

Н. Семенов

MATLAB для DSP. Моделирование приемника DTMF. Часть 2

Заполнение блока "Notch filter"

Построим два одинаковых блока режкторных фильтров. Модель одного из них представлена на рис. 1.

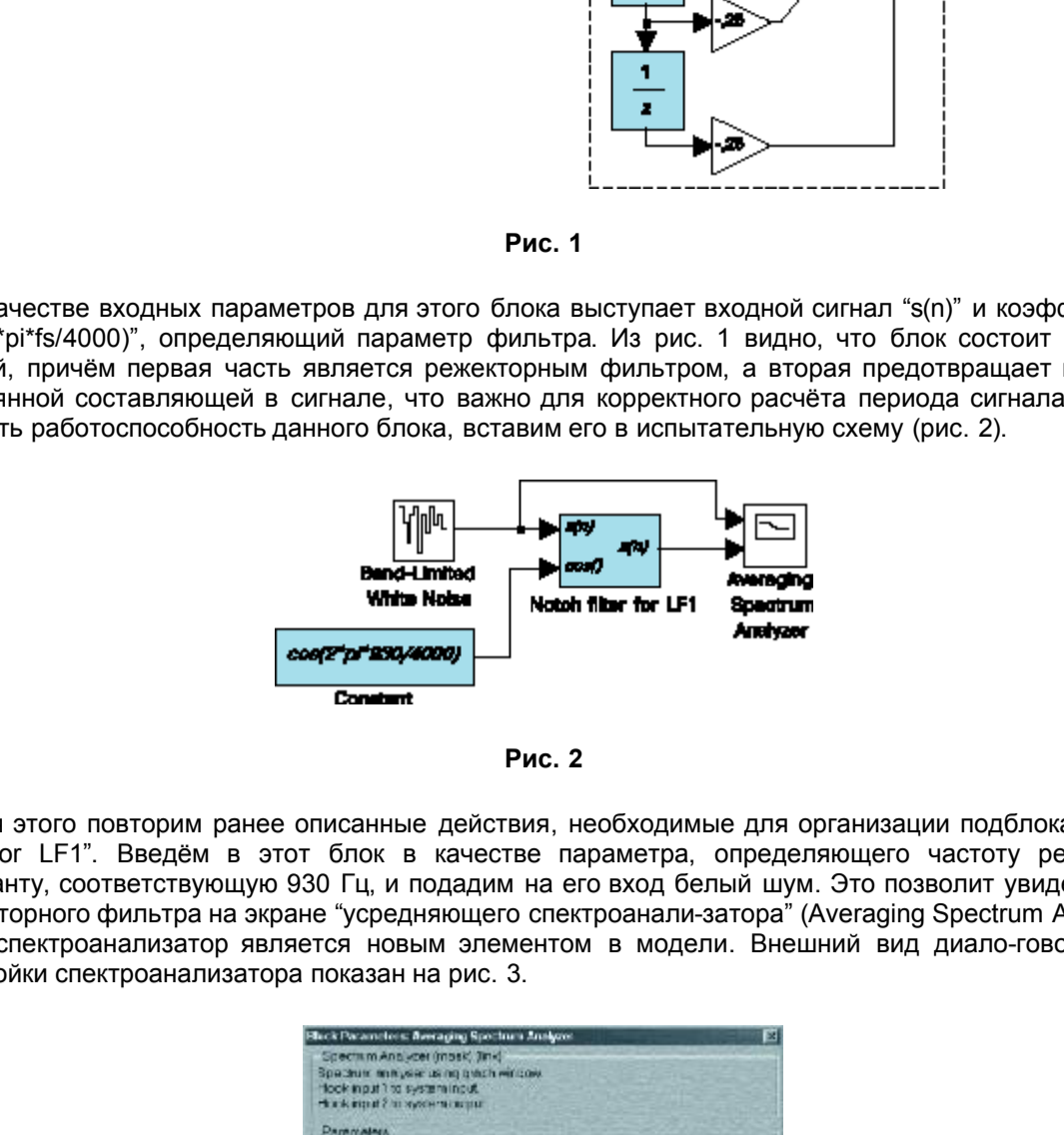


Рис. 1

В качестве входных параметров для этого блока выступает входной сигнал "s(n)" и коэффициент "cos(2*pi*f_s/4000)", определяющий параметр фильтра. Из рис. 1 видно, что блок состоит из двух частей, причем первая часть является режкторным фильтром, а вторая предотвращает наличие постоянной составляющей в сигнале, что важно для корректного расчёта периода сигнала. Чтобы оценить работоспособность данного блока, вставим его в испытательную схему (рис. 2).

