



Кафедра конструирования
и производства радиоаппаратуры

А.С. Шостак, И.И. Горелкин

ОСОБЕННОСТИ ПРИМЕНЕНИЯ САПР SYSTEM VUE

Учебно-методическое пособие по дисциплине:

«ПРИЕМО-ПЕРЕДАЮЩИЕ МОДУЛИ СВЕРХВЫСОКОЙ ЧАСТОТЫ»

предназначено для студентов специальности «11.03.03» и других
специальностей радиотехнического профиля

Томск 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

А.С. Шостак, И.И. Горелкин

ОСОБЕННОСТИ ПРИМЕНЕНИЯ САПР SYSTEM VUE

Учебно-методическое пособие по дисциплине:

«ПРИЕМО-ПЕРЕДАЮЩИЕ МОДУЛИ СВЕРХВЫСОКОЙ ЧАСТОТЫ»

предназначено для студентов специальности «11.03.03», но может быть
использовано и студентами других специальностей радиотехнического профиля
при изучении различных систем автоматического проектирования
радиоаппаратуры

Рецензент: профессор кафедры КИПР ТУСУР, д. т. н. Масалов Е.В.

Технический редактор: доцент кафедры КИПР ТУСУР, к. т. н. Кривин Н.Н.

А.С. Шостак, И.И. Горелкин, Особенности применения САПР System Vue.
Учебно-методическое пособие по дисциплине: «Приемо-передающие модули
сверхвысокой частоты».

Томск: Томский государственный университет систем управления и
радиоэлектроники, 2019 –77с.

Пособие посвящено наиболее мощно развивающейся области моделирования – структурно-функциональному моделированию. Упор сделан на приложения, связанные с радиоэлектроникой. В настоящем пособии рассмотрены наиболее распространенные программные продукты, использующиеся на следующих двух основных этапах проектирования радиоэлектронного устройства: моделирования структурно-функциональной схемы и моделирования принципиальных электрических схем. Основное внимание уделено применению САПР System Vue для проектировании радиоаппаратуры.

© А.С. Шостак, И.И. Горелкин

©Кафедра КИПР Томского государственного
университета систем управления и
радиоэлектроники, 2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 ПРИМЕНЕНИЕ СТРУКТУРНО-ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ В РАДИОТЕХНИКЕ.....	7
2 ПРОГРАММНЫЕ ПАКЕТЫ МОДЕЛИРОВАНИЯ СТРУКТУРНО- ФУНКЦИОНАЛЬНОЙ СХЕМЫ.....	9
3 МОДЕЛИРОВАНИЕ ПРИНЦИПИАЛЬНЫХ ЭЛЕКТРИЧЕСКИХ СХЕМ.....	20
4 ПРОГРАММНЫЕ ПАКЕТЫ СОЗДАНИЯ И ПРОВЕРКИ SPICE МОДЕЛЕЙ ЭЛЕКТРОННЫХ КОМПОНЕНТОВ	26
5 ПРОГРАММНЫЕ ПАКЕТЫ МОДЕЛИРОВАНИЯ ПРИНЦИПИАЛЬНЫХ ЭЛЕКТРИЧЕСКИХ СХЕМ	29
6 ЛАБОРАТОРНЫЕ РАБОТЫ В СРЕДЕ SYSTEM VUE	34
ЛАБОРАТОРНАЯ РАБОТА 1. ГЕНЕРИРУЕМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ БИТОВ	34
ЛАБОРАТОРНАЯ РАБОТА 2. ФОРМИРОВАНИЕ СИМВОЛОВ	43
ЛАБОРАТОРНАЯ РАБОТА 3. КОД УОЛША	46
ЛАБОРАТОРНАЯ РАБОТА 4. ПРОЕКТИРОВАНИЕ ФИЛЬТРА	49
ЛАБОРАТОРНАЯ РАБОТА 5. МОДУЛЯЦИЯ.....	52
ЛАБОРАТОРНАЯ РАБОТА 6. ВЧ КАНАЛ	54
ЛАБОРАТОРНАЯ РАБОТА 7. ДЕМОДУЛЯТОР.....	60
ЛАБОРАТОРНАЯ РАБОТА 8. ПОЛНАЯ СИСТЕМА	66
7 ЗАКЛЮЧЕНИЕ	75
СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ.....	76

ВВЕДЕНИЕ

Структурно-функциональное моделирование – вид моделирования, при котором моделями являются блок-схемы, графики, чертежи, диаграммы, таблицы, рисунки и взаимосвязи между ними. Структурно-функциональное моделирование берет свои истоки в основах электрических цепей и радиотехнике.

Впервые понятие "структурный анализ" было использовано в конце 60-х годов прошлого столетия профессором Массачусетского технологического института Дугласом Россом, который разработал основы современных методов структурно-функционального анализа моделирования сложных систем. Методология структурно-функционального моделирования и анализа сложных систем (SADT) широко использовалась для эффективного решения целого ряда таких проблем, как:

- управление финансами и материально-техническим снабжением крупных фирм;
- разработка программного обеспечения АСУ телефонными сетями;
- долгосрочное и стратегическое планирование деятельности фирм;
- проектирование вычислительных систем и сетей и др.

Методы структурно-функционального моделирования основаны на следующих принципах:

- расчленение систем на части типа "черные ящики";
- иерархическая организация этих "черных ящиков";
- использование графических средств для определения функциональных связей между блоками.

Вкратце рассмотрим эти принципы. Удобство применения принципа "черного ящика" при изучении систем заключается в том, что нет необходимости знать внутреннее устройство каждого отдельного блока, следует лишь знать набор его входов и выходов, а также назначение, т.е. функцию, согласно которой происходит преобразование входного сигнала в выходной.

Разбиение на «черные ящики» должно удовлетворять следующим критериям:

- каждый "черный ящик" реализует единственную функцию системы;
- функция каждого "черного ящика" является легко понимаемой независимо от сложности ее внутренней структуры;
- связь между "черными ящиками" вводится только при наличии связи между соответствующими функциями системы;
- связи должны быть простыми, с минимальным количеством перекрестных и обратных связей между составными блоками, для обеспечения, насколько это возможно, их независимости друг от друга.

Идея, или принцип иерархии (расположение частей или элементов системы в порядке от высшего к низшему), заключается в последовательной детализации структурно-функционального описания системы – сначала система описывается как взаимодействие наиболее крупных блоков, затем каждый блок детализируется. Это значительно облегчает понимание работы сложных систем.

Третий момент, касающийся методов структурно-функционального моделирования – использование графических средств описания связей. Очевидно, что именно графическое представление взаимодействия блоков системы в полной мере реализует принципы, отмеченные выше.

1 ПРИМЕНЕНИЕ СТРУКТУРНО-ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ В РАДИОТЕХНИКЕ

Типовой цикл проектирования радиоэлектронной аппаратуры можно разбить на несколько этапов:

- разработка структурной схемы;
- разработка принципиальной схемы, включая моделирование работы;
- предварительное определение конструктивных требований;
- разработка (трассировка) печатной платы (пластины из диэлектрика, на поверхности и/или в объёме которой сформированы электропроводящие цепи электронной схемы);
- разработка корпуса и конструкции прибора в целом;
- оценка электромагнитной совместимости;
- оценка тепловых режимов;
- оценка надежности.

В современных условиях почти на всех перечисленных этапах предлагается использование специализированных программных пакетов автоматизированного проектирования электронных устройств (САПР). Поэтому основные направления и этапы проектирования можно разделить на следующие задачи:

- моделирование на уровне структурных схем;
- моделирование смешанных аналого-цифровых устройств;
- моделирование и синтез логики для ПЛИС;
- схемотехническое и электромагнитное моделирование СВЧ-устройств;
- проектирование печатных плат;
- анализ электромагнитной совместимости;
- тепловое моделирование.

Общие преимущества использования систем САПР при проектировании радиоэлектронных устройств можно сформулировать следующим образом:

- нет необходимости в экспериментальных исследованиях, для проведения которых требуется приобретение дорогостоящих измерительных приборов, радиодеталей, трудоемкая сборка и длительная настройка макетов;
- всестороннее исследование разрабатываемых устройств в различных режимах работы (например, в предельно допустимых режимах), что не всегда возможно при натурных испытаниях;
- оценка влияния статистического разброса параметров отдельных компонентов на работы системы в целом;
- исследование влияния дестабилизирующих внешних факторов.

2 ПРОГРАММНЫЕ ПАКЕТЫ МОДЕЛИРОВАНИЯ СТРУКТУРНО-ФУНКЦИОНАЛЬНОЙ СХЕМЫ

Разработка любого электронного устройства начинается с идеи, которая первоначально воплощается в виде структурной схемы. Быстро проверить работоспособность и все возможные варианты развития будущей системы можно с помощью специальных программ функционального моделирования. На функциональном уровне важно оценить поведение каждого структурного элемента системы, а также обеспечить правильное соединение элементов друг с другом.

Здесь можно порекомендовать программные пакеты – Simu Link, Sys Calc, System View, Lab View, Hyper-Signal Block Diagram, Vis Sim, VSS – которые позволяют построить моделируемую систему из «кубиков» в точной аналогии со структурной схемой.

Обычно работа таких систем представляет собой конструктор, с помощью которого из стандартных библиотечных "кубиков" строится структурная схема. В библиотеке выбирают нужный элемент, который затем переносят на схему.

Рассмотрим вкратце каждый из перечисленных программных пакетов

1. Simulink

Производитель: The Math Works, Inc.

Сайт продукта: <http://www.mathworks.com/products/simulink/> Программа Simulink является приложением (расширением) к пакету MATLAB [1]. При моделировании с использованием Simulink реализуется принцип так называемого визуального программирования, в соответствии с которым пользователь на экране монитора из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от прежних классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний, требующихся при работе на компьютере и, естественно, знаний той предметной области, в которой он работает.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны, доступ к функциям MATLAB и другим его инструментам остается открытым их можно использовать в Simulink. Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset - набор блоков для разработки цифровых устройств и т.д.).

При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, также составлять целые библиотеки новых блоков.

При моделировании пользователь может выбирать стратегию и технику моделирования, например, численный метод решения дифференциальных уравнений, способ изменения модельного времени (с фиксированным или переменным шагом) и т.д. В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения», входящие в состав библиотеки Simulink. Результаты проведенного моделирования могут быть наглядно представлены в виде графиков или таблиц.

Преимущество Simulink заключается также в том, что он позволяет пользователю самостоятельно пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на классических языках алгоритмического программирования(C ++, Fortran).

На рисунке 1 показан пример моделирования ветрогенератора в пакете Simulink.

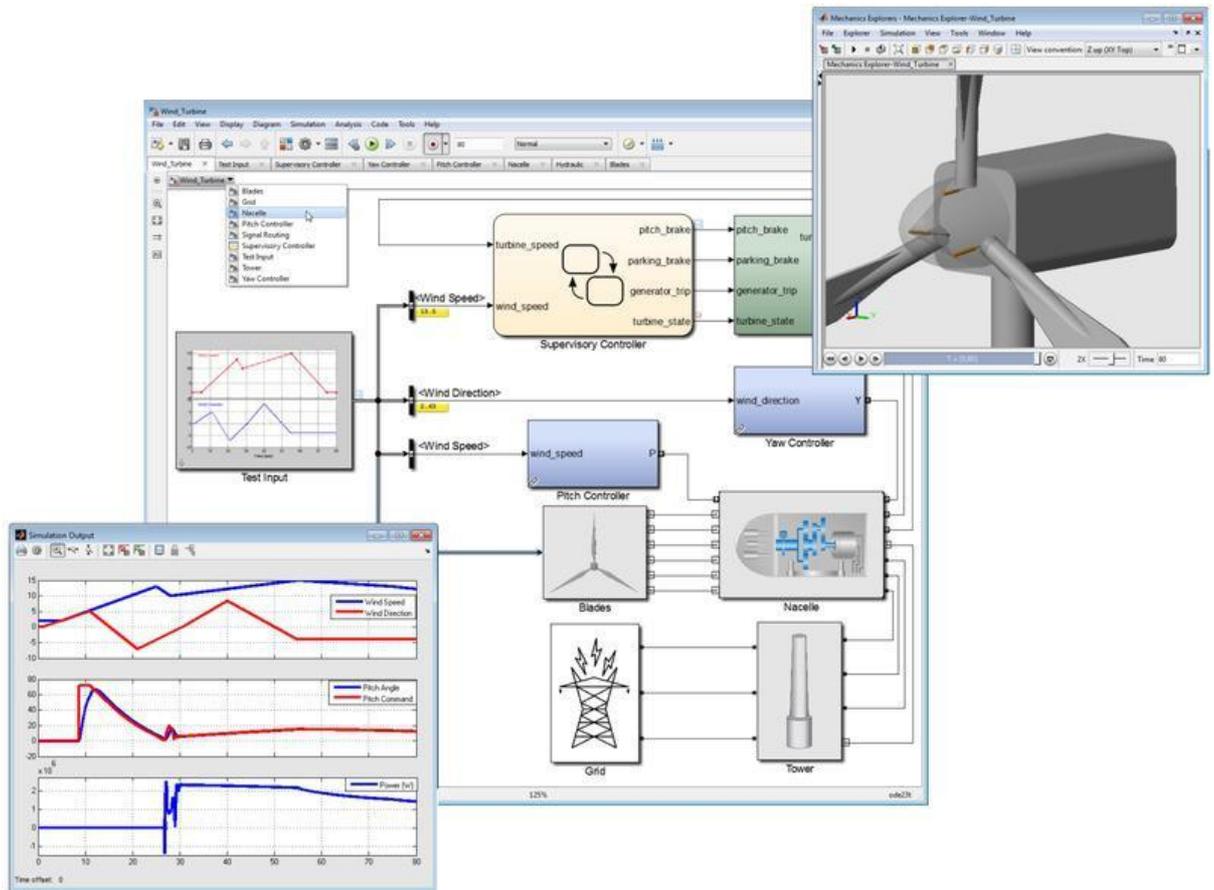


Рис. 1. Пример моделирования в Simulink

2. SysCalc

Производитель: Arden Technologies, Inc.

Сайт продукта: <http://www.ardentech.com/>

Sys Calc – это программный пакет для расчета на системно-техническом уровне различных характеристик: шумовых, коэффициента усиления, интермодуляционных характеристик и суммарного коэффициента усиления последовательности каскадно включенных компонентов, где каждый компонент описывается своими параметрами [2].

Программа Sys Calc позволяет на этапе эскизного проектирования осмысленно подбирать такие параметры системы и отдельных блоков, как динамический диапазон, чувствительность, уровни побочных гармоник, и найти их оптимальное по каскадному распределению. Используя данную программу, можно оценить, какие узлы в тракте вносят ухудшение в результирующие характеристики всего тракта, а также исчерпывающе документировать

разрабатываемые проекты. Sys Calc использует иерархические закладки, чтобы объединить компоненты в связанные «страницы» автономной системы. Страницы затем объединяются, формируя законченную систему. Каждая Системная страница может быть отдельно настроена, чтобы отображать только важную для пользователя информацию и поля данных. Число компонентов, которые могут быть помещены в страницу или число страниц, которые могут быть созданы в проекте, произвольно. Вы можете также присоединять один или большее количество Отчетов (таблиц и графиков) к каждой системной странице. Система отчетов программы SysCalc позволяет производить следующие виды анализа: моделирование фазового шума, анализ линейности и ее диапазон, оценка энергетического потенциала канала связи, анализ побочного электромагнитного излучения и т.п. (см. рисунок 2).

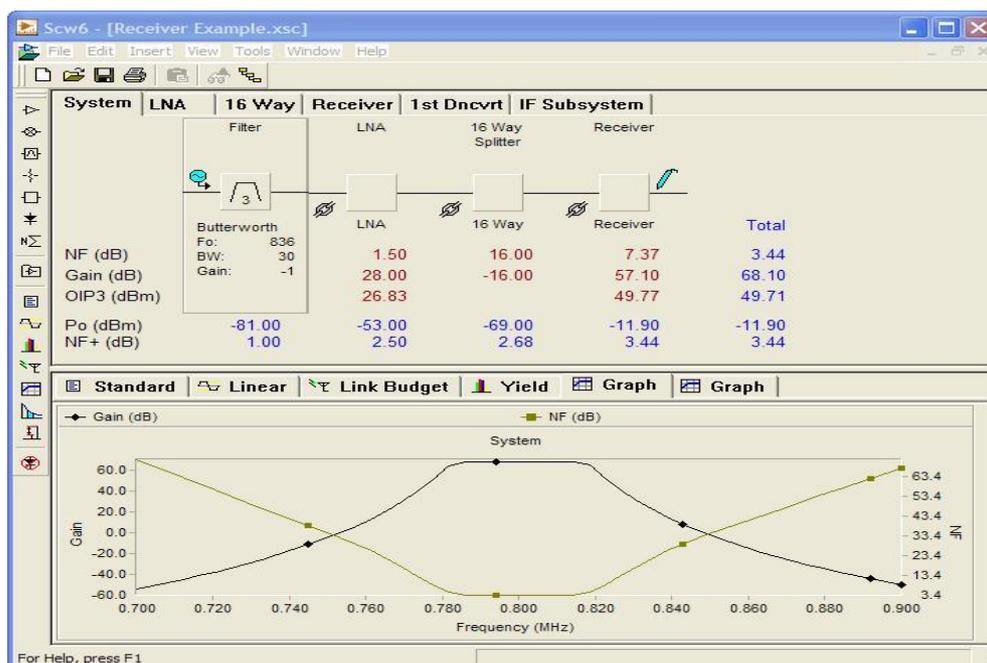


Рис. 2. Пример моделирования в SysCalc

3. SystemVue

Производитель: Agilent Technologies

Сайт продукта: <http://www.agilent.com/find/eesof-systemvue> SystemVue является специализированным программным средством автоматизированного проектирования электронных устройств на системном уровне. System Vue позволяет разработчиками радиосистем проектировать физический уровень

наземных беспроводных и аэрокосмических оборонных систем связи и имеет ценность для разработчиков, программирующих алгоритмы работы с помощью цифровых сигнальных процессоров и ПЛИС [3].

Платформа System Vue предоставляет собой простую в использовании среду с современными технологиями моделирования, возможностью подключения к реальной измерительной аппаратуре и проведения испытаний. Она позволяет создавать алгоритмы работы и прототипы архитектур для сложных систем связи. System Vue имеет интуитивно понятный блочный интерфейс среды разработки с обширными библиотеками блоков для построения коммуникационных систем, элементов адаптивного управления и компонентов цифровой обработки сигналов. System Vue поддерживает алгоритмы разработки прототипов и реализации через генерацию VHDL кода для ПЛИС (FPGA) и ANSI C кода для встраиваемых цифровых сигнальных процессоров.

На рисунке 3 приведен пример моделирования радиосистемы в пакете System Vue.

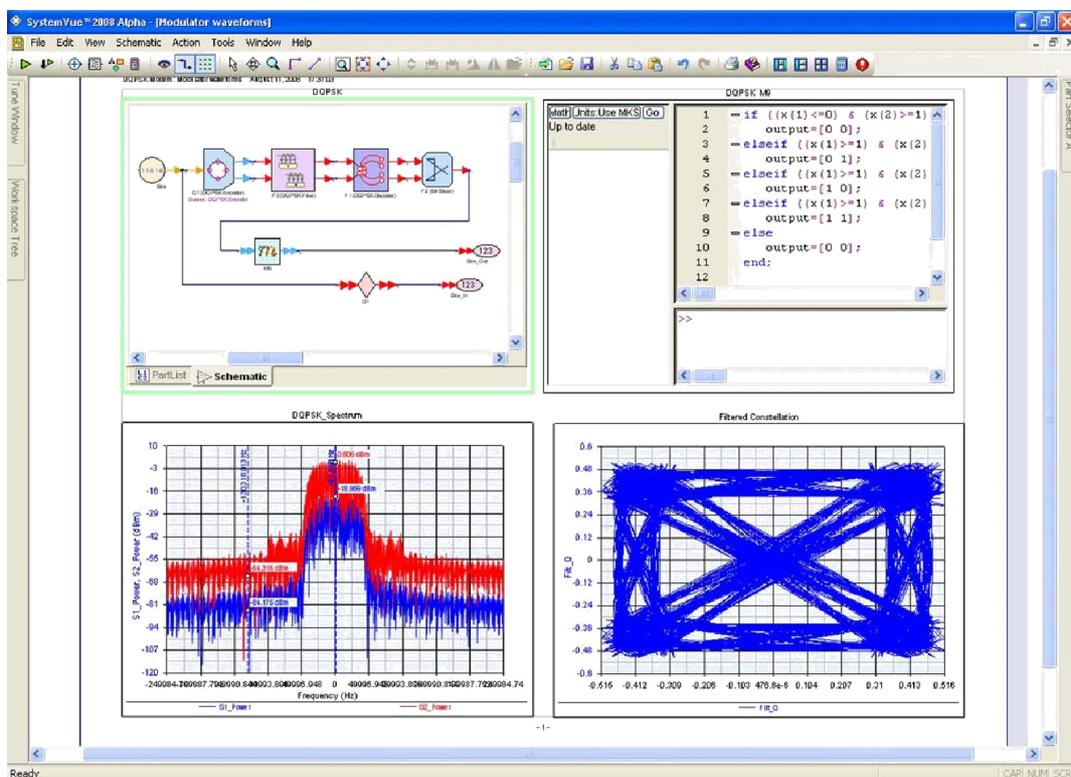


Рис. 3. Пример моделирования в SystemVue

4. Lab VIEW

Производитель: National Instruments

Сайт продукта: <http://www.labview.ru/LabVIEW> (Laboratory Virtual Instrument Engineering Workbench

– среда разработки лабораторных виртуальных приборов) – это среда прикладного графического программирования, которая используется разработчиками, инженерами, преподавателями и учеными по всему миру для быстрого создания комплексных приложений в задачах измерения, тестирования, управления, автоматизации научного эксперимента, а также в области образования.

В основе Lab VIEW лежит концепция графического программирования («графический язык G»), заключающаяся в описании системы в виде графической блок-диаграммы, состоящей из функциональных блоков и связей между ними [4]. Несмотря на графический ввод, в языке G используются те же конструкции и методы, что и в классических языках программирования: типы данных, циклы, переменные, рекурсия, обработка событий и объектно-ориентированное программирование.

Lab VIEW является идеальным программным средством для создания информационно-измерительных систем, а также систем автоматизации управления на основе технологии виртуальных приборов. Lab VIEW позволяет взаимодействовать с реальными аппаратными средствами, такими как встраиваемые в компьютер многоканальные измерительные аналого-цифровые платы, платы захвата и синхронизации видеоизображения для систем машинного зрения, платы управления движением и исполнительные механизмы, также измерительные приборы, подключаемые к компьютеру через стандартные интерфейсы RS-232, RS-485, USB, GPIB (КОП), PXI, VXI. Все это в комплексе позволяет разрабатывать системы измерения, контроля, диагностики и управления практически любой сложности.

Кроме того, Lab VIEW может интегрировать в себя программы, написанные в среде MATLAB и на языке C. Большое количество встроенных алгоритмов цифровой обработки одномерных и двумерных сигналов позволяет

осуществлять весьма сложную обработку сигнала, изображения и экспериментальных данных во временной, пространственной и спектральной областях. Программная среда Lab VIEW постоянно расширяется новыми средствами обработки сигналов на основе вейвлет-анализа, алгоритмов нечеткой логики, сетевых технологий и т. д.

Эффективность использования среды Lab VIEW в научных исследованиях состоит в том, что, оставаясь в ее рамках, можно как разрабатывать математическую модель объекта, так и снабжать эту модель экспериментальными данными с помощью аппаратных средств ввода-вывода, сопряженных с реальным объектом.

На рисунке 4 приведен пример графического моделирования в пакете Lab VIEW.

