

## GPSS

[Общие сведения](#)

[Блоки и транзакты GPSS](#)

[Принципы построения и организации](#)

[Классификация блоков GPSS](#)

[Формат команды в GPSS](#)

[Построение выражений в GPSS](#)

[Структура управляющей программы в GPSS](#)

[Объекты запоминающей категории, ячейки, матрицы ячеек](#)

[Моделирование СМО](#)

[Методика отладки в GPSS](#)

[Графические возможности GPSS](#)

[Table, Tabulate](#)

[Блоки GPSS](#)

[Управляющие блоки](#)

[Блоки создания и уничтожения транзактов](#)

[Блоки по работе с обслуживающими устройствами](#)

[Блоки по синхронизации транзактов](#)

[Блоки PREEMPT и RETURN](#)

[Блоки, изменяющие параметры транзактов](#)

## GPSS

### Общие сведения

**GPSS** – общецелевая система моделирования. Содержит словари и грамматику, с помощью которых разрабатываются имитационные модели сложных дискретных систем.

Возможности GPSS:

- Многозадачность
- Использование виртуальной памяти
- Интерактивность
- Графический интерфейс пользователя
- Визуализация процесса моделирования

Пакет построен в предположении, что моделью сложной дискретной системы является описание ее элементов и логических правил их взаимодействия в процессе функционирования системы.

Для определения класса моделируемых систем можно выделить конечный набор абстрактных элементов, называемых **объектами**.

Набор логических правил также ограничен и может быть описан небольшим количеством стандартных **операций**.

Объекты GPSS делят на 7 категорий и 14 типов:

- Динамическая - транзакты
- Операционная - блоки
- Аппаратная - устройства, память, ключи
- Вычислительная - переменные, функции
- Статистическая - таблицы, очереди
- Запоминающая - ячейки, матрицы
- Группирующая - списки пользователя, группы

**Динамические объекты** – транзакты (сообщения), которые представляют единицы исследуемых потоков и производят ряд определенных действий, продвигаясь по фиксированной структуре (построенной из блоков).

**Операционные объекты** – блоки, задающие логику функционирования модели и пути движения транзактов. Как правило между объектами аппаратной категории.

**Аппаратные объекты** – абстрактные элементы, на которые может быть декомпозировано оборудование реальной системы. Транзакты, воздействуя на эти объекты могут изменять их состояние, влиять на движение других транзактов.

**Вычислительная категория** - применяется, когда связи между компонентами моделируемой системы наиболее просто и компактно описываются при помощи математических соотношений.

Каждому объекту присущи **атрибуты**, описывающие его состояние в данный момент времени. Значения атрибутов могут быть арифметическими или логическими. Большая часть атрибутов недоступна пользователю.

Атрибуты, которые необходимо адресовать называются **стандартными**.

В процессе моделирования системы и взаимодействия объектов происходит изменение атрибутов и преобразование их значений.

**События** - преобразования значений атрибутов.

Управляющие средства языка - команды (start, show ...).

## Блоки и транзакты GPSS

Блок - статический объект, (представляющий собой подпрограмму, реализованную на языке макроассемблера). Последовательность блоков модели отражает направления, по которым происходит перемещение транзактов.

Транзакт - динамический элемент, перемещающийся по GPSS модели. Представляют собой не только единицы потока, но и описания динамических процессов в реальных системах. Они могут описывать реальные физические объекты (автомобиль на бензоколонке) и нефизические (канальные программы).

Любая модель GPSS должна содержать блоки и транзакты.

## Принципы построения и организации

Передача управления от блока к блоку осуществляется посредством движения транзактов в модельном времени (обращение к подпрограммам блоков происходит через движение транзактов от блока к блоку).

С блоками непосредственно связаны:

- Операционные блоки - изменяют процесс моделирования
- Блоки ввода - производят печать промежуточных результатов
- Команды - управляющие процессом моделирования
- Команды редактирования.

## Классификация блоков GPSS

Блоки используются для описания функций и управляют движениями транзактов. У каждого блока имеются два стандартных числовых атрибута (СЧА):  $W_n$  – счетчик входов в блок (ожидаящий счетчик), который содержит в себе номер текущего транзакта, находящегося в блоке.  $N$  – общий счетчик транзактов, поступивших в блок с начального момента моделирования или с момента обнуления.

Управляющие команды:

- RESET – сбрасывает временные счетчики
- CLEAR – обнуляет всю модель.

