



Кафедра вычислительной техники

В.А. Алехин

Наименование дисциплины
**Проектирование цифровых устройств
инфокоммуникационных систем**

Лекционные материалы

Для магистрантов
по направлению подготовки
«Информатика и вычислительная техника»
09.04.01.68

Москва, 2019

1

Оглавление

Оглавление	2
Глава 1. Инфокоммуникационные системы. Назначение. Архитектура. Структура. Реализация.....	11
Введение.....	11
1.1. Основные понятия и определения.....	11
Классификация информационных систем.....	12
1.2. Телекоммуникационные сети и системы.....	17
1.3. Структурные схемы телекоммуникационных систем.....	20
1.4. Виды модуляции в телекоммуникационных системах	27
1.5. Аппаратные средства телекоммуникационных систем	28
1.5.1. Компьютеры.....	28
1.5.2. Микропроцессоры и микроконтроллеры.....	28
1.5.3. Цифровые логические элементы на интегральных схемах	33
1.5.4. Мультиплексоры и демультиплексоры.....	37
1.5.5. Шифраторы и дешифраторы.....	39
1.5.6. Триггеры.....	39
1.5.7. Счетчики импульсов и регистры	40
1.5.8. Регистры	42
1.6. Цифровые постоянные запоминающие устройства (ROM).....	43
1.6.1. Постоянное ЗУ (ПЗУ)	44
1.6.2. Программируемое ПЗУ (ППЗУ), PROM.....	47

1.6.3. Стираемое программируемое ПЗУ (EPROM)	48
1.6.4. EEPROM - электрически стираемое программируемое ПЗУ	49
1.6.5. Обзор энергонезависимых ЗУ	50
1.7. Оперативные запоминающие устройства ОЗУ (RAM)	51
1.7.1. Статическое ОЗУ RAM.....	51
1.7.2. Динамические ОЗУ (DRAM)	53
1.8. Программируемые логические схемы	55
1.8.1. Логические схемы, программируемые изготовителем	55
1.8.2. Логические схемы, программируемые потребителем.....	55
1.8.3. Разновидности программируемых логических устройств.....	58
1.9. Аналого - цифровые и цифроаналоговые	61
преобразователи	61
1.9.1. Принцип аналого-цифровое преобразования.....	62
1.9.2. Этапы аналого-цифрового преобразования	63
1.9.3. АЦП прямого параллельного преобразования.....	66
1.9.4. АЦП, работающий по весовому принципу	68
1.9.5. АЦП, использующие методы счета.....	70
1.9.6. Компенсационный числовой АЦП.....	70
1.9.7. Метод пилообразного напряжения.....	72
1.9.8. Метод двойного интегрирования	73
1.9.9. Цифроаналоговое преобразование	77
1.9.10. ЦАП с суммированием весовых токов	77

1.9.11. ЦАП с резистивной матрицей постоянного импеданса	78
1.10. Применение дельта – сигма модуляции в АЦП и ЦАП	80
1.10.1. Краткий исторический обзор.	81
1.10.2. Дельта-модуляция	81
1.10.3. Дельта-сигма модуляция	83
1.10.4. Шумы дельта-сигма модуляции	86
1.10.5. Информационные параметры	88
1.10.6. Применение.....	89
Глава 2. Современное проектирование информационно-коммуникационных систем	93
2.1. Современное проектирование ИКС	93
2.2. Высокоуровневое проектирование встраиваемых систем.....	94
2.3. Платформы проектирования в информационно-коммуникационной технике	99
2.4. Проектирование электронных устройств в САПР Cadence.....	101
Глава 3. Анализ и проектирование цифровых устройств в среде TINA .	104
Введение.....	104
3.1. Интерфейс программы TINA 12- Education	105
3.1.1. Главное окно схемного редактора.....	106
3.1.2. Редактирование схемы с помощью мыши.....	114
3.1.3. Размещение компонентов.....	116
3.1.4. Основные режимы работы TINA.....	119

3.2. Моделирование простых цифровых схем.....	121
3.3. Анализ цифровых цепей с использованием цифровых HDL имитационных моделей	126
3.3.1. Анализ цифровой схемы с использованием цифрового VHDL моделирования.....	127
3.3.2. Отладчик HDL: отладка VHDL кодов.....	134
3.4. Анализ цифровой схемы с использованием цифрового Verilog моделирования.....	141
3.5. Анализ цепей с использованием моделей Verilog-A.....	144
3.6. Анализ цепей с использованием моделей Verilog-AMS	147
3.7. Анализ цепей с использованием SystemC	150
3.7.1. Требования к компилятору	150
3.7.2. Компиляция модели.....	151
3.7.3. Требования к моделированию	153
3.7.4. Настройка анализа в TINA	157
3.8. Схемы с микроконтроллерами (MCU).....	158
3.8.1. Отладка кода ASM	162
3.8.2. Пример обработки прерываний PIC	162
3.8.3. Редактирование кода ASM в отладчике.....	167
3.8.4. Создание точки останова в ASM	167
3.8.5. Программирование микроконтроллеров с использованием C	168
3.8.6. Отладка кода C в MCU	173

3.8.7. Использование редактора блок-схем и отладчика в ТИНА.....	177
3.8.8. Редактор блок-схем.....	178
3.9. Моделирование в смешанном режиме (Spice - VHDL – MCU совместное моделирование).....	181
3.9.1. Генерация сигналов с использованием VHDL и Spice подсхемы..	182
3.9.2. Генерация сигналов с использованием SystemC и Spice подсхемы	189
3.9.3. Импульсный источник питания на микроконтроллере.....	191
3.9.4. Схемы для формирования сигналов в инфокоммуникационных системах.....	193
3.9.5. Модели устройств компании Infineon.....	194
3.10. Проектирование печатных плат в среде TINA.....	195
Глава 4. Проектирование цифровых устройств в среде OrCAD 17.2	203
4.1. Цифровое моделирование в среде OrCAD 17.2	203
4.1. Модели цифровых устройств.....	204
4.1.1. Функциональное поведение	204
4.2. Цифровые цепи.....	210
4.2.1. Моделирование цифрового счетчика.....	211
4.2.2. Профиль цифрового моделирования.....	214
4.2.3. Отображение цифровых сигналов.....	214
4.3. Контрольные вопросы	217
Глава 5. Смешанное моделирование.....	218
5.1. Исследование аналогового компаратора с цифровым выходом	218

5.2. Исследование цифро-аналогового преобразователя	221
5.3. Контрольные вопросы	225
Глава 6. Создание иерархических проектов	227
6.1. Создание иерархического проекта	227
6.1.1. Создание плоского проекта полусумматора HalfAdd	229
6.1.2. Иерархические порты и off-page разъемы	230
6.1.3. Создание иерархического проекта Full Adder	234
6.1.4. Восходящий метод	234
6.1.5. Создание схемы полного сумматора	235
6.1.6. Добавление в проект аналоговых компонентов	241
6.1.7. Создание и сохранение компонентов для новых проектов	242
6.1.8. Нисходящий метод	245
6.2. Перемещение по иерархической конструкции	252
6.3. Моделирование полного сумматора	254
6.4. Контрольные вопросы	256
Глава 7. Испытательные стенды	258
7.1. Использование частичного моделирования проекта	259
7.2. Работа с тестовым стендом	261
7.2.1. Создание тестового стенда	262
7.2.2. Активация компонентов	264
7.3. Сравнение и обновление основного проекта	267
7.4. Контрольные вопросы	269

Глава 8. Обработка схемы	271
8.1. Добавление ссылок для компонентов	271
8.2. Создание отчёта перекрёстных ссылок.....	275
8.3. Создание списка материалов	277
8.4. Добавление специфических свойств редактора PCB	280
8.5. Проверка правил проектирования	282
8.6. Контрольные вопросы	286
Глава 9. Проектирование печатных плат с использованием	287
OrCAD PCB Editor.....	287
9.1. Обзор	287
9.2. Подготовка в Capture	289
9.2.1. Создание списка соединений для редактора печатных плат	289
9.3. Начало работы в PCB Editor.....	295
9.4. Создание печатной платы.....	296
9.4.1. Создание контура платы.....	297
9.4.2. Добавление монтажных отверстий	302
9.5. Размещение компонентов.....	307
9.5.1. Выбор компонентов с помощью RefDes.....	308
9.5.2. Поиск компонентов на плате	313
9.5.3. Проверка правил разработки.....	316
9.6. Использование категории DataTip	317
9.7. Выделение области	319

9.8. Выбор элементов дизайна с помощью Superfilter.....	320
9.9. Общие параметры на всплывающих меню.....	322
9.10. Описание режима привязки	323
9.11. Использование панели окна WorldView	325
9.12. Маршрутизация	326
9.12.1. Руководство по маршрутизации	327
9.12.2. Ручная маршрутизация цепей VCC и GND.....	328
9.12.3. Маршрутизация остальных сетей вручную.....	332
9.13. Автоматическая маршрутизация с помощью PCB Editor	336
9.14. Автоматическая маршрутизация с помощью OrCAD PCB Router .	336
9.15. Пост-обработка.....	341
9.15.1. Переименование компонентов вручную.....	342
9.15.2. Автоматическое переименование компонентов	342
9.15.3. Обратное аннотирование.....	344
9.16. Кросс-зондирование и кросс-выделение между редактором	348
печатных плат и Capture	349
9.17. Генерация вывода.....	351
9.18. Выходные файлы.....	351
9.19. Отчеты	355
9.20. Резюме	355
9.21. Контрольные вопросы	356
Основная литература	359

Дополнительная литература.....	361
--------------------------------	-----

Глава 1. Инфокоммуникационные системы. Назначение. Архитектура.

Структура. Реализация

Введение

В настоящее время тенденции развития информационных систем и средств обработки и распределения информации характеризуются тем, что, с одной стороны, развитие телекоммуникационных сетей требует применения цифровых каналов и систем передачи данных, средств вычислительной техники для обработки информации в процессе её передачи, а с другой - развитие средств обработки информации и вычислительной техники требует все большего применения средств связи для организации обмена информацией в интересах решения прикладных задач. И как результат происходили процессы интеграции и конвергенции телекоммуникационных сетей и средств информатизации, которые способствовали превращению телекоммуникационных сетей в инфокоммуникационные сети (ранее применялись также термины «информационная сеть», «телекоммуникационная вычислительная сеть» и др.).

1.1. Основные понятия и определения

Для того чтобы задачи проектирования цифровых устройств в составе инфокоммуникационных систем стали для нас конкретными, сначала необходимо изучить основные понятия этой области науки и техники, а также современные способы и средства технической реализации инфокоммуникационных систем.

Согласно существующим представлениям, инфокоммуникационная система (ИКС) – это совокупность, включающая сущности информационной и телекоммуникационной систем.

Информационная система (ИС) включает в себя информацию и пользователя.

Классификация информационных систем

- По уровню автоматизации процессов управления: системы поддержки принятия решений, интеллектуальные , информационно -управленческие , информационно - справочные , информационно – поисковые;
- По уровню и сфере деятельности: государственные, территориальные, по отраслям, объединений, технологических процессов, предприятий;
- По степени централизации обработки информации: централизованные, децентрализованные, коллективного использования;
- По степени интеграции функций: многоуровневые с интеграцией по уровню управления, многоуровневые с интеграцией по уровню планирования.

Информационно-коммуникационная система (ИКС)– это сложная распределённая в пространстве система, состоящая из множества локальных подсистем, располагающих программно-аппаратными средствами реализации информационных технологий, и множества средств, обеспечивающих соединение и взаимодействие этих подсистем с целью предоставления территориально удалённым пользователям широкого набора услуг из сферы информационного обслуживания.

Классификация информационно-коммуникационных систем.

Информационно-коммуникационные системы разделяют на 2 группы:

- а) системы информационного обеспечения;

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

б) системы, имеющие самостоятельное целевое назначение и область применения.

Системы самостоятельного или целевого назначения разделяются на:

- информационно-поисковые системы (ИПС)
- информационно-справочные системы (ИСС)
- информационно-управляющие системы (ИУС)

Телекоммуникационная система (ТКС) обеспечивает перенос информации от источника к потребителю.

Таким образом, инфокоммуникационную систему образует совокупность сети телекоммуникаций (телекоммуникационной подсистемы), прикладной подсистемы (средств хранения и обработки информации, прикладных процессов), а также подсистемы источников и потребителей информации (пользовательские подсистемы).

Наряду с терминами «инфокоммуникационная система» и «инфокоммуникационная сеть» используются термины:

Информационно-телекоммуникационные системы - класс систем, реализующий множество технологических процессов по сбору, обработке, хранению, поиску информации и доступа к ней, переносу (транспортировке) всех видов сообщений путём их объединения в единые транспортные потоки.

Информационно-телекоммуникационная сеть - технологическая система, содержащая линии связи, узлы и пользовательское оборудование, обеспечивающая возможность предоставления услуг по доставке информации пользователям и, частично, по её хранению и обработке в процессе передачи и доставки.

В целях обеспечения корректности использования таких терминов как «инфокоммуникации», «инфокоммуникационная сеть», «инфокоммуникационные технологии» и др. рассмотрим их определения и взаимосвязь.

Инфокоммуникации (Infocommunications) - это совокупность средств обработки, накопления, хранения информации и переноса её в пространстве, имплементированных (исполненных) в единую сетевую структуру, посредством которой обеспечивается доступность информационных ресурсов и информационный обмен.

Инфокоммуникационная сеть (Infocommunication Network) - это совокупность территориально рассредоточенных информационных, вычислительных ресурсов, программных комплексов управления, размещаемых в оконечных системах сети и терминальных системах пользователей, взаимодействие между которыми обеспечивается посредством телекоммуникаций, и которые совместно образуют единую мультисервисную платформу.

Инфокоммуникационные технологии (Infocommunication Technologies) - это совокупность методов и способов обработки, накопления, хранения, отображения и обеспечения целостности информации, а также способов реализации режимов её переноса в пространстве, обеспечивающих некоторый гарантированный уровень качества обслуживания.

Информационные процессы - сбор, обработка, накопление, хранение, поиск и распространение информации.

Телекоммуникационные процессы - передача и коммуникация информации.

Под **информационной технологией** (*Information Technology*) понимаются процессы, методы поиска, сбора, хранения, обработки,

предоставления, распространения информации и способы осуществления этих процессов и методов.

Более широко в настоящее время используется термин **«информационно-коммуникационные технологии»** (Information and Communication Technology), отражающий совокупность информационных процессов и методов работы с информацией, осуществляемых с применением средств вычислительной техники и средств телекоммуникации.

Информационно-коммуникационные технологии (ИКТ) - технологии, предназначенные для совместной реализации информационных и коммуникационных процессов.

Таким образом, инфокоммуникационные сети предназначены для предоставления пользователям услуг, связанных с обменом информацией, её потреблением, а также обработкой, хранением и накоплением.

В качестве *пользователей* могут выступать как физические лица, так и юридические (фирмы, организации, предприятия).

Пользователь, организуя запрос на предоставление той или иной услуги, активизирует в своей оконечной системе некоторый прикладной процесс, выполняющий обработку информации для конкретной услуги связи или приложения.

Оконечными системами инфокоммуникационной сети могут быть:

- терминальные системы, обеспечивающие доступ к сети и её ресурсам;
- рабочие системы, предоставляющие сетевой сервис (управление доступом к файлам, программам, сетевым устройствам, обслуживание вызовов и т.д.);
- административные системы, реализующие управление сетью и отдельными её частями.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Базовым компонентом, ядром инфокоммуникационной сети, является **телекоммуникационная сеть**, состоящая из взаимодействующих телекоммуникационных систем.

Телекоммуникационная система – упорядоченная совокупность методов, правил, протоколов, технических и программных средств в их взаимосвязи и взаимодействии, обеспечивающих передачу электронного сообщения от источника к получателю по сетям электросвязи.

Информационно-коммуникационная система (ИКС)– это сложная распределённая в пространстве система, состоящая из множества локальных подсистем, располагающих программно-аппаратными средствами реализации информационных технологий, и множества средств, обеспечивающих соединение и взаимодействие этих подсистем с целью предоставления территориально удалённым пользователям широкого набора услуг из сферы информационного обслуживания.

Сеть электросвязи – технологическая система, включающая в себя средства и линии связи и предназначенная для электросвязи.

Коммуникационный узел (коммутационный узел) – совокупность технических средств, предназначенных для обработки вызовов или заявок на обслуживание, поступающих по абонентским и соединительным линиям сети, для предоставления инициаторам этих вызовов/заявок основных и дополнительных услуг связи, а также для учёта информации о предоставлении услуги.

Терминальное оборудование пользователя, терминал (абонентское устройство, оконечное оборудование пользователя) – оборудование, подключённое к сети электросвязи, для обеспечения доступа к одной или нескольким определённым службам.

Глобальная информационная инфраструктура – это совокупность сетей связи, оконечного оборудования пользователей, информации, которая может быть использована для коммуникации пользователей и передаётся по доступным ценам с заданным качеством.

Платформы поддержки коммуникаций – это оконечное оборудование данных, модемы, устройства доступа различного назначения, которые непосредственно преобразуют данные для передачи по сетям связи.

Протоколы обмена – это набор правил и форматов представления информации, которая определяет взаимосвязанное поведение взаимодействующих объектов

Сеть доступа – совокупность абонентских линий и станций местной сети, которые обеспечивают доступ оконечного оборудования пользователей (ООП) транспортной сети, а так же местная связь без выхода в транспортную сеть.

1.2. Телекоммуникационные сети и системы

Телекоммуникационные сети и системы являются технологической системой в составе инфокоммуникационных систем и содержат линии связи, узлы и пользовательское оборудование, обеспечивающие возможность предоставления услуг по доставке информации пользователям и, частично, по её хранению и обработке в процессе передачи и доставки.

Телекоммуникационные сети и системы представляют собой комплекс технических средств, обеспечивающих передачу разнообразных сообщений между любыми пользователями, на произвольные расстояния с заданными параметрами качества.

Телекоммуникационные системы (далее ТС) используют многоканальные системы передачи, работающие в различных физических средах (электрический

кабель, волоконно-оптический кабель, радиоспектр) с целью формирования типовых каналов и трактов.

На базе телекоммуникационных систем строятся телекоммуникационные сети, представляющие собой совокупность узлов передачи/обработки информации и линий их соединяющих. В совокупности телекоммуникационные системы и сети образуют систему электросвязи – комплекс технических средств, обеспечивающих электросвязь определённого вида.

На рис. 1.1. показана обобщённая схема инфокоммуникационной системы. В ней два Информационных устройства, имеющие Информацию, Платформы поддержки приложений и Платформы поддержки коммуникаций, осуществляют обмен информацией через Сеть связи с телекоммуникационной инфраструктурой.

Широкое внедрение компьютеров в производственную деятельность и повседневную жизнь стимулирует развитие телекоммуникаций, требует существенной скорости передачи информации, повышения качества обслуживания. Цифровые методы обработки телекоммуникационных сигналов, цифровых технологий распределения информации и управления сетями приводят к слиянию информационных и телекоммуникационных технологий.

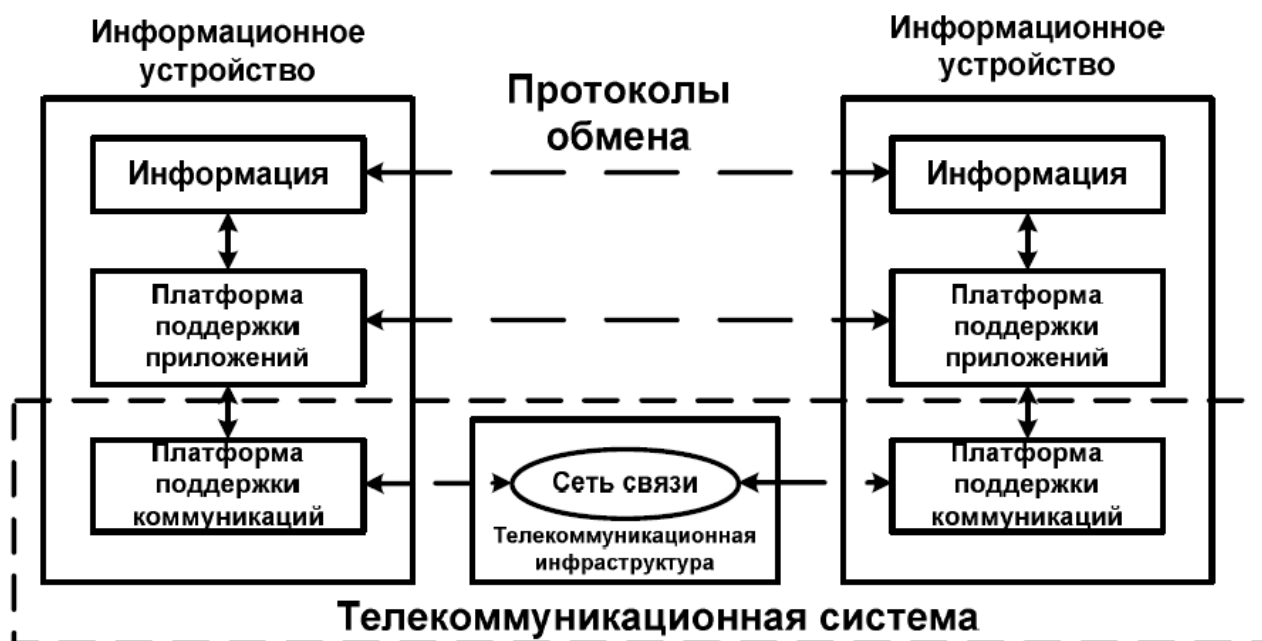


Рис. 1.1. Структурная схема инфокоммуникационной системы

Основные услуги телекоммуникаций можно разделить на следующие виды:

- телеметрия,
- телефония,
- радиовещание,
- передача данных ПД (доступ в Интернет, электронная почта и др.),
- телевидение,
- цифровое телевидение.

Эти услуги различаются как по необходимой скорости передачи информации, так и по времени сеанса. Самые низкие скорости передачи и время сеанса требуются для телеметрии, когда сигналы с нескольких датчиков (давления, температуры, влажности и т. п.) передаются в центр управления. Самые большие объемы скорости и длительности сеанса необходимы для телевидения (TV), телевидения высокого качества (ТВВК) и скоростного

доступа в Интернет. Телефония и радиовещание занимают промежуточное положение.

ТС предназначены для передачи информации. Сформулируем некоторые дополнительные определения:

Информация — совокупность сведений, данных, знаний о каких-либо процессах, явлениях, объектах и т. п., способных храниться, передаваться и преобразовываться для деятельности человека.

1.3. Структурные схемы телекоммуникационных систем

Носителем информации является сообщение, которое, в свою очередь, преобразуется в первичный электрический сигнал $U(t)$. Передача сообщения в телекоммуникационной системе схематично представлена на рис. 1.2, где $S(t)$ — электрический сигнал, удобный для передачи по линии связи, а $n(t)$ — помехи и шумы.

Введём дополнительные определения.

Линия передачи — физическая среда распространения электромагнитных волн, несущих сообщение. После прохождения линии передачи сигнал испытывает обратное преобразование. В качестве линии передачи используются разные виды кабелей. На сегодняшний день одними из самых популярных являются витая пара и оптоволокно. Также довольно широко используются радиорелейные и спутниковые линии связи.

Канал передачи начинается и оканчивается у абонентов, но не включает в себя оконечные терминальные устройства.

Система передачи — канал передачи вместе с оконечными устройствами.

Рассмотренный канал передачи является двухточечным и односторонним, т. е. передача сообщений осуществляется в одну сторону.

Если источник и приемник поочередно меняются местами, то для обмена сигналами необходимо использовать двухсторонний канал связи, допускающий передачу как в одну, так и в другую сторону. Наиболее эффективный способ передачи - это организация двух встречных каналов, работающих одновременно. Такой спаренный канал называется дуплексным.

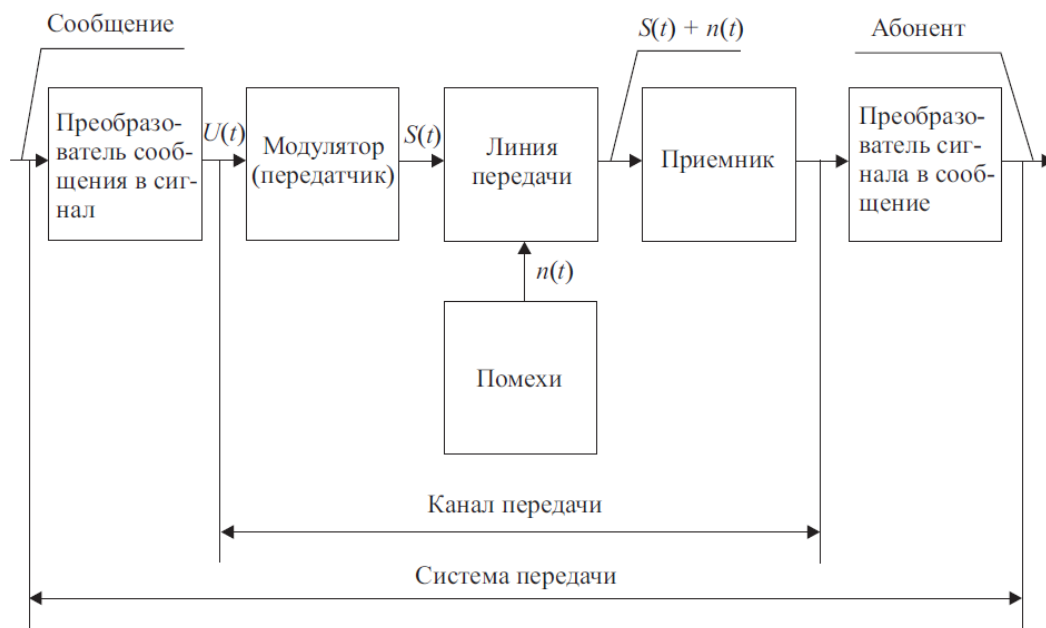


Рис. 1.2. Упрощённая схема системы передачи и приёма сообщений

Отличительной особенностью современных телекоммуникационных систем является то, что это, как правило, цифровые системы связи (ЦСС, digital communication system — DCS). Упрощённая схема ЦСС показана на рис. 1.3.

