

5. Построение моделей систем с многоканальными устройствами и переключателями

Два или более обслуживающих устройств могут быть промоделированы на GPSS двумя или более ОКУ, располагаемыми рядом, т. е. параллельно. Так нужно поступать, когда отдельные устройства являются разнородными, то есть характеризуются различными свойствами, например, различной интенсивностью обслуживания.

Однако часто различные параллельно работающие устройства являются однородными. GPSS представляет для моделирования однородных параллельных устройств специальное средство, именуемое многоканальным устройством. МКУ может быть использовано несколькими транзактами одновременно. Ограничений на число МКУ в модели нет. Для различия им дают имена.

МКУ определяется до его использования командой STORAGE. Формат команды:

```
Name      STORAGE      A
```

Name — имя МКУ. Символическому имени может быть поставлен в соответствие номер командой EQU. Это необходимо, если требуется обращаться к нескольким МКУ в блоках SELECT и COUNT. Операнд A может быть только целым положительным числом. Иные способы задания емкости вызывают ошибку.

В модели можно организовать функционирование МКУ в двух режимах:

- занятие и освобождение МКУ;
- недоступность МКУ.

5.1. Занятие многоканального устройства и его освобождение

Занятие и освобождение МКУ имитируется блоками ENTER (войти) и LEAVE (выйти). Форматы блоков:

```
ENTER      A, [B]  
LEAVE     A, [B]
```

Операнд A в обоих блоках используется для указания имени, соответствующего МКУ. Операнд B задает число устройств (элементов памяти), которое

должно быть занято в блоке ENTER или освобождено в блоке LEAVE. По умолчанию операнд $B = 1$. При $B = 0$ блок считается неработоспособным.

Когда транзакт входит в блок ENTER, операнд A используется для нахождения МКУ с указанным именем. Если такого МКУ нет, происходит останов по ошибке «Обращение к несуществующей памяти». Если МКУ существует и задан операнд B, он вычисляется, округляется до целого и полученный результат используется для оценки свободной емкости. Транзакт может войти в блок ENTER, если МКУ находится в доступном состоянии и достаточно емкости для выполнения запроса. В противном случае транзакт помещается в список задержки устройства в соответствии с приоритетом.

Когда транзакт входит в блок ENTER, планировщик выполняет следующие действия:

- увеличивает на 1 счетчик входов МКУ;
- увеличивает на значение операнда B (по умолчанию на 1) текущее содержимое МКУ;
- уменьшает на значение операнда B (по умолчанию на 1) доступную емкость МКУ.

Если транзакт при входе в блок ENTER запрашивает больше устройств (элементов памяти), чем определено командой STORAGE, т. е. ее операнд A меньше операнда B блока ENTER, возникает ошибка «Запрос элементов памяти превышает ее общую емкость».

МКУ никогда не может быть удалено из текущей модели, даже если команда STORAGE удаляется из рабочей программы. МКУ можно переопределить, т. е. изменить емкость другой командой STORAGE с тем же самым именем.

Например:

Batr	STORAGE	18
------	---------	----

Повторное описание

Batr	STORAGE	24
------	---------	----

Имитация обслуживания в течение какого-то промежутка времени также осуществляется блоком ADVANCE.

Пример 1.

Nak	STORAGE	20
	. . .	
	ENTER	Nak, 2
	ADVANCE	120, 40
	LEAVE	Nak, 2
	. . .	

Командой STORAGE определяется МКУ с именем Nak емкостью 20 единиц. При входе транзакта в блок ENTER занимается 2 единицы и столько же освобождается в блоке LEAVE при выходе из МКУ.

Пример 2.

Nak	STORAGE	20
	. . .	
	ENTER	Nak, 21
	ADVANCE	120, 40
	LEAVE	Nak, 2
	. . .	

При входе транзакта в блок ENTER произойдет останов по ошибке, так как транзакт будет пытаться занять больше каналов (21), чем определено (20) командой STORAGE. То же самое произойдет, если при выходе из блока LEAVE транзакт будет пытаться освободить каналов больше, чем определено командой STORAGE.

Пример 3.

Pun1	EQU	1
Pun2	EQU	2
Pun3	EQU	3
Pun1	STORAGE	6
Pun2	STORAGE	5
Pun3	STORAGE	3
	. . .	
	ENTER	*1
	ADVANCE	MX\$NorVr(P2, P3)
	LEAVE	*1
	. . .	

В данном примере определены три МКУ с именами Pun1, Pun2, Pun3 и емкостями 6, 5 и 3 соответственно. Именам командами EQU поставлены в соответствие номера 1, 2 и 3. Предполагается, что при входе транзакта в блок ENTER в его первом параметре содержится какой-либо один из трех номеров. Согласно этому номеру и занимается МКУ, а затем освобождается. Операнд В в блоках ENTER и LEAVE не используется, поэтому транзактом занимается и освобождается одна единица емкости МКУ.

5.2. Перевод многоканального устройства в недоступное состояние и восстановление доступности

Недоступность МКУ моделируется блоком SUNAVAIL (символ S означает МКУ, UNAVAIL — недоступный). Формат блока:

SUNAVAIL A

Операнд A — имя или номер МКУ, может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Например:

SUNAVAIL Batr

Когда транзакт входит в этот блок, МКУ *Watr* становится недоступным. Если при переводе в недоступное состояние в МКУ находились транзакты, т. е. текущее содержимое МКУ не равнялось нулю, то обслуживание этих транзактов продолжается, пока текущее содержимое не станет равным нулю. Транзакты, которые пытаются занять МКУ во время нахождения его в недоступном состоянии, не входят в блок *ENTER* и помещаются в список задержки МКУ.

Нахождение в недоступном состоянии продолжается до тех пор, пока транзакт не войдет в блок *SAVAIL*. Формат блока:

SAVAIL A

Операнд *A* — имя или номер МКУ. Может быть таким же, как операнд *A* в блоке *SUNAVAIL*.

Если в момент перевода МКУ в доступное состояние в его списке задержки были транзакты, им предоставляется возможность занять МКУ в соответствии с дисциплиной «first-fit-with-skip» (первый подходящий с пропусками). Транзакты, которым будет отказано в занятии МКУ, остаются в списке задержки.

; Пример 5.1

; Определение МКУ

Кап STORAGE 3

; Сегмент имитации поступления и обслуживания транзактов

GENERATE , , , 3 ; Источник транзактов

ENTER Кап ; Занять МКУ

ADVANCE 10 ; Обслуживание

LEAVE Кап ; Освободить каналы МКУ

TERMINATE ; Обслуженные транзакты

; Сегмент имитации недоступности

GENERATE 5, , , 1 ; Транзакт-инициатор недоступности

SUNAVAIL Кап ; Перевод в состояние недоступности

ADVANCE 6 ; Восстановление доступности

SAVAIL Кап ; Перевод в состояние доступности

TERMINATE 1

Блок *GENERATE* первого сегмента генерирует в $t = 0$ три транзакта, которые занимают МКУ *Кап* емкостью 3, определенной командой *STORAGE*.

В $t = 5$ блок *GENERATE* второго сегмента генерирует транзакт, который входит в блок *SUNAVAIL* и переводит МКУ *Кап* в недоступное состояние.

Приведем результаты моделирования.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	11.000	10	0	1

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	3	0	0
	2	ENTER	3	0	0
	3	ADVANCE	3	0	0
	4	LEAVE	3	0	0
	5	TERMINATE	3	0	0
	6	GENERATE	1	0	0
	7	SUNAVAIL	1	0	0
	8	ADVANCE	1	0	0
	9	SAVAIL	1	0	0
	10	TERMINATE	1	0	0

МКУ Кан в недоступном состоянии находилось 6 единиц модельного времени и в $t = 11$ блоком SAVAIL была восстановлена его доступность. Время обслуживания транзактов равно 10 единицам. По результатам моделирования видно, что и в период недоступности обслуживание находившихся в МКУ трех транзактов продолжалось еще 5 единиц модельного времени и завершилось в $t = 10$. Все три обслуженных транзакта вошли в блок TERMINATE первого сегмента.

Пример 5.2. В примере 5.1 обслуживание трех транзактов закончилось до того, как устройство вновь стало доступным. Рассмотрим случай, когда обслуживание некоторых транзактов не закончилось до восстановления доступности.

; Пример 5.2

; *Определение МКУ*

```

Кан      STORAGE      3
;
;          Сегмент 1
GENERATE  3, , , 3      ; Источник транзактов
ENTER     Кан          ; Занять МКУ
ADVANCE   4            ; Обслуживание
LEAVE     Кан          ; Освободить каналы МКУ
TERMINATE ;            ; Обслуженные транзакты
;
;          Сегмент 2
GENERATE  11, , , 1    ; Транзакт-инициатор недоступности
SUNAVAIL  Кан          ; Перевод в состояние недоступности
ADVANCE   1            ; Восстановление доступности
SAVAIL    Кан          ; Перевод в состояние доступности
ADVANCE   1            ; Задержать
TERMINATE 1

```

Блок GENERATE первого сегмента генерирует три транзакта с интервалами в 3 единицы модельного времени. Первый транзакт поступит в модель в $t = 3$, займет один канал МКУ и обслужится в $t = 7$.