По назначению вводится следующая классификация блоков:

1. Блоки, осуществляющие модификацию атрибутов транзактов.
  - a. Временная задержка:
    - i. ADVANCE (при использовании функций – потребуется производить масштабирование)
  - b. Генерирование и уничтожение транзактов:
    - i. GENERATE (второй параметр – модификатор)
    - ii. TERMINATE
    - iii. SPLIT – создает копии одного «ансамбля» транзактов
    - iv. ASSEMBLE
  - c. Синхронизация движения нескольких транзактов:
    - i. MATCH
    - ii. GATHER
  - d. Изменение параметров транзактов:
    - i. ASSIGN
    - ii. INDEX
    - iii. MARK
  - e. Изменение приоритета
    - i. PRIORITY
2. Блоки, изменяющие последовательность передвижения транзактов (блоки передачи управления)

- a. TRANSFER
  - b. LOOP
  - c. TEST (устройство – занято ли, память – свободна ли)
  - d. GATE
3. Блоки, связанные с группирующей категорией
- a. JOIN
  - b. REMOVE
  - c. EXAMINE
  - d. SCAN
  - e. ALTER
4. Блоки, сохраняющие необходимые значения для дальнейшего использования
- a. SAVEVALUE
  - b. MSAVEVALUE
5. Блоки, организующие использование объектов аппаратной категории
- a. Устройств:
    - i. SEIZE – RELEASE
    - ii. PREEMPT – RETURN
    - iii. FAVAIL (facilities avail)
    - iv. MFAVAIL
    - v. FUNAVAIL
  - b. Памяти:
    - i. ENTER – LEAVE
    - ii. SAVAIL – SUNAVAIL
  - c. Ключи:
    - i. LOGIC
6. Блоки, обеспечивающие получение статистических результатов
- a. QUEUE – DEPART
  - b. TABLE
  - c. TABULATE
7. Специальные блоки
- a. TRACE, SELECT, HELP
8. Блоки для организации цепей
- a. LINK
  - b. UNLINK
9. Вспомогательные блоки
- a. WRITE
  - b. SAVE
  - c. LOAD
  - d. REPORT
  - e. UPDATE

## Формат команды в GPSS

При описании схемы алгоритма с помощью языковых средств, каждый блок определяется с помощью отдельной команды независимо от редактора. Команда может включать в себя слева направо:

- Поле метки: символический адрес блока
- Поле операции: наименование типа блока
- Поле операндов: перечисляются все операнды, присущие блоку, A..H
- ; комментарий

## Построение выражений в GPSS

Выражения могут применяться в переменных и операторах GPSS. При применении в переменных выражения определяются командами GPSS. При применении в операторах GPSS выражения определяются как часть языка PLUS.

Язык PLUS является встроенным в GPSS World языком программирования и предназначен для удовлетворения потребностей пользователей в управлении данными, особых вычислительных алгоритмах и других операциях, которые не могут быть обеспечены средствами операторов и команд GPSS. Новый блок PLUS позволяет вызывать PLUS-процедуру, которая выполняется как любой другой блок GPSS. Это позволяет создавать собственные блоки с очень сложной структурой. Кроме того, выражения, записанные с помощью синтаксиса языка PLUS (так называемые PLUS-выражения) могут использоваться в качестве операндов блоков и команд. И выражения могут содержать вызовы встроенных процедур или процедур пользователя.

Язык PLUS включает в себя следующие операторы:

