

Проектирование аналоговых и аналого-цифровых интегральных схем с использованием языка Verilog-AMS

Часть 2. Аналоговые события

Дмитрий Осипов (Москва)

Во второй статье цикла рассматривается обработка аналоговых событий в Verilog-A модулях. Рассмотрены примеры использования аналоговых событий для описания компараторов и самотактируемых схем.

Аналоговые события

Событие – это частичное изменение состояния схемы. Например, событие может быть связано с перепадом входного или внутреннего сигнала. Язык Verilog-A позволяет описывать модули, детектирующие или регистрирующие какие-либо события (например, перепад напряжения) и реагирующие на них. Таким образом, языковая конструкция Verilog-A может быть разбита на две части: первая – это описание события, а вторая – описание реакции схемы на это событие. Общий вид описания аналогового события и его обработки таков:

```
@ ( описание_события ) обработка_
события ;
```

Описание аналогового события отличается от описания события в языке Verilog тем, что аналоговые события не являются блокирующими. Это означает, что модуль будет выполняться даже в том случае, если событие не происходит. Просто та часть, где описана

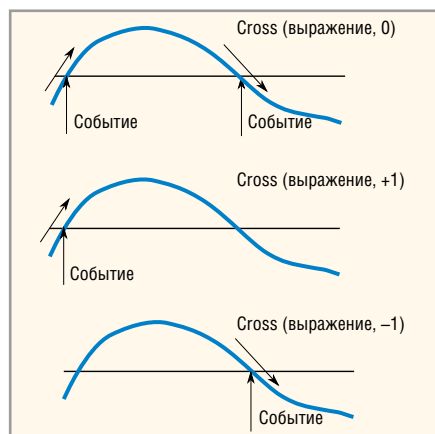


Рис. 1. Поведение функции Cross в зависимости от значения аргумента <направление>

реакция на событие, будет пропускаться. В случае если событие произошло, то тогда его обработка и будет выполнена исключительно в момент наступления события. Поэтому описание реакции схемы на событие не должно содержать оператор «<+>», то есть изменение значения сигнала невозможно внутри обработчика события, что естественно, так как в аналоговом моделировании сигналы должны быть описаны в каждой точке моделирования.

Можно описывать реакцию схемы не только на одно событие, но и на группу событий, используя оператор or (или):

```
@ ( описание_события1 or
описание_события2 ) обработка_
события ;
```

Язык Verilog-A содержит несколько «детекторов» событий, различающихся по классу обрабатываемых событий. К основным типам событий относятся, прежде всего, регистрируемые детекторами пересечения порога, поведение которых схоже с конструкциями Posedge и Negedge в языке Verilog. То есть события, наступающие при определённом изменении входного сигнала. Второй тип событий в Verilog-A – это события по таймеру, то есть наступающие в определённое время моделирования. Такие события обычно используются при написании тестовых окружений на Verilog-A.

Рассмотрим эти типы событий подробнее.

ДЕТЕКТОРЫ ПЕРЕСЕЧЕНИЯ ПОРОГА

Для детектирования событий, связанных с пересечением порога, Verilog-A предлагает две функции Cross (пересечение) и Above (превышение). Функция

Above во многом аналогична функции Cross и в настоящее время не рекомендуется к использованию, поскольку в новых версиях аналоговых симуляторов она может и не поддерживаться.

Функция Cross используется для описания аналогового события, которое происходит при пересечении аргументом функции нуля в заданном направлении. Кроме того, Cross позволяет контролировать временной шаг для достижения необходимой точности пересечения. Ниже представлен общий вид функции Cross:

```
cross (arg, <направление>,
<временной_допуск>, <допуск_по_
значению_выражения> )
```

Параметры <направление>, <временной_допуск>, <допуск_по_значению_выражения> являются опциональными. Все аргументы, за исключением направления, являются вещественными числами или выражениями. Направление же задаётся целым числом.

Если параметр <направление> устанавливается равным 0 или не устанавливается вовсе, то событие наступает, когда значение выражения пересекает ноль в любом направлении. Если значение направления равно +1, то событие считается наступившим только при условии, что значение выражения пересекает ноль во время возрастания. При установке направления –1, событие происходит, если значение выражения пересекает ноль во время спада (см. рис. 1).

