

# Разработка моделей цифровых элементов Digital SimCode для Altium Mixed Sim

Юрий Леган (yuri.legan@altium.com)

В статье описываются этапы разработки цифровых элементов Digital SimCode для устройства Altium Mixed Sim. Digital SimCode схож по функциональности с другими языками программирования и описывает аппаратуру, основываясь на принципе событийного моделирования.

## Введение

Современные САПР электронных устройств, в частности Altium Designer, обладают колоссальной функциональностью, когда речь заходит об управлении данными проектов, данными проектирования схем и печатных плат, а также инженерного анализа.

Инженерный анализ в Altium Designer выполняется различными расчётными модулями, применяемыми в зависимости от инженерной задачи. С помощью программы можно выполнять пред- и посттопологический анализ целостности сигналов, расчёты электрических режимов функционирования элементов схемы и посттопологических статических электрических нагрузок на цепи питания.

Altium Designer оснащён высококачественным имитатором электрических схем Altium Mixed Sim. Данный имитатор позволяет выполнять различные виды расчётов электрических режимов схемы на этапе проектирования. В том числе он имеет алгоритмы моделирования смешанных сигналов, реализующих расчёт поведения электрической схемы, содержащей как «аналоговые», так и «цифровые» компоненты. Расчёт поведения электрической схемы выполняется как расчёт переходного процесса во временной области.

Данный подход давно известен и реализован во многих известных имитаторах электрических схем. Естественным затруднением в этом случае становится описание моделей цифровых компонентов. Такие компоненты характеризуются обычно большим количеством внутренних элементов, функционирующих в ключевом (или иначе – вентиляном) режиме и реализующих такие внутренние состояния своих схем, которые обычно приближены по напряжениям к потенциалам питания. Моделировать схемы на уровне транзисторов – чрезвы-

чайно сложно и затратно с точки зрения ресурсов вычислительной системы. Поэтому все цифровые компоненты моделируются на логическом уровне, интерпретируя значения внутренних функций цифровых компонентов в логической форме, имеющей несколько допустимых статических состояний (0, 1, X и Z) и известные динамические взаимные переходы между статическими состояниями. Для решения задачи моделирования цифровых компонентов в имитаторе Altium Mixed Sim реализована поддержка специфических моделей имитатора XSPICE, описывающего цифровые компоненты.

Изначально XSPICE обладал открытой программной спецификацией и допускал «достраивание» функциональности за счёт разработки дополнительных моделей компонентов в виде программных модулей на языке C++, обладающих определённой совместимостью по программному интерфейсу с ядром имитатора. Этот подход давал широкие возможности для построения моделей, но требовал от инженера нетривиальной специализации дополнительно как программиста. Кроме того, каждое расширение функциональных возможностей имитатора требовало новой сборки программных библиотек и зависело от применяемого компилятора. Используя разные компиляторы, разработчики расширений ядра имитатора могли получать различные результаты даже при одном исходном коде.

Чтобы снизить требования к навыкам программирования, в ядро имитатора Altium Mixed Sim введена универсальная модель цифрового компонента. Данная модель реализует определяемую пользователем функциональность через восприятие плоского текстового файла, содержащего описание цифрового элемента на языке Digital SimCode.

С таким подходом нет необходимости в расширении имитатора дополнительными пользовательскими программами, а значит, можно снизить уровень требуемых навыков для разработки ПО. Пользователю, разрабатывающему модель цифрового компонента, теперь не нужны специальность программиста, знание C++, достаточно овладеть гораздо более простым языком Digital SimCode.

## Назначение Digital SimCode

Digital SimCode является языком описания аппаратуры (HDL), схожим по основной функциональности с другими языками программирования. SimCode предназначен для описания функционирования цифровых элементов на логическом уровне.

