МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

МАИ

ИЗДАТЕЛЬСТВО

УЧЕБНОЕ ПОСОБИЕ

Г.А. ЗВОНАРЕВА
Е.Н. КЛИМОВЕЦ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

МОСКВА • 1994

ГОСУДАРСТВЕННЫЙ КОМИТЕТ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПО ВЫСШЕМУ ОБРАЗОВАНИЮ

МОСКОВСКИЙ
ОРДЕНА ЛЕНИНА И ОКТЯБРЬСКОЙ РЕВОЛЮЦИИ
АВИАЦИОННЫЙ ИНСТИТУТ имени СЕРГО ОРДЖОНИКИДЗЕ

Г.А. ЗВОНАРЕВА Е.Н. КЛИМОВЕЦ

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Учебное пособие

Под редакцией проф. О.М. Брехова

Утверждено
на заседании редсовета
7 июня 1993 г.

Москва
Издательство МАИ
1994

Звонарева Г.А., Климовец Е.Н. Имитационное моделирование вычислительных систем : Учеб. пособие/ Под ред.О.М. Брехов: — М. : Изд-во МАИ, 1994. — 44с. : ил.

Цель пособия — ознакомить студентов с системой GPSS как средством имитационного моделирования вычислительных систем. Логика моделирования рассмотрена на конкретных примерах из области вычислительной техники, приведены практические рекомендации по моделированию работы вычислительных систем в среде общецелевой системы моделирования GPSS.

Рецензенты: Н.И. Дементьев, Г.Н. Воробьев

ISBN 5-7035-1163-1

ОБЩИЕ СВЕДЕНИЯ О ЯЗЫКЕ МОДЕЛИРОВАНИЯ GPSS

1.1. Принципы моделирования в среде GPSS

GPSS является общецелевой системой моделирования. В качестве программы, интерпретирующей модель на языке GPSS используется так называемый интерпретатор GPSS. Модель на GPSS представляет собой совокупность блоков. Передача управления от блока к блоку и их выполнение проходят при движении динамических объектов, называемых транзактами. Транзакты вводятся и выводятся из модели в соответствии с условиями моделирования. В начальный момент моделирования в модели нет ни одного транзакта. В общем случае в модели может существовать большое число транзактов, однако в один и тот же момент времени в модели продвигается только один транзакт. При продвижении транзактов блока (оператора) к блоку выполняются действия, предписанные данному блоку, т.е. происходит обращение к соответствующей подпрограмме. Движение транзакта происходит до тех пор, пока транзакт не входит в блок, функцией которого являются вывод транзакта из модели, задержка транзакта на некоторое время либо отказ транзакту во входе до изменения определенных условий. При прекращении движения одного транзакта в модели начинается продвижение следующего. Соответствие между транзактами и элементами моделируемой системы устанавливает разработчик в процессе создания модели.

В процессе моделирования в системе в хронологической последовательности совершаются некоторые события. В целью фиксирования правильной временной последовательности наступления событий в интерпретатор включена специальная переменная, которая называется таймером модельного времени. Таймер регистрирует только целые значения. Соответствие между единицей модельного времени и моделируемым интервалом устанавливается разработчиком модели (в зависимости от условий моделируемой задачи это может быть секунда, минута, час, год и т.п.). Заметим, что иногда для моделирования одной минуты модельного времени требуется несколько часов реального машинного времени и наоборот.

1.2. Основные сведения о формате записей блоков в GPSS-модели

Строку записи можно условно разделить на несколько полей:

1. *Поле имени.* Это поле занимает со 2-й по 6-ю позиции строки. В нем указываются метки операторов либо имена функций и переменных. Символические имена и метки составляются из алфавитно-циф-

ровых символов, причем их число не должно превышать пяти, к тому же первые три символа должны быть алфавитными. Приведем примеры символический имен:

LBL1
JOB2
FLO2M
PROC

2. *Поле оператора.* Данное поле располагается с 8-й по 18-ю позицию строки. Некоторые блоки, например, GATE, TEST, LOGIC, имеют вспомогательные операнды, которые отделяются пробелом от оператора.

3. *Поле операндов.* Операнды блоков дают информацию, специфичную для действия данного блока. Число операндов каждого блока зависит от типа блоков. Ни один из блоков не использует более семи операндов. Операнды (кроме вспомогательного) записываются с 19-й позиции и разделяются запятыми. Пробелы между операндами не допускаются. Комментарии могут располагаться на отдельной строке. В этом случае она должна начинаться с символа *.

1.3. Блок GENERATE

Посредством блок GENERATE транзакты вводятся в модель. Операнды A и B блока GENERATE задают величину интервала времени, через который транзакты поступают в модель. При равномерном распределении интервалов прихода операнд A определяет среднее время между поступлениями транзактов в модель, а операнд B задает половину поля допуска равномерно распределенного интервала.

Пример: GENERATE 8,2

Операнд A равен 8, операнд B равен 2.

Интервал времени, через который транзакты вводятся в модель, является случайным числом и с равной вероятностью равен 6, 7, 8,9 и 10 тактам модельного времени. Если какой-либо из операндов отсутствует, то по умолчанию его значение равно 0.

Пример: GENERATE *5.*

Операнд A равен *5*, операнд B равен 0.

Интервал времени поступления транзактов в модель является детерминированной величиной и равен *5* тактам.

Более сложный вид распределения интервалов времени прихода транзактов в модель может быть задан при использовании функции (см. блок FUNCTION) языка GPSS.

Операнд C блока GENERATE задает смещение интервалов поступления транзактов. Смещение интервалов времени определяет момент времени, когда первый транзакт поступит в модель, все последующие транзакты поступают в соответствии с операндами A и B. Е если операнд C равен 0, момент поступления первого транзакта в модель задается операндами A и B.

Пример: GENERATE 6,2,10

Первый транзакт поступит в модель в 10-с такте модельного времени. Последующие транзакты поступят в модель через интервалы времени, подчиненные равномерному закону распределения случайных величин со средним временем прихода транзактов, равным 6 тактам и половиной поля допуска, равной 2 тактам.

Операнд D определяет число транзактов, которые могут войти в модель. При поступлении в модель числа транзактов, равного D, генерация транзактов прекращается.

Пример: GENERATE 7,1,12,5

Всего в модель поступит 5 транзактов, первый транзакт поступит в 12-м такте модельного времени, а оставшиеся 4 транзакта — через 71 такт. Операнд E задает уровень приоритета транзактов, входящий в модель. Чем больше значение операнда E, тем выше уровень приоритета транзактов. Операнд E может принимать значения от 0 до 127 включительно.

Пример: GENERATE 5,1,10,6,10

Всего в модель будет введено 6 транзактов с уровнем приоритета 40. Первый транзакт поступит в 10-м такте, остальные транзакты — через 51 такт.

Пример: GENERATE ,,,1,10

Операнды A, B и C отсутствуют. В модель поступит один транзакт с уровнем приоритета, равным 10. Поскольку отсчет модельного времени начинается с первого такта, а операнды A, B и C по умолчанию равны 0, то транзакт поступит в модель в первый такт модельного времени.

1.4. Блоки TERMINATE и START

Блок TERMINATE предназначен для вывода транзактов из модели. Транзакты,: попадая в данный блок, удаляются из модели. В модели может быть несколько блоков TERMINATE, каждый из которых может использоваться без операндов или иметь операнд А. Операнд А блока TERMINATE задает величину, которая вычитается из специального счетчика завершений. Начальное значение счетчика завершений задается операндом А блока START. Всякий раз, как только транзакт входит в блок TERMINATE с операндом А, из счетчика завершений вычитается величина, равная операнду А. При равенстве 0 счетчика завершений моделирование прекращается (см. примеры из раздела 2).

```
Пример: GENERATE 4000
        TERMINATE 1
        START 1
```

Данная связка блоков может использоваться для завершения моделирования про истечении 4000 тактов модельного времени. Блоком GENERATE в 4000-м такте сгенерируется транзакт, который далее поступает на блок TERMINATE с операнда А, равным 1. При попадании в бок TERMINATE транзакт выводится из модели, а из счетчика завершений вычитается 1. Поскольку операнд А блока START равен 1, то начальное значение счетчика завершений равно 1. После вычитания из него значения операнда А блока TERMINATE счетчик завершений примет нулевое значение и моделирование завершится.