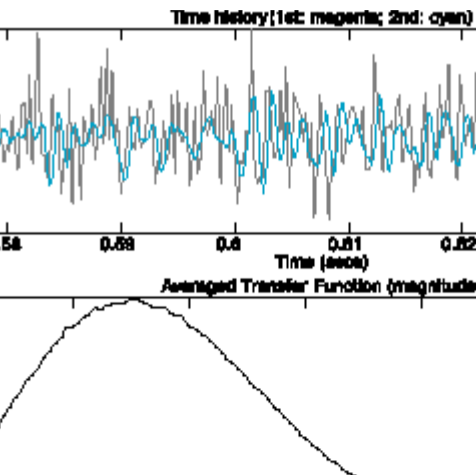


Рис. 2

Для этого повторим ранее описанные действия, необходимые для организации подблока "Notch filter for LF1". Введём в этот блок в качестве параметра, определяющего частоту режкции, константу, соответствующую 930 Гц, и подадим на его вход белый шум. Это позволит увидеть АЧХ режкторного фильтра на экране "усредняющего спектрального анализатора" (Averaging Spectrum Analyzer). Этот спектральный анализатор является новым элементом в модели. Внешний вид диалогового окна настроек спектрального анализатора показан на рис. 3.

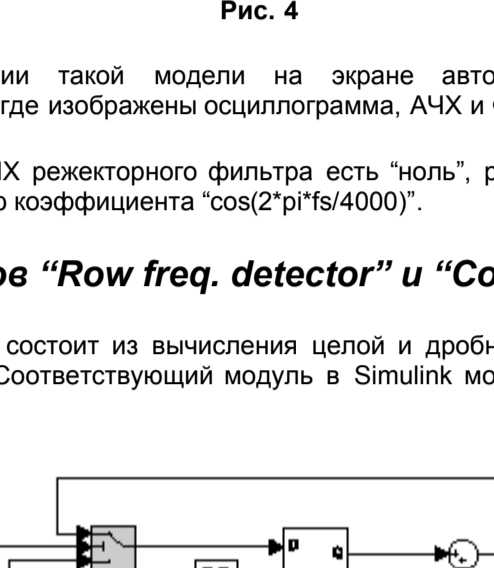


Рис. 3

В качестве параметров настройки необходимо указать длину буфера, определяющую число отсчётов сигнала, отображаемого на экране, число точек для БТФ, шаг (число отсчётов) обновления окна и интервал дискретизации.