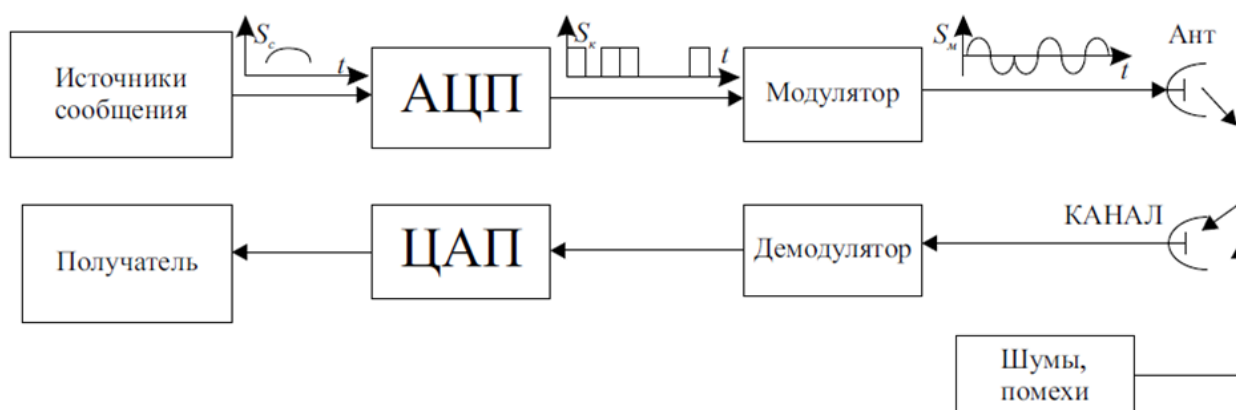


Рис. 1.3. Упрощённая схема цифровой системы связи

На передающей стороне аналоговый сигнал источника сообщений преобразуется аналого-цифровым преобразователем (АЦП) в форму цифровых кодов дискретных выборок сигнала. Цифровые коды поступают на модулятор, в котором осуществляется, например, кодово- импульсная модуляция несущего высокочастотного или оптического сигнала, который излучается антенной или оптическим излучателем.

На приёмной стороне сигнал с помехами и шумами принимается антенной или оптическим приёмником, демодулируется в форму цифровых кодов, преобразуется в цифроаналоговом преобразователе (ЦАП) в аналоговую форму и поступает к получателю информации.

Структурная схема (рис. 1.3) содержит следующие цифровые и аналого-цифровые устройства: *АЦП, модулятор, ЦАП, демодулятор*.

Обычно на практике используются многоканальные системы передачи (МСП). С их помощью мы можем передавать несколько сообщений одновременно по одному каналу. На рис. 1.4 представлена обобщенная структурная схема МСП.

В МСП первичные сигналы, поступающие от абонента, преобразуются модуляторами (M_i). Полученные сигналы называются канальными сигналами.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Устройство объединения каналов (УОК) объединяет каналные сигналы в так называемый групповой сигнал.

В настоящее время в телекоммуникационных системах используют частотное разделение каналов (ЧРК), временное (ВРК) и кодовое (КРК).

При ЧРК каждому каналу соответствует своя полоса частот Δf_i (рис. 1.5). Обычно все Δf_i равны, хотя может быть и не так. Все каналные сигналы в группе генерируются одновременно. Для того чтобы подчеркнуть, что форма спектра каналных сигналов несимметрична и при объединении и разделении каналов спектр может быть инвертирован (зеркально отражен относительно f_i), на рис. 1.5 они изображены в виде «пилообразных треугольников». Роль преобразующих устройств в системе с ЧРК играют преобразователи частоты (смесители), а в качестве УОК применяются сумматор и передатчик (усилитель мощности). В качестве устройства разделения каналов (УРК) используется система полосовых фильтров, настроенных на f_i . Выделенные каналные сигналы преобразуются в первичные с помощью детекторов.

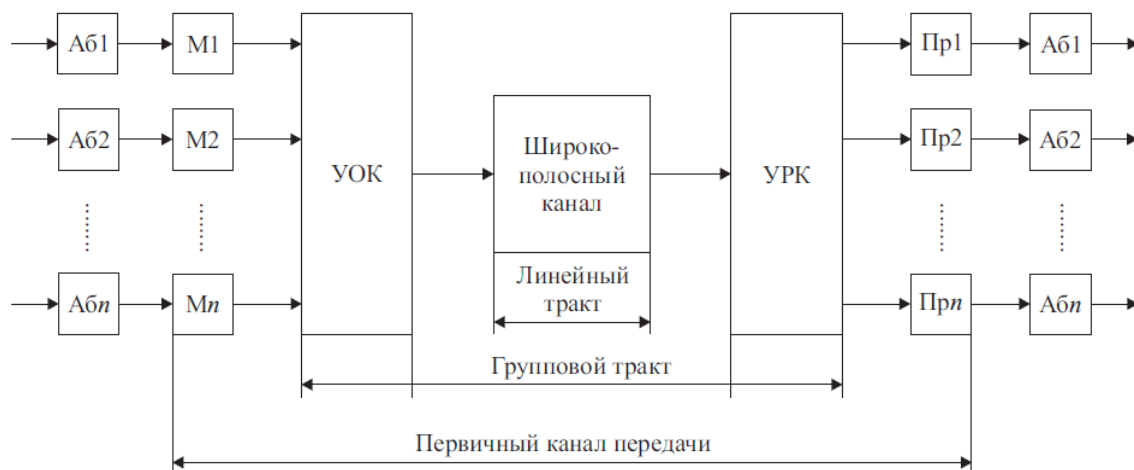


Рис. 1.4. Многоканальная система передачи

При временном разделении каналов (ВРК) из первичных сигналов делаются дискретные выборки с шагом дискретизации τ_d . Моменты

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

дискретизации для соседних первичных сигналов сдвинуты на время τ_k . При этом за время τ_d можно передать $N = \tau_d/\tau_k$ дискретных сигналов (рис. 1.6).

Здесь все сигналы группы передаются в одной полосе частот, но в разные моменты времени. Устройствами объединения и разделения каналов в этом случае будут соответственно *мультиплексор* и *демультиплексор*.

Эти устройства являются цифровыми и мы будем изучать их в курсе.

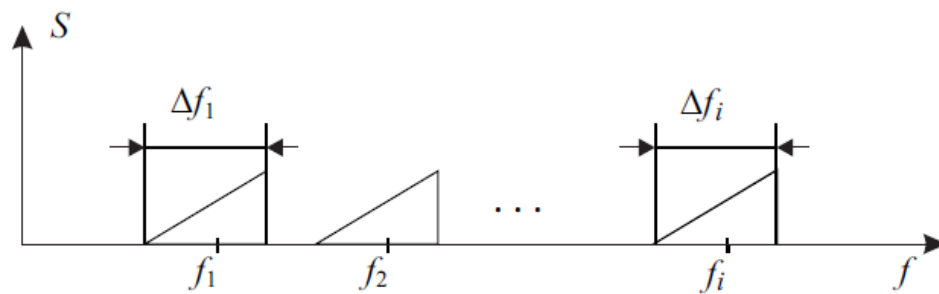


Рис. 1.5. Частотное разделение каналов

При КРК все первичные сигналы преобразуются в цифровой код и символы этого кода передаются одновременно и в одной и той же полосе частот. Различение первичных сигналов осуществляется за счёт внутреннего кодирования каждого символа (рис. 1.7) с помощью специального кодера. На приёмном конце каждому каналу соответствует свой *коррелятор* или *согласованный фильтр*.

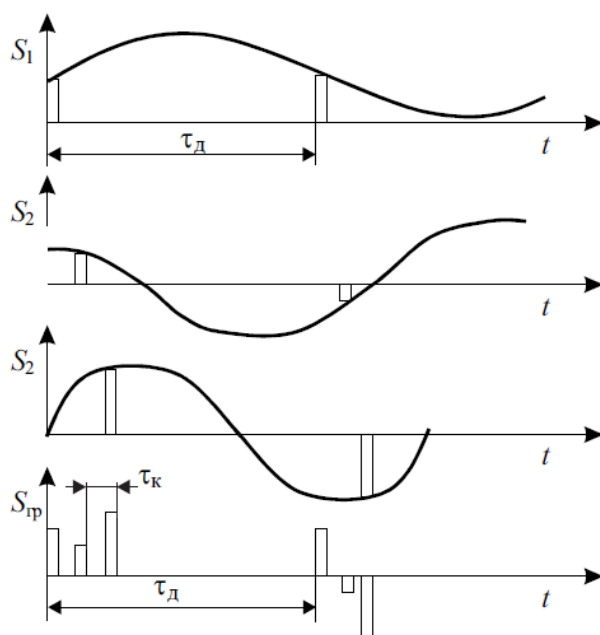


Рис. 1.6. Временное разделение каналов

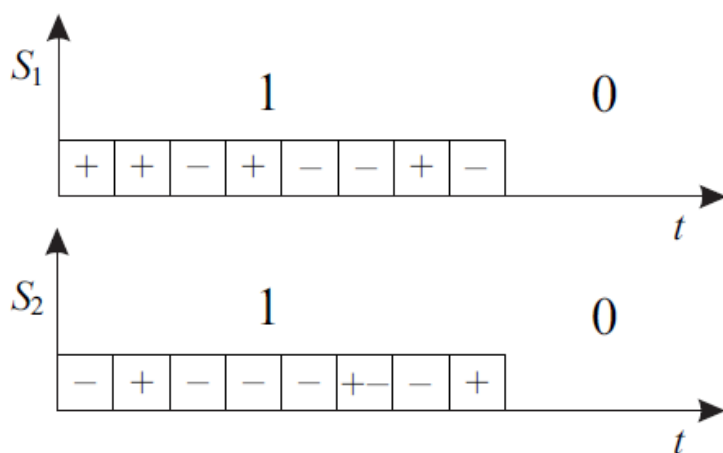


Рис. 1.7. Кодовое разделение каналов

Поскольку база такого сигнала B (число импульсов внутри символа) должна быть большой, порядка 100, то спектр сигнала при сохранении скорости передачи расширяется в B раз. Поэтому сигналы при КРК называют также широкополосными или шумоподобными.

Ещё более полная функциональная схема типичной цифровой системы связи показана на рис. 1.8.

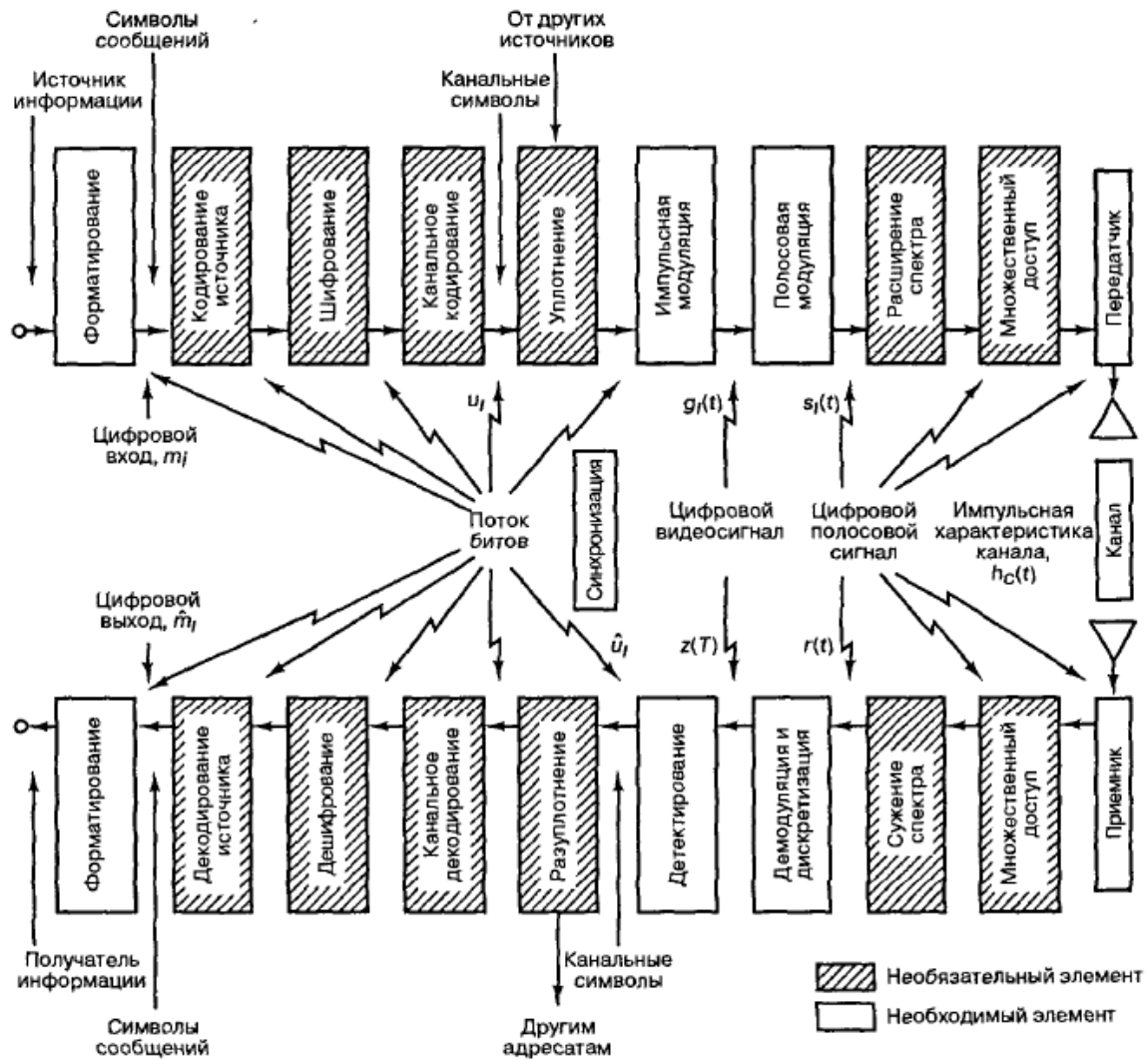


Рис. 1.8. Функциональная схема типичной цифровой системы связи

Верхние блоки — форматирование, кодирование источника, шифрование, канальное кодирование, уплотнение, импульсная модуляция, полосовая модуляция, расширение спектра и множественный доступ — отражают преобразования сигнала на пути от источника к передатчику.

Нижние блоки диаграммы — преобразования сигнала на пути от приемника к получателю информации, и, по сути, они противоположны

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

верхним блокам. Блоки модуляции и демодуляции/детектирования вместе называются модемом. Для беспроводных приложений передатчик состоит из схемы повышения частоты в область радиочастот (RF), усилителя мощности и антенны, а приемник — из антенны и малошумящего усилителя (LNA). Обратное понижение частоты производится на выходе приемника и/или демодулятора.

К цифровым устройствам в схеме (рис. 1.8) относятся, в частности, кодеры и декодеры, шифраторы и дешифраторы, каналные кодеры и декодеры и т.п.

1.4. Виды модуляции в телекоммуникационных системах

В цифровых системах связи для передачи на большие расстояния по радиолиниям или электрическим кабелям используют высокочастотные модулированные сигналы.

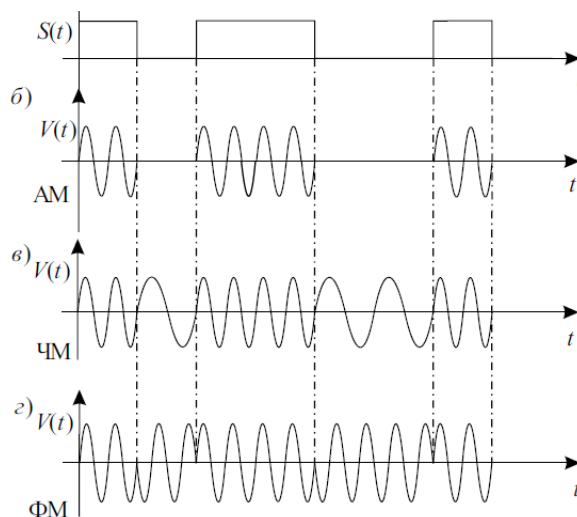


Рис. 1.9. Виды модуляции цифровых сигналов

а) цифровой сигнал; б) амплитудная модуляция;

в) частотная модуляция; г) фазовая модуляция (манипуляция)

Модуляция, которая в данном случае называется манипуляцией, может быть амплитудной (АМ), фазовой (ФМ), частотной (ЧМ) или их комбинацией.

На рис. 1.9 упрощённо показаны виды модуляции цифровых сигналов.

1.5. Аппаратные средства телекоммуникационных систем

Основными компонентами аппаратного обеспечения телекоммуникационных систем являются компьютеры различных типов, терминальное оборудование пользователей, коммуникационные узлы и иные средства связи.

Аппаратные средства используются в составе коммутационного узла, который представляет собой совокупность технических средств, предназначенных для обработки вызовов или заявок на обслуживание, поступающих по абонентским и соединительным линиям сети, для предоставления инициаторам этих вызовов/заявок основных и дополнительных услуг связи, а также для учета информации о предоставлении услуги.

Также аппаратные средства используются в составе оборудования пользователя (абонентское устройство, оконечное оборудование пользователя) – оборудование, подключенное к сети электросвязи, для обеспечения доступа к одной или нескольким определенным службам.

Рассмотрим кратко наиболее важные цифровые устройства и компоненты телекоммуникационных систем.

1.5.1. Компьютеры

Компьютеры телекоммуникационных систем могут отличаться производительностью, конфигурацией, выполняемыми функциями, что позволяет разделить их на рабочие станции и серверы сети.

1.5.2. Микропроцессоры и микроконтроллеры

Микропроцессор, МПр представляет собой синхронное цифровое вычислительное устройство обработки данных, функционирующее на основе загружаемой программы для электронно-вычислительных машин, ЭВМ. Конструктивно микропроцессор выполняется в виде одной или нескольких интегральных микросхем. МПр предназначен для интерпретации (выполнения) программ для ЭВМ.

Микропроцессор включает набор различных аппаратных компонентов, которые функционально объединены в виде центрального вычислительного (процессорного) устройства ЦПУ (central processing unit, CPU). ЦПУ, которое также называется центральным процессором, выполняет основные функции по обработке информации и управлению работой других аппаратных компонент микропроцессорной системы.

Микропроцессор (процессор) выполняет арифметические, логические операции и операции управления, записанных в машинном коде, реализованный в виде одной микросхемы или комплекта из нескольких специализированных микросхем.

ЦПУ выполняет обработку данных с помощью арифметико-логического устройства (АЛУ) и встроенных устройств кратковременного хранения данных – регистров. АЛУ позволяет выполнять вычислительные и логические операции на аппаратном уровне, для чего имеет в своем составе сумматор, схемы базовых логических операций.

Современные микропроцессоры представляют собой весьма сложные по устройству изделия микроэлектроники. Многочисленные типы микропроцессоров характеризуются различными архитектурными решениями и функциональными возможностями. Микропроцессорная техника стремительно и многонаправленно развивается и совершенствуется, интегрируя новейшие достижения микроэлектроники и схемотехники.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Первые семейства 8-разрядных микропроцессорных БИС: 8080, 8085 (Intel), Z80 (Zilog), MC6800, MC6809 (Motorola), MCS6500 (MOS Technology) были разработаны к середине 80-х годов XX века. Отечественными аналогами стали микропроцессорные комплекты серий К580 и К1821.

Микропроцессорный комплект кроме собственно центрального процессора содержит около 20 вспомогательных микросхем (шинные формирователи, приемо-передатчики, контроллеры ввода-вывода, дисплея, интерфейса системы, прямого доступа к памяти и т.п.).

Физическая память – аппаратная часть микропроцессорной системы, в которую могут записываться и храниться, а при необходимости – из которой считываются данные и команды. Физическая оперативная память – память, в которой размещаются данные, обрабатываемые командами, и сами команды в ходе непосредственного выполнения (интерпретации) программ.

Структура микропроцессорного устройства

В микропроцессорных устройствах применяется магистрально-модульный принцип построения микропроцессорных систем (МПС). Упрощенная структура такой МПС приведена на рис. 11.1. Центральным устройством в системе является микропроцессор (МП), выполняющий арифметические и логические операции над данными, осуществляющий управление выборкой команд и данных из памяти и организующий взаимодействие всех устройств, входящих в систему.

Работа МП происходит под воздействием тактовых сигналов, вырабатываемых схемой синхронизации, часто выполняемой в виде отдельной микросхемы (генератора тактовых импульсов).

Программы работы МПС размещаются в модулях постоянных запоминающих устройств (ПЗУ) и оперативных запоминающих устройств (ОЗУ).

Микропроцессорные устройства и МПС содержат различные средства ввода-вывода информации. Шины адреса, данных и управления объединяют все устройства в единую систему. Периферийные устройства подсоединяются к шинам через программируемые периферийные адаптеры, осуществляющие передачу информации в параллельном или последовательном кодах. Наличие программно-настраиваемых адаптеров делает весьма гибкой и функционально богатой систему ввода-вывода информации в МПС.

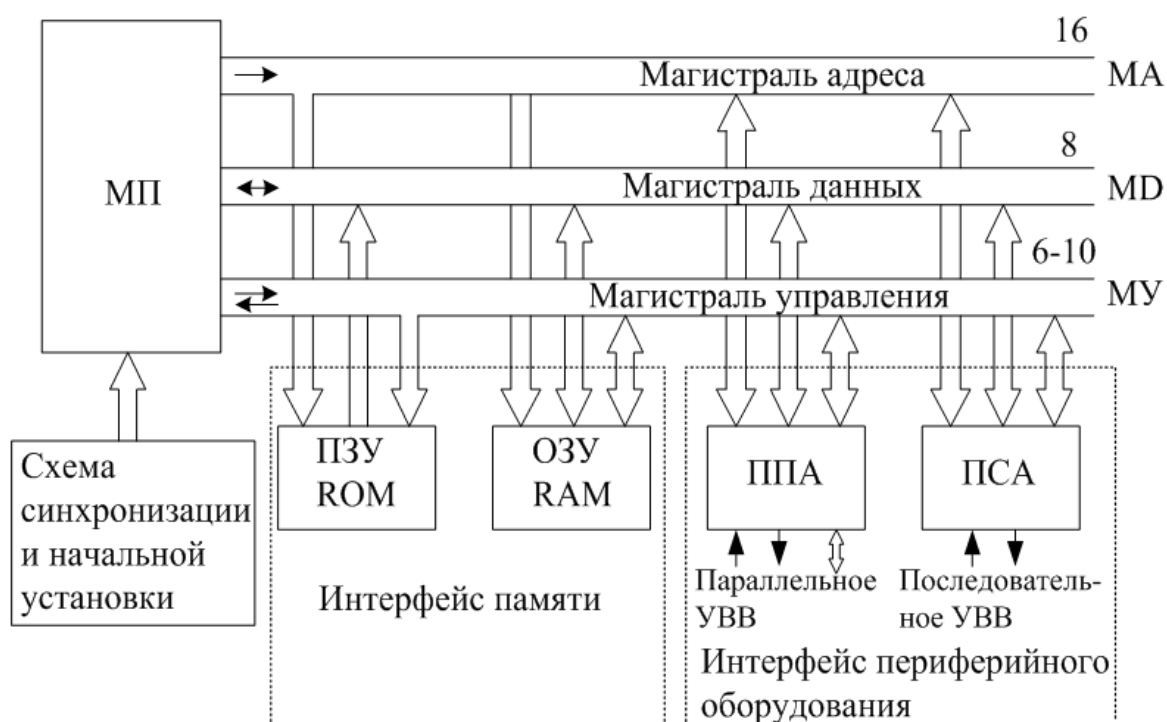


Рис. 1.10. Структура микропроцессорной системы

Подготовка списка команд называется программированием, а подготовленный список команд представляет собой программу, которая заносится в ПЗУ или ОЗУ и предписывает микропроцессору определенные

действия. МП считывает из памяти команду и выполняет ее. Далее МП формирует адрес следующей команды в программе, считывает, выполняет и формирует адрес новой команды. Так функционирует микро-ЭВМ.

Микроконтроллеры

Микроконтроллером называют микросхему, предназначенную для управления электронными устройствами. В отличие от микропроцессора типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств: содержит ОЗУ и (или) ПЗУ, интегрированные в микросхему устройства ввода-вывода, таймеры, АЦП и ЦАП, другие периферийные устройства.

По сути, это однокристальный компьютер, способный выполнять относительно простые задачи.

Микроконтроллеры выпускают многие фирмы (Microchip Technology, Atmel, Texas Instruments и др.).