1.5. Блоки SEIZE и RELEASE

Средства языка GPSS в процессе моделирования позволяют описывать поведение элементов, предназначенных непосредственно для обслуживания заявок. Такими элементами являются приборы (устройства), которые в любой момент времени могут обслуживать только одну заявку. Для занятия прибора используется блок SEIZE. Имя может быть либо числовым, либо символическим. Максимально допустимое числовое значение имени определяется объемом памяти, доступной модели. Так, при использовании 64 Кбайт памяти в модели может быть до 35 устройств, а при использовании 256 Кбайт памяти — до 300 устройств. При использовании символического имени необходимо помнить о том, что ряд версий языка GPSS позволяет использовать идентификаторы, состоящие из 5 латинских позволяет использовать идентификаторы, состоящие из 5 латинских бука и цифр, причем первые три символа должны быть буквами. При необходимости

занять некоторый прибор транзакт сможет войти в блок SEIZE , если данный прибор не занят другим транзактом. ДО освобождения прибора транзакт будет ожидать на входе в блок SEIZE , образуя некоторую очередь. Прибор освобождается, попадая в блок RELEASE. Если на входе блока SEIZE образована очередь из транзактов, претендующих на занятие прибора, то после его освобождения прибор будет занят транзактом с наивысшим приоритетом. При наличии очереди к блоку SEIZE, состоящей из равноприоритетных транзактов, первым поступит на обслуживание транзакт, который раньше всех пришел. При отказе во входе в блок SEIZE транзакте остается в блоке Л, из которого он пытался войти в блок SEIZE.

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN
1	.292	62	9.435
2	.316	60	10.566
3	.367	60	12.266
PAM	.990	180	11.011

SEIZING TRANS. NO.	PREEMPTING TRANS. NO.
4	
3	

Рис. 1

В процессе моделирования для каждого прибора собирается стандартная статистика, которая распечатывается в конце моделирования. На рис. 1 приведена распечатка стандартной статистики по приборам. В колонке FACILITY распечатываются идентификаторы приборов, используемые в процессе моделирования. В колонке AVERAGE UTILIZATION для каждого из приборов фиксируется доля времени, в течение которого прибор был занят. NUMBER ENTRIES определяет число входов в каждый прибор. В колонке AVERAGE TIME/TRAN распечатывается среднее время обслуживания одной заявки для каждого прибора. SEIZING TRANS.NO фиксирует номер транзакта, находившегося на обслуживании в приборе в момент окончания моделирования. Из приведенной статистики следует, что в процессе моделирования использовалось 4 прибора. ТРи прибора имели числовые имена от 1 до 3, а один прибор имел символическое имя PAM-180. Среднее время обслуживания одной заявки на приборе 1, 2, 3 СОСТАВИЛО 9.435; 10.566; 12.266 тактов соответственно, а для прибора PAM — 1170 —11 тактов модельного времени.

1.6. Блок ADVANCE

Блок ADVANCE предназначен для задержки продвижения транзакта в течение некоторого времени. Операнды A и B блока ADVANCE задают интервал задержки, подчиненный равномерному закону распределения случайных величин: Операнд A предназначен для определения среднего времени задержки, а операнд B задает половину поля допуска равномерно распределенного интервала времени задержки.

Пример: ADVANCE 4,1

При попадании в данный блок транзакт будет задержан на время, подчиненное равномерному закону распределения случайных величин со средним временем, равным 4 тактам, и половинной поля допуска, равной 1 такту.

Пример: ADVANCE 8

При прохождении через указанный блок транзакт задерживается на 8 тактов модельного времени. Для задания более сложных видов распределения времен задержки необходимо использовать функции языка GPSS. Без указания операндов блок ADVANCE используется в качестве фиктивного блока. Часто блок ADVANCE используется совместно с блоками SEIZE и RELEASE для описания работы прибора.

```
Пример: SEIZE AAA
        ADVANCE 10
        RELEASE AA
```

Транзакт занимает прибор AAA и в течение 10 единиц модельного времени будет обслуживаться прибором. По истечении времени обслуживания транзакт покидает прибор AAA. Следует отметить: несмотря на то, что связка блоков, рассмотренная в примере,Д используется достаточно часто, блок ADVANCE можно располагать не только между блоками SEIZE и RELEASE.

1.7. Блоки QUEUE и DEPART

При моделировании часто возникают ситуации, когда заявка не может быть обслужена из-за занятости некоторого ресурса ив связи с этим помещается в очередь до освобождения ресурса. При описании блоков SEIZE, RELEASE отмечалось, что если прибор занят, то транзакт не может войти в блок SEIZE и остается в предшествующем блоке, т.е. происходит имитация образования очереди. При этом заявка, пытавшаяся занять прибор, присоединяется к очереди, в течение некоторого времени ожидает своей очередности для занятия прибора и затем занимает прибор. ДЛя регистрации очереди используются блоки QUEUE и DEPART. Блок QUEUE предназначен для регистрации присоединения к очереди. Операнд A блока QUEUE определяет имя очереди. Имена очередей так же, как и имена приборов могут быть числовыми или символическими. Максимальное число регистраторов очередей для объема памяти 64 Кбайт равно 70, а для объема памяти 256 Кбайт равно 300. Блок DEPARTиспользуется для регистрации ухода из очереди. Операнд A блока DEPART определяет имя очереди. Необходимо обратить внимание на то, что блоки QUEUE и DEPART используются не для образования очередей, а для сбора статистики по очередям. В конце моделирования по очередям распечатывается стандартная статистика. В колонке MAXIMUM CUNTENTS фиксируется наибольшее значение содержимого очереди за время моделирования. Колонка TOTAL ENTRIES предназначена для подсчета общего числа заявок, вошедших в очередь. В колонке ZERO ENNTRIES подсчитывается число заявок, для который время пребывания в очереди равнялось 0 (нулевые входы). PERCENT ZEROS соответствует проценту нулевых входов от общего числа заявок, находившихся в очереди. Среднее время, проведенное одной заявкой в очереди, включая нулевые входы, фиксируется в колонке AVERAGE TIVE/TRANS. Текущее содержимое очереди на момент завершения моделирования определяется в колонке CURRENT CONTENS.

Из приведенной статистики следует, что в процессе моделирования использовались три регистратора очереди с именами SSS, AAA и BBB. Максимальное содержимое очередей равнялось 11, 6 и 5 заявкам соответственно. Среднее содержимое очередей составило 4.699; 2.699; 2 заявки. Суммарное число входов в очередь SSS равнялось 15, из них одна заявка имела нулевое время ожидания в очереди. Для очереди BBB число входов равнялось 5, все заявки ожидали в очереди своего обслуживания. Процент нулевых входов для очереди SSS составил 6,6%, очереди AAA — 9,9%, а для очереди BBB — 0 %. В среднем одна заявка в очереди SSS ожидала 3,133 такта, без учета нулевых входов — 3,357 тактов. В очереди AAA с учетом нулевых входов среднее время ожидания с учетом и без учета нулевых входов совпадает и составляет блоком SEIZE для регистрации очереди и устройству.

```
Пример: QUEUE SSS
        SEIZE AAA
        DEPART SSS
```

QUEUE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES
SSS	11	4.699	15	1
AAA	6	2.699	10	1
BBB	5	2.000	5	

PERCENT ZEROS	AVERAGE TIME/TRANS	*AVERAGE TIME/TRANS	TABLE NUMBER	CURRENT CONTENTS
6.6	3.133	3.357		11
9.9	2.699	3.000		6
.0	4.000	4.000		5

Рис. 2

Очередь на входе прибора AAA регистрируется с помощью регистратора очереди SSS. При этом приборе AAA транзакты не могут войти в блок SEIZE и образуют на входе некоторую очередь. Блок QUEUE регистрирует момент присоединения транзакта в очередь. После освобождения прибора AAA некоторым транзактом очередной транзакт из очереди занимает прибор и проходит через блок DEPART. При этом фиксируется время ухода транзакта и очереди. Моделирование систем с использованием приборов и регистраторов очередей рассматривается в разд. 2.1.

1.8. Блок PRIORITY

При вводе транзакта в модель его уровень приоритета определяется операндом E блока GENERATE. Блок PRIORITY позволяет изменять приоритет транзакта в процессе моделирования. Операнд A блока PRIORITY определяет уровень приоритета транзакта, проходящего через данный блок. При прохождении через блок PRIORITY транзакту назначается новый уровень приоритета и в цепи текущий событий он занимает последнее место среди транзактов с таким же уровнем приоритета. Поэтому при повышении уровня приоритета некоторого транзакта блок PRIORITY целесообразно ставить перед блоком ADVANCE. Тогда, попадая в блок ADVANCE, транзакт из цепи текущих событий переводится в цепь будущих событий в соответствии с новым значением приоритета. При достижении модельного времени, с учетом задержки в блоке ADVANCE, транзакт из цепи будущих событий переводится в цепь текущий событий в порядке расположения транзактов в цепи будущих событий.

Пример: PRIORITY 20

При прохождении транзакта через данный блок уровень его приоритета будет равен 20.

1.9. Блок TRANSFER

В данном разделе будет рассмотрено использование блока TRANSFER в двух режимах: в режиме безусловной передачи и в режиме статистической передачи.

В режиме безусловной передачи блок TRANSFER используется в случае, когда необходимо передать транзакт в блок, отличный от последующего. Операнд A блока TRANSFER в режиме безусловной передачи не используется, а операнд B определяет метку блока, в который необходимо передать транзакт. При использовании метода необходимо помнить о том, что ряд версий языка GPSS позволяет использовать идентификаторы, состоящие из 5 латинских букв и цифр , причем первые три символа должны быть буквами.

Пример: TRANSFER, LAB

1 транзакт будет направлен в блок с меткой LAB1.