Работа модели Digital SimCode основана на принципе событийного моделирования: имитатор отслеживает входящие сигналы, проверяет на изменение эквивалентного логического состояния и отслеживает внутренние состояния модели. При наличии события изменения состояния входящего сигнала или внутреннего состояния модели формируется обработка события. Изменённые выходные сигналы отражаются во внешних цепях моделируемого цифрового компонента. Сигналы модели, обращённые к внешней схеме, называют выводами или портами модели.

## Основные возможности

Digital SimCode позволяет:

- описывать перечни входных и выходных сигналов, указывать выводы питания, выводы типа «вход-выход», формировать подгружающие эффекты входов;
- указывать выходные нагрузочные характеристики для выходных выводов;
- указывать законы преобразования электрических уровней в логические состояния, а законы логических состояний – в электрические уровни;
- описывать логические функции устройства на основе логических термов;
- реализовать циклы, использовать условные ветвления;

- объявлять целочисленные и действительные переменные для вычисления промежуточных результатов или внутренних состояний;
- указывать задержки передачи сигналов;
- формировать события, привязанные к определённому времени исполнения.

Модель допускает формирование трёх типов характеристик задержки распространения сигнала: минимальной, типовой и максимальной. При реализации модели указывается нужный тип характеристики, или он назначается по какому-либо параметру имитатора Mixed Sim. Это увеличивает гибкость процесса моделирования, позволяя имитировать технологический разброс быстродействия при изготовлении микросхем.

Функции языка Digital SimCode описывают 4 статических состояния: 1, 0, X и Z, а также динамические переходы между ними. Кроме того, эти состояния показывают эквивалентные внутренние сопротивления выходов, что, в свою очередь, позволяет учитывать перекрёстное взаимодействие выводов двух и более цифровых моделей, имеющих общие внешние цепи. Этот механизм аккуратно вычисляет актуальные действующие напряжения и токи.

Digital SimCode содержит функции, предназначенные для улучшения качества отладки моделей за счёт формирования сообщений потока моделирования.

Также работа модели цифрового устройства может опираться на дополнительные файлы, содержащие переменные данные. Например, это могут быть коды команд цифрового устройства в определённом порядке. Меняя последовательность кодов в файле данных, можно влиять на порядок операций, выполняемых моделируемым цифровым устройством. Примерно так, как это происходит в однокристальных микроконтроллерах и подобных устройствах. Файлы с такими последовательностями данных можно формировать в иных программах или вручную. Так можно менять поведение модели без изменения описания на Digital SimCode. Все функции языка в совокупности позволяют формировать сколь угодно сложное поведение моделируемого цифрового устройства.

## Кодированные модели

Обычно модель цифрового устройства описывают в виде плоского текстового файла в формате ASCII. Для файла незашифрованного описания модели цифрового устройства на Digital SimCode используют расширение .txt. Это позволяет в полной мере использовать и модифицировать модель. В процессе запуска расчёта имитатором производится предварительная сборка модели в сжатом зашифрованном виде. Этот вид можно извлечь из log-файла запуска расчёта и перенести в плоский файл в формате ASCII. Для файлов зашифрованного описания модели цифрового устройства следует использовать расширение .scb.

Файлы описания моделей в зашифрованном формате защищают модель от преднамеренного внесения изменений. Однако это порождает необходимость держать совместно с зашифрованной моделью описание способа назначения портов модели к выводам УГО элемента (иначе называемого шаблоном назначения) для имитатора Mixed Sim. При отсутствии сведений о перечне, порядке и назначении внешних выводов модели нет возможности получить доступ к функциональности.

## Назначение модели Digital SimCode для Altium Mixed Sim. Требования к разработке модели

Модель используется с указанием свободного шаблона модели устройства Altium Mixed Sim. В диалогe Sim Model на закладке Model Kind нужно выбрать General в поле Model Kind и Generic Editor – в поле Model Sub-Kind. Затем указывается значение A в поле SPICE Prefix, формируется шаблон на нижней закладке Netlist Template и выбирается путь к файлу, содержащему описание модели-прототипа цифрового устройства имитатора Altium Mixed Sim в поле Model Name. Если файл с описанием модели-прототипа указан корректно, то на нижней закладке Model File будет приведено содержание файла.

