4$ . Эти же номера будут записаны в параметр с именем Adr копий порождающей заявки. Сама же порождающая заявка со значением  $n+1$  параметра с именем Adr перейдет к следующему оператору.

Каждая новая копия, создаваемая оператором SPLIT, становится членом семейства заявок, порожденных исходной заявкой оператора GENERATE. Заявки, принадлежащие к одному семейству, объединяются планировщиком в список. Если оператором SPLIT не производится нумерацию порождающей заявки и ее копий, то внутри семейства нельзя установить, какая из заявок является порождающей. Число заявок в семействе произвольно. Каждая заявка, созданная оператором GENERATE, является отдельным семейством (номер заявки равен номеру семейства, т. е. сколько сгенерировано оператором GENERATE в модели заявок, столько и семейств). Таким образом, число семейств (ансамблей) в системе произволь-

но и семейство существует до тех пор, пока в нем находится хотя бы одна заявка. К семейству заявок можно применять операторы MATCH, ASSEMBLE, GATHER.

*ASSEMBLE* – объединение заявок. Оператор ASSEMBLE (объединить) используется для объединения заданного числа заявок, принадлежащих одному семейству, в одну заявку. Из оператора ASSEMBLE выходит только одна заявка, который переходит в следующий оператор. Оператор ASSEMBLE имеет следующий формат:

#### ASSEMBLE A

Операнд A это число заявок одного семейства, участвующих в сборке (начальное число счетчика сборки). Может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА\*параметр. В одном и том же операторе ASSEMBLE возможна одновременная сборка заявок нескольких семейств. Так как сборка длится в течение какого-то промежутка модельного времени, оператор ASSEMBLE имеет список синхронизации. В список синхронизации помещаются первые вошедшие в оператор ASSEMBLE заявки каждого собираемого семейства. Они ожидают прихода заявок своих семейств.

При входе заявки в оператор ASSEMBLE проверяется: есть ли в списке синхронизации заявка такого же семейства. Если заявки такого же семейства в списке синхронизации нет, т. е. вошедшая заявка первая из собираемого семейства, то вычисляется операнд A, округляется до целого числа и уменьшается на 1. В зависимости от полученного результата S, сохраняемого в ячейке (счетчике сборки) первой заявки собираемого семейства, выполняются следующие действия:

- если  $S < 0$ , то происходит останов по ошибке «Счетчик сборки не положителен»;
- если  $S = 0$ , то заявка сразу пытается войти в следующий оператор;
- если  $S > 1$ , то заявка исключается из списка текущих событий и помещается в список синхронизации оператора ASSEMBLE.

Если в списке синхронизации уже есть заявка того же семейства, что и вошедшая в оператор ASSEMBLE, то вошедшая заявка уничтожается, а счетчик сборки, сохраненный в ячейке находящейся в списке синхронизации заявки, уменьшается на 1. Когда этот счетчик сборки становится равным нулю, т. е. в оператор ASSEMBLE вошло заданное число заявок, ожидающая заявка выводится из списка синхронизации. Если обслуживание этой заявки не было прервано ни одним из устройств, она пы-

тается перейти к следующему оператору и в случае успеха возвращается в список текущих событий. Например:

– ASSEMBLE 7 – собирает 7 заявок одного семейства, 6 уничтожаются, одна переходит в следующий оператор.

– ASSEMBLE \*1 – собирает число заявок, равное значению параметра 1 первого пришедшего в оператор ASSEMBLE заявки семейства. Этим способом задания счетчика сборки удобно пользоваться, когда неизвестно заранее количество собираемых заявок, и оно определяется в ходе моделирования. В этом случае нужно в параметр номер один всех собираемых заявок записать значение счетчика сборки, так как неизвестно, какая заявка придет первой и установит счетчик сборки.

Прерванной заявке запрещается покидать оператор ASSEMBLE до тех пор, пока все его прерванные обслуживания не будут закончены. Поэтому рекомендуется строить модель так, чтобы заявки, обслуживание которых было прервано без удаления (без освобождения устройства), не входили в оператор ASSEMBLE.

*Пример совместного использования операторов SPLIT и ASSEMBLE.* На пункт управления, состоящий из 4-х автоматизированных рабочих мест (АРМ), поступают группы сообщений. Интервалы времени поступления групп сообщений подчинены экспоненциальному закону со средним значением 5 часов. Количество сообщений в группе распределено по нормальному закону с математическим ожиданием 20 сообщений и стандартным отклонением 3 сообщения. Время обработки одного сообщения на одном АРМ распределяется по экспоненциальному закону со средним значением 25 мин. После обработки сообщения вновь собираются и хранятся вместе. Ограничений на количество обрабатываемых сообщений нет.

Для имитации группы сообщений и одного сообщения следует использовать заявки. АРМ в модели будет представлять многоканальное устройство (МКУ), заданное накопителем (см. ЛРН<sup>№4</sup>).

Общая идея построения модели заключается в следующем. Оператор GENERATE имитирует поступление заявок. Одна заявка – одна группа сообщений. Затем случайным образом определяется количество сообщений в группе. Далее поступившая группа сообщений – исходная заявка входного потока копируется оператором SPLIT и результат копирования – число заявок по числу сообщений в группе, поступают на АРМ – МКУ и обрабатываются. После обработки сообщения, входящие в группу, собираются оператором ASSEMBLE. Когда все заявки-сообщения будут обработаны и

поступят в оператор ASSEMBLE, из него выйдет одна заявка, имитирующая снова собранную группу сообщений. Реализация в GPSS имеет вид, приведенный в примере 1.

### *Пример 1*

#### *; Задание исходных данных*

```
Tpt EQU 300 ; Средний интервал поступления групп сообщений
Moka EQU 30 ; Матожидание числа сообщений в группе
Soka EQU 5 ; Стандартное отклонение числа сообщений в группе
Troa EQU 25 ; Среднее время обработки одного сообщения
VrMod EQU 4320 ; Время моделирования
PObr STORAGE 4 ; Число каналов МКУ
```

#### *; Сегмент имитации поступления и обработки сообщений*

```
GENERATE (Exponential(13,0,Tpt)) ; Источник групп сообщений
ASSIGN 1,(Normal(451,Moka,Soka)) ; Определение числа сообщений в
; группе

SPLIT *1 ; Деление группы на сообщения
QUEUE Rr1 ; Занять очередь
ENTER PObr ; Занять канал МКУ
DEPART Rr1 ; Покинуть очередь
ADVANCE (Exponential(13,0,Troa)) ; Обработка сообщения
LEAVE PObr ; Освободить один канал МКУ
ASSEMBLE *1 ; Собрать сообщения в группу
TERMINATE ; Выход обработанных групп
; сообщений
```

#### *; Сегмент задания времени моделирования*

```
GENERATE VrMod
TERMINATE 1
```

В операторе ASSIGN случайным образом определяется количество сообщений в поступившей группе и запоминается в первом параметре заявки. Сделано это потому, что в операторе ASSEMBLE использована запись ASSEMBLE \*1, значит, он собирает число заявок, равное значению параметра 1 первой пришедшего в оператор ASSEMBLE заявки семейства, а какая именно заявка придет первым после обработки на МКУ PP1 – исходная или копия – неизвестно.

*GATHER* – пакетирование заявок. Оператор GATHER (собирать) предназначен для сбора заданного количества заявок одного семейства, движущихся по одному и тому же пути. Отличие оператора GATHER от оператора ASSEMBLE состоит в том, что в нем заявки не уничтожаются

(не «сливаются» в одну), а после сбора все направляются к следующему оператору. Формат оператора:

### GATHER A

Операнд A задает число заявок, принадлежащих одному семейству, которое нужно собрать при их движении по одному пути (начальное число счетчика сбора). Может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА\*параметр.

Оператор GATHER также может одновременно производить сбор заявок нескольких семейств и также имеет список синхронизации.

Пусть при входе заявки в оператор GATHER в результате проверки оказалось, что в его списке синхронизации нет заявок этого же семейства, т. е. вошедшая заявка из данного семейства является первой. Тогда вычисляется операнд A, округляется до целого числа, уменьшается на 1 и запоминается полученный результат S в ячейке заявки – счетчике сбора. В зависимости от результата S возможны следующие действия:

- если  $S < 0$ , то происходит останов программы по ошибке «счетчик сбора не положителен»;
- если  $S = 0$ , то заявка сразу пытается войти в следующий оператор;
- если  $S > 1$ , то заявка помещается в список синхронизации оператора для ожидания прибытия других заявок своего семейства.

Если проверкой в списке синхронизации обнаружены заявки семейства, к которому принадлежит только что вошедшая в оператор GATHER заявка, то она помещается в список синхронизации, а счетчик сбора уменьшается на 1. Если его значение становится равным нулю, т. е. в оператор GATHER вошло заданное число заявок данного семейства, все собранные заявки исключаются из списка синхронизации и помещаются в список текущих событий.

Если среди собранных заявок одного семейства обнаруживаются прерванные заявки, то после окончания сбора они не переводятся в СТС. Прерванные заявки будут находиться в операторе GATHER, но не в списке синхронизации, до тех пор, пока все прерванные обслуживания не будут закончены. Поэтому в модели нужно предусмотреть, чтобы прерванные без удаления, т. е. без освобождения устройства, заявки, если в этом нет необходимости, не входили в оператор GATHER.

Дополнительный пример, демонстрирующий различие в использовании операторов GATHER и ASSEMBLE в режиме пошагового прогона, приведен в Приложении 4.

*MATCH* – синхронизация движения заявок. Оператор MATCH (синхронизировать) предназначен для синхронизации движения заявок одного

семейства, продвигающихся по разным путям. Для синхронизации необходимы два оператора MATCH, находящихся в соответствующих местах модели и называемых сопряженными. Формат записи:

Name MATCH A

Операндом A каждого оператора MATCH указывается метка или номер сопряженного ему оператора. Операнд может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Например:

Kont1 MATCH Kont2

Kont2 MATCH Kont1

В модели эти операторы будут помещены отдельно в параллельных путях продвижения заявок. При входе заявки в оператор MATCH операнд A вычисляется и округляется до целого числа. По полученному значению определяется сопряженный оператор MATCH. При его отсутствии происходит останов по ошибке.

Если сопряженный оператор есть, проверяется наличие в нем заявки из того же семейства. Если в операторе нет ни одной заявки данного семейства, поступившая заявка помещается в список синхронизации и будет ожидать в нем входа заявки своего семейства в сопряженный оператор MATCH. При поступлении такой заявки в сопряженный оператор MATCH обе заявки одного семейства будут исключены из списка синхронизации и одновременно будут пропущены в следующие за оператором MATCH операторы. Одна и та же пара операторов MATCH может одновременно синхронизировать любое число пар заявок из разных семейств. Заявки одного семейства также можно синхронизировать в любом числе пар операторов MATCH. Если одна из синхронизируемых заявок будучи прерванной вошла в оператор MATCH, ей не разрешается выйти из него до тех пор, пока все его прерванные обслуживания не будут закончены. Поэтому строить модель нужно так, чтобы заявки, обслуживание которых прервано без удаления (без освобождения устройств), не входили в операторы MATCH.

Оператор MATCH может быть сопряжен сам себе (в этом случае его действие будет аналогично действию оператора GATHER с операндом A, равным 2), например:

Six MATCH Six

*Пример применения оператора MATCH.* Техническое обслуживание (ТО) средств связи (СС) проводят в два этапа две группы специалистов. Первый этап выполняется специалистами первой и второй групп параллельно со средним временем соответственно 80 и 70 минут. После первого этапа производится уточнение мероприятий ТО, время выполнения которого принимается равным нулю. После уточнения специалистами первой и второй групп параллельно выполняется второй этап со средним временем 30 и 40 минут соответственно. Затем третья группа специалистов производит комплексную проверку СС со средним временем 60 минут. Интервалы поступления СС и время их ТО подчиняются экспоненциальному закону. В модели группы специалистов целесообразно представить ОУ, а СС – заявками. Пусть первой группе специалистов соответствует ОУ  $Gto1$ , второй группе –  $Gto2$ , третьей группе –  $Gto3$ . реализация модели показана в примере 2.