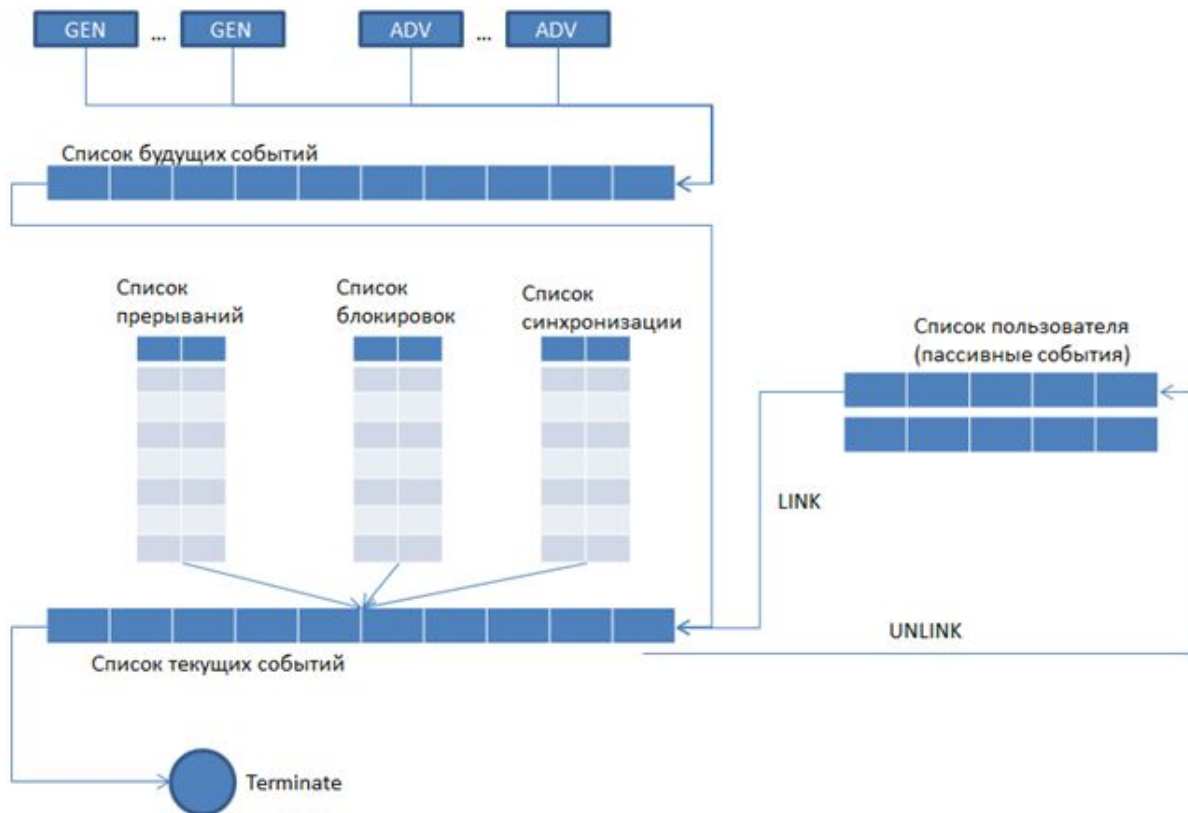
- Оператор присваивания;
- Вызов процедуры;
- BEGIN;
- DO...WHILE;
- END;
- EXPERIMENT;
- GOTO;
- IF...THEN...ELSE;
- PROCEDURE;
- RETURN;
- TEMPORARY.

EXPERIMENT – предназначен для определения специальной процедуры пользователя, которая управляет несколькими повторяющимися процессами моделирования.

PROCEDURE – используется для определения процедур пользователя. Процедуры обладают глобальной областью действия, т.е. могут вызываться в любом месте модели.

TEMPORARY – предназначен для создания временных переменных пользователя и матриц, существующих только во время выполнения процедуры.

## Структура управляющей программы в GPSS



Поддерживается сложная структура списков. С целью уменьшения времени на просмотр этих списков система ведет два основных списка: СТС и СБС.

Список **текущих событий** – включает все события, запланированные на текущий момент модельного времени, независимо от того условные они или безусловные.

Действия управляющей программы:

1. Просматривает этот список
2. Пытается переместить по модели те транзакты, для которых выполнены условия (время).
3. Если в этом списке таких транзактов нет, то обращается к списку будущих событий
4. Переносит все события, которые запланированы на ближайший момент модельного времени
5. Заново просматривает список.

Перенос также осуществляется в случае совпадения текущего времени со временем первого события в СБС.

В СТС транзакты размещены в порядке уменьшения приоритетов. Транзакты с одинаковыми приоритетами размещаются в соответствии с последовательностью поступления в список. Каждый транзакт СТС может находиться или в активном состоянии, или в состоянии задержки.

1. В начальный момент (при выполнении оператора управления START) управляющая программа обращается ко всем блокам GENERATE
2. Каждый из них планирует момент появления транзактов и заносит их в СБС.

3. Программа просматривает CTC (на данный момент в нем пусто).
4. Программа просматривает СБС все транзакты на ближайшее время, заносит в CTC и начинает продвигать модельное время.
5. Если продвижение транзакта было задержано по какой-либо причине, не связанной с блоком ADVANCE, то он остается в списке текущих событий и управляющая программа требует заново перемещать его далее по блокам.
6. Если транзакт входит в блок аванс, то выход из этого блока для транзакта планируется и заносится в СБС.

