

# FASTMEAN 5.0

1 апреля 2010

<b>1. СОВМЕСТИМОСТЬ</b> .....	2
<b>2. ВЕРСИЯ И ВРЕМЯ СБОРКИ</b> .....	2
<b>3. ЭЛЕМЕНТ «ПОРТ»</b> .....	2
<b>4. ИМЕНА ЭЛЕМЕНТОВ</b> .....	3
<b>5. РЕДАКТОР СХЕМЫ</b> .....	4
<b>6. НОВАЯ МОДЕЛЬ MOSFET КЛЮЧА</b> .....	4
<b>7. АВТОМАТИЗАЦИЯ РАСЧЕТОВ</b> .....	7
7.1. ДИРЕКТИВА .WRITE.....	7
7.2. ДИРЕКТИВА .WRITE AFTER .....	7
7.3. ЗАПУСК РАСЧЕТА ИЗ КОМАНДНОЙ СТРОКИ.....	8

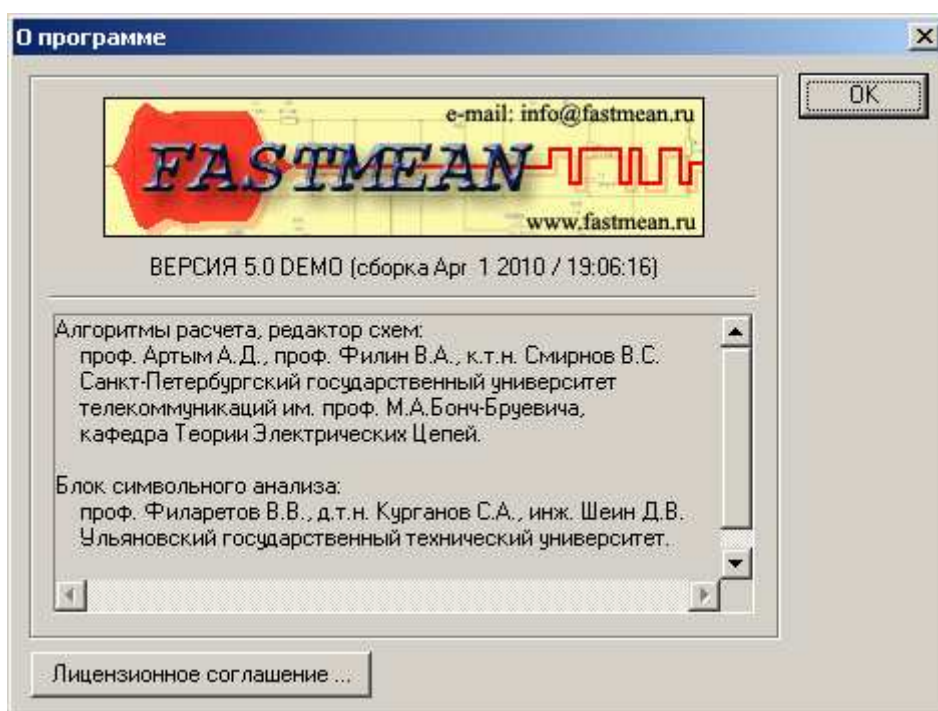
## 1. Совместимость

Версия 5.0 открывает все схемы предыдущих версий FASTMEAN.

Предыдущие версии FASTMEAN не могут открывать схемы, сохраненные версией 5.0.

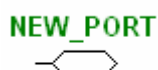
## 2. Версия и время сборки

В окне About показывается номер версии и время сборки, чтобы можно было отличать варианты программы, собранные в разное время.



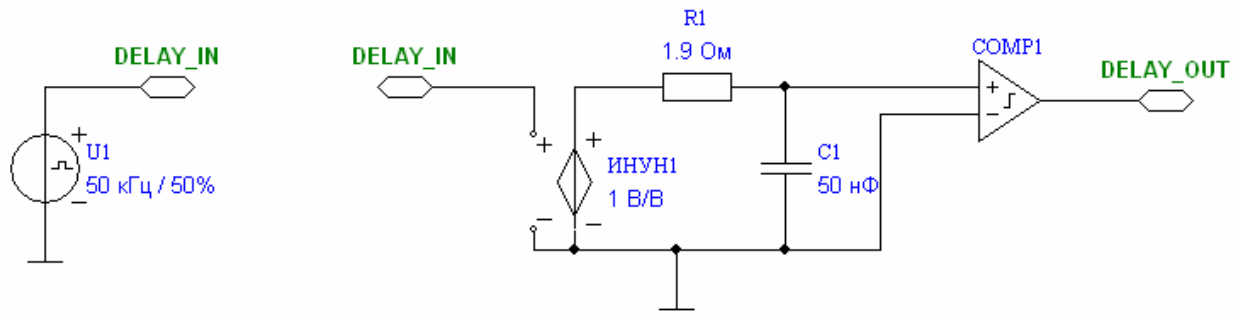
## 3. Элемент «Порт»