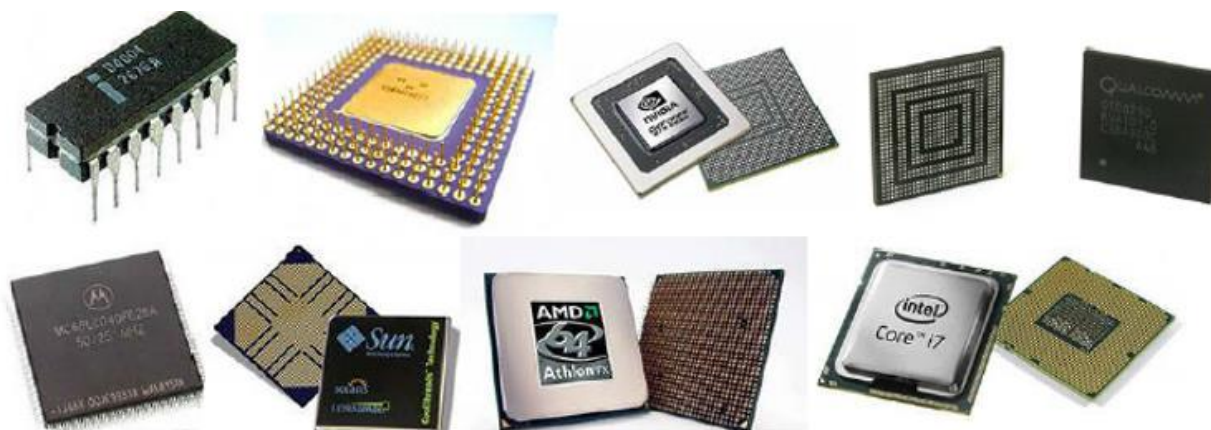


Рис. 1.10а. Общий вид современных микропроцессоров

Микроконтроллеры объединяют все передовые технологии, имеют широкую номенклатуру и используются в устройствах, предназначенных для

разнообразных сфер применения, в частности, в телекоммуникационных системах.

Высокая скорость выполнения команд в PIC контроллерах достигается за счет использования двухшинной гарвардской архитектуры, основанной на наборе регистров с разделенными шинами и адресными пространствами для команд и данных.

Микроконтроллеры PIC содержат RISC-процессор с симметричной системой команд, позволяющей выполнять операции с любым регистром, используя произвольный метод адресации. Пользователь может сохранять результат операции в самом регистре-аккумуляторе или во втором регистре, используемом для операции.

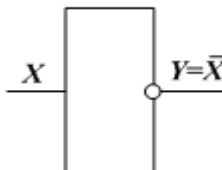
RISC (*restricted (reduced) instruction set computer*) - компьютер имеет сокращённый набор команд. Быстродействие в нем увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения - меньшим.

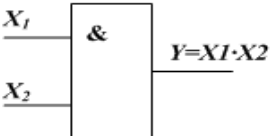
1.5.3. Цифровые логические элементы на интегральных схемах

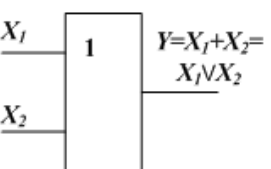
Цифровые логические элементы, выполненные на интегральных микросхемах (ИМС), предназначены для преобразования и обработки дискретных сигналов и выполняют основные логические функции, представленные в таблице 1.1.

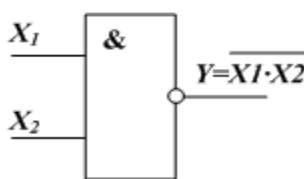
Таблица 1.1

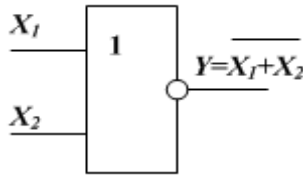
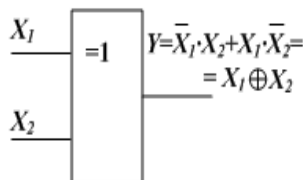
Элемент	Обозначение отечественных ИМС	Выполняемая функция и схема	Таблица истинности

НЕ (отрицание)	ЛН		x	$y = \bar{x}$	
			1	0	
			0	1	

И (логическое умножение)	ЛИ		x_1	x_2	y
			0	0	0
			1	0	0
			0	1	0
			1	1	1

ИЛИ (логическое сложение)	ЛЛ		x_1	x_2	y
			0	0	0
			1	0	1
			0	1	1
			1	1	1

И-НЕ (логическое умножение с отрицанием) (Штрих	ЛА		x_1	x_2	y
			0	0	1
			1	0	1
			0	1	1

Шефера)			1	1	0
ИЛИ-НЕ (логическое сложение с отрицанием) (Стрелка Пирса)	ЛЕ		x_1	x_2	y
			0	0	1
			1	0	0
			0	1	0
			1	1	0
Исключающее ИЛИ	ЛП		x_1	x_2	y
			0	0	0
			1	0	1
			0	1	1
			1	1	0

В таблице 1.1 использованы обозначения: \bar{x} - отрицание значения x ; $x_1 \cdot x_2$ - логическое умножение (конъюнкция); $X_1 + X_2 = X_1 \vee X_2$ - логическое сложение (дизъюнкция).

Цифровое устройство преобразует входную информацию, представленную в виде двоичного кода (нулей и единиц), в значение выходной функции $y(x_1, x_2, \dots, x_n)$ (рис. 1.11). В цифровых комбинационных устройствах выходные сигналы однозначно определяются только действующей в настоящий момент комбинацией входных переменных и не зависят от значений переменных, действовавших на входе ранее.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

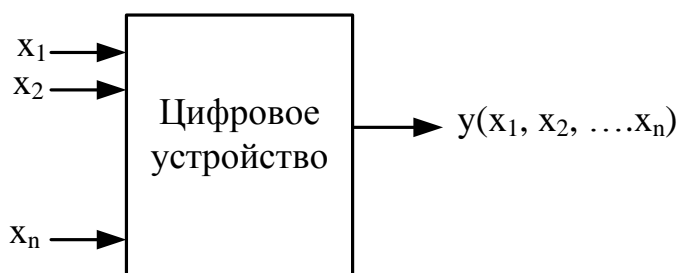


Рис. 1.11. Цифровое устройство комбинационного типа

Этапу построения комбинационной схемы должно предшествовать упрощение логических формул или минимизация. Целью минимизации является получение минимального необходимого количества элементов в схеме. Причем устройство, выполняющее любую логическую функцию, можно построить, имея элементы только одного вида («ИЛИ-НЕ» либо «И-НЕ»). Эти элементы называют универсальными базисами.

Цифровые устройства телекоммуникационных систем, как правило, создают с использованием разнообразных интегральных микросхем.

Современная интегральная (микро)схема – микроэлектронное изделие окончательной или промежуточной формы, предназначенное для выполнения функций электронной схемы, элементы и связи которой нераздельно сформированы в объеме и (или) на поверхности материала, на основе которого изготовлено изделие. Таким материалом в настоящее время является кремний. Интегральная микросхема имеет высокую плотность упаковки электрически соединенных элементов, компонентов или кристаллов, которые с точки зрения требований к испытаниям, поставке и эксплуатации рассматриваются как единое целое. Под плотностью упаковки понимается отношение суммы элементов интегральной микросхемы к объему интегральной микросхемы. Можно выделить сверхбольшую интегральную микросхему, СБИС где плотность упаковки составляет свыше 100000 элементов на объём микросхемы.

1.5.4. Мультиплексоры и демультиплексоры

Мультиплексором называют функциональный узел, который обеспечивает передачу цифровой информации, поступающей по нескольким входным линиям связи, на одну выходную линию. Мультиплексор можно представить в виде коммутатора, управляемого входной логической схемой. Входные логические сигналы X_i поступают на входы коммутатора и через коммутатор передаются на выход Y . На вход логической схемы поступают адресные сигналы A_k , определяющие, какой входной сигнал в данный момент передается на выход. Некоторые мультиплексоры могут иметь выход с тремя состояниями: два состояния 0 и 1 и третье состояние – отключённый выход (выходное сопротивление равно бесконечности).

На рис. 1.12 показана схема мультиплексора ($4 \rightarrow 1$), выполненная на элементах И и ИЛИ. Работа мультиплексора отображается таблицей истинности 1.2.

Графическое изображение мультиплексора ($8 \rightarrow 1$) показано на рис. 1.13. Входами являются выходы микросхемы $D_0 \dots D_7$, выходами Q и \bar{Q} , адресные входы A_0, A_1, A_2 , вход разрешения работы E .

Демультиплексор (DMX) выполняет преобразование, обратное мультиплексированию: входной сигнал X поступает на вход демультиплексора и передается на выходы $Y_1, Y_2 \dots Y_n$. Адресные сигналы $A_0 \dots A_k$ имеют такое же значение, как и у мультиплексора. Графическое изображение демультиплексора ($1 \rightarrow 4$) показано на рис. 1.14.

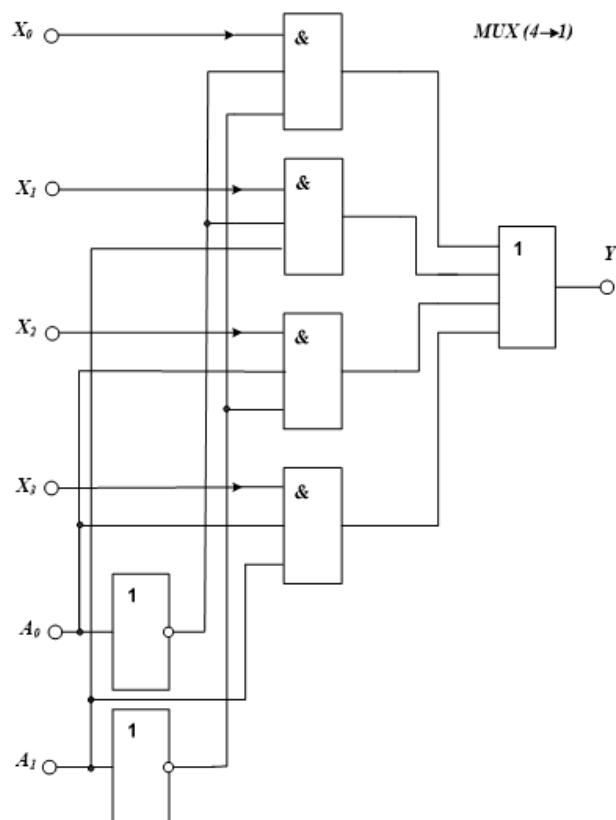


Рис. 1.12. Схема мультиплексора

Таблица 1.2.¶

A_0	A_1	Y
0	0	X_0
0	1	X_1
1	0	X_2
1	1	X_3

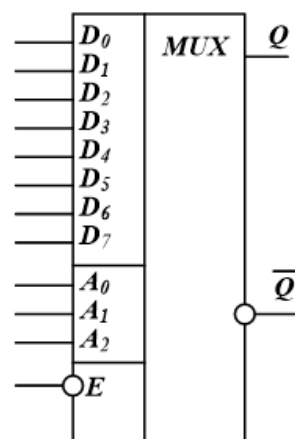
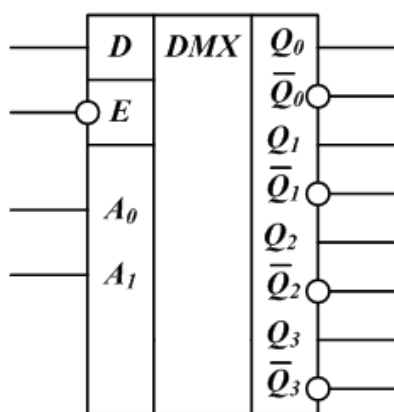
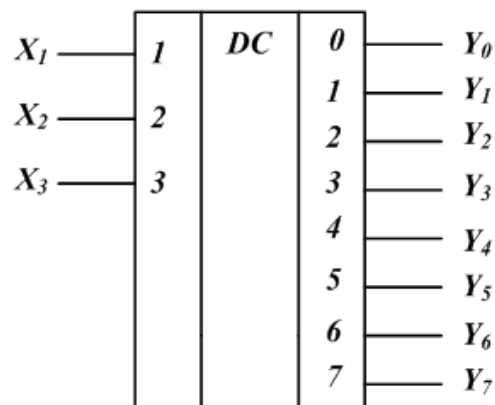
Рис. 1.13. Условное изображение
мультиплексора (8→1)Рис. 1.14. Условное изображение де-
мультиплексора (1→4)

Рис. 1.15. Условное графическое

изображение дешифратора 3х8

1.5.5. Шифраторы и дешифраторы

Дешифратором называют преобразователь двоичного n - разрядного кода в унитарный 2^n - разрядный код, все разряды которого за исключением одного равны нулю. Дешифраторы бывают полные и неполные. Для полного дешифратора выполняется условие $N=2^n$, где n – число входов, N – число выходов. В неполных дешифраторах имеется n – входов, но выходов $N < 2^n$.

На рис. 1.15 показана схема дешифратора 3 х 8, преобразующего двоичный трехразрядный код в унитарный восьмиразрядный, в котором, например, входное двоичное число 100 соответствует выходному коду 00010000.

В цифровой технике связи применяют также преобразователи кодов импульсных сигналов, шифраторы, преобразующие, например, 8-разрядный единичный код в двоичный код, и другие комбинационные устройства, построенные на цифровых логических элементах. Особенностью комбинационных устройств является то, что их выходные сигналы однозначно определяются только действующей в настоящий момент на входе комбинацией переменных и не зависят от значений переменных, действовавших на входе ранее.

1.5.6. Триггеры

Триггером называется устройство, способное формировать два устойчивых значения выходного сигнала и скачкообразно изменять эти значения под действием внешнего управляющего сигнала.

Триггеры относятся к цифровым схемам последовательного типа. Состояние выхода последовательной схемы (цифрового автомата) зависит от текущего состояния входа X и от внутреннего состояния схемы Q :

$$Y=F(X,Q).$$

Таким образом, триггер является хранителем предшествующей и источником текущей информации. Триггер называют бистабильной схемой, он является элементарной ячейкой памяти. Два устойчивых состояния триггера обозначаются: $Q=1$ и $Q=0$.

Основные типы триггеров в интегральном исполнении получили следующие названия: RS -триггер, D - триггер, T - триггер, JK - триггер.

1.5.7. Счетчики импульсов и регистры

Счетчиком называют цифровое устройство, предназначенное для подсчета числа импульсов. С поступлением каждого импульса на вход C счетчик меняет свое состояние на единицу. Счетчик можно реализовать на нескольких триггерах, при этом состояние счетчика будет определяться состоянием его триггеров. В суммирующих счетчиках каждый входной импульс увеличивает его состояние на единицу. В вычитающих счетчиках состояние уменьшается на единицу каждым входным импульсом.

Наиболее простыми являются двоичные счетчики, в которых состояние счетчика определяется двоичным кодом на его выходах. Схема асинхронного четырехразрядного суммирующего счетчика на D – триггерах показана на рис. 1.16.

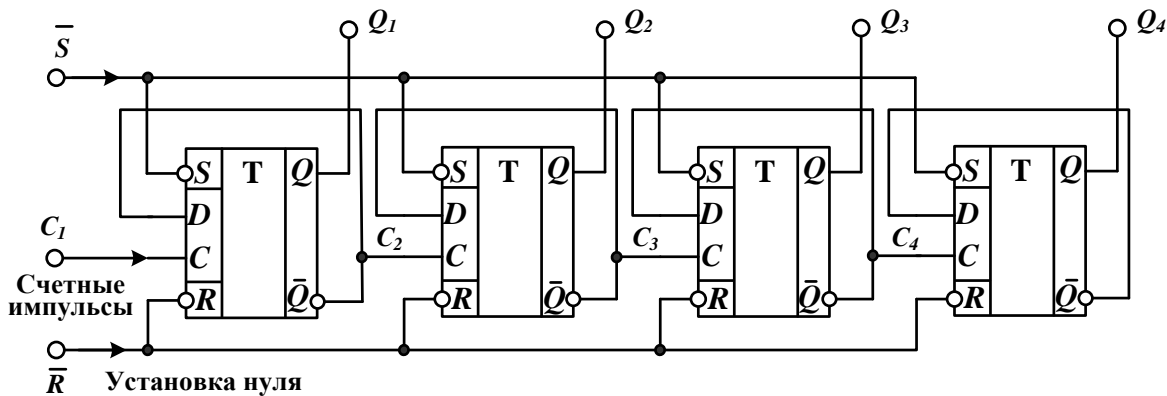


Рис. 1.16. Асинхронный счетчик на D-триггерах

В начале счета все триггеры устанавливаются в нуль и с входов \bar{R} и \bar{S} снимаются активные уровни. На счетный вход C_1 первого триггера поступают счетные импульсы. Каждый импульс изменяет состояние триггеров так, что на выходах $Q1 - Q4$ формируется двоичный код, соответствующий числу счетных импульсов. Длина списка разрешенных состояний счетчика называется модулем счета $K_{сч}$. Число разрешенных состояний определяется количеством триггеров. Для счетчика (рис. 1.16) $K_{сч} = 2^4 = 16$. После 16 импульсов все триггеры обнуляются и начинается новый цикл счета. Счетчики можно использовать в качестве делителей частоты с коэффициентом деления $K_{сч}$. Введением дополнительных обратных связей между триггерами можно произвольно изменять модуль счета в сторону уменьшения. Так введение в обратные связи двух элементов 2И-НЕ устанавливает в счетчике (рис. 1.17) модуль счета равный 9.

В цифровых устройствах связи применяют разнообразные счётчики. Они классифицируются следующим образом:

- по модулю счета: двоичные, двоично-десятичные (декадные) или с другим основанием счета; с произвольным постоянным модулем; с переменным модулем.

- по направлению счета: суммирующие; вычитающие; реверсивные.

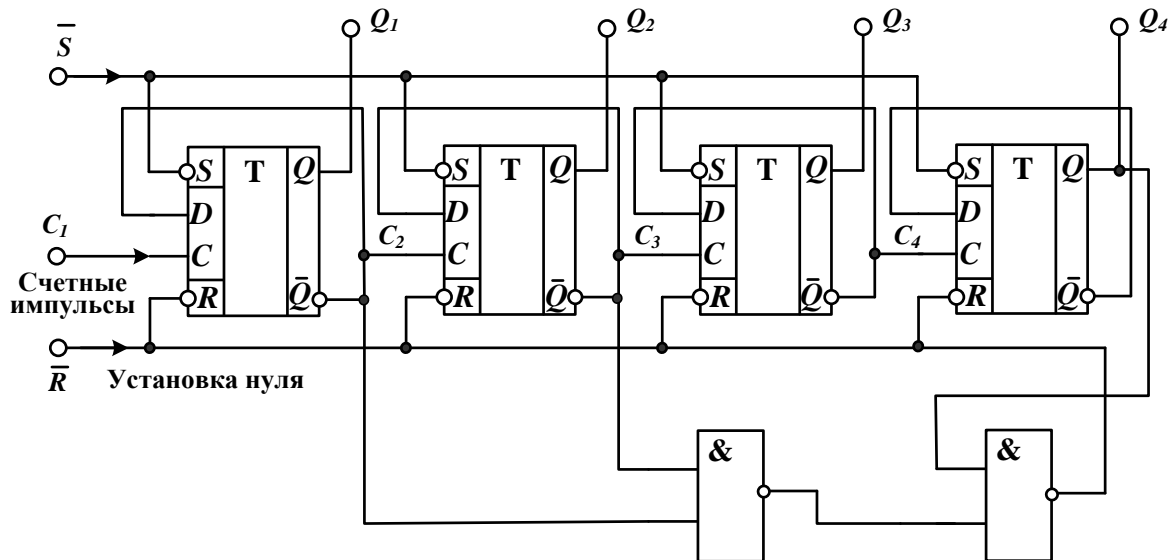


Рис. 1.17. Асинхронный счётчик с модулем счета $K_{сч}=9$

1.5.8. Регистры

Регистром называется устройство цифровой техники, предназначенное для записи, хранения и (или) сдвига информации, представленной в виде многоразрядного двоичного кода.

По способу приёма информации регистры подразделяют на:

последовательные (сдвигающие), в которых информация записывается и считывается только в последовательной форме;

параллельные (статические), в которых информация записывается и считывается только в параллельной форме;

последовательно-параллельные, в которых информация записывается или считывается как в последовательной, так и в параллельной формах.

Простейшие регистры выполняют на триггерах. Схема последовательного сдвигающего регистра на *JK*- триггерах показана на рис. 1.18.

Сдвигающий регистр работает следующим образом. В начале работы управляющими сигналами S и R все триггеры устанавливаются в нулевое состояние, активный уровень сигналов S и R снимается, на вход D (данные) подается первый импульс цифрового кода (например, единица кода 1101 на рис. 1.18). С первым тактовым импульсом, поступающим на вход C , в первый триггер будет записана единица младшего разряда.

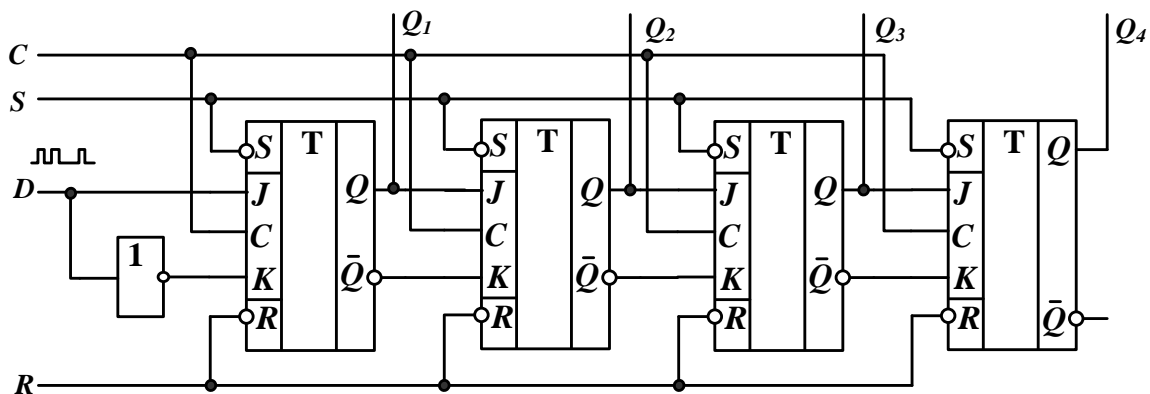


Рис. 1.18. Четырёхразрядный сдвигающий регистр с последовательным вводом

Со следующим тактовым импульсом эта единица будет записана во второй триггер и окажется на его выходе. Одновременно в первый триггер поступит нуль (следующий разряд кода). После четырёх тактовых импульсов код на выходах Q_4-Q_1 будет соответствовать входному коду и может быть считан внешним устройством. Таким образом, регистр преобразует последовательный код в параллельный. В современных микроэлектронных устройствах отображения и регистрации информации количество параллельных выходов регистров может достигать нескольких тысяч.

1.6. Цифровые постоянные запоминающие устройства (ROM)

Блоки запоминающих устройств (ЗУ) служат для запоминания больших массивов данных и становятся важнейшей составляющей частью цифровых компьютеров.

Схемы ЗУ подключаются, как правило, к шинной системе. Благодаря этому большое число различных ЗУ может быть подключено параллельно. Система шин должна давать возможность записывать в определенную ячейку ЗУ или считывать из неё. При этом различают:

Адресные шины

Адресная шина подключается к каждому узлу ЗУ адресов, под которыми могут храниться данные или проводиться их поиск.

Управляющие шины

Управляющая шина содержит все линии, необходимые для управления схемным узлом. К ним относится линия, обеспечивающая выбор узла и содержащая вывод для выбора чипа (chip select, CS).

Поскольку все узлы подключены к одной и той же адресной шине, то с помощью этой линии производится выбор соответствующего узла. С помощью линии считывание-запись можно осуществлять переключение между чтением и записью (read-write, RD-W).

Шины данных

Шина данных подключается ко всем схемным узлам. Чтобы в линиях не создавались конфликты, выходы блоков ЗУ, влияющие на шину данных, должны быть выходами типа tristate («с тремя состояниями»). Их делают свободными для включения с помощью линии output enable (OE, «разрешающий выход»), входящей в управляющую шину.

1.6.1. Постоянное ЗУ (ПЗУ)

Постоянное ЗУ называют ROM (read only memory), что означает ЗУ только с чтением.

ROM является схемой ЗУ, содержание данных в котором определяется изготовителем с помощью маски. Поэтому содержание данных фиксировано и их можно только читать. Запомненные данные являются энергонезависимыми. ROM-схемы различаются числом битов, которые запомнены. Обычно используются ЗУ с длинами слов 1, 4, 8 и 16 бит. В ROM, как и в большинстве других цифровых ЗУ, отдельные ячейки ЗУ расположены в виде матрицы (рис. 9.33). Ячейки ЗУ лежат в точках пересечений линий. К ним обращаются, подав напряжения 1 на строчную линию и на линию столбцов. Строчную линию называют числовой шиной (word line), линию столбцов называют шиной данных (data line). Преимущество данного устройства заключается в экономии шин. Для n^2 ячеек требуется $2n$ шин против n^2 в линейных устройствах.