Второй транзакт займет второй канал МКУ в $t = 6$. Его обслуживание закончится в $t = 10$.

Третий транзакт в $t = 9$ займет один из двух свободных каналов МКУ. В $t = 11$ блок GENERATE второго сегмента сгенерирует транзакт, который войдет в блок SUNAVAIL и МКУ станет недоступным. Обработка третьего транзакта продолжается. В $t = 12$ МКУ вновь будет доступным. Обработка третьего транзакта продолжится и завершится в $t = 13$.

Приведем результаты моделирования.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	13.000	11	0	1

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	3	0	0
	2	ENTER	3	0	0
	3	ADVANCE	3	0	0
	4	LEAVE	3	0	0
	5	TERMINATE	3	0	0
	6	GENERATE	1	0	0
	7	SUNAVAIL	1	0	0
	8	ADVANCE	1	0	0
	9	SAVAIL	1	0	0
	10	ADVANCE	1	0	0
	11	TERMINATE	1	0	0

Они точно такие же, как и в примере 5.1. То есть недоступность только за-прещает поступление транзактов в МКУ. Обработка же их продолжается, как будто и не было никакой недоступности.

Таким образом, осуществить имитацию выхода МКУ из строя, при котором все транзакты, находившиеся в МКУ на обслуживании теряются, блоками SUNAVAIL и SAVAIL невозможно.

5.3. Проверка состояния многоканального устройства

5.3.1. Проверка состояния блоком GATE

Состояние МКУ, как и состояние ОКУ, проверяется блоком GATE такого же формата:

GATE X A, [B]

Отличие состоит в значениях условного оператора X, которые могут быть следующими:

- SE — МКУ, заданное операндом A, пусто;
- SF — МКУ, заданное операндом A, заполнено;
- SNE — МКУ, заданное операндом A, не пусто;
- SNF — МКУ, заданное операндом A, не заполнено;
- SNV — МКУ, заданное операндом A, не доступно;

○ SV — МКУ, заданное операндом A, доступно.

Блок GATE также работает в двух режимах:

- отказа во входе;
- разрешении во входе и альтернативном выходе.

Например:

```
GATE SNF      Can, Met5
```

Если МКУ с именем Can не заполнено, т. е. имеются свободные каналы (элементы памяти), заданное в блоке GATE условие выполняется и транзакт будет направлен к следующему блоку. Если МКУ будет заполнено, транзакт будет направлен к блоку с меткой Met5.

Пример 5.3. МКУ переводится в недоступное состояние. Обслуживание транзактов, находившихся в МКУ к моменту перевода его в недоступное состояние, продолжается, но вновь поступающие транзакты не помещаются в список задержки МКУ. Восстановление доступности МКУ начинается после того, как МКУ будет пусто.

; Пример 5.3

; Определение МКУ

```
Sist STORAGE 7
```

; Сегмент имитации поступления и обслуживания транзактов

```
GENERATE 2,,5 ; Источник транзактов
```

```
GATE SV Sist, Met1 ; МКУ доступно?
```

```
GATE SNF Sist, Met1 ; Да, тогда МКУ не заполнено? Нет, тогда
```

```
ENTER Sist, 3 ; Занять, если есть свободные каналы МКУ
```

```
ADVANCE 5 ; Обслуживание
```

```
LEAVE Sist, 3 ; Освободить каналы МКУ
```

```
TERMINATE 1 ; Обслуженные транзакты
```

```
Met1 TERMINATE 1 ; Потерянные транзакты
```

; Сегмент имитации недоступности МКУ

```
GENERATE ,,1 ; Транзакт-инициатор недоступности
```

```
Met2 ADVANCE 7 ; Время нахождения в доступном состоянии
```

```
SUNAVAIL Sist ; Перевод в состояние недоступности
```

```
GATE SE Sist ; МКУ пусто?
```

```
ADVANCE 1 ; Да, тогда восстановление доступности
```

```
SAVAIL Sist ; Перевод в состояние доступности
```

```
TRANSFER ,Met2 ; Отправить на очередной цикл
```

В $t = 0$ блок GENERATE сегмента 2 генерирует один транзакт ($XN_1=2$), который входит в блок задержки ADVANCE.

В $t = 2$ первый транзакт ($XN_1=1$) блока GENERATE сегмента 1 проходит первый и второй блоки GATE и занимает три канала из семи МКУ Sist. Обслуживание этого транзакта закончится в $t = 2 + 5 = 7$.

В $t = 4$ второй транзакт ($XN1=3$) также проходит оба блока GATE сегмента 1 и занимает следующие три канала МКУ Sist. Остается свободным один канал. Обслуживание второго транзакта закончится в $t = 4 + 5 = 9$.

В $t = 6$ третий транзакт ($XN1=4$) пройдет первый блок GATE сегмента 1 и войдет во второй блок GATE. Так как МКУ Sist имеет один свободный канал, т. е. не заполнено, условие во втором блоке GATE выполняется и транзакт пройдет к блоку ENTER. Но одного свободного канала недостаточно для удовлетворения запроса, требуется три, поэтому третий транзакт помещается в список задержки МКУ Sist.

В $t = 7$ МКУ Sist блоком SUNAVAIL переводится в недоступное состояние. В это же время закончится обслуживание первого транзакта, но третий транзакт не займет МКУ, а так и останется в списке задержки.

Так как в МКУ Sist находится один второй транзакт, т. е. оно не пусто, то транзакт, вызвавший перевод в недоступное состояние, будет задержан блоком GATE сегмента 2, потому что заданное в нем условие не выполняется.

В $t = 8$ четвертый транзакт ($XN1=5$) первым блоком GATE сегмента 1 вследствие недоступности МКУ Sist направляется к блоку TERMINATE с меткой Met1 и уничтожается.

В $t = 9$ закончится обслуживание второго транзакта. МКУ станет пусто, условие в блоке GATE сегмента 2 выполнится и сгенерировавший недоступность транзакт войдет в блок задержки ADVANCE.

В $t = 10$ пятый транзакт ($XN1=6$) первым блоком GATE сегмента 1 вследствие недоступности МКУ Sist направляется к блоку TERMINATE с меткой Met1 и уничтожается.

Также в $t = 9 + 1 = 10$ блок SAVAIL переведет МКУ в доступное состояние. Третий транзакт из списка задержки МКУ займет три канала МКУ. Обслуживание третьего транзакта ($XN1=4$) закончится в $t = 10 + 5 = 15$.

ЗАМЕЧАНИЕ

Как видно из примера, блок GATE позволяет только определить состояние незаполненности МКУ, т. е. наличие свободных каналов, но достаточно ли их для удовлетворения запроса он не определяет. Поэтому если в блоке ENTER операнд $B \neq 1$ и емкость МКУ не делится на значение операнда B без остатка, транзакт проходит через блок GATE и помещается в список задержки МКУ. При $B \neq 1$ и делении емкости МКУ на значение операнда B нацело или $B = 1$ список задержки МКУ всегда пуст.

5.3.2. Проверка состояния булевой переменной и блоком TEST

При проверке состояния МКУ блок TEST имеет тот же формат записи и те же условные операторы, что и при проверке состояния ОКУ.

В булевой переменной допускается использовать следующие логические операторы, связанные с МКУ:

- SF равен 1, если МКУ заполнено полностью, иначе — 0;
- SE равен 1, если МКУ пусто, иначе — 0;

○ SV равен 1, если МКУ доступно, иначе — 0,
а также СЧА МКУ:

- S — число занятых каналов МКУ;
- SA — среднее значение занятых каналов МКУ;
- SC — счетчик использования МКУ;
- SR — коэффициент использования МКУ;
- SM — максимальное число занятых каналов МКУ;
- ST — среднее время использования одного канала МКУ.

Как видно, SE , SF и SV можно использовать как в блоке GATE, так и в булевых переменных. Если же в блоке GATE в качестве условного оператора X указать, например, SA , то при трансляции будет ошибка «Недопустимое использование спецификатора класса СЧА». Если же в булевой переменной указать, например, SNV , также будет ошибка «Предполагается фактор выражения».

Пример 5.4. На трех параллельно работающих МКУ требуется организовать обслуживание так, чтобы их коэффициенты использования были одинаковыми.