Блок TRANSFER в режиме статистической передачи используется в случае, когда необходимо передавать транзакт в один из двух блоков случайным образом. Операнд A блока TRANSFER, используемого в режиме статистической передачи, определяет частоту передачи транзакта в блок, метка которого указана в операнде C. В противном случае транзакт переходит в блок, метка которого указана в операнде B.

Пример: TRANSFER .3,LAB1, LAB2

С частотой 0.3 транзакт будет переходить в блок с меткой LAB2 и с частотой, равной 0.7, транзакт будет переходить в блок с меткой LAB1. Если операнд B опущен, то это означает , что транзакт с частотой 1 минус значение операнда A будет переходить в следующий блок за блоком TRANSFER. Операнд C опущен быть не может.

Пример: TRANSFER .3,,LAB2

Транзакт с частотой 0.3 перейдет по метке LAB2 и с частотой , равной 0.7, поступит в блок, следующий за TRANSFER. Необходимо отметить, что первым символом операнда A блока TRNASFER в режиме статистической передачи должна быть десятичная точка. Остальная часть операнда A соответствует частоте, с которой транзакт переходит в блок, метка которого указана в операнде C.

1.10. Блок PRINT

Наряду со стандартной печатью в конце моделирования с помощью блока PRINT имеется возможность распечатывать статистические данные по заданным элементам в процессе моделирования. Операнды A и B задают наименьшее и наибольшее значения номеров элементов, то информация распечатывается по всем элементам данного типа. Операнд C определяет мнемоническое обозначение типа распечатываемых элементов. Так, обозначения F м Q используются для печати статистики по приборам и регистраторам очередей соответственно: N — счетчик входов в блок, W — счетчик содержимого блоков, B — счетчик входов и содержимого блоков, C — таймер модельного времени. В приложении 1 приводятся мнемонические обозначения для различных типов элементов, наиболее часто используемых в блоке PRINT.

Пример: PRINT,,F

При прохождении транзакта через рассматриваемый блок будет распечатана стандартная статистика по всем приборам, используемым в модели. Часто блок PRINT используется при отладке модели, когда через заданные интервалы времени необходимо проанализировать изменение различных типов элементов. (См. разд.2.1).

```
Пример: GENERATE ,,,1
        MMM PRINT ,, B
        PRINT ,,Q
        ADVANCE 1
        TRANSFER, MMM
```

Рассматриваемый фрагмент предназначен для печати статистики по очередям и по блокам модели каждый такт модельного времени. В первый такт модельного времени будет сгенерирован один транзакт, который, пройдя через блоки PRINT, вызовет печать соответствующей статистики в первом такте. Далее транзакт попадает в блок ADVANCE и во втором такте транзакт, вновь пройдя через блоки PRINT, вызовет печать статистики. Статистика по очередям и блокам модели будет распечатываться каждый такт до завершения моделирования.

1.11. Блок TEST

С помощью блока TEST производится сравнение двух стандартных числовых атрибутов. Стандартные числовые атрибуты — это набор данных, доступных в процессе моделирования. Имя стандартного

числового атрибута состоит из двух частей: из имени группы и имени элемента внутри группы. Первая часть имени определяет тип элемента (т.е. устройство, очередь и т.п.). Вторая часть определяет конкретный элемент группы (т.е. конкретное устройство, очередь и т.п.). Имя элемента внутри группы может быть, как указывалось вышел, либо числовым, либо символическим. Если используется символическое имя, то между групповым именем и символическим ставится знак $, если используется числовое имя, то групповое и символическое имя записываются подряд без разделителя.

В приложении 2 приведены наиболее часто используемых стандартные числовые атрибуты.

Пример: FC — число занятий 2-го прибора
Q$SSS — текущее значение длины очереди SSS

Операндами A и B блока TEST являются имена сравниваемых стандартных числовых атрибутов. в блоке TEST используется вспомогательный оператор, который может принимать следующие значения:

G —A больше B;
GE —B больше или равно B;
E —B равно B;
NE — A не равно B;
LE — A меньше или равно B;
L — A меньше B.

Вспомогательный оператор записывается через пробел после слова TEST. Блок TEST может работать либо в режиме отказа, либо в режиме условной передачи. При работе блока TEST в режиме отказа операнд C не используется, а присутствуют только операнды A и B.

В случае выполнения оператора отношения транзакт, поступающий в блок TEST, проходит в следующий блок. При невыполнении условия заданным вспомогательным оператором транзакт задерживается на входе блока TEST в предшествующем блоке.

Пример: TEST GE QAAA, QBBB

Производится сравнение текущего содержимого очереди AAA и очереди BBB. Если текущее содержимое очереди AAA больше или равно текущему содежимому очереди BBB, то транзакт проходит в следующий блок, в противном случае транзакт задерживается в предыдущем блоке до момента выполнения условия.

При работе блока TEST в режиме условной передачи используется операнд C. Операнд C блока TEST определяет метку блока, на который перейдет транзакт в случае невыполнения условия, заданного

оператором отношения. При выполнении условия транзакт переходи в блок, следующий за блоком TEST.

Пример: TEST L FC1, FC3, LAD

Если число занятий 1-го прибора меньше числа занятий 3-го прибора, то транзакт переходит в следующий блок. В противном случае транзакт переходит в блок с меткой LAB.

1.12. Блок SAVEVALUE

Блок SAVEVALUE используется для изменения значения сохраняемой величины в процессе моделирования. Сохраняемые величины существуют на протяжении всего времени моделирования и позволяют транзактам обмениваться данными. Различаютм полусловные и полнословные сохраняемые величины. Полусловные сохраняемые величины могут изменяться от -32768 до +32767, а полнословные — от -2147483648 до +214748347. При объеме памяти 64 Кбайт максимальное значение полусловных величин рано 500, полнословных величин равно 1000. Сохраняемые величины являются целыми числами. Групповое имя полусловных сохраняемых величин обозначается XH, а полнословных — обозначаеся X.

Пример: XH5 - полусловная сохраняемая величина с числовым именем 5; X$NUM — полнословная сохраняемая величина с символическим именем NUM.

Операнд A блока SAVEVALUE определяет имя изменяемой сохраняемой величины, операнд B определяет величину, используемую в процессе модификации. Операнд C является необязательным. Если операнд C отсутствует, то по умолчанию подразумевается, что сохраняемая величина полнословная, если в качестве операнда C используется символ H, то это означает, что сохраняемая величина полусловная.

Пример: SAVEVALUE 10, HX$DATA, H

При прохождении транзакта через данный блок десятая полусловная сохраняемая величина примет значение, равное полусловной сохраняемой величине с символическим именем DATA.

Пример: SAVEVALUE NUM, -20

Полнословная сохраняемая величина с символическим именем NUM примет значение -20 после прохождения транзакта через данный блок. Блок SAVEVALUE может использоваться в режиме приращения

либо уменьшения. В режиме приращения (уменьшения) значение сохраняемой величины, указанной в операнде A, увеличивается (уменьшается) на величину, указанную в операнде B. Для обозначения режимов увеличения или уменьшения после операнда A до запятой, разделяющей операнды A и B, ставится, соответственно, знак «плюс» или « синус».

Пример: SAVEVALUE DATA 1 +, 15, H

Полусловная величина с символическим именем DATA1 увеличивается на 15 при прохождении транзакта через данный блок.

Пример: SAVEVALUE 10-, Q$SSS

Полнословная величина 10 уменьшается на текущее значение очереди с символическим именем SSS при прохождении транзакта через рассматриваемый блок.

1.13. Блок INITIAL

С помощью блока INITIAL до начала моделирования отдельным сохраняемым величинам могут быть назначены начальные значения, отличные от нуля. В поле операндов блока INITIAL через разделитель / задаются пары величин. Первой величиной в паре является имя сохраняемой величины, второй величиной — назначаемое начальное значение. В модели разрешается использовать несколько блоков INITIAL , которые рекомендуется располагать до блока GENERATE.

Пример: INITIAL X8,-25/XH$NNN, 10

До начала моделирования 8-й полнословной величине присвоено значение -25, полусловной сохраняемой величине с символическим именемNNN будет присвоено значение 1-.

1.14. Блок SPLIT

Блок SPLIT предназначен для расщепления транзактов. Это является вторым способом ввода транзактов в модель (первый способ был рассмотрен при описании блока GENERATE) . При входе транзакта в блок SPLIT в модель дополнительно вводятся один или несколько транзактов. Транзакт, входящий в блок SPLIT, называется «родителем», дополнительно вводимые транзакты в модель называются «потомками». Операнд A блока SPLIT указывает число дополнительно вводимых транзактов, т.е. число «потомков». Операнд B блока SPLIT

определяет метку блока, в который направляются «потомки». Транзакт «родитель», после выхода из блока SPLIT направляется в следующий блок. Потомки имеют тот же уровень приоритета, что и транзакт «родитель». Число параметров и их тип у «потомков» также совпадает с числом параметров и типом транзакта «родителя».

Пример: SPLIT 2, LLL

После входа транзакта «родителя» в блок SPLIT дополнительно в модель будут введены еще два транзакта, которые поступят в блок с меткой LLL.