### Шаблон модели Mixed Sim

Шаблон должен содержать хотя бы одну строку вида:

```
@designator [перечень имён внешних цепей для входных выводов]
[перечень имён внешних цепей выходных выводов] @model.
```

Причём перечни имён цепей должны быть приведены в том же поряд-

ке, в котором они приводятся в описании модели цифрового устройства на языке Digital SimCode (из этого следует необходимость для зашифрованной модели предоставлять ещё и шаблон). Для упрощения формирования шаблона можно сделать предварительное сопоставление вида: <вывод УГО>-<порт модели Mixed Sim>-<вывод модели Digital SimCode>.

### Модель Mixed Sim (параметры модели)

Файл описания модели-прототипа цифрового устройства Altium Mixed Sim должен содержать одну строку вида:

- .model <имя прототипа> xsimcode(file=<имя файла> func=<имя прототипа> [data=<имя файла данных>] [mntpmx]);
- <имя файла> – полное имя файла, содержащего описание прототипа устройства на Digital SimCode;
- <имя прототипа> – наименование функции-прототипа описания устройства Digital SimCode;
- <имя файла данных> – полное имя файла данных, содержащего исходные данные, обрабатываемые моделью;
- [mntpmx] – указание передачи следующих параметров внутрь модели:
  - Propagation – задержка распространения в устройстве;
  - Loading – входная нагрузочная характеристика;
  - Drive – выходная нагрузочная характеристика;
  - Current – ток потребления устройства.

Установка значений MIN или MAX для параметров Propagation, Loading, Drive, Current указывает на минимальное и максимальное значения компонента. По умолчанию используется типичное значение.

PWR value – напряжение питания положительной полярности. Определение параметра перегружает определение зарезервированной переменной в модели-прототипе. Если данный параметр определён, то совместно с ним должен быть определён параметр GND value.

GND value – напряжение питания отрицательной полярности. Определение параметра перегружает определение зарезервированной переменной в модели-прототипе. Если данный параметр определён, то совместно с ним должен быть определён параметр PWR value.