; Пример 5.4

; *Определение МКУ и булевых переменных*

```
Use11 STORAGE 2 ; Емкость МКУ1
Use12 STORAGE 2 ; Емкость МКУ2
Use13 STORAGE 2 ; Емкость МКУ3
Prov1 BVARIABLE BV$Prov2'AND'BV$Prov3'AND'BV$Prov4
Prov2 BVARIABLE S$Use11'OR'(SR$Use11>400)
Prov3 BVARIABLE S$Use12'OR'(SR$Use12>400)
Prov4 BVARIABLE S$Use13'OR'(SR$Use13>400)
Prov5 BVARIABLE BV$Prov2'AND'BV$Prov3
```

; **Сегмент имитации МКУ1**

```
GENERATE (Exponential(310,0,2.3)) ; Источник транзактов
TEST E BV$Prov1,0,Met1 ; Все МКУ заполнены?
TEST E BV$Prov5,0,Met3 ; МКУ1 и МКУ2 заполнены?
TEST E BV$Prov2,0,Met2 ; МКУ1 заполнено?
ENTER Use11 ; Занять МКУ1
ADVANCE 12,4.3 ; Обслуживание МКУ1
LEAVE Use11 ; Освободить каналы МКУ1
TERMINATE ; Обслуженные транзакты МКУ1
```

; **Сегмент имитации МКУ2**

```
Met2 TEST E BV$Prov3,0,Met3 ; МКУ2 заполнено?
ENTER Use12 ; Занять МКУ2
ADVANCE 14.8,12.35 ; Обслуживание МКУ2
LEAVE Use12 ; Освободить каналы МКУ2
TERMINATE ; Обслуженные транзакты МКУ2
```

; **Сегмент имитации МКУ3**

```
Met3 TEST E BV$Prov4,0,Met1 ; МКУ3 заполнено?
ENTER Use13 ; Занять МКУ3
```

```

ADVANCE      (Exponential(310,0,22.3)) ; Обслуживание МКУ3
LEAVE        Use13                       ; Освободить каналы МКУ3
TERMINATE    ; Обслуженные транзакты МКУ3
Met1         TERMINATE                   ; Потерянные транзакты
; Сегмент задания времени моделирования
GENERATE     3600
TERMINATE    1
    
```

При входе транзакта в первый блок TEST вычисляется булева переменная Prov1. Она равна 1 тогда, когда все три МКУ заполнены или их коэффициенты использования больше 0,4. В этом случае заданное в блоке TEST условие не выполняется и транзакт направляется к блоку TERMINATE с меткой Met1 для уничтожения.

Булева переменная Prov5 вычисляется при входе транзакта во второй блок TEST. Она равна 1, если МКУ Use1 и Use2 не имеют свободных каналов или их коэффициенты использования больше 0,4. При Prov5 = 1 условие во втором блоке TEST также не выполняется и транзакт направляется к блоку TEST с меткой Met3 на МКУ Use3.

Если МКУ Use3 не заполнено и его коэффициент использования меньше 0,4 (значение булевой переменной Prov4 равно 0), условие в блоке TEST выполняется, транзакт проходит к следующему блоку и занимает один из свободных каналов МКУ Use3. При невыполнении условия в блоке TEST (значение булевой переменной Prov4 равно 1) транзакт направляется к блоку TERMINATE с меткой Met1 для уничтожения.

Если при входе транзакта в первый и второй блоки TEST булевы переменные Prov1 и Prov5 равны 0, транзакт войдет в третий блок TEST, который пропустит его для обслуживания на МКУ Use1 в случае, если МКУ Use1 имеет свободные каналы или его коэффициент использования меньше 0,4. В противном случае транзакт направляется к блоку TEST с меткой Met2.

Блок TEST с меткой Met2 проверяет наличие свободных каналов у МКУ Use2 и величину его коэффициента использования. Если булева переменная Prov3 равна 0, транзакт пропускается к следующему блоку ENTER и занимает один из свободных каналов МКУ Use2.

Если МКУ Use2 не имеет свободных каналов или его коэффициент использования больше 0,4, транзакт направляется к блоку TEST с меткой Met3, где осуществляются аналогичные проверки состояния МКУ Use3.

Приведем результаты моделирования.

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
CPU1	2	1	0	1	2402	1	0.800	0.400
CPU2	2	2	0	1	1965	1	0.800	0.400
CPU3	2	2	0	1	1233	1	0.800	0.400

Все три МКУ имеют равные коэффициенты использования.

ЗАМЕЧАНИЕ

Использование булевой переменной и блока TEST не только расширяет возможности, но и позволяет сократить машинное время за счет того, например, что сразу выводятся из модели транзакты, для которых выполняются условия в булевой переменной Prov1. Для данного примера таких транзактов оказалось 65%.

Пример 5.5. В условиях примера 5.3 организуем такую дисциплину обслуживания, при которой в случае незаполненности МКУ, но недостаточности свободных каналов для удовлетворения запроса, транзакт не пропускается к блоку ENTER и, следовательно, не помещается в список задержки.

; Пример 5.5

; *Определение МКУ и булевых переменных*

```
Zap      EQU      3
Sist     STORAGE  7
Kont1    BVARIABLE SV$Sist'AND'((7-S$Sist)>=Zap))
Kont2    BVARIABLE SE$Sist
```

; *Сегмент имитации поступления и обслуживания транзактов*

```
GENERATE 2,,5      ; Источник транзактов
TEST E    BV$Kont1,1,Met1 ; МКУ свободно?
ENTER    Sist,3    ; Занять МКУ
ADVANCE  5        ; Обслуживание
LEAVE    Sist,3    ; Освободить каналы МКУ
TERMINATE 1       ; Обслуженные транзакты
Met1     TERMINATE 1 ; Потерянные транзакты
```

; *Сегмент имитации недоступности*

```
GENERATE ,,,1      ; Транзакт-инициатор недоступности
Met2     ADVANCE  7 ; Время нахождения в состоянии доступности
SUNAVAIL Sist      ; Перевод в состояние доступности
TEST E    BV$Kont2,1 ; МКУ пусто?
ADVANCE  1        ; Восстановление доступности
SAVAIL    Sist     ; Перевод в состояние доступности
TRANSFER ,Met2    ; Отправить на очередной цикл
```

В $t = 0$ блок GENERATE сегмента 2 генерирует один транзакт ($XN1=2$), который входит в блок задержки ADVANCE.

В $t = 2$ первый транзакт ($XN1=1$) блока GENERATE сегмента 1 входит блок TEST. Так как МКУ свободно и доступно, булева переменная $Kont1=1$, условие в блоке TEST выполняется, транзакт его проходит и занимает три канала из семи МКУ Sist. Обслуживание первого транзакта закончится в $t = 2 + 5 + 7$.

В $t = 4$ второй транзакт ($XN1=3$) также проходит блок TEST сегмента 1 и занимает следующие три канала МКУ Sist. Остается свободным один канал. Обслуживание второго транзакта закончится в $t = 4 + 5 = 9$.

В $t = 6$ третий транзакт ($XN1=4$) не пройдет блок TEST сегмента 1. Значение булевой переменной будет равно 0, так как МКУ Sist имеет один свободный канал. Но одного канала недостаточно для удовлетворения запроса, требуется

три, поэтому третий транзакт направляется в блок TERMINATE с меткой Met1 и уничтожается.

В $t = 7$ МКУ Sist блоком SUNAVAIL переводится в недоступное состояние. В это же время закончится обслуживание первого транзакта.

Так как в МКУ Sist находится один второй транзакт, т. е. оно не пусто, то транзакт, вызвавший перевод в недоступное состояние, будет задержан блоком TEST сегмента 2, потому что заданное в нем условие не выполняется (булева переменная Kont2 равна 0).

В $t = 8$ четвертый транзакт ($XN1 = 5$) блоком TEST сегмента 1 вследствие недоступности МКУ Sist (булева переменная Kont1 равна 0) направляется к блоку TERMINATE с меткой Met1 и уничтожается.

В $t = 9$ закончится обслуживание второго транзакта. МКУ станет пусто, условие в блоке TEST сегмента 2 выполнится и сгенерировавший недоступность транзакт войдет в блок задержки ADVANCE.

В $t = 10$ пятый транзакт ($XN1 = 6$) первым блоком GATE сегмента 1 вследствие недоступности МКУ Sist направляется к блоку TERMINATE с меткой Met1 и уничтожается.

Также в $t = 9 + 1 = 10$ блок SAVAIL переведет МКУ в доступное состояние. Транзакт, генерирующий недоступность и доступность, блоком TRANSFER будет направлен к блоку ADVANCE с меткой Met2.

Таким образом, в примере 5.3 обслуживаются три транзакта, а в данном примере — только два транзакта, так как третий транзакт из-за недостаточности свободной емкости не пропускается блоком TEST к блоку ENTER и, естественно, не помещается в список задержки МКУ.

5.4. Моделирование неисправностей многоканальных устройств

Ранее отмечалось, что осуществить имитацию выхода МКУ из строя, при котором все транзакты, находившиеся в МКУ на обслуживании, теряются, блоками SUNAVAIL и SAVAIL невозможно.

Рассмотрим, как это можно осуществить с применением блока DISPLACE.

Пример 5.6. Многоканальная СМО с отказами без очереди. СМО имеет конечную надежность. При выходе СМО из строя заявки, находившиеся на обслуживании, теряются. Новые заявки на обслуживание не принимаются.