В блоке SPLIT имеется возможность упорядочения транзактов. Это достигается с помощью операнда С. Операнд С блока SPLIT определяет номер параметра транзакта «родителя» и транзактов «потомков», в котором транзакты будут упорядочены. Параметр, указанный в операнде С, для транзакта родителя будет увеличен на 1, а для каждого из потомков соответственно на 2, на 3 и т.д.

Пример: SPLIT 3, LLL, 1

Предположим, что транзакт «родитель» до входа в блок SPLIT имел первый параметр, равный 0. Тогда, после выхода из блока SPLIT, первый параметр у транзакта родителя станет равным 1, у первого «потомка» — 2, у второго «потомка» —3, у третьего «потомка» —4. Все потомки поступят на блок с меткой LLL. Параметр D блока SPLIT задает число параметров, которое должно быть у каждого «потомка». Если «потомки» имеют большее число параметров, чем транзакт «родитель», то дополнительным параметрам присваивается нулевое значение.

Работа блока SPLIT с использованием упорядочения транзактов рассматривается в разд.2.2.

1.15. Блоки LINK, UNLINK

Язык GPSS содержит в своем арсенале такой элемент, как цепь пользователя. Цепь пользователя представляет собой то место, где могут находиться транзакты, присутствующие в модели, но не находящиеся ни в цепи текущих, ни в цепи будущих событий. Другими словами, транзакты можно временно заблокировать в цепи пользователя. В модели может быть несколько цепей пользователя. Каждая цепь пользователя может иметь либо численное, либо символическое имя. Допустимое число различных цепей пользователя зависит от объема машинной памяти.

Возможность поместить транзакт в цепь пользователя обеспечивается блоком LINK. Блок LINK может использоваться в безусловном и условном режимах.

В начале рассмотрим использование блока в безусловном режиме. В данном случае блок обладает двумя операндами: А и В. А — имя цепи пользователя, к которой присоединяется транзакт. Операнд В задает место, которое входящий транзакт занимает в цепи пользователя. В качестве операнда В могут быть использованы следующие обозначения:

FIFO — транзакт помещается в конец цепи;

LIFO — транзакт помещается в начало цепи;

P_j, где j — целое число, в интервале от 1 до 100, транзакты помещаются в цепь пользователя в порядке возрастания их параметра P_j.

Заметим, что транзакты, входящие в блок LINK, работающий в безусловном режиме, всегда помещаются в цепь пользователя. Рассмотрим пример использования данного блока:LINK BUF, P3

В данном случае блок LINK используется в безусловном режиме. Входящие транзакты присоединяются к цепи пользователя с именем BUF в порядке возрастания их третьего параметра. Так, при вхождении в блок транзактов со значениями третьего параметра 3, 29 и 14, они будут размещены в порядке 3, 14, 29. Если теперь в блок войдет транзакт со значением P3, равным 21, он займет третье место в цепи.

В условном режиме в блоке LINK используется так же операнд С. Транзакт, входящий в блок LINK, который работает в условном режиме, будет либо присоединяться к цепи пользователя, либо направляться в блок с меткой С.

Так, например, в случае, если операнд С блока LINK является меткой блока SEIZE, то транзакт будет присоединяться к данной цепи пользователя, если соответствующий прибор занят, и будет пересылаться на блок SEIZE, если прибор свободен.

Блок UNLINK является дополнительным к LINK. Функцией блока UNLINK является вывод одного или нескольких транзактов из цепи пользователя и помещение их обратно в цепь текущих событий.

При рассмотрении этого блока следует различать транзакт-инициатор вывод (т.е. транзакт, входящий в блок UNLINK) и выводимые транзакты. Данный блок имеет следующие операнды:

А — имя цепи пользователя;

В — метка блока, в который переходит выведенный из цепи транзакт (или транзакты);

С — число выводимых транзактов (счетчик вывода), может быть константой, стандартным числовым атрибутом или символом ALL. Если в операнде С используется ALL, то всеб транзакты, принадлежащие соответствующей цепи пользователя, будут выведены. D и E используются для того, чтобы определить, с какого края цепи пользователя

следует брать выводимые транзакты. Если оба операнда пусты, транзакты выбираются из начала цепи пользователя. Если в качестве операнда D используется BACK, а операнд E не используется, транзакты выводятся из конца цепи.

Операнд F блока UNLINK может не использоваться. Если он пуст, то транзакт-инициатор переходит из блока UNLINK в следующий блок. Если операнд F используется, то в нем ставится метка блока, в который перейдет транзакт-инициатор вывода из цепи, если при попытке вывести транзакты из цепи пользователя, вывода не произошло. Такая ситуация возможна в случае, если цепь пуста.

Рассмотрим пример использования блока UNLINK:

UNLINK HOLD, 1 ,,, MET1

В данном случае из цепи пользователя с именем HOLD выводится один транзакт. Если цепь пользователя пуста, блок-инициатор вывода пересылается в блок с меткой MET1. По окончании моделирования интерпретатор выдает стандартную статистику по цепям пользователя, используемым в модели.

1.16. Параметры транзакта, блок ASSIGN

Транзакты в моделях GPSS могут иметь 100 параметров. Число параметров транзакта задается операндом f блока GENERATE, через который транзакт входит в модель. По умолчанию транзакт имеет 12 параметров. Имя параметра состоит из двух частей: группового имени, которым является буква P и номера конкретного члена этой группы, заданного с помощью целых чисел от 1 до 100. P3 означает Имя третьего транзакта. Параметры не могут иметь символических имен. Значениями параметров могут быть целые числа со знаком. Максимальное значение параметра зависит от того, определен ли он в полуслове или в полном слове. Тип задается операндом G блока GENERATE. Значение по умолчанию является H (полуслово). Для полного слова необходимо использовать символ F в качестве операнда G. В полусловах диапазон составляет от -32768 до +322767.

Начальным значением всех параметров является нуль. Как и другие стандартные числовые атрибуты, параметр можно использовать в операндах блоков и в качестве аргументов функций. В случае, когда параметры используются в качестве операндов блоков или аргументов функций, то данный параметр принимает значение данного параметра того транзакта, который в данный момент входит в данный блок. Итак, если в качестве некоторых данных требуется использовать значения параметров активного транзакта, т.е. того транзакта, который обрабатывается в данный момент. При распечатке цепей текущих и будущих событий также распечатываются и параметры транзактов, входящих в эти цепи.

Значения параметров могут назначаться и изменяться при входе транзактов в блок ASSIGN (назначить).

Блок ASSIGN содержит 2 операнда A и B. При входе транзакта в блок ASSIGN данные из операнда B фиксируются в параметре, номер которого задается операндом A. В результате предыдущее значение параметра заменяется на новое.

Пример: ASSIGN 5,67

При входе транзакта в данный блок значение его пятого параметра станет равным 67. Предыдущее значение P5 теряется. Помимо задания операндом в виде констант, существует возможность их косвенного задания через стандартные числовые атрибуты. Например, рассмотрим блок

ASSIGN P7,/ X$NNN

В данном случае параметру, номер которого равен значению параметра P7 данного транзакта, присваивается значение сохраняемой величины с именем NNN.

Выше был рассмотрен режим замещения, однако, блок ASSIGN может использоваться также в режиме приращения и вычитания.

В режиме приращения новое значение параметра вычисляется путем сложения значения операнда B со старым значением параметра. В режиме вычитания новое значение параметра вычисляется путем вычитания значения операнда B из старого значения. Режимы приращения и вычитания отмечаются указанием знаков + или - соответственно, стоящих перед запятой, разделяющей операнды A и B.

Пример использования блока в режиме вычитания:

ASSIGN P2-, *5*

При входе транзакта в этот блок из значения параметра, номер которого задан параметром P2, из его значения вычитается 5.

1.17. Блок MATRIX

Понятие сохраняемой величины в GPSS может быть так же применено к двумерным массивам или матрицам. Такие сохраняемые величины называются матричными или матрицами. Матричные сохраняемые величины должны быть заданы интерпретатору GPSS в начале моделирования. С этой целью используется блок MATRIX. В поле имени находится числовое или символическое имя задаваемой матрицы. В поле операции стоит слово MATRIX. В поле операндов операнд A содержит единственный символ H или X в зависимости от того, будет ли матрица состоять из полусловных или полнословных ячеек памяти соответственно. Операнды B и C являются константами,

определяющими соответственно число строк и столбцов, из которых составляется матрица.

В следующем примере матрица, имеющая символическое имя TAB, определена полусловной и имеет четыре строки и восемь столбцов.

TAB MATRIX H,4,8

Оператор MATRIX должен быть использован до первой ссылки на рассматриваемую матрицу. Каждый элемент матрицы занимает место в определенной строке и в определенном столбце данной матрицы. Для обозначения полусловного или полнословного типа матрицы употребляют соответственно MH и MX. Далее следует символическое имя рассматриваемой матрицы (в случае использования символического имени между MX и MH необходимо помещать символ $). Затем указывается номер строки и номер столбца, заключенные в круглые скобки.