### *Пример 2*

#### ***; Задание исходных данных***

Post	EQU	11400	; Интервал поступления СС на ТО
Obs2	EQU	4800	; Среднее время ТО первой группой на этапе 1
Obs3	EQU	4200	; Среднее время ТО второй группой на этапе 1
Obs4	EQU	1800	; Среднее время ТО первой группой на этапе 2
Obs5	EQU	2400	; Среднее время ТО второй группой на этапе 2
Obs6	EQU	3600	; Среднее время ТО третьей группой на этапе 3

#### ***; Сегмент имитации поступления средств связи на ТО***

GENERATE (Exponential(215,0,Post)) ; Источник средств связи  
 SPLIT 1, Anb ; Разделение на две заявки

#### ***; Сегмент имитации работы первой группы специалистов***

SEIZE Tg1 ; Занять первую группу специалистов  
 ADVANCE (Exponential(215, 0, Obs2)) ; Первый этап обслуживания  
 Par1 MATCH Par2 ; Уточнение мероприятий ТО  
 ADVANCE (Exponential(215,0,Obs4)) ; Второй этап обслуживания  
 RELEASE Tg1 ; Освободить первую группу  
 TRANSFER ,Ank

#### ***; Сегмент имитации работы второй группы специалистов***

Anb SEIZE Tg2 ; Занять вторую группу специалистов  
 ADVANCE (Exponential(215,0,Obs3)) ; Первый этап обслуживания  
 Par2 MATCH Par1 ; Уточнение мероприятий ТО  
 ADVANCE (Exponential(215,0,Obs5)) ; Второй этап обслуживания  
 RELEASE Tg2 ; Освободить вторую группу  
 ; специалистов

Ank	ASSEMBLE 2	; Передача СС третьей группе
; Сегмент имитации работы третьей группы специалистов		
SEIZE	Tg3	; Занять третью группу специалистов
ADVANCE	(Exponential(215,0,0bs6))	; Третий этап обслуживания
RELEASE	Tg3	; Освободить третью группу
TERMINATE	1	; Готовые СС

В примере 2 после поступления в модель исходной заявки – одного образца средства связи – осуществляется расщепление ее с помощью оператора SPLIT на две заявки. Первая из них направляется на цепочку операторов, имитирующих работу специалистов первой группы, а вторая (копия) – на операторы, моделирующие работу специалистов второй группы. После выполнения первого этапа обслуживания специалистами обеих групп обе заявки задерживаются операторами MATCH и одновременно начинается второй этап работ. После его выполнения две заявки оператором ASSEMBLE объединяются в один и далее с использованием одной заявки имитируется проведение комплексной проверки.

*INDEX* – изменение параметра заявки. Изменить значение параметра заявки можно оператором INDEX следующего формата:

INDEX A,B

Операнд А – номер параметра заявки, он может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Операнд В – числовое значение, которое прибавляется к значению параметра, указанного операндом А. Может быть именем, числом, выражением в скобках, СЧА или СЧА\*параметр. Полученный результат заносится в первый параметр заявки.

Так, например, INDEX Slag1,(Normal(21,Mat,SreOtk)+X\$post) – при входе заявки в такой оператор вычисляется выражение в скобках, которым указан операнд В, и складывается со значением параметра с именем Slag1. Результат сложения присваивается параметру номер один заявки. Если этот параметр не существует, он создается. Если же нет параметра, указанного операндом А, происходит останов по ошибке «Обращение к несуществующему параметру».

*PLUS*. В GPSS World значения параметров заявок можно также изменять оператором PLUS. Оператор PLUS вычисляет выражение и может записать его в параметр. Оператор PLUS имеет следующий формат:

PLUS A,[B]

Операнд А – выражение. Может быть именем, числом, строкой, выражением в скобках, СЧА или СЧА\*параметр. Операнд В – номер параметра заявки, в котором сохраняется результат. Операнд может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Так, например, PLUS (Exponential(23,0,32.7)+48.75),Itog – при входе заявки в такой оператор PLUS вычисляется выражение в скобках, которым задан операнд А. Затем полученный результат присваивается параметру с именем Itog. Если такой параметр не существует, он создается.

### **Операторы изменения последовательности движения заявок**

К операторам, изменяющим последовательность продвижения заявок по системе, относятся операторы: DISPLACE (см. ЛРН№6), TRANSFER (см. ЛРН№4), TEST (см. ЛРН№5), GATE (см. ЛРН№4) и LOOP.

*LOOP – организация циклов.* Оператор LOOP предназначен для организации в модели циклов. Он имеет следующий формат:

LOOP A,B

Операнд А – параметр заявки или параметр цикла, в котором содержится число повторений какого-либо участка модели. Операнд В – метка оператора, с которого начинается цикл. Операнды А и В могут быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр, например: LOOP KolPovt,Start1 – оператор с меткой Start1 является началом цикла, т. е. должен быть расположен раньше оператора LOOP. Когда заявка, пройдя участок модели, начинающийся оператором с меткой Start1, войдет в оператор LOOP, значение ее параметра с именем KolPovt уменьшается на 1. Если это значение не равно нулю, заявка переходит к оператору с меткой Start1, т. е. цикл повторяется. Если же после вычитания 1 значение параметра цикла равно нулю, следовательно, выполнено заданное число повторений, заявка переходит к следующему оператору. Если при входе заявки в оператор LOOP окажется, что параметра с именем KolPovt нет, произойдет останов по ошибке «обращение к несуществующему параметру».

При каждом входе заявки в оператор LOOP значение параметра цикла уменьшается на 1. Значит, значение параметра цикла можно использовать для выбора новых значений каких-либо факторов при очередном повторении. Однако это не всегда удобно, так как величина параметра цикла изменяется от большего к меньшему значению, а не наоборот. Функции,

выполняемые оператором LOOP, могут быть реализованы, например, с использованием оператора TEST следующим образом:

```

...
    ASSIGN  2,1
Met1  ...
...
    ASSIGN  2+,1
    TEST G  P2,10,Met1

```

Первым оператором ASSIGN в параметр 2 активной заявки заносится 1. Оператор с меткой Met1 является началом цикла. После выполнения последовательности операторов, составляющих тело цикла, второй оператор ASSIGN увеличивает значение параметра 2 на 1. Оператор TEST проверяет значение параметра 2. Если оно меньше десяти, заявка направляется к оператору с меткой Met1 и вновь повторяется цикл, если больше – заявка проходит к следующему по порядку оператору. Значения параметра 2 изменяются в порядке возрастания.

Рассмотрим использование оператора LOOP для моделирования одноканальной СМО с ожиданием и ограниченной очередью на три заявки. Пусть требуется в любой момент времени знать номера заявок, находящихся в очереди и на обслуживании. Реализация такой модели показана в примере 3.

### *Пример 3*

***; Задание длины очереди и булевой переменной***

Emk EQU 3

Kont1 BARIABLE (Q\$Dlina<Emk) ; Проверка длины очереди

***; Сегмент имитации поступления и обслуживания заявок***

GENERATE 2,,7 ; Источник заявок

TEST E BV\$Kont1,1,Met1 ; Есть ли место в очереди?

QUEUE Dlina ; Да, встать в очередь

***; Учет номеров заявок, вошедших в очередь***

ASSIGN 1, Emk ; Запись в параметр цикла

Met2 TEST E X\*1,0,Met3 ; Есть ли свободное место?

SAVEVALUE P1,XN1 ; Да, записать номер заявки

TRANSFER ,Met4 ; Выйти из цикла

Met3 LOOP 1,Met2 ; Повторить или конец цикла

Met4 SEIZE Stan ; Занять устройство

DEPART Dlina ; Покинуть очередь

***; Учет номеров заявок, покинувших очередь***

ASSIGN 1, Emk ; Запись в параметр цикла

Met5 TEST E X\*1,XN1,Met6 ; *Есть ли заявка с таким номером?*  
SAVEVALUE P1,0 ; *Да, тогда удалить*  
TRANSFER ,Met7 ; *Выйти из цикла*  
Met6 LOOP 1,Met5 ; *Повторить или конец цикла*  
; **Учет номера заявки, находящейся на обслуживании**  
Met7 SAVEVALUE (Emk+1),XN1 ; *Записать номер заявки*  
ADVANCE 4.5 ; *Задержка на время обслуживания*  
RELEASE Stan ; *Освободить ОУ*  
TERMINATE ; *Обслуженные заявки*  
Met1 TERMINATE ; *Потерянные заявки*  
; **Задание времени моделирования**  
GENERATE 15  
TERMINATE 1

Оператор GENERATE генерирует семь заявок с интервалами 2 единицы модельного времени, время прохождения которых представлено в таблице 39. В их нумерации, данной в скобках, отсутствует номер два (XN1 = 2), так как этот номер присвоен единственной заявке второго оператора GENERATE сегмента задания времени моделирования.

Таблица 39

События	Заявки						
	1(1)	2(3)	3(4)	4(5)	5(6)	6(7)	7(8)
Вход в модель	2	4	6	8	10	12	14
Занятие очереди	2	4	6	8	10	12	14
В очереди заявки	–	3	3, 4	4, 5	4, 5, 6	5, 6, 7	5, 6, 7
Занятие устройства	2	6,5	11	–	–	–	–
Освобождение устройства	6,5	11	15,5	–	–	–	–
Вывод из модели	6,5	11	15,5	–	–	–	14

Первый оператор TEST пропускает заявку к следующему оператору QUEUE, если в очереди на три заявки есть место. После постановки в очередь начинает работать сегмент учета номеров заявок, занявших место в очереди. Так как длина очереди ограничена тремя заявками, то в первый параметр – параметр цикла записывается число три. Далее организуется с помощью оператора LOOP цикл, в котором отыскивается возможность записи номера очередной заявки. Записать номер заявки можно будет тогда, когда значение какой-нибудь одной из трех ячеек сохраняемых величин X1, X2 или X3 будет равно нулю. Таких ячеек может быть больше одной, а заявка поступает одна, поэтому после записи номера осуществляется вы-

ход из цикла. Заявка занимает оператор SEIZE с меткой Met3 и покидает очередь с именем Dlina. Теперь начинает работать сегмент учета номеров заявок, покинувших очередь. Вначале в параметр цикла записывается число три (по длине очереди). Затем в цикле находится одна из сохраняемых ячеек X1, X2 или X3, в которой записан номер заявки, покидающей очередь. Эта ячейка обнуляется. Так как учет номеров заявок, находящихся в очереди и на обслуживании, ведется отдельно, то в сохраняемую ячейку с номером (Emk+1), т. е.  $3 + 1 = 4$  записывается номер заявки, занявшей устройство Stan. После завершения обслуживания устройство Stan освобождается, а сохраняемая ячейка с номером (Emk+1) обнуляется, т. е. обслуженная заявка списывается с учета и выводится из модели. Анализ результатов моделирования подтверждает правильность работы модели. В  $t = 14$  восьмая заявка удалена из модели, так как очередь заполнена. Моделирование завершается в  $t = 15$ . В очереди находятся пятая, шестая и седьмая заявки, а на обслуживании – четвертая.

### **Операторы объектов аппаратной категории**

Операторы, описывающие объекты аппаратной категории, а именно одноканальные устройства обслуживания и многоканальные устройства (памяти или накопители) рассматриваются в лабораторных работах №3-6. Помимо названных, к объектам аппаратной категории относятся ключи – логические переключатели.

*LOGIC* – моделирование переключателей. Для моделирования переключателей, имеющих только два состояния, в GPSS используются логические ключи. Логический ключ может находиться в одном из двух состояний: включен (ON или 1) или выключен (OFF или 0). В зависимости от состояния ключа может изменяться направление движения заявок. Логический ключ моделируется оператором LOGIC:

#### **LOGIC X A**

Операнд A это имя или номер логического ключа. Может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Логический оператор X – состояние логического ключа устанавливается в зависимости от следующих его значений:

- S – логический ключ, заданный операндом A, включается;
- R – логический ключ, заданный операндом A, выключается;
- I – логический ключ инвертируется, т. е. состояние его меняется на противоположное, например, если он был включен, то будет выключен.

Проверка состояния логического ключа может быть организована с использованием оператора GATE, который имеет такой же формат, как и при проверке состояний ОУ и накопителей, и такие же два режима работы:

GATE X A,[B]

Операнд A это имя или номер проверяемого ключа. Может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр. Операнд B – метка оператора, к которому будет направлена заявка в случае невыполнения условия, заданного условным оператором X. Условный оператор может принимать значения LS (равен 1, если логический ключ, заданный операндом A, включен; 0 – если выключен) или LR (равен 1, если логический ключ, заданный операндом A, выключен; 0 – если включен).