CTC и СБС можно вывести на экран с помощью управляющей команды EVENTS(?) или в окне списков GPSS.

Список **блокировок** – список транзактов, которые ожидают изменения состояния ресурса. Существует шесть видов таких списков, связанных с устройствами, семь связанных с памятьми, и два связанных с логическими ключами.

С устройствами используются списки для: занятых и незанятых, доступных и недоступных, работающих с прерываниями и без прерываний.

Памяти: списки для заполненного, незаполненного, пустого непустого доступного недоступного.

Ключи: списки для включенных\выключенных ключей.

Список **прерываний** – содержит прерванные во время обслуживания транзакты, а также транзакты, вызвавшие прерывания. Данный список используется для организации обслуживания одноканальных устройств по абсолютным приоритетам, что дает возможность организовать приоритетные дисциплины обслуживания.

Список **синхронизации** – содержит транзакты, которые на текущий момент ждут сравнения. Список работает с транзактами, полученными с помощью блока SPLIT – создает транзакты-копии, принадлежащие одному ансамблю; синхронизацию движения транзактов одного семейства выполняется блоком MATCH (также ASSEMBLE собирает все транзакты-копии и выдает один начальный, GATHER собирает заданное количество транзактов и задерживает их пока не соберется нужное число копий).

Остановленные процессы находятся в СБС, СС и списке блокировок.

Список пользователя содержит транзакты, выведенные пользователем с помощью LINK и помещенные в список прерываний как неактивные. Управляющей программе они будут недоступны до UNLINK.

## Объекты запоминающей категории, ячейки, матрицы ячеек

Объекты запоминающей категории обеспечивают обращения к сохраненным значениям. Ячейки и матрицы ячеек сохраняемых величин используются для сохранения некоторой пользовательской числовой информации. Любой транзакт может осуществить запись в эти объекты. Впоследствии, записанную информацию может считать любой транзакт. Таким образом, эти объекты являются глобальными и доступны из любой части модели.

Ячейки используются для записи и хранения в процессе моделирования текущих значений СЧА.

Занесение информации в ячейку производится при помощи блока SAVEVALUE формата:

**SAVEVALUE A, B, C**, где A - имя ячейки, B - присваиваемое значение, C - тип ячейки. Если после A стоит +/-, то значение поля B прибавляется/вычитается из текущего значения ячейки.

Типы ячеек:

- Полусловная – XH
- Полнословная – XF (по умолчанию)
- С плавающей точкой – XL

Начальное значение по умолчанию - 0. Инициализация может быть проведена с помощью блока

**INITIAL Ячейка1, Значение1, ... ЯчейкаN, ЗначениеN**

Матрицы ячеек дают возможность упорядочить значения, которые нужно сохранить. Матрицы перед использованием необходимо описать. Это производится при помощи блока MATRIX в формате:

**j MATRIX A, B, C**, где j - имя матрицы. A - неиспользуемое поле (для совместимости с прежними версиями GPSS). B - количество строк матрицы. Операнд должен быть константой. C - количество столбцов матрицы.

Изменение элементов матриц во время моделирования производится блоком:

**MSAVEVALUE A B C D**, где A - имя или номер матрицы, B - номер строки, C - номер столбца, D - новая величина элемента матрицы.

Обращение к i, j-ой ячейке матрицы: **MX\$ИМЯ\_МАТРИЦЫ(i,j)**

## Моделирование СМО

СМО моделируются при помощи объектов аппаратной категории:

- Одноканальные устройства (ОКУ)
- Многоканальные устройства (МКУ)
- логические ключи

Каждое ОКУ в любой момент времени может обслуживать только один транзакт. При появлении нового он должен либо дождаться освобождения устройства, либо пойти другим маршрутом, либо прервать (если у него более высокий приоритет).

Когда транзакт поступает на ОКУ в модели требуется пропустить время, необходимое для его обслуживания.

Каждому ОКУ присваивается символическое имя (аналогично имени блока).



Варианты функционирования ОКУ: занятие и освобождение, прерывание, недоступность и ее восстановление.

Параллельные однородные устройства в GPSS могут быть представлены в виде МКУ. МКУ может использоваться несколькими транзактами одновременно. МКУ создается командой

**Name STORAGE A** , где Name - имя МКУ, A - емкость МКУ (целое положительное).

## Методика отладки в GPSS

Отладка производится пошагово при помощи команды **STEP A**, A - количество входов активного транзакта в блоки, которое производится при каждом выполнении команды. Обычно A = 1 и транзакт продвигается к следующему блоку.

