

Рис. 2. Моделирование частотной избирательности слуха

В результате выходом алгоритма является массив коэффициентов ДПФ той же размерности, что и входной, но с некоторыми обнуленными коэффициентами.

В процессе эксперимента использовались 2 тестовых звуковых файла с музыкальным и речевым фрагментами, а также 3 человека, прослушивающие исходные файлы и файлы, прошедшие цифровую обработку в модели.

Результат моделирования с использованием свойства частотной избирательности человека показал, что 25 % спектральных составляющих аудиоданных при их сжатии можно удалить. При этом ухудшения качественного восприятия обработанных программой тестовых аудиоданных не наблюдалось.

Литература

1. Цвикер Э., Фельдкеллер Р. Ухо как приемник информации / пер. с немец, под ред. Б. Г. Белкин. М.: Связь, 1971 – 255 с.
2. Воскобойников Ю. Регрессионный анализ данных в пакете Mathcad. М: Лань, 2011 – 224 с.
3. Ковалгин Ю. А., Вологдин Э. И. Цифровое кодирование звуковых сигналов.–СПб.: КОРОНА-принт, 2004–240 с., ил.
4. Стефанова И. А., Павлов К.В. Моделирование свойств слухового анализатора. Уфа: Международный центр инновационных исследований «Символ науки» № 11, часть 1, 2015 – 58-62 с.

А.В. Папе, И.А. Стефанова

ИСПОЛЬЗОВАНИЕ ПОДСИСТЕМ ДЛЯ МОДЕЛИРОВАНИЯ УСТРОЙСТВ В SIMULINK

(Поволжский государственный университет
телекоммуникаций и информатики)

Современная математическая система *MATLAB* в комплексе с пакетом расширения *Simulink* обладает огромным количеством различных библиотек, позволяющих проводить как сложные математические вычисления, так и моде-



лизовать работу любого сколь угодно сложного технического устройства. Базируясь на принципах визуально-ориентированного программирования, *Simulink* позволяет выполнять моделирование сложных устройств с высокой степенью достоверности и обладает наглядными средствами представления результатов. Это позволяет пользователям разработать новые устройства, понять работу уже созданных систем, просчитать характеристики тех или иных блоков системы, представить наглядно их работу и результаты этой работы. Все это существенно упрощает процесс обучения студентов.

Однако зачастую схемы, создаваемые на базе простейших блоков из библиотек *Simulink*, получаются объемными, размытыми по рабочей области окна моделирования, имеют большое количество переплетающихся между собой соединений. Всё это значительно затрудняет процесс изучения схемы и её работы, мешает в целом восприятию изучаемого объекта моделирования. Чтобы избежать данной проблемы можно использовать приём, про который часто забывают пользователи *Simulink* – создание и маскирование подсистем.

Подсистема – это фрагмент *Simulink*-модели, оформленный в виде отдельного функционального блока. Использование подсистем при составлении моделей имеет следующие положительные стороны:

- уменьшение количества одновременно отображаемых блоков на экране, что облегчает восприятие модели (в идеале модель полностью должна отображаться на экране монитора);
- создание и отладку фрагментов модели по отдельности, что повышает технологичность создания модели;
- создание собственных библиотек;
- возможность синхронизации параллельно работающих подсистем;
- включение в модель собственные справочные средства;
- возможность связывания подсистемы с каким-либо *m*-файлом системы *MATLAB*, обеспечивая запуск этого файла при открытии подсистемы (нестандартное открытие подсистемы).

Использование подсистем и механизма их соединения в единый блок позволяет создавать блоки, не уступающие стандартным по своему оформлению (собственное окно параметров блока, пиктограмма, справка и т.п.). Этот процесс представляет собой маскирование систем [1].

Количество подсистем в модели не ограничено, кроме того подсистемы могут включать в себя другие вложенные подсистемы. Уровень вложенности подсистем друг в друга также не ограничен.

Связь подсистемы с моделью (или подсистемой верхнего уровня иерархии) выполняется с помощью входных (блок *Inport* библиотеки *Sources*) и выходных (блок *Outport* библиотеки *Sinks*) портов. Добавление в подсистему входного или выходного порта приводит к появлению на изображении подсистемы метки порта, с помощью которой внешние сигналы передаются внутрь подсистемы или выводятся в основную модель. При моделировании подсистем в *Simulink* имеется возможность переименования блоков *Inport* или *Outport* со стандартных названий *In / Out* на те, которые нужны пользователю.



Пакет расширения *Simulink* содержит как виртуальные подсистемы (*Subsystem*), так и монолитные (*Atomic Subsystem*). Отличие этих видов подсистем заключается в порядке выполнения блоков во время расчета [3]. Если подсистема является виртуальной, то *Simulink* игнорирует наличие границ отделяющих такую подсистему от модели при определении порядка расчета блоков. Иными словами в виртуальной системе сначала могут быть рассчитаны выходные сигналы нескольких блоков, затем выполнен расчет блоков в основной модели, а затем вновь выполнен расчет блоков входящих в подсистему. Монолитная подсистема считается единым (неделимым) блоком и *Simulink* выполняет расчет всех блоков в такой подсистеме, не переключаясь на расчеты других блоков в основной модели.