; Пример 5.6

; Определение МКУ и булевых переменных

Emk	EQU	7
Zap	EQU	2
Sist	STORAGE	7
Kont1	BVARIABLE	SV\$Sist'AND'((7-\$Sist)>=Zap)
Kont2	BVARIABLE	SE\$Sist

; Сегмент имитации поступления и обслуживания заявок

```

GENERATE      2, , , 7      ; Источник заявок
Met12 TEST E      BV$Kont1,1, Met1 ; Есть ли место в МКУ?
SAVEVALUE    KoIPovt, (INT(Emk/Zap))
ENTER        Sist, Zap      ; Занять МКУ

```

; Учет номеров транзактов, занявших МКУ

```

ASSIGN       KoIPovt, X$KoIPovt ; Запись в параметр цикла
Met5  TEST E      X*KoIPovt, 0, Met4 ; Есть ли в списке место?
SAVEVALUE    P$KoIPovt, XN1      ; Да, записать номер транзакта
TRANSFER     , Met6              ; Выйти из цикла
Met4  LOOP       KoIPovt, Met5    ; Повторить или конец цикла

```

```

;
Met6  ADVANCE    5                ; Имитация обслуживания
LEAVE      Sist, Zap              ; Освободить МКУ

```

; Учет номеров транзактов, освободивших МКУ

```

ASSIGN       KoIPovt, X$KoIPovt ; Запись в параметр цикла
Met8  TEST E      X*KoIPovt, XN1, Met7 ; Есть ли этот номер транзакта?
SAVEVALUE    P$KoIPovt, 0        ; Да, тогда удалить
TRANSFER     , Met9              ; Выйти из цикла
Met7  LOOP       KoIPovt, Met8    ; Повторить или конец цикла

```

```

;
Met9  TERMINATE  1                ; Обслуженные заявки
Met1  TERMINATE  1                ; Потерянные заявки

```

; Сегмент имитации неисправностей

```

GENERATE     , , , 1
Met2  ADVANCE   7.5                ; Интервал отказов
SUNAVAIL    Sist                  ; Перевод в недоступность

```

; Удаление транзактов из МКУ

```

ASSIGN       KoIPovt, X$KoIPovt ; Запись в параметр цикла
Met10 TEST NE    X*KoIPovt, 0, Met11 ; Есть ли транзакт в МКУ
DISPLACE    X*KoIPovt, Met3      ; Да, тогда удалить
SAVEVALUE    P$KoIPovt, 0        ; Снять с учета
Met11 LOOP   KoIPovt, Met10      ; Повторить или конец цикла

```

```

;
TEST E      BV$Kont2, 1          ; МКУ пуст?
ADVANCE    1                    ; Да, тогда ремонтируем
SAVAIL     Sist                 ; Перевод в доступность
TRANSFER   , Met2
Met3  LEAVE     Sist, 2          ; Освободить МКУ
TERMINATE  1                    ; Удаленные из МКУ транзакты

```

Блок GENERATE с интервалами две единицы модельного времени генерирует семь транзактов. Следующий за ним блок TEST проверяет МКУ Sist на исправность (доступность) и наличия в нем свободных каналов, достаточных для удовлетворения запроса. Если булева переменная Kont1 равна 1, транзакт пропускается и занимает МКУ Sist. Но перед этим вычисляется и заносится целое

число $\text{INT}(\text{Emk}/\text{Zap}) = \text{INT}(7/2) = 3$ в сохраняемую ячейку с именем KolPovt, которое определяет, сколько транзактов может одновременно находиться в МКУ.

После выхода транзакта из блока ENTER начинает работать сегмент учета номеров транзактов, занявших МКУ. Блоком ASSIGN в параметр с именем KolPovt — параметр цикла заносится число, находящееся в сохраняемой ячейке с именем KolPovt. Далее в цикле, тело которого начинается с блока TEST с меткой Met5 и заканчивается блоком LOOP с меткой Met4, находится свободное место в списке для записи номера транзакта. Свободное место определяется блоком TEST как равенство нулю значения какой-либо одной из трех сохраняемых ячеек X1, X2 или X3 (по числу транзактов, одновременно находящихся в МКУ). Такая ячейка всегда есть и блоком SAVEVALUE в нее записывается номер занявшего МКУ транзакта. А так как сохраняемых ячеек, значения которых равны нулю, может быть несколько (особенно вначале работы модели), после записи номера транзакта осуществляется выход из цикла.

Транзакт направляется к блоку ADVANCE с меткой Met6 и входит в него. Имитируется обслуживание заявки. После обслуживания транзакт освобождает МКУ Sist, пройдя блок LEAVE.

Начинает работать сегмент учета номеров транзактов, освободивших МКУ. Его работа аналогична рассмотренной ранее работе сегмента учета номеров транзактов, занявших МКУ. Отличие состоит в том, что отыскивается сохраняемая ячейка X1, X2 или X3, которая содержит номер обслуженного транзакта. Найденная ячейка обнуляется, т. е. обслуженный транзакт снимается с учета. Так как номер записывается только в одну ячейку, то после ее обнуления осуществляется выход из цикла, тело которого начинается с блока TEST с меткой Met8 и заканчивается блоком LOOP с меткой Met7.

Теперь рассмотрим работу модели при возникновении неисправности МКУ. Транзакт, инициирующий неисправность, после задержки блоком ADVANCE сегмента имитации неисправностей, входит в блок SUNAVAIL, который переводит МКУ Sist в недоступное (неисправное) состояние.

Далее начинается работа сегмента удаления транзактов из МКУ. Суть ее заключается в следующем. В цикле, также организованном с помощью блока LOOP, просматривается список номеров транзактов, занявших МКУ. Если значение какой-либо сохраняемой ячейки X1, X2 или X3 не равно нулю, значит, в ней записан номер транзакта, находящегося в данный момент в МКУ. Блоком DISPACE этот транзакт перемещается к блоку LEAVE с меткой Met3, освобождает МКУ и выводится из модели.

После удаления из МКУ всех транзактов, транзакт-инициатор неисправности входит в следующий за блоком LOOP с меткой Met11 блок TEST. Так как Sist пусто (булева переменная Kont2 равна нулю), блок TEST пропускает его и начинается имитация восстановления работоспособности МКУ.

Результаты моделирования представлены в табл. 5.1. Из результатов следует, что обслужены только две заявки (первый и шестой транзакты). Пять заявок не обслужены: пятый транзакт поступил, когда МКУ было неисправным,

а третий и четвертый, седьмой и восьмой транзакты потеряны: в моменты возникновения неисправностей ($t = 7,5$ и $t = 16$ соответственно) они находились в МКУ.

Коэффициент использования (Util.) МКУ $Sist$ равен 0,375. Он рассчитывается по формуле:

$$k = \frac{\sum_{i=1}^n \alpha_i \cdot \Delta t_i}{T \cdot E}, \quad (5.1)$$

где n — число транзактов, занимавших МКУ за время T моделирования;

E — число каналов МКУ (емкость);

α_i — число каналов, занимаемых i -м транзактом;

Δt_i — промежуток времени, в течение которого i -м транзактом было занято α_i каналов, рассчитывается как

$$\Delta t_i = t_{2i} - t_{1i}, \quad (5.2)$$

t_{1i} и t_{2i} — абсолютное модельное время занятия и освобождения соответственно α_i каналов МКУ.

Таблица 5.1

Результаты моделирования примера 5.6

События	Транзакты						
	1	3	4	5	6	7	8
Вход в модель	2	4	6	8	10	12	14
Занятие МКУ	2	4	6		10	12	14
В МКУ транзакты	1	1, 3	1, 3, 4		6	6, 7	6, 7, 8
Удаление		7.5	7.5			16	16
Освобождение МКУ	7	7.5	7.5		15	16	16
Вывод из модели	7	7.5	7.5	8	15	16	16

Для рассматриваемого примера 5.6 (см. табл. 5.1) имеем:

$$E = 7, T = 16, \alpha = 2;$$

$$\Delta t_1 = 7 - 2 = 5, \Delta t_2 = 7,5 - 4 = 3,5; \Delta t_3 = 7,5 - 6 = 1,5; \Delta t_4 = 0;$$

$$\Delta t_5 = 15 - 10 = 5; \Delta t_6 = 16 - 12 = 4; \Delta t_7 = 16 - 14 = 2;$$

$$k_1 = \frac{2(5+3,5+1,5+0+5+4+2)}{16 \cdot 7} = \frac{2 \cdot 21}{16 \cdot 7} = \frac{3}{8} = 0,375.$$

Таким образом, коэффициент использования МКУ рассчитан с учетом времени занятия его и теми транзактами, обслуживание которых было прервано.

Если в результате моделирования требуется определить коэффициент использования МКУ без учета удаленных из него транзактов, т. е. транзактов, обслуживание которых было прервано, пользователь может это сделать самостоятельно. Для этого нужно знать, сколько заявок будет обслужено полностью. Введем в модель следующие строки:

```

KIsp      FVARIABLE      (Zap#N$Met9#5)/(Emk#16)
          SAVEVALUE      KIsp,V$KIsp
    
```

Получим

$$k_2 = \frac{2 \cdot (5 + 5)}{16 \cdot 7} = \frac{5}{28} = 0,179.$$

Пример 5.7. Многоканальная СМО с ограниченной очередью обладает конечной надежностью. При выходе из строя обслуживание находящихся в каналах заявок прекращается, но они не теряются, а после восстановления работоспособности обслуживаются сначала. Новые заявки принимаются, если в очереди есть место.

Так как заявки, находящиеся на обслуживании, при выходе СМО из строя не должны теряться, то их нужно сохранять в очереди до тех пор, пока они не будут обслужены.

Модель функционирования такой СМО представлена ниже.