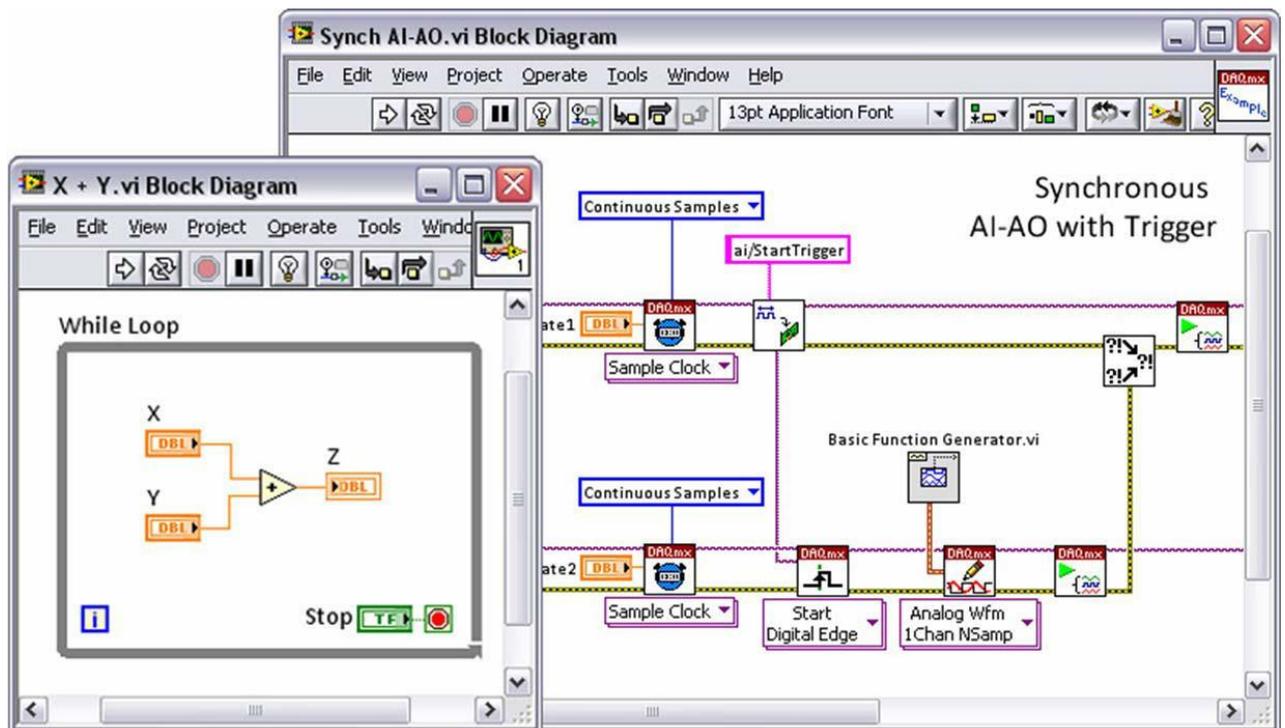


Рис. 4. Пример моделирования в Lab VIEW

5. Hypersignal RIDE (Hypersignal Block Diagram)

Производитель: National Instruments

Сайт продукта: <http://www.ni.com/support/ride.htm>

1990 г. фирма Hyperception выпустила систему автоматизированного проектирования Hypersignal Block Diagram для визуального проектирования и

моделирования на ПК сложных динамических систем с использованием вычислительных возможностей ПК. В 2003 году компания National Instruments объявила о приобретении Hyperception. Логическим развитием пакета Block Diagram стала САПР Hypersignal RIDE, которая поддерживает аппаратные средства цифровой обработки сигналов (ЦОС), установленные в ПК или связанные с ним по последовательному интерфейсу, и позволяет комбинировать в одном проекте как функции, выполняющиеся в среде процессора ПК, так и функции, выполняющиеся в среде процессора ЦОС.

В состав Hypersignal входят несколько сотен тематически сгруппированных функций-блоков [5]. Среди них: блоки генераторов сигналов, блоки арифметических функций, блоки матричных и векторных операций, блоки функций ЦОС, блоки файловых операций, блоки визуализации сигналов и так далее. В состав САПР Hypersignal также входят блоки управления: клавиатуры, переключатели, линейные и стрелочные индикаторы и так далее. Наличие этих функциональных блоков позволяет создавать пользовательский интерфейс разрабатываемой системы совместно с разработкой алгоритма функционирования. В дополнение поставляются специализированные библиотеки функций для обработки речи (Advanced Speech Library), библиотеки коммуникационных функций (Advanced Transmission Library) и библиотек и функций для обработки изображений (Image Processing Library).

Пользователям САПР Block Diagram и RIDE дополнительно предлагается генератор ANSI Си-кода. Генератор Си-кода генерирует ANSI Си-код, соответствующий "визуализованному" алгоритму, разработанному с помощью Hypersignal RIDE или Hypersignal Block Diagram. В дальнейшем этот код может быть встроен в различные программные продукты для ПК, перенесен на альтернативные UNIX-платформы или встроен с применением соответствующих средств кросс компиляции в автономные системы реального времени, построенные на базе различных процессоров.

На рисунке 5 приведен пример моделирования системы амплитудной модуляции/демодуляции в пакете Hypersignal.

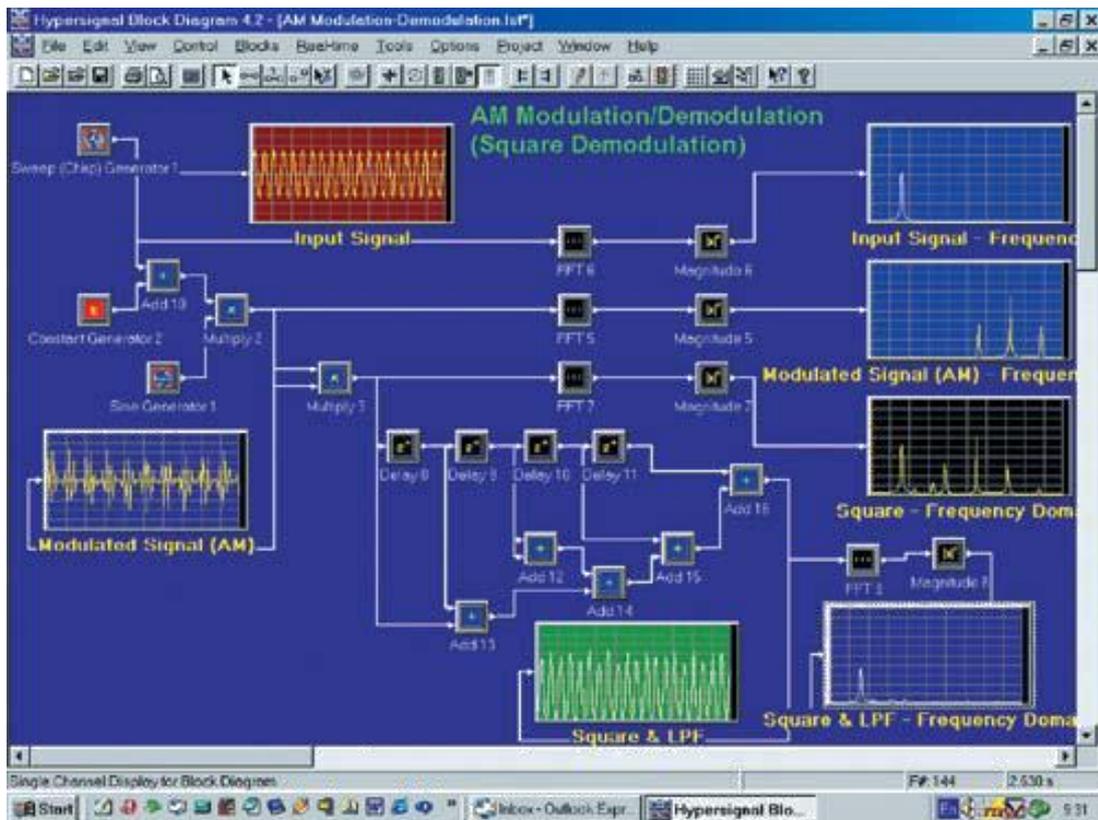


Рис. 5. Пример моделирования в Hypersignal

6. Vis Sim

Производитель: Visual Solutions

Сайт продукта: <http://www.vissim.com/>

Программа Vis Sim предназначена для построения, исследования и оптимизации виртуальных моделей физических и технических объектов, в том числе и систем управления [6]. Vis Sim

– это сокращение выражения Visual Simulator – визуальная, т.е. непосредственно воспринимаемая зрением, программная среда и средство моделирования.

Программный пакет предоставляет человеку развитый графический интерфейс, используя который исследователь создает модель из виртуальных элементов (с некоторой степенью условности и идеализации компонентов) так же, как если бы он строил реальную систему из настоящих элементов (см. рисунок 6). Это позволяет создавать, а затем исследовать и оптимизировать модели систем широкого диапазона назначения и различного уровня сложности.

При использовании Vis Sim'a не требуется владеть программированием на языках высокого уровня или на ассемблере. В то же время, специалисты, владеющие программированием, могут создавать собственные блоки, дополняя ими богатую библиотеку стандартных блоков Vis Sim'a.

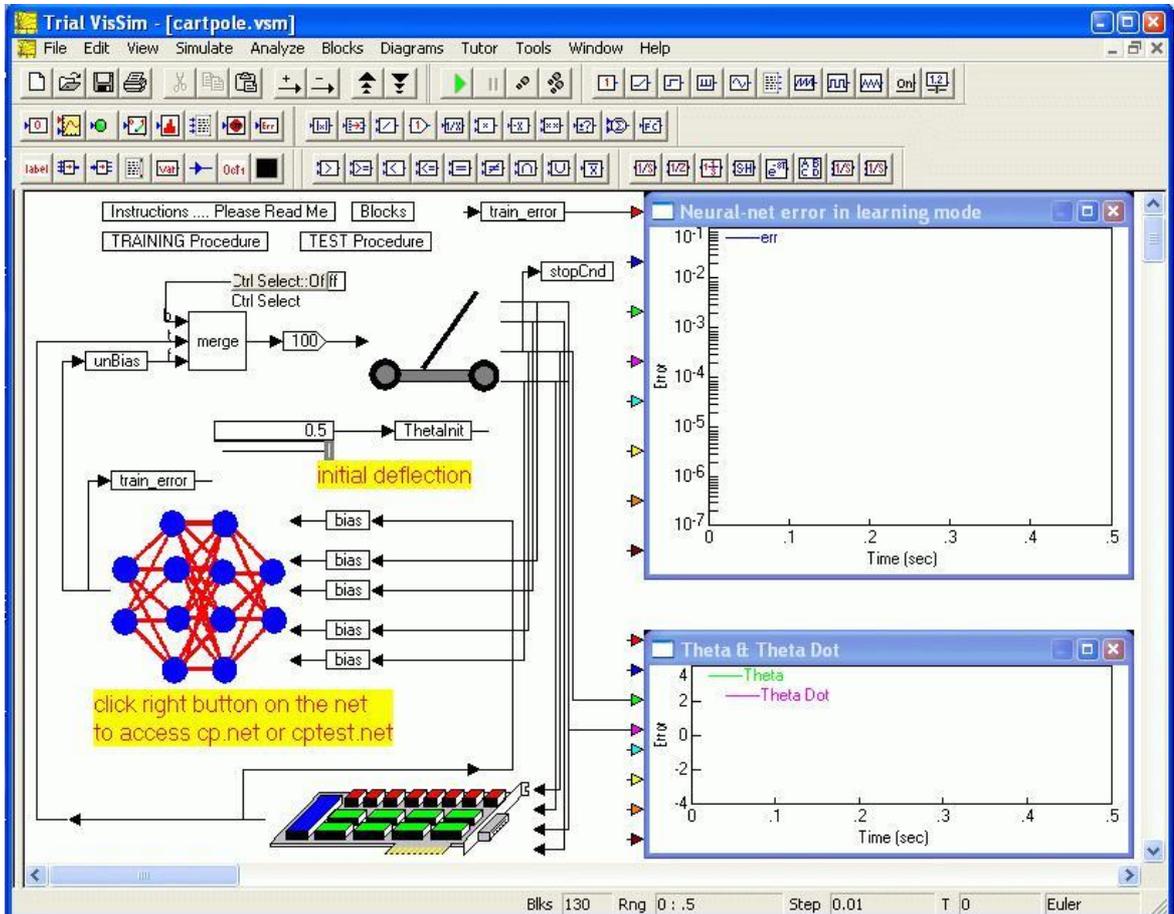


Рис. 6. Пример моделирования в VisSim

7. Visual System Simulator

Производитель: AWR Corporation

Сайт продукта: <http://www.awrcorp.com/ru/products/visual-system-simulator>

Visual System Simulator (VSS) - программа моделирования коммуникационных систем на уровне функциональных блоков.

Ключевые возможности [7]:

- интеграция схем и систем на уровне блоков;
- высокая скорость моделирования на системном уровне;
- обширная библиотека моделей, описывающих стандарты связи GSM, CDMA и др.;

3 МОДЕЛИРОВАНИЕ ПРИНЦИПИАЛЬНЫХ ЭЛЕКТРИЧЕСКИХ СХЕМ

После моделирования структурной схемы проектируются принципиальные электрические схемы самих составных блоков, осуществляются различные проверки работоспособности и выбирается элементная база.

Детальное проектирование и исследование радиоэлектронных устройств на ЭВМ на уровне принципиальных схем можно провести с помощью целого ряда специальных прикладных программных пакетов. В учебных целях и на начальных этапах работы целесообразно применять более простые программные пакеты: Milti Sim (Electronics Workbench), Micro CAP, Circuit Maker и др. Большинство этих пакетов использует данные в текстовом формате SPICE, чем обеспечивается их совместимость со многими САПР.

Для более сложных задач используются такие профессиональные пакеты, как OrCAD (PSPice A/D), P-CAD 2000-200X (ACCEL EDA), Altium Designer (Protel), Product Designer, Power PCB, CAM 350, View logic (Analog), BETA Soft, MATLAB+Simulink (Sim Electronics Toolbox) и др.

При проектировании устройств сверхвысокочастотного диапазона могут быть использованы программы Microwave Office, CST Microwave Studio, EMPro, Momentum и др.

Все современные программные продукты предполагают графический ввод проекта в редакторе принципиальных схем, после чего автоматически генерируется список схемных соединений, необходимый для работы программы моделирования. В качестве вычислительного ядра почти во всех пакетах используется программа SPICE (Simulation Program with Integrated Circuit Emphasis).

Различные версии этой программного продукта были в разное время заимствованы производителями схемотехнических САПР для использования в своих прикладных пакетах в качестве основного вычислительного ядра. Сейчас для моделирования аналоговых устройств в основном используется версия

SPICE 3, а для моделирования смешанных аналого-цифровых устройств – версия XSPICE. Рассмотрим SPICE подробнее.

3.1 Симулятор электронных схем SPICE

SPICE (Simulation Program with Integrated Circuit Emphasis) – симулятор электронных схем общего назначения с открытым исходным кодом. Является мощной программой, используемой в разработке печатных плат для проверки целостности схемы и для анализа ее поведения. В настоящее время SPICE де-факто стала эталонной программой-ядром моделирования аналоговых и аналого-цифровых электронных цепей.

Первая версия SPICE была разработана в середине 70-х годов в Electronics Research Laboratory в Калифорнийском университете в Беркли. Она позволяла анализировать линейные и нелинейные цепи во временной области, рассчитывать частотные характеристики линейных цепей. Программа SPICE 1 была целиком написана на языке FORTRAN, и для анализа линейных резистивных цепей использовался метод узловых напряжений. Библиотека моделей электронных компонентов включала модели диода и биполярного транзистора (модель Молла-Эберса).

В последующем программа SPICE совершенствовалась, появлялись новые версии, расширялся список моделей компонентов, уже в конце 70-х годов SPICE стала широко использоваться в промышленности для проектирования и моделирования электронных схем.

В 1975 году вышла вторая версия программы SPICE 2, которая так же была написана на FORTRAN, но имела больше элементов, позволяла изменять временной шаг при моделировании, а уравнения цепей формулировались при помощи модифицированного метода узловых потенциалов.

Последняя редакция второй версии SPICE 2G.6 появилась в 1983 году. В ней были переработаны модели биполярных и полевых МОП-транзисторов. Модель биполярного транзистора в версии SPICE 2G.6 основана на уравнениях Гуммеля-Пуна.

В 1985 году была разработана следующая версия программы, SPICE 3. Она базируется на версии SPICE 2G.6 и является ее развитием. Математическое ядро программы было полностью переписано на языке С. Для представления результатов моделирования стал использоваться графический интерфейс (эпюры напряжений, графики частотных характеристик и т.д.). В программу были включены программные модели новых электронных компонентов: длинные линии с потерями, неидеальные ключи и т.д. Появились новые модели МОП-транзисторов, учитывающие физические эффекты, возникающие при уменьшении геометрических размеров приборов.

В середине 80-х годов появились программы схемотехнического моделирования, предназначенные для персональных компьютеров. В 1984 г. корпорация Micro Sim представила свою версию SPICE для персональных компьютеров, назвав ее PSpice. Эта программа и все ее последующие версии используют математические модули SPICE, а также SPICE формат представления входных и выходных данных.

Первые версии PSpice компании Micro Sim позволяли моделировать только аналоговые радиоэлектронные устройства. Рассчитывались переходные процессы при действии сигналов различной формы, частотные характеристики, рабочие точки нелинейных приборов. В начале 90-х годов были созданы версии, позволяющие моделировать не только аналоговые, но и смешанные аналого-цифровые устройства. Программа получила более удобный интерфейс, обеспечивающий графический ввод схем вместо текстового.

Современные версии PSpice представляют вычислительную среду, предназначенную для моделирования аналоговых и цифровых электронных схем. Существенное достоинство программы заключается в возможности моделирования смешанных аналого-цифровых схем без применения вспомогательных программных модулей согласования аналоговых и цифровых сигналов. Это достигается за счет автоматического использования специальных интерфейсов и значительно облегчает моделирование смешанных аналого-цифровых устройств.

Процедура моделирования электронных схем в программе PSpice состоит из трёх этапов:

- Создание принципиальной схемы в графическом редакторе;
- Непосредственно моделирование работы схемы;
- Представление результатов моделирования в удобной для пользователя форме (эпюры напряжений, АЧХ и т.д.).

Базовый набор электронных элементов PSpice включает резисторы, конденсаторы, катушки индуктивности, диоды, биполярные транзисторы, полевые транзисторы (с управляющим р–п переходом и изолированным затвором), длинные линии, источники напряжения и тока, генераторы сигналов различной формы. Аналоговые интегральные схемы, а также некоторые электронные компоненты (тиристоры, некоторые виды полевых транзисторов) представляются подсхемами, параметры которых задает пользователь.

Базовый набор электронных компонентов PSpice содержит также цифровые функциональные блоки, выполняющие логические операции. Кроме того, существуют обширные библиотеки серийно выпускаемых цифровых компонентов.

Для примера на рисунке 8 показано описание модели n-канального МОП-транзистора NMOS5P0.

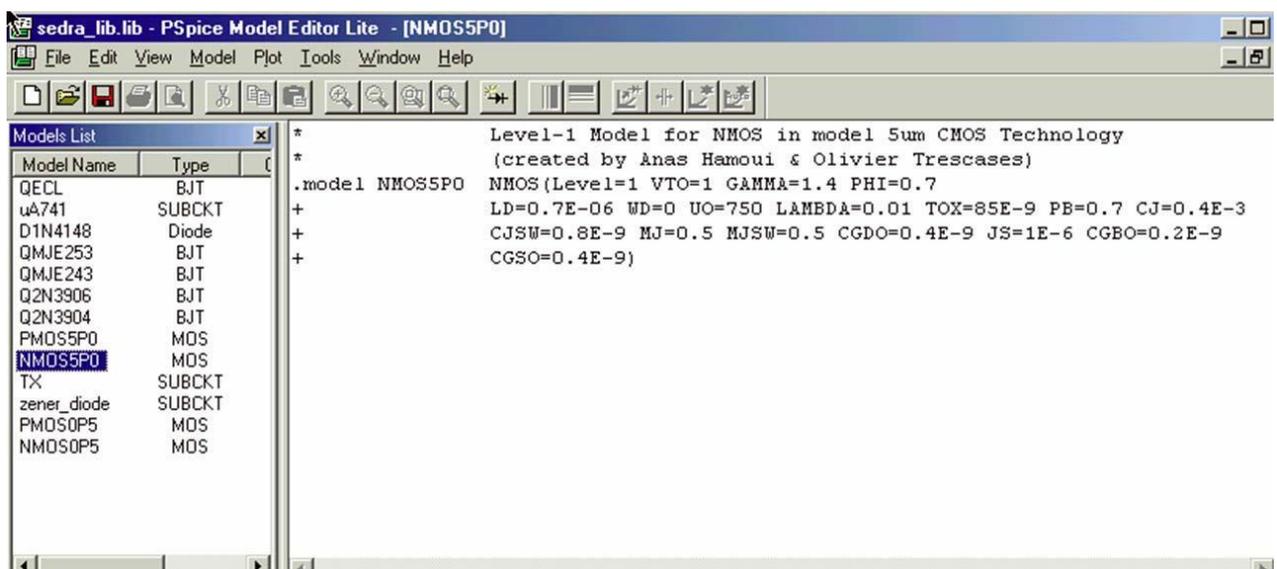


Рис. 8. Модель n-канального МОП-транзистора в PSpice

В PSpice (SPICE) возможны следующие виды анализа.

- Bias – Анализ по постоянному току для больших сигналов;
- DC (DirectCurrent) – Анализ по постоянному току для малых сигналов;
- AC (AlternatingCurrent) – Анализ по переменному току (расчет частотных характеристик линейных цепей)
 - Transient – расчет переходных процессов в нелинейных цепях при действии сигналов произвольной формы;
 - Fourier Analysis – Анализ спектров сигналов;
 - Parametric Sweep – Режим вариации параметров компонентов в цепи;
 - Sensitivity – Расчет характеристик чувствительности линейных цепей к вариациям параметров компонентов в режимах постоянного и переменного тока (метод наихудшего случая);
 - Noise Analysis – Анализ спектральной плотности мощности шума на входе и выходе схемы;
 - Вероятностный анализ разброса параметров электронных компонентов методом Монте-Карло.

Алгоритм функционирования SPICE приведен на рисунке 9 ниже.

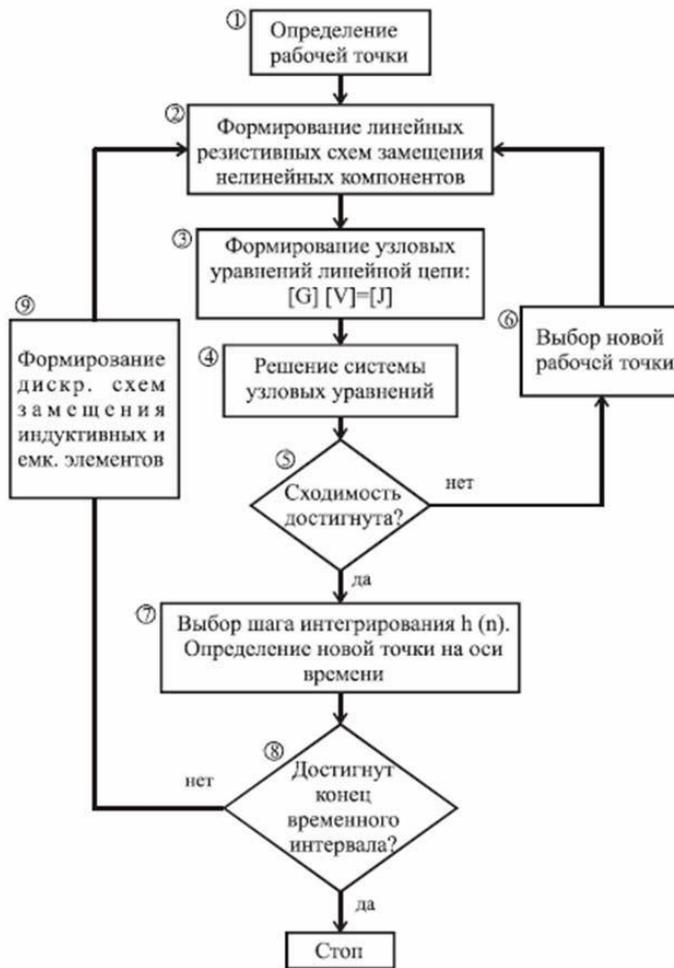


Рис. 9. Блок-схема алгоритма функционирования SPICE

4 ПРОГРАММНЫЕ ПАКЕТЫ СОЗДАНИЯ И ПРОВЕРКИ SPICE МОДЕЛЕЙ ЭЛЕКТРОННЫХ КОМПОНЕНТОВ

1. Model Builder

Производитель: Agilent Technologies

Сайт продукта: <http://www.agilent.com/find/eesof-mbp> Программа Model Builder (MBP) представляет собой

комплексное решение для моделирования полупроводниковых приборов на основе кремния и экстракции полученных моделей.

Основные возможности программы Model Builder [8]:

- создание SPICE моделей
- экстракции моделей для полевых МОП-транзисторов, биполярных транзисторов, диодов и пассивных компонентов
- поддержка современных моделей, включая BSIM6, BSIM-IMG, BSIM-CMG и др.;
- возможность создания моделей старения, моделей функционирования в условиях перегрузки, моделей высоковольтных устройств;
- дружелюбный пользовательский интерфейс и широкий набор вспомогательных программ для моделирования.

На рисунке 10 показан пример моделирования транзистора в данном программном пакете.