В свою очередь, подсистемы могут быть также управляемыми или неуправляемыми. Управляемые подсистемы всегда являются монолитными. Они имеют дополнительные (управляющие) входы, на которые поступают сигналы, активизирующие данную подсистему. Управляющие входы, как правило, расположены сверху или снизу подсистемы. Когда управляемая подсистема активизирована – она выполняет вычисления. В том случае если управляемая подсистема пассивна, то она не выполняет вычисления, а значения сигналов на ее выходах определяются настройками выходных портов.

Существуют два способа создания в модели подсистемы:

1) Использование команды создания подсистем. Для этого необходимо скопировать нужную подсистему из библиотеки *Subsystem* в модель. Выделить с помощью мыши нужный фрагмент модели и выполнить команду *Create Subsystem* из меню *Edit* окна модели. Выделенный фрагмент будет помещен в подсистему, а входы и выходы подсистемы будут снабжены соответствующими портами. Этот способ позволяет создать виртуальную неуправляемую подсистему. В дальнейшем, при необходимости можно сделать подсистему монолитной, изменив ее параметры, или управляемой, добавив управляющий элемент из нужной подсистемы, находящейся в библиотеке.

2) Использование механизма маскирования подсистем, который позволяет оформить подсистему как полноценный библиотечный блок, т.е. снабдить подсистему собственным окном параметров, пиктограммой, справочной системой и т.п.

Маскирование подсистем дает пользователю следующие преимущества:

- расширение возможности по управлению параметрами модели;
- создание более понятного интерфейса подсистемы;
- повышение наглядности блок-диаграмм;
- расширение возможности построения сложных моделей;
- повышение защищенности модели от несанкционированной модификации.

Для выполнения маскирования имеющейся подсистемы необходимо предварительно выполнить следующие действия:



1. Определить какие параметры подсистемы должны задаваться пользователем в будущем окне параметров. Задать эти параметры в подсистеме с помощью идентификаторов;
2. Определить каким образом параметр должен задаваться в окне диалога (с помощью строки ввода, выбором из раскрывающегося списка или установкой флажка);
3. Разработать эскиз пиктограммы блока;
4. Создать комментарии (справку) по использованию подсистемы.

Маскирование подсистемы выполняется с помощью редактор маски (*Mask Editor*). Для выделенных блоков запускается *Mask Editor*, при этом выводится окно с тремя вкладками *Icon* (Пиктограмма), *Initialization* (Инициализация), *Documentation* (Документация), с помощью которых настраивается, маскируется и сохраняется будущая подсистема.

После того как маскирование системы будет выполнено, двойной щелчок по ее изображению будет открывать уже окно параметров подсистемы, а не окно модели.

Для демонстрации возможностей использования подсистем в *Simulink* было смоделировано арифметико-логическое устройство в виде небольшой блок-схемы *Simulink*. Продемонстрируем её работу ниже.

Рассмотрим работу арифметико-логического устройства, описываемого формулой [2]:

$$F_i = (A_i B_i S_3 \vee A_i \bar{B}_i S_2) \oplus (\bar{B}_i S_1 \vee B_i S_0 \vee A_i) \oplus (C_0 \vee M),$$

где F_i – значение i -го разряда результата операции, A_i и B_i – значения i -го разряда двоичных операндов, S_i – значение i -го разряда кода операции АЛУ, C_0 – перенос в младший разряд АЛУ, M – выбор режима функционирования АЛУ. При $M=1$ выполняется логическая операция, при $M=0$ выполняется арифметическая операция над входными операндами A и B .

Из приведенного выражения видно, что для реализации АЛУ необходимы логические элементы *AND*, *OR*, *NOT* и *XOR*, которые реализуют логические операции конъюнкции (\wedge), дизъюнкции (\vee), отрицания ($\bar{}$), равнозначности (\oplus). По этому выражению была собрана модель АЛУ, представленная на рисунке 1. С целью упрощения восприятия АЛУ все элементы модели рисунка 1 а) были объединены в единую подсистему, которая приведена на рисунке 1 б).

В модели $A1$ и $B1$ – значения одного разряда двоичных операндов A и B соответственно.

Представленная на рисунке 1 модель одnorазрядного устройства АЛУ способно выполнять 32 различные операции, 16 из которых – логические и 16 – арифметические, реализуемые в соответствии с таблицей работы типовых АЛУ.