Так, MH 4(5,8) — обращение к элементу полусловной матрицы 4, стоящему в пятой строке и восьмом столбце. Номера строк и столбцов могут быть так же заданы косвенно. Например, MX $ TAB (P4, P8) — ссылка на элемент полнословной матрицы с символическим именем TAB. Номера строки и столбца этого элемента находятся соответственно из четвертого и восьмого параметров обрабатываемого в данный момент транзакта. Число сохраняемых величин зависит от размера доступной машинной памяти. Элементы матриц могут быть использованы как стандартные числовые атрибуты для косвенного ввода данных.

Перед началом моделирования значения всех элементов матрицы устанавливаются равными нулю. Пользователь может присвоить некоторым элементам ненулевые значения при использовании блока INITIAL.

Если ряд последовательных элементов в столбце должен принять одно и то же начальное значение, то может быть использована запись вида AMH 5(10-15,8),1. Такая запись приведен к присвоению шести элементам матрицы 5 значения 1.

Пример: INITIAL MX2 (1,4),-157/MH/$/SVN (1,1),33

В данном блоке задается начальное значение элемента полнословной матрицы 2, равное 164, а также элементу полусловной матрицы SVN, расположенному в первом столбце, и первой строке присваивается значение 33.

1.18. Блок MSAVEVALUE

Значение одного элемента матрицы изменяется при прохождении транзакта через блок MSAVEVALUE (сохранить значение элемента матрицы). Блок имеет операнды A, B, C, D и E. При входе транзакта в блок MSAVEVALUE величина операнда D присваивается элементу

матрицы, стоящему в строке B и столбце C в A-матрице. Операнд E обеспечивает задание рассматриваемого типа. По умолчанию матрица подразумевается полнословной.

Блок MSAVEVALUE может использоваться в режиме приращения и вычитания, а также замещения.

Режим приращения и вычитания задается помещением соответственно знака плюс или минус непосредственно перед запятой, отделяющей операнды A и B.

Ниже рассмотрим примеры использования блока:

MSAVEVALUE 1,5,6,8

MSAVEVALUE BOT+,2,3,P3

В первом примере элементу матрицы 1. расположенному в пятой строке и шестом столбце, присваивается значение 8. Во втором примере к значению элемента матрицы BOT, расположенному во второй строке и третьем столбце, прибавляется значение третьего транзакта, входящего в данный блок.

1.19. Блок LOOP

Блок LOOP служит для организации цикла. В операнде A блока стоит номер так называемого параметра цикла. Когда транзакт входит в этот блок, указанный параметр уменьшается на единицу, а затем проверяется, равно ли его значение нулю. Если нет, то транзакт переходит к блоку, метка которого задана в операнде B. Когда параметр цикла равен нулю, транзакт переходит в следующий блок.

Блок LOOP может быть использован только для убывающего счета, а именно, для уменьшения значения параметра. Значение параметра всегда уменьшается на единицу. Если транзакт входит в блок LOOP с параметром цикла, меньшим или равным нулю, происходит указание на ошибку и моделирование прекращается.

Пример: LOOP 5,LBL

Данный блок организует цикл по пятому параметру, осуществляя передачу управления к блоку с меткой LBL до тех пор, пока значение параметра P5 не станет равным нулю.

1.20. Логические переключатели. Блок LOGIC

Такие элементы GSSP, как логические переключатели могут использоваться для моделирования управления событиями в системе. В модели может существовать условие прохождения транзакта в тот или иной блок. Например, при моделировании конвейерной вычислитель-

ной системы, обрабатывающей линейно зависимые команды, возникает ситуация, когда очередная команда может поступить на обработку в первую ступень конвейера только по завершении обработки на последней ступени конвейера команды, от которой она линейно зависит. Другими словами, при условии, если последняя ступень конвейера обработала команду, результаты которой используются в последующей, эта линейно зависимая команда может поступить на обработку в конвейер.

Аналогичные проблемы возникают при моделировании буфера предварительной выборки заявок из памяти, будь то буфер команды или данных. Действительно, генерируемые памятью заявки могут поступать в буфер ограниченной емкости лишь в случае, когда он еще не заполнен. Итак, условием поступления очередной заявки в буфер является наличие в нем свободных мест.

Реализация подобных условий средствами GPSS может осуществляться двояко.

Первый подход реализуется с помощью использования некоторой сохраняемой величины, принимающей различные значения, в частности 1 либо 0 («открыто» или «закрыто») и блока, который, в зависимости от значения сохраняемой величины, пропускает транзакт в следующий (либо определенный меткой) блок или нет. Такой подход рассмотрен в главе 2 при моделировании буфера команд.

При втором подходе используются логические переключатели. Они являются двухпозиционными, имея два различных положения — «установлено» и «сброшено». Установка логических переключателей может изменяться в процессе моделирования для отображения изменяющихся условий в модели. Состояния логических переключателей можно проверять и использовать для оказания влияния на движение транзактов в модели.

При разработке модели следует учесть, что в использовании для моделирования управления логических переключателей вместо сохраняемых величин существует ряд преимуществ. Во-первых, время выполнения, требуемое для изменения и проверки положения логических переключателей меньше, чем время для проведения тех же процедур с сохраняемыми величинами.

Во-вторых, программисту психологически проще представлять управление в терминах «установлено-сброшено», чем в терминах числовых кодов, рассматриваемых при использовании сохраняемых величин.

Подобно приборам, таблицам и сохраняемым величинам логические переключатели должны иметь числовое или символическое имя.

Перед началом моделирования все логические переключатели устанавливаются в состояние «сброшено». Если того требует логика работы модели, пользователь может поместить ряд переключателей до начала моделирования в состояние «установлено» с помощью оператора INITIAL. В поле операндов оператора INITIAL перечисляются имена устанавливаемых ключей, сопровождаемые символами LS, имена разделяются косой чертой. Информация в строке не должна занимать более 71 позиций. При необходимости следует использовать несколько карт INITIAL.

Например:

INITIAL LS5/LS12 - LS22/LS $ TAG/LS $ KEY

Пример демонстрирует, что числовые и символические имена могут быть использованы в одном и том же операторе, при этом интерпретатор GPSS устанавливает соответствие между символическими именами и их числовыми эквивалентами. Так, первому встречающемуся логическому переключателю с символьным именем TAAG будет установлен числовой эквивалент 1, а переключатель KAY получит в соответствие эквивалент 2.

Таким образом, очевидно, что в данной модели следует использовать числовые имена переключателей, начиная с 3, поскольку эквиваленты 1 и 2 уже заняты.

Кроме того, из примера видно, что если номера логических переключателей, начальное состояние которых должно быть «установлено», образуют непрерывную последовательность, соответствующая запись в карте INITIAL может быть сокращена до вида LS12-LS22.

В конце моделирования автоматически распечатываются имена всех логических переключателей, которые находятся в состоянии «установлено».

Для изменения состояния логического переключателя служит блок LOGIC.

Блок имеет операнд A и вспомогательный оператор X. Формат оператора выглядит следующим образом:

LOGIC X A

Оператор X может принимать следующие значения:

R — «сброшено»;

S — «установлено»;

I — инверсия.

Состояние логического переключателя обязательно измениться лишь в том случае, если оператором отношения будет I. Для операторов R и S в момент входя транзакта в блок LOGIC соответствующий переключатель может уже находиться в нужном состоянии. Операнд A служит для обозначения имени логического переключателя.

1.21. Блок GATE

БЛОК GATE (впустить) служит для проверки состояния логического переключателя. Операнд A данного блока указывает имя про-

веряемого переключателя. Оператор записывается в следующем формате:

GATE X F, (B),

где X — вспомогательный оператор, представляющий собой логический указатель, показывающий требуемое состояние логического переключателя для того, чтобы проверка была истинной.

Логический указатель X может принимать следующие значения:

LX — проверка истинна, если переключатель «установлен», в противном случае проверка ложна;

LR — проверка истинна, если переключатель «сброшен». Операнд B может не использоваться. Если операнд B не используется, оператор работает в режиме отказа. В данном режиме переключатель закрыт, если проверка ложна. Если транзакт обнаруживает, что клапан закрыт, он задерживается в блоке, предшествующем блоку GATE.

Заблокированный транзакт находится в цепи текущих событий. В дальнейшем, если блок LOGIC изменит состояние данного логического переключателя, транзакт будет переведен в активное состояние интерпретатор откроет блок GATE.

Если операнд B используется, проверка производится в режиме условной передачи. Транзакт, поступающий в блок GATE, переходит к следующему по порядку блоку, если логический переключатель имеет заданное состояние, в противном случае он поступает в другой блок, указанный посредством задания его имени в операнде B. Логика работы блока GATE в режиме условной передачи совпадает с логикой оператора TEST. Транзакт не переходит из блока GATE к последующему блоку, если логический переключатель не имеет статуса, указанного во вспомогательном операторе блока GATE.

Примеры использования блока GATE.

1. GATE LS SYGN

В данном примере блок используется в режиме отказа. Блок закрыт, если логический переключатель SYGN «сброшен».