Проверка состояния логического ключа также может быть организована с использованием оператора TEST (см. ЛР№5) или булевой переменной (см. ЛР№6), для которой разрешено использование только СЧА LS.

### **Операторы получения статистических результатов**

Для сбора статистических данных о работе очередей устройств обслуживания используются парные операторы очереди QUEUE (см. ЛР№4) и DEPART (см. ЛР№4). Помимо этого в GPSS может быть организован сбор любых статистических данных по желанию пользователя.

*TABULATE* – сбор данных в таблицу. Для получения плотности распределения, среднего и стандартного отклонения некоторых аргументов, которыми могут быть СЧА (например, времени нахождения заявки в модели или задержки в ее отдельных частях, длин очередей, содержимого накопителя и т. д.), используются статистические таблицы TABLE и QTABLE. Команда описания таблицы TABLE имеет следующий формат:

Name TABLE A,B,C,D

Команда определяет аргумент, а также число и ширину частотных интервалов (классов). Метка Name определяет имя таблицы. Операндом A задается аргумент таблицы – элемент данных, чье частотное распределение будет табулироваться. Операнд может быть именем, выражением в скобках или СЧА. Операндом B задается верхний предел первого интервала. Операнд C задает ширину частотного интервала – разницу между верхней и нижней границей каждого частотного класса. Операнды B и C могут быть числами или рядом. Операндом D задается число частотных интервалов (положительное целое число).

Для сбора данных заявка должна войти в оператор TABULATE с тем же именем таблицы, которое определено в операторе TABLE. Оператор помещается в ту точку модели, которая соответствует исследуемому объекту. Оператор TABULATE имеет следующий формат:

TABULATE A,[B]

Операндом А задается имя таблицы, в которую табулируется значение аргумента. Операндом В (весовой коэффициент) задается число единиц, которые должны быть занесены в тот частотный интервал, куда попало значение аргумента. Если операнд В отсутствует, то по умолчанию эта величина равна 1. Операнды А и В могут быть именем, выражением в скобках, СЧА или СЧА\*параметр. Операнд А может быть только положительным целым числом, а операнд В – положительным числом. Например:

VrRem TABLE P\$ReaLvs,8.2,5,5,10

...

TABULATE VrRem

Оператором TABLE описывается таблица с именем VrRem. Аргументом таблицы является СЧА P\$ReaLvs с верхним пределом первого интервала 8,2; шириной 5,5 и числом интервалов 10. Каждое значение табулируемого аргумента P\$ReaLvs, меньшее или равное 8,2, увеличивает частоту первого частотного класса таблицы на 1. Если аргумент таблицы не попадает в первый частотный класс, класс определяется делением значения аргумента на операнд С оператора TABLE. Например, значение P\$ReaLvs равно 30,25. Тогда  $30,25/5,5 = 5,5$  и будет увеличена на 1 частота шестого класса. Если аргумент А таблицы больше  $B + C \cdot D = 8,2 + 5,5 \cdot 10 = 63,2$ , то изменен будет на 1 последний (десятый) класс. Одновременно корректируются текущие значения СЧА таблицы: счетчик входов в таблицу ТС, среднее время ожидания ТВ и стандартное отклонение времени ожидания TD. Собранная в таблице статистика выводится в стандартный отчет.

При использовании в операнде В оператора TABULATE весового коэффициента, например:

TABULATE VrRem,3

последний будет добавляться к значению частоты частотного класса. Весовой коэффициент применяется также для среднего и стандартного отклонения, что равносильно нескольким входам в оператор TABULATE.

Таким образом, команда TABLE вместе с оператором TABULATE служат для табулирования любого СЧА.

Кроме таблиц TABLE могут использоваться Q-таблицы, являющиеся средством получения распределения только времени пребывания заявки в очереди. Формат команды описания Q-таблицы такой же, как и TABLE. Отличие состоит в том, что операндом A задается имя очереди. Назначение операндов B, C, D такое же, что и в команде описания TABLE. Операнд B может быть нулем или положительным числом. Для создания в модели такой таблицы ее нужно предварительно определить с помощью команды QTABLE формата:

Name QTABLE A,B,C,D

Например, в операторе формата:

VTime QTABLE Dlina,18.2,4.3,6

Dlina – это имя очереди к устройству, а не СЧА, как в случае использования TABLE.

При прохождении заявки через операторы QUEUE и DEPART ее время ожидания фиксируется, и к счетчику частотного интервала таблицы, в который попало это время, добавляется 1. Одновременно в таблице накапливается информация для вычисления среднего значения и среднеквадратического отклонения времени ожидания. Следует обратить внимание, что при использовании QTABLE информация в таблицу заносится автоматически при входе заявки в операторы QUEUE и DEPART и никаких специальных мер, т. е. оператора TABULATE, при этом не требуется (см. ЛР№4). По окончании моделирования собранная в таблице информация также выводится в стандартном отчете GPSS.

### **Операторы организации списка пользователя**

Списки пользователя используются для сокращения времени моделирования, а также для создания сложных дисциплин обслуживания. При движении по модели заявки могут быть заблокированы, например, при проверке состояния устройств операторами GATE и TEST. Если заблокированные заявки находятся в СТС, то при их большом количестве планировщик расходует много времени на просмотр СТС с целью выбора очередной заявки для продвижения. Для экономии машинного времени заблокированные заявки целесообразно помещать в списки пользователя и оставлять их там до тех пор, пока не будут выполнены условия, позволяющие их дальнейшее продвижение. Кроме того, размещение ожидающих

заявок в списках пользователя позволяет организовать различные дисциплины очередей, отличающиеся от дисциплины FIFO (первым пришел – первым обслужен), реализованной в списке текущих событий.

Список пользователя представляет собой некоторый буфер, в который могут временно помещаться заявки, выведенные из СТС. В отличие от списков текущих и будущих событий заявки вводятся в список пользователя и выводятся из него не автоматически, а по решению пользователя в соответствии с логикой модели при помощи специальных операторов.

Для ввода заявок в список пользователя служит оператор LINK (ввести в список), который может быть использован в двух режимах: условном и безусловном.

*LINK* – ввод заявок в список пользователя. Рассмотрим безусловный режим, в котором оператор LINK имеет следующий формат записи:

[имя] LINK A,B

Операндом А задается имя или номер списка пользователя, в который безусловно помещается заявка, вошедший в оператор LINK. Операнд В определяет, в какое место списка пользователя следует поместить вошедшую заявку. Допустимые значения:

- FIFO – заявка помещается в конец списка;
- LIFO – заявка помещается в начало списка;
- PR – заявки упорядочиваются по убыванию приоритета;
- P – заявки помещаются позади тех заявок, значения соответствующего параметра которых меньше (в порядке возрастания);
- M1 – заявки помещаются в порядке возрастания относительного времени пребывания в модели.

В качестве операнда В могут использоваться и другие СЧА, кроме указанных СЧА заявок: арифметическая переменная, функция, а также выражение в скобках. В этом случае выполняется вычисление указанного операндом В для активной заявки и для всех остальных заявок, уже находящихся в списке пользователя, начиная с начала очереди. После этого производится упорядочивание заявок в списке пользователя по убыванию вычисленного значения. Например:

- LINK 3,FIFO – помещает заявки в конец списка пользователя с номером 3 в порядке их поступления в оператор;
- LINK Otst,P\$Pol – помещает заявки в список пользователя с именем Otst, упорядочивая их по возрастанию параметра с именем Pol.

Условия, при которых заявка помещается в список пользователя, в безусловном режиме проверяются средствами, предусмотренными разра-

ботчиком модели. Например, направить заявку в список пользователя в случае занятости устройства можно так следующим образом:

```
...
      GATE  NU   Rem1,Wait
      SEIZE Rem1
...
Wait  LINK   Otst,FIFO
...
```

Если устройство Rem1 занято, то оператор GATE не впускает заявку в оператор SEIZE, а направляет ее в оператор LINK с именем Wait, и заявка вводится в конец списка пользователя с именем Otst. Аналогично:

```
...
      GATE  U    Rem1,Met1
      LINK  Otst,FIFO
Met1  SEIZE  Rem1
...
```

Здесь устройство Rem1 проверяется на занятость. Если устройство занято, заявка проходит к следующему оператору LINK и помещается в список пользователя с именем Otst. В случае незанятости устройства, заявка направляется к оператору SEIZE с меткой Met1 и занимает свободное устройство.

В рассмотренных примерах предполагается, что список пользователя неограничен. При моделировании реальных систем список пользователя может использоваться для имитации, например, входного накопителя, емкость которого ограничена. Это ограничение можно реализовать следующим образом:

```
Emk  EQU    10
...
      GATE  NU   Rem1,Wait
      SEIZE          Rem1
...
Wait  TEST   L    CH$Otst,Emk,Term1
      LINK          Otst.LIFO
```

Если устройство Rem1 занято, то оператор GATE не позволяет заявке войти в оператор SEIZE, а направляет ее в оператор TEST с меткой

Wait, находящийся перед оператором LINK. Если текущее содержимое списка пользователя с именем Otst меньше заданной емкости Emk, заявка проходит в список пользователя, в противном случае направляется к оператору с меткой Term1.

*Ввод заявок в список пользователя в условном режиме.* Для ввода заявок в список пользователя используется оператор LINK формата:

[имя] LINK A,B,[C]

Операндом C указывается метка оператора, к которому переходит активная заявка в случае, если индикатор (флаг) списка пользователя установлен в 0 (выключен). Индикатор имеется у каждого списка пользователя. Когда операнд C не используется, т. е. оператор LINK работает в безусловном режиме, индикатор устанавливается в 1 (включается) и все вошедшие в оператор LINK заявки помещаются в список пользователя.

Рассмотрим использование списка пользователя в условном режиме:

```
...
      LINK      Nak,M1,Met1
Met1 SEIZE      Can
      ADVANCE   V$Zad
      RELEASE   Can
      UNLINK    Nak,Met1,1
...
```

Заметим, что если бы оператор LINK в этом примере был использован без операнда C, то модель не работала бы, так как все заявки, входящие в оператор LINK, помещались бы в список пользователя и ни одна из них не попала бы в устройство Can.

В примере с использованием оператора C индикатор списка пользователя управляется операторами LINK и UNLINK (см. вывод заявок из списка пользователя). Если в операторе LINK используется операнд C, оператор UNLINK при обнаружении того, что список пользователя пуст, сбрасывает индикатор в 0 (выключает). Первая заявка, вошедший в оператор LINK, находит индикатор в выключенном состоянии, поэтому не помещается в список пользователя, а направляется к оператору сметкой Met1, указанной операндом C. После этого оператор LINK устанавливает индикатор в 1 (включает).

Следующая заявка может войти в оператор LINK либо раньше, чем тот закончит обслуживание предыдущей, либо после того, как это обслуживание завершится. Если следующий заявка войдет в оператор LINK

раньше, чем будет закончено обслуживание в устройстве с именем Can предыдущей заявки, она будет помещена в список пользователя. Если обслуживание предыдущей заявки закончится раньше, чем очередная заявка войдет в оператор LINK, то обслуженная заявка (она же выводющаяся) войдет в оператор UNLINK и будет обнаружено, что список пользователя пуст. Оператор UNLINK установит индикатор списка пользователя в 0 (выключит). Поэтому следующая заявка вновь будет направлен к оператору с меткой Met1, а не помещен в список пользователя.

Таким образом оператор LINK в условном режиме и оператор UNLINK управляют помещением заявок в список пользователя: *если список пуст, заявка в него не помещается, а направляется к оператору с меткой, указанной операндом С.*

Операторы LINK и UNLINK предоставляют возможность пользователю формировать свои списки в динамике вне зависимости от списков задержки, которые автоматически управляются системой GPSS World. При применении операторов LINK и UNLINK операторы QUEUE/DEPART для сбора статистики об очереди не используются, так как почти все те же данные можно получить из статистики о списке пользователя.