Новый элемент «Порт» позволяет соединять отдельные части схемы без протягивания проводов. Для этого различным элементам «Порт» нужно присвоить одинаковое имя.



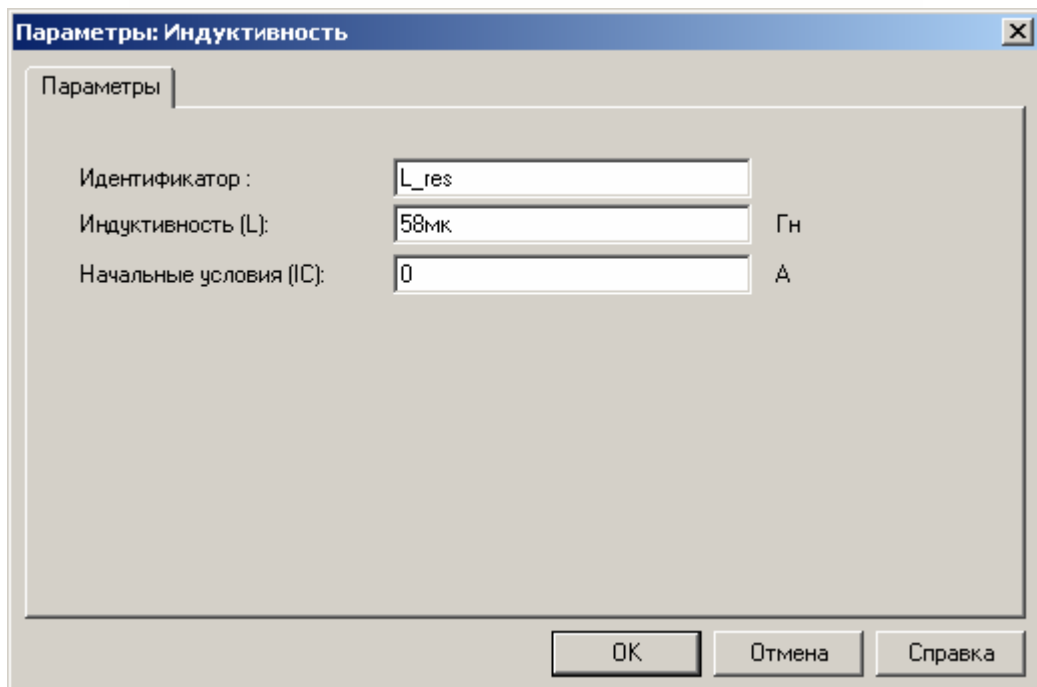
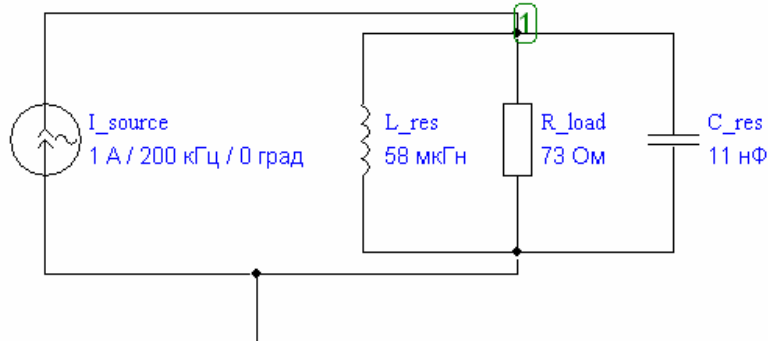
Этот элемент также удобно использовать для именованя узлов и контрольных точек в схеме. В отличие от номеров узлов, которые могут меняться в FASTMEAN, имя порта не меняется и придает осмысленность выражениям в расчетах. Например: U(CLOCK\_OUT).