2. GATE LR 5, MET1

В данном случае транзакты, входящие в рассматриваемый блок, будут переходить в следующий блок, если ключ 5 «сброшен», в противном случае, т.е. если он «установлен», транзакты будут переходить в блок с меткой MET1.

Необходимо заметить, что блок GATE может быть использован для проверки занятости приборов. Таким образом, блок GATE может управлять движением транзактов в зависимости от логического состояния приборов.

Ниже представлены некоторые значения вспомогательного оператора X, используемые в данном контексте:

U — проверка занятости прибора;

NU — проверка незанятости прибора.

Пример: GATE NU PROC,MET2

Данный блок будет пропускать транзакты к следующему блоку, если прибор PROC не занят, в случае занятости прибора, транзакт переходит к блоку MET2.

Данное логическое условие может быть описано также и с помощью блока TEST:

TEST E F\$\PRO, 0, MET,

где F $ PROC — стандартный числовой атрибут, принимающий значение 0 или 1 в случае незанятости и занятости прибора PROC соответственно.

Однако использование блока GATE в данном случае более предпочтительно, поскольку с точки зрения машинного времени блок GATE экономичнее блока TEST.

1.22. Блок FUNCTION

В GPSS существует возможность задания следующих типов функций;

дискретной (тип D);

непрерывной (C);

атрибутивно-значимой (E типа).

1.22.1. Дискретные функции GPSS

При задании дискретной функции необходимо задать следующую информацию:

1. Символическое или числовое имя. Если имена числовые, то они должны быть положительными целыми числами. Если имена символические, то они состоят из алфавитно-цифровых символов (от трех до пяти), причем первые три символа должны быть алфавитными.

2. Аргумент функции. Аргументом дискретной функции является генератор случайных чисел, используемый распределением, заданным функцией. Аргумент задается в виде RN_j, где $j=1\div8$. Пользователем может быть выбран любой источник случайных чисел.

3. Число различных значений, которые может принимать случайная величина.

4. Значения переменной и соответствующие указания функций распределения.

Оператор описания функции содержит две или более строки. Первая строка содержит имя функции, указывает на генератор случайных чисел (операнд A), а также символ D, идентифицирующий тип

функции и целого числа, определяющего число различных значений, которые может принимать случайная переменная. На второй строке начинается описание значений случайной переменной и соответствующие им значения функции распределения. Они могут быть записаны на одной и более строках. Основной единицей информации описания функции является пара X_i, Y_i, где X_i — i-я суммарная частота, а Y_i — соответствующее значение случайной величины. Последовательные пары чисел разделяются знаками «/». Основные единицы должны следовать по порядку так, чтобы суммарные частоты шли в возрастающем порядке. Запись соответствующих пар чисел должна начинаться с первой позиции.

Рассмотрим пример описания дискретной функции:

```
DIM RUNCTION RN2, D5
.15,2/.35,5/.6,8/.82,9/1,12
```

Функция имеет символическое имя DIM. В качестве источника случайных чисел выступает RN2.

Случайная переменная может иметь пять различных значений. Суммарные частоты и соответствующие им пять значений записаны как пять пар чисел на следующей строке. Длина записи на строке не должна превышать 71 позицию. За последней парой разделитель не следует. При использовании функций в различных блоках ссылки на нее записываются как FN_j, где j — номер функции. Если ссылка символическая, она имеет вид FN$ имя.

1.22.2. Определение непрерывных функций

Дискретные функции могут иметь только фиксированное число значений. Непрерывные в отличие от них могут иметь неограниченное число различных значений. Часто используют подход, при котором непрерывную функцию можно дискретизировать, задав ее набором дискретных величин, однако в GPSS существует также возможность задания непрерывных функций. Графически непрерывная функция в GPSS состоит из прямых отрезков, представляя собой ломаную линию.

Пример:
```
1 FUNCTION RN2,C2
0,7/1,14
```

Первый операнд определяет номер генератора случайных чисел, второй операнд содержит символ C, идентифицирующий тип функции и число точек, задающих функцию.

График данной функции представляет собой прямую, соединяющую две точки, определяемые парами чисел 0,7 и 1,14. Следует отметить, что при использовании функцией в качестве операндов различных блоков (например, GENERATE) значениями операндов будут целые части функции.

1.22.3. Атрибутивно-значимые функции

При задании атрибутивно-значимой функции в качестве первого операнда может использоваться любой стандартный числовой атрибут, операнд B определяется символом F. Как и в случае дискретных и непрерывных функций, первым элементом каждой упорядоченной пары точек, задающих функцию, является константа. Вторым элементом каждой пары является стандартный числовой атрибут. Это означает, что каждое значение функции задается косвенно.

Например:
```
FAX FUNCTION P5,E3
9,X1/11,X2/12.X3
```

Значениями функции являются различные сохраняемые величины. Для значений параметра, равных 9, 11,12, значение функции равно значениям сохраняемых величин с номерами 1, 2 и 3 соответственно. Для аргументов, выходящих за границы, интерпретатор GPSS обрабатывает атрибутивно-значимые функции так же, как дискретные.

Так, для функции FAX при значении P5, меньшем 9, значением функции будет X1; если величина P5 превышает 12, значением функции будет X3.

В качестве значений одной функции могут использоваться различные стандартные атрибуты, а также другие функции.

Например:
```
1 FUNCTION P1, E3
4,FN2/8,Q$QUICK/10,X$NNN
```

Аргументом атрибутивно-значимой функции может являться и датчик случайных чисел.

Например:
```
2 FUNCTION RN5,E2
.5,FN$1/1,Q4
```

Предположим, ЦП обслуживает заявки, поступающие от различных сателлитных процессов в соответствии с различными функциональными зависимостями. Если вспомогательные процессоры заданы параметрически, то для задания времени обслуживания заявки в ЦП целесообразно использовать атрибутивно-значимую функцию с аргументом P1, используемым для параметрического задания вспомогательных процессоров. Пример такого использования атрибутивно-зна-

чимой функции рассмотрен при моделировании мультиплексорной ВС в примере 2.2.

1.23. Переменные. Блок VARIABLE

В GPSS предусмотрена возможность создания арифметических выражений. Арифметическая переменная является стандартным числовым атрибутом. Ссылка на арифметическую переменную имеет вид: V_j или V$ имя, где j — номер переменной. Величиной арифметической переменной является величина заданного арифметического выражения, определяющего эту переменную. В арифметических выражениях допускаются следующие арифметические операции:

\+ — сложение;
\- — вычитание;
* — умножение;
/ — деление;
@ — деление по модулю.

При делении результатом операции является только целая часть частного. Дробная часть, если она есть, отбрасывается. Данные в выражениях представляются целыми константами либо стандартными числовыми атрибутами, а также ссылками на другие арифметические переменные. Все стандартные числовые атрибуты являются целочисленными. Таким образом, арифметические выражения являются только целыми числами.

Переменная задается блоком VARIABLE, который представляется в следующем формате: имя VARIABLE арифметическое выражение, определяющее переменную.

Выражение не может располагаться далее 71 позиции. В случае, если в качестве данных в арифметическом выражении использованы параметры, рассматриваются значения параметров того транзакта, который вошел в блок, содержащий ссылку на арифметическую переменную, определяемую данным арифметическим выражением.

В выражениях, определяющих арифметические переменны, могут использоваться круглые скобки с целью задания последовательности арифметических операций.

Пример:
K11 VARIABLE V$K10+6

Данный оператор описывает переменную K11, которая определяется выражением, использующим значение арифметической переменной K10.

1.24. Блок FVARIABLE

В GPSS существует возможность определения действительных переменных, переменных с плавающей запятой. Для задания действительной переменной используется блок FVARIABLE. Форма записи данного блока совпадает с блоком VARIABLE. Вычисление значения действительной переменной имеет ряд особенностей:

1. Если в арифметическом выражении используется функция, то берется ее полное значение. Дробная часть ее не отбрасывается.
2. При использовании операции деления дробная часть частного не отбрасывается. Константы, используемые при определении действительных переменных, являются целыми.

Ссылка на действительные переменные осуществляется аналогично арифметическим переменным, т.е. с помощью V_j или V$ имя, где j — номер переменной, поэтому действительные и арифметические переменные в одной модели должны иметь различные имена.

1.25. Блок BVARIABLE

Блок BVARIABLE служит для здания булевских переменных. значением булевской переменной может быть либо 1, либо 0, в зависимости от того, будет ли булевское выражение, определяющее переменную, иметь значение «истина» или «ложь». Булевское выражение состоит из ссылок на логические условия элементов или ссылок на числовые свойства элементов, выраженных в логических величинах.

В качестве ссылок на логическое состояние элементов используются следующие логические операторы:

LS — логический переключатель установлен;
LR — логический переключатель сброшен;
F — прибор используется;
FNU — прибор не используется.

Условие, которое может существовать между двумя отдельными величинами, выраженными численно, определяется оператором отношения типа:

'G' — больше;
'GE' — больше или равно и т.д.