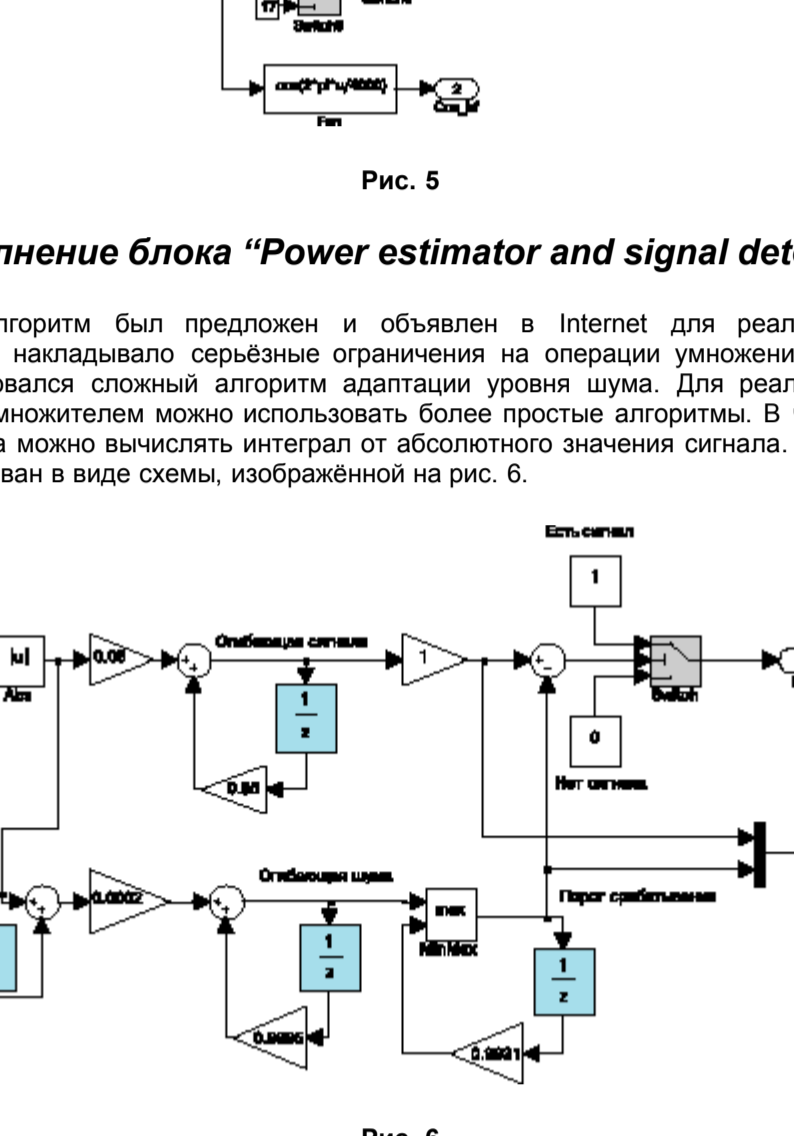


Рис. 4

После запуска симуляции такой модели на экране автоматически появится окошко спектрального анализатора (рис. 4), где изображены осциллограмма, АЧХ и ФЧХ сигнала.

Как видно из рис. 4, у АЧХ режкторного фильтра есть "ноль", расположенный на вырезаемой частоте, заданной с помощью коэффициента "cos(2*pi*f_s/4000)".

Заполнение блоков "Row freq. detector" и "Column freq. detector"

Подсчёт частоты сигнала состоит из вычисления целой и дробной частей периода сигнала в интервалах дискретизации. Соответствующий модуль в Simulink может быть реализован так, как показано на рис. 5.