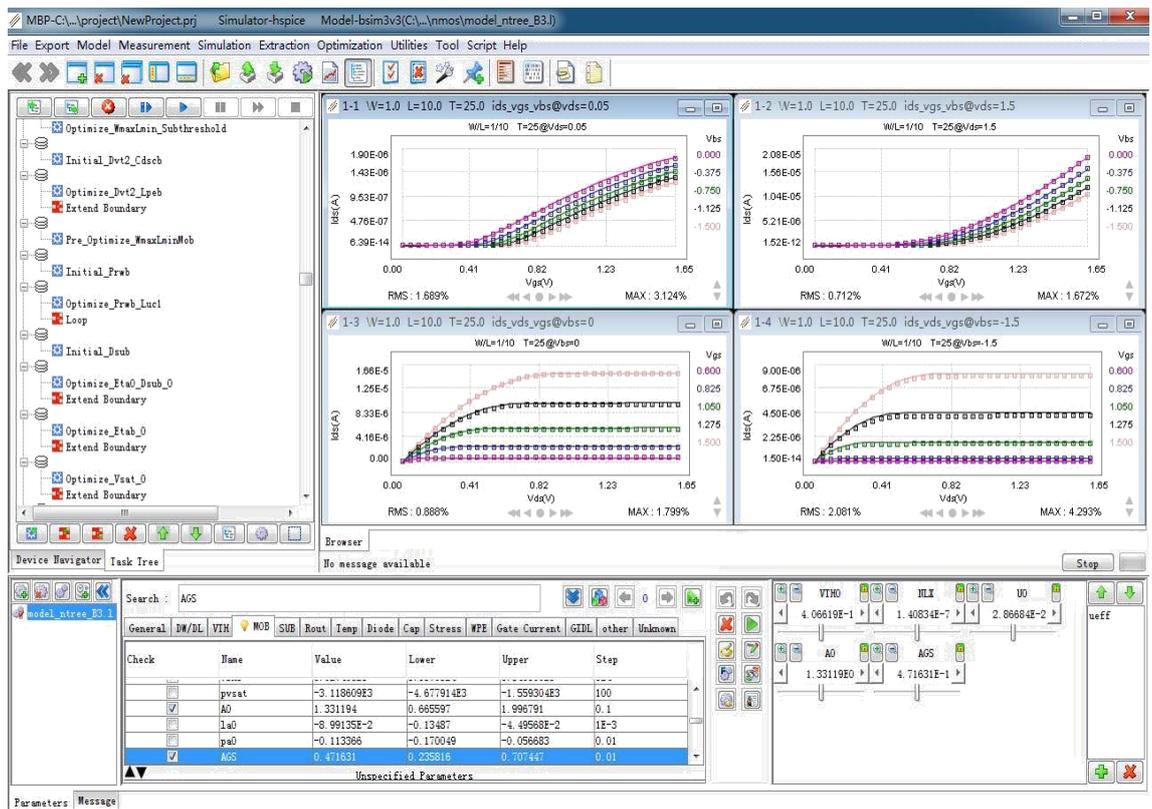


Рис. 10. Пример моделирования в ModelBuilder (MBP)

2. MQA

Производитель: Agilent Technologies

Сайт продукта: <http://www.agilent.com/find/eesof-mqa> Программа проверки качества модели (MQA) позволяет разработчикам и производителям интегральных схем проводить проверку достоверности, сравнение и документирование моделей SPICE.

Основные возможности программы MQA [9]:

- проверка качества и достоверности моделей SPICE;
- отладка результатов моделирования SPICE;
- проверка достоверности заводских моделей; создание детализированных отчетов по проверке качества моделей;
- сравнение различий между версиями модели, симуляторами SPICE и технологиями производства.

Пример анализа моделей SPICE в пакете MQA приведен на рисунке 11.

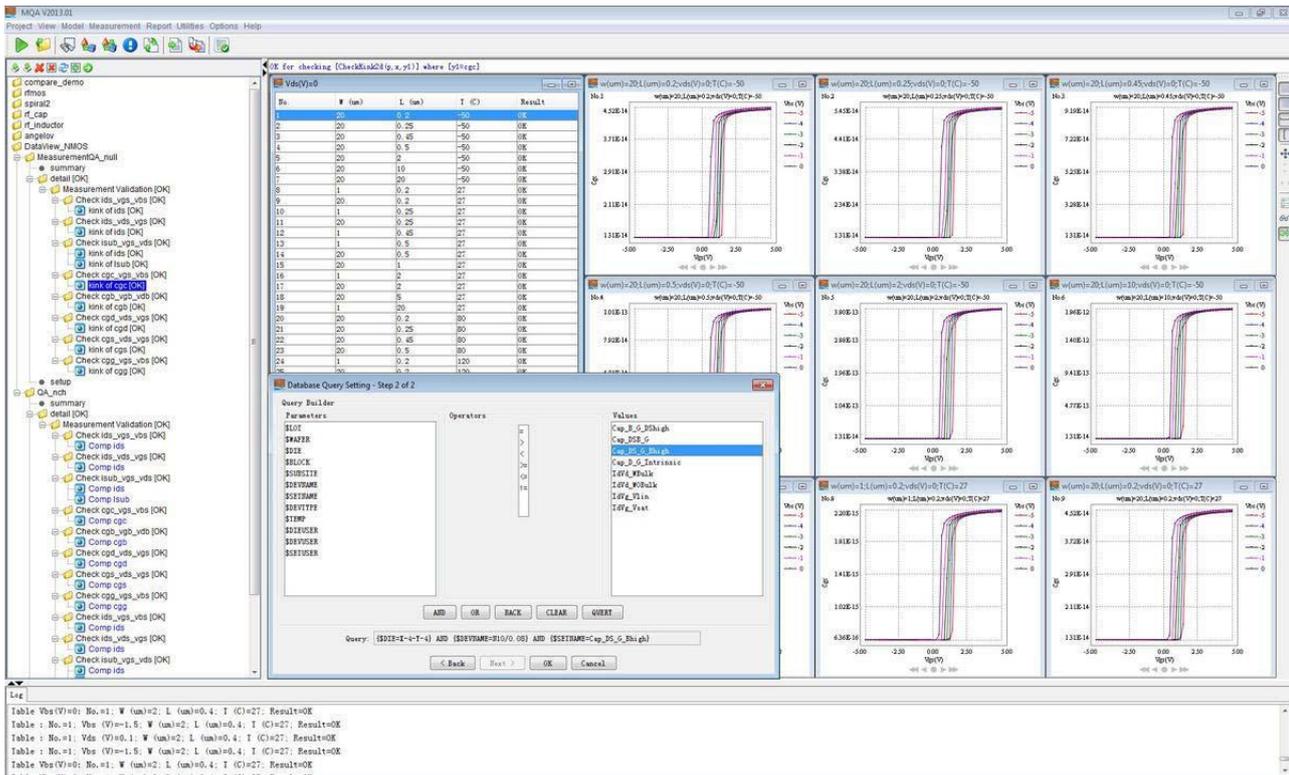


Рис. 11. Пример моделирования в MQA

5 ПРОГРАММНЫЕ ПАКЕТЫ МОДЕЛИРОВАНИЯ ПРИНЦИПИАЛЬНЫХ ЭЛЕКТРИЧЕСКИХ СХЕМ

1. NI Multisim (Electronics Workbench)

Производитель: National Instruments

Сайт продукта: <http://www.ni.com/multisim/>

Программа NI Multisim (ранее Electronics Workbench) представляет собой программную среду создания и моделирования схем электронных устройств [10]. Multisim является программным пакетом схемотехнического моделирования, использующим оригинальные алгоритмы моделирования вычислительного ядра SPICE Беркли. Multisim первоначально была создана компанией Electronics Workbench, которая в настоящее время является подразделением National Instruments. Пакет Multisim включает всебя моделирование микроконтроллеров (Multi MCU), а также интегрированные функции импорта и экспорта принципиальных схем, разработанных в других САПР. Multisim широко используется образовании и промышленности для изучения работы электронных схем и их моделирования.

В программе имеется большой набор виртуальных измерительных приборов (вольтметры, амперметры, генераторы сигналов, осциллограф, измеритель амплитудно-частотной характеристики и т.п.). Испытуемая схема "монтируется" на виртуальном лабораторном столе (work bench), и затем проводятся необходимые измерения (см. рисунок 12). При этом настройка виртуальных измерительных приборов осуществляется практически так же, как и при работе с органами управления передней панели реальных приборов.

Благодаря интеграции пакетов Multisim и Lab VIEW, специалисты могут точнее определять и анализировать поведение схем и детектировать ошибки еще на ранних стадиях разработки.

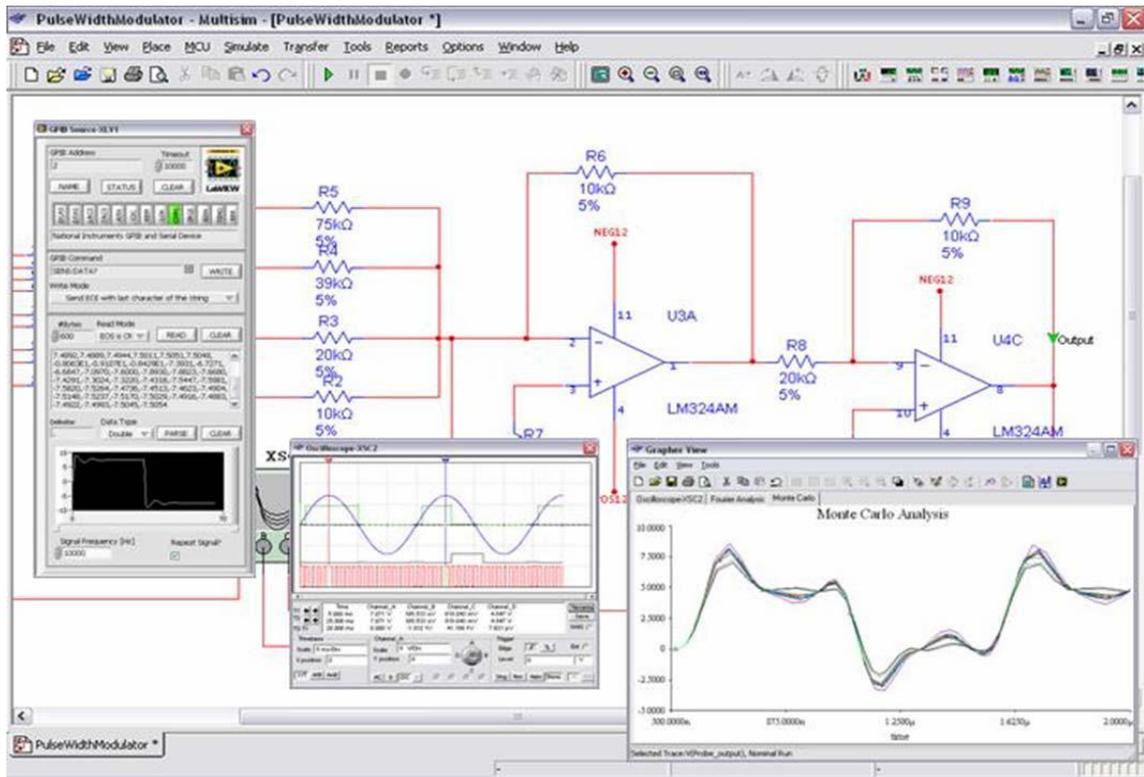


Рис. 12. Пример моделирования в NI Multisim

2. Micro-Cap

Производитель: Spectrum Software

Сайт продукта: <http://www.spectrum-soft.com/>

Семейство программ схемотехнического анализа Micro-Cap (Microcomputer Circuit Analysis Program) пользуется также достаточно большой популярностью. Это связано с тем, что эти программы традиционно имеют удобный, дружелюбный интерфейс и достаточно скромные требования к программно-аппаратным средствам компьютера. Но при этом предоставляемые возможности достаточно велики. Micro-Cap позволяет анализировать не только аналоговые, но и цифровые устройства [11]. Возможно также и смешанное моделирование аналого-цифровых электронных устройств. Опытные пользователи пакета могут также в нестандартной ситуации создавать собственные макромодели, облегчающие имитационное моделирование без потери существенной информации о поведении системы.

Пакет Micro-Cap рекомендуется для выполнения исследовательских работ, не предусматривающих немедленной конструкторской реализации (т.е. разводки

печатной платы и оформления конструкторской документации). В состав программы входит модуль расчета параметров моделей аналоговых элементов по результатам экспериментальных исследований (таким способом создаются новые модели). В системе предусмотрен режим исследования чувствительности выходного сигнала к изменению параметров любого элемента схемы. Есть возможность определить входное и выходное сопротивления устройства. Предусмотрена возможность разработки активных и пассивных фильтров с заданными параметрами.

Полная совместимость со SPICE-моделями и SPICE-схемами и развитие возможности конвертирования позволяет пользователю Micro-Cap успешно пользоваться всеми разработками других схемотехнических САПР (например, ORCAD), а полученные навыки проектирования и моделирования позволят в случае необходимости гораздо быстрее осваивать более сложные профессиональные пакеты моделирования.

Пример моделирования принципиальной электрической схемы в пакете Micro-Cap приведен на рисунке 13.

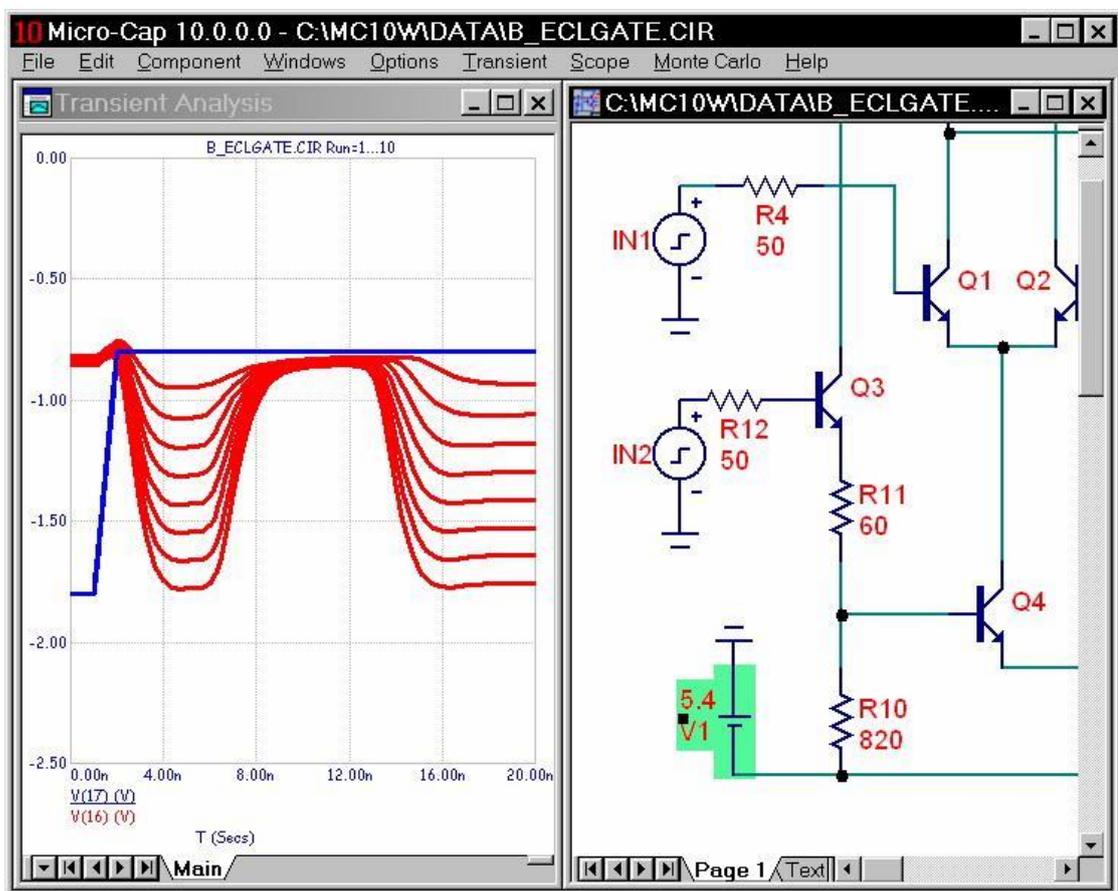


Рис. 13. Пример моделирования в Micro-Cap

3. Cadence (OrCAD) PSpice A/D

Производитель: Cadence Design Systems

Сайт продукта: http://www.cadence.com/products/orcad/pspice_simulation/Pages/default.aspx

Cadence Pspice A/D является полнофункциональным симулятором аналоговых и смешанных электрических цепей и де-факто является промышленным стандартом среди программ моделирования, основанных на SPICE. Данный программный пакет позволяет симулировать сложные схемы, содержащими как аналоговые, так и цифровые компоненты, и поддерживает широкий спектр имитационных моделей, таких как транзисторы, широтно-импульсные модуляторы, ЦАП и АЦП [12]. Встроенные математические функций и поведенческие методы моделирования позволяют осуществлять быстрое и точное моделирование схем, а также эффективно выполнять отладку. PSpice A/D также позволяет пользователям разрабатывать имитационные модели для трансформаторов и дросселей постоянного тока.

Дополнительные возможности анализа основаны на интеграции с MATLAB Simulink для совместного моделирования. Продвинутые функции, такие как моделирование температуры компонентов и их стресс-анализ, электромеханическое моделирование, анализ наихудшего случая, моделирование по методу Монте-Карло, – предназначены для разработки высокопроизводительных систем, которые должны быть надежными и выдерживать вариации различных параметров.

Пример моделирования принципиальной электрической схемы в пакете Pspice A/D приведен на рисунке 14.

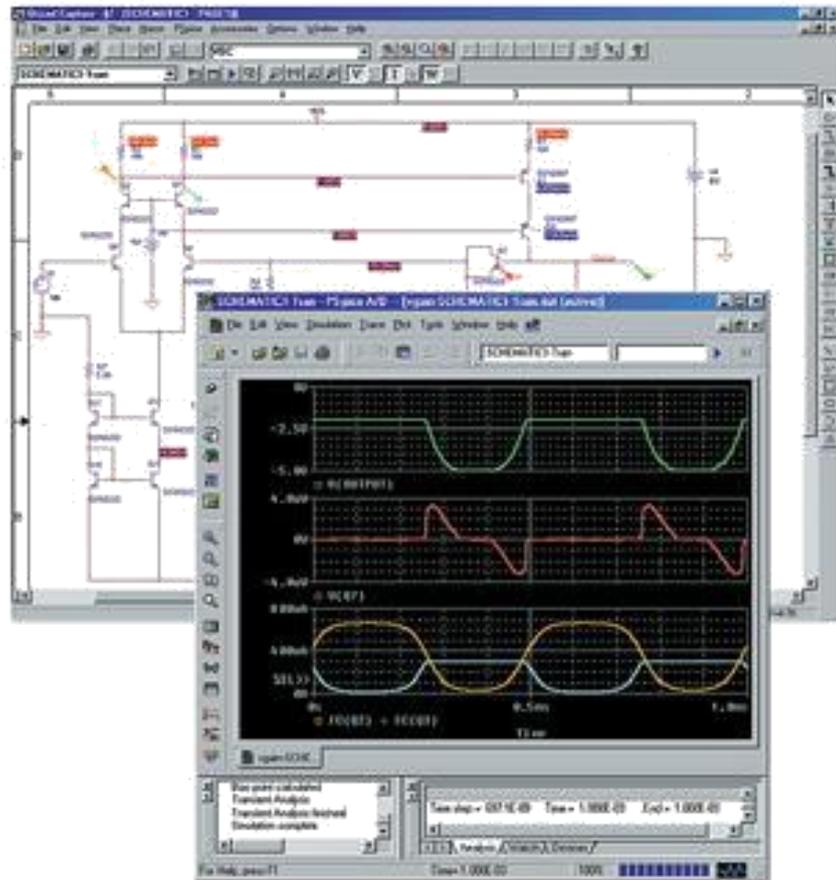


Рис. 14. Пример моделирования в PSpice A/D

6 ЛАБОРАТОРНЫЕ РАБОТЫ В СРЕДЕ SYSTEM VUE

ЛАБОРАТОРНАЯ РАБОТА 1. ГЕНЕРИРУЕМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ БИТОВ

В этой лабораторной работе мы создадим очень простую последовательность битов и проанализируем её выход в форме глазковой диаграммы.

- Запустите System Vue (в этом учебном пособии используется версия 2011.10), дважды щёлкнув на соответствующем значке рабочего стола.
- После загрузки откроется страница приветствия с учебными видеороликами. Пропустите её, щёлкнув- кнопку «Заккрыть».
- Следующее окно позволяет открыть имеющиеся рабочие среды или шаблоны. Щёлкнув на «отмену», создайте пустую рабочую среду.

Откроется окно схемы, в котором и будет выполняться работа по проектированию (рис. 9.1). В верхней части экрана расположена панель инструментов; с левой стороны вы увидите структуру директорий, которая представляет ваше «дерево рабочей среды». Дерево рабочей среды позволяет организовать схемы, представления, уравнения и т.п. в виде иерархической структуры папок. Под деревом рабочей среды расположено «окно настройки», которое является одной из самых мощных функций SystemVue, позволяющих использовать настраиваемые переменные (настройка значений переменных в режиме реального времени) в любом месте проекта. Справа расположена библиотека компонентов (эта часть называется селектором компонентов). В нижней части экрана отображается информация об ошибках. Кроме того, обратите внимание на вкладки под экраном ошибок, показывающие прочую информацию о событиях.

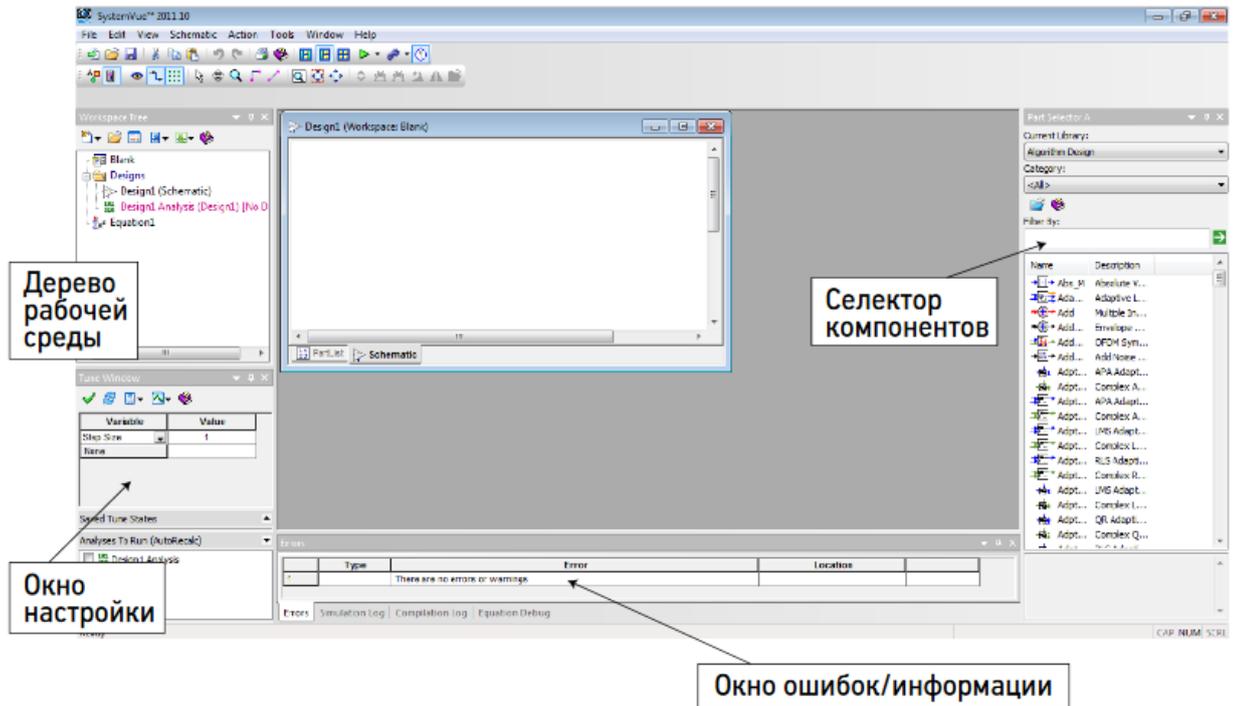


Рис. 9.1. Окно схемы SystemVue.

Итак, начнём.

Откройте меню **File > Save As > My_CDMA_Sys** (Файл>Сохранить как> My_CDMA_Sys). В результате вся рабочая среда сохранится под именем My_CDMA_Sys. Рекомендуется сохранять рабочую среду после каждой лабораторной работы, выполняя команду **File>Save**.

По умолчанию System Vue создаёт папку и сохраняет в ней проект и анализ. Переименуйте папку “Designs” (Проекты), наведя на неё указатель мыши. Щёлкните правой кнопкой и выберите **Rename** (переименовать). Назовите папку “Step 1_Bits”. Переименуйте схему Design в “Bits”, а Design Analysis в “DF_Bits”. Это анализ процесса проектирования, который собирает информацию о наборе значений и запускает нужную программу – примерно так, как это делает компьютер

Дважды щёлкните на “DF_Bits” и введите значения, показанные на рис. 9.2.