Пример:



#### 4. Имена элементов

Имена элементов теперь можно менять. Имена элементов теперь не меняются FASTMEAN при удалении других элементов. Это придает большую осмысленность схеме и удобство в работе.



Конфликты имен устраняются автоматически. Если Вы ввели новое имя, а изменения не были приняты или было присвоено другое имя, значит, был устранен конфликт с существующим именем в схеме.

Для символьного анализа важно сохранять неизменной начальную часть имени (R, L, ИНУН, U и т.п.) – по ней определяется тип элемента.

## 5. Редактор схемы

Новое поведение клавиатуры и мыши:

R – повернуть элемент

F – отобразить элемент слева направо

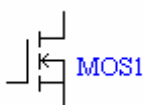
V – отобразить элемент сверху вниз

Ctrl + перетаскивание элементов мышью – копирует элементы

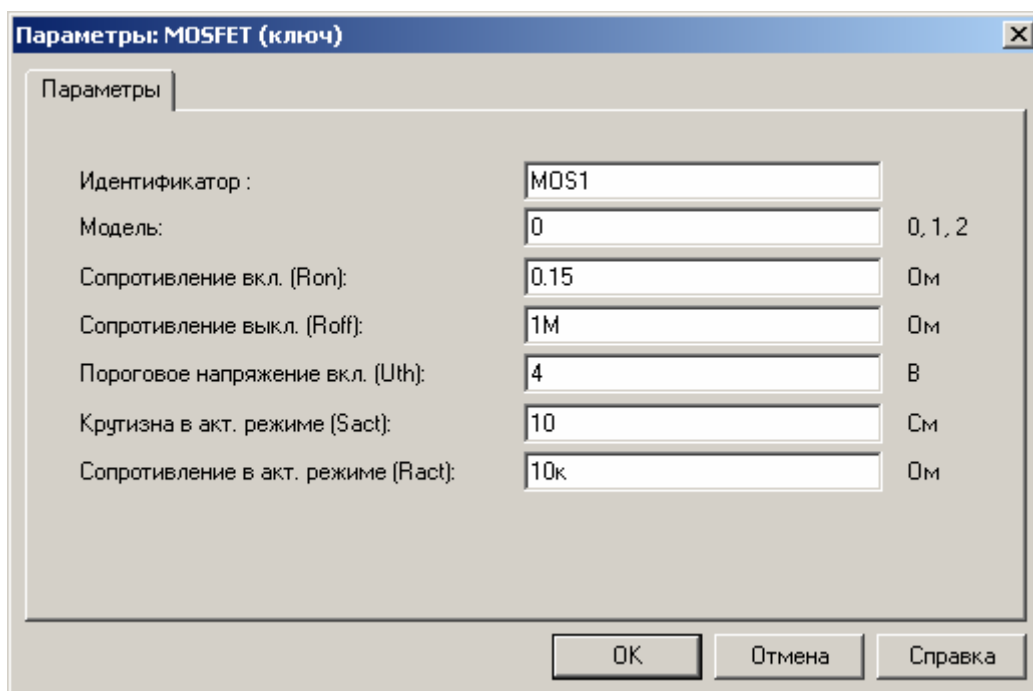
Ctrl + колесо мыши – изменение масштаба (zoom)

Shift + колесо мыши – скроллинг листа в горизонтальном направлении

## 6. Новая модель MOSFET ключа



Изменена старая и добавлены 2 новые модели MOSFET ключа. Теперь у MOSFET 3 модели: 0 (старая), 1 и 2. Модель задается в окне параметров.