*UNLINK – вывод заявок из списка пользователя.* Вывод заявок из списка пользователя. Для вывода одного или нескольких заявок из списка пользователя и помещения их обратно в список текущих событий служит оператор UNLINK (вывести из списка), имеющий следующий формат:

[имя] UNLINK X A,B,C,[D],[E],[F]

Операндом А указывается имя или номер списка пользователя. Операнд В – метка оператора, в который переходят выведенные из списка пользователя заявки.

Операндом С указывается число выводимых заявок или ключевое слово ALL (по умолчанию). Операнды D и E вместе с условным оператором X определяют способ и условия вывода заявок из списка пользователя. Значения оператора X те же, что и в операторе TEST (см. ЛР№5). В случае, когда условный оператор X должен использоваться, но не указан, по умолчанию он принимает значение E (равенство). Если операнды D и E не используются, не указывается и условный оператор X. В этом случае заявки выводятся с начала списка, а количество выводимых заявок определяется обязательным операндом С.

Операнд D может быть: булевой переменной; номером параметра заявки; ключевым словом BACK. Если операнд D является булевой переменной, операнд E и оператор X не используются. Булева переменная вы-

числяется относительно заявки, находящейся в списке пользователя. Если результат не равен нулю, т. е. условие вывода выполняется, заявка выводится. Количество выводимых заявок определяется операндом С. Однако выведено может быть и меньше, чем указано операндом С: по числу ненулевых результатов вычисления булевой переменной. Кроме того, и заявок в списке пользователя может быть меньше, чем указано операндом С. Если операндом D указано ключевое слово BACK, операнд E и условный оператор X также не используются, а заявки выводятся с конца списка в количестве, определяемом обязательным операндом С. Если операнд D не булева переменная и не ключевое слово BACK, должны быть указаны операнд E и условный оператор X. Операнд D вычисляется относительно заявки, находящейся в списке пользователя, и используется в качестве номера параметра, значение которого сравнивается с результатом вычисления операнда E. Если операнд D задает параметр, а операнд E не используется, значение параметра заявки из списка пользователя сравнивается со значением такого же параметра выводящей заявки. Если они равны, заявка выводится из списка пользователя. И в этом случае количество выводимых заявок определяется операндом С.

Операндом F указывается имя оператора, куда переходит заявка, выходящая из оператора UNLINK, если из списка пользователя не выведена ни одна заявка. Если операнд F не используется, выходящая заявка переходит в следующий оператор независимо от количества выведенных заявок. Например:

– UNLINK 4,Apd,1 – выводит из списка пользователя с номером 4 одну заявку с начала списка и направляет ее в оператор с меткой Apd;

– UNLINK Otst,Mars,1,BACK – выводит из списка пользователя с именем Otst одну заявку с конца списка и перенаправляет ее в оператор с меткой Mars;

– UNLINK E P\$Wiw,App1,ALL,App2,P\$App2,App3 – выводит из списка пользователя, номер которого записан в параметре Wiw выводящей заявки, и направляет в оператор с меткой App1 все заявки, содержимое параметра App2 которых равно содержимому одноименного параметра выводящей заявки. Если таких параметров в списке не окажется, то выводящая заявка будет направлена в оператор с меткой App3, в противном случае – к следующему оператору.

Отметим следующие особенности выполнения оператора UNLINK. Во-первых, если операнды D и E содержат ссылки на СЧА заявок, операнд D вычисляется относительно заявок в списке пользователя, а операнд E –

относительно активной заявки. Во-вторых, после вывода заявок из списка планировщик продолжает или начинает продвижение заявки с наивысшим приоритетом, а при равенстве приоритетов отдает предпочтение заявке-инициатору вывода.

Рассмотрим примеры использования списков пользователя для организации сложных дисциплин обслуживания. Пусть в одноканальной СМО с ожиданием требуется организовать дисциплину обслуживания, при которой приоритет отдается заявкам с наименьшим временем обслуживания. Реализация такой модели приведена в примере 4.

#### *Пример 4*

	GENERATE	(Exponential(23,0,200))	; Источник заявок
	ASSIGN	Nbl,(Exponential(34,0,120))	; Записать в параметр время ; обслуживания
	GATE NU	Evm.Otst	; ОУ свободно?
App1	SEIZE	Evm	; Занять ОУ
	ADVANCE	P\$Nbl	; Обслуживание
	RELEASE	Evm	; Освободить ОУ
	UNLINK	Pol,App1,1	; Вывести из списка пользователя
	TERMINATE		; Выход обслуженных заявок из СМО
Otst	LINK	Pol, P\$Nbl	; Список пользователя
	GENERATE	3600	; Время моделирования
	TERMINATE	1	

В параметр с именем Nbl поступающих в модель заявок в операторе ASSIGN записывается случайное время обслуживания, вычисляемое с использованием встроенного генератора экспоненциального распределения. Если ОУ Evm свободно, то оператор GATE впускает заявку в оператор SEIZE, и ОУ занимается на время P\$Nbl. Если же в момент поступления заявки ОУ занято, то оператор GATE перенаправляет заявку в оператор LINK с меткой Otst, который вводит ее в список пользователя Pol, упорядочивая заявки по возрастанию времени обслуживания, записанного в параметре P\$Nbl. Оператор UNLINK по освобождении ОУ выводит с начала списка заявку с наименьшим временем обслуживания, обеспечивая тем самым заданную дисциплину.

Пусть в одноканальной СМО с ожиданием требуется организовать дисциплину обслуживания, при которой заявки не обслуживаются, если они находятся в очереди более Vrem единиц модельного времени. Реализация такой модели показана в примере 5.

### Пример 5

```
Vrem EQU 250 ; Время ожидания в очереди
Preb BVARIABLE (AC1-P1) 'LE' Vrem ; Определение булевой переменной
GENERATE (Exponential(23,0,120)) ; Источник заявок
ASSIGN Nbl,(Exponential(34,'0','150)) ; Записать в параметр
GATE NU Evm,Otst ; ОУ свободно?
App1 SEIZE Evm ; Занять ОУ
ADVANCE P$Nб ; Обслуживание
RELEASE Evm ; Освободить ОУ
UNLINK Pol,App1,1,BV$Preb, ,Term2 ; Вывести из списка
Term1 TERMINATE ; Обслуженные заявки
Term2 TERMINATE ; Потерянные заявки
Otst ASSIGN 1,AC1 ; Записать в параметр абсолютное
; модельное время
LINK Pol,P1 ; Список пользователя
GENERATE 3600 ; Время моделирования
TERMINATE 1
```

В параметр Nbl поступающих в модель заявок в операторе ASSIGN записывается случайное время обслуживания. В список пользователя с именем Pol заявки помещаются также, как в предыдущем примере. Перед оператором LINK вставлен оператор ASSIGN с меткой Otst, который в первый параметр заявки P1 записывает абсолютное модельное время в момент помещения ее в список пользователя.

В качестве операнда D оператора UNLINK используется булева переменная Preb, в которой сравнивается время пребывания заявки в очереди с заданным ограничением Vrem. Время нахождения в очереди определяется как разность между абсолютным модельным временем в момент входа выводящей заявки в оператор UNLINK и абсолютным модельным временем помещения заявки в список пользователя, хранящимся в параметре P1. При входе выводящей заявки булева переменная вычисляется относительно заявки, находящейся в списке пользователя. Если результат вычисления не нулевой, т. е. заданное в булевой переменной условие выполняется, выводится одна заявка, так как операнд C равен 1.

По количеству заявок, вошедших в оператор TERMINATE с меткой Met2, можно увидеть в скольких случаях в списке пользователя не оказалось заявок, время обслуживания которых было меньше или равно заданному ограничению пребывания в очереди.

## Специальные операторы

*COUNT* – определение числа объектов, удовлетворяющих заданному условию, с записью в параметр активной заявки. Пусть необходимо узнать число свободных ОУ или число накопителей с коэффициентом использования меньше 700 (в долях от 1 000) и др. Для этого используется оператор *COUNT*, который имеет следующий формат записи:

*COUNT X A,B,C,[D],[E]*

Оператор *COUNT* работает в двух режимах: условном или логическом. Режим работы определяется логическим или условный оператором *X*, который задает условие, например, подсчет числа свободных накопителей, занятых ОУ и др. Для ОУ используются следующие логические операторы: *NU, U, NI, I, FV, FNV*, для накопителей: *SE, SNE, SF, SNF, SV, SNV*, для логических ключей: *LR, LS*. Содержание логических операторов изложено при рассмотрении оператора *GATE* (см. *ЛР№4*). При использовании логических операторов операнды *D* и *E* могут быть опущены.

В качестве условных операторов используются следующие: *L, NE, LE, E, G, GE, MAX, MIN, NE*. Здесь *MAX (MIN)* – наибольшее (наименьшее) значение из всех значений СЧА объектов, удовлетворяющих заданному условию, а *NE* – значение проверяемого СЧА должно быть не равно заданному значению. Содержание остальных условных операторов – см. *ЛР№5-7*. Если используются условные операторы, то операнды *D* и *E* должны быть заданы.

Операнд *A* это номер параметра, вошедшей в оператор *COUNT* заявки, в котором будет организован счетчик числа объектов. Операнд *A* может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА\*параметр.

Операнды *B* и *C* определяют соответственно нижнюю и верхнюю границы диапазона изменения номеров или имен объектов, для которых проверяется заданное условие. Значение операнда *C* должно быть больше или равно значению операнда *B*. Операнды *A* и *B* могут быть такими же, как и операнд *A*.

Операндом *D* задается значение, которое сравнивается со значением операнда *E*. Операнды *D* и *E* используются только совместно с условными операторами *L, LE, E, NE, G, GE*. Если условный оператор не используется, то операнд *D* можно не задавать.

Операнд *E* это какое-либо из СЧА проверяемых объектов. Необходимо записывать только мнемоническое обозначение атрибута, поскольку диапазон изменения номеров объектов задан операндами *B* и *C*.

Примеры использования:

– COUNT L 3,4,8,X\$Bb2,FT – оператор COUNT работает в условном режиме. Подсчитывается число ОУ в диапазоне от 4 до 8 (операнды В и С), среднее время обслуживания (операнд Е) в которых одной заявки меньше времени, записанного в сохраняемой ячейке Bb2 (операнд D), и заносится в параметр 3 (операнд А) вошедшей в оператор COUNT заявки.

– COUNT SNF 2,Pogr1,Pogr3 – оператор COUNT работает в логическом режиме. Определяется и заносится в параметр 2 (операнд А) вошедшей в оператор заявки число незаполненных (оператор SNF) накопителей в диапазоне от Pogr1 до Pogr3 (операнды В и С). Операнды D и Е не используются. В данном примере накопители указаны именами. По условиям оператора COUNT необходимо этим именам предварительно присвоить командой EQU последовательные номера.

– COUNT E KolQ\_0,ObsQ1,ObsQ6,0,Q – оператор COUNT работает также в условном режиме. Определяется количество очередей (операнд Е) в диапазоне от ObsQ1 до ObsQ6 (операнды В и С), длины которых равны нулю (операнд D), и заносится в параметр активной заявки с именем KolQ\_0) (операнд А). Именам очередей ObsQ1-0bsQ6 командами EQU должны быть присвоены последовательные номера.

*SELECT* – выбор первого объекта в заданном диапазоне с записью номера в параметр активной заявки. Оператор SELECT выбирает первый объект в заданном диапазоне, который удовлетворяет определенному условию. Номер этого объекта заносится в заданный параметр вошедшей в оператор заявки. Оператор SELECT имеет следующий формат записи:

SELECT X A,B,C,[D],[E],[F]

Логические и условные операторы X здесь имеют тот же смысл, что и в операторе COUNT. Назначение операндов А, В, С, D и Е также аналогично. Операнд F задает номер альтернативного оператора для входящей заявки, если заданному условию не удовлетворяет ни один объект в указанном диапазоне. Если операнд F не используется, то заявка будет направлена к следующему по номеру оператору. Например:

– SELECT MIN 2,1,6,,FR – оператор SELECT работает в условном режиме (условный оператор MIN). В диапазоне от 1 до 6 (операнды В и С) определяется первое ОУ с минимальным коэффициентом использования (операнд Е). Номер выбранного ОУ записывается в параметр 2 (операнд А) вошедшей в оператор заявки. Операнды D и F не используются. Заявка из оператора SELECT будет направлен к следующему оператору.