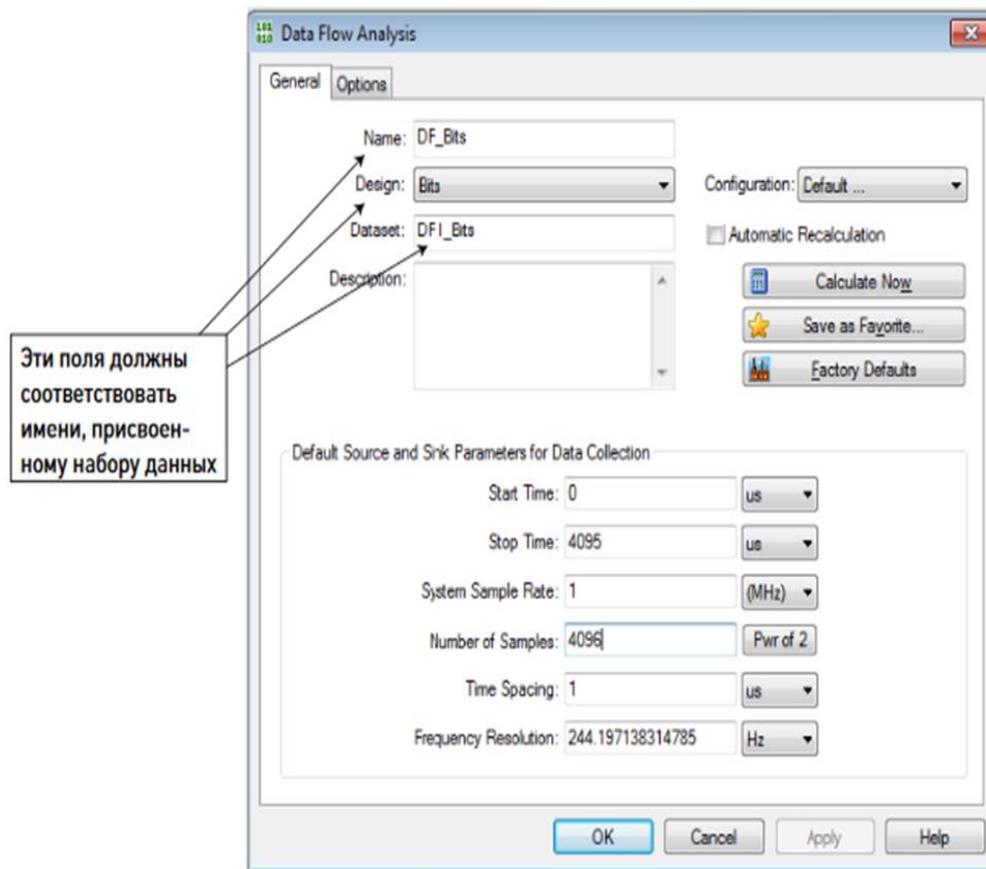


Рис. 9.2. Ввод значений в окно анализа процесса проектирования.

Ключевым параметром является System Sample Rate (системная частота дискретизации). Другим важным параметром является Number of Samples (число выборок), который определяет длину моделирования. Время запуска и останова устанавливается автоматически. Разрешение по частоте рассчитывается по времени останова.

- Щёлкнув ОК, закройте окно.
- Разверните рабочую среду на весь экран.
- Из “Part Selector A” (селектор компонента A) введите “Bits” в поле “filter by” (фильтрация по) и нажмите Enter. Вы увидите все компоненты, содержащие слово Bits. Однократно щёлкните на “Bits” и перетащите компонент в пустую схему. Щёлкните снова и отпустите компонент.
- Дважды щёлкните на компоненте “Bits”. Одна из замечательных и полезных особенностей System Vue заключается в том, что многие компоненты являются полиморфными моделями

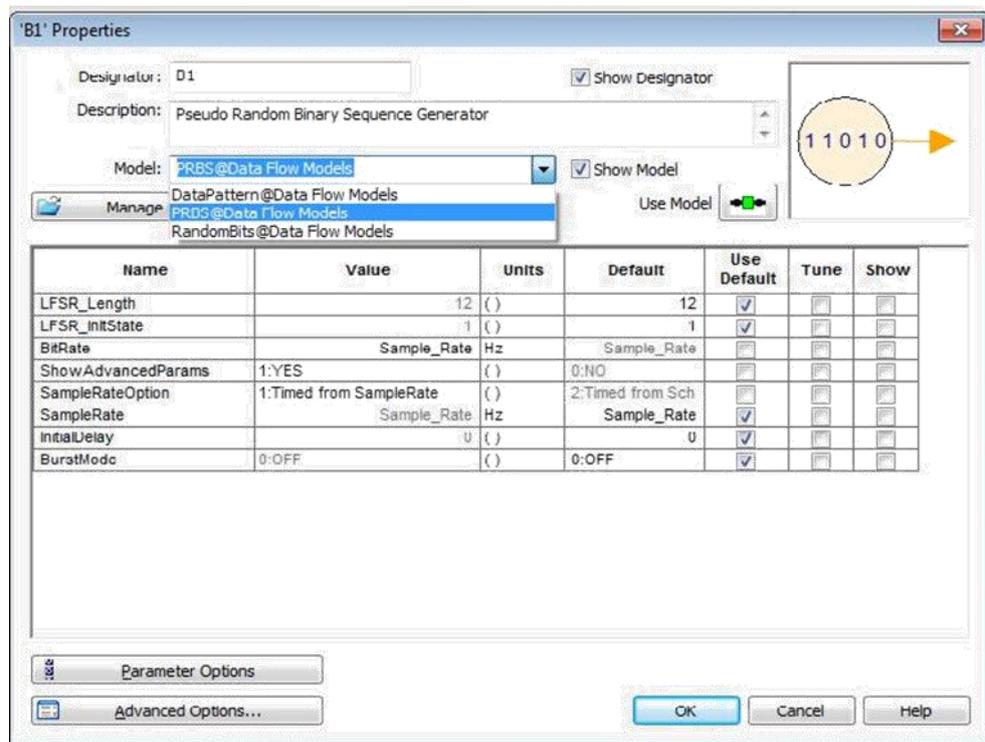


Рис. 9.3. Выпадающее меню свойств В1.

- Щёлкните на выпадающем меню возле поля “model” (модель) и выберите “PRBS @ Data Flow Models”. Эта операция создаст псевдослучайную последовательность.

- Заполните поля, как показано на рис. 9.3. Обратите внимание, что вы можете добавлять в этот список собственные модели, что характеризует System Vue как гибко настраиваемую систему.

Обратите внимание, что “Bit Rate” (битовая скорость) установлена в значение “Sample Rate” (частота дискретизации). Благодаря таким настройкам мы получаем одну выборку на бит.

Теперь давайте подключим приёмник к источнику битов и отобразим нашу битовую последовательность. Введите “Sink” (приёмник) в Part Selector. Щёлкните на “Data Sink” (приёмник данных) и установите его рядом с источником битов. Сформируется соединение, как показано на рис. 9.4.

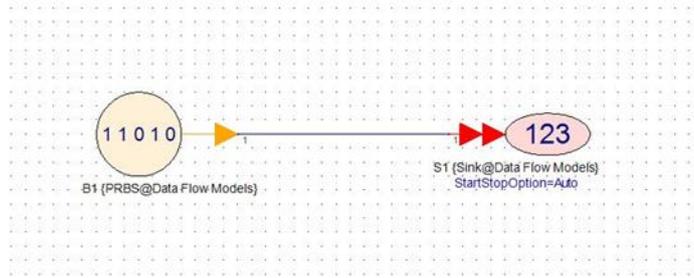


Рис. 9.4. Подключение приёмника к источнику битов.

Полезный совет. Чтобы установить приёмник, можно нажать на клавиатуре “s” и затем щёлкнуть на схеме. Чтобы установить соединение, нажмите “w” (перетащите соединение от одного компонента к другому), чтобы установить фильтр нажмите “f” и т.д.

Щёлкните правой кнопкой на “DF_Bits (Bits)” и выберите “Run (calculate now)” (запустить (рассчитать сразу)). Вы увидите, что к модели добавился набор данных. С помощью этого набора данных можно наблюдать выход

Чтобы сохранить структуру, щёлкните правой кнопкой на наборе данных и выберите Rename (переименовать). Назовите его “DF1_Bits”.

Поскольку имя набора данных изменилось, нужно убедиться в том, что моделирование выполняется с нужным набором данных.

- Дважды щёлкните на анализе “DF_Bits” рядом с полем Dataset (набор данных). Напечатайте “DF1_Bits” и щёлкните ОК.

Заметьте, что теперь Data Flow (поток данных) и Dataset окрасились в красный цвет. Это значит, что модель не соответствует актуальной информации, и мы внесли изменения в наш проект.

- Щёлкните правой кнопкой на “DF_Bits” и выберите Run (calculate now).

После безошибочного моделирования построим диаграмму битов.

- Дважды щёлкните на наборе данных “DF1_Bits”, он откроется, и вы увидите переменные, соответствующие приёмнику.

- Щёлкните правой кнопкой на S1 и выберите Add to Graph>New Graph Series Wizard (Добавить к диаграмме > Мастер создания диаграмм) (рис. 9.5).

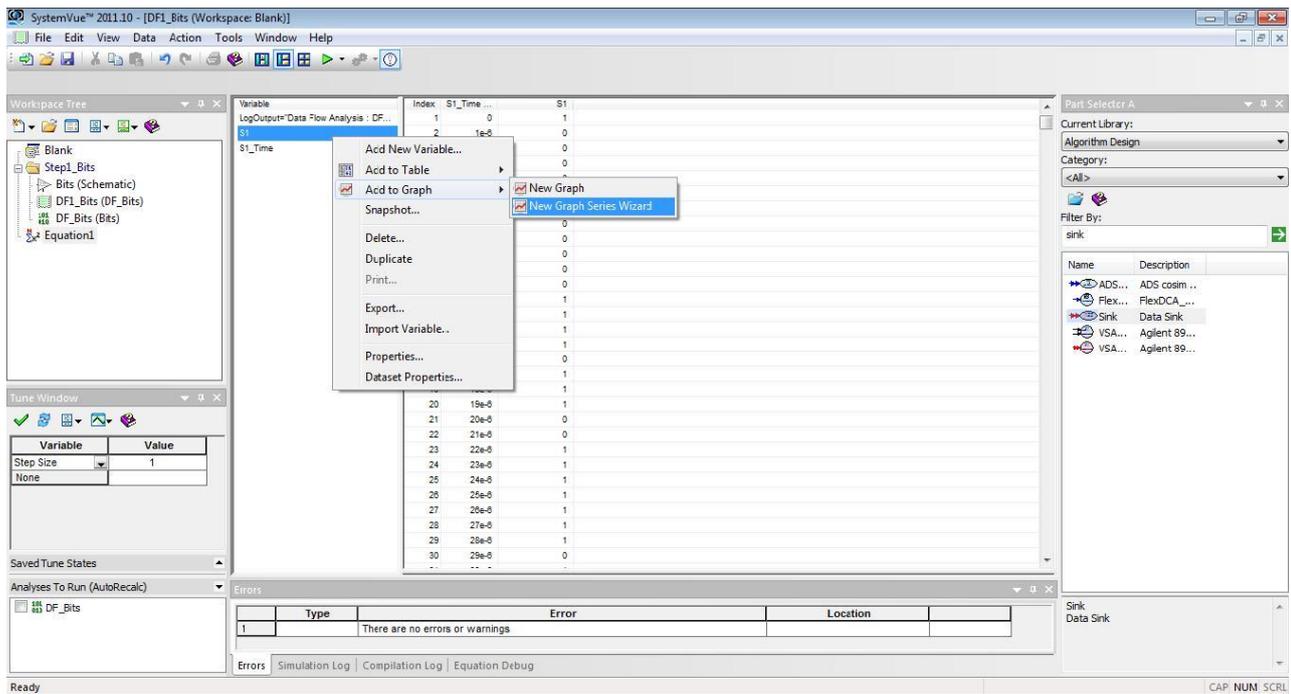


Рис. 9.5. Выбор мастера создания диаграмм.

- В поле “Type of series elected ” (выбранный тип последовательности) выберите “Time” (время). Убедитесь, что в поле “Data Selected” (выбранные данные) отмечен селектор “S1”, поскольку это и есть выход, диаграмму которого мы хотим построить (рис. 9.6).

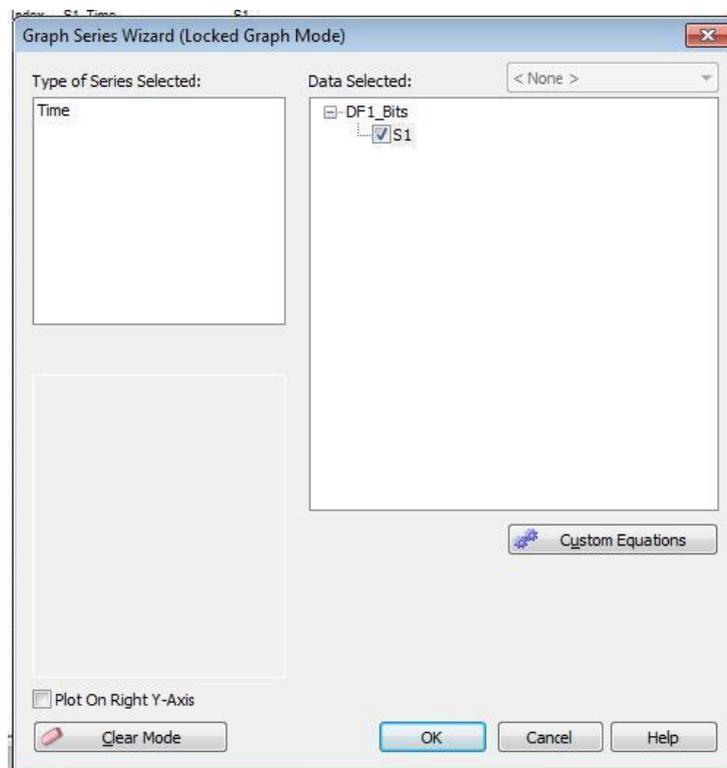


Рис. 9.6. Выберите “S1”, чтобы построить диаграмму этого выхода.

- Нажмите ОК. При этом откроется окно свойств диаграммы, в котором в качестве единственной переменной будет указана “S1” (она будет называться My Var).
- Нажмите ОК. Откроется график проекта “Bits” во временной области (рис. 9.7).

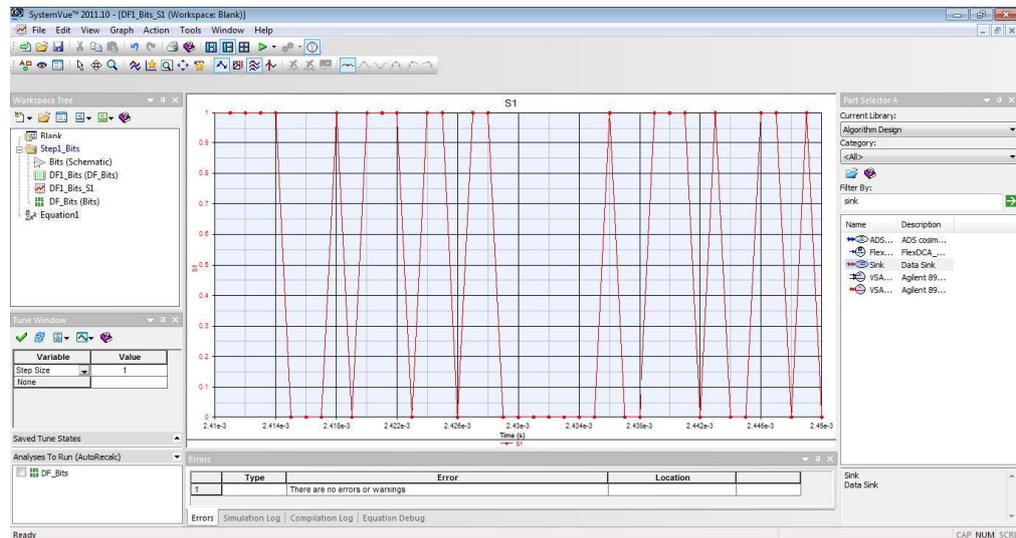


Рис. 9.7. График проекта “Bits”.

- Увеличьте масштаб изображения по оси X колёсиком мыши, как показано на рис. 9.7. Прокручивая колёсико вперёд, вы более чётко увидите случайные единицы и нули.

Часто для анализа систем связи используются глазковые диаграммы, которые позволяют глубже понять характеристики сигнала.

1. Измените схему так, чтобы она соответствовала рис. 9.8, для чего скопируйте созданную ранее схему “Bits” (убедитесь, что используемые компоненты имеют те же переменные, что и раньше).

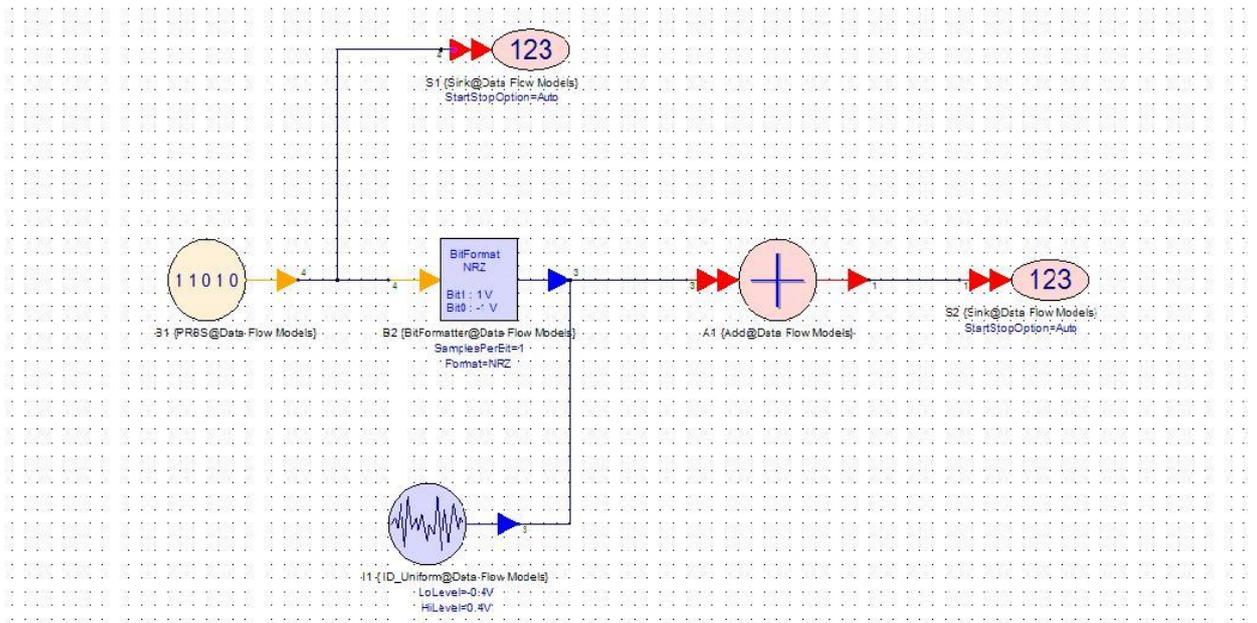


Рис. 9.8. Для схемы можно построить глазковую диаграмму, чтобы лучше понять характеристики сигнала.

2. Напечатав “NRZ” в поле поиска селектора компонентов, можно найти BitFormat. Аналогичным образом для поиска идентификатора источника однородного шума напечатайте “Noise”, а чтобы найти в библиотеке компонентов сумматор, выполните поиск по слову “Add”. Компонент NRZ преобразует цифровые разряды в заданное напряжение. Для того, чтобы сделать глазковую диаграмму более интересной и реалистичной, мы добавили немного шума.

3. Дважды щёлкните на источнике шума и установите напряжение Lo Level (нижний уровень) на $-0,4$ В, а напряжение Hi Level (верхний уровень) на $0,4$ В. Для этого щёлкните в соответствующих полях и введите значения.

4. Запустите моделирование, щёлкнув правой кнопкой на “DF_Bits” и выбрав Run.

5. Дважды щёлкните на наборе данных, щёлкните правой кнопкой на S2 и выберите Addto Graph>New Graph Series Wizard>Eye>Custom Equations (Добавить к диаграмме > Мастер создания диаграмм > Глазковая > Специальные уравнения). Измените уравнения на:

6. Щёлкнув три раза ОК, закройте все окна. Теперь вы увидите глазковую диаграмму, как на рис. 9.9.

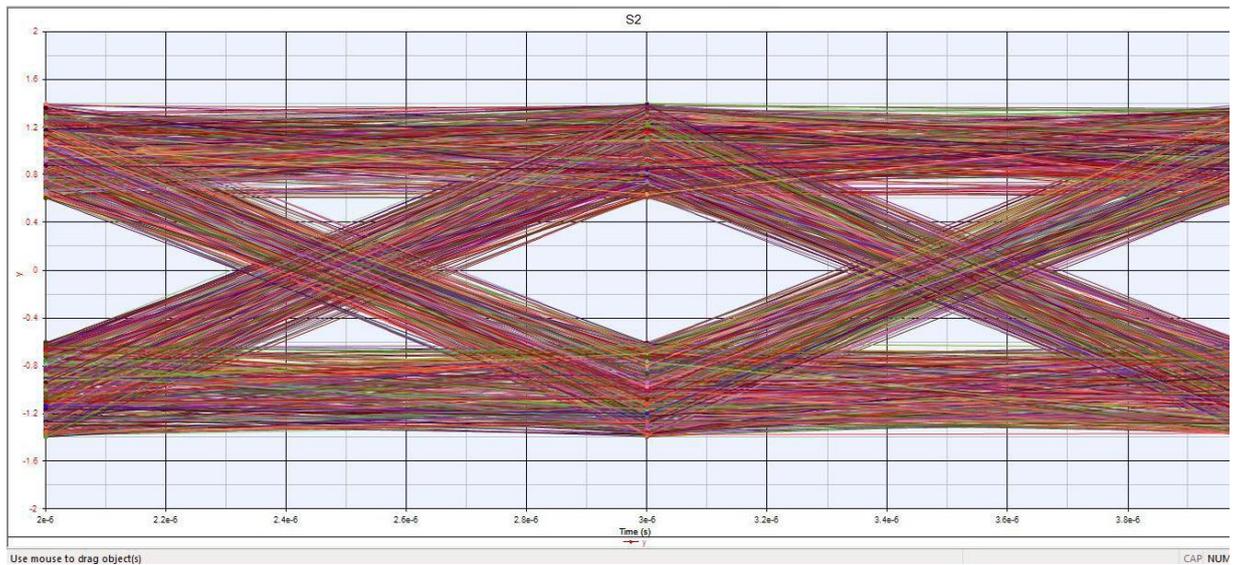


Рис. 9.9. Глазковая диаграмма схемы, показанной на рис. 9.8.

Чтобы изменить диаграмму, дважды щёлкните на ней. При этом откроется диалоговое окно свойств диаграммы. Обратите внимание, что переменная “y” получила название “<продолжение обработки>”. Это значит, что вы можете продолжить обработку данных во временной области для построения глазковой диаграммы. Нажав “edit” (изменить) в диалоговом окне, вы вернётесь в мастер создания диаграмм, где можете изменить уравнения диаграммы.

**ЛАБОРАТОРНАЯ РАБОТА 1 ЗАВЕРШЕНА. НЕ ЗАБУДЬТЕ
СОХРАНИТЬ РАБОЧУЮ СРЕДУ.**

ЛАБОРАТОРНАЯ РАБОТА 2. ФОРМИРОВАНИЕ СИМВОЛОВ

В этой лабораторной работе мы преобразуем данные в формат распространённой цифровой модуляции. Затем мы рассмотрим эти данные на сигнальном созвездии.

Создайте новую папку “Step 2_Mapping”, для чего выполните следующие шаги.

1. Щёлкните правой кнопкой на ветви “My_CDMA_Sys”, которая расположена в верхней части дерева рабочей среды, и выберите Add>Add Folder (Добавить > Добавить папку). Назовите папку “Step 2_Mapping”.
2. Щёлкните правой кнопкой на папке “Step 2_Mapping” и выберите Add>Designs>Add Schematic (Добавить > Проекты > Добавить схему). Назовите проект “Mapping” (формирование символов).
3. Добавьте набор данных, для чего щёлкните правой кнопкой на папке “Step 2_Mapping” и выберите Add>Add Data (Добавить > Добавить данные).
4. Щёлкните правой кнопкой на папке “Step2_Mapping” и выберите Add>Analysis>Add Data Flow Analysis (Добавить > Анализ > Добавить анализ потока данных). Назовите его “DF_Mapping”. Проверьте, что информация в всплывающем окне “Data Flow Analysis” соответствует показанной на рис 9.10.

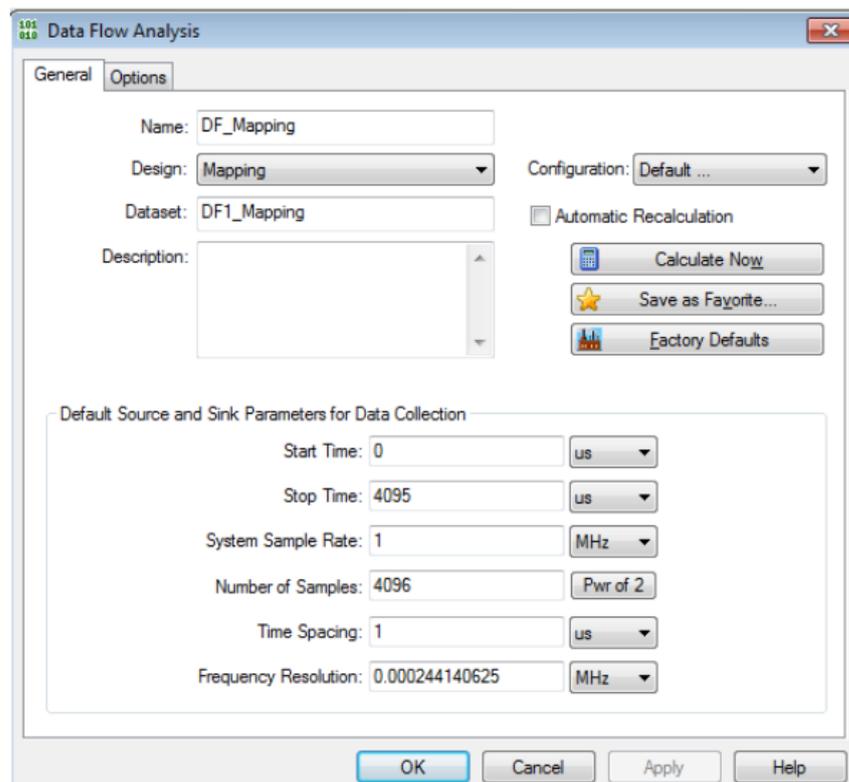


Рис. 9.10. Поля проекта и данных должны соответствовать модели.