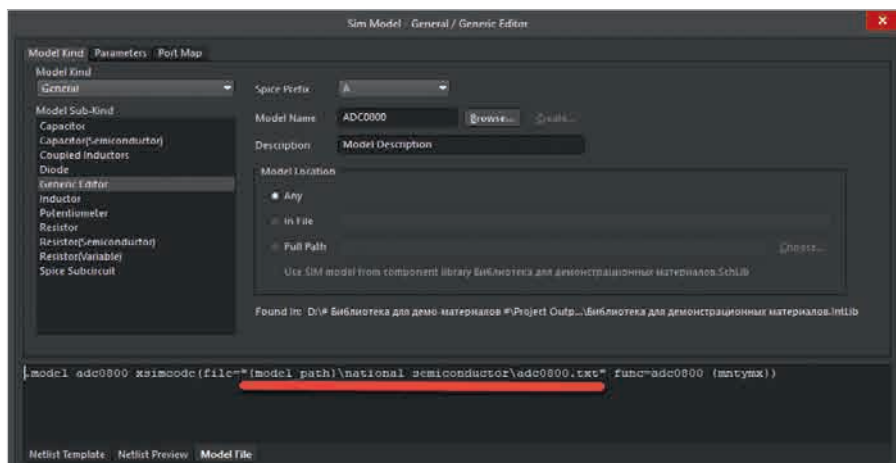


Рис. 1. Определение местоположения файла модели-прототипа Digital SimCode

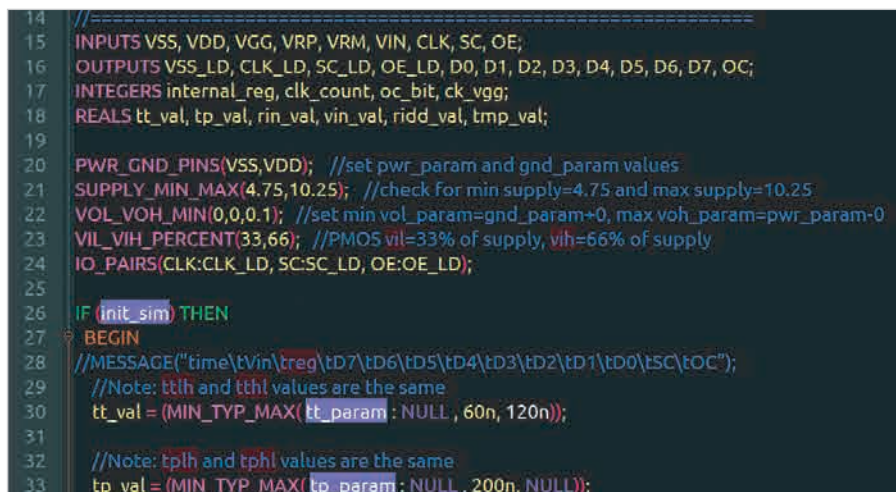


Рис. 2. Код модели в текстовом редакторе

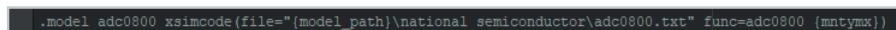


Рис. 3. Файл mdl с описанием модели-прототипа Mixed Sim

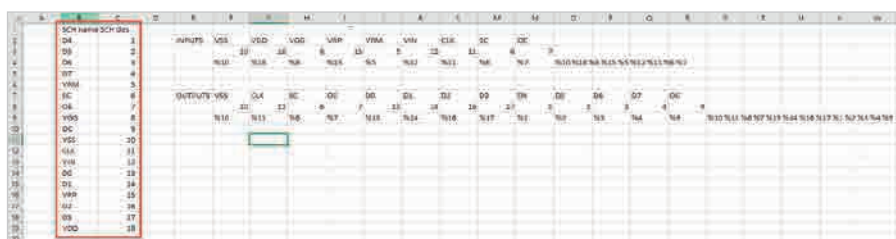


Рис. 4. Перенос перечня входных и выходных выводов в таблицу

VIL value – входное напряжение, интерпретируемое как сигнал низкого уровня.

VIN value – входное напряжение, интерпретируемое как сигнал высокого уровня.

VOL value – выходное напряжение, эквивалентное сигналу низкого уровня.

VOH value – выходное напряжение, эквивалентное сигналу высокого уровня. Определение параметров VIL value, VIN value, VOL value, VOH value перегружает определение зарезервированной переменной в модели-прототипе.

WARN – установка значения ON является флагом для ошибок: времени установки сигнала и удержания, ширины импульса, нарушения минимально-максимальных частот и напряжения питания. Сообщение об ошибке появляется в течение всей имитации после возникновения нарушения.

### Особенности разработки модели Digital SimCode

При разработке модели Digital SimCode следует учитывать следующее:

- описание функции-прототипа устройства на Digital SimCode не должно содержать кириллических символов;
- <имя файла> должно быть обрамлено двойными кавычками;
- <имя файла> не должно содержать кириллических символов;
- расширение <имени файла> должно быть .txt или .scb;
- <имя файла данных> должно быть обрамлено двойными кавычками;
- <имя файла данных> не должно содержать кириллических символов;
- расширение <имени файла данных> должно быть .dat;
- <имя файла> и <имя файла данных> могут содержать специальный префикс {MODEL\_PATH};
- признак {mntpmx} указывается в описании модели для Mixed Sim при необходимости передать специальные параметры модели.