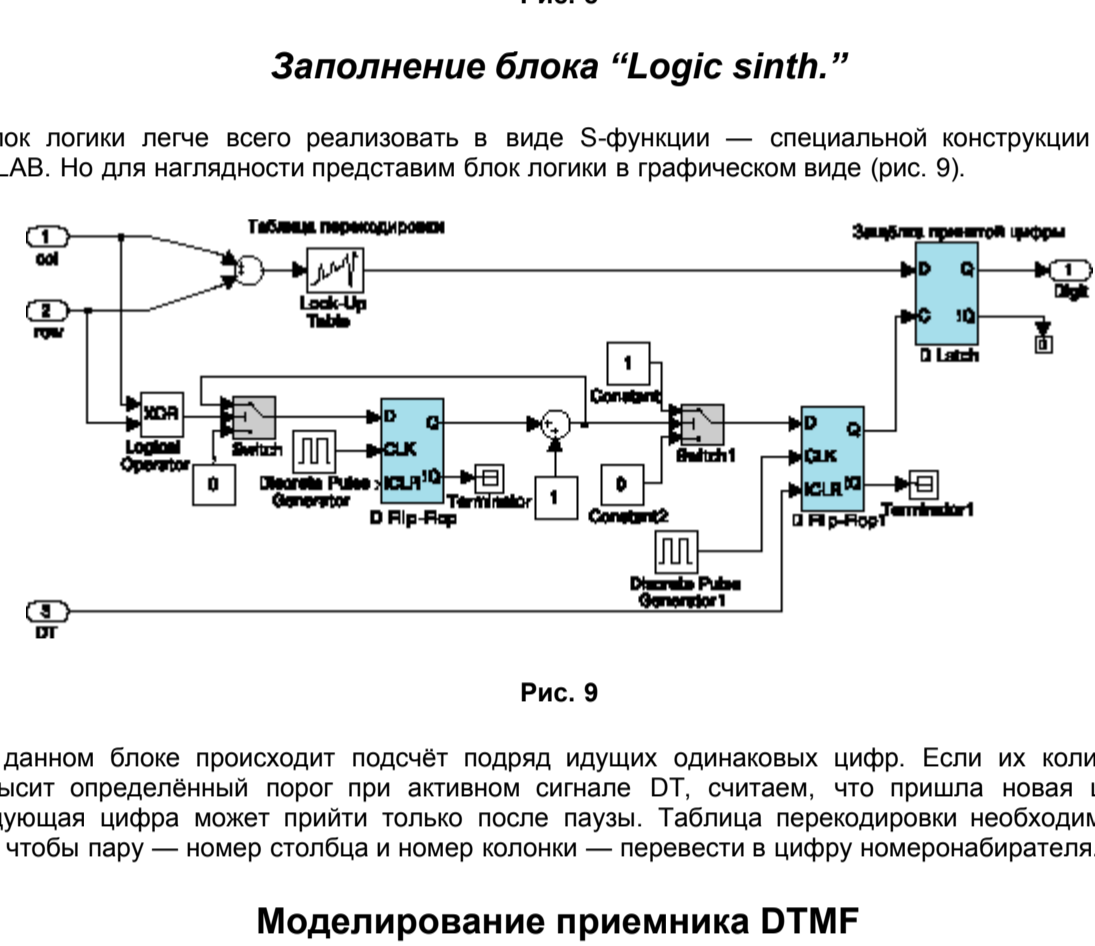


Рис. 5

Заполнение блока "Power estimator and signal detector"

Изначально алгоритм был предложен и объявлен в Internet для реализации на 8-бит процессорах, что накладывало серьёзные ограничения на операции умножения с накоплением, поэтому использовался сложный алгоритм адаптации уровня шума. Для реализации на 16-бит ЦПСОС с 32-бит умножением можно использовать более простые алгоритмы. В частности, вместо мощности сигнала можно вычислять интеграл от абсолютного значения сигнала. Соответствующий алгоритм реализован в виде схемы, изображённой на рис. 6.

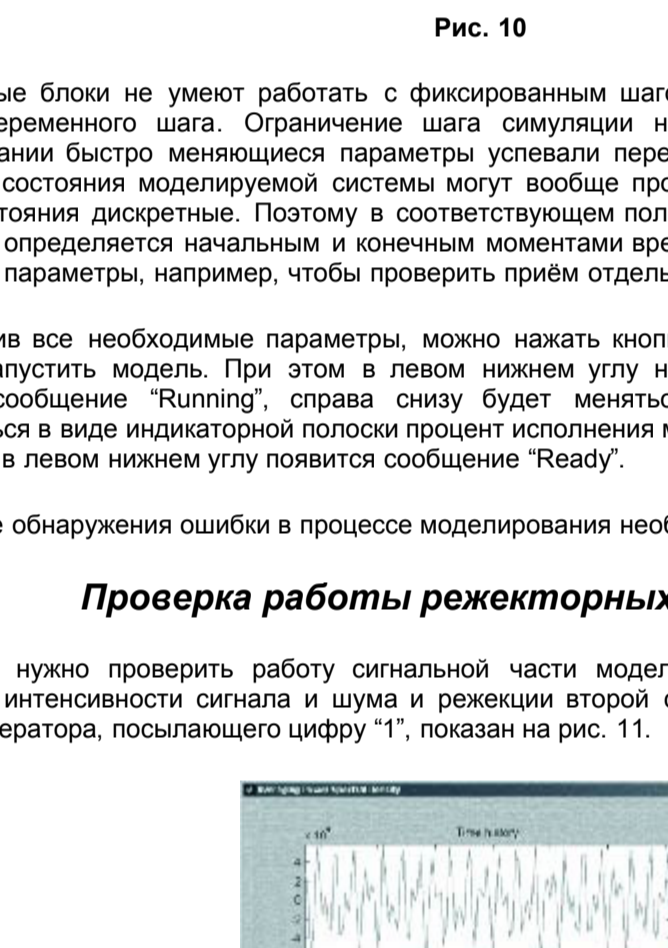


Рис. 6

В этом блоке (рис. 6) сначала происходит вычисление абсолютного значения сигнала (блок "Abs"), потом полученные значения фильтруются с помощью рекурсивного фильтра НЧ, результат сравнивается с порогом срабатывания, и принимается решение: 1 — сигнал DT активен, 0 — неактивен. Работу этого блока можно проверить, подав на него сигнал (рис. 7).