Давайте построим распределение битов модуляции QPSK, как показано на рис. 9.11. Для компонента Bits установите параметры так же, как вы делали это в лабораторной работе 1, в пункте 4.

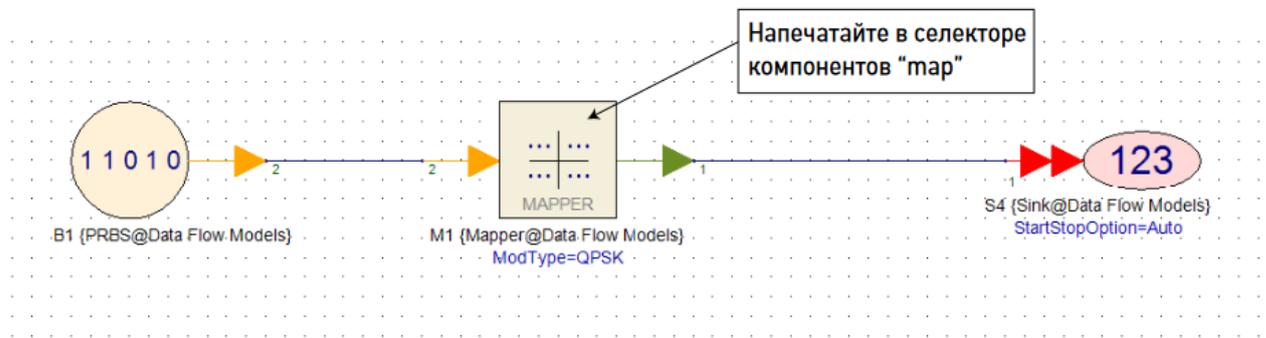


Рис. 9.11. Распределение битов.

Запустите моделирование.

5. Щёлкните правой кнопкой на “DF_Mapping” и выберите Run (Calculate Now).
6. Дважды щёлкните на наборе данных, щёлкните правой кнопкой на приёмнике (S1) и выберите Addto Graph>NewGraph Series Wizard>Constellation>Ok>Ok (Добавить к диаграмме > Мастер создания диаграмм > Сигнальное созвездие > ОК > ОК).

Вы должны увидеть идеальное сигнальное созвездие (рис. 9.12).

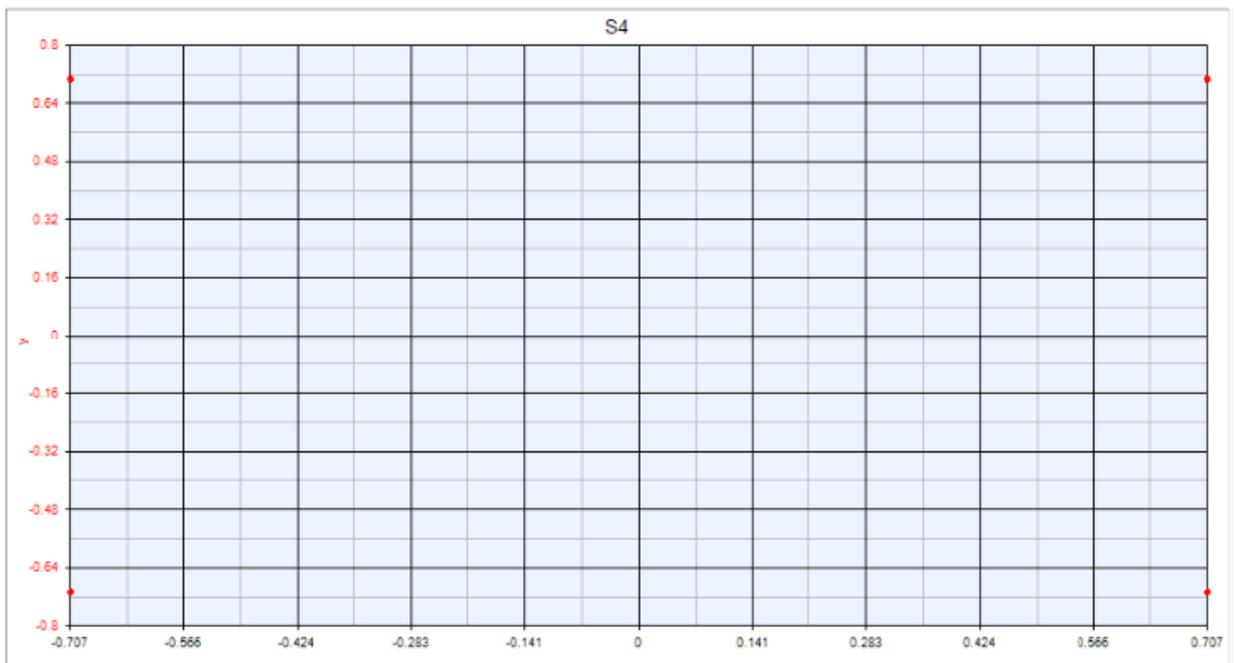


Рис. 9.12. Идеальное сигнальное созвездие.

7. Измените проект, как показано на рис. 9.13.

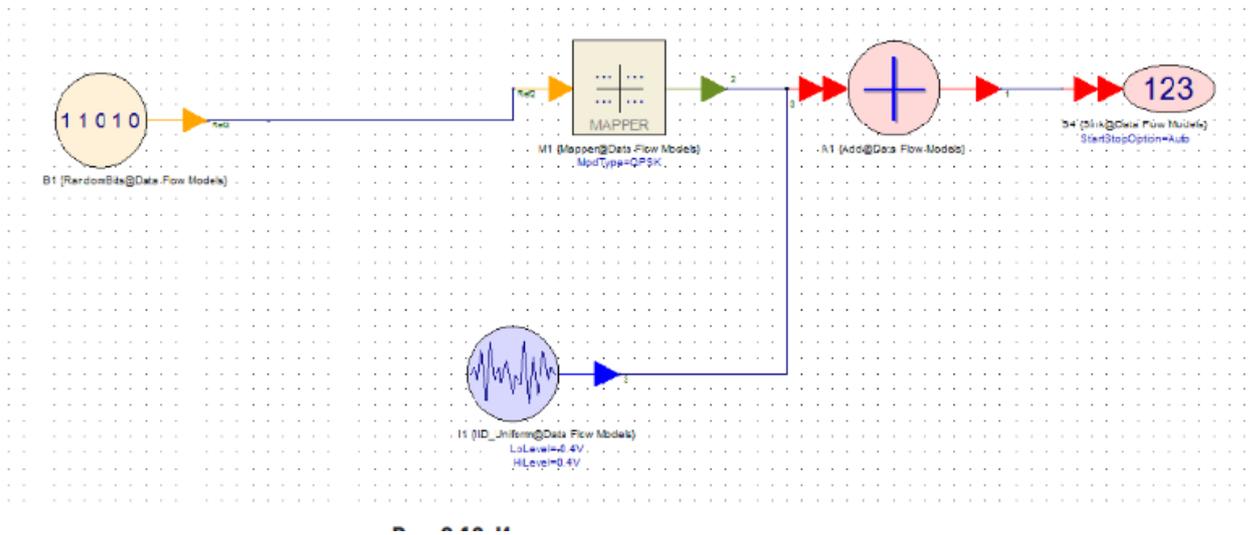


Рис. 9.13. Изменение сигнального созвездия.

Рекомендуется дать соединениям имена. Для этого щёлкните правой кнопкой на соединении и выберите Net>Edit Netname (Цепь > Изменить имя цепи). В данном случае мы выбрали имя Ref.

8. Теперь запустите моделирование, щёлкнув правой кнопкой на “DF_Mapping” и выбрав Run.

9. Вернитесь к предыдущей диаграмме, и вы увидите, что к сигнальному созвездию добавился шум (рис. 9.14).

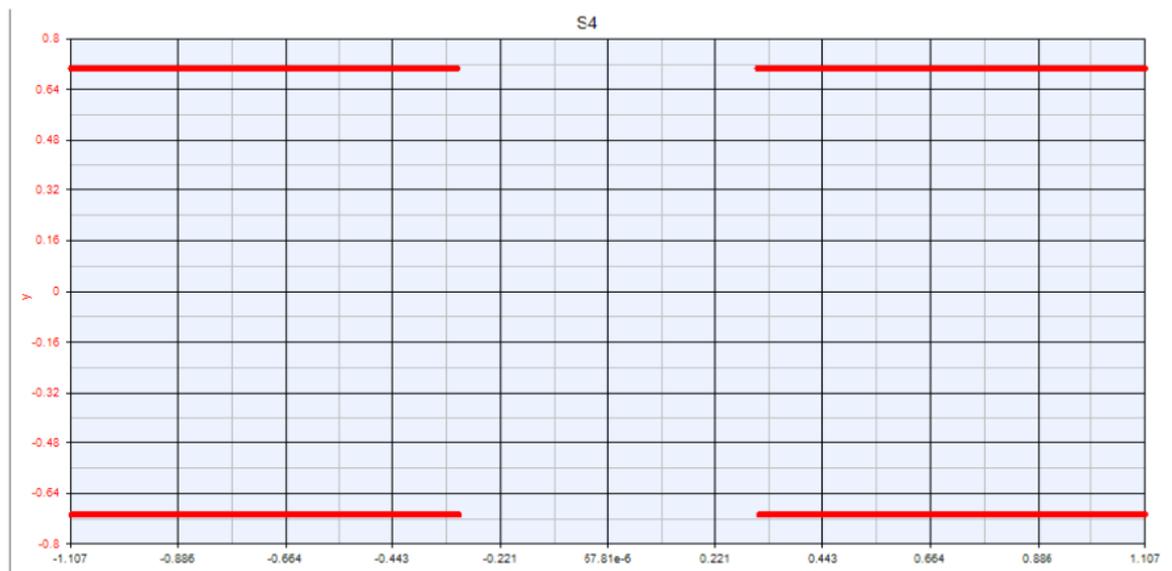


Рис. 9.14. Сигнальное созвездие с добавленным шумом.

КОНЕЦ ЛАБОРАТОРНОЙ РАБОТЫ 2.

ЛАБОРАТОРНАЯ РАБОТА 3. КОД УОЛША

В этой лабораторной работе мы создадим код Уолша для наших данных.

1. Щёлкните правой кнопкой на “My_CDMA_Sys” и выберите Add>Add folder. Назовите папку “Step 3_ Walsh Code”.

2. Щёлкните правой кнопкой на папке “Step 3_ Walsh Code” и выберите Add>Designs>Add Schematic. Назовите схему “Walsh Code”.

3. Щёлкните правой кнопкой на папке “Step 3_ Walsh Code” и выберите Add>Analysis>Data Flow Analysis (Добавить > Анализ > Анализ потока данных). Назовите его “DF_ Walsh Code”.

4. Щёлкните правой кнопкой на папке “Step 3_ Walsh Code” и выберите Add>Add Data. Назовите данные “DF1_ Walsh Code”.

5. Щёлкните правой кнопкой на Data Flow Analysis и измените параметры так, чтобы проект и набор данных соответствовали друг другу, то есть, чтобы в поле design было написано “Walsh Code”, а в поле data set было написано “DF1_ Walsh Code”. Все остальные параметры должны соответствовать ранее настроенному анализу.

Скопируйте схему из “Step 2_Mapping”, и измените проект, как показано на рис. 9.15.

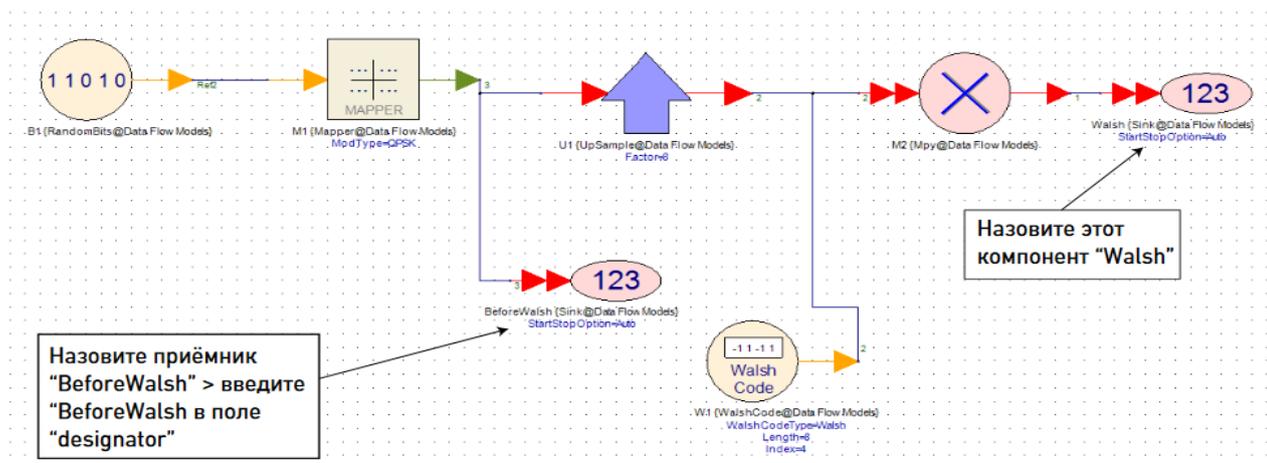


Рис. 9.15. Видоизменённая схема.

После построения графика данных для формирования символов, символьная скорость снижается в 4 раза из-за двойной поляризации и QPSK.

Передискретизатор до некоторой степени повышает частоту дискретизации. В данном случае мы повышаем частоту дискретизации в 8 раз, и символьная скорость становится равной 4 МГц, что создаёт достаточную полосу для передачи на несущей по каналу связи.

6. Дважды щёлкните на компоненте Up Sample (повышающий передискретизатор) и установите множитель 8. Выберите режим “Hold Sample”. Входная выборка повторится на выходе 8 раз.

7. Дважды щёлкните на компоненте Walsh Code Generator (генератор кода Уолша). Установите значения, показанные на рис. 9.16.

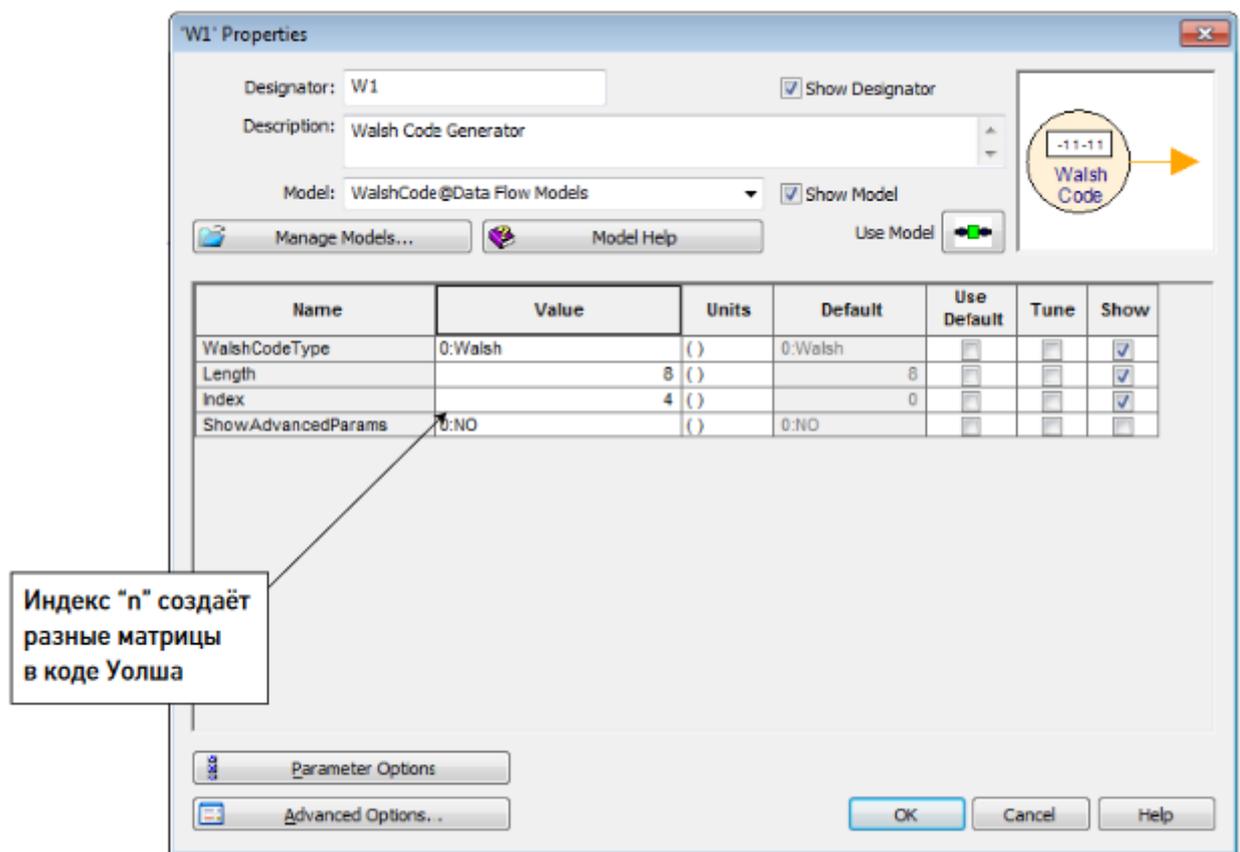


Рис. 9.16. Установка параметров генератора кода Уолша.

Давайте рассмотрим значения кода до и после генерации кода Уолша.

8. Запустите моделирование.

9. Дважды щёлкните на наборе данных, щёлкните правой кнопкой на “Before Walsh” (до Уолша) и выберите Add to Table > New Table (Добавить в таблицу > Создать таблицу). Щёлкните «Close» (закреть).

10. Щёлкните правой кнопкой на верхней части таблицы рядом с именами значений (рис. 9.17).

Index	BeforeWalsh...	re(BeforeWalsh)	im(B...
1	1e-8	0.707	
2	3e-8	-0.707	
3	5e-8	0.707	
4	7e-8	-0.707	
5	9e-8	-0.707	
6	11e-8	-0.707	-0.707
7	13e-8	-0.707	-0.707
8	15e-8	0.707	-0.707
9	17e-8	0.707	0.707

Рис. 9.17. В таблице щёлкните на “Index” (индекс).

11. Щёлкните на “Index” и на “Before Walsh_Time(s)”. Вы увидите, что в таблице пропали соответствующие столбцы.

12. Дважды щёлкните на наборе данных Walsh Code (код Уолша), щёлкните правой кнопкой на Walsh и выберите Addto Table>Addto “DF _Walsh Code_Before Walsh”.

13. Разверните окно на весь экран. Столбец “Walsh_Time(s)” можно удалить, потому что он не нужен.

Рассмотрите и сравните значения до и после генерации кода Уолша. Обратите внимание, что они отличаются. Это значит, что наш код надёжно защищён кодом Уолша. Теперь мы должны сделать так, чтобы такой же код Уолша применялся в приёмнике для декодирования данных и извлечения исходной информации. В последней лабораторной работе мы используем некоторые примеры и посмотрим, что происходит с отдельными битами, когда они сопоставляются с кодом Уолша.

ЛАБОРАТОРНАЯ РАБОТА 3 ЗАВЕРШЕНА

ЛАБОРАТОРНАЯ РАБОТА 4. ПРОЕКТИРОВАНИЕ ФИЛЬТРА

В этой лабораторной работе мы создадим папку для расчёта полосы модуляции.

1. Создайте папку и назовите её “Step 4_Filter”.
2. Добавьте схему с именем “Filter” (фильтр).
3. Добавьте набор данных с именем “DF 1_Filter”.
4. Добавьте анализ потока данных с именем “DF_Filter”. Установите Number of Samples (число выборок) 4096.

Скопируйте схему, созданную на шаге 3, и вставьте её во вновь созданную пустую схему. Измените проект, как показано на рис. 9.18.

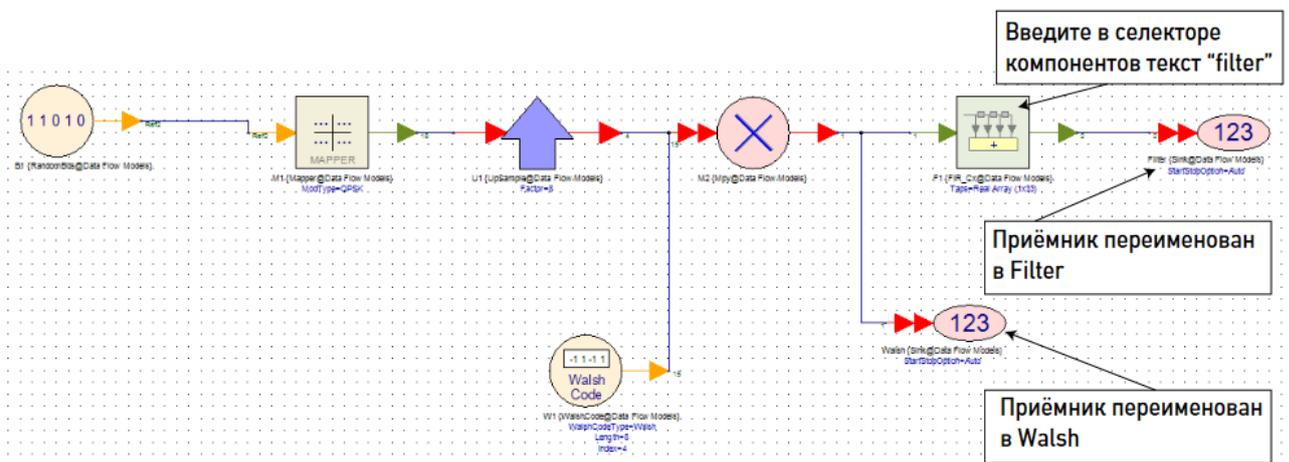


Рис. 9.18. Измените проект, как показано на этом рисунке.

5. Дважды щёлкните на компоненте Filter. Откроются свойства проекта. Нажмите на кнопку Filter Designer (конструктор фильтров). Измените свойства так, чтобы они соответствовали рис. 9.19. Внеся изменения, закройте страницу свойств.

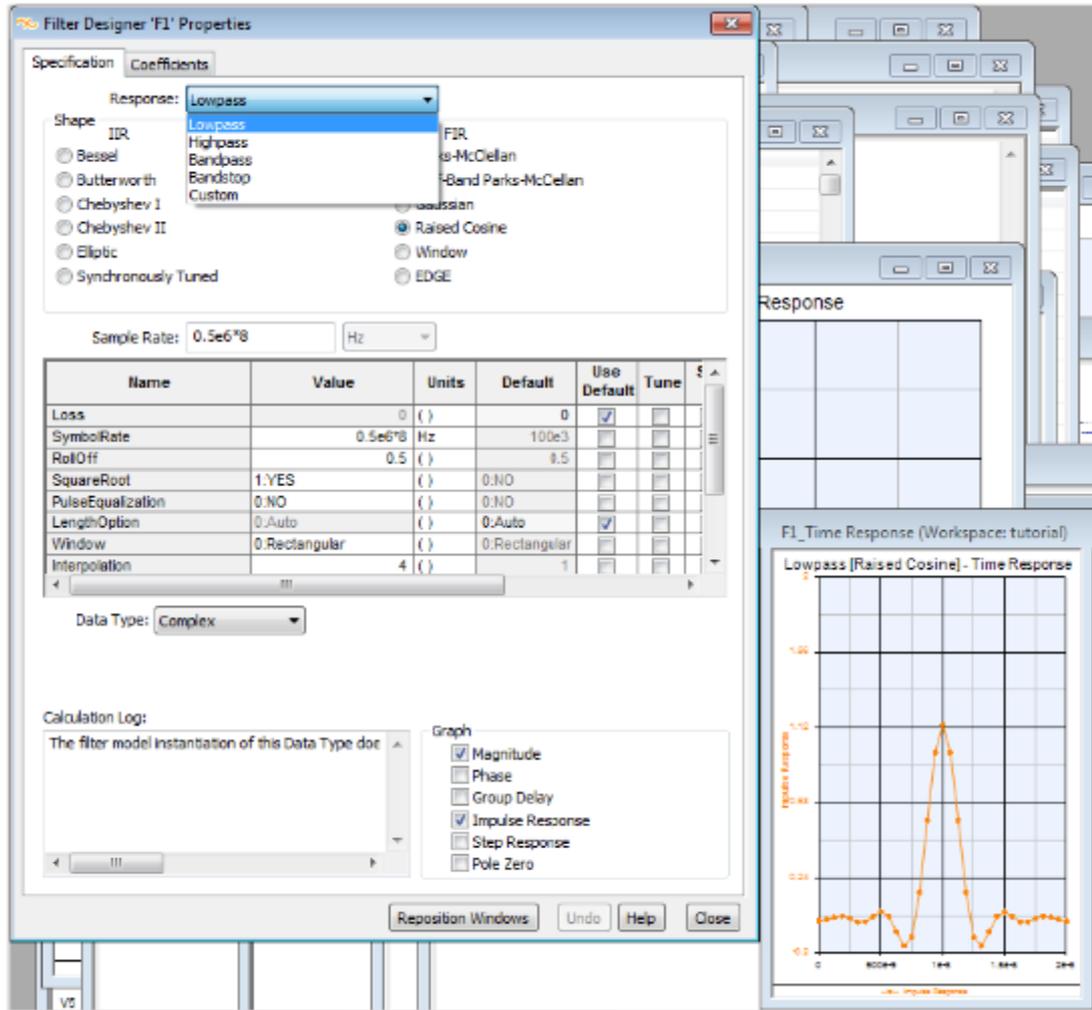


Рис. 9.19. Меню свойств проекта.