Форма записи оператора BVARIABLE сходна с оператором VAARIAABLE.

Пример: 1 BVARIABLE LS1 * F $ PROC

Булевская переменная BV1 имеет значение истинно, если переключатель 1 установлен, а устройство PROC занято. Если не выполняется одно из этих условий, переменная имеет значение ложно.

2. ПРИМЕРЫ МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ СРЕДСТВАМИ ЯЗЫКА GPSS

2.1. Моделирование системы, включающей два ЦП и ОП

ЦП выполняют каждый «свою» программу, обращаясь к ОП за командами и за данными. При возникновении конфликтных ситуаций, как правило, запросам от ЦП1 назначается более высокий приоритет, чем запросам от ЦП2. Примем, что запросы от ЦП1 поступают в ОП через время T_1, а запросы от ЦП" поступают в ОП через время T_2, ОП обрабатывает запросы в течение времени T_3. Промоделировать поведение системы в течение T_4, проанализировать изменение содержимого очереди к ОП каждый такт.

Распечатка программы моделирования данной системы приведена на рис. 3. Поскольку содержимое очереди анализируется каждый такт, то моделирование системы проводится в течение 10 тактов. Такой прием используется при отладке моделей. Также при отладке моделей целесообразно выбирать небольшие значения T_1, T_2, T_3, чтобы за время моделирования можно было проанализировать разрешение конфликтных ситуаций на входе ОП. Так, в приведенной распечатке T_1 = 1 такту, T_2 = 2 тактам, T_3 = 3 тактам. ОП описывается прибором MEM. Очередь заявок от ЦП1 к ОП фиксируется с помощью регистратора очереди AAA, от ЦП2 с помощью регистратора BBB, а от ЦП1 и ЦП2 совместно с помощью регистратора очереди SSS.

Модель включает в себя четыре сегмента. Первые два сегмента имитируют поведение ЦП1-ОП, ЦП2-ОП. Третий сегмент каждый такт распечатывает статистику по всем очередям и по блокам модели, четвертый сегмент управляет завершением моделирования.

В табл. 1 для первых пяти тактов модельного времени приведено число входов и содержимое очередей SSS, AAA, BBB, полученное в результате моделирования.

Таблица 1

Очередь	Номер такта				
	1 такт	2 такт	3 такт	4 такт	5 такт
Очередь SSS	1/0	3/2	4/3	6/4	7/5
Очередь AAA	1/0	2/1	3/2	4/2	5/3
Очередь BBB	0/0	1/1	1/1	2/2	2/2


```
            SIMULATE
*
*           MODEL SEGMENT 1
*
            GENERATE    1,,,,5
            QUEUE       SSS
            QUEUE       AAA
            SEIZE       MEM
            DEPART      AAA
            DEPART      SSS
            ADVANCE     3
            RELEASE     MEM
            TERMINATE
*
*           MODEL SEGMENT 2
*
            GENERATE    2,,,,3
            QUEUE       SSS
            QUEUE       BBB
            SEIZE       MEM
            DEPART      BBB
            DEPART      SSS
            ADVANCE     3
            RELEASE     MEM
            TERMINATE
*
*           MODEL SEGMENT 3
*
            GENERATE    1,,,1
  KKK       PRINT       ,,Q
            PRINT       ,,B
            ADVANCE     1
            TRANSFER    ,KKK
*
*           MODEL SEGMENT 4
*
            GENERATE    10
            TERMINATE   1
            START       1
            END
```

Рис. 3

В каждой клетке таблицы над чертой приводится число входов в очередь, а под чертой — текущее содержимое очереди.

Как следует из приведенной распечатки программы моделирования, в первом такте будет сгенерирован 1 транзакт, который имитирует заявку от ЦП1, с приоритетом 5 (см. блок GENERATE 1-го сегмента). Этот транзакт проходит через регистраторы очереди SSS и AAA и занимает прибор MEM. Поэтому в первом такте для очередей SSS и AAA число входов, равное 1, в очередь зафиксировано, а текущее содержимое очереди равно 0. Поскольку транзакты, сгенерированные в 1-м и во 2-м сегментах модели и имитирующие заявки от канала и от ЦП, образуют очередь к прибору MEM. Текущее содержимое очередей AAA и BBB становится равным по 1, а текущее содержимое суммарной очереди SSS равно 2. Число входов в очередь AAA равно 2, в очередь BBB — 1, а в очередь SSS — 3.

В третьем такте будет сгенерирован один транзакт в первом сегменте, имитирующий заявку от ЦП1, который будет пытаться занять прибор MEM. В связи с тем, что прибор занят, транзакт поступит в очередь и изменить текущее содержимое и число входов в очереди AAA и SSS. Таким образом, число входов в очередь AAA станет равным 3, в очередь SSS — 4, а текущее содержимое очередей AAA и SSS будет равно 2 и 3 соответственно. В четвертом такте прибор MEM завершит обслуживание транзакта. Блоки GENERATE 1-го и 2-го сегмента сгенерируют по одному транзакту, имитирующему заявки от канала и от ЦП к памяти. Поскольку приоритет заявок ЦП1, равный 5, выше приоритета заявок ЦП (приоритет равен 3), то транзакт, имитирующий заявку от канала из очереди AAA займет прибор MEM. В связи с этим текущее содержимое очереди AAA не изменится и будет равно 2, содержимое очереди BBB увеличится на 1 и примет значение, так же равное 2, а содержимое очереди SSS будет равно 4. Число входов в очередь AAA, BBB и SSS будет равно соответственно 4, 2 и 6. Читателям предлагается проанализировать состояние очередей в течение 10 тактов модельного времени.

2.2. Моделирование мультипроцессорной вычислительной системы

Моделируемая система включает N ЦП, каждый из которых, обрабатывая свою программу, обращается к общей ОП за данными. При обращении к ОП имеется возможность считывать блок данных переменной длины, что используется при работе с массивами. Для определенности положим, что N=3. Запросы от ЦП1, ЦП2, ЦП3 поступают к общей ОП через интервалы времени T_1, T_3, T_3 соответственно. Время выборки информации из памяти зависит от величины считываемого

или записываемого блока данных и определяется номером ЦП и равно T_4, T_5 и T_6 соответственно. Промоделировать функционирование системы в течение времени T_7. Собрать статистику по каждому из ЦП и по ОП. Определить максимальные число заявок, находившихся в очереди к ОП, среднее время пребывания одной заявки в очереди, процент заявок, для которых время ожидания в очереди было равно 0.

Распечатка программы моделирования мультипроцессорной ВС приведена на рис. 4. Программа состоит из двух сегментов. Первый сегмент предназначен для моделирования самой системы, второй сегмент управляет завершением моделирования. В первом сегменте в первый момент модельного времени будет сгенерирован 1 транзакт, который поступает на блок SPLIT работающий в режиме упорядочения (см. разд. 1.14). В первом параметре транзакта «родителя» и двух «потомков» будут занесены соответственно числа 1, 2, 3. Далее транзакты попадают на блок SEIZE, в качестве операнда которого используется P1. Таким образом, транзакт «родитель» займет прибор 1, а транзакты «потомки» — приборы 2 и 3, что соответствует значению первого параметра транзактов. Затем транзакты поступают на блок ADVANCE, операндом A которого является значение атрибутивно-значимой функции KKK (см, раздел 1.22.3). Блоком ADVANCE для каждого ЦП имитируется обработка программы до очередного обращения в ОП. В качестве аргумента функции KKK используется значение P1, т.е. в зависимости от номера ЦП интервалы времени обращения к ОП будут определяться дискретными функциями (см. разд. 1.221). При этом для первого ЦП T_1 определяется по функции с именем VER1, для второго ЦП T_2 определяется по функции с именем VER2 и для третьего ЦП T_3 определяется по функции с именем VER3. Далее транзакты проходят через блок RELEASE с операндом A, равным P1, моделируя тем самым освобождение соответствующего прибора. Заявки из каждого ЦП поступают в общую ОП, которая имитируется прибором PAM, а для сбора статистики по общей очереди к ОП используется регистратор очереди с символическим именем PAM1. Время обслуживания транзакта в приборе PAM, т.е. имитация обслуживания требования в ОП, производится с помощью блока ADVANCE, в качестве операнда A которого используется атрибутивно-значимая функция с именем MMM. В зависимости от номера ЦП, пославшего заявку в ОП, длительность обслуживания заявки в памяти для первого ЦП T_4 определяется дискретной функцией VER11, для второго ЦП T_5 определяется дискретной функцией VER12, а для третьего ЦП — функцией VER13. По завершении обслуживания в ОП заявки поступают на «свои» ЦП. Это достигается с помощью блока TRANSFER, рабо-

```
        SIMULATE
 KKK    FUNCTION    P1,E3
1,FN*VER1/2,FN*VER2/3,FN*VER3
 MMM    FUNCTION    P1,E3
1,FN*VER11/2,FN*VER12/3,FN*VER13
 VER1   FUNCTION    RN1,D3
.3,5/.7,10/1,15
 VER11  FUNCTION    RN1,D4
.1,8/.4,6/.9,10/1,21
 VER2   FUNCTION    RN2,D3
.2,6/.7,15/1,10
 VER12  FUNCTION    RN2,D4
.15,7/.35,20/.7,10/1,15
 VER3   FUNCTION    RN3,D2
.5,10/1,15
 VER13  FUNCTION    RN3,D3
.4,5/.8,10/1,15
 PAM    EQU         4,F
*
*       MODEL SEGMENT 1
*
        GENERATE    ,,,1
        SPLIT       2,MET,1
 MET    SEIZE       P1
        ADVANCE     FN*KKK
        RELEASE     P1
        QUEUE       PAM1
        SEIZE       PAM
        DEPART      PAM1
        ADVANCE     FN*MMM
        RELEASE     PAM
        TRANSFER    ,MET
*
*       MODEL SEGMENT 2
*
        GENERATE    2000
        TERMINATE   1
        START       1
        END
```

Рис. 4

тающего в режиме безусловной передачи (см. разд. 1.11). Моделирование завершится в 2000 такте, что следует из сегмента 2 (см. разд. 1.3).