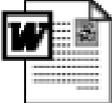
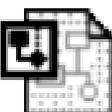
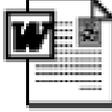
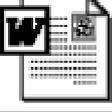
– SELECT SV PProg, Pogr1, Pogr3, Met4 – оператор SELECT работает в логическом режиме. Из диапазона от Pogr1 до Pogr3 (операнды B и C) определяется первый накопитель, который является доступным (логический оператор SV). Номер найденного накопителя записывается в параметр заявки с именем PProg (операнд A). Операнды D и E не используются. Если доступный накопитель не будет найден, активная заявка будет направлен к оператору с меткой Met4 (операнд F). В противном случае – к следующему оператору. Именам накопителей Pogr1-Pogr3 командами EQU должны быть присвоены последовательные номера.

– SELECT LE 4, Aa1, Aa7, X\$Bb3, Q – оператор SELECT также работает в условном режиме (условный оператор LE). Из очередей Aa1-Aa7 (операнды B и C) выбирается первая очередь (операнд E), длина которой меньше или равна значению ячейки Bb3 (операнд D). Номер очереди записывается в параметр 4 (операнд A) вошедшей в оператор заявки. Операнд F не используется, поэтому заявка из оператора SELECT направляется к следующему оператору. Именам очередей Aa1-Aa7 также командами EQU должны быть присвоены последовательные номера [30, 31].

## ЭЛЕКТРОННЫЕ МАТЕРИАЛЫ ПО КУРСУ

Помимо данного учебно-методического комплекса для успешного освоения курса «Моделирование», выполнения лабораторных работ и курсового проекта рекомендуется дополнительно ознакомиться с материалами, которые выдаются в электронном виде. Характеристика материалов приведена в таблице 40.

Таблица 40

Папка	Файл	Описание
 Пример Курсового Проекта	 Пояснительная записка Документ Microsoft Word 3 815 КБ	Пример задания на проектирование многопоточной СМО с отказами доступа входных очередей и пояснительной записки к курсовому проекту.
	 Базовая ИМ СМО GPSS World Model 46 КБ	Базовая имитационная модель (ИМ) многопоточной СМО с отказами доступа входных очередей.
	 Модификация ИМ СМО_2.1 GPSS World Model 46 КБ	Модификация базовой ИМ, реализованная для решения первой задачи анализа второй части курсового проекта.
	 Модификация ИМ СМО_2.2 GPSS World Model 56 КБ	Модификация базовой ИМ, реализованная для решения второй задачи анализа второй части курсового проекта.
	 ГСЧ для лимитирующего процесса Лист Microsoft Excel	Генератор случайных чисел для определения числа прогонов ИМ для обеспечения заданной точности моделирования по лимитирующему процессу.
	 Сеть Петри СМО Microsoft Visio Drawing 178 КБ	Граф сети Петри модели многопоточной СМО с отказами доступа входных очередей.
	 Бланк защиты КП Документ Microsoft Word 69 КБ	Бланк защиты курсового проекта. Распечатанный бланк вкладывается (не подшивается!) в сброшюрованную пояснительную записку для оформления протокола проверки и защиты проекта.
	 Титульный лист КП для заочников Документ Microsoft Word	Образец оформления титульного листа курсового проекта для студентов заочного отделения.
	 Правила оформления ПЗ по ГОСТ2.105-95 Документ Microsoft Word	Правила оформления пояснительной записки к курсовому проекту.

Папка	Файл	Описание
 Программное обеспечение моделирования	 Tau	Программа моделирования систем автоматического управления. Используется в лабораторной работе №2.
	 CircuitMaker	Дистрибутив системы автоматизации функционального и схемотехнического проектирования Circuit Maker. ПО используется в лабораторных работах №1 и №2.
	 GPSS World	Дистрибутив ПО моделирования GPSS World. ПО используется в лабораторных работах №3–6 и в курсовом проектировании.
	 IngProject	Программа моделирования временных ингибиторных сетей Петри. Используется в лабораторной работе №7 и, при необходимости, в курсовом проектировании.
 Работа №1	 Пример (ЛР№1) Лист Microsoft Excel 22 КБ	Расчет статического риска сбоя для схемы примера выполнения лабораторной работы №1.
	 Пример (ЛР№1) Файл "СКТ" 4 КБ	Реализация схемы примера выполнения лабораторной работы №1 в Circuit Maker. Используется для модельных экспериментов анализа статического сбоя.
 Работа №2	 Пример (ЛР№2).zvn Файл "ZVN" 80 КБ	Реализация схемы примера выполнения части 1 лабораторной работы №2 в программе tau.exe.
	 Пример (ЛР№2) Файл "СКТ" 3 КБ	Реализация схемы примера выполнения части 1 лабораторной работы №2 в Circuit Maker.
	 Варианты индивидуальных заданий (часть 2) Лист Microsoft Excel	Варианты индивидуальных заданий для выполнения части 2 лабораторной работы №2 (к задаче идентификации).
	 Пример (ЛР№2) Лист Microsoft Excel 278 КБ	Расчеты по примеру решения задачи идентификации к части 2 лабораторной работы №2.

Папка	Файл	Описание
<p>Работа №7</p> 	 <p>пример1.rne Файл "PNE" 3 КБ</p>	<p>Пример реализации в IngProject сети Петри модели процессов возникновения/устранения отказов к лабораторной работе №7.</p>
	 <p>пример2.rne Файл "PNE" 3 КБ</p>	<p>Пример реализации в IngProject сети Петри модели СМО параллельного обслуживания с переменным размером пакета заявок.</p>
	 <p>пример3.rne Файл "PNE" 6 КБ</p>	<p>Пример реализации в IngProject сети Петри модели магистрального канала передачи данных.</p>
<p>Примеры GPSS</p> 	 <p>пример с переменным размером входного пакета GPSS World Model</p>	<p>Дополнительный пример реализации в GPSS СМО с равномерно распределенным размером (в знаках) пакетов входного потока и зависимостью времени обслуживания в ОУ от размера пакета.</p>
	 <p>пример с управлением количеством заявок GPSS World Model</p>	<p>Дополнительный пример реализации в GPSS СМО с управлением количеством единовременно обслуживаемых заявок.</p>

## ЛИТЕРАТУРА

1. Шалагинов, А. В. Моделирование: экскурс в дисциплину / А. В. Шалагинов. – Новосибирск: НГТУ, 1998. – 24 с.
2. Шеннон, Р. Имитационное моделирование систем – искусство и наука / Р. Шеннон. – М. : Мир, 1978. – 418 с.
3. Основы автоматизированного проектирования радиоэлектронной аппаратуры: УМК / сост. и общ. ред. О. Е. Шестопаловой. – Новополюк : ПГУ, 2004. – 224 с.
4. Барышников, Ю. А. Моделирование: конспект лекций для специальности «Вычислительные машины, комплексы, системы и сети» / Ю. А. Барышников. – Екатеринбург: ГОУ ВПО «УГТУ-УПИ», 2005. – 87 с.
5. Советов, Б. Я. Моделирование систем. Учебник для вузов / Б. Я. Советов, С. А. Яковлев. – М. : Высш. шк., 2005. – 343 с.
6. Тарасик, В. П. Математическое моделирование технических систем / В. П. Тарасик. – Минск : Дизайн ПРО, 2004. – 640 с.
7. Володин, Д. О. Моделирование информационных процессов и систем / Д. О. Володин [и др.]. – М. : МГДД(Ю)Т, МИРЭА, ГНИИ ИТТ «Информика», 2002. – 50 с.
8. Громов, Ю. Ю. Системный анализ в информационных технологиях: Учеб. пособие / Ю. Ю. Громов [и др.]. – Тамбов : ТГУ, 2004. – 176 с.
9. Семенов, А. Д. Идентификация объектов управления: Учебн. пособие / А. Д. Семенов, Д. В. Артамонов, А. В. Брюхачев. – Пенза : Изд-во Пенз. гос. ун-та, 2003. – 211 с.
10. Моделирование дискретных устройств: методические указания по курсу «Проектирование элементов и устройств систем управления» / сост. В. В. Муханов, А. В. Серегин. – Екатеринбург : ГОУ ВПО «УГТУ-УПИ», 2006. – 43 с.
11. Анхимюк В.Л., Опейко О.Ф., Михеев Н.Н. Теория автоматического управления. – Мн.: Дизайн ПРО, 2000. – 352 с.
12. Дрейпер, Н. Прикладной регрессионный анализ: В 2-х кн. Кн.1 / Н. Дрейпер. – М. : Финансы и статистика, 1986. – 365 с.
13. Дрейпер, Н. Прикладной регрессионный анализ: В 2-х кн. Кн.2 / Н. Дрейпер. – М. : Финансы и статистика, 1987. – 352 с.
14. Яворский, В. А. Планирование научного эксперимента и обработка экспериментальных данных / В. А. Яворский. – Долгопрудный : МФТИ, 2006. – 24 с.
15. Жирабок, А. Н. Планирование эксперимента для построения математических моделей / А. Н. Жирабок // Соросовский образовательный журнал, 2001. – том 7, №9. – С. 121 – 127.
16. Соболев, И.М. Метод Монте-Карло / И. М. Соболев. – М. : Наука, 1978. – 64 с.
17. Паньгина, Н. Н. Статистическое моделирование: метод Монте-Карло / Н. Н. Паньгина, А. А. Паньгин // Компьютерные инструменты в образовании, 2002. – №5. – С. 30 – 43.
18. Голуб, В. А. Моделирование случайных величин: Учеб. пособие / В. А. Голуб, Т. М. Жукова, М. А. Соколова. – Воронеж : ВГУ, 2005г. – 27 с.
19. Афанасьева, О. В. Теория и практика моделирования сложных систем: Учеб. пособие / О. В. Афанасьева, Е. С. Голик, Д. А. Первухин. – СПб: СЗТУ, 2005г. – 131 с.

20. Блюмин, С. Л. Дискретное моделирование систем автоматизации и управления / С. Л. Блюмин, А. М. Корнеев. – Липецк : ЛЭГИ, 2005г. – 124 с.
21. Murata, T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE. – 1989. – Voll.77. – №4.
22. Khoromskaia, V. Petri Nets Based Modelling of Control Flow // Вестник РУДН, серия «Прикладная и компьютерная математика». – 2004. – №1. – С. 74 – 84.
23. Shestopalova, O. E. Ingproject – Petri Nets Software Tool for Queuing Systems Simulation // Проблемы проектирования и производства РЭС: Сборник материалов Международной научно-технической конференции. Секция «Информатика». – Новоолоцк : ПГУ, 2008. – С. 91 – 94.
24. Моделирование систем с использованием теории массового обслуживания: Учеб. пособие / под ред. Д. Н. Колесникова. – СПб : СПбГПУ, 2003. – 180 с.
25. Башарин, Г. П. Модели информационно-вычислительных систем: Сборник научных трудов / Г. П. Башарин. – М. : Наука, 1994. – 78 с.
26. Вычислительные системы, сети и телекоммуникации / под ред. В. Л. Брейдо. – СПб : Питер, 2003. – 688 с.
27. Лаврусь, О. Е. Теория массового обслуживания: методические указания, учебная программа и задания для контрольных работ для студентов заочной формы обучения специальности 071900 «Информационные системы в технике и технологиях» / О. Е. Лаврусь, С. Ф. Миронов. – Самара : СамГАПС, 2002. – 38 с.
28. Щеклеин, В. С. Моделирование информационных систем: конспект лекций / В. С. Щеклеин. – Ульяновск : УлГТУ, 2002. – 46 с.
29. Окольнішников, В. В. Представление времени в имитационном моделировании / В. В. Окольнішников // Вычислительные технологии. – 2005. – том 10. – №5. – С. 57 – 78.
30. Кудрявцев, Е. М. GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – М. : ДМК Пресс, 2004. – 320 с.
31. Боев, В. Д. Моделирование систем. Инструментальные средства GPSS World: Учеб. пособие / В. Д. Боев. – СПб : БХВ-Петербург, 2004. – 368 с.