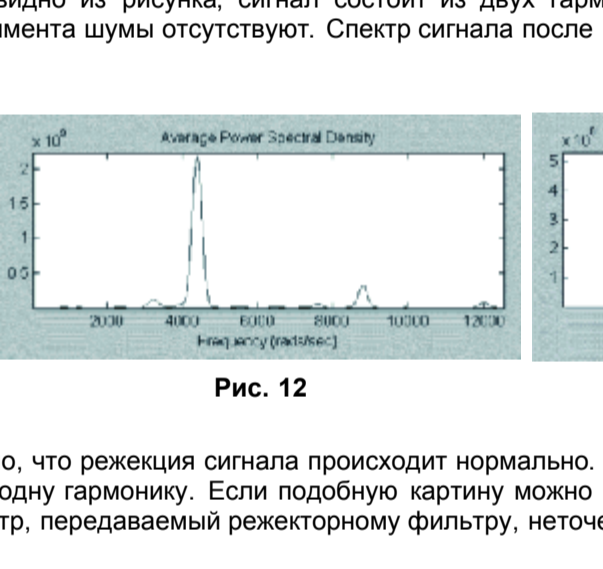


Рис. 7

Такая тестовая схема позволяет увидеть как сам сигнал, так и результат работы тестируемого блока. На рис. 8 показано, что можно увидеть на экране осциллографа при данных настройках.

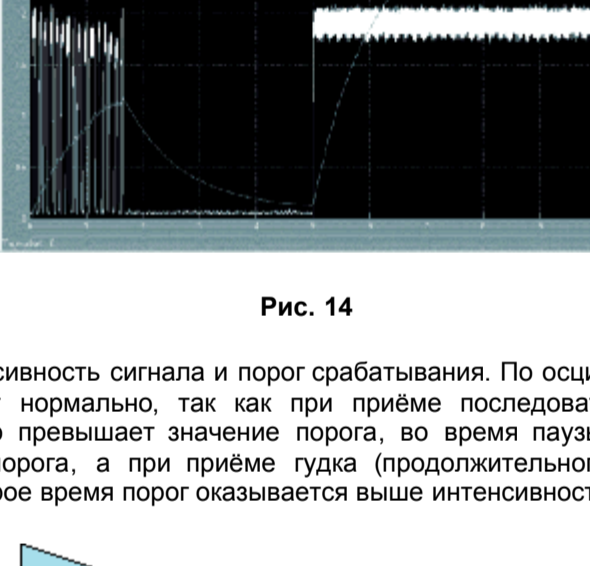


Рис. 8

Заполнение блока "Logic synth."

Блок логики легче всего реализовать в виде S-функции — специальной конструкции языка MATLAB. Но для наглядности представим блок логики в графическом виде (рис. 9).