6. Запустите анализ.
7. Дважды щёлкните на наборе данных DF 1_Filter, щёлкните правой кнопкой на Walsh и выберите Addto Graph>NewGraphWizard>Spectrum (Добавить к диаграмме > Мастер новых диаграмм > Спектр). Дважды нажмите ОК. Появится экран, показанный на рис. 9.20.

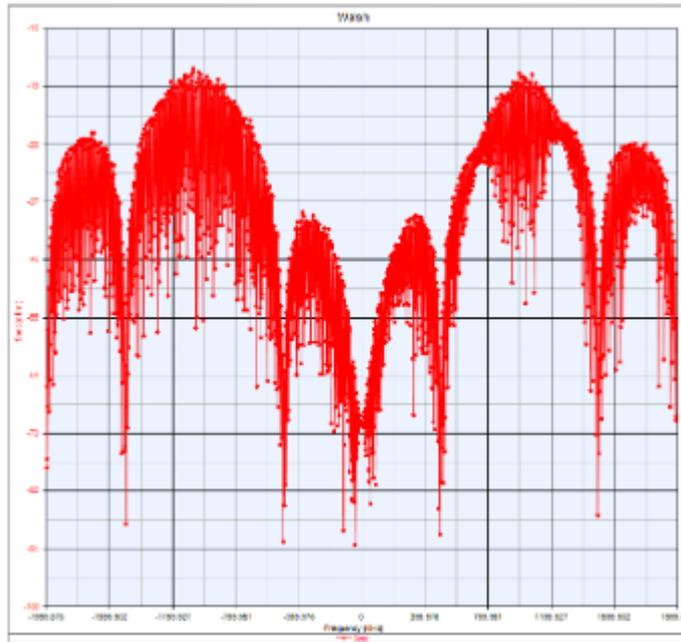


Рис. 9.20. Результирующая диаграмма.

8. Теперь постройте спектр для “Filter”. Дважды щёлкните на наборе данных фильтра (DF 1_Filter), щёлкните правой кнопкой на Filter и выберите Addto Graph>New Graph Wizard>Spectrum. Дважды нажмите ОК. Вы увидите, как это отразится на фильтре (рис. 9.21).

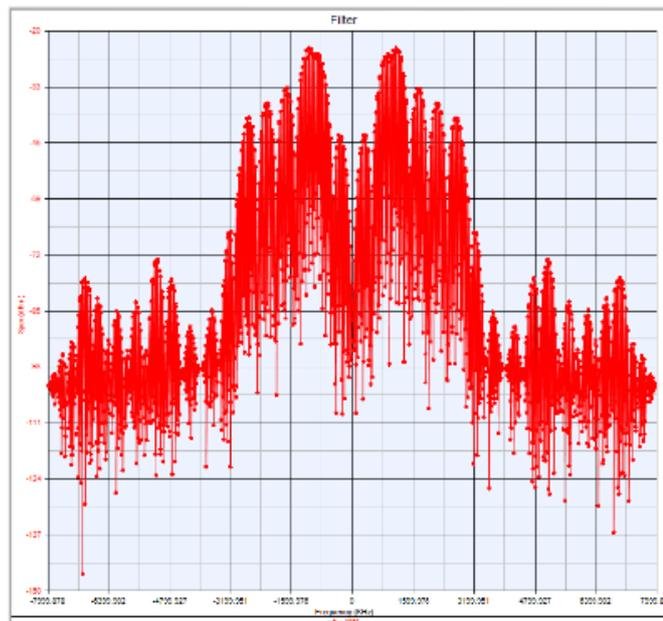


Рис. 9.21. Диаграмма спектра.

КОНЕЦ ЛАБОРАТОРНОЙ РАБОТЫ 4

ЛАБОРАТОРНАЯ РАБОТА 5. МОДУЛЯЦИЯ

Создайте новую папку и назовите её “Step 5_Modulation”. Добавьте новую схему с именем “Modulation” (модуляция). Добавьте анализ потока данных (“DF_Modulation”) и набор данных (“DF1_Modulation”). Установите число выборок анализа потока данных 4096.

1. Скопируйте схему, созданную на шаге 4, и вставьте её во вновь созданную пустую схему.
2. Удалите выходной приёмник.
3. В селекторе компонентов выполните поиск по ключевому слову “sxt” и выберите компонент “Sxt To Rect”. Он нужен для разделения реальной и мнимой части сигнала.
4. Присоедините этот компонент к выходу фильтра.
5. Добавьте модулятор (введите “modulator” для поиска в библиотеке селектора компонентов).
6. Присоедините его к выходу преобразователя комплексных чисел в реальную и мнимую часть.
7. Установите компонент плотности шума, напечатав в селекторе компонентов “noise density”. Подключите к выходу анализатор спектра.
8. Измените параметры компонентов, как показано на рис. 9.22.

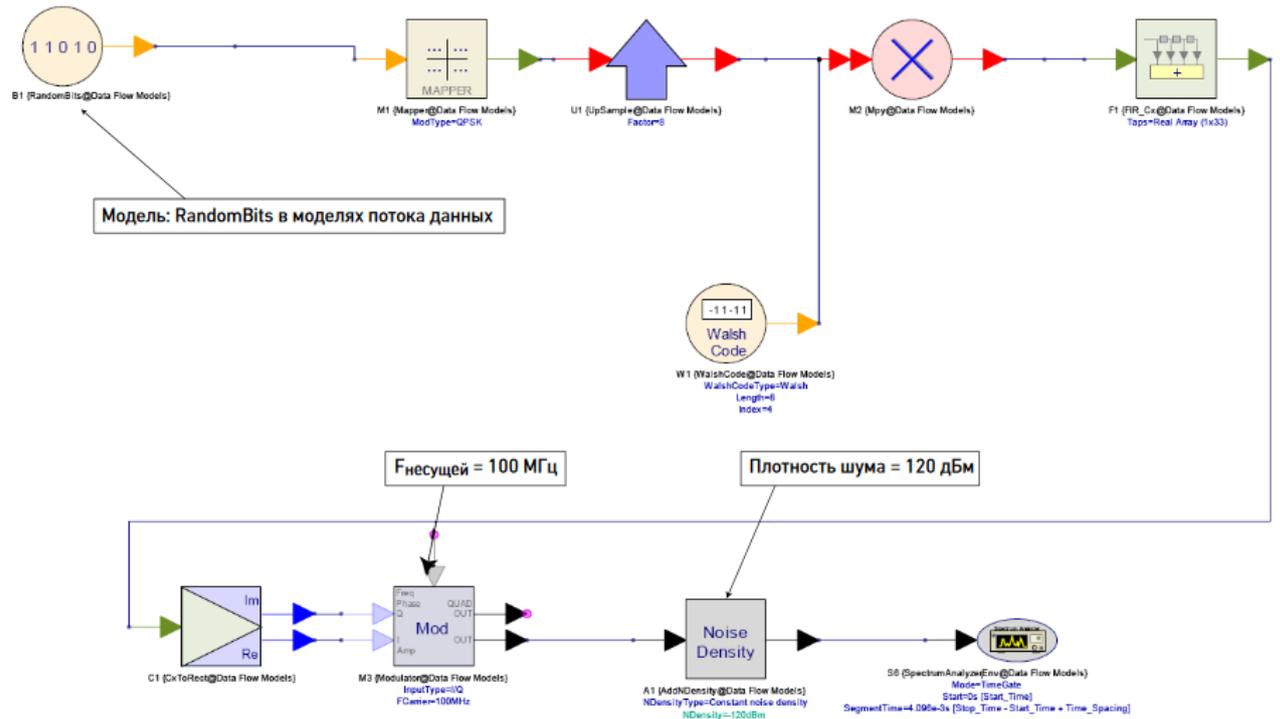


Рис. 9.22. Установите в схему компонент плотности шума.

9. Давайте рассмотрим спектр мощности в System Vue с помощью анализатора спектра. Дважды щёлкните на наборе данных “DF1_Modulation”, щёлкните правой кнопкой на ‘S 6_Power’ и выберите Add to the Graph>New Graph Series Wizard>General (Добавить в диаграмму > Мастер создания диаграмм > Общие). Дважды нажмите ОК (рис. 9.23).

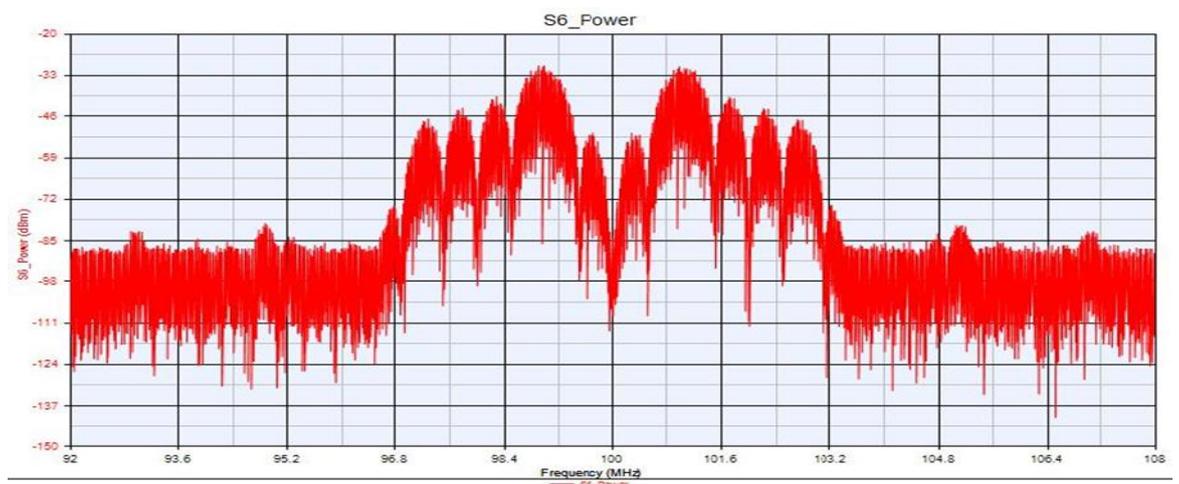


Рис. 9.23. Спектр мощности.

ЛАБОРАТОРНАЯ РАБОТА 5 ЗАВЕРШЕНА.

ЛАБОРАТОРНАЯ РАБОТА 6. ВЧ КАНАЛ

В этой лабораторной работе мы спроектируем ВЧ канал для передачи модулированного сигнала в приёмник.

1. Добавьте новую папку для шага 6. Назовите её “Step 6_RF Link”.
2. Создайте папку с именем “Step 6_RF Link” и вставьте в неё схему (RF Link).
3. В пустой схеме перейдите в селектор компонентов, прокрутите меню “Current Library” (текущая библиотека) и выберите “RF Design” (ВЧ проекты).
4. Оставьте категорию “All”, чтобы отображались все компоненты.
5. Создайте ВЧ канала в соответствии с приведённым ниже рисунком. Чтобы выбрать компоненты для схемы, введите в селекторе компонентов показанные на рис. 9.24 сокращения.

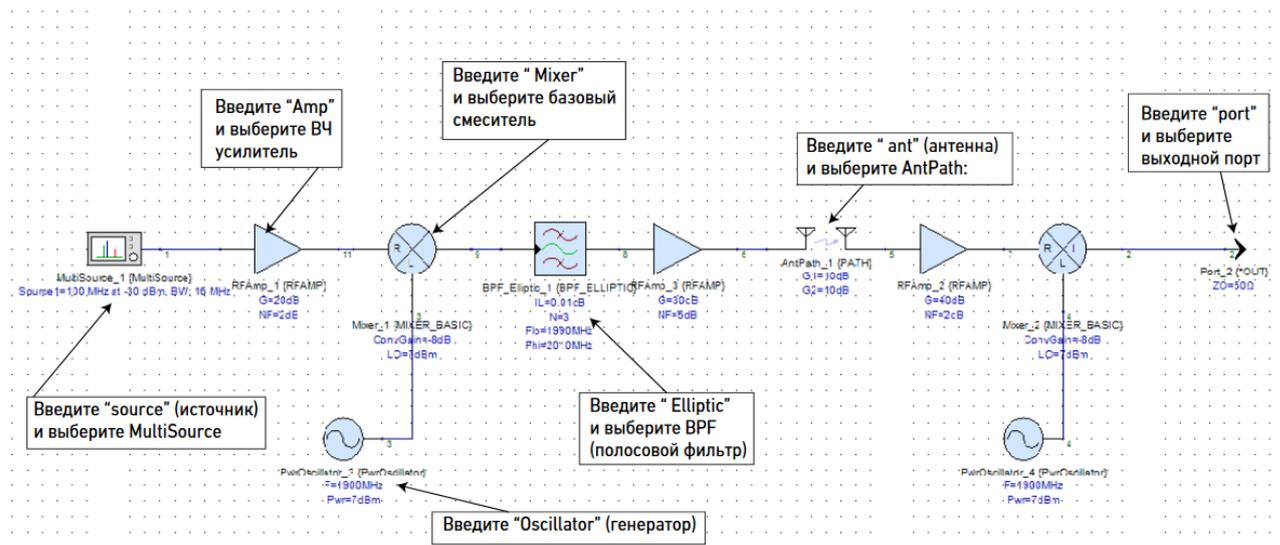


Рис. 9.24. Проект ВЧ канала.

С помощью колёсика мыши увеличьте масштаб страницы и установите параметры. Все параметры подробно описаны ниже.

Параметры:

-- Multi Source: дважды щёлкните на компоненте и в окне свойств источника щёлкните на “edit” (изменить). Убедитесь, что селектор установлен на “wide

band” (широкая полоса), и полоса равна 16 МГц. Установите центральную частоту 100 МГц и мощность –30 дБм. Оставьте фазу в 0 градусов.

-- ВЧ усилитель: дважды щёлкните на компоненте. Установятся значения по умолчанию. Измените коэффициент шума на 2 дБ. Дважды проверьте, что усиление равно 20 дБ.

-- Смеситель. По умолчанию должны быть установлены значения: усиление –8 дБм и гетеродин (LO) 7 дБм. Установите параметр SUM в значение 1:Sum.

-- Генератор: установите частоту 1900 МГц. Мощность по умолчанию должна равняться 7 дБм.

-- Полосовой фильтр: установите нижнюю частоту среза (F_{lo}) 1990 МГц, а верхнюю частоту среза (F_{hi}) 2010 МГц. Вносимые потери уже должны быть установлены на 0,01, а порядок фильтра на 3.

-- ВЧ усилитель на выходе полосового фильтра: установите усиление 30 дБ, а коэффициент шума 5 дБ.

-- Антенна: установите $G1=10$ дБ и $G2=10$ дБ.

-- ВЧ усилитель на выходе антенны: установите усиление 40 дБ, а коэффициент шума 2 дБ.

-- Смеситель после ВЧ усилителя: установите Conv Gain на -8 дБ и LO на 7 дБм. Кроме того, установите параметр SUM в значение 1:Sum.

-- Мощный генератор: установите частоту 1900 МГц и мощность 7 дБм.

Добавьте выходной порт и установите его сопротивление в 50 Ом.

6. Щёлкните правой кнопкой на папке Step 6_RF Link и выберите Add>Add data (для набора данных). Назовите набор данных “System 1_Data”.

7. Снова щёлкните правой кнопкой на папке Step 6_RF Link и выберите Add>Analysis>Add RF System Analysis (Добавить > Анализ > Анализ ВЧ систем).

8. Дважды щёлкните на анализе (рис. 9.25).

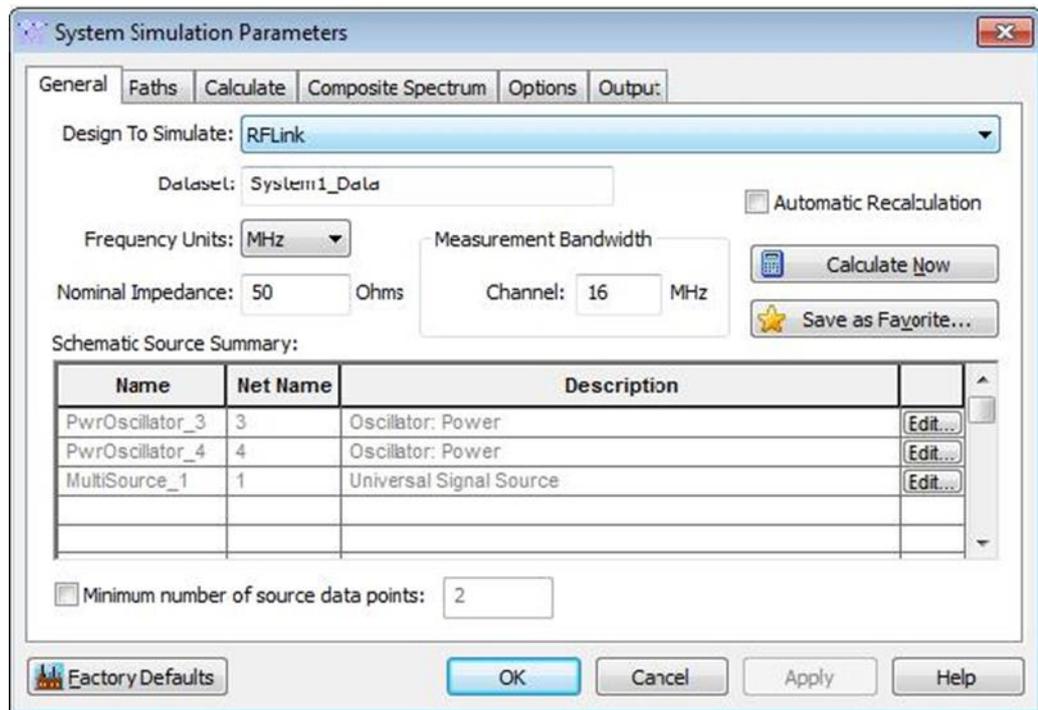


Рис. 9.25. Окно параметров моделирования системы.

На вкладке General (общие) убедитесь, что в поле моделируемого проекта (design to simulate) стоит значение RF Link, и что выбран набор данных (dataset) “System 1_Data”.

9. Перейдите на вкладку “Paths” (маршруты). Выберите “Add all paths from all sources” (добавить все маршруты из всех источников).

10. Нажмите OK.

11. Запустите моделирование, щёлкнув правой кнопкой на System1 (RF Link) и выбрав Run. Вы можете обнаружить, что создана дополнительная папка “System1_Data_Folder”.

На схеме наведите курсор на значения. Для начала давайте рассмотрим значение на выходе мульти источника. Если навести курсор на значение, то рядом со стрелкой появится прямоугольный курсор.

12. Щёлкните правой кнопкой (рис. 9.26).

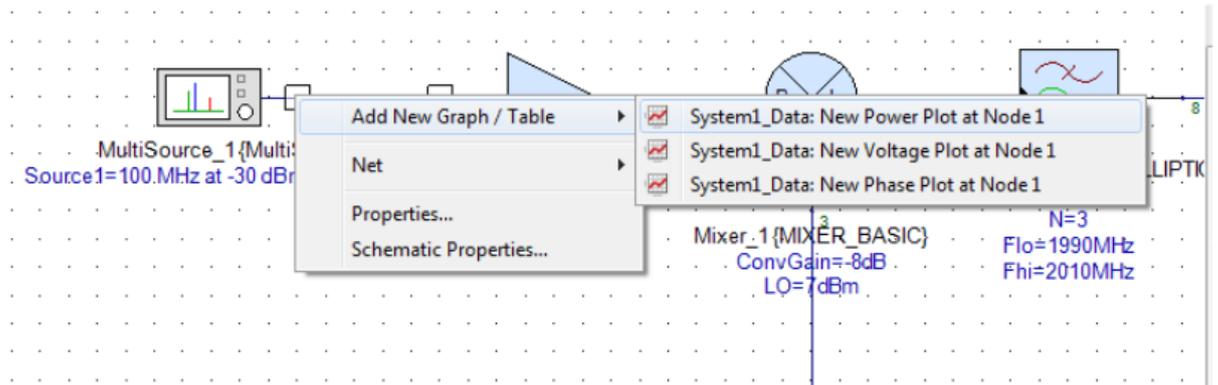


Рис. 9.26. На схеме выберите сначала значение на выходе мульти источника.

13. Выберите **Add New Graph/Table>Select** (Добавить новую диаграмму/таблицу > Выбрать), чтобы построить диаграмму мощности (рис. 9.27).

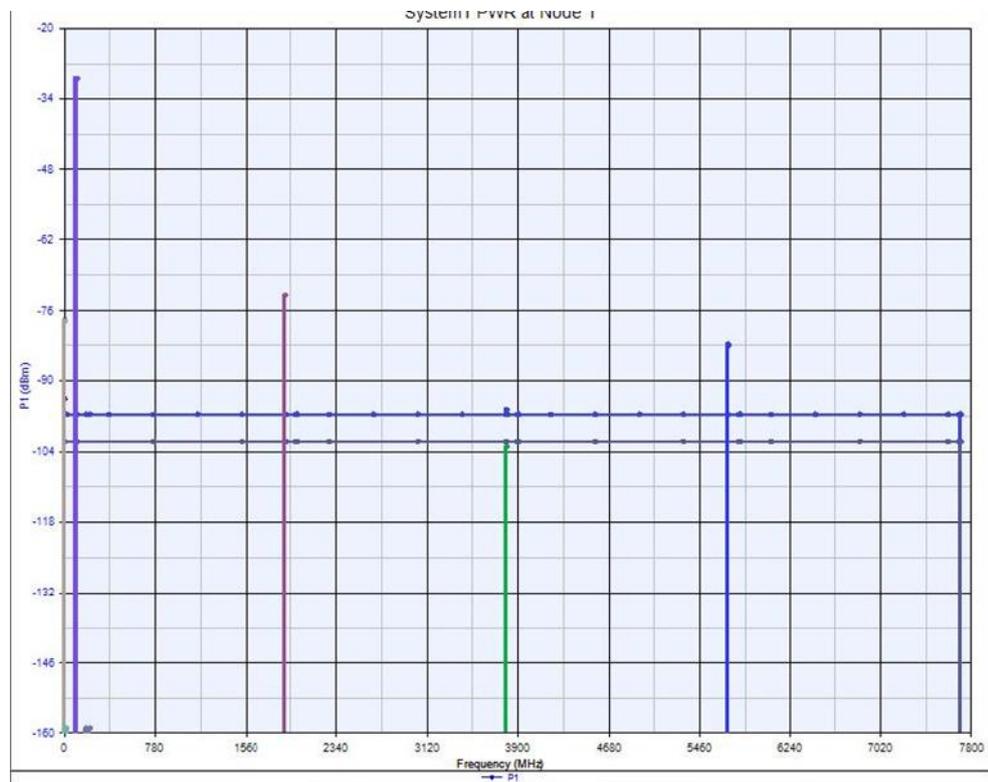


Рис. 9.27. Диаграмма мощности.

Здесь мы можем наблюдать, насколько чист сигнал нашей системы, и присутствуют ли источники помех. Сигнал данных занимает частоту 100 МГц (заданную нашим источником). Это центральная частота, и установлена она достаточно высоко для ВЧ канала связи. Кроме того, мы видим на этой диаграмме другие частотные составляющие и можем подтвердить, что сигналы, создающие помехи для наших данных, отсутствуют.

14. Отобразите мощность на выходном порту (рис. 9.28).

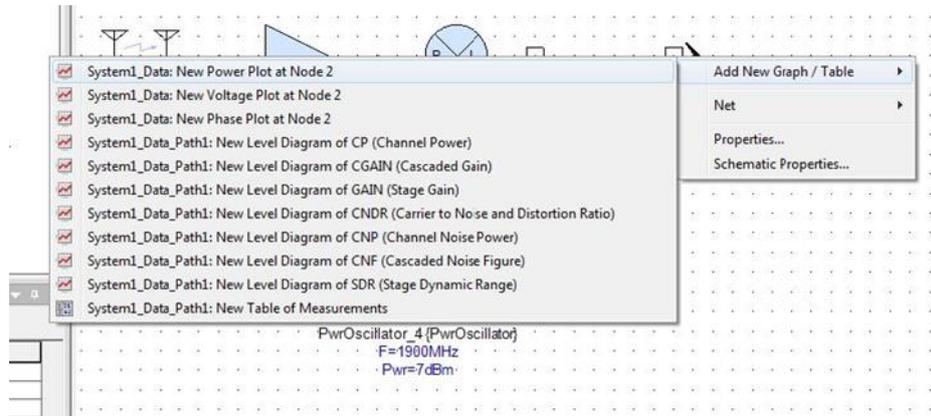


Рис. 9.28. Отображение сигнала на выходном порту.