## ПРИЛОЖЕНИЕ 1. ЭЛЕМЕНТЫ СТАНДАРТНОГО ОТЧЕТА GPSS

Строка заголовка стандартного отчета содержит имя файла модели, который создал отчет. Также в нее включена дата и время прогона модели:

GPSS World Simulation Report – Пример\_2.8.24.2

Tuesday, September 02, 2003 11:45:55

### 1. *Общая информация о результатах работы модели:*

- START TIME – начальное время. Абсолютное модельное время в момент начала моделирования. Устанавливается равным абсолютному модельному времени с помощью оператора RESET или CLEAR;
- END TIME – конечное время. Абсолютное модельное время, когда счетчик завершения принимает значение 0;
- BLOCKS – количество операторов, использованных в текущей модели, к моменту завершения моделирования;
- FACILITIES – количество устройств, использованных в модели, к моменту завершения моделирования;
- STORAGES – количество накопителей, использованных в текущей модели к моменту завершения моделирования.

2. *Информация об именах.* Файл статистики содержит информацию об именах, которые просматривает GPSS в ходе моделирования:

- NAME – перечень заданных пользователем имен, содержащихся в программе модели;
- VALUE – числовое значение, присваиваемое имени.

### 3. *Информация о блоках (операторах):*

- LABEL – метка, алфавитно-цифровое имя данного оператора (если оно задано);
- LOC – числовой номер позиции данного оператора в модели;
- BLOCK TYPE – тип оператора GPSS;
- ENTRY COUNT – количество заявок, вошедших в данный оператор, с начала работы программы или после последнего выполнения оператора RESET или CLEAR;
- CURRENT COUNT – количество заявок, находящихся в данном операторе к моменту завершения моделирования;
- RETRY – количество заявок, ожидающих специального условия, зависящего от состояния данного оператора.

4. *Информация об объектах типа «устройство».* Элементы статистики, представленные в данном разделе, имеют следующее содержание:

- FACILITY – имя или номер устройства;

- ENTRIES – количество раз, когда устройство было занято или занято с прерыванием с начала моделирования или после последнего выполнения оператора RESET или CLEAR;
- UTIL. – коэффициент использования, доля времени моделирования, в течение которого устройство было занято;
- AVE.TIME – среднее время занятия устройства одной заявкой в течение времени моделирования с начала моделирования или после выполнения оператора RESET или CLEAR;
- AVAIL. – состояние устройства в конце моделирования (равно 1, если устройство доступно; 0 – если недоступно);
- OWNER – номер заявки, который занимает устройство (0 – устройство не занято);
- PEND – количество заявок, ожидающих выполнения с прерыванием других заявок (т. е. вошедших в операторы PREEMPT в режиме прерывания);
- INTER – количество заявок, прерванных на данный момент (количество заявок в списке прерываний);
- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данного устройства;
- DELAY – количество заявок, ожидающих занятия устройства (входят так же заявки, ожидающие занятия устройства в режиме прерывания с помощью операторов PREEMPT).

##### **5. Информация об объектах типа «очередь»:**

- QUEUE – имя или номер очереди;
- MAX – максимальное содержимое очереди в течение периода моделирования, который начинается с момента трансляции модели или применения оператора RESET или CLEAR;
- CONT. – текущее содержимое очереди в конце процесса моделирования;
- ENTRY – общее количество входов заявок в очередь в течение времени моделирования;
- ENTRY(0) – общее количество входов заявок в очередь с нулевым временем ожидания;
- AVE.CONT. – среднее значение содержимого очереди в течение времени моделирования;
- AVE.TIME – среднее время пребывания одной заявки в очереди с учетом всех входов в очередь;
- AVE.(–0) – среднее время пребывания одной заявки в очереди без учета «нулевых» входов в очередь;

- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния очереди.

**6. Информация об объектах типа «накопитель»:**

- STORAGE – имя или номер накопителя;
- CAP. – емкость накопителя, заданная оператором STORAGE;
- REM. – число единиц накопителя свободных в конце моделирования;
- MIN. – минимальное число единиц накопителя, использовавшихся за период моделирования;
- MAX. – максимальное число единиц накопителя, использовавшихся за период моделирования;
- ENTRIES – количество входов в накопитель за период моделирования;
- AVL. – состояние накопителя в конце моделирования (1 – доступно; 0 – недоступно);
- AVE.C – среднее значение занятой емкости за период моделирования;
- UTIL. – коэффициент использования накопителя;
- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данного накопителя;
- DELAY – количество заявок, ожидающих в операторах ENTER, связанных с данным накопителем.

**7. Информация о таблицах:**

- TABLE – имя или номер таблицы или Q-таблицы;
- MEAN – средневзвешенное значение табулируемого аргумента;
- STD.DEV. – взвешенное среднеквадратическое отклонение;
- RANGE – нижний и верхний пределы частотного класса;
- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данной таблицы;
- FREQUENCY – суммарная величина, которая формируется при попадании табулируемого аргумента в указанные границы (значения операнда В суммируются оператором TABULATE);
- CUM.% – величина частоты в процентах к общему количеству значений табулируемого аргумента.

При попадании табулируемого аргумента в интервал, который имеет значение больше или меньше нижней границы частотного класса или равно верхней границе, изменяется значение частоты (FREQUENCY). Операнд В (весовой коэффициент) оператора TABULATE может быть

использован для определения величины, которая добавляется в частотный класс при попадании табулируемого значения в этот частотный класс. Частотные классы, суммарное значение которых равно 0, в файл статистики не выводятся.

**8. Информация о списках пользователя:**

- USER CHAIN – имя или номер списка пользователя;
- SIZE – количество заявок в списке пользователя в конце времени моделирования;
- RETRY – количество заявок, ожидающих наступления специального условия, зависящего от состояния данного списка пользователя;
- AVE.CONT. – среднее содержимое списка пользователя в течение времени моделирования;
- ENTRIES – общее число заявок, входивших в список пользователя в течение времени моделирования;
- MAX – максимальное количество заявок в списке пользователя за период моделирования;
- AVE.TIME – среднее время пребывания заявки в списке пользователя.

**9. Информация о группах заявок:**

- XACT GROUP – имя или номер объекта группы заявок;
- GROUP SIZE –I число заявок в группе в конце моделирования;
- RETRY – число заявок, ожидающих выполнения специального условия, зависящего от состояния данной группы заявок.

**10. Информация о числовых группах:**

- NUMERIC GROUP – имя или номер числовой группы;
- GROUP SIZE – количество заявок, содержащихся в числовой группе в конце моделирования;
- RETRY – число заявок, ожидающих выполнения специального условия, зависящего от состояния данной числовой группы.

**11. Информация о логических переключателях:**

- LOGICSWITCH – имя или номер логического ключа;
- VALUE – значение ключа в конце моделирования (1 (true) – «установлен»; 0 (false) – «сброшен»);
- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данного логического ключа.

**12. Информация о сохраняемых величинах (ячейках):**

- SAVEVALUE – имя или номер ячейки;
- VALUE – значение сохраняемой величины в конце моделирования;

- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данной ячейки.

### **13. Информация о матрицах:**

- MATRIX – имя или номер матрицы;
- RETRY – количество заявок, ожидающих выполнения специального условия, зависящего от состояния данной матрицы;
- INDICES – до шести целых чисел, определяющих элемент матрицы;
- VALUE – значение элемента матрицы в конце моделирования (элементы, равные 0, выводятся в отчете группами).

**14. Информация о списках текущих и будущих событий.** Списки текущих (CEC) и будущих (FEC) событий выводятся в файл статистики, если в команде START значение операнда D равно 1. Элементы статистики для CEC, представленные в данном разделе, имеют следующее содержание:

- XN – номер каждой заявки, находящегося в списке текущих событий;
- PRI – приоритет заявки;
- M1 – время входа заявки в модель или время ее самого раннего предка (породившего данную заявку);
- ASSEM – номер семейства данной заявки;
- CURRENT – номер оператора, в котором находится заявка в конце моделирования;
- NEXT – номер следующего оператора, в который должна была войти заявка;
- PARAMETER – имя или номер параметра заявки;
- VALUE – значение параметра.

В содержании статистики для FEC имеется только одно отличие. Вместо поля M1 присутствует поле BDT, которое определяет момент абсолютного модельного времени, когда заявка покинет список будущих событий.

## ПРИЛОЖЕНИЕ 2. ОСНОВНЫЕ ОПЕРАТОРЫ GPSS

В таблице допустимые типы операндов указаны цифрами, которые соответствуют:

- 1 – Null (нет ввода);
- 2 – Name (имя);
- 3 – Number (неотрицательное число);
- 4 – String (строковый);
- 5 – ParenthesizedExpression (выражение в скобках);
- 6 – SNA (стандартный числовой атрибут – СЧА);
- 7 – SNA\*Parameter (косвенная адресация);
- 8 – PosInteger (положительное целое число);
- 9 – Integer (целое число со знаком (необязательно));
- 10 – DirectSNA (прямой СЧА, т. е. не использующий косвенную адресацию).

Название и формат	Операнды		
	имя	значение	тип
1. Источник потока заявок. Формат: GENERATE A,B,C,D,E	[A]	Среднее время	1...5, 10
	[B]	Полуинтервал или модификатор-функция	1...5, 10
	[C]	Время начальной задержки	1...5, 10
	[D]	Количество генерируемых заявок	1...5, 8, 10
	[E]	Приоритет	1, 2, 4, 5, 8, 9
2. Удаление из модели активной заявки. Формат: TERMINATE A	[A]	Величина уменьшения счетчика завершения	1, 2, 5...8
3. Задержка заявки на заданный отрезок модельного времени. Формат: ADVANCE A,B	A	Среднее время задержки	2...7
	[B]	Полуинтервал или модификатор-функция	1...7
4. Присвоение или изменение значения параметра. Формат: ASSIGN A[±],B,C	A	Параметр активной заявки	2...7
	B	Значение	1...7
	[C]	Номер функции	1, 2, 5...8
5. Установка приоритета активной заявки. Формат: PRIORITY A,B	A	Новое значение приоритета	2, 4...7, 9
	[B]	Помещение заявки в СТС	1, BU

Название и формат	Операнды		
	имя	значение	тип
6. Занятие устройства. Формат: SEIZE A	A	Имя или номер устройства	2, 5...8
7. Освобождение занятого устройства. Формат: RELEASE A	A	Имя или номер устройства	2, 5...8
8. Захват устройства. Формат: PREEMPT A,B,C,D,E	A	Имя или номер устройства	2, 5...8
	[B]	Режим	1, PR
	[C]	Метка оператора	1,2, 5...8
	[D]	Имя или номер параметра	1,2, 5...8
	[E]	Режим удаления	RE
9. Освобождение устройства от захвата. Формат: RETURN A	A	Имя или номер устройства	2, 5...8
10. Перевод устройства в недоступное состояние. Формат: FUNAVAIL A,B,C,D,E,F,G,H	A	Имя или номер устройства	2, 5...8
	[B]	Режим удаления или продолжения	1, RE, CO
	[C]	Метка оператора выхода	1, 2, 5...8
	[D]	Имя или параметр устройства	1, 2, 5...8
	[E]	Режим удаления или продолжения	1, RE, CO
	[F]	Метка оператора выхода	1,2, 5...8
	[G]	Режим удаления или продолжения	1, RE, CO
	[H]	Метка оператора выхода	1,2, 5...8
11. Перевод устройства в доступное состояние. Формат: FAVAIL A	A	Имя или номер устройства	2, 5...8
12. Занятие накопителя. Формат: ENTER A,B	A	Имя или номер накопителя	2, 5...8
	[B]	Число занимаемых элементов накопителя	1, 2, 5...8
13. Освобождение накопителя. Формат: LEAVE A,B	A	Имя или номер накопителя	2, 5...8
	[B]	Число освобождаемых элементов накопителя	1, 2, 5...8
14. Перевод накопителя в недоступное состояние. Формат: SUNAVAIL A	A	Имя или номер накопителя	2, 5...8