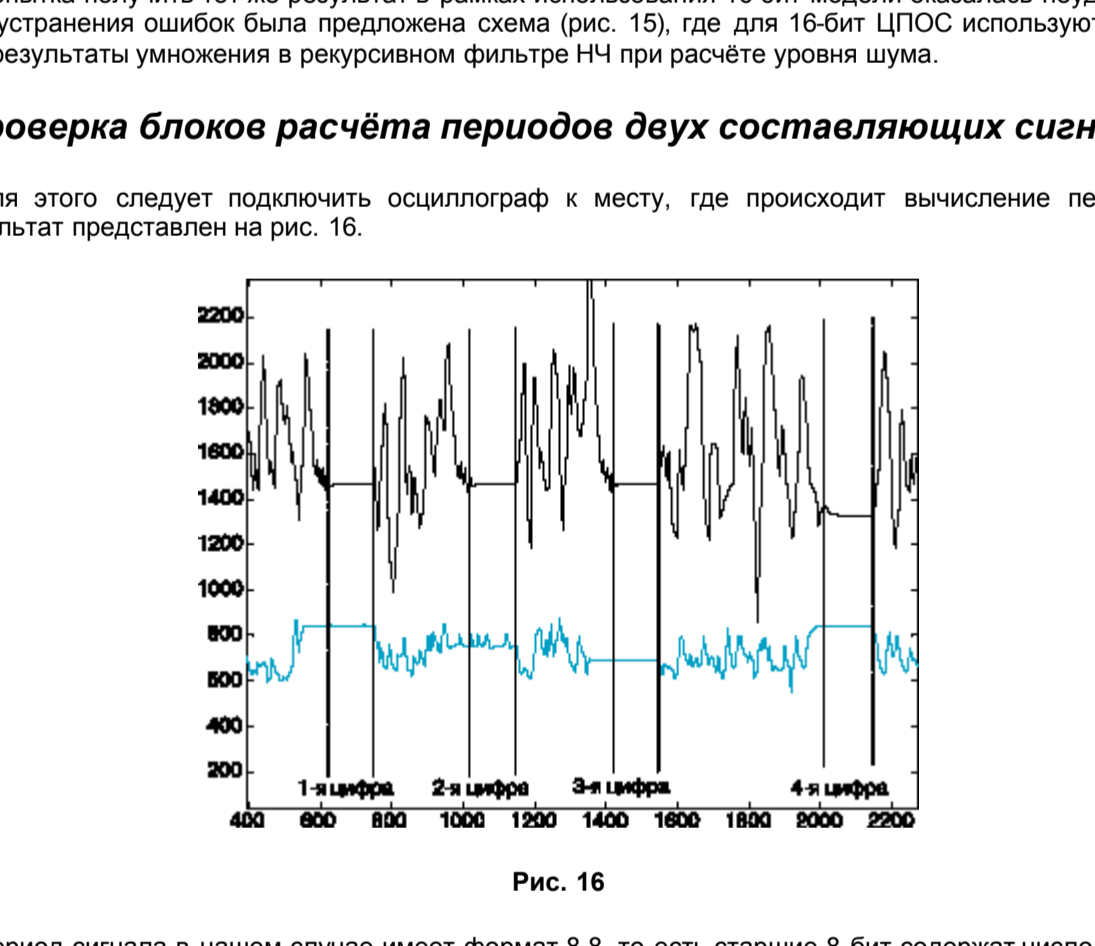


Рис. 9

В данном блоке происходит подсчёт подпара идущих одинаковых цифр. Если их количество превышает определённый порог при активном сигнале DT, считаем, что пришла новая цифра. Следующая цифра может прийти только после паузы. Таблица перекодировки необходима для того, чтобы пару — номер столбца и номер колонки — перевести в цифру номеронабрателя.

Моделирование приемника DTMF

Настройка симулятора

После того, как вся модель построена и скомпонована, можно запустить симуляцию и проследить на разных стадиях весь процесс детектирования цифры. Перед запуском симуляции модели необходимо задать параметры симуляции, к которым относятся время, способ и максимальный шаг симулятора, если выбран не фиксированный шаг. Пример диалогового окна настройки показан на рис. 10.

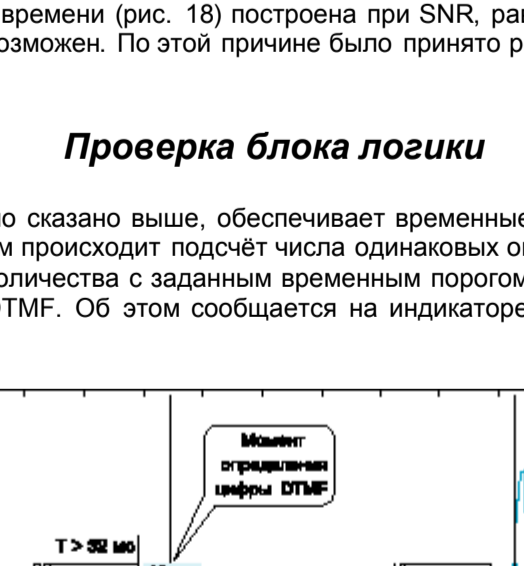


Рис. 10

Некоторые блоки не умеют работать с фиксированным шагом симулятора, поэтому вводится понятие переменного шага. Ограничение шага симуляции необходимо для того, чтобы при моделировании быстро меняющиеся параметры успевали пересчитываться, в противном случае некоторые состояния моделируемой системы могут вообще пропущаться симулятором. В нашем случае состояние индикатора. Поэтому в соответствующем поле стоит значение "discrete". Время симуляции определяется начальным и конечным моментами времени. В процессе отладки удобно менять эти параметры, чтобы проверить работу отдельной цифры.

Установив все необходимые параметры, можно нажать кнопку "Start/Pause Simulation" и таким образом запустить модель. При этом в левом нижнем углу на полосу статуса окошка модели появится сообщение "Running", справа снизу будет меняться текущее значение времени и отображаться в виде индикаторной полоски процент исполнения моделирования. После завершения симуляции в левом нижнем углу появится сообщение "Ready".