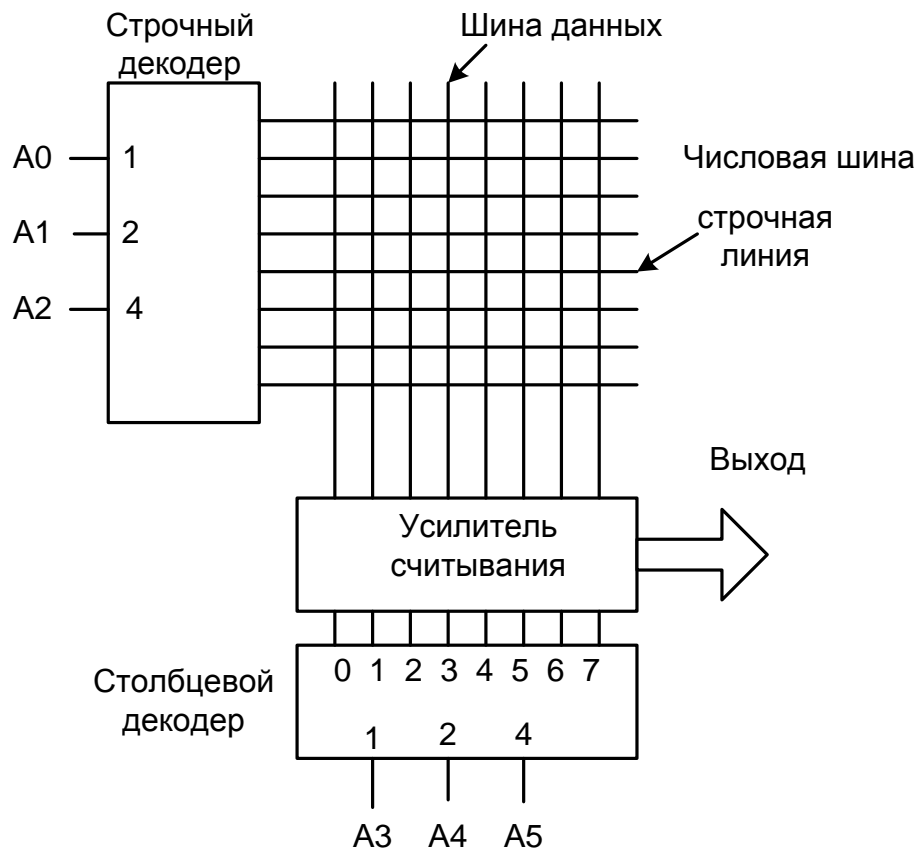


Рис. 1.19. Структурная схема ROM ёмкостью 8x8

Распределение числовых шин (word line) по адресам осуществляет строчный декодер. Столбцовый декодер осуществляет выбор шин данных и работает как мультиплексор. Поскольку шина данных наряду с выбором столбца осуществляет вывод накопленной информации к выходу, между декодером и матрицей ЗУ включается усилитель считывания. На рис. 1.19 показано ROM ёмкостью 8x8. Данные выводятся последовательно по сигналу чтения RD.

ПЗУ часто производят в N-МОП –технологии. На пересечении линий строк и столбцов в местах, соответствующих сигналу «0» формируют N-МОП транзисторы. В местах, соответствующих сигналу «1», транзисторы отсутствуют (рис. 1.20).

В другом типе ПЗУ запись информации производится при помощи последней операции производства микросхемы — металлизации. Металлизация производится при помощи маски, поэтому такие ПЗУ получили название **масочных ПЗУ**.

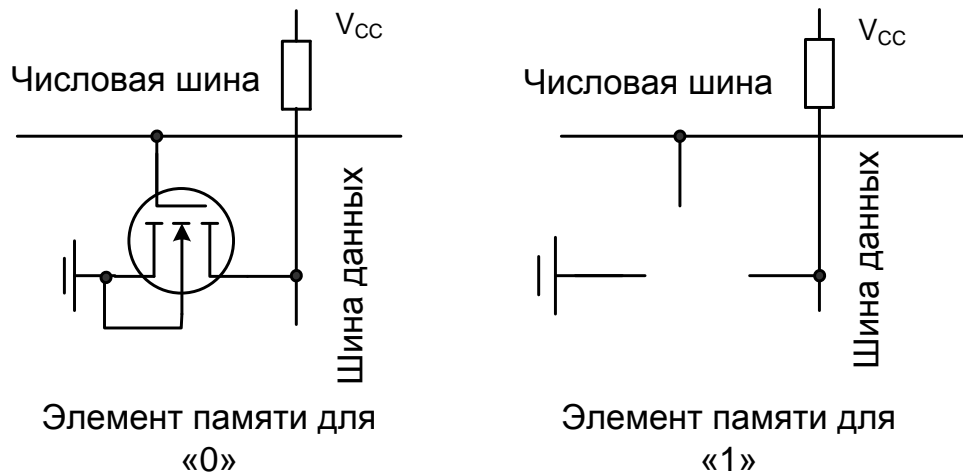


Рис. 1.20. Элемент памяти ПЗУ (N – МОП - технология)

На микросхемах ПЗУ обозначают символом ROM.

1.6.2. Программируемое ПЗУ (ППЗУ), PROM

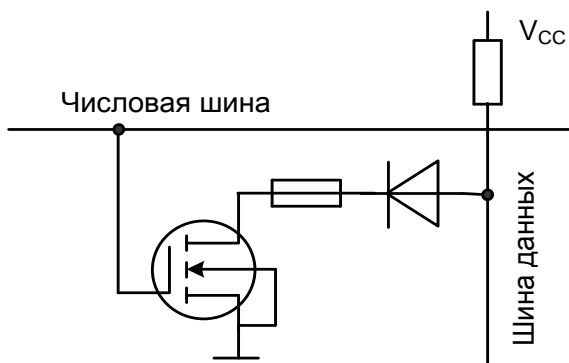


Рис. 1.21. Ячейка ПЗУ типа PROM

PROM соответствует по структуре ROM, но с тем отличием, что она может программироваться пользователем. PROM также выполнена в виде матрицы, со столбцевым и строчным декодерами для адресного декодирования. Возможная реализация может быть разработана на основе рис. 1.19. Стоки транзисторов в ячейках ПЗУ PROM могут

быть закоротены вместо прямого соединения проводящими дорожками с плавкими перемычками (fusible link).

Это соединение показано на рис. 1.21. В том случае, если должно быть запомнено состояние Н, плавкая перемычка при программировании расплавляется током. Как правило, для этого необходимо специальное программирующее устройство. Плавкая перемычка подобна плавкому предохранителю. Записанная информация является энергонезависимой. Однажды расплавленная плавкая перемычка не может быть вновь восстановлена. Поэтому подобные схемы обозначаются как OTP-ROM (OTP — one time programmable), то есть как ROM с однократным программированием.

1.6.3. Стираемое программируемое ПЗУ (EPROM)

Через EPROM обозначают erasable programmable ROM, стираемое программируемое ПЗУ, СППЗУ. EPROM по своей структуре соответствует ROM или PROM только на месте плавких перемычек или программируемых с помощью маски соединений находятся «стираемые» запоминающие элементы.

В качестве таких запоминающих элементов используются МОП ПТ с плавающим затвором (рис. 1.22). Эти МОП ПТ являются нормально-закрытыми транзисторами с дополнительным затвором, который не имеет внешнего вывода и называется плавающим затвором.

Изначально этот затвор имеет нулевой потенциал. С помощью наведения заряда на плавающем затворе может быть осуществлена запись информации в ячейки.

При отсутствии заряда на плавающем затворе транзистор функционирует как обычный n-канальный транзистор нормально-закрытого типа. Достаточно большое положительное напряжение на затворе включает транзистор. Это состояние ведёт к уровню L на шине данных в том случае, если данная ячейка ЗУ выбрана числовой шиной.

В том случае, если должен быть запомнен уровень Н, на плавающем затворе должен быть накоплен отрицательный заряд. Теперь транзистор всегда заперт и при выборе ячейки с помощью числовой шины шина данных остается на уровне Н. Отрицательный заряд на плавающем затворе создается за счёт туннелирования электронов через оксид. При повышенном напряжении между стоком и подложкой поле между затворным электродом и каналом достигает высоких значений и за счёт лавинного пробоя появляются «горячие» электроны.

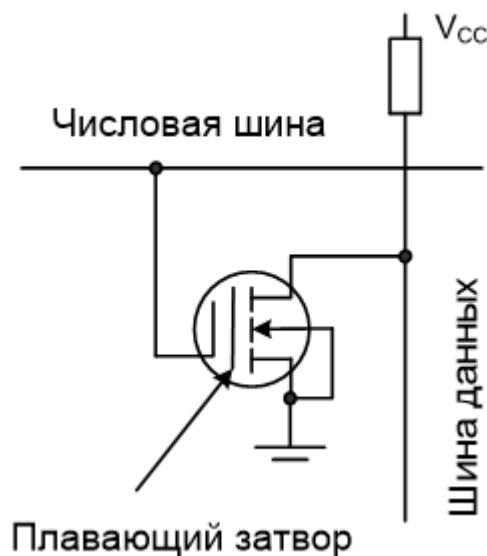


Рис. 1.22. МОП ПТ с плавающим затвором

Определённое количество электронов при этом может туннелировать через оксид. На плавающем затворе таким образом накапливается отрицательный заряд, который запирает транзистор. Облучение УФ светом в течение приблизительно 20 минут ионизирует изолирующий материал, лежащий между затвором и истоком, благодаря чему заряд рассасывается. При этом информация стирается. Благодаря хорошим диэлектрическим свойствам оксида накопленный заряд остаётся стабильным в течение многих лет.

1.6.4. EEPROM - электрически стираемое программируемое ПЗУ

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Через EEPROM обозначают электрически стираемое программируемое ПЗУ, ЭСППЗУ (electrically erasable programmable ROM). Эта схема ЗУ электрически записывается и электрически стирается. Единичная ячейка ЗУ также как и в EPROM, построена на одном МОП ПТ с плавающим затвором. Однако оксид между плавающим затвором и каналом имеет меньшую толщину. Благодаря этому оказывается возможным при повышенном напряжении между затвором и каналом перемещать электроны с затвора в канал и обратно. Это происходит за счет эффекта туннелирования.

EEPROM со специальными ячейками ЗУ иногда называют flash-EEPROM («мигающее» EEPROM). Они стираются не только полностью, но и поблочно.

Коммерчески доступные ROM, PROM, EPROM, EEPROM часто совместимы по выводам корпуса так, что возможно применение в фазе разработки PROM или EPROM, которые при переходе к серийной продукции можно заменить на ROM или на PROM. Логический символ EEPROM идентичен логическому символу ROM.

1.6.5. Обзор энергонезависимых ЗУ

В таблице 1.3 показан обзор энергонезависимых ЗУ.

Таблица 1.3

Обозначение	Программирование		Стирание
	Маска	Однократное	
ROM (read only memory)	Маска	Однократное	Невозможно
PROM (программируемое ROM), программируемое полем ROM, однократное PROM (OTPROM)	Эл.	Однократное	Невозможно

EPROM (erasable ROM)	Эл.	Многократное	УФ облучение (20 минут), все содержимое ЗУ
EEPROM (electrically erasable ROM) flash-EEPROM	Эл.	Многократное	Электрическое, все содержи- мое ЗУ или по битам (20-100 мс)

1.7. Оперативные запоминающие устройства ОЗУ (RAM)

RAM является сокращением для random access memory (то есть ЗУ с произвольной выборкой, ЗУПВ). Под этим понимается схема ЗУ, в которой можно произвольно записывать и из которой можно произвольно считывать информацию. Ячейки памяти выбирают по их адресам. В ячейках хранится записанная информация. Для вывода данных ячейки также выбирают по адресам. Считывание данных не стирает содержание ячейки. Если информация больше не нужна, она может быть удалена, а в ячейку записана новая информация.

ОЗУ выпускают в виде интегральных микросхем. Различают статические RAM и динамические RAM (DRAM). В статических RAM в качестве ячеек памяти применяют триггеры. Схемы с высокой степенью интеграции работают в основном с использованием динамического запоминания информации в конденсаторах, управляемых одним транзистором.

1.7.1. Статическое ОЗУ RAM

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Ячейка статического ЗУ RAM показана на рис. 1.23. Она построена из двух КМОП-инверторов. Обращаться к ячейке ЗУ можно через числовую шину.

Для выбора ячейки на числовую шину подается Н. Вследствие этого T_5 и T_6 становятся низкоомными. В процессе записи можно, подав уровень Н на шину данных DL, записать Н в ЗУ. В таком случае T_3 будет открыт, а T_4 — закрыт. Правый инвертор выдаст L. После этого будет настроен на Н левый инвертор. Схема работает как триггер. Точно также можно записать L в ЗУ, подав Н на шину данных \overline{DL} . При считывании ячейка вновь выбирается с помощью числовой шины. На шинах данных может быть считан запомненный бит.

Для считывания с обеих шин данных используется один симметричный усилитель считывания. Имеется также усилитель записи.

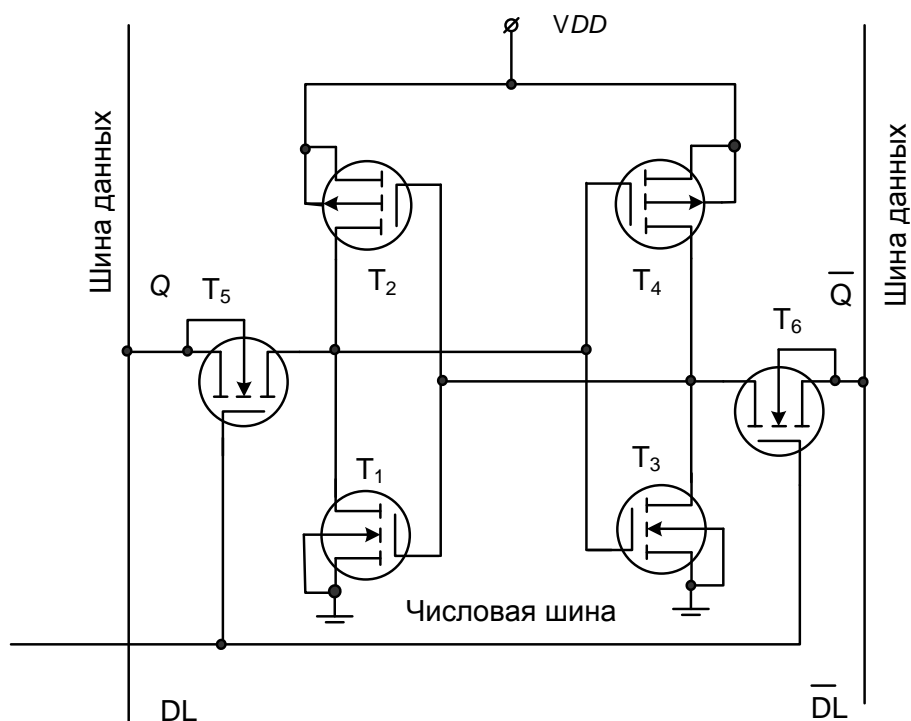


Рис. 1.23. Ячейку ЗУ типа RAM, выполненная по КМОП-технологии

1.7.2. Динамические ОЗУ (DRAM)

DRAM (динамическое RAM) представляет собой энергозависимое полупроводниковое ЗУ, в котором информация запоминается в конденсаторах. Благодаря очень простой структуре ячейки ЗУ подобные DRAM имеют очень большую информационную емкость. Запоминающий DRAM элемент показан на рис. 1.24. Приложение Н к числовой линии позволяет выбрать ячейку ЗУ. Тогда накопленный на конденсаторе заряд может стекать по шине данных. Наличие заряда означает содержимое ЗУ со значением Н, а значению L соответствует отсутствие заряда. Чтение разрушает накопленный заряд, так что после каждого считывания заряд должен быть снова восстановлен (регенерирован).

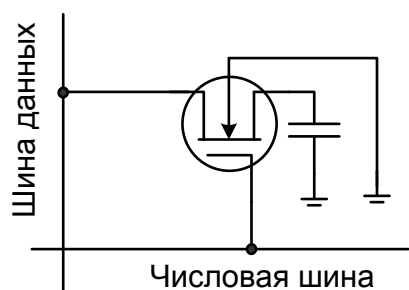


Рис. 1.24. Ячейка динамического RAM

Динамическое RAM организовано так, что процесс считывания автоматически связан с регенерацией заряда. Если ячейка ЗУ не считывается определенное время, заряд рассасывается, и информация теряется. Поэтому все запомненное содержимое должно периодически регенерироваться с помощью процесса считывания. Поскольку сопротивление МОП ПТ в выключенном состоянии очень велико,

то достаточно очень малых емкостей конденсаторов, чтобы поддерживать время разряда в диапазоне миллисекунд.

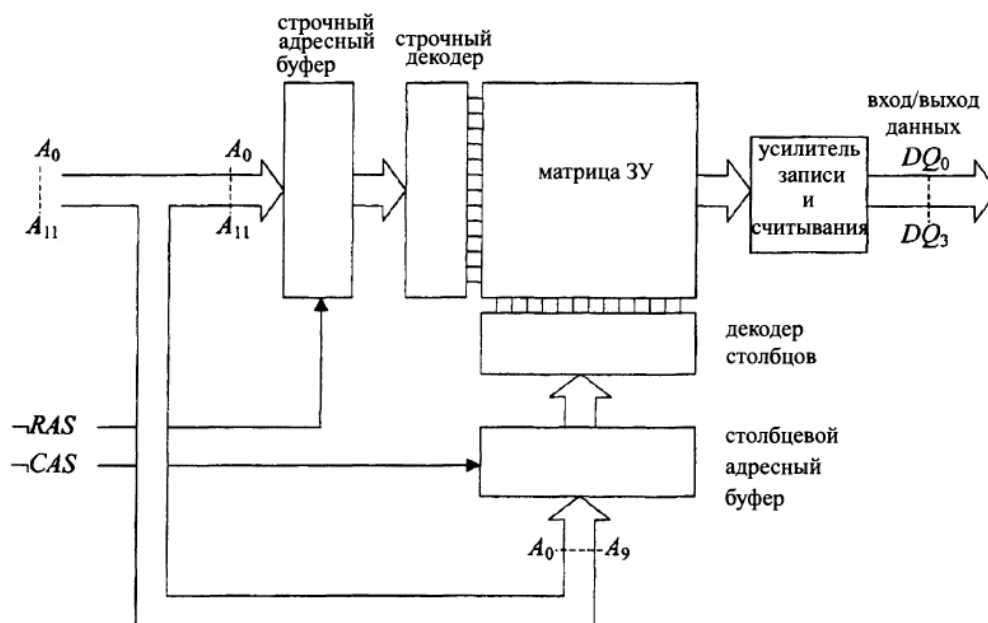


Рис. 1.25. Принципиальная схема DRAM с емкостью 4М x 4 бит

В качестве примера на рис. 1.25 представлена схема DARM емкостью 4М x 4 бит. Данные в каждой ячейке ЗУ должны регенерироваться каждые 64 мс. В эту схему строчные и столбцевые адреса должны загружаться через те же самые контактные площадки, чтобы схема была достаточно малой площади. Как показано на принципиальной схеме (рис. 1.25), в DRAM для выборки столбцов используется 10 бит и для выборки строк 12 бит. Для введения строчных адресов используется контактная площадка \overline{RAS} (row address strobe, выборка строчных адресов), а для введения столбцевых адресов контактная площадка \overline{CAS} (column address strobe, выборка столбцевых адресов). С помощью усилителя записи и считывания вводятся и считываются слова длиной 4 бита.

DRAM работает в режимах: считывание, запись, регенерация, хранение. Для каждого режима заданы временные диаграммы выдачи адресов и управляющих импульсов. Эти диаграммы надо соблюдать при проектировании устройств с оперативной памятью.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

1.8. Программируемые логические схемы

Для выполнения специальных задач требуются интегральные микросхемы с различными функциями алгебры логики (ФАЛ). Выпускать ограниченное количества ИС с заданной ФАЛ для конкретного заказчика очень дорого и нерентабельно. Поэтому разработаны микросхемы, которые программируются производителем или самими пользователями.

1.8.1. Логические схемы, программируемые изготовителем

Схемы с большими логическими структурами называют программируемыми вентильными матрицами. Это схемы полуфабрикаты. Имеющиеся на них вентили могут быть соединены друг с другом по желанию пользователя. Линии связи производят масками - это стеклянные фотошаблоны, на которые нанесены фотографически уменьшенные логические структуры. Эти маски переносят на полупроводниковые пластины, используя процесс фотолитографии, и такое программирование называют масочным.

Матричные кристаллы, базовые ячейки и схемы, ориентированные на заказчика – это разновидности ASIC микросхем (Application Specific Integrated Circuit). Микросхемы встраивают в корпус, который имеет от 16 до 144 выводов. Особенно высокую плотность размещения вентиляей можно достигнуть с N-МОП технологией.

1.8.2. Логические схемы, программируемые потребителем

Логические схемы, программируемые потребителем, называют PLD (Programmable Logic Devices) – программируемые логические устройства.

Мы изучали, что логические связи любой схемы могут быть выражены в *Совершенной дизъюнктивной нормальной форме* (СДНФ), которую получают из таблицы истинности логическим суммированием конститuent единицы

(записанных по определенному алгоритму логических произведений входных переменных, при которых выходная функция равна «1»).

СДНФ можно реализовать, используя логические элементы И и ИЛИ, соединенные определенным образом с помощью *программируемой логической матрицы* (ПЛМ).

На рис. 1.26 показана схема ПЛМ с тремя входными сигналами X_1, X_2, X_3 и двумя выходными Y_1, Y_2 .

Схема содержит две программируемые логические матрицы в виде сетки перпендикулярных проводников. В местах пересечения проводников находятся полупроводниковые элементы – диоды и транзисторы, включенные через легкоплавкие перемычки к проводникам матрицы. При программировании ненужные перемычки пережигаются импульсами тока.

Матрица М1 работает следующим образом. Если на горизонтальную шину подана «1», то диод заперт, и ток через R не течет. На вертикальной шине, соединенной с анодом запертого диода, остается высокий потенциал, соответствующий «1». Если на горизонтальной шине низкий потенциал «0», то протекающий через R и открытый диод ток создает падение напряжения на R и на вертикальной шине будет «0». Для сохранения высокого потенциала на вертикальной шине матрицы М1 необходимо, чтобы все диоды этой шины были заперты, т.е. на всех входах была логическая «1». Таким образом на вертикальных шинах реализуется операция И от аргументов, помеченных крестиками, в которых диоды действуют. Матрицу М1 называют матрицей И. В примере (рис.9.40) на выходах P_i реализуются функции:

$$P_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3; P_2 = x_1 x_3; P_3 = x_1 x_2; P_4 = \bar{x}_1 x_2 x_3$$

Конъюнкции, образованные на выходах матрицы И, называют термами.

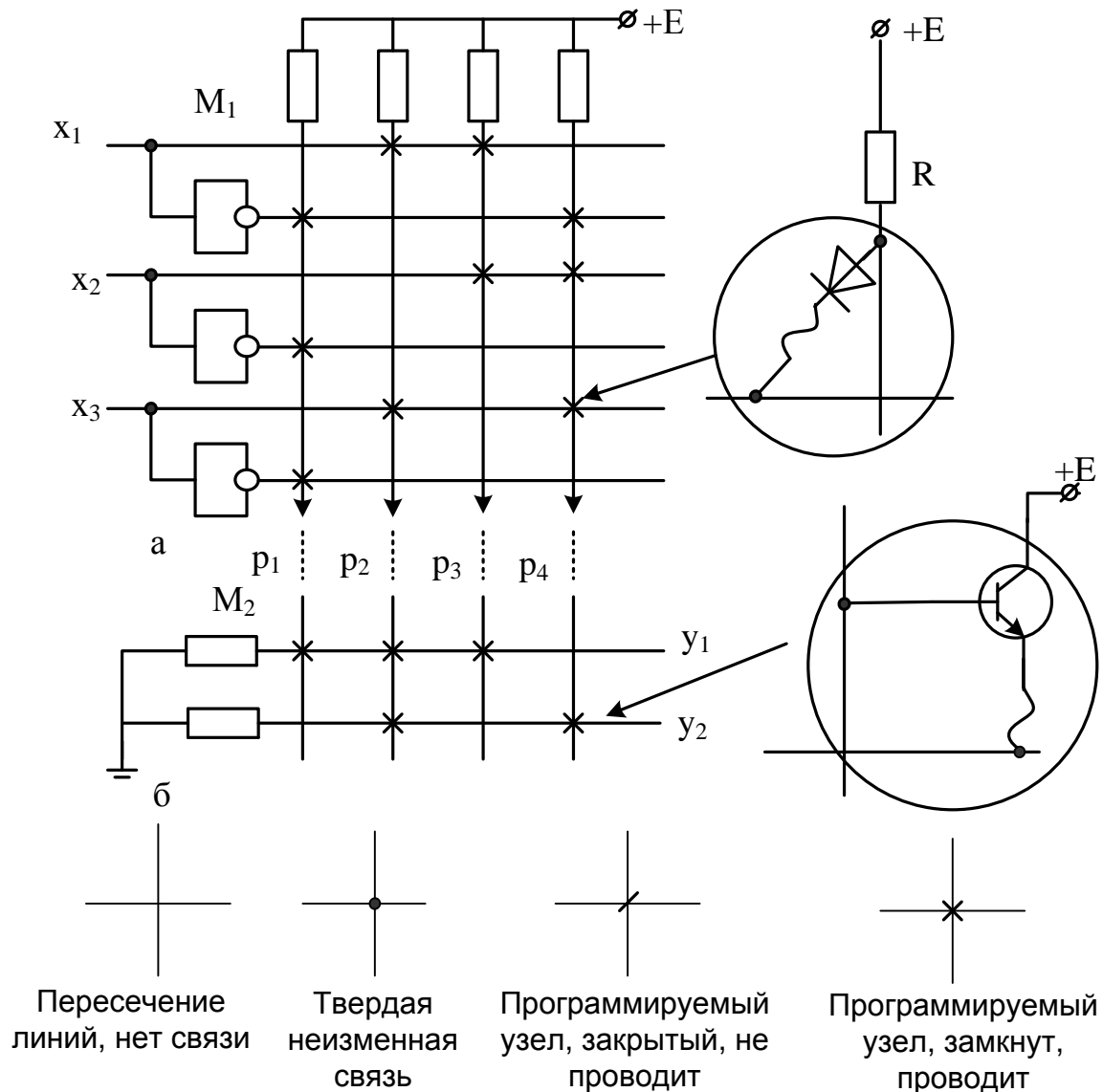


Рис. 1.26. Программируемые логические матрицы:

а- матрица И; б – матрица ИЛИ

Рассмотрим работу матрицы M_2 . Если на вертикальной шине высокий потенциал, то транзистор открыт и высокий потенциал $+E$ поступает на горизонтальную шину y , т.е. на ней будет «1». Если транзистор закрыт (на затворе «0»), то на шине y будет логический «0». Открытие любого транзистора приводит к появлению на горизонтальной шине y потенциала «1», поэтому на

ней реализуется функция ИЛИ от входных переменных P_i , помеченных крестиками. Так как каждая из переменных P_i реализует функцию И над аргументами x , то в итоге на выходах y реализуются функции:

$$y_1 = P_1 + P_2 + P_3 = \bar{x}_1\bar{x}_2\bar{x}_3 + x_1x_3 + x_1x_2 \quad \text{и}$$

$$y_2 = P_2 + P_4 = x_1x_3 + \bar{x}_1x_2x_3.$$

Таким образом, ПЛМ осуществляет реализацию СДНФ логических функций. Основными параметрами ПЛМ являются число входов аргументов x , число выходов y и число термов (переменных P), которые могут входить в выходную функцию y .