Название и формат	Операнды		
	имя	значение	тип
15. Перевод накопителя в доступное состояние. Формат: SAVAIL A	A	Имя или номер накопителя	2, 5...8
16. Увеличение длины очереди. Формат: QUEVE A,B	A	Имя или номер очереди	2, 5...8
	[B]	Число, на которое увеличивается длина очереди	1,2, 5...8
17. Уменьшение длины очереди. Формат: DEPART A,B	A	Имя или номер очереди	2, 5...8
	[B]	Число, на которое уменьшается длина очереди	1, 2, 5...8
18. Присоединение заявки к списку пользователя. Формат: LINK A,B,C	A	Имя или номер списка пользователя	2, 5...8
	B	Задание дисциплины обслуживания	5...7, LIFO, FIFO
	[C]	Метка альтернативного оператора	1, 2, 5...8
19. Удаление заявки из списка пользователя. Формат: UNLINK X A,B,C,D,E	[X]	Условный оператор	1,G,GE, L, LE, E, NE
	A	Имя или номер списка пользователя	2, 5...8
	B	Метка оператора выхода	2, 5...8
	[C]	Число удаляемых заявок	1, 2, 5...8 ALL
	[D]	Имя или номер параметра	1, 2, 5...8 BACK
	[E]	Аргумент для сравнения	1...7
	[F]	Метка альтернативного оператора	1, 2, 5...8
20. Модификация значения ячейки. Формат: SAVEVALUE A[±], B	A	Имя или номер ячейки	2, 5...8
	B	Модификатор	2...7
21. Изменение состояния логического ключа. Формат: LOGIC X A	X	Логический оператор	S, R, I
	A	Имя или номер логического ключа	2, 5...8
22. Управление потоком заявок в зависимости от результата сравнения операндов. Формат: TEST X A,B,C	X	Условный оператор	G, GE, L, LE, N, NE
	A	Первая сравниваемая величина	2...7
	B	Вторая сравниваемая величина	2...7
	[C]	Метка оператора	1, 2, 5...8

Название и формат	Операнды		
	имя	значение	тип
23. Переход активной заявки в указанный оператор. Формат: TRANSFER A,B,C,D	[A]	Режим перехода: безусловный; статистический; параметрический; функциональный; оба; все; выборочный; одновременный; подпрограммный	1, 2, 5...8 [.] .XXX P FN BOTH ALL PICK SIM SBR
	[B]	Имя или номер оператора (параметра, функции)	1, 2, 5...8
	[C]	Имя или номер оператора (аргумент для режимов P и FN)	1, 2, 5...8
	[D]	Шаг нумерации операторов для режима ALL	1, 2, 5...8
24. Управление потоком заявок в зависимости от состояния устройства. Формат: GATE X A,B	X	Проверяемое условие: занято или нет; захвачено или нет; доступно или нет	U, NU, I, NI, FV, FNV
	A	Имя или номер устройства	2, 5...8
	[B]	Метка оператора	1, 2, 5...8
25. Управление потоком заявок в зависимости от состояния накопителя. Формат: GATE X A,B	X	Проверяемое условие: пусто или нет; заполнено или нет; доступно или нет	SE, SNE, SF, SNF, SV, SNV
	A	Имя или номер памяти	2, 5...8
	[B]	Метка оператора	1, 2, 5...8
26. Управление потоком заявок в зависимости от состояния ключа. Формат: GATE X A,B	X	Проверяемое условие: ключ установлен ключ не установлен	LS, LR
	A	Имя или номер логического ключа	2, 5...8
	[B]	Метка оператора	1,2, 5...8
27. Организация циклического прохождения заявок в модели. Формат: LOOP A,B	A	Параметр, содержащий число циклов	2, 5...8
	[B]	Метка оператора – начало цикла	1, 2, 5...8

Название и формат	Операнды		
	имя	значение	тип
28. Управление потоком заявок в зависимости от состояния синхронизации проверяемой заявки. Формат: GATE X A,B.	X	Проверяемое условие: есть заявка, ожидающая условия синхронизации, в указанном операторе MATCH или нет	M, NM
	A	Метка оператора MATCH	2, 5...8
	[B]	Метка оператора	1, 2, 5...8
29. Создание необходимого числа копий (семейства) активной заявки. Формат: SPLIT A,B,C	A	Количество создаваемых копий	2, 5...8
	[B]	Метка оператора для входа копий	1,2,5...8
	[C]	Имя или номер параметра, в который помещаются порядковые номера копий	1,2,5...8
30. Объединение заявок одного семейства. Формат: ASSEMBLE A	A	Количество собираемых заявок	2, 5...8
31. Накопление заданного числа заявок одного семейства. Формат: GATHER A	A	Количество заявок, которые необходимо накопить	2, 5...8
32. Синхронизация движения двух заявок одного семейства. Формат: MATCH A	A	Метка сопряженного оператора MATCH	2, 5...8
33. Разрешение заявке на выполнение операции любого другого оператора Формат: EXECUTE A	A	Метка оператора, операция которого выполняется активной заявкой	2, 5...8
34. Определение или смена семейства активной заявки Формат: ADOPT A	A	Имя или номер семейства. Записывается в специальный параметр активной заявки (Assemble Set)	2, 5...8

### ПРИЛОЖЕНИЕ 3. СТАНДАРТНЫЕ ЧИСЛОВЫЕ АТТРИБУТЫ GPSS

Код	Значение СЧА	Примечание
<b>Системные числовые атрибуты</b>		
RNj	Случайное число. Целочисленное значение.	Датчик генерирует последовательность равномерно распределенных случайных целых чисел от 0 до 999.
C1	Относительное модельное время. Вещественное значение.	Модельное время с момента последней команды RESET. Изменяется автоматически.
AC1	Абсолютное модельное время. Вещественное значение.	Модельное время с момента последней команды CLEAR. Изменяется автоматически.
TG1	Число, равное текущему значению счетчика завершения. Целочисленное значение.	Начальное значение TG1 задается оператором START и указывает на завершение моделирования, когда становится равным нулю. Вошедшие в оператори TERMINATE с ненулевым положительным операндом A заявки уменьшают это значение на число, равное значению операнда A.
<b>Стандартные числовые атрибуты заявок</b>		
Pj	Значение j-го параметра активной заявки. Целочисленное, вещественное или строковое значение.	Для косвенной адресации используются: СЧА*j ; СЧА*имя параметра; СЧА*\$имя параметра.
XN1	Номер активной заявки. Целочисленное значение.	–
PR	Приоритет активной заявки. Целочисленное значение.	Может изменяться оператором PRIORITY. По умолчанию приоритет равен 0.
MBj	Соответствие в операторе MATCH. Целочисленное значение.	1, если в сопряженном операторе MATCH находится заявка того же семейства, что и активная; 0 – иначе.
MPj	Время прохождения заявкой некоторого участка модели. Вещественное значение.	Вычисляется как разность текущего абсолютного модельного времени и значения j-го параметра активной заявки. Разность – транзитное время – сохраняется в j-м параметре.
M1	Время пребывания в модели активной заявки. Вещественное значение.	Вычисляется как разность абсолютного модельного времени и отметки времени активной заявки, в которой хранится либо время поступления заявки в модель, либо время последнего прохождения заявкой оператора MARK без операнда A.

Код	Значение	Примечание
<b>Стандартные числовые атрибуты операторов</b>		
Nj	Общее число входов заявок в j-й оператор. Целочисленное значение.	Увеличивается при каждом входе заявки в j-й оператор.
Wj	Текущее число заявок, находящихся в j-м операторе. Целочисленное значение.	Увеличивается при каждом входе заявки в j-й оператор и уменьшается при каждом выходе заявки из j-го оператора.
<b>Стандартные числовые атрибуты накопителей</b>		
Sj	Текущее содержимое накопителя j. Целочисленное значение.	Изменяется операторами ENTER и LEAVE.
Rj	Число свободных единиц накопителя j. Целочисленное значение.	Изменяется операторами ENTER и LEAVE.
SRj	Коэффициент использования накопителя j. Вещественное значение.	Выражается в тысячных долях, но значение СЧА – целое, т. е. если коэффициент равен 0,65, то SRj равно 650. Может быть нецелочисленным.
SAj	Среднее значение занятого накопителя j. Вещественное значение.	Взвешенное по времени среднее количество занятых элементов накопителя.
SMj	Максимальное количество элементов занятого накопителя j. Целочисленное значение.	Максимальное количество элементов накопителя, которое было занято одновременно в процессе моделирования.
SCj	Счетчик использования накопителя j. Целочисленное значение.	Общее количество использовавшихся элементов накопителя j. Увеличивается при каждом входе заявки в оператор ENTER.
STj	Среднее время использования одного элемента накопителя j.	Определяется системой автоматически.
SEj	Занятость накопителя j.	1 – не занят; 0 – занят.
SFj	Заполненность накопителя j.	1 – заполнен; 0 – не заполнен.
SVj	Доступность накопителя j.	1 – доступен; 0 – недоступен.
<b>Стандартные числовые атрибуты обслуживающих устройств</b>		
Fj	Занятость устройства j. Целочисленное значение.	1 – занято; 0 – не занято. Может быть изменен операторами SEIZE, RELEASE, PREEMPT и RETURN.
FIj	Прерванность устройства j. Целочисленное значение.	1 – прервано; 0 – не прервано. Может быть изменен операторами PREEMPT и RETURN.
FVj	Доступность устройства j. Целочисленное значение.	1 – доступно; 0 – не доступно. Может быть изменен операторами FAVAIL и FUNAVAIL.

Код	Значение	Примечание
FRj	Коэффициент использования устройства j.	Выражается в тысячных долях, но значение СЧА – целое, т. е. если коэффициент равен 0,88, то FRj равен 880.
FCj	Количество заявок, занимавших устройство j.	Увеличивается при входе заявок в оператор SEIZE или PREEMPT.
FTj	Среднее время использования устройства j одной заявкой.	Определяется автоматически.
<b>Стандартные числовые атрибуты очередей</b>		
Qj	Текущая длина очереди j. Целочисленное значение.	Увеличивается при входе заявки в оператор QUEUE и уменьшается при входе в оператор DEPART.
QAj	Средняя длина очереди j. Вещественное значение.	Взвешенное по времени количество заявок в очереди j.
QMj	Максимальная длина очереди j. Целочисленное значение.	Максимальное количество заявок, находившихся одновременно в очереди j.
QCj	Общее число входов в очередь j. Целочисленное значение.	Автоматически увеличивается системой при входе заявки в очередь j.
QZj	Число входов с нулевым временем пребывания в очереди j.	Число входов заявок, время нахождения которых в очереди равно нулю.
QTj	Среднее время пребывания заявки в очереди j.	Рассчитывается с учетом нулевых входов.
QXj	Среднее время пребывания заявки в очереди j.	Рассчитывается без учета нулевых входов.
<b>Стандартные числовые атрибуты ячеек и матриц ячеек</b>		
Xj	Содержимое ячейки j. Целочисленное, вещественное или строковое значение.	Изменяется оператором SAVEVALUE.
MXj (a,b)	Содержимое ячейки матрицы j. Целочисленное, вещественное или строковое значение.	Значение ячейки в строке <b>a</b> и столбце <b>b</b> . Изменяется оператором SAVEVALUE. Для матриц, имеющих более двух измерений, все остальные индексы считаются равными 1. В отличие от СЧА класса MX, PLUS-выражения могут обращаться к любому элементу многомерных матриц.
<b>Системные числовые атрибуты вычислительных объектов</b>		
FNj	Результат вычисления значения функции j. Вещественное значение.	–
Vj	Результат вычисления целочисленной переменной j или переменной j с плавающей точкой. Вещественное значение.	–

Код	Значение	Примечание
BVj	Результат вычисления булевой переменной j. Вещественное значение.	Принимает значения: 1 (true); 0 (false).
<b>Системные числовые атрибуты групп, списков и ключей</b>		
GNj	Текущее число членов в j-й числовой группе.	–
GTj	Текущее число членов в j-й группе заявок.	–
CHj	Текущее число заявок в j-м списке пользователя.	Список пользователя создается и ведется самим программистом в отличие от остальных списков модели. Используются операторы LINK и UNLINK. Значения всех СЧА определяются аналогично значениям СЧА очередей.
CAj	Среднее число заявок в j-м списке пользователя.	–
CMj	Максимальное число заявок в j-м списке пользователя.	–
CCj	Общее число заявок в j-м списке пользователя.	–
CTj	Среднее время пребывания заявки в j-м списке пользователя.	–
LSj	Состояние j-го логического ключа.	Принимает значения: 1 – установлен; 0 – не установлен
<b>Системные числовые атрибуты таблиц</b>		
TBj	Среднее значение невзвешенных аргументов таблицы j.	Для занесения в таблицу используется оператор TABULATE.
TCj	Число включений в таблицу j.	Целочисленное значение.
TDj	Среднеквадратическое отклонение для таблицы j.	Вещественное значение.