Постройте новую диаграмму мощности (рис. 9.29).

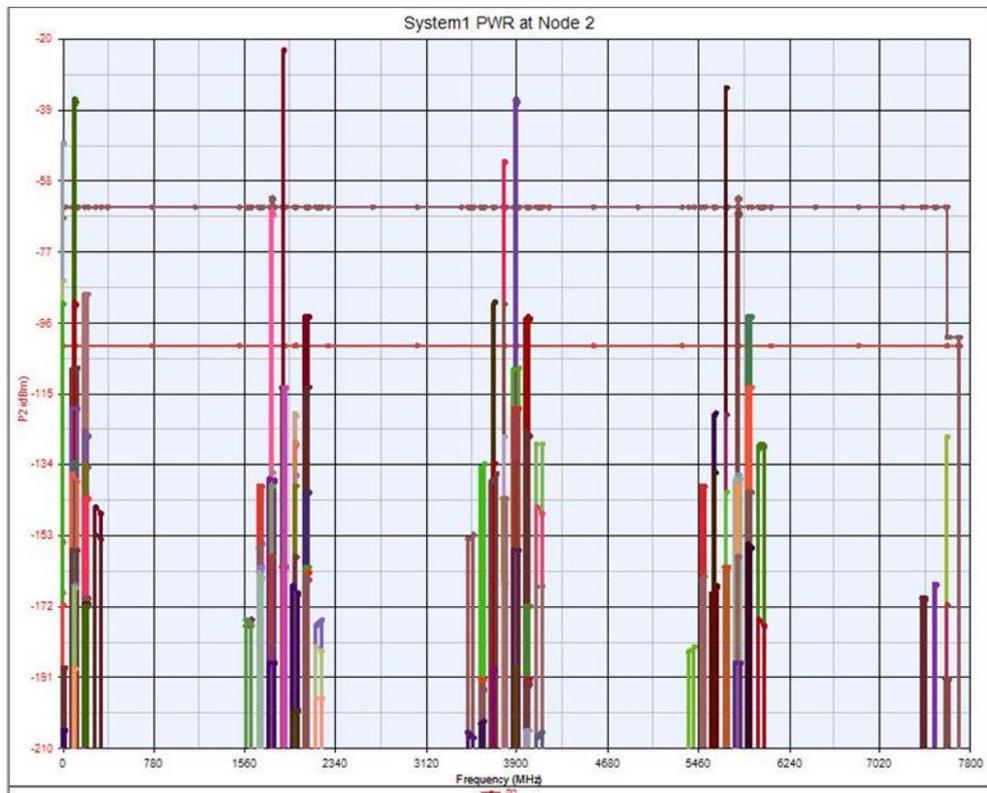


Рис. 9.29. Диаграмма мощности.

Мы можем наблюдать уровень нашего сигнала на частоте 1900 МГц, которая является центральной частотой. Вокруг частоты несущей видны гармоники. Кроме того, мы видим другие частотные составляющие на верхних и нижних частотах, демонстрирующие реальные условия системы передачи. Также мы можем заключить, что после передачи по каналу связи уровень нашего сигнала

ещё достаточно высок и, следовательно, мы можем выполнить его демодуляцию и извлечение исходных передаваемых данных без ошибок.

НА ЭТОМ ЛАБОРАТОРНАЯ РАБОТА 6 ЗАВЕРШЕНА.

ЛАБОРАТОРНАЯ РАБОТА 7. ДЕМОДУЛЯТОР

В этой лабораторной работе мы демодулируем сигнал и спроектируем приёмную часть системы.

1. Создайте новую папку с именем “Step 7_Demodulation”. Добавьте новую пустую схему и назовите её “Demodulation” (демодуляция). Добавьте анализ потока данных с именем “DF_Demodulation”. Создайте набор данных с именем “DF1_Demodulation”.

2. Скопируйте проект, созданный на этапе 5 Step5_Modulation, и вставьте его в новую пустую схему “Demodulation”.

3. В дереве рабочей среды разверните папку “Step 6_RF Link”. Щёлкните на схеме RF Link и перетащите её в любое место текущей схемы “Demodulation”.

Замечательной функцией System Vue является возможность изменять имена цепей на те, что вам нужны, выполнять соединения без физических проводов, не нарушая организации проекта.

4. Щёлкните на соединении между “Noise density” (плотность шума) и “Spectrum Analyzer” (анализатор спектра) (рис. 9.30).

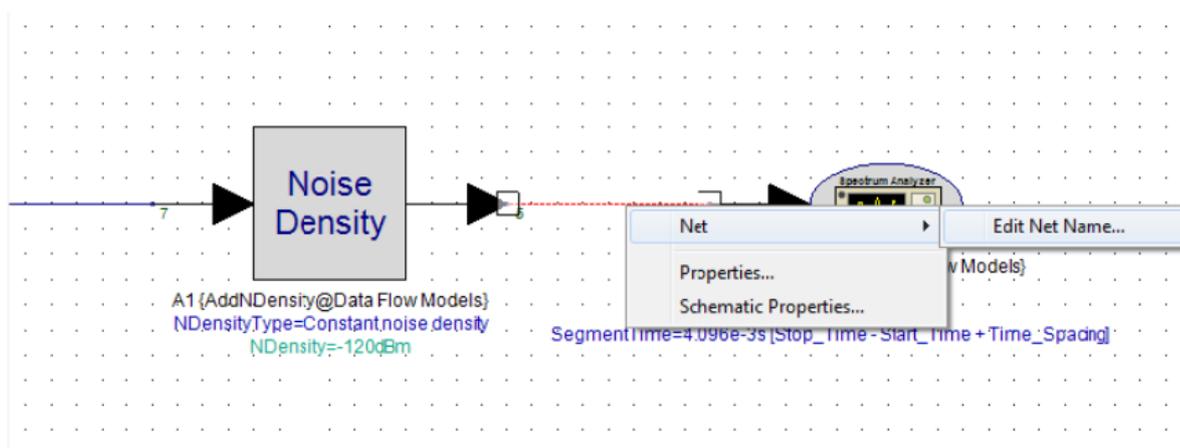


Рис. 9.30. Измените имя цепи.

5. Щёлкните правой кнопкой на соединении и выберите Net>Edit Net Name (Цепь > Изменить имя цепи). Введите новое имя “IF_in”.

Слева от RF_Link создайте соединение, позволяющее нам выделить эту цепь, и дайте ей имя (рис. 9.31).

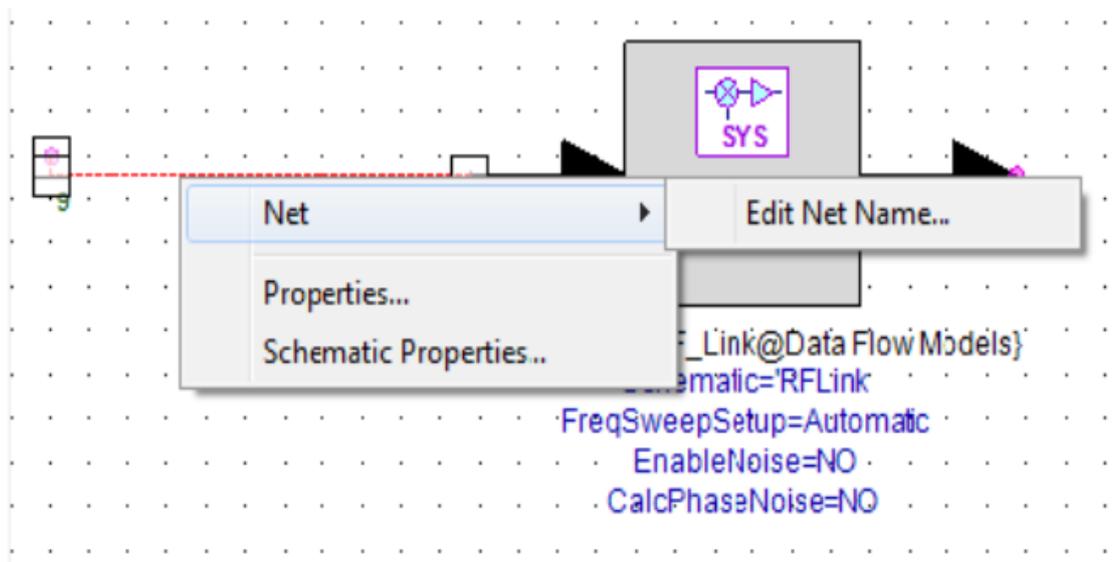


Рис. 9.31. Создайте соединение, выделите цепь и дайте ей имя.

Дайте соединению то же имя, что и выходному соединению компонента плотности шума (“IF_in”). Это соединит оба компонента между собой. Теперь компонент плотности шума и RF_Link соединены.

6. Создайте систему демодуляции. В сущности, эта система зеркальна уже созданной системе модуляции.

7. Добавьте после RF_Link анализатор спектра. Вам может понадобиться изменить “Current Library”(текущая библиотека) в селекторе компонентов на Algorithm Design (конструктор алгоритмов). Мы будем наблюдать снижение мощности после передачи. Скопируйте соответствующие компоненты из Step5_Modulation, и их параметры тоже сохранятся (рис. 9.32).

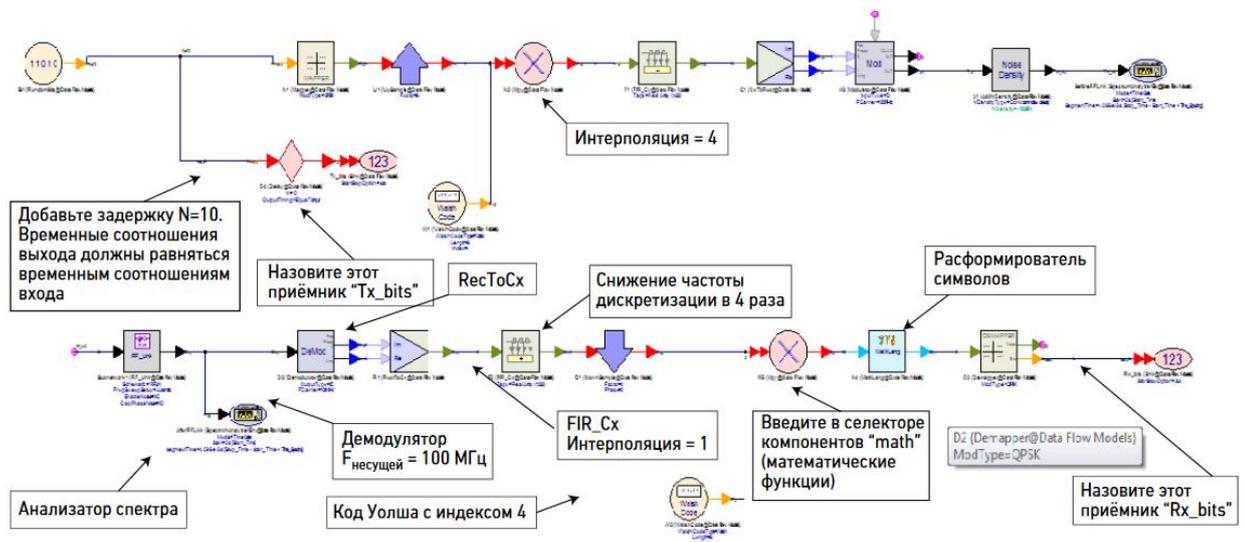


Рис. 9.32. Добавление анализатора спектра.

Модель Math Lang использует для обработки входных данных язык математических уравнений и создаёт выходные данные. Мы можем использовать это для поддержания нужной частоты дискретизации при расформировании символов.

8. Дважды щёлкните на компоненте Math Lang и на вкладке Equations (уравнения) скопируйте то, что показано на рис. 9.33 ($output = \text{sum}(input)/8$).

На вкладке “I/O” (ввод/вывод) установите число входных портов равным 8 (число выходных портов оставьте равным 1).

9. Щёлкните на ОК.

10. Запустите моделирование, следя за тем, чтобы анализ потока данных получил корректные значения (рис. 9.34).

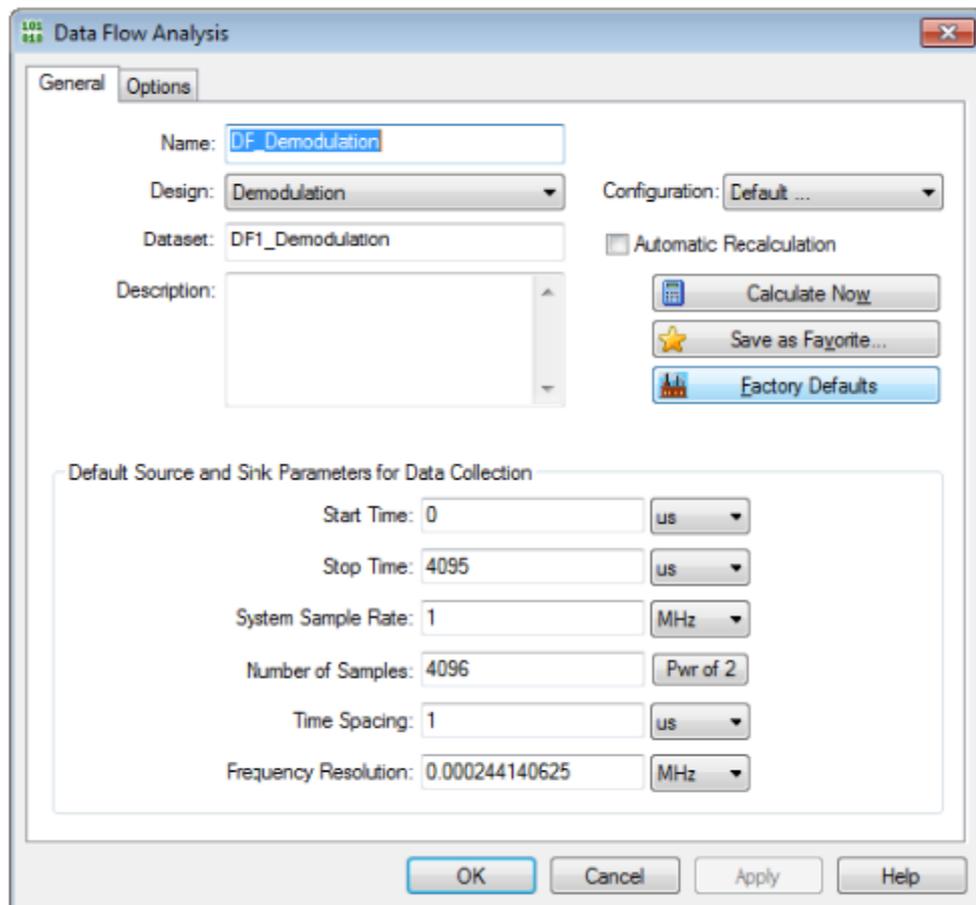


Рис. 9.34. Используйте для моделирования показанные здесь значения.

Давайте рассмотрим мощность на анализаторе спектра до и после канала RF Link, по которому передаётся поток данных.

11. Щёлкните на наборе данных и создайте диаграмму спектра мощности сигнала до RFLink. Дважды щёлкните на наборе данных “DF1_Demodulation” и, щёлкнув правой кнопкой на ‘S6_Power’, выберите Add to the Graph>New Graph Series Wizard>General. Дважды нажмите на ОК (рис. 9.35).

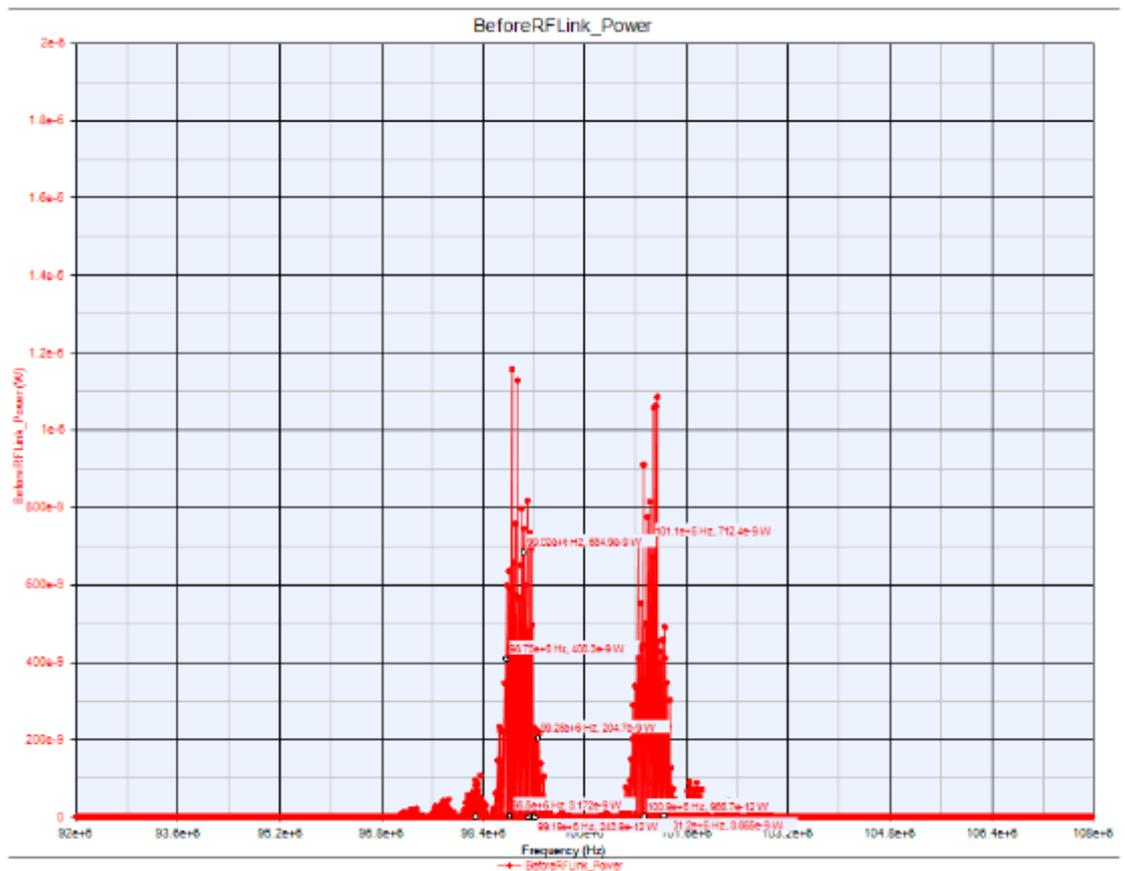


Рис. 9.35. Спектр мощности до RFLink.

12. Теперь давайте рассмотрим спектр мощности после прохождения потока данных по ВЧ каналу (рис. 9.36).

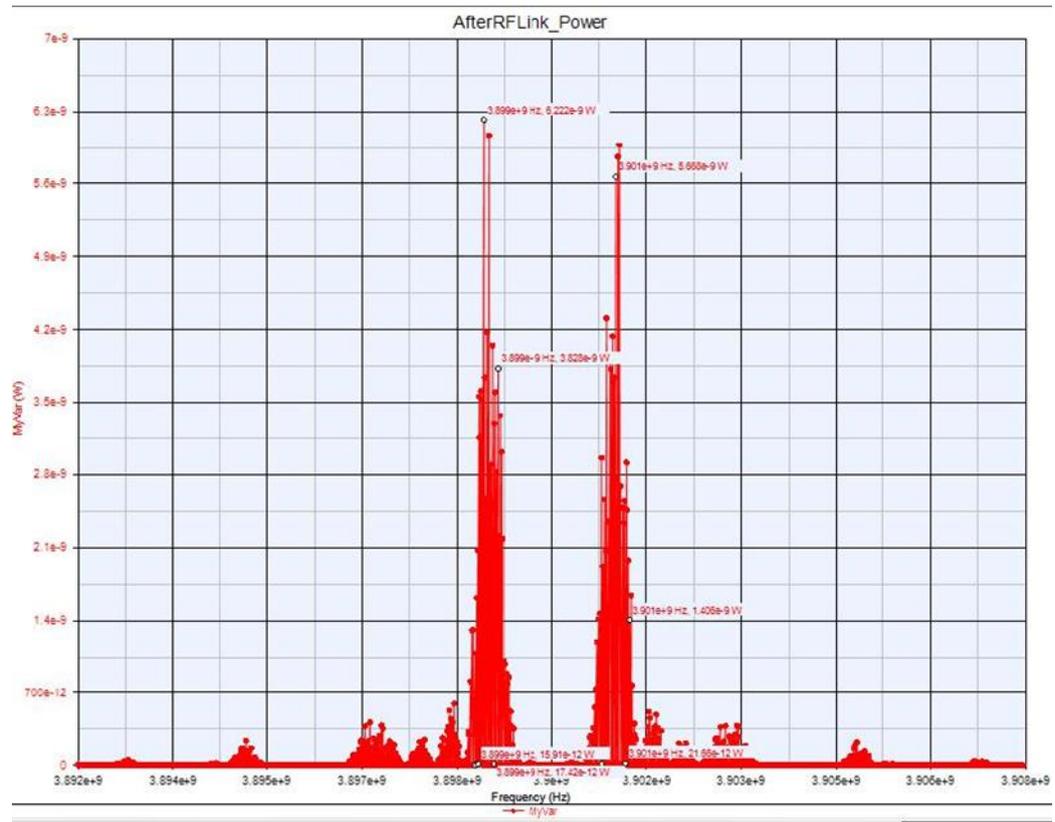


Рис. 9.36. Спектр мощности после RF Link.

Как видите, мощность на частоте 10 МГц, после того, как несущая промодулирована нашими данными, равна примерно $1,18 \cdot 10^{-6}$ Вт. После прохождения сигнала через ВЧ канал мощность падает примерно до $6,38 \cdot 10^{-9}$ Вт. Эта потеря мощности имитирует реальные условия распространения сигнала по эфиру между передающей и приёмной антеннами. При этом мощность сигнала может теряться из-за деревьев, зданий и других объектов ландшафта. Кроме того, мы видим боковые полосы. Можно смело утверждать, что они не влияют на данные на стороне приёма, поскольку код Уолша реконструирует информацию, превышающую уровень шумов.

Теперь давайте исследуем отдельные биты и посмотрим, насколько хороша связь.

13. В наборе данных вместо диаграммы откройте табличное представление переданных битов и сравните их с принятыми битами (Rx_bits и Tx_bits), как показано на рис. 9.37.

Если вам нужны пошаговые инструкции, вернитесь к лабораторной работе 3, где мы создавали таблицу для просмотра кодов Уолша.

Index	Rx_bits	Ix_bits
1	0	0
2	1	0
3	1	0
4	0	0
5	1	0
6	0	0
7	1	0
8	0	0
9	0	0
10	1	0
11	0	0
12	1	1
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	1	1
28	0	0
29	0	0
--	-	-

Рис. 9.37. Сравнение переданных и принятых битов.

По принятым битам можно заключить, что система работает хорошо.

ЛАБОРАТОРНАЯ РАБОТА 7 ЗАВЕРШЕНА

ЛАБОРАТОРНАЯ РАБОТА 8. ПОЛНАЯ СИСТЕМА

В этой лабораторной работе мы протестируем характеристики системы, рассчитав коэффициент битовых ошибок (BER).

1. Добавьте новую папку с именем “Step 8_CompletedSystem”.

2. В папке создайте схему “Completed System” (полная система), создайте набор данных “DF1_Completed System” и анализ потока данных “DF_Completed System” (установите число выборок 4096).

3. Скопируйте компоненты схемы, созданной на этапе 7, и вставьте их в новую пустую схему.

Давайте сопоставим два наших сигнала. Это позволит сравнить сигналы и оценить степень их подобия. В нашей системе CDMA мы сравним отправленный код с принятым кодом, полученным с помощью генератора ко-дов Уолша. Если коды подобны, то приёмник их декодирует и воспроизведёт исходный переданный сигнал.

4. Установите в свободном месте схемы кросс-коррелятор, для чего введите в селекторе компонентов текст “cross” и выберите компонент кросс-коррелятора.

5. Добавьте два приёмника – один с задержкой (назовите его “Delay_Estimate”), а второй с именем “Cross_Estimate”.

6. Добавьте к каждому входу по соединению и измените их имена. Вход Y назовите “Test” (тест), а вход X – “Ref” (опорный). Для этого дважды щёлкните на соединении и измените стандартное имя (рис. 9.38).

7. Соответственно назовите выходное соединение генератора случайных битов “Ref”, а выходное соединение системы – “Test” (рис. 9.38).