Наряду с матрицами в ПЛМ входят блоки входных и выходных буферных каскадов. Входные буферы дают прямое и инверсное представление аргументов, формируют необходимую мощность для питания элементов матрицы. Выходные буферы обеспечивают необходимую нагрузочную способность выходов, разрешают или запрещают выходы ПЛМ на внешние шины.

ПЛМ могут выполняться как на биполярных, так и на МОП- транзисторах. Программирование ПЛМ заключается в пережигании перемычек, соединяющих горизонтальные и вертикальные шины матриц, которое выполняется в программаторах. Если аргумент не участвует в формировании переменной P , пережигают две перемычки, а если участвует – пережигают одну из перемычек для прямого или инверсного значения аргумента. Если переменная P входит в функцию y , то перемычку сохраняют, а если не входит – пережигают.

1.8.3. Разновидности программируемых логических устройств

К настоящему времени создано и используется значительное число разнообразных программируемых логических устройств. Средства программируемой матричной логики получили свое развитие в направлении, называемом «программируемые логические интегральные схемы» (ПЛИС), иностранное название PLD. Рассмотрим кратко основные их них.

Программируемая матричная логика – ПМЛ (Programmable Array Logic – PAL). В ПМЛ (рис. 1.27) выходы элементов И первой матрицы жестко расположены между элементами ИЛИ (входами матрицы ИЛИ). Для m входов и n выходов требуется m инверторов, n схем ИЛИ и $4n$ элементов И. По сравнению с ПЛМ эта схема имеет меньшую функциональную гибкость, но ее изготовление и использование проще.

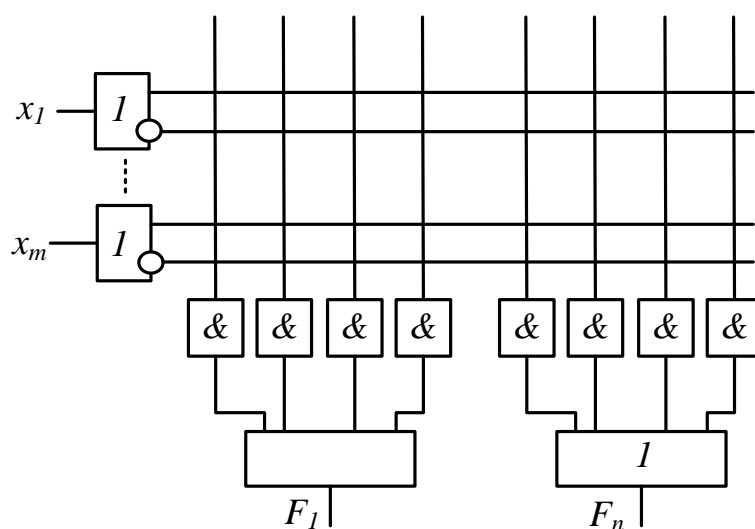


Рис. 1.27. Базовая структура ПМЛ

Логика на базовых матричных кристаллах GAL (Genetic Array Logic). GAL-схемы были спроектированы с целью улучшить некоторые известные недостатки PAL-схем. Программирование прожигом очень тонких перемычек не применяется. GAL-схемы являются полностью EPLD (стираемые ультрафиолетом Erasable PLD) или EEPLD (стираемые электрически Electrical

Erasable PLD). Их можно программировать более ста раз. GAL-используют программируемую И-матрицу, а матрица ИЛИ фиксирована.

Сложные программируемые логические устройства CPLD (complex programmable logic device). Содержат центральную коммутационную матрицу, множество функциональных блоков, в которых осуществляется логическая обработка сигналов, и блоки ввода/вывода, расположенные на периферии кристалла и осуществляющие прием и передачу сигналов между кристаллом микросхемы и контактными площадками.

Программируемые пользователем вентильные матрицы FPGA (field-programmable gate array) содержат блоки умножения-суммирования, которые широко применяются при обработке сигналов, а также логические элементы (как правило, на базе таблиц перекодировки — таблиц истинности) и их блоки коммутации. FPGA обычно используются для обработки сигналов, имеют больше логических элементов и более гибкую архитектуру, чем CPLD. Программа для FPGA хранится в распределённой памяти, которая может быть выполнена как на основе энергозависимых ячеек статического ОЗУ (в этом случае программа не сохраняется при исчезновении электропитания микросхемы), так и на основе энергонезависимых ячеек flash-памяти или перемычек antifuse (в этих случаях программа сохраняется при исчезновении электропитания). Если программа хранится в энергозависимой памяти, то при каждом включении питания микросхемы необходимо заново конфигурировать её при помощи начального загрузчика, который может быть встроен и в саму FPGA. Альтернативой ПЛИС FPGA являются более медленные цифровые процессоры обработки сигналов. FPGA применяются также, как ускорители универсальных процессоров в суперкомпьютерах.

СБИС система на кристалле (СнК – система-на-кристалле) , (SoC – System on Chip), интегральная схема, содержащая компоненты компьютерных
Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

или иных электронных систем на одном кристалле. Может включать в себя цифровую, аналоговую и радиочастотную функции. Состоит, как правило, из готовых электронных компонент - специализированных СБИС (ASIC), полужаказных ИС (ПЛИС, БМК), ИС средней и малой степени интеграции, а также дискретных элементов, размещенных на одной подложке интегральной схемы. SoCs очень распространены на рынке мобильной электроники из-за их низкого энергопотребления, для более мощных процессоров, способных работать с программным обеспечением, таких как настольные версии Windows, и Linux. Высокая степень интеграции чипа приводит к снижению затрат на производство и позволяет реализовать производство целых цифровых систем, включающих процессор, память, интерфейсное устройство.

Основой для реализации СБИС «система на кристалле» стало направление FPGA. Производители ПЛИС обозначают их как усовершенствованные СБИС с архитектурой FPGA, которые содержат сложные функциональные узлы, выполненные в виде аппаратных ядер.

1.9. Аналого - цифровые и цифроаналоговые преобразователи

Электронные устройства телекоммуникационных систем делятся на аналоговые и цифровые. В системах управления различными процессами участвуют устройства обоих типов. Аналоговые устройства используют для съёма первичной информации с микрофонов в звуковых системах, информационных датчиков систем управления, телеметрии и т.п. Передачу информации в современных цифровых системах связи выполняют цифровые устройства. После цифровой обработки сигналов и их приёма требуется обратное преобразование в аналоговую форму для воспроизведения звуковых сигналов в динамиках, включения двигателей постоянного или переменного тока и т.п. Взаимодействие между аналоговой частью системы и цифровой

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

обеспечивают аналого-цифровые преобразователи (АЦП) и цифроаналоговые преобразователи (ЦАП).

1.9.1. Принцип аналого-цифровое преобразования

Аналого-цифровые преобразователи (АЦП) преобразуют аналоговую информацию в цифровую.

Теоретической основой аналого-цифрового преобразования является **теорема В.А. Котельникова:**

Аналоговый непрерывный сигнал с ограниченным спектром можно представить последовательностью отсчетов с периодом дискретизации

$T \leq \frac{1}{2f_B}$, где f_B - верхняя частота спектра сообщения.

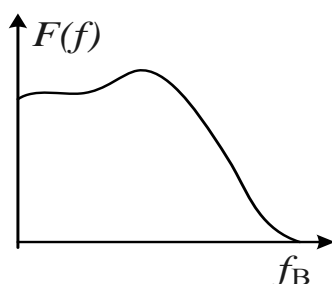


Рис. 1.28. Ограниченный спектр сигнала

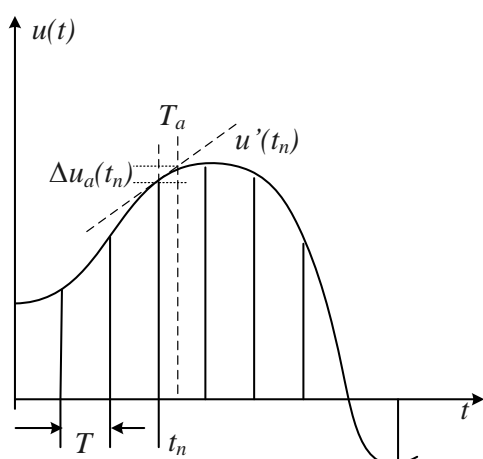
Верхнюю частоту спектра определяют так, чтобы практически вся энергия сигнала была сосредоточена в полосе частот от нуля до f_B (рис. 1.28). Например, аналоговый звуковой сигнал находится в частотном диапазоне от 50 Гц до 20 кГц. Для его «оцифровки» частота отсчётов должна быть не менее 40 кГц. В этом случае на приемной стороне импульсный сигнал из отсчетов можно с высоким качеством преобразовать обратно в аналоговый. В современных

цифровых системах звукозаписи частота отсчетов (дискретизации) составляет не менее 44кГц.

1.9.2. Этапы аналого-цифрового преобразования

1. Дискретизация по времени

На первом этапе проводят дискретизацию аналогового сигнала по времени и представляют исходный сигнал в виде совокупности отсчетов. Для этого в АЦП присутствует блок или устройство выборки и хранения (УВХ), в котором в момент поступления тактирующего импульса текущее значение напряжения аналогового сигнала запоминается на емкости.



Процесс дискретизации по времени может сопровождаться апертурной погрешностью, которую рассчитывают следующим образом.

На рис. 1.29 показана временная дискретизация аналогового сигнала с периодом T .

Параметры временной дискретизации:

T_a - апертурное время преобразования;

$u'(t_n)$ - скорость изменения сигнала;

$\Delta u_a(t_n)$ - апертурная погрешность.

Рис. 1.29. Апертурная погрешность

временной дискретизации

погрешность. Для гармонического сигнала относительная апертурная

погрешность:

$$\delta_a = \frac{\Delta u_a}{U_m} = \omega T_a$$

2. Квантование по амплитуде

Для того, чтобы дискретные отсчёты преобразовать в цифровые значения, выполняют квантование по амплитуде, т.е. представляют непрерывный аналоговый сигнал в виде ступенчатого сигнала (рис. 1.30). Высоту минимальной ступеньки называют шагом квантования h . Максимальная погрешность квантования $\pm 0,5h$. Сред-

некватратическая погреш-

ность $\sigma_{kv} = \frac{h}{\sqrt{12}}$.

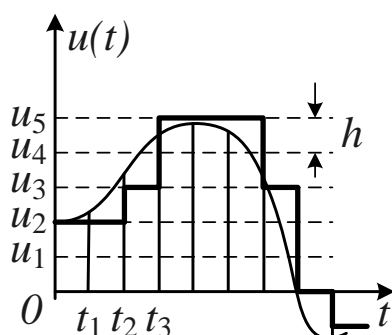


Рис. 1.30. Квантование по амплитуде

Апертурная погрешность будет меньше погрешности квантования, если:

$$T_a < \frac{h}{2U_m \omega_m}$$

Многие современные микроконтроллеры имеют встроенные модули АЦП.

Важной характеристикой АЦП является разрядность. Разрядность показывает на сколько дискретных значений или уровней разделен диапазон изменения входного аналогового сигнала. Дискретные значения хранятся обычно в бинарном коде, их количество является степенью 2 и поэтому разрешение выражают в битах. Так АЦП с разрядностью 8 бит может

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

кодировать аналоговый сигнал на $2^8=256$ уровней. Значения могут быть в диапазоне от 0 до 255 (т.е. целое без знака) или от -128 до 127.

Разрешение АЦП может быть выражено через электрические величины. Разрешение в вольтах, которое называют шаг квантования, равно полному диапазону измерения напряжения, деленному на количество дискретных интервалов:

$$h = \frac{U_{max} - U_{min}}{2^m} = \frac{U_{max} - U_{min}}{N},$$

где: h – шаг квантования (дискретное значение напряжения в вольтах, приходящееся на единичное изменение выходного кода);

$(U_{max} - U_{min})$ – диапазон изменения входного напряжения;

m – разрядность АЦП;

N – количество интервалов (число градаций), на которые делится входной сигнал.

Например, используя АЦП с разрядностью 12 бит, входное напряжение в диапазоне 0 – 10В можно преобразовать на $2^{12}=4096$ квантованных уровня. Разрешение АЦП в вольтах составит:

$$h = \frac{U_{max} - U_{min}}{2^m} = \frac{10 - 0}{4096} = 2,44 мВ.$$

Шаг квантования, равный напряжению младшего разряда кода, обозначают U_{LSB} (least significant bit).

АЦП характеризуют следующими параметрами:

разрядность – количество бит;

разрешающая способность;

точность – ошибка в % от результата, или % от максимального значения;
скорость – длительность преобразования;
диапазон от мин. до макс. преобразуемого напряжения.

1.9.3. АЦП прямого параллельного преобразования

АЦП прямого параллельного преобразования являются наиболее быстрыми и работают с компараторами, которые всегда переключают выходное состояние «0» в выходное состояние «1», если на положительном входе напряжение равно или больше, чем на отрицательном. Каждый выходной бит привязан к отдельному компаратору. Схема АЦП показана на рис. 1.31.

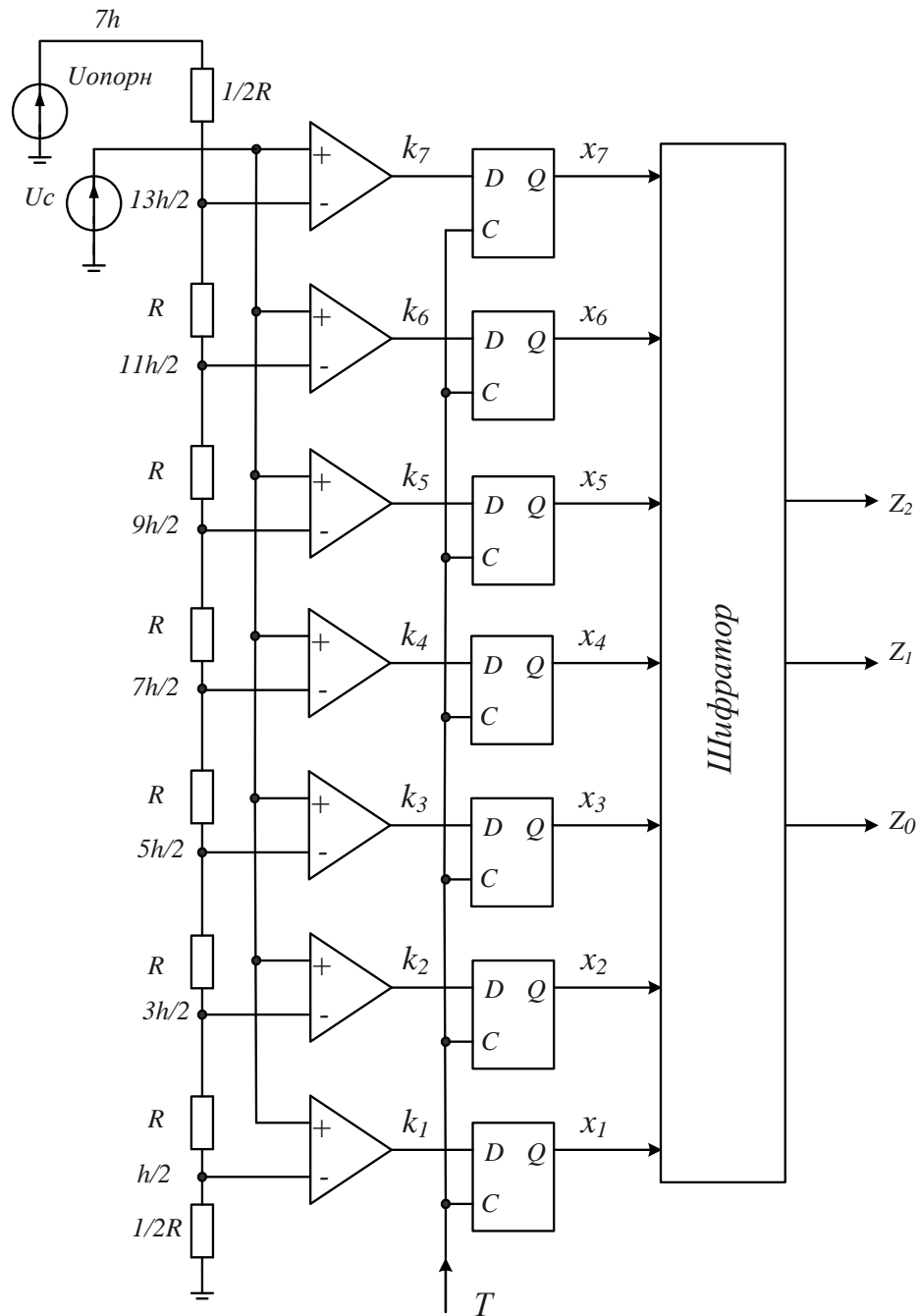


Рис. 1.31. АЦП прямого параллельного преобразования

АЦП преобразует аналоговое напряжение U_c в трехразрядный код. На делитель из резисторов подано опорное напряжение $U_{опорн} = 7h$, где h - шаг квантования. Компараторы на ОУ сравнивают входное напряжение U_c с

напряжениями на резисторах делителя и устанавливают «1», если $U_c > (n - \frac{1}{2})h$. Тактовый импульс записывает состояния компараторов в триггеры. Шифратор преобразует данные в трёхразрядный код. В результате получаем 8 уровней квантования.

АЦП прямого преобразования работают быстрее, чем все другие виды АЦП. Время преобразования определяется временем переключения компараторов. Оно равняется от 40 до 50 нс. Значит, при 100 нс на одно преобразование каждую секунду возможны 10 миллионов преобразований. Стоимость такого АЦП большая. Для аналого-цифрового преобразования с 256 уровнями напряжения нужен АЦП с разрядностью 8 бит. Для 10-битового АЦП с 1024 шагами напряжения необходимо иметь 1024 компаратора.

Высокая степень интеграции современных микросхем позволяет создать АЦП прямого преобразования с хорошим разрешением. Точность преобразования зависит от точности опорных напряжений и от степени точности компараторов.

1.9.4. АЦП, работающий по весовому принципу

Блок-схема устройства АЦП, использующего метод взвешивания, представлена на рис. 1.32. Компаратор К сравнивает занесенное в устройство выборки и хранения (УВХ) значение измеряемой величины U_c с выходным напряжением ЦАП $U(Z)$.

Рассмотрим этапы работы восьмиразрядного АЦП:

1. Память обнуляется.

2. Записываем в регистр $Z_7 = 1$. На выходе ЦАП получим $U_Z = 2^7 h = \frac{U_{max}}{2}$. Если $U_c > U_Z$, сохраняем $Z_7 = 1$. Если $U_c < U_Z$, обнуляем $Z_7 = 0$.

На этом завершается «взвешивание» старшего разряда. Далее эта процедура выполняется с каждым из разрядов вплоть до самого младшего (LSB).

3. Записываем $Z_6 = 1$. Сравниваем $U_c - (Z_7 2^7 h + Z_6 2^6 h)$. Если $U_c > U_Z$, сохраняем $Z_6 = 1$. Если $U_c < U_Z$, обнуляем $Z_6 = 0$.

После восьми шагов в памяти будет двоичное число $Z = \frac{U_c}{h}$.

Таким образом, в регистре появляется число, которое с помощью ЦАП превращается в напряжение, совпадающее с U_c с точностью до U_{LSB} .

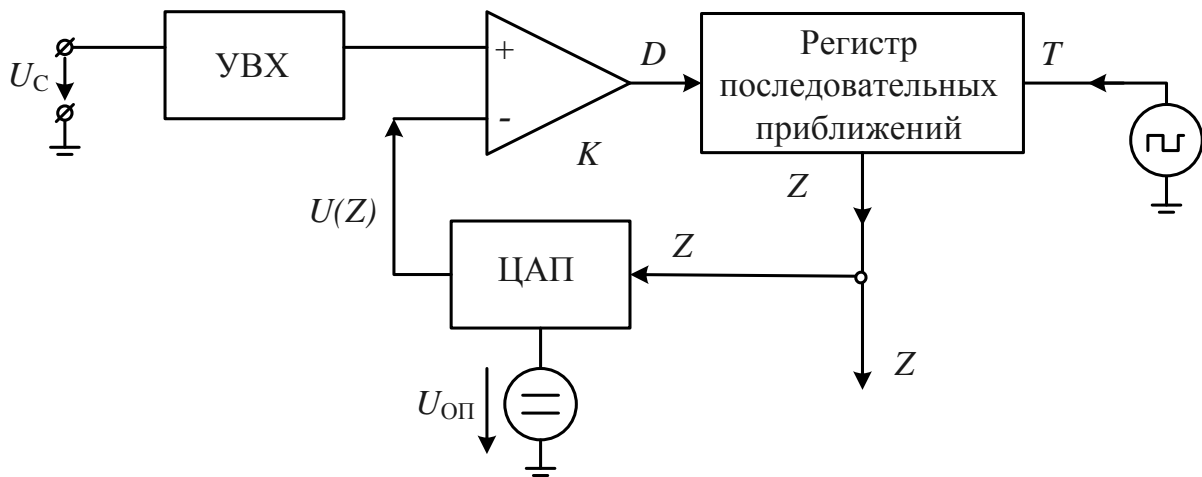


Рис. 1.32. АЦП, работающий по методу взвешивания

Отсюда следует: $U(Z) = U_{on} \frac{Z}{Z_{max} + 1} = U_c$ и

$$Z = (Z_{max} + 1) \frac{U_c}{U_{on}}.$$

Если входное напряжение за время преобразования изменяется, то необходимо УВХ в качестве буфера для хранения считанного значения функции с тем, чтобы все разряды формируемого числа относились к одному и тому же моменту времени. В отсутствие УВХ может возникнуть погрешность, равная приращению входного напряжения за время преобразования.

1.9.5. АЦП, использующие методы счета

Аналого-цифровое преобразование, использующие метод счета, требует наименьших схмотехнических затрат, но длительность преобразования заметно больше по сравнению с остальными методами и, как правило, составляет от 1 мс до 1 с. Этого достаточно для медленно изменяющихся сигналов (например, связанных с измерением температуры, в цифровых вольтметрах и т.п.).

Рассмотрим наиболее важные способы реализации метода счета.

1.9.6. Компенсационный числовой АЦП

Компенсационный АЦП (рис. 1.33) близок к методу взвешивания, отличаясь от последнего тем, что вместо регистра последовательных приближений в нем используется реверсивный счётчик.

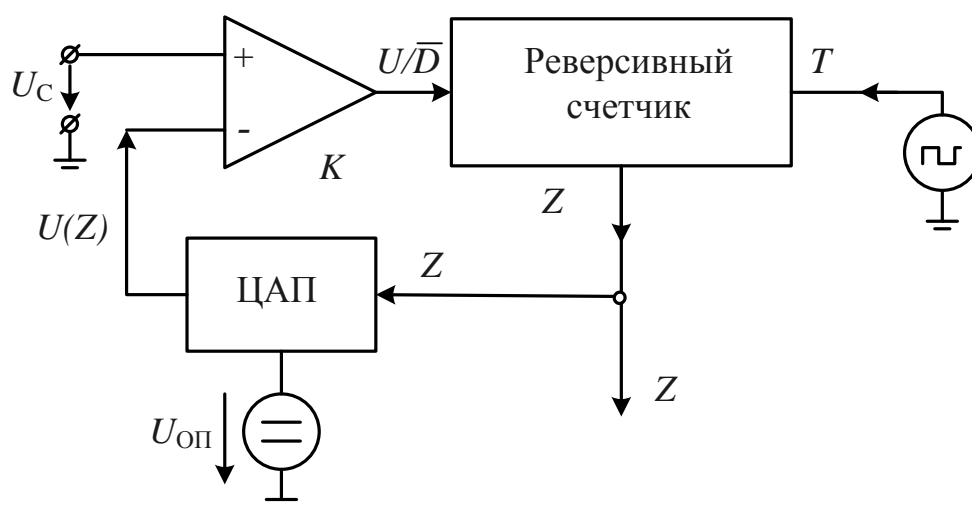


Рис. 1.33. Компенсационный цифровой АЦП

Входное напряжение сравнивается с компенсирующим. Если

$U_c - U(Z) > \frac{h}{2}$, счётчик суммирует импульсы, если

$U_c - U(Z) < \frac{h}{2}$ - вычитает. Компенсирующее напряжение отслеживает

входное. На выходе счетчика формируется двоичный код входного напряжения. Такие АЦП называют следящими аналого-цифровыми преобразователями.