Статистика по каждому ЦП и по ОП собирается автоматически в конце моделирования и соответствует стандартной статистике по приборам (см. разд. 1.5). Для определения максимального числа заявок, находившихся в очереди, среднего времени пребывания одной заявки в очереди, процента заявок, для которых время ожидания в очереди было равно 0. в модели использовался регистратор очереди PAM1. Использование регистратора очереди было вызвано необходимостью сбора и выдачи стандартной статистики по очереди заявок и ОП в конце моделирования, которая позволит ответить на поставленные вопросы (см. разд. 1.7).

2.3. Моделирование вычислительной системы с КЭШ-памятью

рассмотрим функционирование ВС, включающей ЦП, ОП и КЭШ команд. Иерархическая память имеет секторный способ организации отображения информации ОП на КЭШ. Напомним, что при секторном способе отображения j-й блок k-го сегмента ОП может быть отображен только на j-е место m-го сектора буфера, причем блоки одного сегмента могут располагаться в одном секторе буфера. Регистр номеров присутствующих сегментов служит для отображения информации о том, какой сегмент ОП расположен в том или ином секторе буфера. В качестве дисциплины замещения используется сквозная память. Это означает, что при возникновении изменения информации в КЭШ, оно одновременно заносится в соответствующий блок ОП. Поскольку в модели рассматривается КЭШ команд, обращение к буферу осуществляется только по чтению.

Запрос из ЦП в память осуществляется по исполнительному адресу, который будет представлять состоящим из номера сегмента и номера блока. Для задания исполнительного адреса будем использовать параметры транзакта.

Параметр P1 будет служить для задания номера блока, P2 определяет номер сегмента. Значение параметров P1 и P2 задаются по функциям FN1 и FN2 соответственно.

В соответствии со стратегией функционирования КЭШ-памяти, в случае отсутствия искомого сегмента в буфере, необходимо осуществить вытеснение из КЭШ какого-либо сегмента.

Будем задавать номер сектора, содержащего вытесняемый сегмент, с помощью параметра P4, определяемого по функции FN4. Параметры P3 и P5 служат для организации циклов.

В данном примере предлагается моделировать функционирование КЭШ-памяти посредством матриц. В модели используется 2 матрицы:

CASH и RNPS. Причем будет рассматривать матрицу CASH, состоящую из признаков присутствия блоков. Пусть КЭШ содержит 3 сегмента по 4 блока в каждом, тогда матрица CASH будет иметь размерность 4x3, матрица RNPS, которая соответствует регистру номеров присутствующих сегментов, будет состоять из одной строки и 3-х столбцов.

Рассмотрим логику работы модели. В начале моделируется 1 транзакт, интерпретирующийся как заявка ЦП, далее в блоках ASSIGN ей присваивается соответствующий исполнительный адрес. Далее заявка поступает в иерархическую память, причем первоначальное обращение происходит в КЭШ-память. Осуществляется просмотр элементов матрицы RNPS на совпадение их с номером сегмента в исполнительном адресе заявки, то есть с P2. Просмотр осуществляется в цикле. Результат просмотра может быть двояким. Если сегмент обнаружен в КЭШ, проверяется наличие искомого блока. Для этого элемент матрицы CASH, находящийся в строке под номером P1 (номер блока) и в столбце с номером P3 (номер сектора, в котором обнаружен искомый сегмент) сравнивается с 1. Если блок присутствует в КЭШ, т.е. MX$RNPS (P1, P3)=1, заявка отправляется на обработку в процессор. В случае отсутствия искомого блока происходит обращение к ОП (SEIZE OP) и блок переписывается в КЭШ, после чего заявка так же поступает на обслуживание в ЦП.

При отсутствии сегмента в КЭШ осуществляется обнуление битов присутствия в секторе, подлежащем вытеснению, после чего в соответствующей ячейке матрицы RNPS записывается номер заносимого сегмента, а далее при обращении к ОП переписывается соответствующий блок. По завершении обслуживания в ЦП транзакт с помощью блока TRANSFER снова поступает на обработку в КЭШ-память.

Заметим, что в данном примере моделирование рабочей нагрузки осуществлено вероятностным образом. С вероятностью P_1=0.8 новое обращение происходит к тому же информационному блоку, с вероятностью P_2=0.1=0.2x0.5 произойдет обращение к другому блоку сегмента и с вероятностью P_3 = 0.1 следующее обращение будет проходить к другому сегменту.

Время обслуживания заявки в процессоре распределено по функции F3, для ОП используется функция F4. Данная модель позволяет определить эффективность функционирования ЦП и ОП.

В качестве задания предлагается определить количество обращений в КЭШ за время моделирования, количество промахов, т.е. обращений в КЭШ при отсутствующем сегменте, а также при отсутствующем блоке.


```
        SIMULATE
 2      FUNCTION    RN1,C2
0,1/1,65
 1      FUNCTION    RN2,C2
0,1/1,5
 4      FUNCTION    RN3,C2
0,1/1,4
 RNPS   MATRIX      H,1,3
 CASH   MATRIX      H,4,3
*
*       MODEL SEGMENT 1
*
        GENERATE    ,,,1
 LAB3   ASSIGN      2,FN2
 LAB4   ASSIGN      1,FN1
 LAB2   ASSIGN      3,3
 LAB1   TEST NE     MH*RNPS(1,P3),P2,LAB11
        LOOP        3,LAB1
        ASSIGN      4,FN4
        MSAVEVALUE  RNPS,1,P4,P2,H
        ASSIGN      5,4
 LAB5   MSAVEVALUE  CASH,P5,P4,0,H
        LOOP        5,LAB5
        ASSIGN      3,P4
 LAB6   SEIZE       PAM
        ADVANCE     10,3
        RELEASE     PAM
        MSAVEVALUE  CASH,P1,P3,1,H
 LAB7   SEIZE       PROC
        ADVANCE     5,2
        RELEASE     PROC
        TRANSFER    ,2,,LAB2
        TRANSFER    .5,LAB4,LAB3
 LAB11  TEST NE     MH*CASH(P1,P3),1,LAB7
        TRANSFER    ,LAB6
*
*       MODEL SEGMENT TIMER
*
        GENERATE    2000
        TERMINATE   1
        START       1
        END
```

Рис. 5

Время моделирования предлагается ограничить 2000 единицами модельного времени. Программа моделирования КЭШ памяти представлена на рис. 5.

ЛИТЕРАТУРА

Шрайбер Т.Дж. Моделирование на GPSS. — М.: Машиностроение, 1980.

Наиболее часто используемые значения операнда С блока PRINT

C — модельное время
W — счетчик текущего содержимого блоков
N — счетчик входов в блоки
B — счетчики текущего содержимого и входов в блоки
XH — полусловные сохраняемые величины
X — полнословные сохраняемые величины
MH — полусловные матрицы
MX — полнословные матрицы
F — статистика по приборам
Q — статистика по очередям
U — статистика по цепям пользователя
LG — логические переключатели

Список наиболее часто используемых стандартных числовых атрибутов

C1* — значение относительного модельного времени
X — значение полнословной сохраняемой величины
XH — значение полусловной сохраняемой величины
MX(i,j) — элемент полнословной матрицы, стоящий в i-й строке и j-м столбце
MH(I,J) — элемент полусловной матрицы, стоящий в i-й строке и j-м столбце
N — счетчик входов
W — счетчик текущего содержимого
RN — генератор случайных чисел
Q — текущее содержимое очереди
QA — среднее содержимое очереди, округленное до целого
QC — счетчик числа входов в очередь
QM — максимальное содержимое очереди
QT — среднее время пребывания в очереди, округленное до целого
QZ — счетчик нулевых входов
F — состояние прибора (1 — занят, 0 — свободен)
FC — счетчик числа занятий прибора
FT — среднее время обработки одной заявки прибором, округленное до целого
CH — текущее содержимое цепи пользователя, округленное до целого
PR* — приоритет транзакта
FN — значение функции
P — величина параметра

ОГЛАВЛЕНИЕ