; Пример 5.7

; *Определение МКУ, арифметических и булевых переменных*

```

Emk      EQU          7      ; Число каналов МКУ
Zap      EQU          2      ; Число занимаемых одним транзактов каналов
Smo      STORAGE     7      ; Задание числа каналов МКУ
Dlina   EQU          3      ; Длина очереди
Kont1    BVARIABLE   (Dlina-(S$Smo/Zap))>0
Kont2    BVARIABLE   SV$Smo 'AND' (R$Smo>Zap)
Kont3    BVARIABLE   SV$Smo
Rasm     VARIABLE    Dlina+1+P$KolTp
KIsp     VARIABLE    (N$Met18#7#Zap)/(Emk#19)
    
```

; *Сегмент имитации поступления и обслуживания заявок*

```

GENERATE 2,,7          ; Источник заявок
TEST E   BV$Kont1,1,Met1 ; Есть ли место в очереди?
SAVEVALUE KolTp,(INT(Emk/Zap)) ; Расчет числа повторений
    
```

; *Учет номеров транзактов, вставших в очередь*

```

SSIGN    Dlina,Dlina   ; Запись в параметр цикла
Met4     TEST E        X*Dlina,0,Met5 ; Есть ли в списке место?
SAVEVALUE P$Dlina,XN1 ; Да, записать номер транзакта
TRANSFER ,Met6         ; Выйти из цикла
Met5     LOOP         Dlina,Met4     ; Повторить или конец цикла
;
Met6     TEST E        BV$Kont2,0,Met3 ; Есть ли в очереди место?
    
```

```

LINK      Накоп, LIFO      ; Да, поставить транзакт в очередь
Met3     ENTER      Smo, Zap      ; Занять МКУ
;      Учет номеров транзактов, занявших МКУ
ASSIGN    KolTp, X$KolTp      ; Запись в параметр цикла
Met8     ASSIGN    1, V$Rasm      ; Расчет индекса
TEST E    X*1, 0, Met7      ; Есть ли в списке место?
SAVEVALUE P1, XN1      ; Да, записать номер транзакта
TRANSFER  , Met9      ; Выйти из цикла
Met7     LOOP      KolTp, Met8      ; Повторить или конец цикла
;
Met9     ADVANCE   7      ; Имитация обслуживания
LEAVE    Smo, Zap      ; Освободить МКУ
;      Учет номеров транзактов, освободивших МКУ
ASSIGN    KolTp, X$KolTp      ; Запись в параметр цикла
Met10    ASSIGN    1, V$Rasm      ; Расчет индекса
TEST E    X*1, XN1, Met11      ; Есть ли этот номер транзакта?
SAVEVALUE P1, 0      ; Да, удалить
TRANSFER  , Met12      ; Выйти из цикла
Met11    LOOP      KolTp, Met10      ; Повторить или конец цикла
;      Учет номеров транзактов, покинувших очередь
Met12    ASSIGN    Dlina, Dlina      ; Запись в параметр цикла
Met15    TEST E    X*Dlina, XN1, Met16      ; Есть ли этот номер транзакта?
SAVEVALUE P$Dlina, 0      ; Да, тогда удалить
TRANSFER  , Met17      ; Выйти из цикла
Met16    LOOP      Dlina, Met15      ; Повторить или конец цикла
;
Met17    UNLINK    Накоп, Met3, 1      ; Одна заявка из очереди в МКУ
TERMINATE      ; Обслуженные заявки
Met1     TERMINATE      ; Потерянные заявки
;      Сегмент имитации неисправностей
GENERATE  , , , 1
Met20    ADVANCE   9.5      ; Имитация времени между отказами
SUNAVAIL  Smo      ; Перевод в недоступность
;      Удаление транзактов из МКУ
SAVEVALUE UdTp, 0      ; Обнуление счетчика
ASSIGN    KolTp, X$KolTp      ; Запись в параметр цикла
Met21    ASSIGN    1, V$Rasm      ; Расчет индекса
TEST NE   X*1, 0, Met22      ; Есть ли транзакт в МКУ?
DISPLACE  X*1, Met23      ; Да, тогда удалить
SAVEVALUE P1, 0      ; Снять с учета
Met22    LOOP      KolTp, Met21      ; Повторить или конец цикла
;
TEST E    SE$Smo, 1      ; МКУ пусто?
ADVANCE   1      ; Да, тогда ремонт МКУ

```

	SAVAIL	Smo		; Перевод в доступность
	TEST NE	X\$UdTp, X\$KолTp, Met20		; Выводить из Nakop?
	ASSIGN	1, (X\$KолTp-X\$UdTp)		; Да, записать в параметр цикла
Met24	UNLINK	Nakop, Met3, 1		; Одна заявка из очереди в МКУ
	LOOP	1, Met24		; Повторить или конец цикла
	TRANSFER	, Met20		
Met23	LEAVE	Smo, Zap		; Освободить МКУ
	SAVEVALUE	UdTp+, 1		; Счет удаленных транзактов
	TEST E	BV\$Kont3, 1		; Исправно ли МКУ?
	TRANSFER	, Met3		; Да, удаленные транзакты в МКУ
; Сегмент задания времени моделирования				
	GENERATE	19		
	SAVEVALUE	KIsp, V\$KIsp		; Расчет коэффициента использования
	TERMINATE	1		

Блок TEST, стоящий сразу после блока GENERATE сегмента имитации поступления и обслуживания заявок, проверяет наличие свободного места в очереди. При отсутствии свободного места транзакт направляется к блоку TERMINATE с меткой Met1 и уничтожается. Если свободное место есть (булева переменная Kont1 равна 1), учитывается номер транзакта и сохраняется в одной из ячеек X1...X3 (число ячеек равно длине очереди).

После этого, если СМО неисправно и свободное число каналов не может удовлетворить запрос (булева переменная Kont2 равна 0), блок TEST с меткой Met6 пропускает транзакт. Транзакт помещается блоком LINK в список пользователя с именем Nakop. В противном случае булева переменная Kont2 равна 1, транзакт направляется к блоку ENTER с меткой Met3 и занимает Zap каналов МКУ Smo.

При дальнейшем продвижении транзакта до блока ADVANCE с меткой Met9 учитывается его номер, как занявшего МКУ.

После обслуживания транзакт освобождает Zap каналов МКУ. Его номер снимается с учета номеров транзактов, занимающих МКУ и находящихся в очереди. При входе транзакта в блок UNLINK с меткой Met17 (этот транзакт является выводящим) из списка пользователя с именем Nakop выводится очередной транзакт и направляется к блоку ENTER с меткой Met3. Поскольку в МКУ есть свободные каналы для удовлетворения запроса, транзакт занимает их. Выводящий, он же обслуженный транзакт, входит в блок TERMINATE с меткой Met18 и уничтожается.

При возникновении неисправности транзакт сегмента имитации неисправностей войдет в блок SUNAVAIL и переведет МКУ Smo в недоступное состояние. Проходя далее через последовательность блоков до блока LOOP с меткой Met22, транзакт вызывает их срабатывание, вследствие чего все транзакты, которые находятся в это время в МКУ, удаляются из него и направляются к блоку LEAVE с меткой Met23. Удаленные транзакты проходят этот блок и освобождают МКУ Smo. Но так как МКУ недоступно, то блок TEST не пропускает дальше удаленные из МКУ транзакты.

В то же время, так как МКУ стало пустым, блок TEST пропускает транзакт-инициатор неисправности к блоку ADVANCE. Начинается имитация ремонта МКУ Smo. По истечении необходимого для этого времени МКУ Smo блоком SAVAIL переводится в доступное состояние.

Как только МКУ Smo стало доступным, удаленные из МКУ транзакты пропускаются блоком TEST (булева переменная Kont3 равна 1), направляются блоком TRANSFER к блоку ENTER с меткой Met3, подсчитываются блоком SAVEVALUE и занимают МКУ Smo. Обслуживание этих транзактов начинается сначала.

Во время недоступности МКУ при наличии места в очереди могут приниматься новые заявки, которые помещаются в список пользователя с именем Nakop. МКУ после восстановления доступности может иметь свободные каналы. Поэтому из списка пользователя нужно вывести, если есть, транзакты и направить их в МКУ. Если количество удаленных транзактов равно количеству транзактов, которые одновременно могут находиться в МКУ, то после восстановления доступности в МКУ нет свободных каналов для удовлетворения запроса. Вывод из списка пользователя не производится. В противном случае транзакты из списка пользователя выводятся. В качестве выводящего транзакта используется транзакт-инициатор неисправности.

Результаты пошагового моделирования сведены в табл. 5.2.

Таблица 5.2

Результаты моделирования примера 5.7

События	Транзакты						
	1(1)*	2(4)	3(5)	4(6)	5(7)	6(8)	7(9)
Вход в модель	2	4	6	8	10	12	14
В очереди транзакты	1	1, 4	1, 4, 5	1, 4, 5	4, 5, 7	4, 5, 7	4, 5, 7
Занятие МКУ	2	4/10,5	6/10,5		10,5	17,5	17,5
Удаление транзактов		9,5	9,5				
Освобождение МКУ	9	9,5/ 17,5	9,5/ 17,5		17,5		
Вывод из модели	9	17,5	17,5	8	10	12	14

* *Примечание.* Даны порядковые номера транзактов, которые должны обслуживаться МКУ Smo. В скобках даны номера этих же транзактов, присвоенные системой GPSS. Номера два (XN1=2) и три (XN1=3) имеют транзакты блоков GENERATE сегмента имитации неисправностей и сегмента задания времени моделирования соответственно.