Отладку удобно проводить находясь в окне блоков. Для продолжения моделирования необходимо ввести команду CONTINUE в командную строку.

Для упрощения процесса отладки в GPSS World и GPSS Studio предусмотрены различные средства (в GPSS Studio - режим отладки) кнопки stop, continue и т.п.

## Графические возможности GPSS

В системе моделирования GPSS World имеется возможность графического отображения моделируемых величин (например, длин очередей, сохраняемых величин и т.д.) в процессе моделирования. Другими словами, в ходе моделирования можно наблюдать на экране графики изменения анализируемых величин во времени. Для этого требуется выполнить следующее:

- 1) подготовить текст модели и откомпилировать ее, используя команду меню Command – Create Simulation;
- 2) выбрать команду меню Window – Simulation Window – Plot Window;
- 3) в появившемся окне Edit Plot Window указать параметры графиков, которые требуется построить (эти действия будут подробно рассмотрены в примере 8.3);
- 4) запустить модель на выполнение, используя команду меню Command – Start. В процессе моделирования отображаются заданные графики.

Горизонтальная ось (абсцисса) всегда представляет собой ось времени, вертикальная ось (ордината) – ось моделируемой величины. В ходе моделирования может строиться один или несколько графиков. В одной системе координат (т.е. в одном окне) можно отобразить изменение одной или нескольких моделируемых величин.

Необходимо, однако, учитывать, что построение графиков в процессе моделирования значительно замедляет работу модели.

## Table, Tabulate

Таблицы в GPSS используются для получения частотных распределений различных числовых величин (длины очередей, времена задержки и т.п.) по диапазонам.

Эти данные выводятся в файл результатов моделирования в табличной форме: указывается количество случаев за время моделирования, когда заданная величина принимает значения из диапазонов  $[B, B+C]$ ,  $[B+C, B+2C]$ ,  $[B+2C, B+3C]$  и т.д., где  $B$  и  $C$  - конкретные числа (границы и размеры диапазонов), задаваемые при объявлении таблицы.

**имя\_таблицы TABLE A,B,C,D**, где:

- имя\_таблицы – произвольное имя, обычно символьное;
- $A$  - имя СЧА, распределение которого по диапазонам требуется показать в таблице. Этот СЧА называется также аргументом таблицы или табулируемой величиной;
- $B$  - верхняя граница первого диапазона аргумента таблицы;
- $C$  - шаг таблицы (разность между верхней и нижней границей каждого диапазона);
- $D$  - количество диапазонов.

Регистрация значения СЧА, заданного в операнде  $A$  при объявлении таблицы, производится при прохождении транзакта через следующий оператор:

TABULATE  $A$ , где  $A$  - имя таблицы.

Оператор TABULATE указывается в тексте GPSS-модели в том месте, где требуется получить данные о значении табулируемой величины.

**Имеется возможность графического представления табулируемой величины в виде гистограммы.**

## Функции

Используя функции, пользователь может производить вычисления непрерывных или дискретных функциональных зависимостей между аргументом функции (независимая величина) и зависимым значением функции. Функции широко применяются, например, для задания случайного интервала времени между генерацией двух транзактов. Все функции в GPSS задаются табличным способом с помощью операторов описания функций.

Функция – стандартный числовой атрибут, название и численная зависимость, обозначаемые в виде FN\$name. Описывается пользователем в виде численной зависимости от другого СЧА. Два основных типа – дискретные и непрерывные.

## Блоки GPSS

### Управляющие блоки

Блок **GATE** - вспомогательный блок, проверяющий состояния устройств, памяти, логических ключей.

**GATE\_R A,B**, где Внутренний операнд  $R$  определяет проверяемое условие в виде логического атрибута и может принимать следующие значения:

- U - устройство занято;
- NU - устройство не занято;
- 1 - устройство прервано;
- NI - устройство не прервано;
- SF - память заполнена;
- SNF - память не заполнена;
- SE - память пустая;
- SNE - память не пустая;
- LR - ключ выключен;
- LS - ключ включен;
- M - транзакт находится в состоянии синхронизации;
- MN - транзакт не находится в состоянии синхронизации.