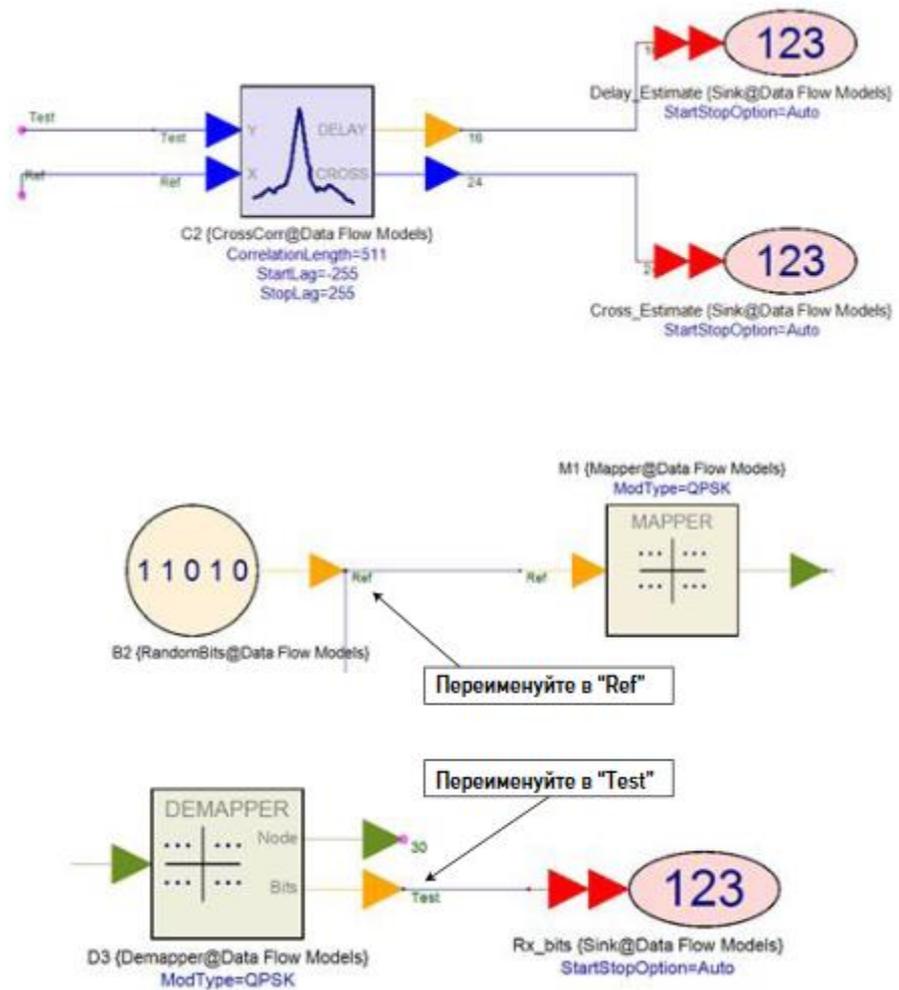


Рис. 9.38. Добавление соединения к каждому входу и изменение их имён.

8. Запустите моделирование.
9. Постройте график задержки, в виде графика кросскорреляции (Dataset>Delay_Estimate>New Series Graph Wizard>Cross Correlation).
10. Теперь постройте график перекрёстной оценки (рис. 9.39).

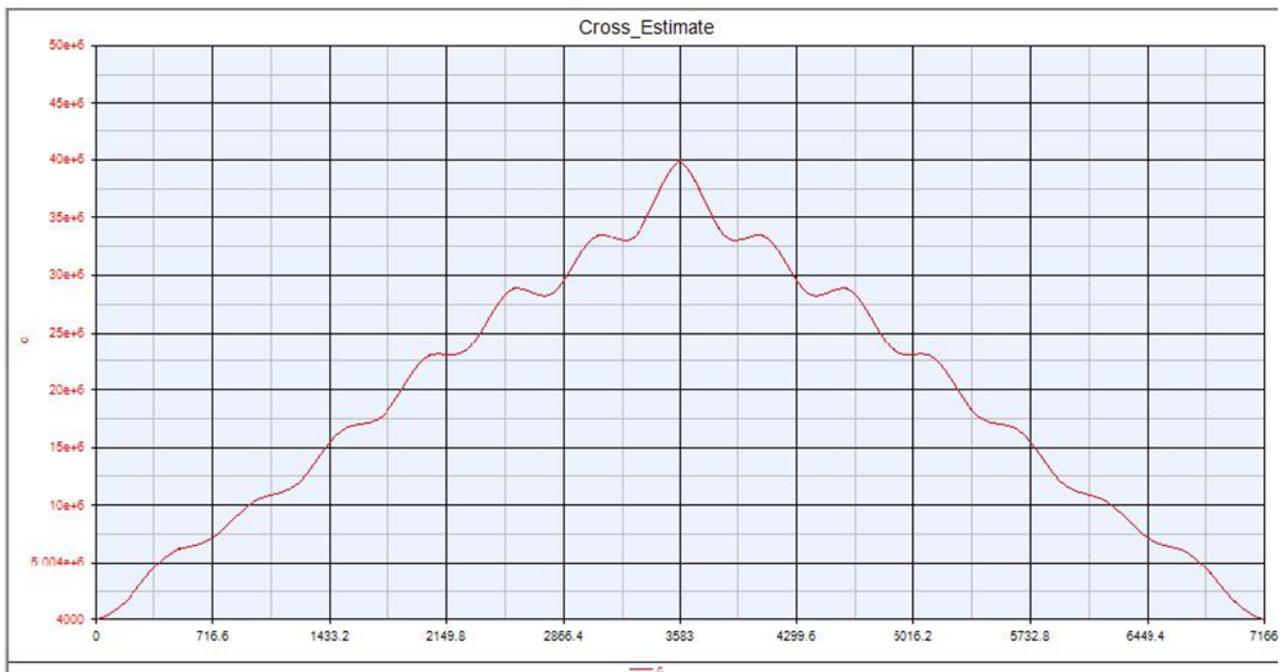


Рис. 9.39. График перекрёстной оценки.

Мы можем представить, что переданный сигнал и сигнал, сохранённый в приёмнике (с помощью генератора кода Уолша), скользят один по-другому, и в это время коды становятся все более совпадающими, и пики кросскорреляции образуют график, напоминающий треугольник. Пик представляет смещение, при котором коды почти идеально совпадают.

Теперь давайте проверим характеристики системы, применив метод BER. BER используется для оценки параметров системы связи, поскольку он показывает интенсивность появления ошибок в системе передачи. BER определяется уравнением E.9.1:

$$BER = \text{Число ошибочных битов} / \text{Общее число переданных битов} \quad E.9.1$$

Если среда между передатчиком и приёмником обладает хорошими характеристиками, то отношение сигнала к шуму будет большим, и BER будет малым. Если система обладает большим уровнем шумов, и методы кодирования выбраны неправильно – в нашем случае схема кодирования Уолша, – то принимаемые биты будут отличаться от исходно переданных битов, что приведёт к росту BER.

1. Найдите в схеме свободное место.

2. Введите в селекторе компонентов “BER” и вставьте в проект соответствующий компонент. Параметр Sample Start модели BER_FER должен быть установлен в 10, Bits Per Frame в 1000, а Start Stop Option = Samples.

3. Добавьте по входу REF задержку $N=10$. Эта задержка обеспечивает синхронизацию битовых потоков, без которой значение BER будет неверным.

4. Добавьте к обоим входам соединения, чтобы мы могли дать им имена. Подключите вход TEST к сигналу TEST. Для этого можно назвать соединение “TEST”. Другому соединению на входе REF дайте имя “REF”. Это позволит измерителю BER рассчитать коэффициент битовых ошибок передаваемого сигнала и принимаемого сигнала, после декодирования его с помощью кода Уолша (рис. 9.40).

5. Запустите моделирование.

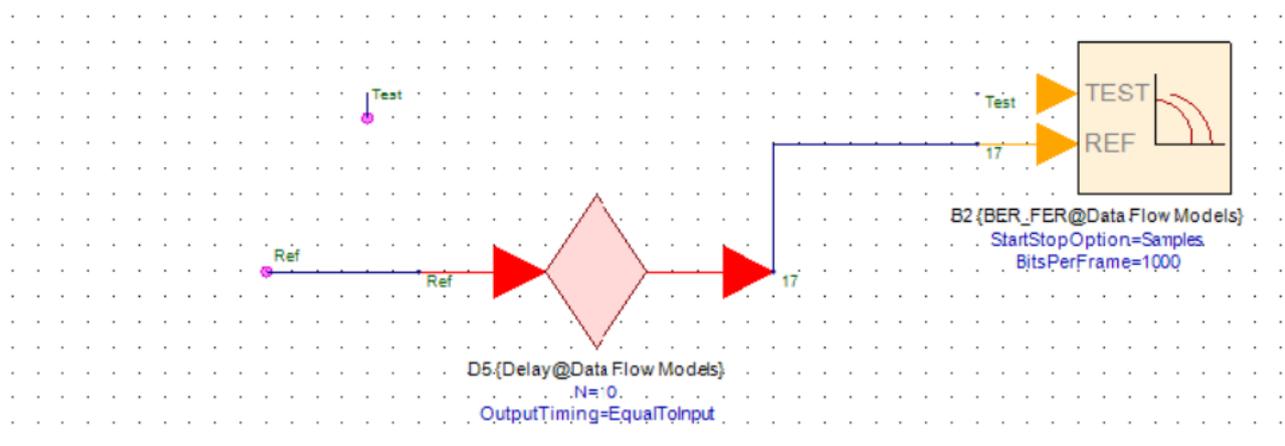


Рис. 9.40. Рассчитайте BER переданного и принятого сигнала с помощью измерителя BER.

Теперь давайте настроим уравнения BER и выведем наши результаты в текстовое окно.

6. Щёлкните в дереве рабочей среды на уравнении 1 (рис. 9.41). В поле кодирования введите следующий текст (B3_BER представляет переменную набора данных BER):

- `using('DF1_CompletedSystem');`
- `BER = B3_BER;`

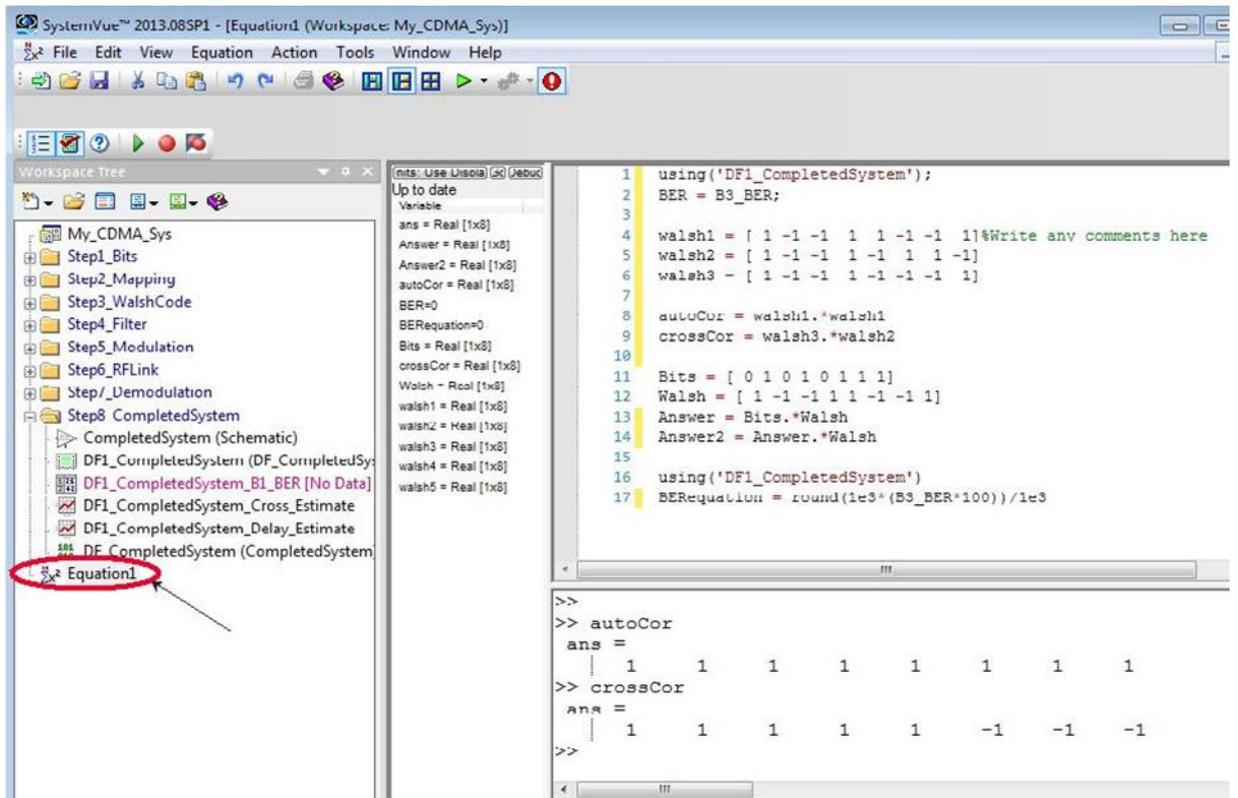


Рис. 9.41. Вставка уравнения BER.

7. Щёлкните на “Show/Hide Annotation Toolbar” (Показать/скрыть панель аннотаций) в верхнем углу приложения System Vue (рис. 9.42).

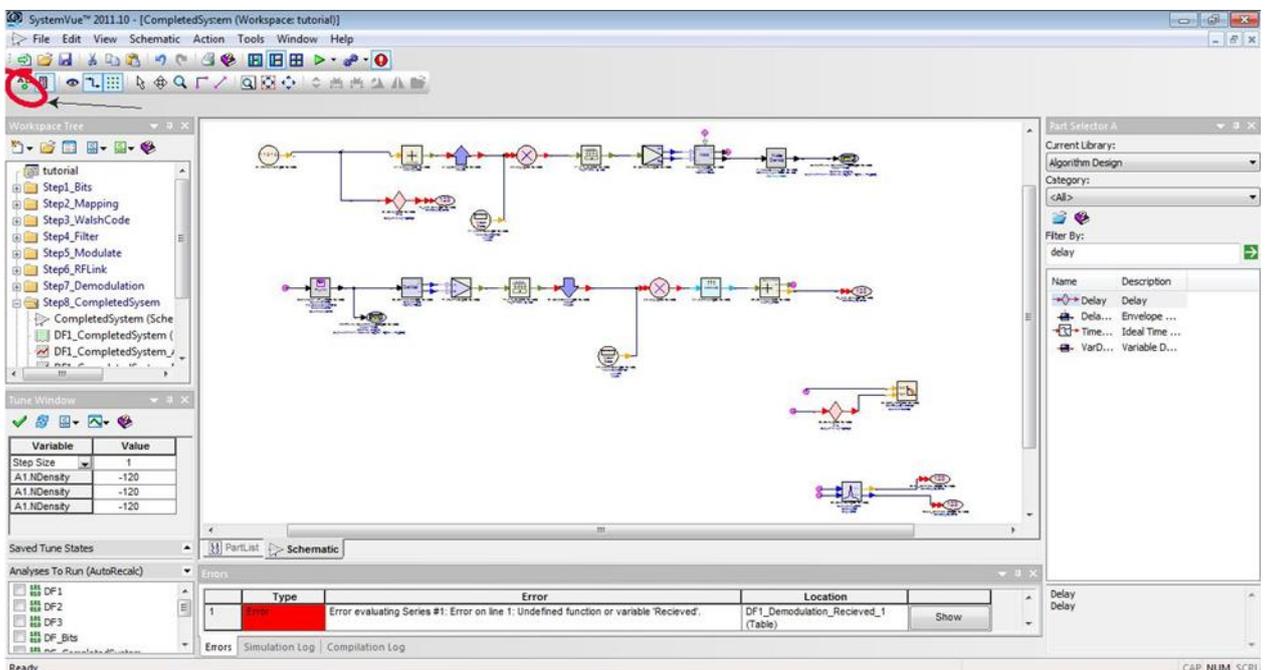


Рис. 9.42. Показать/скрыть панель аннотаций в SystemVue

Появится панель инструментов. Щёлкните на квадратном текстовом значке, как показано на рис. 9.43.



Рис. 9.43. Квадратный текстовый значок

Создайте два текстовых поля и расположите их бок о бок. Одно из них отображает наше значение BER, а второе – имя переменной (рис. 9.44).

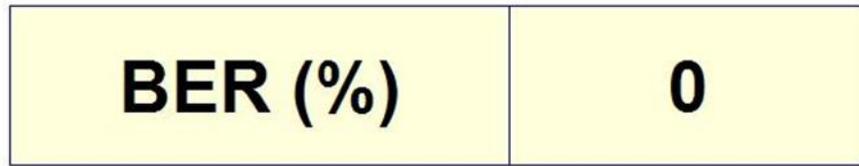


Рис. 9.44. Два расположенных бок о бок текстовых поля.

8. Дважды щёлкните на правом текстовом поле и введите “=BER”, как показано на рис. 9.45.

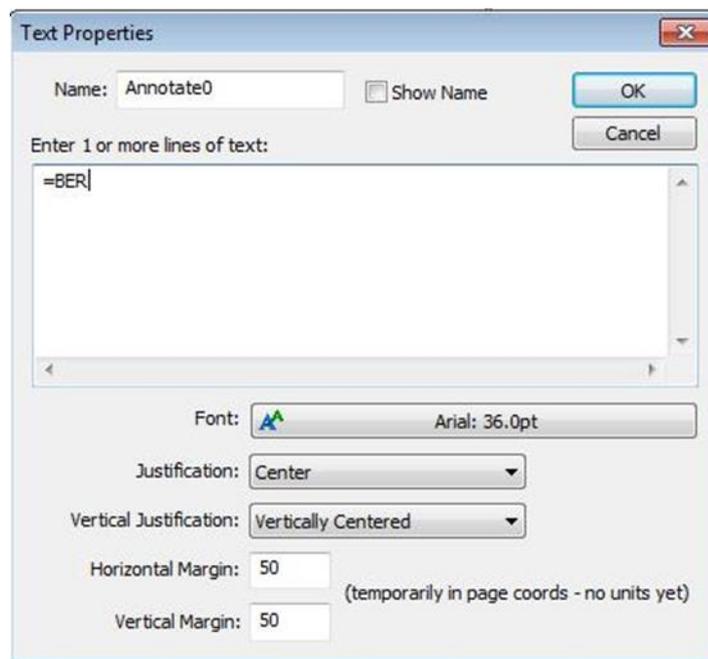


Рис. 9.45. Страница свойств текста.

Здесь вы можете изменить положение текста, его размер, шрифт и цвет.

9. Теперь дважды щёлкните в левом текстовом поле и введите текст “BER (%)”.

Если индекс генератора кода Уолша передатчика совпадает с индексом кода Уолша приёмника, то после щелчка на ОК, BER должен равняться 0 %. Это показывает, что характеристики системы идеальны, и все переданные данные успешно декодированы кодом Уолша и приняты.

Давайте изменим индекс одного из кодов Уолша и посмотрим, что произойдёт.

10. Щёлкните на одном из генераторов кода Уолша и измените его индекс на любое значение до 8 (отличное от того, что было установлено).

11. Запустите моделирование.

Обратите внимание, что BER станет больше 0 %. Попробуйте изменить индекс Уолша, ещё раз запустите моделирование и наблюдайте изменения BER.

Мы завершили построение системы CDMA с распределённым спектром и протестировали её характеристики с помощью BER в САПР System Vue.

В качестве последнего упражнения давайте используем уравнения System Vue для наблюдения реальных битов, кодов Уолша и альтернативного способа отображения BER. В этом упражнении для обработки выходных данных будут использоваться уравнения на языке описания математических формул.

1. В дереве рабочей среды щёлкните правой кнопкой на “tutorial” (учебник) и выберите Add>Add Equation (Добавить > Добавить уравнение).

Давайте создадим некоторые коды Уолша.

2. В окне кода введите следующий текст, начиная со строки 1 (рис. 9.46).

```
1 walsh1 = [ 1 -1 -1 1 1 -1 -1 1]  %Здесь можно написать комментарии
2 walsh2 = [ 1 -1 -1 1 -1 1 1 -1]
3 walsh3 = [ 1 -1 -1 1 1 -1 -1 1]
```

Рис. 9.46. Знак “%” позволяет записывать комментарии

Обратите внимание, что каждый код Уолша имеет равное число единиц и единиц с минусом.

Обратите внимание, что переменные устанавливаются в левой части дерева.

3. Нажав Go (пуск), скомпилируйте код. Эту кнопку надо нажимать после создания новой переменной.

4. В окне команд (в нижней части экрана) введите любую переменную кода Уолша, например, Walsh1. Заметьте, что команды чувствительны к регистру букв.

```
>> walsh1
      ans =
         1    -1    -1     1     1    -1    -1     1
>>
```

Рис. 9.47. Введите переменную кода Уолша, например, Walsh1.

Теперь давайте применим метод корреляции.

5. Примените автокорреляцию к Walsh1, введя в окно кода следующее уравнение – `auto Cor = walsh1.*walsh1`

6. Выполните кросскорреляцию walsh3 и walsh2, введя в окно кода следующее уравнение – `cross Cor = walsh3.*walsh2`

Наблюдайте соответствующие значения в окне команд (рис. 9.41). Обратите внимание, что автокорреляция создаёт один объект из единиц. Здесь мы можем доказать, что автокорреляция, эквивалентная кросскорреляции с таким же кодом, означает, что данные можно декодировать, и информацию можно извлечь (в связи с полным совпадением). Кросс-корреляция двух разных кодов Уолша даёт рассогласование, поскольку создаётся серия единиц и единиц с минусом. Из-за рассогласования данные декодировать нельзя. Таким образом, мы увидели и доказали, что для декодирования и получения исходных данных, генератор кода Уолша на стороне передатчика должен генерировать такой же код Уолша, что и на стороне приёмника.

Если создать битовую последовательность, то можно увидеть, что происходит с данными, когда они шифруются кодом Уолша (рис. 9.48).

```
Bits = [ 0 1 0 1 0 1 1 1]
Walsh = [ 1 -1 -1 1 1 -1 -1 1]
Answer = Bits.*Walsh
```

Рис. 9.48. Шифрование данных кодом Уолша.

Проверьте результирующий массив с помощью окна команд (рис. 9.49).

```
>> Answer
      ans =
         0    -1    -0     1     0    -1    -1     1
>>
```

Рис. 9.49. Результирующий массив.

Если теперь мы возьмём этот код и сопоставим его с нашим кодом Уолша (“Walsh1”), мы сможем получить исходную битовую последовательность (рис. 9.50).

```
Bits = [ 0 1 0 1 0 1 1 1]
Walsh = [ 1 -1 -1 1 1 -1 -1 1]
Answer = Bits.*Walsh
Answer2 = Answer.*Walsh
```

Рис. 9.50. Исходная битовая последовательность.

Проверьте результирующий массив с помощью окна команд (рис. 9.51).

```
>> Answer
      ans =
           0           1           0           1           0           1           1           1
>>
```

Рис. 9.51. Результирующий массив.

Используя уравнения, мы можем наблюдать результаты расчёта BER прямо в окне команд с помощью следующего кода:

```
using('DF_ompletedSystem')
BER = round(1e3*(B2_BER*100))/1e3
```

Это даст нам значение BER в процентах.

7 ЗАКЛЮЧЕНИЕ

Надеемся, упражнение по построению системы CDMA с распределённым спектром было полезным, и вы познакомились со всеми необходимыми инструментами для разработки, моделирования, отображения и анализа результатов – графиками, глазковыми диаграммами, спектрами, диаграммами мощности, графиками в частотной области и BER. Эта информация должна подготовить вас к созданию более сложных систем.

Дополнительные навыки работы с САПР System Vue можно получить с помощью обширной библиотеки примеров, учебных видеороликов и интернет-форумов, доступ к которым предоставляется через меню справки.

СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ

1. Официальный сайт продукта Simulink (производитель The Math Works, Inc.) – <http://www.mathworks.com/products/simulink/>
2. Официальный сайт продукта Sys Calc (производитель ArdenTechnologies, Inc.) – <http://www.ardentech.com/>
3. Официальный сайт продукта System Vue (производитель AgilentTechnologies) – <http://www.agilent.com/find/eesof-systemvue>
4. Официальный сайт продукта LabVIEW (производитель National Instruments) – <http://www.labview.ru/>
5. Официальный сайт продукта Hypersignal RIDE (производитель National Instruments) – <http://www.ni.com/support/ride.htm>
6. Официальный сайт продукта VisSim (производитель Visual Solutions) – <http://www.vissim.com/>
7. Официальный сайт продукта Visual System Simulator (производитель AWR Corporation) – <http://www.awrcorp.com/ru/products/visual-system-simulator>
8. Официальный сайт продукта Model Builder (производитель Agilent Technologies) – <http://www.agilent.com/find/eesof-mbp>
9. Официальный сайт продукта MQA (производитель Agilent Technologies) – <http://www.agilent.com/find/eesof-mqa>
10. Официальный сайт продукта NI Multisim (производитель National Instruments) – <http://www.ni.com/multisim/>
11. Официальный сайт продукта Micro-Cap (производитель Spectrum Software) – <http://www.spectrum-soft.com/>
12. Официальный сайт продукта Cadence (OrCAD) PSpice A/D (производитель Cadence Design Systems) – http://www.cadence.com/products/orcad/pspice_simulation/Pages/default.aspx