За 19 единиц модельного времени обслужены четыре транзакта (первый, четвертый, пятый и седьмой). Обслуживание четвертого и пятого транзактов было прервано в $t = 9,5$ и они обслуживались повторно с $t = 10,5$ до $t = 17,5$. Седьмой транзакт обслуживался одновременно с ними.

Шестой транзакт был удален, так как в $t = 8$ в очереди находились первый, четвертый и пятый транзакты, т. е. в очереди не было места.

В $t = 9$ закончилось обслуживание первого транзакта и в очереди остались два транзакта. В $t = 9,5$ МКУ стало неисправным. В $t = 10$ поступил седьмой транзакт. В очереди было одно свободное место и седьмой транзакт поместили в список пользователя с именем Nakor. В очереди стало три транзакта (четвертый, пятый и седьмой). Поэтому из-за отсутствия свободных мест в очереди восьмой и девятый транзакты были удалены в $t = 12$ и $t = 14$ соответственно.

Седьмой транзакт был помещен в список пользователя с именем Nakor в $t = 10$. Когда в $t = 10,5$ МКУ стало исправным, его заняли только два удаленные ранее транзакта (четвертый и пятый). Вывод седьмого транзакта из списка пользователя осуществляется в цикле транзактом-инициатором неисправности. Число повторений цикла равно разности между количеством транзактов, которое может быть одновременно в МКУ, и количеством удаленных из МКУ транзактов.

Коэффициент использования МКУ S_{mo} равен 0,556. Системой GPSS он рассчитывался по формуле:

$$k = \frac{\sum_{i=1}^n \sum_{j=1}^m \alpha_i \cdot \Delta t_{ij}}{T \cdot E}, \quad (5.3)$$

$$\Delta t_{ij} = \Delta t_{2ij} - \Delta t_{1ij}, \quad (5.4)$$

Δt_{1ij} и Δt_{2ij} — абсолютное модельное время j -го занятия и освобождения МКУ соответственно i -м транзактом.

Для рассматриваемого примера 5.7 (см. табл. 5.2) получим:

$$E = 7, \quad T = 16, \quad \alpha = 2;$$

$$\Delta t_{11} = 9 - 2 = 7;$$

$$\Delta t_{21} = 9,5 - 4 = 5,5; \quad \Delta t_{22} = 17,5 - 10,5 = 7;$$

$$\Delta t_{31} = 9,5 - 6 = 3,5; \quad \Delta t_{32} = 17,5 - 10,5 = 7;$$

$$\Delta t_{41} = 0; \quad \Delta t_{51} = 17,5 - 10,5 = 7; \quad \Delta t_{61} = 0; \quad \Delta t_{71} = 0;$$

$$k_1 = \frac{2[7 + (5,5 + 7) + (3,5 + 7) + 7]}{7 \cdot 19} = \frac{74}{133} = 0,556.$$

Коэффициент использования МКУ без учета времени незавершенного обслуживания составит:

$$k_2 = \frac{2 \cdot 4 \cdot 7}{7 \cdot 19} = \frac{8}{19} = 0,421.$$

Увеличим длину очереди на одну заявку. В этом случае шестой транзакт не должен быть удален, поскольку в очереди будет одно место. Коэффициент k_2

использования МКУ останется неизменным, а k_1 увеличится, так как $\Delta t_{s1} = 17,5 - 10,5 = 7$, а не ноль:

$$k_1 = \frac{2[7 + (5,5 + 7) + (3,5 + 7) + 7 + 0,5 + 1,5]}{7 \cdot 19} = \frac{78}{133} = 0,586.$$

Рассмотрены методы моделирования неисправностей МКУ. Их применение позволит разработчику строить интересующие его модели МКУ с отказами.

5.5. Моделирование переключателей

Для моделирования такого оборудования, как переключатели, имеющие только два состояния, в GPSS используются логические ключи. Логический ключ может находиться в одном из двух состояний:

- включен (ON или 1);
- выключен (OFF или 0).

В зависимости от состояния ключа изменяется направление движения транзактов.

Логический ключ моделируется блоком LOGIC. Формат блока:

LOGIC X A

Операнд A — 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 был включен, будет выключен.

5.5.1. Проверка состояния логического ключа блоком GATE

Блок GATE имеет такой же формат, как и при проверке состояний ОКУ и МКУ, и такие же два режима работы:

GATE X A, [B]

Операнд A — имя или номер проверяемого ключа. Может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА*параметр.

Операнд B — метка блока, к которому будет направлен транзакт 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;

○ LR — равен 1, если логический ключ, заданный операндом A, выключен;
0 — если включен.

Пример 5.8. В условиях примера 5.3 используем логический ключ для организации такой же дисциплины обслуживания, как и с блоками SUNAVAIL и SAVAIL.

; Пример 5.8

; Определение MKУ

Sist STORAGE 7

; Сегмент имитации поступления и обслуживания транзактов

GENERATE 2,,5 ; Источник транзактов

GATE SV Klu, Met1 ; Включен ключ

GATE SNF Sist, Met1 ; MKУ свободно?

ENTER Sist, 3 ; Занять MKУ

ADVANCE 5 ; Обслуживание

LEAVE Sist, 3 ; Освободить каналы MKУ

TERMINATE 1 ; Обслуженные транзакты

Met1 TERMINATE 1 ; Потерянные транзакты

; Сегмент имитации недоступности

GENERATE ,,1 ; Транзакт-инициатор недоступности

LOGIC S Klu ; Включить ключ

Met2 ADVANCE 7 ; Время нахождения в доступном состоянии

LOGIC R Klu ; Выключить ключ

GATE SE Sist ; MKУ пусто

ADVANCE 1 ; Восстановление доступности

LOGIC S Klu ; Включить ключ

TRANSFER , Met2 ; Направить на очередной цикл

В $t = 0$ блок GENERATE сегмента имитации поступления и обслуживания транзактов генерирует один транзакт ($XN1=1$) и перестает быть активным. Этот транзакт входит в блок LOGIC, который включает ключ Klu, и далее задерживается блоком ADVANCE.

Блок GENERATE сегмента имитации поступления и обслуживания транзактов генерирует с интервалом 2 единицы модельного времени пять транзактов и также перестает быть активным.

В $t = 2$ первый транзакт ($XN1=2$) проходит первый и второй блоки GATE, так как ключ Klu включен и MKУ свободно, и занимает три свободных канала. Обслуживание этого транзакта закончится в $t = 2 + 5 = 7$.

Второй транзакт ($XN1=3$) в $t = 4$ также пройдет оба блока GATE и займет следующие три канала MKУ Sist. Свободным останется один канал. Обслуживание второго транзакта закончится в $t = 4 + 5 = 9$.

В $t = 6$ третий транзакт пройдет первый блок GATE сегмента имитации поступления и обслуживания транзактов и войдет во второй блок GATE. Поскольку MKУ Sist имеет один канал свободным, условие, заданное в блоке GATE, выполнится и транзакт пройдет к блоку ENTER. Но одного канала недостаточно

для удовлетворения запроса, поэтому транзакт помещается в список задержки МКУ.

В $t = 7$ ключ Klu будет выключен и МКУ будет закрыто для вновь поступающих транзактов. В это же время закончится обслуживание первого транзакта и третий транзакт из списка задержки займет МКУ.

Так как в МКУ $Sist$ находятся второй и третий транзакты, т. е. оно не пусто, то транзакт, вызвавший выключение ключа, будет задержан блоком GATE сегмента имитации недоступности, потому что заданное в нем условие не выполняется.

В $t = 8$ четвертый транзакт ($XN1=5$) первым блоком GATE сегмента имитации поступления и обслуживания транзактов, так как ключ Klu выключен, направляется к блоку TERMINATE с меткой $Met1$ и уничтожается.

В $t = 9$ закончится обслуживание второго транзакта. В МКУ останется один транзакт, условие в блоке GATE сегмента имитации недоступности снова не выполнится и вызвавший выключение ключа транзакт не войдет в блок задержки ADVANCE.

В $t = 10$ пятый транзакт ($XN1=6$) первым блоком GATE сегмента 1 также направляется к блоку TERMINATE с меткой $Met1$ и уничтожается.

В $t = 7 + 5 = 12$ закончится обслуживание третьего транзакта. Транзакт, вызвавший выключение ключа, пройдет через блок GATE к блоку задержки ADVANCE и в $t = 12 + 1 = 13$ будет включен ключ Klu . Транзактам откроется путь на МКУ $Sist$.

Таким образом, в данном примере будут обслужены три транзакта (первый, второй и третий), в примере 5.3 — также три транзакта (первый, второй и третий). При этом время окончания обслуживания составит 12 и 15 единиц модельного времени соответственно, т. е. при использовании логического ключа время обслуживания первых трех транзактов заканчивается раньше.