Для контроля допусков по точности детектирования пересечения служат аргументы <временной_допуск> и <допуск_по_значению_выражения>. Если задан <допуск_по_значению_выражения>, то событие произойдёт тогда, когда модуль значения выражения станет меньше, чем аргумент <допуск_по_значению_выражения>. Если задан <временной_допуск>, то событие произойдёт внутри задан-

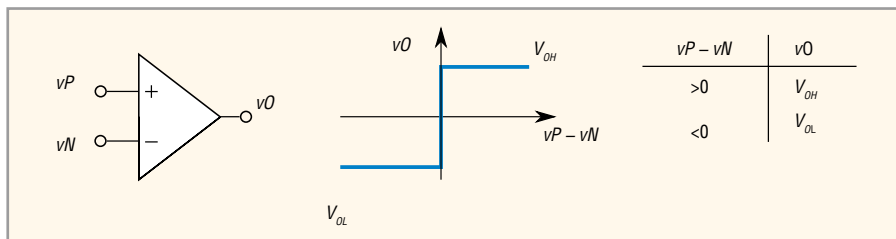


Рис. 2. Принцип работы компаратора напряжений

ного интервала, определяемого относительно действительного момента пересечения выражением нуля.

В качестве примера использования функции Cross рассмотрим Verilog-A описание компаратора. Компаратор – это схема, производящая сравнение одного аналогового сигнала с другим или с опорным сигналом и вырабатывающая цифровой выход, основанный на результате сравнения (см. рис. 2).

Модель идеализированного компаратора представлена на рисунке 3, где f_1 определяется выражением:

$$f_1(v_P - v_N) = \begin{cases} V_{OH}, & (v_P - v_N) > 0 \\ V_{OL}, & (v_P - v_N) < 0. \end{cases}$$

Описание компаратора языком Verilog-A:

```

analog begin
    @(cross(V(vP) - V(vN), 0))
    f1 = ((V(vP) > V(vN)) ?
VOH : VOL);
    V(vO) <+ f1;
end
    
```

В показанном примере аналоговым событием является пересечение выражением $(V(vP) - V(vN))$ нуля. При наступлении события устанавливается новое значение вещественной переменной f_1 . Обратите внимание на то, что когда событие не происходит, то весь блок кода, относящийся к обработке события, не выполняется и, следовательно, сигналу выходного напряжения $V(vO)$ присваивается «старое» значение веще-

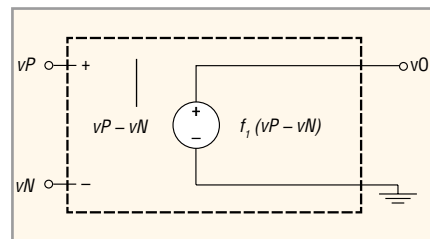


Рис. 3. Модель идеального компаратора

ственной переменной f_1 . Таким образом, в описываемой модели значение выходного напряжения в начальный период работы (до наступления первого события), в общем случае не определено, так как не определено значение f_1 . Для решения этой проблемы можно задать значение переменной f_1 при наступлении особого события Initial_Step, аналогичного блоку Initial в Verilog. Данное событие наступает один раз, в начальный момент времени моделирования:

```

@ ( initial_step ) begin
    f1 = ((V(vP) >
V(vN)) ? VOH : VOL);
end
    
```