Специальные параметры модели имеют соответствия в перечне параметров симулятора (задаются в разделе Advanced диалога Analyses Setup). В результате приоритет определений выставляется от структурно располагающихся наиболее близко к коду модели. Самый низкий приоритет у специальных переменных в модели-прототипе Digital SimCode, затем приоритет за значениями параметров симулятора, для передачи значений параметров симулятора должен быть указан флаг {mntpmx}, самый высокий уровень приоритета имеют определения специальных параметров модели Mixed Sim, должен быть указан флаг {mntpmx}.

### Приём реализации назначения модели

Назначение модели Digital SimCode является само по себе не очень сложным, но в процессе можно быстро запутаться, так как приходится согласовывать выводы УГО, контейнерной модели-прототипа xsimcode для Mixed Sim и выводы модели-прототипа Digital SimCode. Для этого используются данные шаблона экземпляра модели и информация на закладке Port Map.

Однако процедуру можно значительно упростить, используя простой приём назначения. Нужно привести обозначения выводов УГО в соответствие номерам выводов модели Mixed Sim. Таким образом, на этапе сопоставления на закладке Port Mapping диалога Sim Model получился результат вида «первый к первому, второй ко второму» и т.д.



При этом следует учитывать, что выводы модели Mixed Sim появятся только после того, как будут указаны в шаблоне экземпляра. Поэтому предварительной целью для операции назначения портов является разработка шаблона.

Разработать шаблон экземпляра можно либо на основании исходного описания модели Digital SimCode, либо на основании сведений о порядке перечисления и назначении портов в этой модели в разделах Inputs и Outputs. Удобно разрабатывать шаблон уже после разработки mdl-файла, содержащего описание модели-прототипа Mixed Sim. Отсюда можно вывести следующий порядок действий: найти файл модели-прототипа Digital SimCode; выписать из него перечень входных и выходных сигналов, пар выводов «вход-выход» и пары выводов питания; разработать описание модели-прототипа для Mixed Sim; разработать шаблон экземпляра; выполнить назначение портов модели-прототипа Mixed Sim и выводов УГО; проверить результат.

### Практический пример 1 (назначение модели)

Рассмотрим приём назначения модели на основе модели микросхемы АЦП ADC0800. Предварительно имеем разработанное УГО элемента. Рассмотрим набор шагов для настройки назначения модели Digital SimCode элементу.

В качестве первого шага следует определиться с местоположением файла модели-прототипа Digital SimCode и именем файла. Удобно будет скопировать его в пустой каталог поближе к корневому каталогу – это позволит использовать более короткие пути в определениях модели-контейнера (см. рис. 1). Затем открыть файл модели в текстовом редакторе (см. рис. 2). Создать mdl-файл с описанием модели-прототипа Mixed Sim. Следует указать путь к файлу и его имя, а также название модели, описывающей АЦП (см. рис. 3). Перенести перечень входных и выходных выводов в электронную таблицу (см. рис. 4). Перенести в ту же электронную таблицу перечень парных соответствий входных и выходных выводов (см. рис. 5). Выполнить сопоставление. Перечень входных и выходных выводов модели Mixed Sim должен быть дан в том же порядке, в котором перечислены входные и выходные порты модели Digital SimCode (см. рис. 6). Сформировать шаблон экземпляра модели Mixed Sim (см. рис. 7). Проверить