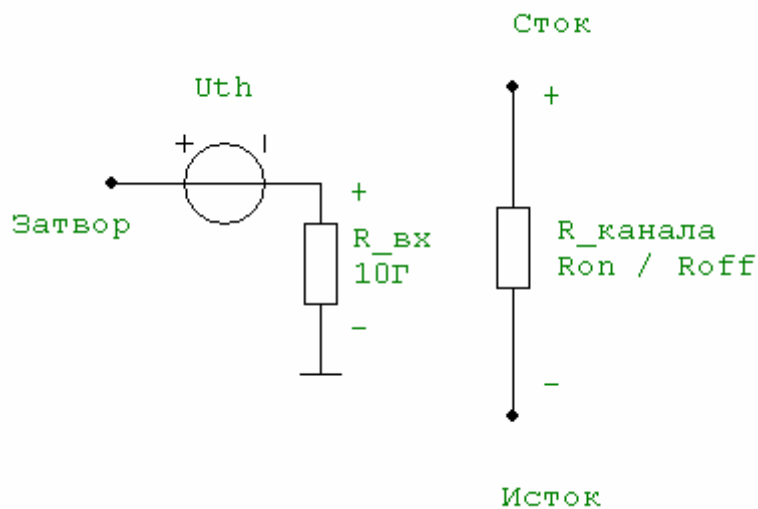
Параметр	Значение	Единица
Идентификатор :	MOS1	
Модель:	0	0, 1, 2
Сопротивление вкл. (Ron):	0.15	Ом
Сопротивление выкл. (Roff):	1M	Ом
Пороговое напряжение вкл. (Uth):	4	В
Крутизна в акт. режиме (Sact):	10	См
Сопротивление в акт. режиме (Ract):	10к	Ом

У старой модели убран параметр «падение напряжения во включенном состоянии», как неприменимый к MOSFET.

Модель MOSFET по-прежнему не содержит встроенный диод. Его необходимо добавлять самостоятельно.

Для модели 2 добавлены параметры: крутизна в активном режиме Sact и сопротивление (выходное) в активном режиме Ract. Для моделей 0 и 1 эти два параметра не используются.

## Модель 0

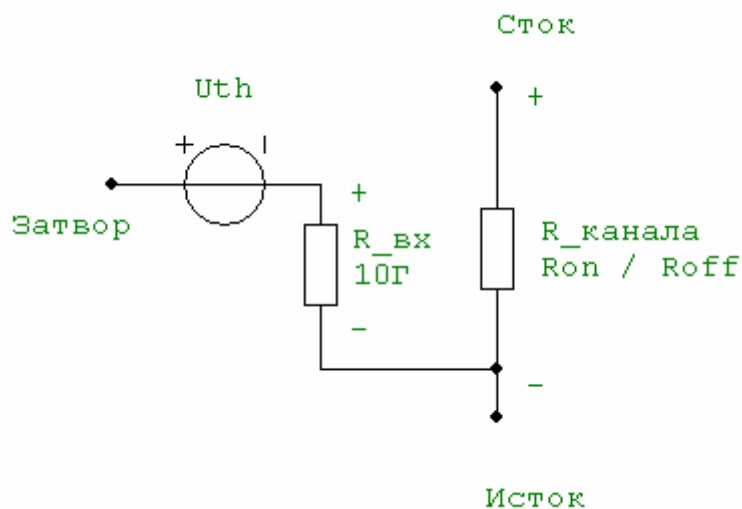


Транзистор управляется напряжением на затворе относительно земли. Сделано для упрощения как в PowerSim, чтобы не моделировать драйверные цепи.

Логика:

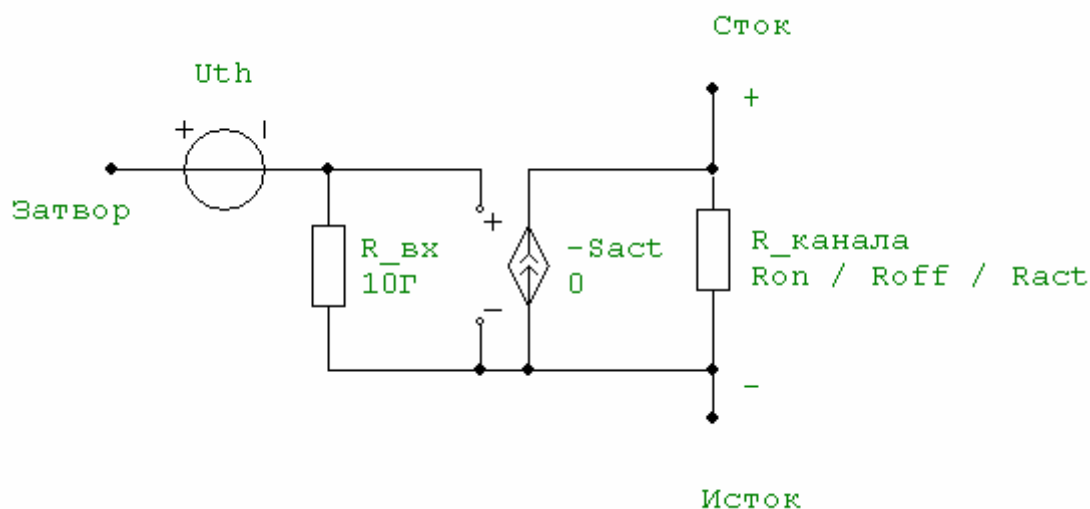
$R_{on}$ , если  $U(R_{вх}) \geq 0$  И  $U(R_{канала}) \geq 0$   
 $R_{off}$ , если  $U(R_{вх}) < 0$  ИЛИ  $U(R_{канала}) < 0$

## Модель 1



Аналогична модели 0, но управляющее напряжение на затворе относительно истока.

