



9. Газизов А.Т., Заболоцкий А.М., Газизов Т.Р. Разложение сверхкороткого импульса в структурах с лицевой связью // Известия высших учебных заведений. Физика. – 2017. – №3. – с. 70-75.

10. Белоусов А.О., Заболоцкий А.М., Газизов Т.Р. Экспериментальное подтверждение модельной фильтрации в многопроводной микрополосковой линии // Доклады Томского государственного университета систем управления и радиоэлектроники. 2016. – №3. – с. 51-54.

Н.В. Решетникова, И.Г. Криволапчук

ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ЗАПОМИНАЮЩИХ УСТРОЙСТВ В СИСТЕМЕ MULTISIM

(Санкт-Петербургский государственный университет аэрокосмического приборостроения)

Процесс разработки сколько-нибудь сложного электронного устройства обязательно включает в себя этап макетирования, позволяющий проверить общую работоспособность устройства, выявить и устранить схемные ошибки, оптимизировать исходную схему. Некоторым недостатком чистого макетирования является то, что при разработке относительно сложного, но штучного изделия макет вносит существенный вклад в общую стоимость разработки. В первую очередь это относится к схемам с использованием дорогостоящих одноразовых элементов, то есть элементов, чья функциональность не может быть в полной мере восстановлена до исходной при демонтаже и последующем повторном использовании.

Существенно улучшить ситуацию помогло появление симулятора электронных схем общего назначения SPICE, разработанного в Electronics Research Laboratory Калифорнийского университета в Беркли. Стоит отметить, что абсолютное большинство современных средств проектирования электронных схем базируется на версии SPICE3f5 данного симулятора. Широкому распространению SPICE способствовала не только его функциональность, но и доступность и открытость исходного кода, что дало толчок появлению таких расширенных вариантов симулятора, как XSPICE[1] (Технологический институт Джорджии), поддерживающего работу со смешанными аналого-цифровыми моделями. Именно этот вариант используется в системе NI Multisim™[2] в качестве ядра моделирования.

Запоминающие устройства различных типов и технологий являются неотъемлемой частью вычислительных устройств. Постоянные запоминающие устройства (ПЗУ), как следует из названия, в первую очередь предназначены для долговременного хранения программного кода. В перезаписываемых ПЗУ могут сохраняться, например, промежуточные результаты работы или конфигурационная информация микропроцессорной системы. Однако эксплуатационные свойства ПЗУ позволяют значительно расширить область их примене-



ния, используя в качестве основы для построения различных цифровых и цифро-аналоговых устройств.

Одним из частых применений микросхем ПЗУ является замена ими сложных или громоздких комбинационных схем, что позволяет в ряде случаев существенно улучшить показатели проектируемого устройства. Среди других применений ПЗУ можно назвать построение на их основе генераторов сложных последовательностей цифровых импульсов, генераторов аналоговых сигналов произвольной формы, функциональных преобразователей. Еще одним интересным применением ПЗУ является создание на их базе микропрограммных автоматов, позволяющих реализовывать достаточно сложные и гибкие алгоритмы управления.

При попытке моделирования описанных выше и подобных им устройств в среде Multisim пользователь сталкивается с определенными трудностями, связанными с отсутствием доступных штатных инструментов для записи информации в моделируемые запоминающие устройства. Ниже рассматриваются несколько методов использования базовых и расширенных возможностей системы моделирования, позволяющих решить поставленную задачу.

Метод редактирования на лету основан на свойствах элемента схемы, имеющего в своей основе базовую модель запоминающего устройства `d_ram` или `d_rom`. Для таких элементов симулятор автоматически создает окно просмотра и редактирования содержимого, доступное при переводе симулятора в состояние паузы.

Будучи предельно простым, этот метод обладает существенным недостатком, серьезно ограничивающим его применение – отсутствует возможность сохранения данных. Любое изменение схемы или прерывание моделирования приводит к необходимости повторного программирования ПЗУ. Рекомендовать к использованию этот метод можно только в случае работы с небольшими, не более нескольких десятков байт, объемами памяти.

Добавление микроконтроллера к моделируемой схеме без его подключения позволяет воспользоваться всеми возможностями, которые система предоставляет при моделировании устройств на базе микроконтроллеров[3]. В этом случае во вспомогательном диалоге создания проекта (MCU Wizard) необходимо выбрать опции использования внешнего файла прошивки и пустого проекта. Далее в созданном проекте через MCU code manager указывается, что прошивка будет загружаться во внешнее ПЗУ. Подготовленный в формате Intel HEX файл прошивки помещается в каталог проекта. При запуске моделирования данные из файла будут автоматически загружены в выбранное ПЗУ.