Если проверяемое условие для объекта, номер которого определяется полем A, выполняется, то транзакт входит в блок GATE. Если условие "ЛОЖЬ", то возможны два случая:

- 1) если поле B задано, то транзакт идет в блок, номер которого указан в поле B;
- 2) если в поле B пробел, то транзакт ждет в предыдущем блоке, пока не выполнится условие.

Блок **TEST** - сравнивает два стандартных числовых атрибута.

**TEST\_R A,B,C**, где

R - внутренний операнд, принимающий значения:

- E - равно;
- NE - не равно;
- L - меньше чем;
- LE - меньше чем или равно;
- G - больше чем;
- GE - больше чем или равно;
- A - стандартный числовой атрибут;
- B - стандартный числовой атрибут;
- C - номер альтернативного блока.

Значения стандартных числовых атрибутов, указанных в поле A и B, сравниваются отношением, определяемым операндом R. Если условие выполняется, транзакт вводится в блок. Если условие не выполняется и определено поле C, транзакт переходит в указанный блок, если же C не задано, транзакт задерживается в предыдущем блоке.

Блок **TRANSFER** используется для изменения движения транзактов.

**TRANSFER A,B,C,D**, где

- A - режим передачи (пробел, „ALL,BOTH,FN,P,PICK,SBR,SIM);

- В - следующий блок;
- С - следующий блок;
- D - значение индекса, используемое в режиме ALL.

Транзакт направляется в блок, определяемый в соответствии с режимом передачи, указанным в поле А. Режимы передачи поля А:

1. Пробел - транзакт передается в блок, определяемый полем В.
2. "." - статистический режим; в поле А указано десятичное число, выражающее вероятность перехода в блок С; его дополнение до единицы указывает вероятность перехода в блок В.
3. ALL - транзакт последовательно пытается перейти в блоки, определяемые значениями В, В+D, В+2D.....С.
4. BOTH - транзакт последовательно пытается войти в блок В, затем в блок С, до тех пор, пока один из них станет доступным.
5. FN - функциональный режим: поле В является номером функции; следующий блок определяется суммой значения этой функции поля С.
6. P - параметрический режим: поле В является номером параметра; следующий блок определяется суммой значения этого параметра и поля С.
7. PICK - выборочный режим: блок выбирается с равной вероятностью из блоков с номерами: В, В+1,..., С.
8. SBR - режим перехода к подпрограмме: номер текущего блока помещается в параметр, указанный в поле С, а транзакт передается в блок, номер которого указан в поле В.
9. SIM - одновременный режим: проверяется одновременное выполнение условий беспрепятственного движения транзактов в задерживающих блоках. Если условие выполняется, транзакт передается в следующий блок, в противном случае транзакт переходит на блок С.

**START** - управление процессом моделирования.

### **START A,B,C,D**

- А - счетчик числа завершений;
- В - подавление вывода на печать ( $B = NP$ );
- С - промежуточный вывод статистики; 0 - распечатка списков.

Поле А определяет счетчик завершений при прогоне модели. При входе транзакта в блок TERMINATE с ненулевым значением поля А, содержимое поля А блока TERMINATE вычитается из остаточного значения счетчика - поля А карты START.

Прогон завершается, когда значение счетчика достигнет нуля. Указатель NP в поле В отменяет вывод стандартного отчета GPSS. Величина, указанная в поле С, определяет число завершений, по достижении которого выдается промежуточный отчет. Таким образом можно получать статистические данные в определенных интервалах моделирования. Если поле D содержит 1, полученный отчет будет содержать списки текущих и будущих событий, а также списки пользователя.

**RESET** – очистка накопленной статистики. Достигнутое состояние объектов при этом сохраняется. Применение блока RESET позволяет уменьшить затраты машинного времени на сбор статистики о стационарном процессе в тех случаях, когда предшествующий ему переходный процесс вносит заметные искажения в накапливаемую статистику.

**CLEAR** – переводит всю модель — всю статистику и все объекты — в исходное состояние. Исключением является лишь датчик случайных чисел — он не возвращается к начальному значению. Применение блока CLEAR позволяет осуществить независимые реализации моделируемого случайного процесса.

## Блоки создания и уничтожения транзактов

Создание транзактов производится при помощи блока GENERATE

**GENERATE A, B, C, D, E**, где:

- A – среднее время между поступлениями транзактов. Поле может быть модифицировано за счет B
- B – разброс или модификатор среднего значения
- C – время появления первого транзакта
- D – число генерируемых транзактов
- E – начальный приоритет транзактов

Для моделирования случайных величин с заданными законами распределения в GPSS возможно использование библиотечных функций или задание требуемой функции распределения в табличном виде путем аппроксимации непрерывными функциями.