Недостатком этой простой схемы является то, что счетчик не приходит в состояние покоя и его показания всегда изменяются на 1 LSB вокруг входного напряжения, ибо тактовые импульсы следуют непрерывно. Если это мешает, можно расширить обычный компаратор до строб-компаратора, который позволит блокировать тактовые посылки, когда компенсирующее напряжение приближается к входному на $\pm 1/2 \text{ LSB}$.

В этом методе начальная скорость преобразования значительно ниже, чем в методе взвешивания, так как компенсирующее напряжение способно изменяться только на величину U_{LSB} . Однако при медленном изменении

входного напряжения вполне возможно добиться малого времени установления благодаря самой процедуре слежения, при которой аппроксимация происходит непрерывно, тогда как в АЦП взвешивания каждый цикл начинается с нуля.

1.9.7. Метод пилообразного напряжения

АЦП по методу пилообразного напряжения (рис. 1.34) не имеет ЦАП и действует по принципу преобразования входного напряжения в пропорциональный ему временной интервал. Для этого служит генератор пилообразного напряжения в сочетании со строб - компаратором K_1 , K_2 и G_1 .

Пилообразное напряжение изменяется от отрицательного значения до положительного по закону: $V_S = \frac{U_{on}}{\tau} t - V_0$.

На выходе элемента эквивалентности G_1 сохраняется состояние «1» все время, пока пилообразное напряжение находится между 0 и U_c . Это время равно $\Delta t = \tau \frac{U_c}{U_{on}}$ и измеряется числом колебаний кварцевого генератора. Если в начале измерений счётчик сбросить на нуль, по достижении верхнего порога компаратора он покажет число:

$$Z = \frac{\Delta t}{T} = \tau f \frac{U_c}{U_{on}}.$$

Если на вход подать отрицательное напряжение, нулевой отсчёт будет пройден после того, как пилообразное напряжение сравняется с входным. По этой последовательности определяют знак измеряемого напряжения. Длительность измерения останется прежней, так как она зависит только от модуля входного напряжения. По окончании измерения счетчик должен быть сброшен на нуль, а пилообразное напряжение снижено до начальной

отрицательной величины. Чтобы данные на выходе отображались непрерывно, предыдущий числовой результат заносят в запоминающее устройство и хранят там, пока не сформируется новое число.

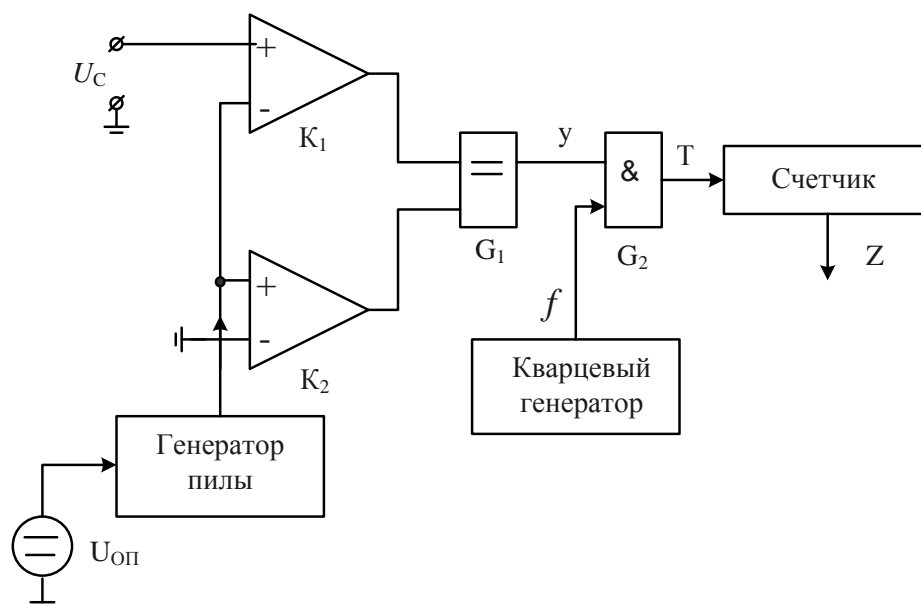


Рис. 1.34. АЦП по методу пилообразного напряжения

Уравнение АЦП показывает, что разброс постоянной времени τ непосредственно сказывается на точности измерений. Постоянная времени определяется RC-звеном, и потому подвержена влиянию температуры и дрейфа ёмкости конденсатора. По этим причинам крайне трудно добиться точности лучше 0,1%.

1.9.8. Метод двойного интегрирования

Метод предусматривает интегрирование не только опорного, но и входного напряжения. В исходном состоянии ключи S_1 и S_2 разомкнуты, а ключ S_3 замкнут (рис. 1.35), благодаря чему напряжение на выходе интегратора равно нулю. Число Z также оказывается равным нулю:

$$Z = (Z_{\max} + 1) \frac{U_c}{U_{on}}.$$

К началу измерений счётчик обнулен, ключ S_3 разомкнут, а S_1 – замкнут, в силу чего происходит интегрирование входного напряжения U_c . Если оно положительно, напряжение на выходе интегратора будет отрицательным, и компаратор K с помощью схемы $И$ разблокирует тактовый генератор. Первая фаза интегрирования t_1 завершается обнулением счетчика вслед за его переполнением после $Z_{\max} + 1$ тактов. После обнуления схема управления изменяет состояние ключей: размыкается ключ S_1 и замыкается S_2 . Затем интегрируется опорное напряжение. Поскольку оно отрицательно, выходное напряжение вновь нарастает. В конце второй фазы интегрирования U_1 достигает нуля. Тогда компаратор сбрасывается на нуль и останавливает счетчик. Его показание равно количеству тактовых импульсов за время t_2 и, следовательно, пропорционально входному напряжению.

Временные диаграммы напряжения на выходе интегратора показаны на рис. 1.36. Видно, что интервал времени t_1 постоянный, а t_2 зависит от входного напряжения.

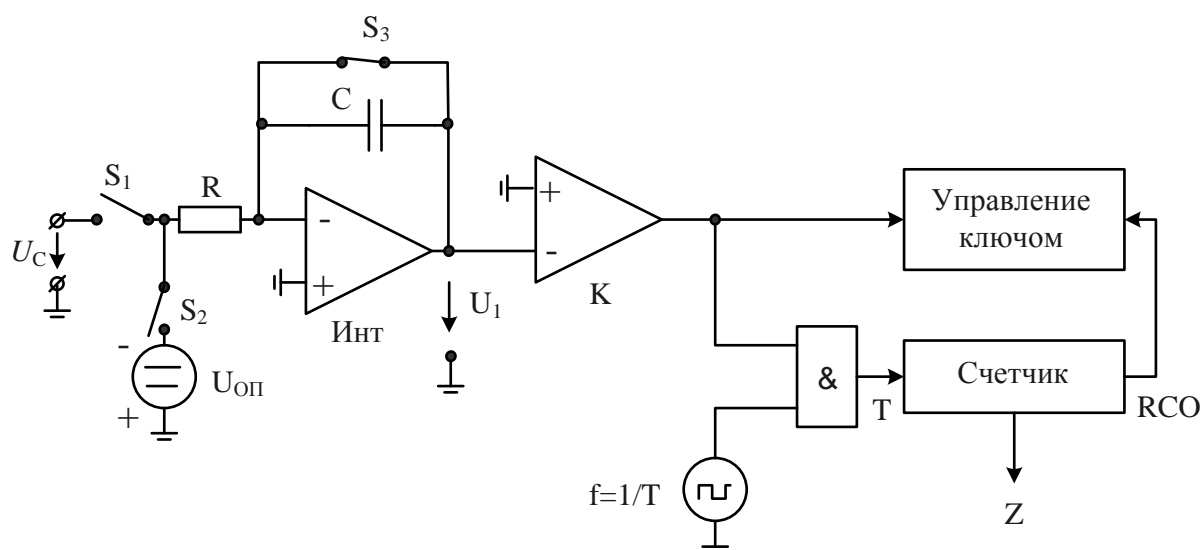


Рис. 1.35. АЦП по методу двойного интегрирования

Зависимость между входным напряжением U_c и результатом Z выражается непосредственно, если рассчитать процесс интегрирования и учесть, что он начинается и завершается при напряжении 0 В. Из уравнения:

$$U_I = -\frac{1}{RC} \int_0^{t_1} U_c dt - \frac{1}{RC} \int_0^{t_2} U_{on} dt$$

при постоянном U_c следует:

$$-\frac{1}{RC} U_c t_1 - \frac{1}{RC} U_{on} t_2 = 0$$

При

$$t_1 = (Z_{max} + 1)T \quad \text{и} \quad t_2 = ZT$$

Получаем:

$$-\frac{1}{RC} U_c (Z_{max} + 1)T - \frac{1}{RC} U_{on} ZT = 0$$

Постоянная времени RC и длительность тактового импульса T в уравнении сокращаются. Тогда:

$$U_c (Z_{max} + 1)T + U_{on} ZT = 0$$

Отсюда окончательно находим:

$$Z = -\frac{U_c}{U_{on}} (Z_{max} + 1)$$

Согласно этому выражению отличительная особенность метода двойного интегрирования заключается в том, что ни тактовая частота $1/T$, ни постоянная

интегрирования $\tau = RC$ не входят в окончательное уравнение. Требуется лишь соблюдать постоянство тактовой частоты за время $t_1 + t_2$. Столь кратковременное постоянство частоты обеспечивается простым тактовым генератором, что позволяет данному методу легко обеспечивать точность 0,01%.

Из окончательного выражения видно, что в него входит не мгновенное

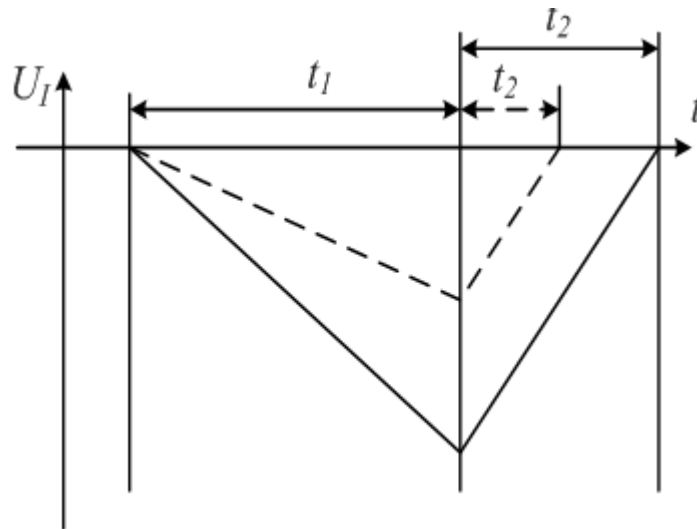


Рис. 1.36. Напряжение на выходе интегратора

значение измеряемого напряжения, а только его уровень, усреднённый за период регистрации t . Отсюда, чем выше частота переменного напряжения, тем сильнее оно ослабляется, и полностью подавляются переменные

напряжения с частотой, кратной целому числу $1/t_1$. Поэтому целесообразно настраивать частоту тактового генератора так, чтобы t_1 равнялось периоду колебаний сетевого напряжения или было ему кратно, что позволит подавить все

Благодаря возможности обеспечить высокую точность и подавление помех при малых затратах метод двойного интегрирования нашел широкое

применение в цифровых вольтметрах, где не важна относительно большая длительность преобразования.

Счётчик на рис. 1.35 может быть и двоично-десятичным. Это обстоятельство используется в цифровых вольтметрах, избавляя от необходимости выполнять двоично-десятичное преобразование результата измерений.

1.9.9. Цифроаналоговое преобразование

Цифроаналоговый преобразователь (ЦАП) преобразует цифровую информацию в аналоговую. В цифровой технике информация существует чаще всего в бинарном виде, который кодируется по определенному коду. Для этого кода должен быть свой цифроаналоговый преобразователь. Код должен быть весовым. Например, каждому элементу двоичного кода, т.е. каждому разряду, поставлен в соответствие вес, или степень числа 2. Двоично-десятичный код также является весовым кодом. Невесовые коды должны быть преобразованы в весовые. Рассмотрим работу нескольких важных типов ЦАП.

1.9.10. ЦАП с суммированием весовых токов

Схема модели ЦАП с суммированием весовых токов показана на рис. 1.37. Опорное напряжение U опор через резисторы с весами 1, 2, 4, 8 и ключи создает на инвертирующем входе ОУ суммарный ток:

$$I_0 + I_1 + I_2 + I_3 = -I_N.$$

При замыкании всех ключей получаем:

$$U_{\text{опорн}} \left(\frac{1}{R_0} + \frac{2}{R_0} + \frac{4}{R_0} + \frac{8}{R_0} \right) = -\frac{U_a}{R_N}$$

В общем случае, когда не все ключи замкнуты, напряжение на выходе равно:

$$U_a = -U_{опорн} \left(\frac{R_N}{R_0} \right) (8Z_3 + 4Z_2 + 2Z_1 + Z_0)$$

Здесь в зависимости от кода $Z_0 \div Z_3 = 0;1$.

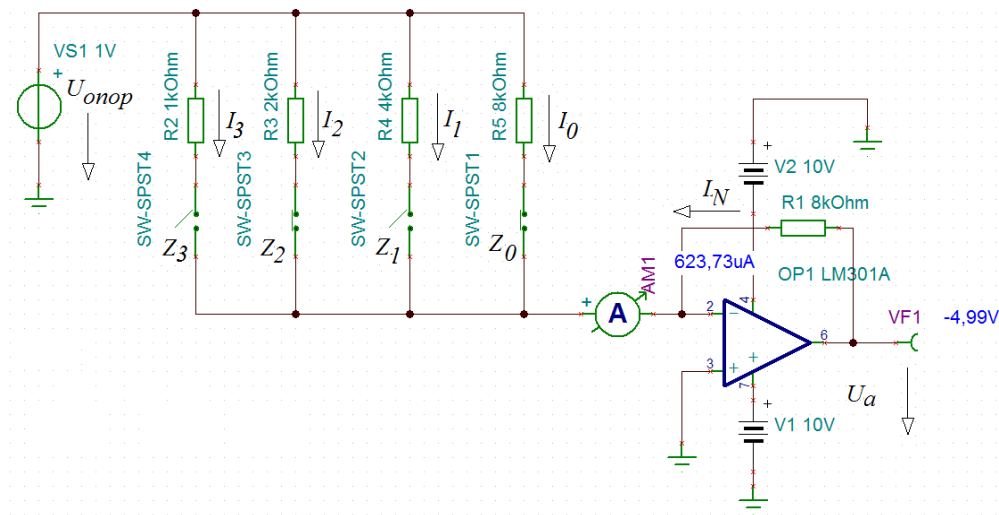


Рис. 1.37. Схема модели ЦАП с суммированием весовых токов

Недостатком схемы является то, что меняется нагрузка опорного источника.

Для устранения этого недостатка применяют перекидные ключи и матрицу постоянного импеданса.

1.9.11. ЦАП с резистивной матрицей постоянного импеданса

Схема модели ЦАП с матрицей постоянного импеданса (матрицей $R-2R$) показана на рис. 1.38.

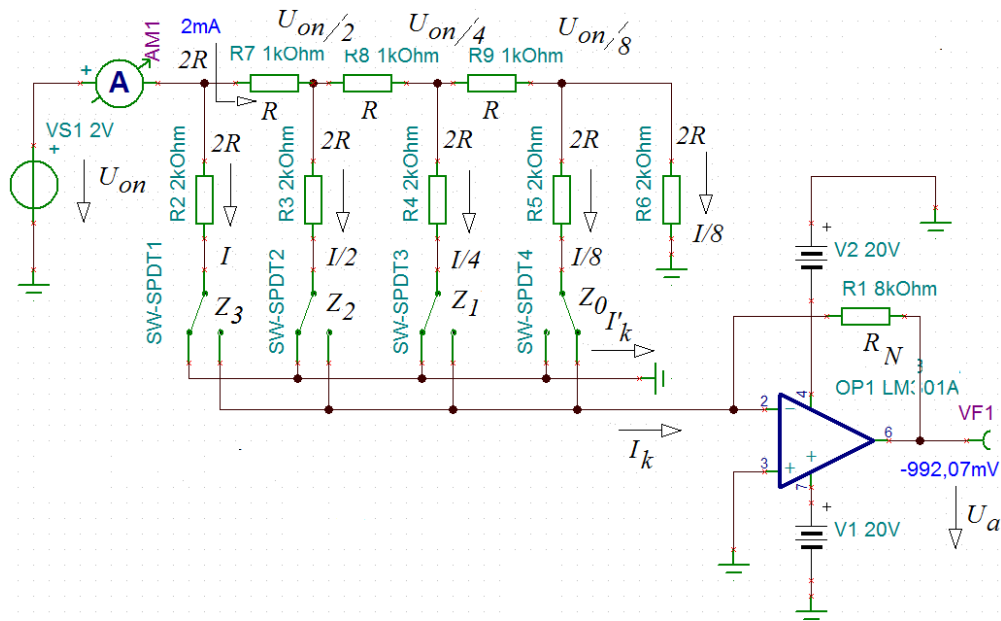


Рис. 1.38. ЦАП с резистивной матрицей постоянного импеданса

Так как неинвертирующий вход ОУ заземлен, на инвертирующем входе потенциал равен виртуальному нулю. При любом положении коммутирующих ключей нагрузка источника опорного напряжения не меняется и равна R .

Вычислим выходное напряжение:

$$I_k = U_{\text{опорн}} \left(\frac{1}{2R} + \frac{1}{4R} + \frac{1}{8R} + \frac{1}{16R} \right) =$$

$$= \frac{U_{\text{опорн}}}{16R} (8Z_3 + 4Z_2 + 2Z_1 + Z_0) = -\frac{U_a}{R_N}.$$

Отсюда получаем:

$$U_a = -U_{\text{опорн}} \frac{R_N}{16R} Z.$$

В реальных ЦАП в качестве ключей используют МОП транзисторы (рис. 1.39).

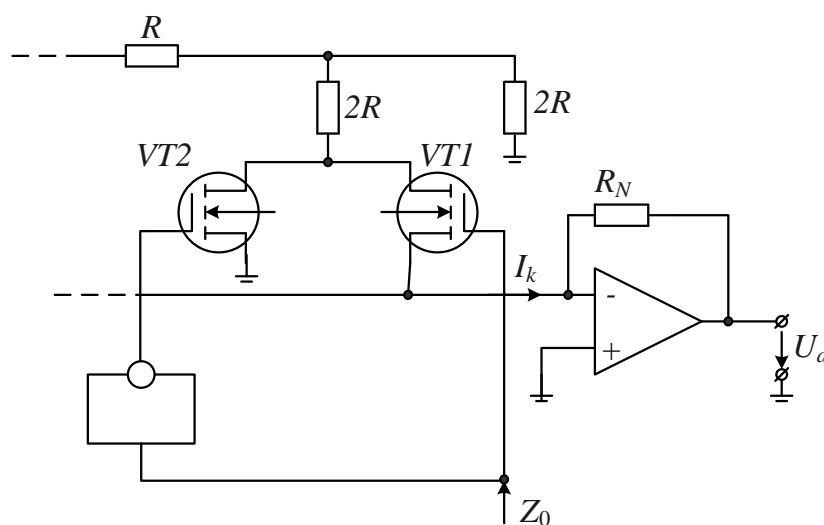


Рис. 1.39. Матрица постоянного импеданса с ключевыми МОП транзисторами

В этой схеме всегда открыт один из транзисторов и нагрузка источника опорного напряжения не меняется.

1.10. Применение дельта – сигма модуляции в АЦП и ЦАП

Модуляция delta-sigma ($\Delta\Sigma$, или sigma-delta $\Sigma\Delta$) это метод кодирования аналоговых сигналов с преобразованием их в цифровую форму, что обычно делают аналого-цифровые преобразователи, АЦП (analog-to-digital converter, ADC). Принцип delta-sigma также используют для преобразования высокоразрядных, низкочастотных цифровых сигналов в высокочастотные, низкоразрядные - как часть преобразования цифровых сигналов в аналоговые, что применяется в цифро-аналоговых преобразователях, ЦАП (digital-to-analog converter, DAC).

Дельта-сигма модуляция — это экономичный метод преобразования аналогового сигнала в цифровой. Несмотря на то, что теория ее была разработана еще в первой половине XX века, только в последнем десятилетии этот метод кодирования аналоговых сигналов начал находить применение на

практике. Все чаще в последнее время встречается термин дельта-сигма модуляция — в описании формата новых SACD аудиодисков, в сверхточных и малозумящих АЦП и ЦАП, в профессиональной звукозаписывающей аппаратуре. Поэтому имеет смысл оценить дальнейший путь развития этого метода аналогово-цифрового преобразования.

1.10.1. Краткий исторический обзор.

В 1939 году Джоном Ривзом из лаборатории Александра Г. Бэлла был изобретен способ преобразования и передачи аналоговых телефонных сигналов в виде дискретных импульсов, названный впоследствии импульсно-кодовой модуляцией (ИКМ). Позднее появилась идея передавать не абсолютное значение сигнала в каждый момент времени, а лишь его изменение относительно предыдущего значения (дифференциальная ИКМ — ДИКМ). Но ДИКМ была все еще многобитной системой. Следующим шагом по направлению к дельта-сигма модуляции стала разработка принципов дельта-модуляции, где изменение сигнала передается всего лишь 1 битом информации. С него и имеет смысл начать.

1.10.2. Дельта-модуляция

Рассмотрим блок-схему дельта-модулятора, изображенную на рис. 1. Принцип его действия можно описать следующим образом: на основании некоторого набора предыдущих выборок сигнала делается предположение о последующей. Затем предполагаемое значение сравнивается с фактическим и выносится решение о знаке их различия, что и является выходным сигналом.

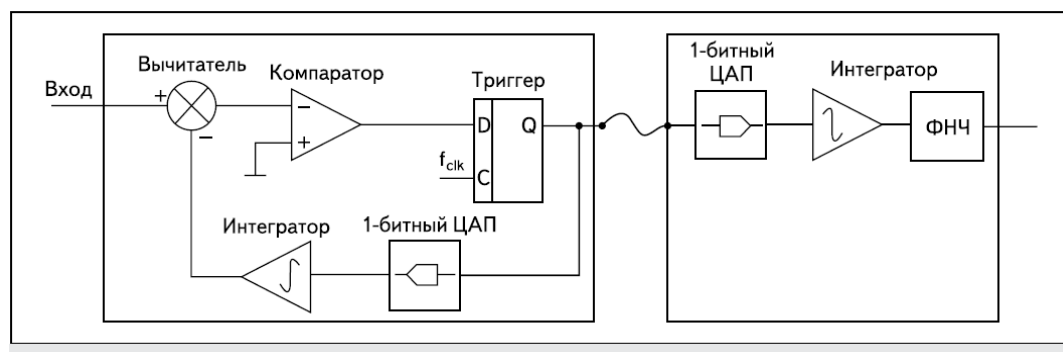


Рис. 1.40. Блок-схема дельта-модулятора: ФНЧ — фильтр низких частот

Напряжение входного сигнала подается на вычитатель, где из него вычитается аппроксимирующее напряжение, созданное на основании предыдущих значений сигнала.

Далее разность поступает на стробируемый компаратор, где сравнивается с нулевым уровнем. Таким образом, логическая единица на выходе компаратора означает, что эта разность положительна или что входной сигнал больше предполагаемого (аппроксимирующего), а логический ноль, соответственно, означает, что входной сигнал меньше аппроксимирующего. Далее последовательность нулей и единиц поступает на однобитный местный ЦАП, который обычно представляет из себя преобразователь уровней однополярного напряжения (лог. «0» и лог. «1») в двухполярное ($\pm U_{\text{пит}}$). С выхода ЦАП сигнал поступает на вход интегратора, на выходе которого формируется аппроксимирующее напряжение, с заданной точностью повторяющее входное. Точность определяется частотой стробирования компаратора и шагом приращения напряжения в интеграторе. Схема приемной части состоит из однобитного ЦАП, интегратора и ФНЧ.

Эта схема имеет ряд существенных недостатков, которые препятствовали ее применению в аппаратуре аналогово-цифрового преобразования. Попытки ее модернизации привели к переходу от дельта-модуляции к дельта-сигма модуляции.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

1.10.3. Дельта-сигма модуляция

Дельта-сигма модуляция обладает всеми достоинствами дельта-модуляции и в то же время лишена многих её недостатков. Для того чтобы разобраться в ее структуре и понять, как был выполнен переход от схемы дельта-модулятора к схеме дельта-сигма модулятора (ДСМ), можно рассуждать следующим образом. Как известно, дельта - модулятор пригоден для работы только с хорошо коррелированными сигналами, поэтому для повышения коррелированности входного сигнала его можно пропустить через интегратор, а на приёмной стороне выходной преобразованный сигнал пропустить, соответственно, через дифференциатор (рис. 1.41).