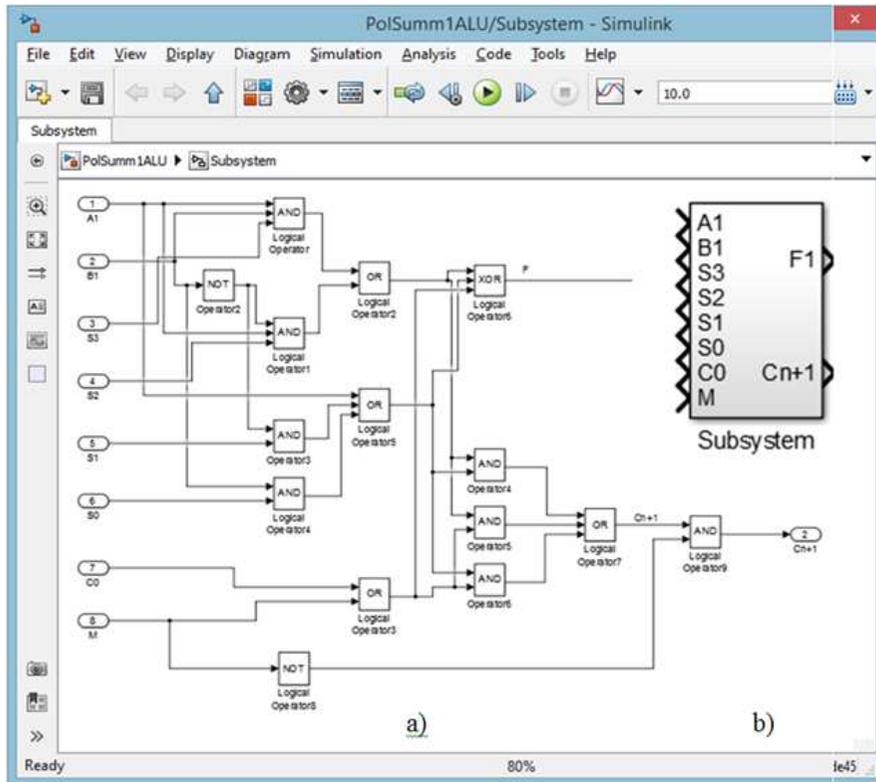


Рис. 1. Модель а) одноразрядного АЛУ и его б) подсистема (Subsystem)

Выводы: С помощью созданного блока рис. 1 б) подсистемы можно создавать устройства с наращиванием возможной разрядности, просто комбинируя и соединяя параллельно управляющие выходы. Соединенные таким образом подсистемы могут образовывать 4-х, 8, 16, 32 и высшие разрядности АЛУ, что можно использовать как в лабораторных работах на практических занятиях, так и при исследовательской деятельности студентами, магистрами и аспирантами.

Литература

1. Дьяконов, В. П. MATLAB и SIMULINK для радиоинженеров / Дьяконов, В. П. – М.: ДМК, 2011, 976 с.
2. Информатика 2007 / ред. А. П. Алексеев. - М.: Солон-Пресс, 2007. 89 - 92 с.
3. Черных, И. В. Моделирование электротехнических устройств в Matlab, SimPowerSystems и Simulink.– М.: ДМК, 2014, 296 с.

М.А. Пеливан, Е.О. Васюкова, А.В. Яковлев

СИСТЕМА МОДЕЛИРОВАНИЯ И ИМИТАЦИИ РЕЖИМОВ БОМБОМЕТАНИЯ

(Тамбовский государственный технический университет)

Бомбометание, применяемое с начала 19 века, активно используется и в наши дни. Эффективное бомбометание позволяет быстро и действенно подавлять силы противника и уничтожать точечные цели. Результат бомбометания зависит от нескольких параметров полета: скорости и высоты самолета и точности их выдерживания, углов пикирования и кабрирования, а также от способов бомбометания. Существует большое количество методов бомбометания, однако на сегодняшний день активно применяются следующие: горизонтальное метание, метание с пикированием и метание с кабрированием (рис. 1).



Рисунок 1 – Методики бомбометания: а) горизонтальное, б) с пикированием, в) с кабрированием

Эффективность поражения цели можно повысить, проведя многочисленные испытания и выявив оптимальные характеристики полета при сбрасывании авиабомб. Однако стоимость одного испытательного вылета требует больших материальных затрат, в связи с чем, сбор статистики превращается в крайне дорогостоящую процедуру. Для решения этой проблемы служит система моделирования и имитации режимов бомбометания. Система вычисляет координаты падения авиабомбы на основе входных значений: скорости и высоты полета, точности их выдерживания, угла пикирования или кабрирования, метода бомбометания, скорости ветра и параметров авиабомбы. Подавая в качестве входных параметров наборы с различными значениями величин, можно получить статистические данные, выявить зависимости и выбрать оптимальные параметры полета при различных погодных условиях, методах бомбометания и видах бомб.

Кроме определения оптимальных параметров для различных методов бомбометания и условий полета, важной задачей является отработка навыков