## Модель 2



Модель 2 поддерживает активный режим работы транзистора, в котором MOSFET работает как источник тока, управляемый напряжением затвор-исток с крутизной  $S_{act}$ , с некоторым большим выходным сопротивлением  $R_{act}$ . Вне активного режима источник тока не работает.

Логика:

УСЛОВИЕ 1  $U(R_{vx}) \geq 0$

УСЛОВИЕ 2  $U(R_{канала}) \geq 0$

УСЛОВИЕ 3  $U(R_{vx}) * S_{act} + U(R_{канала}) / R_{act} - U(R_{канала}) / R_{on} \geq 0$

$R_{on}, 0,$  если УСЛОВИЕ 1 И УСЛОВИЕ 2 И УСЛОВИЕ 3

$R_{off}, 0,$  если НЕ(УСЛОВИЕ1) ИЛИ НЕ(УСЛОВИЕ2)

$R_{act}, S_{act},$  если УСЛОВИЕ 1 И УСЛОВИЕ 2 И НЕ(УСЛОВИЕ 3)

Пример использования модели 2 для транзистора SPW17N80C3 находится в папке Схемы\New MOSFET\. Модель была составлена по данным спецификации производителя и учитывает активный режим работы транзистора и нелинейные паразитные емкости.

## 7. Автоматизация расчетов

### 7.1. Директива `.write`

Директива `.write` позволяет при выполнении расчета переходных процессов записывать сигналы в файл.

Директива `.write` размещается в текстовом блоке в произвольном месте схемы (аналогично `.define`). Директивы могут быть размещены в разных блоках «Текст». Директивы в одном блоке отделяются символом «;». Директиву можно писать на нескольких строчках. Символ «%» означает комментарий до конца текущей строки.

`.write имя_файла_1 выражение_1 выражение_2 ... выражение_N;`

выражение\_i – обычное выражение FASTMEAN при расчете переходных процессов.

Если имя файла или выражение содержит пробелы, заключите его в кавычки. Задача пользователя избегать одинаковых имен файлов в разных директивах.

Примеры:

```
.write      "main signals.txt"  
           t                % время  
           U(R_load)  
           "1 - U(CLOCK_OUT)"; % not CLOCK_OUT
```

`.write pwm_signals.txt t U(MOS_PWM) I(MOS_PWM) U(MOS_GATE);`

Формат файла: текстовый, таблица чисел с плавающей точкой, каждая строка – одна точка расчета переходного процесса.

### 7.2. Директива `.write after`

Директива `.write after` позволяет записывать сигналы в файл после того, как время достигнет заданного момента. Это удобно когда нужно отбросить начальный переходный процесс.

Директива `.write after` является общей для всех директив `.write`. В схеме должна быть только одна директива `.write after`.

`.write after выражение`

выражение – момент времени, после которого нужно записывать сигналы

Пример:

`.write after tmax-10u % записывать в файл последние 10мкс процесса`

### 7.3. Запуск расчета из командной строки

Для автоматизации расчетов FASTMEAN запускается из командной строки с указанием имени файла скрипта:

```
Fastmean.exe -o файл_схемы.scm -s файл_скрипта
```

Файл скрипта может содержать:

**<элемент>.<параметр> = <значение>**

изменение параметров элементов

**sim [end\_time=<выражение>]**

выполнить расчет переходного процесса до момента времени end\_time  
если end\_time не указан, используются настройки схемы

**save**

сохранить схему

**exit**

выйти из программы

Команды отделяются между собой символом «;». Символ «%» означает комментарий до конца текущей строки.

Пример: файл run.txt

```
U_in.U0 = 540;  
R_load.R = 73;  
U_K_pwm.U0 = 0.04;  
save;  
sim;  
exit;
```