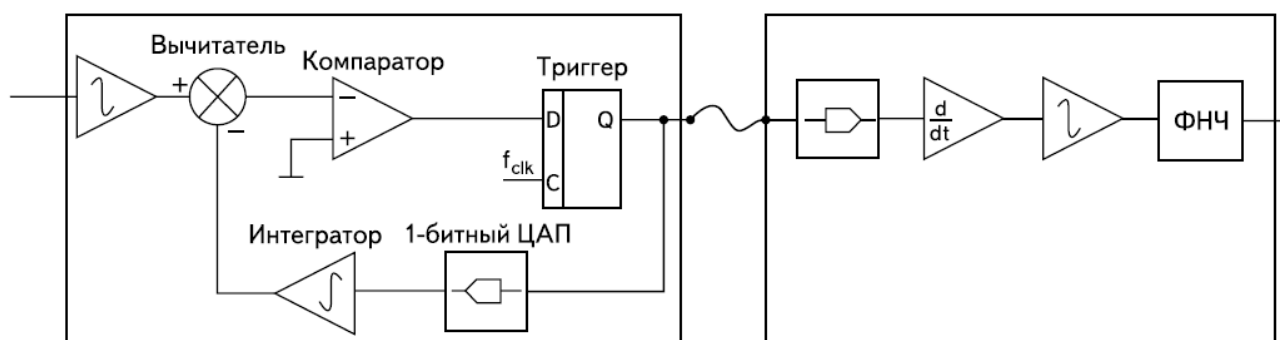


Рис. 1.41. Переход от дельта-модулятора к дельта-сигма модулятору

Поскольку разность интегралов равна интегралу разности, то два интегратора на входах вычитателя можно заменить одним на его выходе. Что касается дифференциатора на приемной стороне, то он вместе с приемным интегратором может быть исключен. Таким образом, схема ДСМ, изображенная на рис. 3, отличается от дельта-модулятора положением интегратора на передающей стороне и его отсутствием на приемной. Такое незначительное изменение в схеме значительно улучшило ее характеристики и, в частности, позволило достичь отношения сигнал/шум -120 дБ.

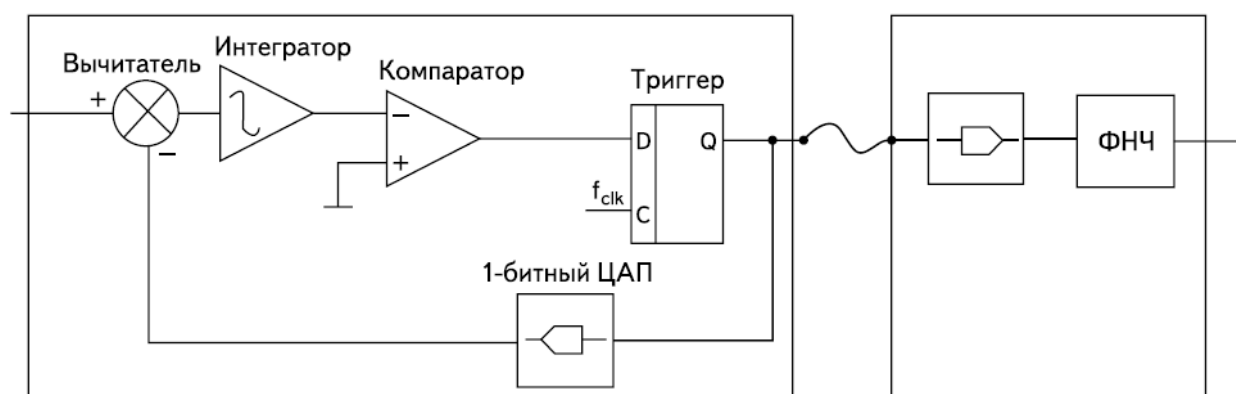


Рис. 1.42. Схема дельта-сигма модулятора

Рассмотрим работу схемы ДСМ. Когда образуется высокоррелированный сигнал, то коррелированными оказываются не только его отсчеты, но и ошибки при каждом квантовании. Следовательно, их легко предсказать и вычесть из сигнала, отправляемого на устройство квантования, прежде чем произойдет квантование. Хорошей оценкой текущей ошибки в таком случае выступает предшествующая ошибка. Предшествующая ошибка, образованная как разность между входом и выходом устройства квантования, помещается в схему задержки (триггер). Таким образом, в контуре обратной связи циркулирует сигнал ошибки.

Выходной сигнал ДСМ представляет собой однобитный поток импульсов. Рассмотрим его в терминах теории вероятности. Так, вероятность появления в потоке логической единицы $P(1)$ и вероятность появления логического нуля $P(0)$ связаны следующим выражением: $P(0) + P(1) = 1$. Более того, если на вход модулятора подается сигнал x (ограниченный в динамическом диапазоне 0-1), то вероятность $P(1) = x$, а $P(0) = (1-x)$. Иными словами, чем плотнее представлены импульсы определённой полярности в потоке, тем выше уровень сигнала в этот момент. Нулевой уровень сигнала кодируется одинаковой плотностью положительных и отрицательных

импульсов. Импульсные последовательности при кодировании синусоидального напряжения представлены на рис. 1.43. Видно, что плотность положительных и отрицательных импульсов одинакова в точках, близких к 0, плотность отрицательных импульсов максимальна в точке -1, и плотность положительных импульсов максимальна в точке +1.

Такие особенности позволяют кодировать в формате дельта-сигма модуляции сигналы с частотой от 0 до 100 кГц. В частности, прямоугольное аналоговое напряжение и уровни постоянного напряжения, последнее актуально при применении дельта-сигма модуляции в датчиках медленно меняющихся сигналов.

В следующих главах мы исследуем модели дельта-сигма АЦП и ЦАП.

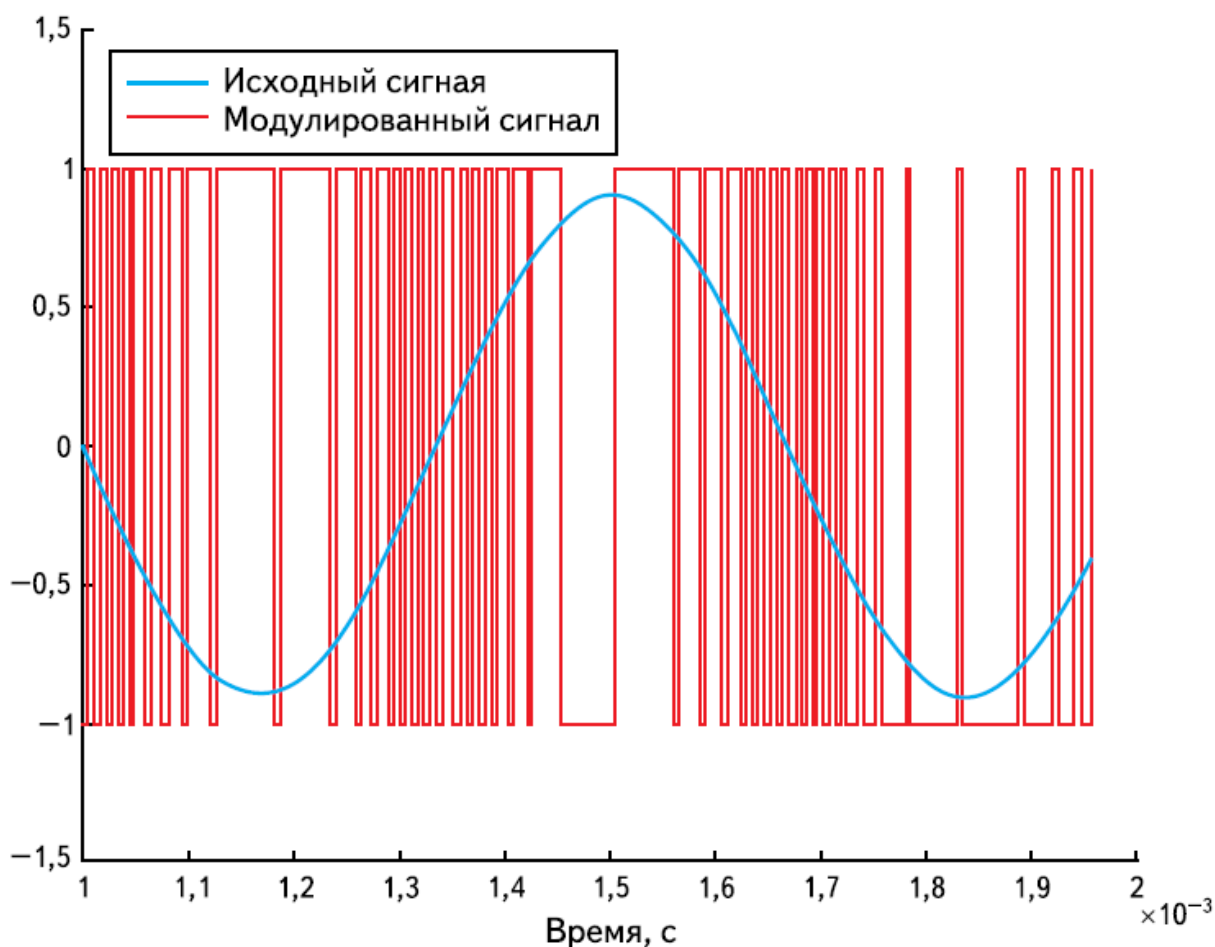


Рис. 1.43. Осциллограмма выходного сигнала дельта-сигма модулятора

1.10.4. Шумы дельта-сигма модуляции

Исследование шумов в ДСМ заслуживает отдельного рассмотрения. Ведь методы достижения отношения сигнал/шум -120 дБ при разрядности 1 бит представляют известный интерес. В 1954 году С. Катлер из той же лаборатории Александра Бэлла предложил концепцию передискретизации и формирования спектра шума. Как известно, каждый дополнительный бит при преобразовании аналогового сигнала в цифровой добавляет 6 дБ к отношению сигнал/шум (рис. 1.43а). Одним из основополагающих принципов дельта-модуляции является превышение частоты Котельникова в K раз. При такой передискретизации эффективная разрядность, а соответственно, и отношение сигнал/шум, увеличивается согласно формуле $K = 2^N$, где K — коэффициент передискретизации, а N — количество дополнительных битов. Обычно применяется $K = 64$, и в этом случае эффективная разрядность будет 7 бит, а отношение сигнал/шум будет равно 42 дБ (рис. 1.43б). Однако передискретизация сама по себе не является эффективным средством. Дальнейшее подавление шума производится благодаря самой структуре дельта-сигма модулятора. В иностранной литературе часто применяется термин «нойзшейпинг», что означает формирование спектра шума. Чтобы понять, как именно происходит формирование, используем линеаризованную дискретную модель системы, в которой входной сигнал представлен последовательностью $j(p)$, выходной сигнал $y(x)$ и шум квантования, вносимый компаратором и триггером, — $e(p)$, что изображено на рис. 1.44.

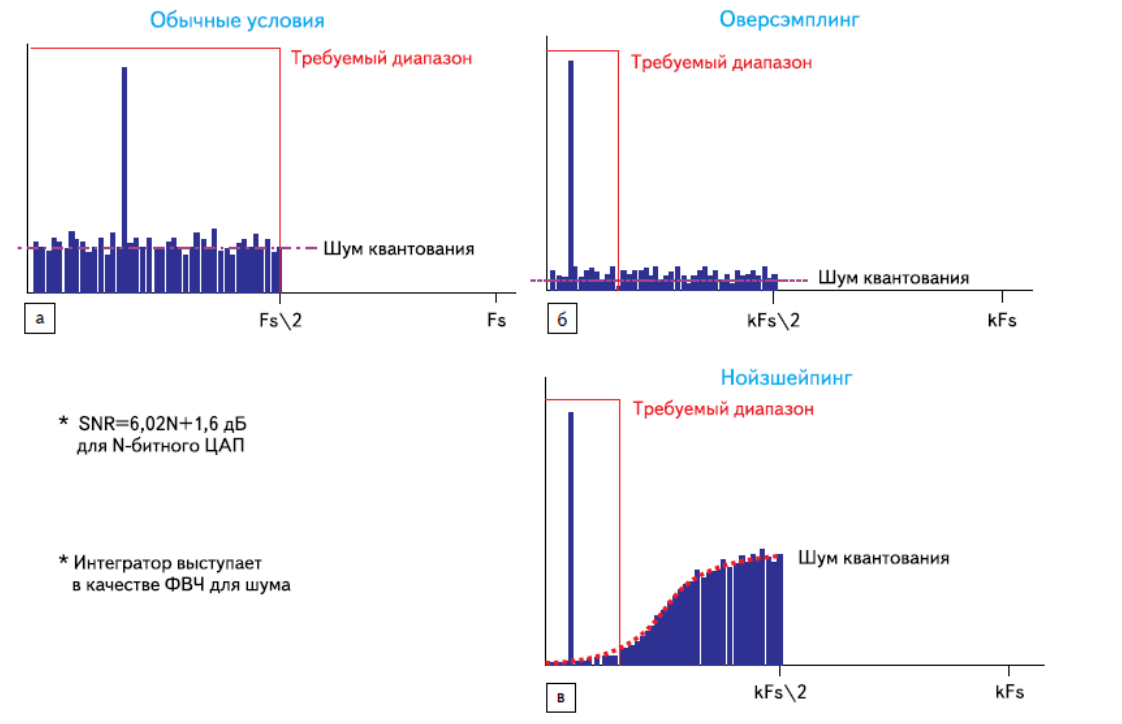


Рис. 1.43. Спектры выходного сигнала

Рассмотрим Z-преобразование этой системы дельта-сигма модулятора:

$$Y(z) = \frac{X(z) - Y(z)}{1 - \frac{1}{z}} + E(z),$$

$$Y(z) = \frac{X(z)}{z} + \left(1 - \frac{1}{z}\right)E(z).$$

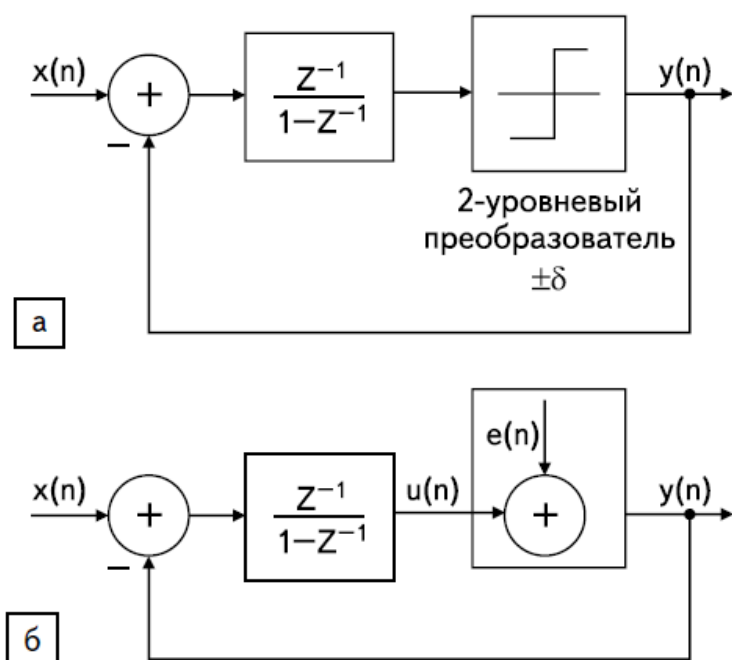


Рис. 1.44. Схема линейризованной дискретной модели системы

Видно, что полезный сигнал $X(t)$ проходит эту цепь без изменений, с задержкой на 1 такт, в то время как для шума возникает препятствие в виде ФНЧ. Таким образом, осуществляется формирование спектра шума в дельта-сигма модуляторе. Интегратор в данном случае выступает в качестве ФНЧ для шумовой составляющей сигнала.

Энергия шума сосредотачивается в области верхних частот, и большая ее часть может быть отфильтрована выходным ФНЧ (рис. 1.44в). Таким образом, в выходном сигнале после демодулирования дельта-сигма последовательности наблюдается намного более низкий уровень шума, чем можно было бы предполагать. Следующим шагом по улучшению параметров по шумам является повышение порядка модулятора. Следует особо отметить, что дельта-сигма АЦП с высочайшей (24 бита) эффективной разрядностью можно построить, всего лишь используя интегратор и стробируемый компаратор.

1.10.5. Информационные параметры

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Еще одним важным на сегодня параметром сигнала является его информационная емкость. Здесь следует отметить, что сигнал в формате дельта-сигма модуляции не требует кадровой синхронизации, а значит, считывать его можно в любой момент времени в записи или в канале передачи. В этом его сходство с аналоговым сигналом. Еще одно важное его отличие — это факт одинаковой информационной емкости каждого бита в потоке, что повышает помехоустойчивость сигнала в формате дельта-сигма модуляции.

Оценим теперь информационные параметры сигнала. В качестве требуемого диапазона возьмем порог слышимости, принятый в различных стандартах звукозаписи, — 22 кГц. Частота Котельникова в ИКМ для такого диапазона, следовательно, будет равняться 44 кГц. Оверсэмплинг в формате дельта-сигма модуляции (например, SACD фирмы Sony) предполагает 64-кратное увеличение частоты Котельникова. Таким образом, получается, что частота дискретизации в формате дельта-сигма модуляции с оверсэмплингом будет равна 2,82 МГц для передачи диапазона от 0 до 22 кГц. Учитывая, что передача цифровых сигналов в обоих форматах ведется в последовательном режиме, оценим количество бит в секунду. При последовательной передаче в формате ИКМ 44 кГц/16 бит поток равен 705 кБод, в формате дельта-модуляции — 2,8 мБод.

Однако качество сигнала в формате дельта-модуляции 2,8 МГц приближается к качеству сигнала в формате ИКМ — 192 кГц/24 бита, поток которого составляет уже 4,8 мБод. Также следует учесть, что, в отличие от дельта-сигма модуляции, при передаче ИКМ-сигналов требуется жесткая кадровая синхронизация.

1.10.6. Применение

В настоящее время дельта-сигма модуляторы широко применяются в системах аналогово-цифрового и цифро-аналогового преобразования благодаря чрезвычайно простой архитектуре при высокой разрядности (как было сказано выше) и высокому отношению сигнал/шум. И хотя такие системы пользуются сейчас заслуженной популярностью, сама по себе дельта-сигма модуляция как метод передачи и хранения информации остается пока лишь вспомогательной, второстепенной технологией.

Пожалуй, единственным примером полноценного использования дельта-сигма модуляции в области записи данных является разработка фирмы Sony под названием Super Audio Compact Disk (SACD), призванная заменить популярную, но уже устаревающую технологию Audio Compact Disk (Audio CD). Не вдаваясь в особенности этой технологии, скажем лишь, что информация на таком диске записана в формате дельта-сигма модуляции, что, по оценкам некоторых специалистов, обеспечивает более реалистичное звучание, чем при обычной, даже сверхвысококачественной записи в формате ИКМ. На рис. 8 приведены спектры шумов в различных форматах на выходе профессиональной звуковоспроизводящей аппаратуры. Нетрудно заметить, как энергия шума сосредоточена преимущественно в области частот более 20 кГц.

В настоящее время системы вида «АЦП-обработка, передача или хранение сигнала ЦАП» строятся так, как показано на рис. 1.45. Структуры «кодер – ЦСП – декодер»: ДСДМ — дельта-сигма демодулятор; ЦСП — цифровой сигнальный процессор.

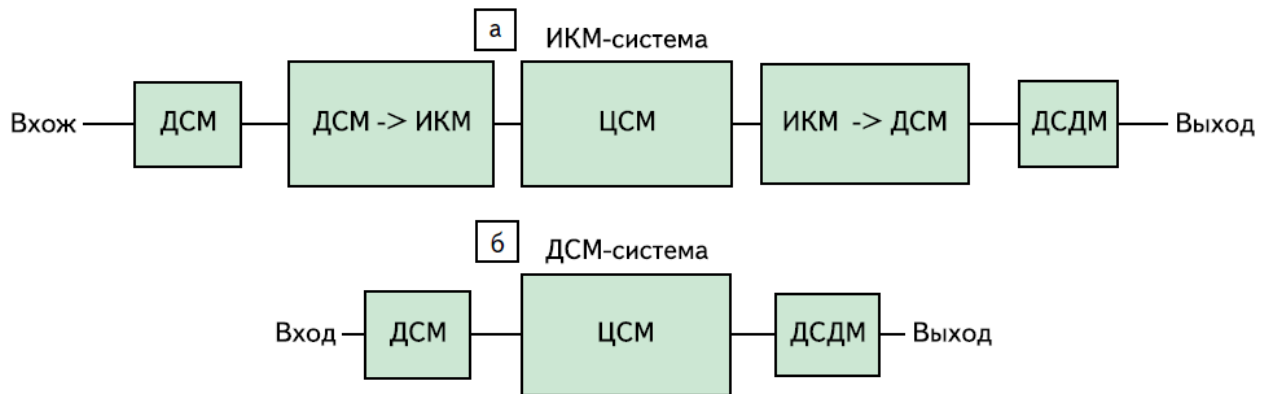


Рис. 1.45. Структуры «кодер – ЦСП – декодер»: ДСДМ — дельта-сигма демодулятор; ЦСП — цифровой сигнальный процессор

Сначала аналоговый сигнал поступает на дельта-сигма модулятор, преобразовывается в 1-битный цифровой поток, поступает на фильтр-преобразователь, который производит одновременно и цифровую фильтрацию высокочастотной шумовой составляющей сигнала, и преобразование в многоразрядный ИКМ-код. Обычно он строится по схеме, в англоязычной литературе именуемой *Integrate and Dump* (или, в переводе на русский, схема накопления и сброса), выполненной на двоичном счетчике и регистре.

Далее многоразрядный ИКМ-код подвергается любым математическим операциям, передается по линии связи или сохраняется в памяти. При обратном преобразовании полученные отсчеты сигнала интерполируются и снова преобразовываются в битовый поток с помощью сложного цифрового ДСМ. С его выхода сигнал поступает на демодулятор, который представляет собой фильтр нижних частот.

С появлением прецизионных и дешевых дельта-сигма АЦП и ЦАП возникает возможность избавиться от лишних ступеней преобразования форматов и перейти к новой схеме обработки и передачи сигналов уже без формата ИКМ. На рис. 1.45б изображена такая схема. Поток с выхода дельта-

сигма модулятора поступает непосредственно на вход специального сигнального процессора (или ПЛИС) и на приемной стороне дельта-сигма ЦАП.

Однако отметим, какие препятствия стоят на пути подобного упрощения системы, и рассмотрим их по порядку. Во-первых, пока не существует отработанных методов цифровой обработки дельта-сигма потоков для многих важных операций. Во-вторых, существующие методы не позволяют поддерживать качество обработанных с их помощью сигналов на приемлемом уровне. Иными словами, при выполнении каждой операции прямой обработки дельта-сигма сигналов существенно возрастает уровень шумов в сигнале.

При реализации в ПЛИС схемы для обработки дельта-сигма потока требуют значительно меньше аппаратных ресурсов.

Уже на простейших операциях видна значительная экономия ресурсов ПЛИС. Используя умножители, сумматоры и цепи D-триггеров в качестве элементов задержки, можно построить более сложные устройства обработки сигналов в формате дельта-сигма модуляции, которые требуют намного меньше ресурсов ПЛИС. Здесь необходимо учесть, что при увеличении числа операций над дельта-сигма сигналом из-за ошибок пропорционально падает и отношение сигнал/шум.

Следовательно, реализация сложных многоступенчатых устройств обработки сигналов становится затруднительной. Это обстоятельство и является главным препятствием, которое необходимо преодолеть исследователям, чтобы формат дельта-сигма модуляции смог стать уверенным конкурентом для ИКМ в области обработки, хранения и передачи аналоговых сигналов.

Глава 2. Современное проектирование информационно-коммуникационных систем

2.1. Современное проектирование ИКС

На сегодняшний день подавляющее большинство ИКС являются распределёнными системами реального времени. Особое место занимают ИКС для мобильных устройств, которые относятся к встраиваемым системам (ВсС) и могут проектироваться по технологии «Система на кристалле» (СнК, SoC).

Выделяются три сценария проектирования ИКС:

- а) заказной или целевой, когда создается специализированная ИКС «под ключ»;
- б) создание вычислительной (проектной) платформы для целевых ИКС;
- в) модификация существующих систем или платформ.

Основными являются первые два сценария. В качестве иллюстрации сложности и важности сценариев можно отметить бурное развитие методологии PBD (платформно - ориентированное проектирование), первоначально явно направленной на создание СнК.

Для разработчика ИКС очень важным является широкое видение и обоснованный выбор вариантов решения задачи. Проектная платформа, как решение архитектурного уровня, в традиционных технологиях создания ИКС действительно характеризует практически весь маршрут проектирования.

На основе этого понятия возможна классификация архитектурных решений для ИКС. Перечень широко используемых в настоящее время и перспективных категорий архитектурных платформ представлен ниже:

- платформы промышленных ПК;
- платформы программируемых логических контроллеров;

- полуфабрикаты от мультимедиа-индустрии (КПК и сотовые телефоны с мобильными операционными системами, системы прикладного программирования в стиле micro-DotNet и т.д.);
- микроконтроллерные модули со средствами программирования (на «железе», RTOS, виртуальные машины и т.д.);
- DSP-платформы; - ПЛИС, ПСнК;
- ASIC, ASSP, SOC и т.д.;
- «свободная» кремниевая компиляция.

Платформы промышленных персональных компьютеров и программируемых логических контроллеров позволяют относительно просто и быстро создать прикладную систему, однако эта эффективность проявляется только в рамках «стандартных» технических заданий.

Проектные платформы микроконтроллеров и сигнальных процессоров предоставляют большую свободу разработчику. Они имеют свою специфику, в первую очередь в организации системного программного обеспечения, в степени открытости архитектуры и т.д.

В качестве платформ с успехом используются ПЛИС и программируемые СнК, сочетающие гибкость программных и аппаратных средств.

Нельзя упускать из виду и предельный вариант, когда в качестве платформы выступает весь электронный логический базис в рамках принципа свободной кремниевой компиляции. Последний вариант — в некотором роде из области фантастики, а перечисленные выше — наши реалии в проектировании.

2.2. Высокоуровневое проектирование встраиваемых систем

Ранее проектирование цифровых устройств начиналось с размещения транзисторов для реализации нужной функции (рис. 2.1). Очевидно, что при таком ручном методе проектирования гибкость заключается в выборе размеров

транзисторов и способе проведения проводных соединений, однако при этом достигается оптимальная реализация заданной функции.