Встроенная библиотека GPSS содержит функции для моделирования случайных величин, имеющих следующие законы распределения:

- равномерное (Uniform);
- экспоненциальное (Exponential);
- гамма (Gamma);
- вейбулловское (Weibull);
- нормальное (Normal);
- логнормальное (LogNormal);
- биномиальное (Binomial);
- геометрическое (Geometric);
- дискретное равномерное (Discrete Uniform);
- пуассоновское (Poisson);
- логистическое (Logistic);
- логлапласово (LogLaplace);
- треугольное (Triangular).

Для моделирования случайной величины, имеющей равномерное распределение, используется библиотечная функция **UNIFORM(Stream,Min,Max)**

Stream - номер генератора случайных чисел (от 1 до 7); Min- наименьшее возможное значение; Max - наибольшее возможное значение.

Например:

GENERATE (UNIFORM(4,2,8)); генерация транзактов через интервалы времени, равномерно распределенные на отрезке [2; 8], с использованием 4-го генератора случайных чисел.

Блок **TERMINATE** используется для удаления из системы транзактов.

**TERMINATE A**, где A - число единиц, на которое блок изменяет счетчик завершения в момент поступления транзакта.

## Блоки по работе с обслуживающими устройствами

В GPSS элементами, которые требуют обслуживания, являются транзакты. Они перемещаются в модели от блока к блоку. Если в какой-то момент активности транзакт занимает ОКУ, то для этого он входит (или пытается войти) в соответствующий блок, описывающий это ОКУ. Блок должен обладать следующими свойствами:

- если ОКУ уже используют, транзакт не может войти в блок и должен ждать в очереди;
- если ОКУ не используют, транзакт может войти в блок, статус ОКУ изменяется на "занято".

Блок, обладающий этими свойствами, является блоком SEIZE. Вход транзакта в блок SEIZE моделирует занятие ОКУ. После обслуживания вход того же транзакта в другой блок моделирует освобождение ОКУ. Назначением этого блока является изменение состояния ранее занятого ОКУ с "занято" в "незанято". Этим блоком является блок RELEASE (освободить).

Форматы блоков:

- **SEIZE A**
- **RELEASE A**

## Блоки по синхронизации транзактов

**MATCH** - синхронизирует два транзакта одного семейства.

Формат: **MATCH A**

A - номер сопряженного блока MATCH.

Первый транзакт, достигнув блока MATCH, задерживается в нем до тех пор, пока другой транзакт семейства достигнет сопряженного блока MATCH, указанного в поле A. Во время задержки устанавливается индикатор синхронизации. Он сбрасывается, тогда транзакт того же семейства входит в соответствующий блок MATCH.

Пример:

AA MATCH BB

...

BB MATCH AA

**ASSEMBLE** - объединение транзактов, принадлежащих одному семейству (или ансамблю).

Формат: **ASSEMBLE A**

A - число объединяемых транзактов.

Первый транзакт семейства, достигнув блока ASSEMBLE, задерживается в нем до тех пор, пока остальные члены семейства не поступят в этот блок. Когда транзакты, число которых указано в поле A, поступят в этот блок, они **будут удалены из модели**, а первый прибывший транзакт **продолжит движение**.

**GATHER** - накапливает транзакты, являющиеся членами семейства.

Формат: **GATHER A**

A - счетчик транзактов, которые должны быть накоплены.

Транзакты одного семейства задерживаются в блоке GATHER до тех пор, пока их число не станет равным значению поля A. Когда последний транзакт войдет в блок GATHER, все **они одновременно выходят из него в том порядке, в котором поступили**. Состояние блока GATHER может быть проверено блоком GATE.

**SPLIT** - создает копии текущего транзакта.

Формат: **SPLIT A,B,C,D**

- A - число создаваемых копий;
- B - следующий блок для копий;
- C - параметр для хранения порядкового номера копии;
- D - число параметров у каждой копии.

Поле A определяет число копий, которые образуются при входе текущего транзакта. Эти вновь созданные транзакты по умолчанию идентичны исходному транзакту. Копии входят в блок, указанный в поле B. И сходный транзакт поступает на следующий блок. Параметр поля C используется для задания порядковых номеров копий. Нумерация осуществляется следующим образом. Порядковый номер исходного транзакта увеличивается первым. Если он был равен нулю, при входе транзакта в блок он станет равным единице. Порядковый номер первой копии станет на единицу больше, чем у исходного транзакта. Номера последующих копий также увеличиваются на единицу. Если

поле D не задано, копии имеют такое же, как у исходного транзакта число и тип параметров.