В случае обнаружения ошибки в процессе моделирования необходимо определить их источники.

Проверка работы режкторных фильтров

Сначала нужно проверить работу сигнальной части модели — блоков генерации сигнала, изменения интенсивности сигнала и шума и режкции первой составляющей. Спектр сигнала на выходе генератора, посылающего цифру "1", показан на рис. 11.

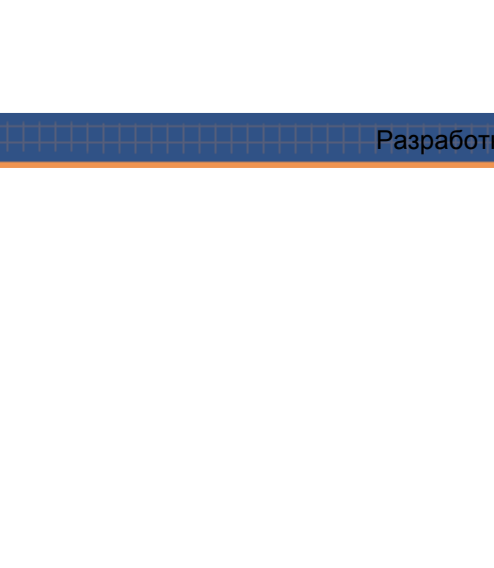


Рис. 11

Как видно из рисунка, сигнал состоит из двух гармоник одинаковой амплитуды. Для частоты эквивалентной шуму отсутствует. Спектр сигнала после режкторных фильтров показан на рис. 12 и 13.

Рис. 12

Рис. 13

Видно, что режкция сигнала происходит нормально. На выходах остаются сигналы, содержащие только одну гармонику. Если подобную картину можно видеть не для всех принимаемых цифр, то параметр, передаваемый режкторному фильтру, неточен или рассчитан неправильно.

Проверка работы детектора наличия сигнала

Детектор наличия сигнала, принимающий решение о том, есть сигнал или нет, должен адаптироваться к интенсивности сигнала и шума. Чтобы проверить положение порога срабатывания, можно вывести уровень шума и интенсивности сигнала на осцилло-граф. Пример того, что может получиться, показан на рис. 14.

Рис. 14

Здесь показаны интенсивность сигнала и порог срабатывания. По осциллограмме можно заметить, что этот блок работает нормально, так как при приёме последовательности цифр значение интенсивности устойчиво превышает значение порога, во время паузы значение интенсивности строго ниже значения порога, а при приёме гудка (продолжительного отрезка гармонического колебания) через некоторое время порог оказывается выше интенсивности сигнала.

Рис. 15

Попытка получить тот же результат в рамках использования 16-бит модели оказалась неудачной. Для устранения ошибок была предложена схема (рис. 15), где для 16-бит ЦПСОС используются 32-бит результаты умножения в рекурсивном фильтре НЧ при расчёте уровня шума.

Проверка блоков расчёта периода двух составляющих сигнала

Для этого следует подключить осциллограф к месту, где происходит вычисление периода. Результат представлен на рис. 16.

Рис. 16

Период сигнала в нашем случае имеет формат 8,8, то есть старшие 8 бит содержат число целых интервалов дискретизации, а младшие — его дробную часть. Видно, что за период измерения сигнал 4 раза стабилизировался около определённых частот. И каждый раз принимались разные цифры. Для проверки вычислим частоты, соответствующие первой цифре:

$$f_s = \frac{f_0}{k} = \frac{4000}{840} \cdot 256 = 1219 \text{ Гц}$$

— для верхней частоты,

$$f_s = \frac{f_0}{k} = \frac{4000}{1459} \cdot 256 = 701 \text{ Гц}$$

— для нижней частоты.

Такой результат в 697±10 Гц. Допустимый стандарт совпадает с тем, что было послано (1209±18 и 697±10 Гц). То есть можно утверждать, что расчёт периода сигнала происходит правильно.

Теперь по полученным значениям производится проверка на попадание значения периода сигнала в один из заданных интервалов для того, чтобы решить, какой номер строки и столбца соответствует заданному интервалу. Часть модели, отвечающая за проверку допустимых границ периода сигнала, показана на рис. 17. У каждого переключателя в качестве параметра указан порог срабатывания. Если период оказывается больше порога, на выход подаётся сигнал с верхнего входа, а если меньше, то с нижнего.