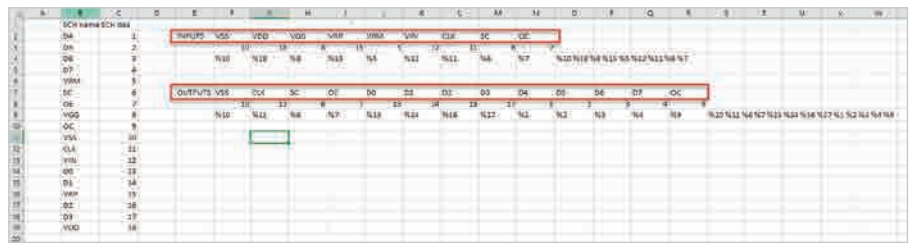


Рис. 5. Перенос перечня парных соответствий входных и выходных выводов в таблицу

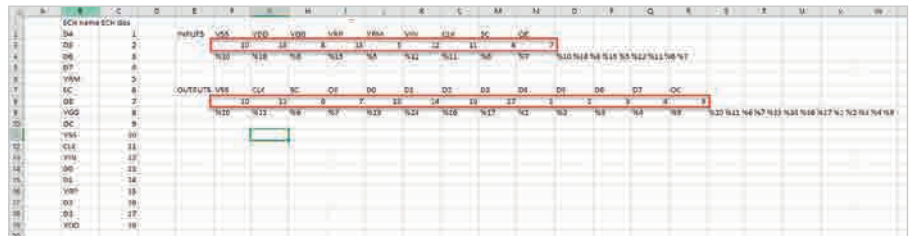


Рис. 6. Сопоставление входных и выходных выводов модели Mixed Sim с входными и выходными портами модели Digital SimCode

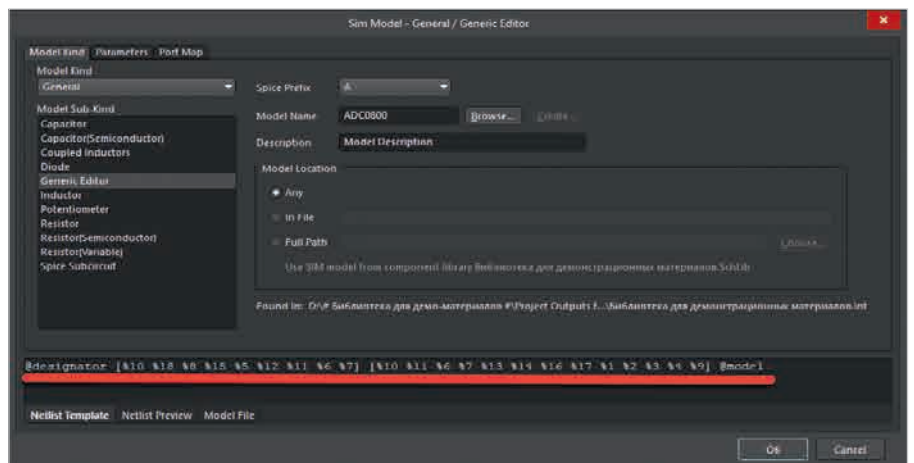


Рис. 7. Формирование шаблона экземпляра модели Mixed Sim

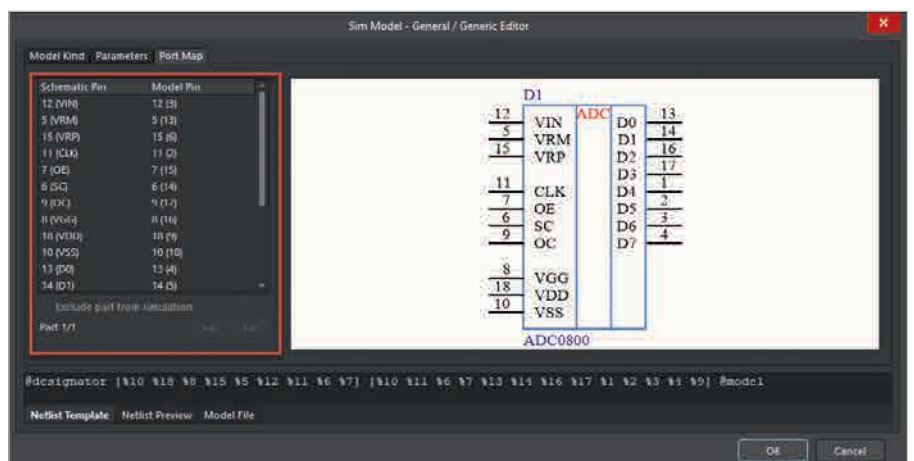


Рис. 8. Проверка информации на закладке Port Map диалога Sim Model

информацию на закладке Port Map диалога Sim Model и откорректировать при необходимости (см. рис. 8). В качестве последнего шага следует проверить корректность назначения mdl-файла на закладке Model File диалога Sim Model (см. рис. 9). Описание