## Блоки PREEMPT и RETURN

Для моделирования работы одноканальных устройств с прерываниями, т. е. с приостановкой обслуживания транзакта, ранее занявшего одноканальное устройство, и захвата устройства прерывающим транзактом используется блок

### PREEMPT A,B,[C],D,E

- A - имя устройства, работа которого прерывается входящим в блок транзактом;
- B - режим прерывания (по умолчанию - обычный, PR - прерывание по приоритету);
- C - метка блока, в который направляется транзакт, обслуживание которого было прервано. Прерванный транзакт покидает устройство, но претендует на право его использования (если не задан операнд E);
- D - номер параметра прерванного транзакта, в который заносится остаток времени обслуживания;
- E - если в поле записывается «RE», то транзакт, обслуживание которого было прервано, не претендует на завершение своего обслуживания в устройстве.

Прерванный транзакт (если отсутствует адрес в поле C) ожидает окончания прерывания устройства и возвращения на устройство для обслуживания. Если таковых транзактов несколько, то они возвращаются на устройство в порядке прерываний (либо приоритетов).

Транзакт, вошедший в блок RETURN, снимает прерывание на устройстве, вызванное вхождением данного транзакта в блок PREEMPT:

**RETURN A**, где A - имя устройства, с которого снимается прерывание.

Прерывание может быть снято только тем транзактом, которым оно было сгенерировано.

Пары блоков SEIZE-RELEASE и PREEMPT-RETURN могут работать с одними и теми же устройствами. В зависимости от логики работы модели пользователь сам должен определить потребность работы устройств с прерываниями.

## Блоки, изменяющие параметры транзактов

**ASSIGN** - изменяет значение параметра транзакта.

Формат: ASSIGN A, B, C

- A - номер изменяемого параметра (+, -);
- B - новое значение параметра.
- C - задает номер модификатора-функции. При использовании C значение операнда B умножается на значение модификатора-функции.



Если за полем А следует знак + или -, то значение поля В соответственно добавляется или вычитается из А. Если знаки - или + не указаны, то значение поля В становится текущим значением параметра.

**MARK** - ставит отметку времени или записывает значение таймера.

Формат: **MARK A**, где А - номер параметра, в который записывается значение таймера абсолютного времени.

Если поле А не используется, отметка времени (время создания транзакта) заменяется значением текущего таймера. Если поле А определено, то текущее значение таймера записывается в параметр, указанный в поле А.

**INDEX A,B** если поле А равно 1, то блок модифицирует значение первого параметра транзакта, вошедшего в данный блок, на величину, задаваемую полем В, если же в поле А записано некоторое другое число, то это число воспринимается как номер параметра транзакта и блок присваивает первому параметру транзакта сумму значения поля В и значения параметра, указанного полем А.

Очереди в GPSS

**QUEUE A,B**

А – номер очереди

В – число добавляемых единиц

Осуществляет сбор статистики об очереди. При записи нового транзакта в очередь определяется длина интервала времени, в течение которого длина очереди оставалась неизменной.

При входе транзакта в данный блок текущая длина очереди увеличивается на число единиц, указанное в поле В. Затем происходит сравнение с максимальной длиной очереди, достигнутой до этого момента времени. Если оно больше старого значения, то оно его заменяет. Кроме того, счетчик общего числа единиц прошедших через очередь увеличивается на тоже число единиц.

**ADVANCE A,B - добавляет элемент в очередь**

Блок задает среднее время выполнения операций в моделируемой системе, а также разброс времени относительно среднего. Задержка – целое число.

Для задания времени пребывания в блоке ADVANCE пользователь указывает среднее время в поле А, а модификатор в поле В. Если поле задержки постоянно, то поле В может быть пустым. А если нулевое, то и поле А может отсутствовать.

**DEPART** - удаляет транзакт из очереди.

Формат: **DEPART A,B**

- А - номер (имя) очереди;

- В - число удаляемых из очереди элементов.

Удаляет текущий транзакт из очереди, указанной в поле А, и уменьшает содержимое очереди на значение поля В. Транзакт может находиться одновременно в двух различных очередях.