ЗАМЕЧАНИЕ

Если до поступления в модель первого транзакта от блока GENERATE сегмента имитации поступления и обслуживания транзактов не включить ключ Klu , то модель работать будет неверно, так как все транзакты будут направляться к блоку TERMINATE с меткой $Met1$ и уничтожаться. Ключ Klu будет включен лишь после первого перевода из недоступного состояния в доступное. Поэтому в сегменте имитации недоступности предусмотрен блок LOGIC, включающий в $t = 0$ ключ Klu .

5.5.2. Проверка состояния логического ключа блоком TEST и булевой переменной

При проверке состояния логического ключа блок TEST имеет такой же формат записи и те же условные операторы, что и при проверке состояний ОКУ и МКУ.

В булевой переменной используется только СЧА LS. Если в булевой переменной записать LR, возникнет ошибка «Предполагается фактор выражения».

Пример 5.9. МКУ переводится в недоступное состояние через случайные интервалы времени. Организуем две дисциплины обслуживания и исследуем как они влияют, например, на вероятность обслуживания запросов.

Для организации недоступности МКУ при первой дисциплине обслуживания используем логический ключ и блок GATE, а не блоки SUNAVAIL и SAVAIL. Так как емкость МКУ *Sist* равна 8 и не делится на 3 без остатка, то блок GATE будет пропускать транзакт при недостаточной для удовлетворения запроса свободной емкости МКУ. Восстановление доступности МКУ начинается после того, как МКУ будет пусто.

; Пример 5.9. Вариант 1

; *Определение МКУ и арифметической переменной*

```
Zap      EQU      3
Sist     STORAGE  8
Ver      FVARIABLE N$Per/N$Vse
```

; Сегмент имитации поступления и обслуживания запросов

```
GENERATE (Exponential(310,0,10.3)) ; Источник запросов
Vse      GATE LS   Klu, Met1        ; Ключ включен?
         GATE SNF  Sist, Met1       ; МКУ свободно
         ENTER    Sist, Zap         ; Занять МКУ
         ADVANCE  (Normal(231,25.3,1.7)) ; Обслуживание
         LEAVE    Sist, Zap         ; Освободить каналы МКУ
Per      TERMINATE ; Обслуженные запросы
```

; Сегмент имитации недоступности

```
GENERATE , , 1 ; Транзакт-инициатор недоступности
LOGIC S   Klu ; Включить в ключ
Met2     ADVANCE (Exponential(11,0,600.7)); Время доступности
LOGIC R   Klu ; Выключить ключ
GATE SE   Sist ; МКУ пусто?
ADVANCE  3.2, 1.3 ; Восстановление доступности
LOGIC S   Klu ; Включить ключ
TRANSFER , Met2 ; Направить на очередной цикл
Met1     TERMINATE
```

; Сегмент задания времени моделирования

```
GENERATE 3600
SAVEVALUE Ver, V$Ver ; Расчет вероятности обслуживания
TERMINATE 1
```

Блок GATE сегмента имитации поступления и обслуживания транзактов, стоящий перед блоком ENTER, проверяет состояние ключа *Klu*. Если ключ включен, транзакт проходит ко второму блоку GATE. Этот блок определяет наличие свободных каналов в МКУ *Sist*. Если свободные каналы имеются, транзакт проходит к следующему блоку ENTER и занимает три из восьми каналов МКУ *Sist*. Если условия в обоих блоках GATE не выполняются, транзакт направляется к блоку TERMINATE с меткой *Met1* для уничтожения.

Перевод МКУ *Sist* в недоступное состояние производится блоками сегмента имитации недоступности. Единственный транзакт, сгенерированный блоком GENERATE сегмента 2, после включения ключа *Klu* и задержки поступает в блок

LOGIC, который переводит логический ключ Klu в состояние выключено. Далее на пути транзакта находится блок GATE. Если MKY Sist пусто, транзакт проходит к блоку ADVANCE и начинается имитация восстановления доступности, например, технического обслуживания. В противном случае транзакт ожидает завершения обслуживания транзактов, находившихся в MKY Sist к моменту перевода его в недоступное состояние.

После включения ключа Klu транзакт блоком TRANSFER направляется к блоку ADVANCE с меткой Met2 и процесс имитации доступности и недоступности MKY Sist повторяется.

Приведем результаты моделирования.

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SIST	8	2	0	6	8049	1	5.658	0.707
SAVEVALUE					RETRY		VALUE	
VER					0		0.759	

Вторая дисциплина обслуживания предусматривает, что в случае незаполненности MKY, но недостаточности свободной емкости для удовлетворения запроса, транзакт не помещается в список задержки MKY. Для организации второй дисциплины обслуживания используем логический ключ, булеву переменную и блок TEST.

; Пример 5.9. Вариант 2

; *Определение MKY, арифметической и булевых переменных*

```
Zap      EQU      3
Sist     STORAGE  8
Ver      VARIABLE N$Per/N$Vse
Kont1    BVARIABLE LS$Klu'AND'((8-S$Sist)>=Zap)
Kont2    BVARIABLE SE$Sist
```

; **Сегмент имитации поступления и обслуживания запросов**

```
GENERATE (Exponential(310,0,10.3)) ; Источник запросов
Vse      TEST E   BV$Kont1,1,Met1 ; Доступно и есть каналы MKY
ENTER    Sist,Zap ; Занять MKY
ADVANCE (Normal(231,25.3,1.7)) ; Обслуживание
LEAVE    Sist,Zap ; Освободить каналы MKY
Per      TERMINATE ; Обслуженные запросы
```

; **Сегмент имитации недоступности**

```
GENERATE ,,1 ; Транзакт-инициатор недоступности
LOGIC S   Klu ; Включить ключ
Met2     ADVANCE (Exponential(11,0,600.7)); Время доступности
LOGIC R   Klu ; Выключить ключ
TEST E    BV$Kont2,1 ; MKY пусто?
ADVANCE 3.2,1.3 ; Восстановление доступности
LOGIC S   Klu ; Включить ключ
TRANSFER ,Met2 ; Направить на очередной цикл
Met1     TERMINATE ; Обслуженные запросы
```

; Сегмент задания времени моделирования

```
GENERATE      3600
SAVEVALUE    Ver, V$Ver           ; Расчет вероятности
TERMINATE    1
```

Блок TEST сегмента 1 пропускает транзакт на МКУ Sist в случае, если булева переменная Kont1 равна 1. Это будет тогда, когда ключ Klu включен и когда оставшейся свободной емкости МКУ достаточно для удовлетворения запроса.

Блок TEST сегмента 2 пропускает транзакт к следующему блоку задержки ADVANCE, если булева переменная Kont2 равна 1, т. е. когда МКУ Sist пусто.

В остальном работа модели аналогична. Приведем результаты моделирования.

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
SIST	8	5	0	6	5529	1	3.891	0.486

SAVEVALUE	RETRY	VALUE
VER	0	0.521

Из сравнения полученных результатов видно, что при одних и тех же данных и второй дисциплине обслуживания коэффициент использования МКУ и вероятность обслуживания уменьшились в 1,52 раза.

5.6. Примеры построения моделей систем с многоканальным устройством обслуживания

Пример 5.10. Пункты наблюдения (ПН) ведут мониторинг средств связи (СС). По мере сбора данных с интервалом времени $T1 \pm T2$ мин ПН передают их для обработки на ЭВМ. На обработку затрачивается $T3 \pm T4$ мин машинного времени. При этом ЭВМ выполняет другие заявки, поступающие с интервалом времени $T5 \pm T6$ мин. Эти заявки обрабатываются в течение времени $T7 \pm T8$ мин. ПН приступают к ведению мониторинга СС практически одновременно.

Построить модель мониторинга СС в течение 10 часов с целью определения среднего времени одного этапа мониторинга одним ПН.

Вариант данных, необходимых для разработки модели, примем следующий: $T1 = 30$ мин, $T2 = 5$ мин, $T3 = 4$ мин, $T4 = 2$ мин, $T5 = 15$ мин, $T6 = 10$ мин, $T7 = 10$ мин, $T8 = 4$ мин, количество ПН — 15.

В данном случае есть два вида ограничивающих условия. Во-первых, имеется лишь одна ЭВМ. Во-вторых, имеется фиксированное число ПН. Тогда ЭВМ — одноканальное устройство, а ПН — транзакт. В реальной системе после того как ПН обработает данные на ЭВМ, он возвращается к сбору данных. В модели этот возврат может быть обеспечен блоком TRANSFER. Для того, чтобы ограничить число транзактов, циркулирующих в модели, числом, равным числу ПН, необходимо использовать операнд D блока GENERATE. Для имитации сбора данных целесообразно использовать МКУ с емкостью, равной числу ПН.

Программа модели и результаты моделирования после 5 000 прогонов приведены ниже.