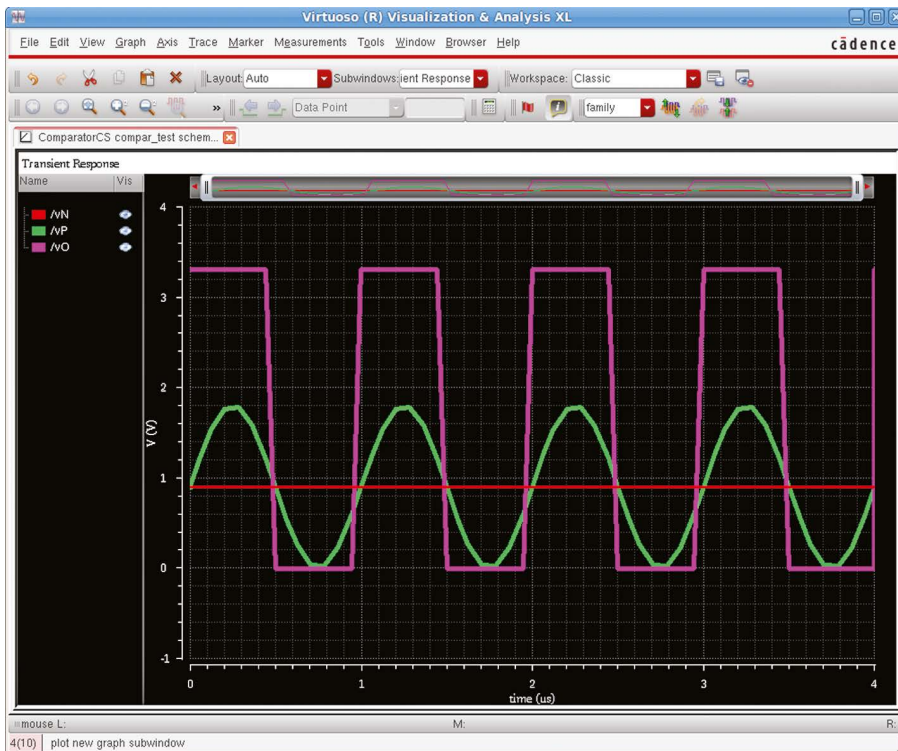


Рис. 4. Входной и выходной сигналы Verilog-A модуля модели компаратора



Рис. 5. Окно настроек модели компаратора в среде Cadence Virtuoso

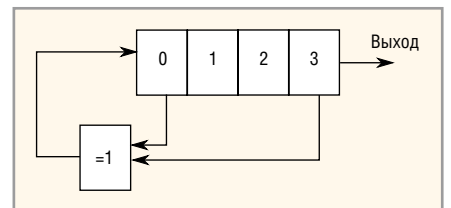


Рис. 6. Схема генератора псевдослучайных чисел

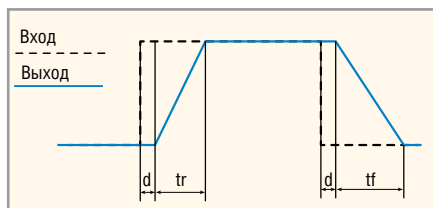


Рис. 7. Иллюстрация работы оператора Transition

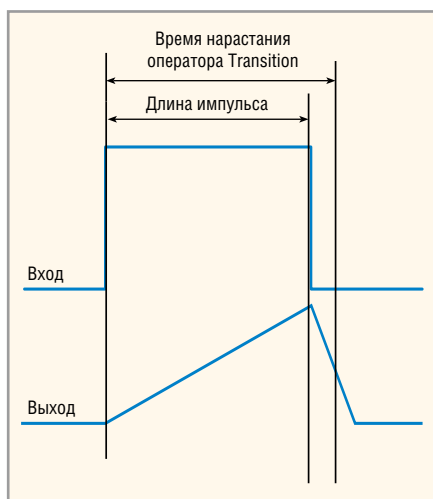


Рис. 8. Результат применения оператора Transition к импульсу с длиной меньше, чем параметр tr оператора

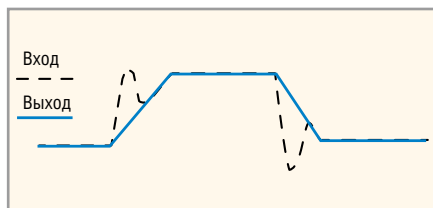


Рис. 9. Иллюстрация работы оператора Slew

Verilog-A описание самоактивируемого генератора псевдослучайных чисел

```

module prng(out);
output out;
electrical out;
parameter integer bit0=0;
parameter integer bit1=0;
parameter integer bit2=0;
parameter integer bit3=0;
parameter real period=100n;
integer feedback_signal;
integer sh_reg[0:3];
integer i;
analog begin
    @(timer(0)) begin
        sh_reg[0]=bit0;
        sh_reg[1]=bit1;
        sh_reg[2]=bit2;
        sh_reg[3]=bit3;
    end
    @(timer(10n, period)) begin
        feedback_
        signal=sh_reg[0]^sh_reg[3];
        for
        (i=3;i>0;i=i-1) begin
            sh_reg[i]=sh_reg[i-1];
        end
        sh_
        reg[0]=feedback_signal;
    end
    V(out)<+transition(sh_reg[3],1n,1n,1n);
end
endmodule
    
```

На рисунке 4 показана реакция нашей разработанной языком Verilog-A модели на синусоидальный входной сигнал в Cadence Virtuoso. Для задания параметров модуля (уровней напряжения логической «1» и логического «0») используется ключевое слово Parameter, рассмотренное в первой части статьи (см. СЭ № 7, 2014). На рисунке 5 показано окно настроек параметров модуля в Cadence Virtuoso. Таким образом, для конечного пользователя разработанная Verilog-A модель ничем не отличается от модели, встроенной в симулятор.

Аналоговые события по таймеру

Оператор Timer (старт, период) используется для генерирования аналоговых событий в определённые моменты времени. Если задан только аргумент «старт», то аналоговое событие детектируется один раз, когда время симуляции достигает значения «старт». Если же задан период, то событие детектируется во все моменты времени (старт + n × период), где n = 0, 1, 2, ... Данный оператор позволяет описывать самоактивируемые схемы или сигналы. В качестве примера можно рассмотреть схему генератора псевдослучайных чисел на основе сдвигового регистра с обратной связью, представленную на рисунке 6.

Описание регистра языком Verilog-A приведено в листинге, где в качестве параметров рассматриваются период

и инициализирующие значения битов регистра.

Конструкция `@(timer(0))` полностью аналогична конструкции `@(initial_step)`, рассмотренной в разделе «Детекторы пересечения порога», и используется для инициализации внутренних переменных Verilog-A модуля.