Рис. 17

В спецификации на DTMF сказано, что сигнал с отклонениями 1,5% по частоте нужно принимать, а с отклонениями больше 3% принимать не следует. Допуски периода сигнала необходимо выбирать так, чтобы приёмник соответствовал спецификации. Первоначальный выбор допуска в 1,5% оказался слишком жестким, что видно из рис. 18.

Рис. 18

Зависимость периода от времени (рис. 18) построена при SNR, равном 15 дБ. В таких условиях уверенно приём цифр невозможен. По этой причине было принято решение увеличить допуски по периоду до 3%.

Проверка блока логики

Блок логики, как уже было сказано выше, обеспечивает временные характеристики приёмника и общую работу. В нём происходит подсчёт числа одинаковых определённых номеров строки и столбца, сравнение этого количества с заданным временным порогом и выработка решения о том, что пришла новая цифра DTMF. Об этом сообщается на индикаторе, расположенном на верхнем уровне модели.

Рис. 19

На рис. 19 показан процесс принятия решения о том, что пришла новая цифра. Голубым цветом показаны периоды низкочастотной и высокочастотной составляющих. Синим — процесс определения цифры; вертикальная чёрная полоска — момент определения новой цифры. Такие результаты получены при обработке тестового сигнала фирмы Mitel, специально предназначенного для тестирования приёмника DTMF. Видно, что блок логики работает нормально.

Проверка всей модели

После проверки всех блоков по отдельности и коррекции неточностей можно снова запустить модель и убедиться, что она работает. Теперь остаётся только проверить работу модели в рабочих условиях, то есть при отклонениях частот, амплитуды, наличия шумов и так далее в том же самый важный тест — несрабатывание приемника на речь.

Так как моделирование должно полностью повторять параметры моделируемой системы, необходимо добавить округление после каждой операции умножения и проверки на допустимые границы после каждой операции сложения. Блоки усиления сигнала и сумматоры имеют встроенную возможность входить в состояние "насыщения" при достижении определённого значения. Для этого в их параметрах необходимо активировать опцию "Saturate on integer overflow". Для округления можно использовать блок из пакета Simulink, называемый "Rounding function". В его параметрах следует указать, какую функцию округления использовать: округление (round), усечение (floor) или другую.

Заключение

Как видно из вышеописанного, пакет Simulink позволяет реализовать довольно сложный алгоритм, разбить его на блоки и анализировать их как все вместе, так и по отдельности. Наглядность представления модели в Simulink позволяет легко менять структуру, "обезьянать" модель различными "датчиками" — осциллографами, спектральными анализаторами и индикаторами. Процесс отладки становится красивым увлекательным занятием, результаты наглядны и однозначны. Для точного соответствия вычислений в Simulink и в ЦПСОС, в модель вводятся элементы, обеспечивающие моделирование в формате с фиксированной запятой. Например, у результатов умножения можно отбрасывать дробную часть, а у сумматоров включить опцию "Integer overflow", то есть режим переполнения 16-разрядного аккумулятора. Построенная с помощью Simulink, модель приёмника DTMF уверенно принимает тестовый сигнал, удовлетворяет требованиям ITU-T и может являться пособием для создания подобных устройств на ЦПСОС и микроконтроллерах. Данная модель была создана сначала на языке MATLAB (m-file), для удобства настройки параметров по блокам была смоделирована в Simulink, после чего её перенесли на DSP. Она действительно работает на 16-разрядном процессоре, выполняя все спецификации ITU-T.

В заключение автор считает своим долгом поблагодарить руководителя проекта "MATLAB для DSP" проф. Ланца А.А. за постановку задачи и помощь в выполнении данной работы, фирму «SoftLine», предоставившую для этой работы лицензионную версию программы MATLAB v5.3, а также инженера «Лонисис» Гусеву М.Н. за помощь в тестировании модели на 16-бит DSP фирмы «Analog Devices».

Литература

1. Анохин В.В., Ланца А.А. MATLAB для DSP. Моделирование аналого-цифрового преобразования. Часть 1 // Chip News. — 2000. — № 2. — С. 3–7.
2. Анохин В.В. MATLAB для DSP. Моделирование аналого-цифрового преобразования. Часть 2 // Chip News. — 2000. — № 3. — С. 26–29.

E-mail: nick@rts.lonis.ru