Пример использования данного подхода приведен на рис.1. Микросхема ПЗУ использована в качестве основы генератора сигналов. Так же, как и в предыдущем методе, в режиме паузы доступно окно просмотра и редактирования. Производимые изменения не сохраняются при перезапуске процесса моделирования, поэтому все вносимые изменения должны быть перенесены в файл прошивки с последующей его заменой в каталоге проекта.

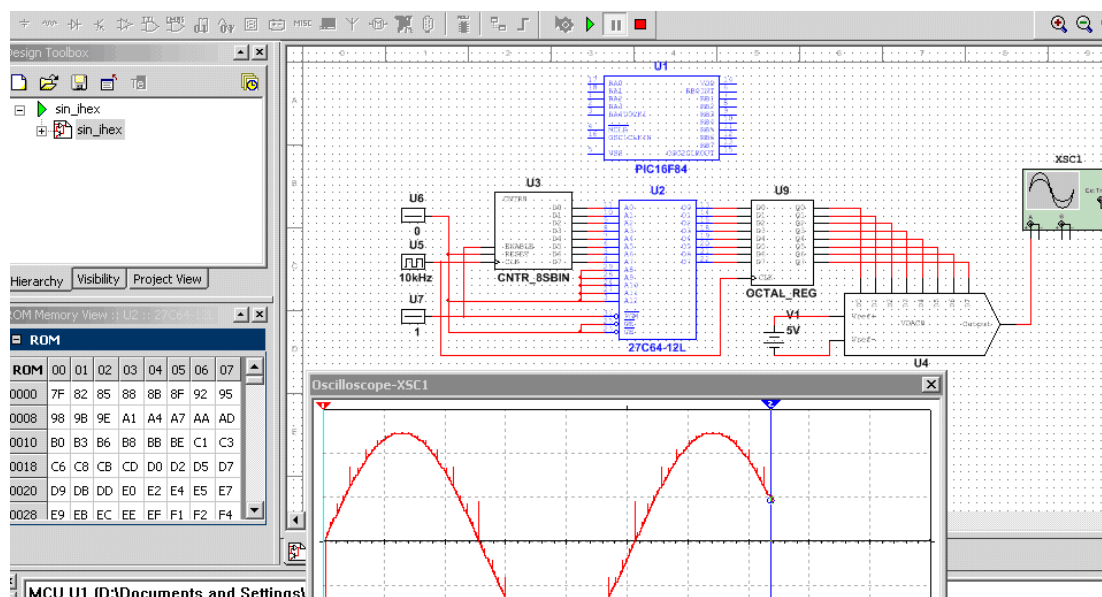


Рис. 1. Генератор сигналов на базе ПЗУ.

Недостатком метода является потребление определенных временных ресурсов присутствующим микроконтроллером и невозможность одновременного использования нескольких микросхем ПЗУ в одном проекте. При моделировании схемы с несколькими запоминающими устройствами, а, следовательно, и с несколькими микроконтроллерами, указанные недостатки могут оказать заметное воздействие на скорость моделирования.

Представленные методы моделирования используют исключительно базовые возможности системы. Методы, описываемые ниже, опираются на расширения симулятора и предполагают наличие у пользователя навыков создания моделей для Multisim и умения создавать программы на языке C.

Пользовательская модель, как и все встроенные в симулятор модели, представляет собой комбинацию визуального элемента – изображения элемента на схеме, и функционального наполнения, описывающего поведение модели. Визуальный элемент (ВЭ), кроме собственно изображения и описания выводов, может содержать в себе элементы анимации, позволяющей визуализировать режимы работы или состояние модели. Для создания и редактирования ВЭ используется инструмент SymEditor, входящий базовый набор программных средств Multisim.

Содержание функциональной части модели зависит от типа моделируемого элемента и возможностей, вкладываемых в модель разработчиком. Минимальное содержание представляет собой шаблон для включения элемента в структуру SPICE-модели. Описание работы элемента может быть выполнено с использованием языка SPICE, скриптовых расширений Multisim на основе FORTH-подобного языка или через ссылку на внешнюю dll-библиотеку, содержащую программную реализацию модели элемента.