модели должно состоять из следующих условных разделов:

- идентификация функции устройства;
- объявление данных;
- инициализация модели;
- определение нагрузочных характеристик;



стояние вывода с состоянием при предыдущем вызове модели;

3. RECOVER – тестирует порты на нарушения времени восстановления;
4. SETUP\_HOLD – тестирует порты на нарушения времени установки/удержания;
5. SUPPLY\_MIN\_MAX – проверяет разницу напряжений на входах питания;
6. WIDTH – сравнивает ширины импульсов для входных портов с указанным временем.

### Определение задержек выходных сигналов и сообщений событий

Эффекты запаздывания и распространения выходного сигнала можно имитировать с помощью оператора: `DELAY <выход1> [<выход2>...] = <задержка>`.

Вместо <длительность> можно использовать несколько связанных операторов CASE, тогда выражение будет иметь вид:

```
DELAY <выход> [<выход> ...] =
CASE (<условное выражение>) :
<задержка>
CASE (<условное выражение>) :
<задержка>
[CASE (<условное выражение>) :
<задержка> ...]
END;
```

Оператор DELAY может иметь два вида синтаксиса и формировать события при помощи оператора EVENT. Если производится отладка модели или модель разрабатывается с внедрением контроля исполнения, то для формирования сообщений используется оператор PROMPT.

### Листинг 3

```
//=====Идентификация функции устройства=====
# adc0800 source
//=====Объявление данных=====
INPUTS VSS, VDD, VGG, VRP, VRM, VIN, CLK, SC, OE; // объявление входных портов модели - принимающих сигналы
OUTPUTS VSS_LD, CLK_LD, SC_LD, OE_LD, D0, D1, D2, D3, D4, D5, D6, D7, OC; //объявление выходных портов модели - формирующих сигналы
INTEGERS internal_reg, clk_count, oc_bit, ck_vgg; // объявление внутренних целочисленных переменных
REALS tt_val, tp_val, rin_val, vin_val, ridd_val, tmp_val; // объявление внутренних действительных переменных
//==Смешанное объявление данных и определение контроля нарушений==
PWR_GND_PINS (VSS,VDD); // установка значений параметров pwr_param и gnd_param
SUPPLY_MIN_MAX (4.75,10.25); // контроль минимального напряжения питания =4.75 и максимального напряжения питания =10.25
VOL_VOH_MIN (0,0,0.1); //установка минимума напряжения для выходных портов vol_param=gnd_param+0, максимума напряжения для выходных портов voh_param=pwr_param-0
VIL_VIH_PERCENT (33,66); // определение значений параметров vil=33% от размаха питания, vih=66% от размаха питания для Р-структуры МДП
IO_PAIRS (CLK:CLK_LD, SC:SC_LD, OE:OE_LD); // определение портов CLK, SC, OE в качестве входов-выходов
//=====Инициализация=====
IF (init_sim) THEN
BEGIN
//MESSAGE ("time\tvin\treg\tD7\tD6\tD5\tD4\tD3\tD2\tD1\tD0\tSC\tOC");
// Приведены параметры сопротивления (системные параметры для модели)
//Примечание: значения ttlh и thl одинаковы
tt_val = (MIN_TYP_MAX(tt_param: NULL, 60n, 120n));
//Примечание: значения tplh и tphl одинаковы
tp_val = (MIN_TYP_MAX(tp_param: NULL, 200n, NULL));
//Параметры выходных портов для низкого уровня напряжения IOI max=2 мА @ vol=0.4 В => rol_param=(0.4-vol_param)/2 мА
rol_param = (MIN_TYP_MAX(drv_param: NULL, 100, NULL));
//Параметры выходных портов для высокого уровня напряжения IOH max=-200 мкА @ voh=4.5 В => roh_param=(voh_param-4.5)/200 мкА
roh_param = (MIN_TYP_MAX(drv_param: NULL, 500, NULL));
//Утечка входа @ 25 °C: IIN=1 мкА @ VDD=5 В => rin= (5/1 мкА);
rin_val = (MIN_TYP_MAX(ld_param: NULL, 5E6, NULL));
```