## ПРИЛОЖЕНИЕ 4. ПАКЕТИРОВАНИЕ ЗАЯВОК

Для организации пакетирования заявок в GPSS World используются операторы GATHER и ASSEMBLE, действие которых принципиально отличается. Оператор GATHER задерживает входящие в него заявки одного семейства до тех пор, пока их количество не достигнет заданного числа. После этого заявки сформированного пакета по одной продвигаются далее по системе, а оператор начинает формирования очередного пакета. В этом случае объединение заявок в пакет (слияния) не происходит.

Если необходимо реализовать пакетирование со слиянием исходных заявок и дальнейшим единовременным продвижением по системе новой заявки-пакета, используется оператор ASSEMBLE.

### *Пример использования оператора GATHER.*

```
GENERATE 1,0      ;генерация заявок с интервалом в 1 сек.
ADOPT 1          ;все заявки становятся членами семейства №1
SEIZE USTR1
ADVANCE 1,0      ;одиночное обслуживание заявок в USTR1 за 1 сек.
RELEASE USTR1
GATHER 3         ;объединение заявок в пакет по три. Слияния
                ;не происходит. Каждая заявка в пакете остается
                ;самостоятельной.

QUEUE O2         ;в очередь заявки каждого пакета поступают
                ;по одной, с момента формирования пакета
                ;из 3-х заявок в блоке GATHER.

SEIZE USTR2
DEPART O2
ADVANCE 3,0      ;одиночное обслуживание заявок в USTR2 за 3 сек.
RELEASE USTR2
TERMINATE 1
```

Рассмотрим результаты моделирования, полученные в результате выполнения модели примера в режиме пошагового прогона. Для моделирования в этом режиме в программной реализации не должно быть команды START, а после компиляции следует выбрать пункт меню Window/Simulation Window/Blocks Window. Ситуация, показанная на рис. П 4.1, соответствует моменту накопления в операторе GATHER 3-х заявок входного потока. Видно, что до момента накопления в блоке GATHER 3-х заявок пакета дальнейшее их продвижение не осуществлялось, и в очередь второго устройства заявки не поступали.

Следующий шаг моделирования показан на рис. П 4.2. Видно, что одна заявка из сформированного в блоке GATHER пакета из 3-х заявок уже поступила в очередь второго устройства. Остальные две заявки пакета остаются в блоке GATHER, и будут по одной поступать в очередь по факту ее освобождения. Новые заявки, поступающие в блок GATHER, задерживаются до формирования нового пакета по 3, т. е. до момента, когда число входов (Entry Count) не станет равным 6.

Loc	Block Type	Current Co...	Entry Co...	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	4	0	1	0
2 ADO	ADOPT	0	4	0	2	0
3 SEI	SEIZE	1	4	0	3	0
4 ADV	ADVANCE	0	3	0	4	0
5 REL	RELEASE	0	3	0	5	0
6 GAT	GATHER	3	3	0	6	0
7 QUE	QUEUE	0	0	0	8	0
8 SEI	SEIZE	0	0	0	11	0
9 DEP	DEPART	0	0	0	12	0
10 ADV	ADVANCE	0	0	0	13	0
11 REL	RELEASE	0	0	0	14	0
12 TER	TERMINATE	0	0	0	15	0

Рис. П 4.1

Loc	Block Type	Current Co...	Entry Count	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	4	0	1	0
2 ADO	ADOPT	0	4	0	2	0
3 SEI	SEIZE	0	4	0	3	0
4 ADV	ADVANCE	1	4	0	4	0
5 REL	RELEASE	0	3	0	5	0
6 GAT	GATHER	2	3	0	6	0
7 QUE	QUEUE	1	1	0	8	0
8 SEI	SEIZE	0	0	0	10	0
9 DEP	DEPART	0	0	0	11	0
10 ADV	ADVANCE	0	0	0	12	0
11 REL	RELEASE	0	0	0	13	0
12 TER	TERMINATE	0	0	0	14	0

Рис. П 4.2

Результат моделирования, полученный в виде стандартного отчета при выполнении программы с командой START 100, показан на рис. П 4.3.

Видно, что на момент завершения моделирования 100 заявок завершили обслуживание (количество входов в оператор №12 TERMINATE), а одна поступила во второе устройство, но обслуживание еще не началось (1 в поле CURRENT COUNT оператора №4 ADVANCE). При этом 199 заявок находятся в очереди второго устройства (199 в поле CURRENT COUNT оператора №7 QUEUE), а 2 заявки задержаны в операторе №6 GATHER до прихода 3-ей, необходимой для формирования очередного пакета.

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	303	0	0
	2	ADOPT	303	0	0
	3	SEIZE	303	0	0
	4	ADVANCE	303	1	0
	5	RELEASE	302	0	0
	6	GATHER	302	2	0
	7	QUEUE	300	199	0
	8	SEIZE	101	1	0
	9	DEPART	100	0	0
	10	ADVANCE	100	0	0
	11	RELEASE	100	0	0
	12	TERMINATE	100	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
USTR1	303	0.997	1.000	1	303	0	0	0	0
USTR2	101	0.987	2.970	1	101	0	0	0	199

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY	
O2	201	200	300	1	99.671	101.000	101.338	0

Рис. П 4.3

*Пример использования оператора ASSEMBLE.*

```

GENERATE 1,0 ;генерация заявок с интервалом в 1 сек.
ADOPT 1 ;все заявки становятся членами семейства №1
SEIZE USTR1
ADVANCE 1,0 ;одиночное обслуживание заявок в USTR1 за 1 сек.
RELEASE USTR1
ASSEMBLE 3 ;объединение заявок в пакет по три со слиянием
;заявок. Далее пакет продвигается по системе
;как самостоятельная единица.

QUEUE O2 ;в очередь пакеты поступают по одному, по факту
;объединения заявок в пакет из 3-х в блоке
;ASSEMBLE.

SEIZE USTR2
DEPART O2
ADVANCE 3,0 ;одиночное обслуживание заявок в USTR2 за 3 сек.
RELEASE USTR2
TERMINATE 1

```

Рассмотрим результаты моделирования, полученные в результате выполнения модели примера в режиме пошагового прогона. Ситуация, показанная на рис. П 4.4, соответствует состоянию системы, в котором в блоке Assemble формируется первый пакет (CURRENT COUNT=1), но заявок для его формирования поступило еще недостаточно (ENTRY COUNT=2), поэтому очередь второго устройства пуста.

Loc	Block Type	Current Count	Entry Count	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	3	0	1	0
2 ADO	ADOPT	0	3	0	2	0
3 SEI	SEIZE	0	3	0	3	0
4 ADV	ADVANCE	1	3	0	4	0
5 REL	RELEASE	0	2	0	5	0
6 ASS	ASSEMBLE	1	2	0	6	0
7 QUE	QUEUE	0	0	0	8	0
8 SEI	SEIZE	0	0	0	10	0
9 DEP	DEPART	0	0	0	11	0
10 ADV	ADVANCE	0	0	0	12	0
11 REL	RELEASE	0	0	0	13	0
12 TER	TERMINATE	0	0	0	14	0

Рис. П 4.4

На рис. П 4.5 показан следующий шаг моделирования: первый пакет из 3-х заявок сформирован: ENTRY COUNT=3; и один этот объединенный пакет уже поступил в очередь второго устройства. Объединенный пакет, по сути, является новой самостоятельной заявкой, не сохранившей никаких признаков заявок-прародителей пакета.

Loc	Block Type	Current Co...	Entry Co...	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	4	0	1	0
2 ADO	ADOPT	0	4	0	2	0
3 SEI	SEIZE	0	4	0	3	0
4 ADV	ADVANCE	1	4	0	4	0
5 REL	RELEASE	0	3	0	5	0
6 ASS	ASSEMBLE	0	3	0	6	0
7 QUE	QUEUE	1	1	0	8	0
8 SEI	SEIZE	0	0	0	10	0
9 DEP	DEPART	0	0	0	11	0
10 ADV	ADVANCE	0	0	0	12	0
11 REL	RELEASE	0	0	0	13	0
12 TER	TERMINATE	0	0	0	14	0

Рис. П 4.5

Ситуация, имеющая место во время формирования второго пакета, показана на рис. П 4.6. В блоке ASSEMBLE формируется следующий, второй пакет (CURRENT COUNT=1). Его формирование закончится, когда ENTRY COUNT станет равным 6. До тех пор в очередь второго устройства ничего не поступает.

На рис. П 4.7 видно, что второй пакет сформирован и поступил в очередь второго устройства. К этому моменту первый пакет уже покинул систему обслуженным.

Loc	Block Type	Current Co...	Entry Co...	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	5	0	1	0
2 ADO	ADOPT	0	5	0	2	0
3 SEI	SEIZE	0	5	0	3	0
4 ADV	ADVANCE	0	5	0	4	0
5 REL	RELEASE	0	5	0	5	0
6 ASS	ASSEMBLE	1	5	0	6	0
7 QUE	QUEUE	0	1	0	8	0
8 SEI	SEIZE	0	1	0	10	0
9 DEP	DEPART	0	1	0	11	0
10 ADV	ADVANCE	1	1	0	12	0
11 REL	RELEASE	0	0	0	13	0
12 TER	TERMINATE	0	0	0	14	0

Рис. П4.6

Loc	Block Type	Current Co...	Entry Count	Retry Ch...	Line Number	Include-file
1 GEN	GENERATE	0	7	0	1	0
2 ADO	ADOPT	0	7	0	2	0
3 SEI	SEIZE	0	7	0	3	0
4 ADV	ADVANCE	1	7	0	4	0
5 REL	RELEASE	0	6	0	5	0
6 ASS	ASSEMBLE	0	6	0	6	0
7 QUE	QUEUE	1	2	0	8	0
8 SEI	SEIZE	0	1	0	10	0
9 DEP	DEPART	0	1	0	11	0
10 ADV	ADVANCE	0	1	0	12	0
11 REL	RELEASE	0	1	0	13	0
12 TER	TERMINATE	0	1	0	14	0

Рис. П4.7

Результат моделирования, полученный в виде стандартного отчета при выполнении программы с командой START 100, показан на рис. П 4.8. На момент завершения моделирования 100 заявок (объединенных пакетов

по 3) завершили обслуживание – через оператор №11 TERMINATE прошли 100 заявок; одна одиночная заявка поступила в первое устройство и обслуживания не завершила (1 в поле CURRENT COUNT оператора №4 ADVANCE). В операторе №6 ASSEMBLE началось формирование очередного, 101-го пакета, 2 одиночные заявки сверх уже использованных для пакетирования 300 задержаны до прихода очередной 3-ей, необходимой для формирования пакета.

LABEL	LOC	BLOCK TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	303		0	0	
	2	ADOPT	303		0	0	
	3	SEIZE	303		0	0	
	4	ADVANCE	303		1	0	
	5	RELEASE	302		0	0	
	6	ASSEMBLE	302		1	0	
	7	QUEUE	100		0	0	
	8	SEIZE	100		0	0	
	9	DEPART	100		0	0	
	10	ADVANCE	100		0	0	
	11	RELEASE	100		0	0	
	12	TERMINATE	100		0	0	

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
USTR1	303	0.997	1.000	1	303	0	0	0	0
USTR2	100	0.987	3.000	1	0	0	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)	RETRY
02	1	0	100	100	0.000	0.000	0.000	0

Рис. П 4.8

*Учебное издание*

ШЕСТОПАЛОВА Ольга Евгеньевна

## МОДЕЛИРОВАНИЕ

Учебно-методический комплекс  
для студентов специальностей

1-40 01 01 «Программное обеспечение информационных технологий»  
и 1-40 02 01 «Вычислительные машины, системы и сети»

Компьютерная вёрстка и техническое редактирование *В. В. Рудак*

Дизайн обложки *В. А. Виноградовой*

---

Подписано в печать 28.09.09. Формат 60×84 1/16. Бумага офсетная. Гарнитура Таймс.  
Ризография. Усл. печ. л. 17,64. Уч.-изд. л. 17,14. Тираж 85 экз. Заказ № 1640.

---

Издатель и полиграфическое исполнение:  
учреждение образования «Полоцкий государственный университет»

ЛИ № 02330/0548568 от 26.06.09      ЛП № 02330/0494256 от 27.05.09

211440 г. Новополоцк, ул. Блохина, 29