Наиболее простым из расширенных методов является использование универсального цифрового элемента. Универсальный цифровой элемент – d_chip[4] – является элементом, специфичным именно для версии XSPICE, ис-



пользуемой в Multisim. Он представляет собой средство моделирования произвольной комбинационной логики, поведение которой описывается таблицей истинности. Особенностью данного элемента является передача описания поведения схемы через строку параметров, что позволяет одновременно использовать в схеме несколько однотипных моделей на базе d_chip с различным поведением.

Основными командами, используемыми при моделировании ПЗУ через d_chip, являются /input, /output и /table. Две первые команды служат для описания связей входных и выходных сигналов внутри d_chip с внешними выводами модели, команда /table задает таблицу истинности элемента. Представленный ниже код описывает модель ПЗУ 16x8 бит с прошивкой, реализующей реверсивный распределитель импульсов управления для бесконтактного двигателя постоянного тока.

```
.MODEL myROM d_chip ( behaviour= "  
+;6 step encoder  
+;      C  B  A  Dir  
+/inputs  A2 A1 A0 A3  
+;      H1 L2 H3 L4 H5 L6  
+/outputs D0 D1 D2 D3 D4 D5  
+/table 13  
+;  
+; C B A  Dir  1 2 3 4 5 6  
+; forward  
+  H L H   L   H L L H L L  
...  
+  H L L   L   L L L H H L  
+; backward  
+  L H H   H   L L L H H L  
...  
+  L L H   H   L H L L H L  
+; disable  
+  X X X   X   L L L L L L  
+" )
```

Преимуществом подобной реализации модели ПЗУ является сравнительная простота создания и модификации прошивки. Несмотря на отсутствие возможности непосредственного просмотра и редактирования содержимого памяти в процессе моделирования, прошивка может быть легко модифицирована встроенным редактором моделей или внешним текстовым редактором. Также к положительным свойствам можно отнести хорошую читаемость модели и возможность её автоматической генерации с использованием скриптовых языков. Недостатки данного типа моделей в основном сводятся к битово-ориентированной записи прошивки и увеличенному времени обработки больших последовательностей при случайном доступе.

Разработка внешнего подключаемого программного модуля[5] является наиболее универсальным по своим возможностям подходом, но при этом до-



статочно сложным по реализации и в значительной мере избыточным. Преимуществом метода является получение пользователем полного контроля работы модели запоминающего устройства, возможности свободного создания, модификации и загрузки прошивки в любом удобном формате. В случае грамотно разработанного программного кода модель будет потреблять минимальное процессорное время для обработки матрицы памяти произвольного объема и организации. Серьезными недостатками являются сложность отладки и возможность вызвать крах системы моделирования при программных сбоях. Рекомендовать к использованию данный метод можно только для очень опытных пользователей, хорошо представляющих себе функционирование системы моделирования и взаимосвязь её компонентов.

В качестве заключения следует отметить, что Multisim предоставляет весьма широкий набор инструментов и средств, позволяющий пользователю, пусть и не всегда очевидным образом, адаптировать возможности системы под решение конкретных задач.

Литература

1. XSPICE Software User's Manual. Computer Science and Information Technology Laboratory, Georgia Tech Research Institute, 1992.
2. NI Multisim™ User Manual, 374483D-01, National Instruments Corporation.
3. Multisim™ MCU Module User Guide, 374486A-01, National Instruments Corporation.
4. Multisim™ SPICE Reference Manual, 374845A, National Instruments Corporation.
5. Cox F.L., Kuhn W.B., Murray J.P., Tynor S.D. Code-level modeling in XSPICE Proceedings of the IEEE International Symposium on Circuits and Systems, 1992 (ISCAS '92), Volume 2 , pp. 871-874, 10-13 May 1992.

Д.А. Рыбаков

МОДЕЛЬ СВЯЗАННОСТИ СОБЫТИЙ НА ОСНОВЕ МЕТОДОВ БАЙЕСА

(Eram systems)

Допустим, требуется разместить инвестиции в нескольких предприятиях, которые могут обанкротиться с вероятностью 5%. Если имеется 2 предприятия, тогда вероятность того, что они оба обанкротятся, и мы потеряем все вложения $P = 0.05 \cdot 0.05 = 0.0025$. Так учит стандартная теория вероятности. Но что будет, если предприятия связаны, и банкротство одного ведет к банкротству другого? Крайним случаем является ситуация, когда предприятия полностью зависимы. Для полностью зависимых предприятий вероятность двойного банкротства и потери всех вложений остается равной $P = 0.05$. Получается, что наивная методика оценки риска имеет большой разброс P от 0.05 до 0.0025, и реальное