При этом останавливается программный поток модели. Действие оператора MESSAGE подобно оператору PROMPT за исключением того, что не происходит остановки программного потока.

### Практический пример 2 (разбор модели ADC0800)

В качестве примера условного описания структуры модели Digital SimCode рассмотрим листинг 3 модели микро-

схемы АЦП National Instruments ADC0800.

### Литература

1. Вебинар «Разработка моделей цифровых элементов Digital SimCode для Altium Mixed Sim». URL: <https://www.youtube.com/watch?v=PNC9QIWlJPw>.
2. Руководство по Digital SimCode. URL: <https://ru.resources.altium.com/russian-guide-books/руководство-digital-simcode>.



## НОВОСТИ МИРА

### ШВЕЙЦАРСКИЙ СТАРТАП СОБИРАЕТ СРЕДСТВА НА РАЗРАБОТКУ ПРОЦЕССОРА EDGE AI

Synthara AG (Цуг, Швейцария), стартап 2017 года, работающий над чипами машинного обучения, привлекла начальное финансирование. Размер финансирования не разглашается, но известно, что в стартап вложились Grunderfonds. Компания Synthara AG создана Ману Наиром, Алессандро Аймаром и Юлией-Александрой Лунгу, которые ранее трудились в Институте нейроинформатики UZH-ETH (INI) в Цюрихе.

Компания разрабатывает двухъядерный RISC-V ASIC с интегрированным ускорителем AdaptiveStorm. В сторожевом режиме потребляемая мощность AdaptiveStorm будет составлять менее 1 мВт и менее микроватта в некоторых случаях использования, а при работе – до 10 Вт.

Adaptiva предназначен для поддержки сверточных нейронных сетей, таких как resNet, VGG и MobileNet, а также многослойных рекуррентных нейронных сетей. Чип будет разработан для поддержки слияния и интерпретации данных от датчиков со стандартными аудио- и видеоинтерфейсами, включая MIPI, I<sup>2</sup>C, SPI и прямые интерфейсы к необработанным данным датчиков, таким как микрофоны, аудиосистемы и акселерометры.

Первый набор продуктов Synthara должен появиться в 2022 году. В настоящее время Synthara готовит библиотеки компиляторов, которые оптимизируются для повышения энергоэффективности. Для обучения нейронных сетей разработчики смогут использовать популярные библиотеки ИИ, такие как PyTorch или Tensorflow. Эти сети будут портированы

на Adaptiva посредством компиляторов, которые также распределяют рабочую нагрузку для обеспечения максимальной энергоэффективности. Прямая запись в Adaptiva алгоритмов машинного обучения может быть выполнена сотрудниками Synthara.

Алессандро Аймар, технический директор компании Synthara, утверждает, что чипы с учётом новых алгоритмов обеспечат в 500 раз лучшую производительность для следующего поколения интеллектуальных сенсорных приложений, использующих инерциальные измерительные блоки, датчики звука и изображения. Примечательно, что Synthara нацелена на применение в носимых устройствах, IoT и интеллектуальном мониторинге, в том числе и медицинском оборудовании.

[www.synthara.ai](http://www.synthara.ai)