; Пример 5.10

```

TR1      STORAGE      15          ; Количество ПН
; Сегмент имитации мониторинга
          GENERATE     ,, ,15      ; Количество транзактов = количеству ПН
Rasv     ASSIGN       1,AC1        ; Время начала этапа мониторинга
          ENTER        Tr1         ; Занятие МКУ – начало этапа сбора данных
          ADVANCE      30,5        ; Сбор данных
          LEAVE        Tr1         ; Окончание этапа сбора данных
          QUEUE        Dlina       ; Встать в общую очередь заявок на ЭВМ
          QUEUE        Dlina1      ; Встать в очередь передачи данных
          SEIZE        Ewm         ; Занять ЭВМ
          DEPART      Dlina1      ; Покинуть очередь передачи данных
          DEPART      Dlina       ; Покинуть общую очередь заявок на ЭВМ
          ADVANCE      4,2         ; Передача данных
          RELEASE     Ewm         ; Освободить ЭВМ
          ASSIGN      2,AC1        ; Время конца этапа мониторинга
          SAVEVALUE   1+, (P2-P1) ; Суммарное время этапов
KolDok   TRANSFER     ,Rasv       ; На очередной этап мониторинга
; Сегмент имитации выполнения других заявок
          GENERATE     15,10       ; Источник других заявок
          QUEUE        Dlina       ; Встать в общую очередь заявок на ЭВМ
          SEIZE        Ewm         ; Занять ЭВМ
          DEPART      Dlina       ; Покинуть общую очередь заявок на ЭВМ
          ADVANCE      10,4        ; Выполнение запроса
          RELEASE     Ewm         ; Освободить ЭВМ
          TERMINATE   ; Выполненные запросы
; Сегмент задания времени моделирования
          GENERATE     600
          SAVEVALUE   Vrem, (X1/N$KolDok) ; Среднее время этапа мониторинга
          TERMINATE   1
QUEUE    MAX    CONT.   ENTRY   ENTRY(0)  AVE.CONT.  AVE.TIME  AVE. (-0)
DLINA    34     20      450076   1          21.905    146.007   146.008
DLINA1   15     13      249954   0          12.167    146.030   146.030
SAVEVALUE      RETRY      VALUE
VREM           0          180.037

```

Среднее время одного этапа мониторинга одним ПН составляет 3 часа: значение сохраняемой ячейки с именем VREM равно 180,037 мин, т. е. 3 часа.

В приведенной модели предполагается, что число ПН остается неизменным. Однако на практике это может быть не так. Например, будут потери или вследствие какой-либо необходимости потребуется уменьшить или увели-

чить число ПН. В случае моделирования с учетом потерь можно, например, добавить блок TRANSFER в статистическом режиме. При заданной доле потерь, в соответствии с ней будет уменьшаться число ПН, т. е. не все они будут возвращаться и передавать данные.

Для имитации восполнения потерь ПН или их числа целесообразно ввести дополнительно блок GENERATE, задав в нем закон пополнения ПН, и направлять эти ПН с помощью еще одного дополнительного блока TRANSFER.

Пример 5.11. Средства связи (СС) после сборки проходят проверку на функционирование на двух пунктах контроля (ПК). Если они функционируют нормально, то отправляются по назначению. В противном случае они возвращаются на настройку, которая выполняется на одном пункте, а затем снова на ПК для проверки на функционирование. Интервалы первичного поступления на ПК, время проверки и время настройки распределены по экспоненциальному закону со средними значениями 50, 90 и 200 мин соответственно. Примерно 85% СС проходят проверку успешно. После настройки около 15% СС бракуются.

Построить модель с целью определения количества СС, проверенных за 8 часов и готовых к применению, а также отправленных в брак.

Программа модели представлена ниже.

Блоки ENTER и LEAVE образуют МКУ с именем Kont, имитирующее работу двух ПК. Блоки SEIZE и RELEASE представляют в модели ОКУ — пункт настройки СС. Первая пара блоков QUEUE и DEPART осуществляет сбор информации об очереди Ar1 к МКУ Kont, а вторая пара — об очереди Ar2 к ОКУ Nastr. Первый блок TRANSFER осуществляет статистическую передачу транзактов, движение которых в модели предположительно соответствует движению СС в реальной системе. Второй блок TRANSFER также в статистическом режиме осуществляет передачу транзактов — СС в брак и на ПК.

; Пример 5.11

Kont STORAGE 2 ; Число пунктов контроля

; Сегмент имитации работы пунктов контроля

```

GENERATE (Exponential(47,0,80)) ; Источник СС
Bac QUEUE Ar1 ; Встать в очередь на пункт контроля
ENTER Kont ; Занять один пункт контроля
DEPART Ar1 ; Покинуть очередь на пункт контроля
ADVANCE (Exponential(67,0,90)) ; Контроль
LEAVE Kont ; Освободить один пункт контроля
TRANSFER .15, Rem1 ; 15% отправить на проверку
SrSv TERMINATE ; 85% готовых СС

```

; Сегмент имитации работы пункта проверки

```

Rem1 QUEUE Ar2 ; Встать в очередь на проверку
SEIZE Nastr ; Занять пункт проверки
DEPART Ar2 ; Покинуть очередь на пункт проверки
ADVANCE (Exponential(77,0,200)) ; Проверка

```

```

RELEASE      Nastr      ; Освободить пункт проверки
TRANSFER     .8, ,Vac   ; 20% отправить в брак
Brak         TERMINATE  ; Бракованные СС

```

; Сегмент задания времени моделирования

```

GENERATE      4800
SAVEVALUE     IspSS, (INT(N$SrSv/5000))
SAVEVALUE     BrakSS, (INT(N$Brak/5000))
TERMINATE     1
START         5000

```

По результатам 5 000 прогонов за 8 часов проверено 58 и забраковано 2 СС. Модель запускается сразу после трансляции, так как указана команда START. Количество прогонов (5 000) используется в расчетах при определении количества исправных и забракованных СС в блоках SAVEVALUE. При изменении числа прогонов нужно внести изменения и в операнд В блоков SAVEVALUE.

Пример 5.12. На узел связи, имеющий три канала, поступают заявки на передачу сообщений с интервалами, распределенными по экспоненциальному закону со средним значением 120 сек. Время передачи сообщения также имеет экспоненциальное распределение со средним значением 300 сек. Если все три канала заняты, то с вероятностью 0,15 в дальнейшем не будет предпринято повторных попыток передать эти же сообщения, и они будут потеряны. С вероятностью 0,85 будут предприняты попытки снова передать эти сообщения. Причем с вероятностями 0,35 и 0,65 повторные попытки будут предприняты через интервалы времени, распределенные по экспоненциальному закону со средними значениями 370 и 425 сек соответственно.

Построить модель с целью определения относительного количества заявок, по которым были переданы сообщения со второй и более попыток.

Программа модели системы с отказами и повторными попытками передачи сообщений приведена ниже.

; Пример 5.12

; Сегмент имитации работы узла связи

```

Met6          GENERATE      (Exponential(54,0,120)) ; Источник заявок
Met1          ASSIGN       1,1                      ; Код 1 – первая попытка
Met1          TRANSFER     BOTH,,Met2               ; Не заполнен МКУ?
Met1          ENTER        Use13                    ; Нет, тогда занять канал
Met1          ADVANCE      (Exponential(45,0,300)) ; Передача
Met1          LEAVE        Use13                    ; Освободить канал
Met1          TEST E       P1,1,Met5                ; Первая попытка?
Met5          TERMINATE    ; Переданные сообщения с первой попытки
Met5          TERMINATE    ; Переданные сообщения не с первой попытки
Met2          TRANSFER     .15,,Met3                ; 0,15 заявок не повторяются
Met2          INDEX        1,1                      ; Фиксация очередной попытки
Met2          TRANSFER     .35,,Met4                ; Деление потока повторных заявок
Met2          ADVANCE      (Exponential(54,0,425)) ; Время очередной попытки

```

```

Met4      TRANSFER      ,Met1          ; Снова на узел связи
          ADVANCE      (Exponential(54,0,370)) ; Время очередной попытки
          TRANSFER      ,Met1          ; Снова на узел связи
Met3      TERMINATE
; Сегмент задания времени моделирования
          GENERATE      4320
          SAVEVALUE     VPovt, (N$Met5/N$Met6) ; Доля повторов
          TERMINATE     1

```

В параметр 1 транзактов, поступающих в модель, заносится код 1 — признак первой попытки передать сообщение. Далее транзакты попадают в блок TRANSFER с меткой Met1, работающий в режиме BOTH. Если в момент поступления транзакта в МКУ User3 хотя бы один канал свободен, то блок TRANSFER направит транзакт в следующий блок, т.е. в блок ENTER. Если же в момент поступления три канала МКУ заняты, и поэтому блок ENTER отказывает во входе, транзакт будет направлен в блок TRANSFER с именем Met2, работающий в статистическом режиме. С вероятностью 0,85 транзакты из этого блока передаются в следующий блок INDEX, который увеличивает значение параметра 1 на 1 — фиксируется очередная попытка передать сообщение. Далее поток повторных попыток блоком TRANSFER в статистическом режиме разбивается на две части. Каждая из частей задерживается в соответствующем блоке ADVANCE на случайное время и с помощью блока TRANSFER, работающего в безусловном режиме, передается вновь на вход модели в блок с именем Met1. С вероятностью 0,15 транзакты из блока с меткой Met1 передаются в блок TERMINATE с меткой Met3 для уничтожения.

По результатам 5 000 прогонов относительное количество сообщений, переданных со второй и более попыток, составляет 0,316.

Примеры моделей с МКУ и переключателями будут приведены еще в последующих главах.