Внутри аналогового блока происходит присваивание значения `sh_reg [3]` напряжению на выходе модуля. При этом для сглаживания дискретных значений переменной `Result` (вычисляется только по положительному перепаду тактового сигнала) применяется оператор `Transition`. Оператор `Transition` (операнд, `<d>`, `<tr>`, `<tf>`), называемый также «аналоговый фильтр», производит преобразование кусочно-постоянного сигнала, указанного в операнде, в сигнал с переходным процессом между постоянными уровнями. Задержка начала переходного процесса определяется параметром `<d>`, время нарастания – `<tr>`, время спада – `<tf>`. Если указано только время нарастания, то время спада будет считаться равным времени нарастания (см. рис. 7).

Так как для переходного процесса необходимо некоторое время, возможна ситуация, когда новый переходный процесс начинается ещё до завершения предыдущего. В этом случае оператор прерывает предыдущий переходный процесс и начинает обработку следующего, но без ущерба непрерывности выходного сигнала (см. рис. 8).

Для DC-анализа, то есть анализа по постоянному току, выход оператора равен значению операнда. Для AC-анализа (малосигнального анализа) сигнал, прошедший через данный оператор, не изменяется.

Кроме аналогового фильтра `Transition`, язык Verilog-A предоставляет

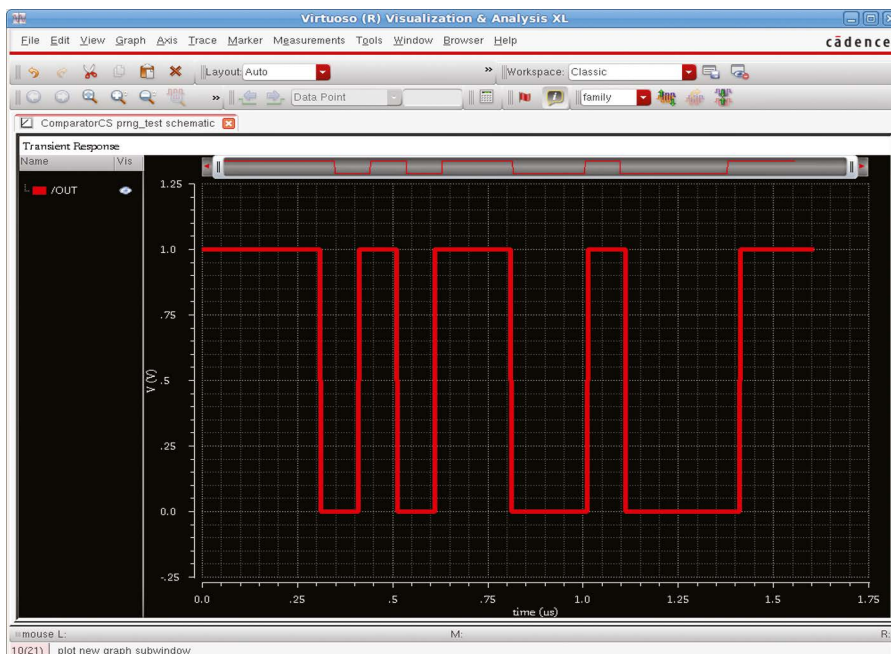


Рис. 10. Выходной сигнал генератора псевдослучайных чисел в среде Cadence Virtuoso

ещё один оператор для сглаживания выбросов во входном сигнале – оператор `Slew` (операнд, `<mPS>`, `<mNS>`). Этот оператор пропускает входной сигнал без изменений, но при этом ограничивает максимальный наклон выходного сигнала (см. рис. 9). Максимальные положительный и отрицательный наклоны задаются аргументами `mPS` и `mNS` соответственно. Для DC-анализа выход оператора равен значению операнда. Для AC-анализа (малосигнального анализа) оператор `Slew` будет представлять собой нулевое усиление, если рабочая точка находилась на участке сигнала, где сработало ограничение по максимальному наклону. В противном случае усиление будет единичное.

Кроме того, если необходимо только задержать входной сигнал, можно применить оператор `Absdelay` (операнд, `<d>`, `<maxD>`), который возвращает сигнал

равный сигналу, указанному как операнд оператора, сдвинутый по времени на `<d>`. Значение `<d>` должно быть положительным. Если задана максимальная задержка `<maxD>`, то задержка `<d>` может изменяться. При этом она не должна превышать значения `<maxD>`. Для DC-анализа выход оператора равен значению операнда. Для AC-анализа передаточная функция оператора `absdelay` равна $e^{j2\pi fT}$, где T – значение `<d>`, f – частота анализа, что отличает данный оператор от оператора `Transition`.

Выходной сигнал модуля для инициализирующего значения сдвигового регистра 1111 показан на рисунке 10. В данном модуле дважды используются аналоговые события по таймеру: сначала для инициализации начального значения сдвигового регистра, а затем для периодического сдвига значения на регистре.