По мере усложнения проектов возникла необходимость в еще более высоком уровне абстракции, при котором количество элементов уменьшилось бы, по сравнению с числом вентилях. Основное внимание на этом уровне абстракции сосредоточено на передаче данных между регистрами, логическими узлами и шинами. Поэтому он и называется уровнем регистровых передач (RTL).

С течением времени проекты ещё более усложнились, и возникла необходимость в переходе на более высокий, по сравнению с регистровыми передачами, уровень абстракции. В настоящее время этот уровень, называемый уровнем электронной системы (Electronic System Level – ESL), или системным уровнем, является наивысшим. На системном уровне разработчик заботится только о функционировании разрабатываемой системы и описывает алгоритм, который должен быть реализован. Алгоритм описывается с помощью процедурного языка, подобного языку программирования C. Описание системы на этом уровне не содержит синхросигналов или временных задержек вентильного уровня.

Средства проектирования системного уровня включают средства ввода, моделирования и, конечно, программы генерации аппаратной части. Генерация аппаратуры из описания системного уровня может выполняться одним из двух возможных способов. Первый способ аналогичен применяемому на других уровнях абстрагирования и заключается в трансляции описания системного уровня на низший уровень абстракции, то есть на уровень регистровых передач.

Альтернативным способом является путь, когда процедурное описание системного уровня может компилироваться для выполнения на заданном

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

процессоре. Этот метод становится возможным только на системном уровне, потому что описание системного уровня является процедурным, и для этого используется язык описания программной части системы, подобный языку С.

Именно последний вышеупомянутый метод генерации аппаратуры из описания системного уровня и должен стать методом проектирования встраиваемых систем.

На сегодняшний день в цикле работ по созданию ВсС все большее значение приобретает этап высокоуровневого проектирования (High Level Design, HLD). На рис. 2.2 показаны подуровни системного уровня представления/проектирования вычислительных систем традиционной Y-диаграммы. Системный уровень, который представляется обычно единым верхним уровнем, в настоящее время разделяется на три части:

- спецификация;
- архитектура (макроархитектура);
- микроархитектура.



Рис. 1.1. Уровни абстракции описания цифровой системы

Рис. 2.1. Уровни абстракции описания цифровой системы



Рис. 2.2. Y – модель проектирования вычислительных систем

Две верхние оси отображают поведенческий и структурный аспекты представления системы, нижняя – её физическую реализацию. С увеличением расстояния от начала оси увеличивается уровень абстракции. Фактически, крайние точки, изображённые на рис. 2.2, и представляют собой системный уровень представления ВС.

Процесс проектирования, отображённый на Y-модель при помощи перемещений между осями, представлен на Рис. 2.3. Стрелки слева направо соответствуют задачам синтеза на различных уровнях абстракции, стрелки справа налево – представляют собой анализ полученных решений. Далее, стрелки сверху вниз – генерация из описаний аппаратуры конкретных реализаций, стрелки снизу-вверх – извлечение описаний из существующих реализаций. стрелки вдоль осей показывают увеличение уровня абстракции или детализацию, закруглённая стрелка – символизирует оптимизацию полученного структурного представления без изменения уровня абстракции.

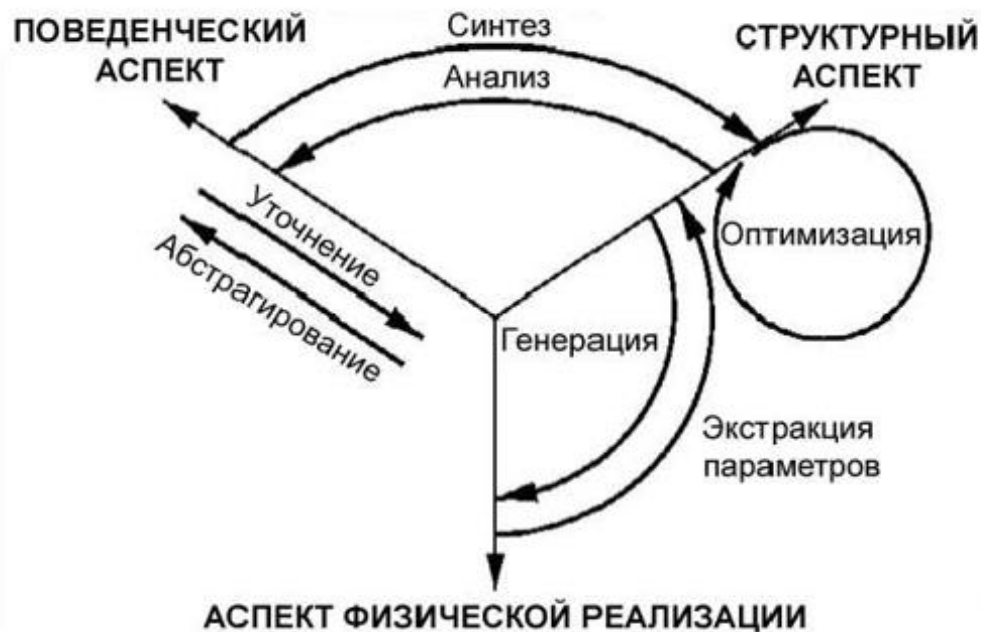


Рис. 2.3. Процесс проектирования по Y – модели

В Y-модели не представлен временной аспект, аспект коммуникаций между объектами и абстракции данных. Кроме того, хотя такое представление можно расширить и на представление ПО, но в явном виде оно в модели не представлено.

SLD (System Level Design) — это работа с абстрактными описаниями некоторой ВС, предполагающая их создание, анализ, модификацию и дальнейшее от того, как они в дальнейшем будут реализованы.

Уровень микроархитектуры должен содержать информацию о концептуальном устройстве отдельных вычислителей и телекоммуникационных устройств, входящих в состав ИКС, независимо от способа реализации вычислителя (например, в виде кремниевого процессора, виртуальной машины, аналого-цифрового смешанного устройства).

Перечислим основные задачи этапа высокоуровневого проектирования ИКС:

- концепция решения целевой задачи, исходные спецификации;

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

- организация вычислительного процесса (модели вычислений — MoC);
- организация инфраструктуры проекта;
- архитектура ИКС, ее оценка и верификация;
- выходные спецификации для этапа реализации.

Важную роль в данном отношении играет понятие модели вычислений (Model of Computation), которое позволяет создавать абстрактное представление ИКС и выполнять комплексное и поэлементное моделирование.

2.3. Платформы проектирования в информационно-коммуникационной технике

Многообразие платформ, применяемых в информационно - коммуникационной технике, представлено на рис. 2.4. Под проектной платформой понимается уровень абстракции в маршруте проектирования, который скрывает несущественные на данном этапе детали множества возможных способов реализации.

Процесс проектирования на основе платформ, называемый в литературе платформно-ориентированным проектированием (platform-based design, PBD), заключается в поэтапном проектирование вычислительных платформ, уточнении высокоуровневой начальной спецификации до готового продукта с использованием платформы на каждом этапе проектирования. Это позволяет повысить эффективность проектирования за счет абстрагирования и повторного использования. Анализ работы показывает, что почти каждая абстракция (платформа), применяемая для преодоления сложности в проектировании вычислительной техники, скрывает важные для ИКС характеристики физического мира.

Основу проектирования составляет вопрос организации целевого вычислительного процесса аппаратными и программными средствами.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Многообразие вариантов реализации вытекает из многоуровневой организации ИКС— своеобразного «слоеного пирога».

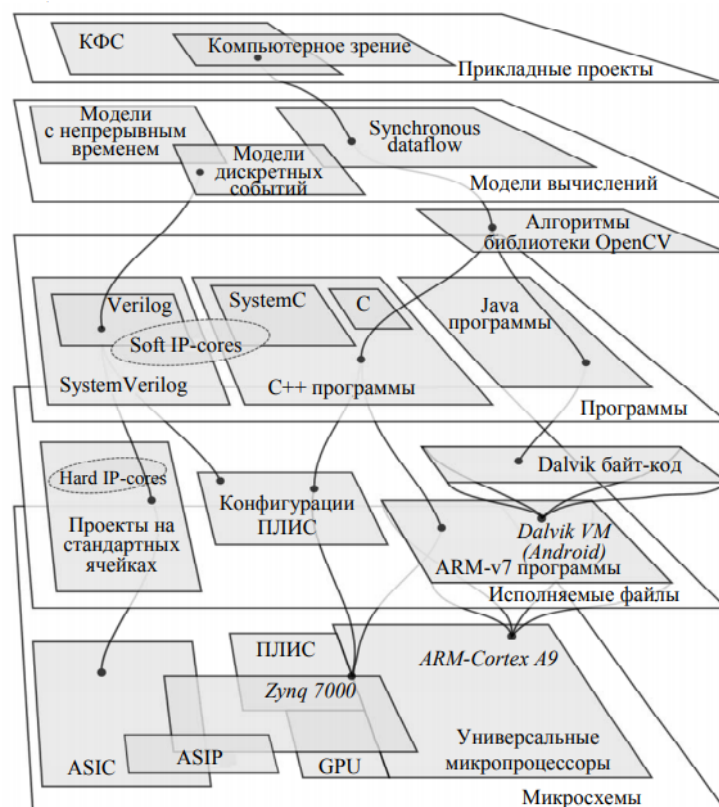


Рис. 2.4. Платформы информационно – коммуникационной техники

Одни и те же функции потенциально могут быть реализованы на любом уровне «пирога». На практике же выбор уровня должен быть компромиссом между стоимостью и получаемыми характеристиками. Этот выбор является одной из центральных проблем проектирования ИКС. На практике он обычно сводится к вопросу распределения функциональности между программной и аппаратной составляющими.

Анализ уровней «погружения» в вычислительную организацию ВС в рамках проекта ИКС показывает, что на практике выполняются работы на всех уровнях, причем в различных сочетаниях. Ниже перечислены основные категории таких проектных работ: выбор или разработка прикладного ПО; выбор или разработка системного ПО; определение состава

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

телекоммуникационных модулей (ТКМ) из готового набора; разработка ТКМ; определение состава центральных вычислительных ресурсов ВС (процессоров, памяти, интерфейсов); разработка центральных вычислительных ресурсов ВС на готовой компонентной базе; разработка компонентной базы. Проектная платформа, как решение архитектурного уровня, в традиционных технологиях создания ИКС определяет практически весь маршрут проектирования и разработки. Поэтому именно она может выступать в качестве классификационного признака технологий создания ИКС и определять уровень «погружения» в вычислительную иерархию, ожидающий разработчика.

В качестве платформ с успехом используются ПЛИС, сочетающие гибкость программных и аппаратных средств. Существуют проекты, включающие создание специальной компонентной базы, в первую очередь SoC, ASIP, ASIC. Особое положение занимают платформы мобильных и интернет-устройств, которые начинают активно использоваться в качестве мобильных терминалов ИКС и сетевые контроллерные платформы, которые выступают системообразующими решениями в ИКС с распределенной организацией.

2.4. Проектирование электронных устройств в САПР Cadence

В настоящее время Cadence предоставляет широкий набор программных средств для проектирования современных электронных устройств от интегральных схем и ПЛИС до систем на кристалле (СНК) и Интернета вещей.

На рис. 2.5 показаны этапы проектирования систем на кристалле, включающие:

Системное проектирование;

Аппаратное проектирование;

Проектирование топологии интегральной схемы (ИС);

Проектирование корпуса ИС;



2.6. Маршрут проектирования Cadence

Важно отметить, что маршрут проектирования Cadence реализует все этапы проектирования, начиная с системного и заканчивая разработкой печатной платы. Это, так называемое, сквозное проектирование электронных систем, которое выполняется с использованием систем автоматизированного проектирования (САПР) электроники или EDA (Electronic Design Automation).

В следующих главах мы будем изучать такие САПР:

TINA-12 компании DesignSoft;

OrCAD 17.2 компании Cadence.

Для проектирования цифровых устройств ИКС на базе ПЛИС применяют среду Quartus II, компании Intel, среду Simulink компании Xilinx и пр.

Глава 3. Анализ и проектирование цифровых устройств в среде TINA

Введение

Современные электронные системы используют широкий диапазон разнообразных цифровых устройств:

простые цифровые интегральные схемы;

интегральные схемы памяти;

программируемые логические интегральные схемы (ПЛИС, ПЛИМ, FPGA (field-programmable gate array), и др.);

микроконтроллеры;

микропроцессоры;

процессоры;

системы на кристалле (СнК, SoC);

системы на программируемых кристаллах (СнПК, SoPC)

и многие другие.

Задачей разработчиков является владение инструментальными средствами анализа и проектирования таких устройств. Такими средствами служат системы автоматизированного проектирования (САПР) электронных устройств, названных общим названием EDA (Electronic Design Automation).

Крупнейшими разработчиками САПР для электроники являются компании Synopsys, Mentor Graphics, Cadence.

Эффективные программы анализа и проектирования электронных устройств TINA, разработала и реализует компания DesignSoft (Венгрия) [8].

Методы работы в среде TINA описаны в книгах автора [9-13].

Проектированию систем на кристалле посвящено учебное пособие [14].

Анализ и проектирование электронных устройств в среде OrCAD 17.2 компании Cadence рассмотрены в [15].

В этой главе мы знакомимся интерфейсом программы TINA 12 – Education и с методами проектирования цифровых устройств в среде TINA. Эта среда достаточно проста для изучения, обладает очень удобным графически интерфейсом, высокой производительностью и много лет используется автором в учебном процессе в РТУ – МИРЭА

3.1. Интерфейс программы TINA 12- Education

TINA Design Suite - это мощный, но доступный программный пакет для анализ, проектирование и тестирование в реальном времени схем с аналоговыми, цифровыми компонентами и микроконтроллерами, определенными в различных языках описания аппаратных средств (PSpice, VHDL, Verilog, Verilog A, Verilog AMS и SystemC) и для разработки макетов их печатных плат. Вы также можете анализировать высокочастотные, связные, оптоэлектронные схемы, и мехатронные приложения с 3D-интерфейсом.

Программу TINA также можно использовать в образовательной среде. TINA включает в себя уникальные инструменты для проверки знаний студентов, мониторинга прогресса обучения и внедрение методов устранения неполадок. С дополнительным оборудованием это может использоваться для проверки реальных цепей для сравнения с результатами, полученными из моделирования. Большое значение для педагогов, имеет то, что пакет включает в себя все инструменты, необходимые для подготовки образовательных материалов.

TINA-12 является мощным инструментом для моделирования аналоговых и цифровых схем, позволяет проводить исследование схем при изменении параметров, оптимизацию, выполнять частотный и спектральный анализ, исследовать переходные характеристики и т.д. Рассмотрим интерфейс программы TINA-12.

3.1.1. Главное окно схемного редактора

После запуска на мониторе появится главное окно схемного редактора (рис. 3.1). Рассмотрим отдельные элементы интерфейса программы.

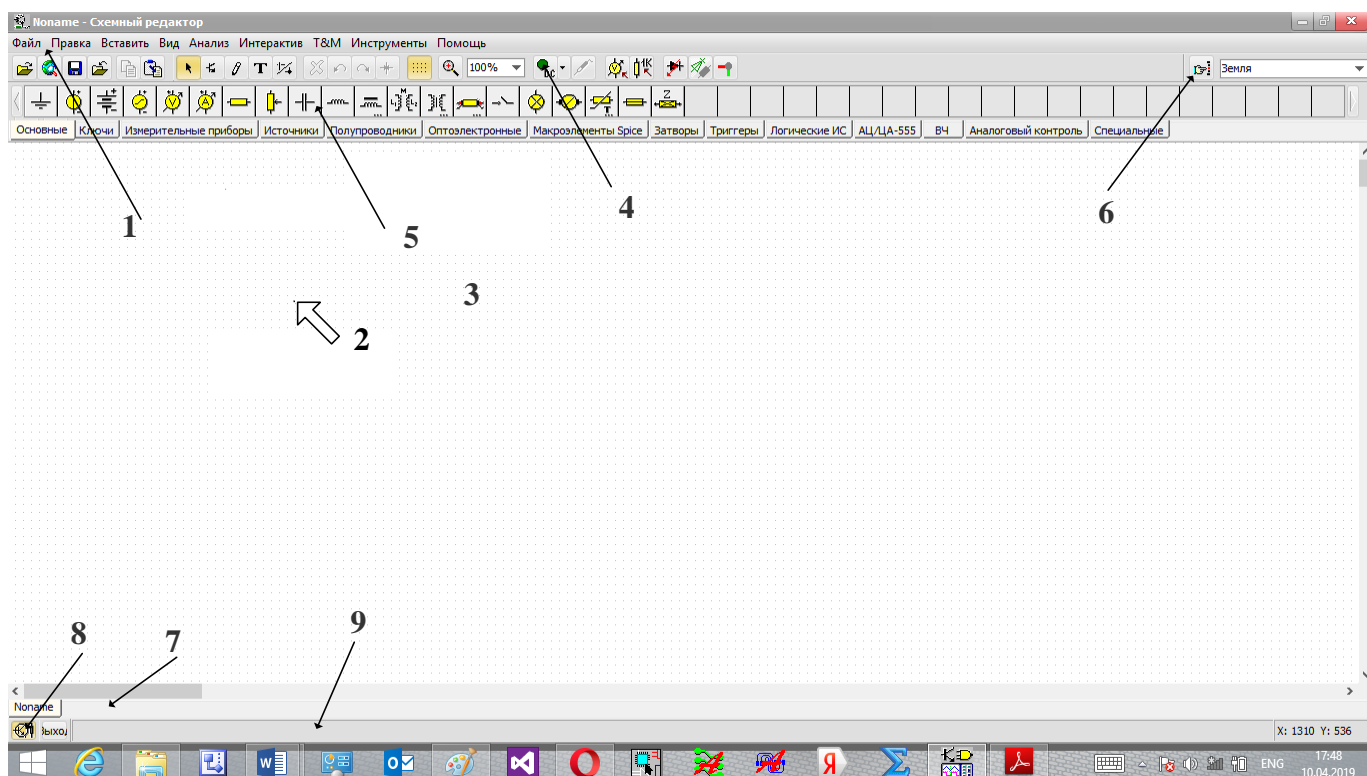


Рис. 3.1. Главное окно редактора схем

1. Панель меню

Основные вкладки главного меню мы изучим позже, а сейчас рекомендуем установить «Язык» и некоторые важные опции редактора (рис. 3.2).

2. Курсор или указатель

Используется для выбора команд и редактировать схемы. Вы можете перемещать курсор только с помощью мыши.

В зависимости от режима работы курсор предполагает одно из следующие формы:

Стрелка, когда в окне редактирования требуется выбор команды.

Символ компонента (сопровождается стрелкой и небольшим прямоугольником), при вставке этого компонента на схему в окно схемы. Пока позиция компонента на схеме не будет выбрана, его движение контролируется мышью.

Ручка при определении конечной точки провода.

Гибкая линия при определении конечной точки проволоки или второго узла входа или выхода.

Гибкая коробка при определении блока после фиксации его первого угла.

Пунктирная линия при размещении метки компонента или текста блок.

Увеличительное стекло, при определении окна масштабирования.

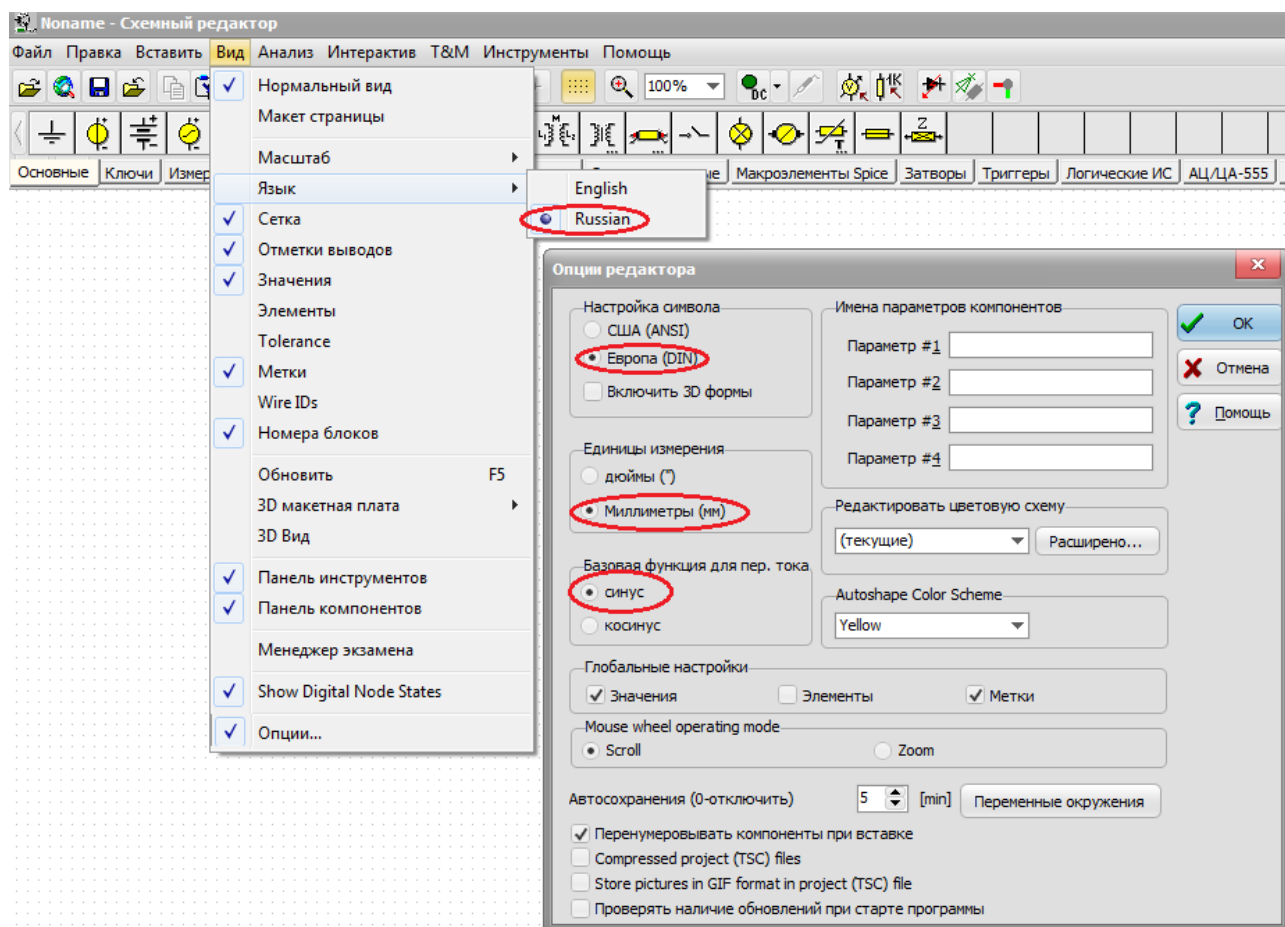


Рис. 3.2. Выбор языка и установка опций редактора

3. Окно схемы

В окне показана принципиальная схема, которая в настоящее время редактируется или анализируется. Окно схемы на самом деле является окном для большой области рисования. Вы можете перемещать экранное окно над всей областью рисования с помощью полос прокрутки справа и в нижней части экрана. При выборе новой команды на меню Файл, система автоматически выравнивает исходное окно редактора с центром всей области рисования редактора. Также происходит, когда загружается существующий файл схемы, так как это значение положения окна по умолчанию. Вы можете думать о схеме TINA, как о существующей на нескольких «слоях». В дополнение к

первичному слою, который содержит компоненты, провода и текст, есть два других слоя рисования, которые вы можете включить или выключить индивидуально. Обычно удобно иметь эти два слоя.

На вкладке главного меню:

Вид/Отметки выводов можно отобразить или скрыть концы выводов компонентов.

Вид/Сетка можно показать или скрыть сетку.

На некоторых уровнях масштабирования вы не увидите точек сетки; тем не менее, все компонентные контакты и соединительные провода будут на сетке. Эти точки представляют собой единственные доступные соединительные точки. Символы компонентов расположены на области рисования по горизонтали и вертикали и привязаны к сетке. Эти символы - жесткие шаблоны с предопределенными позициями штифтов и обрабатываются как отдельные блоки. Это позволяет программному обеспечению однозначно распознавать сеть узлы.

4. Панель инструментов

Панель инструментов позволяет выбрать многие команды редактирования: выделение, масштаб, проволочное соединение и т.д. Рассмотрим наиболее важные команды на панели инструментов. Панель инструментов содержит меню команд (рис. 3.3).



Рис. 3.3. Панель инструментов

Вы можете выбрать большинство команд редактора (например, выбрать, масштаб, провод и т. д.) с этой панели инструментов.

Кроме обычных команд (открытие файла, сохранение и т.п.) в меню команд входят:



- открытие файла из сети;



-режим выделения позволяет выделять или перетаскивать компоненты курсором при нажатии левой кнопки мыши. Выделенные компоненты окрашиваются в красный цвет. Для снятия выделения надо щёлкнуть левой кнопкой мыши на свободном участке поля. Выделенный компонент можно удалять, поворачивать и т.п., щёлкнув правой кнопкой мыши.



- вставка последнего компонента.



- провод, служит для вставки проводников в схему (пишущий карандаш).



- вставка текста или комментариев в схемы и результаты анализа.



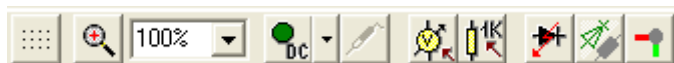
- позволяет разъединять компоненты или удалять соединяющие точки между проводниками и проводными соединениями.



- повороты выделенного компонента.



- зеркальное отражение выделенного компонента.



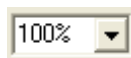
- группа кнопок имеет следующее назначение:



- включение/выключение сетки;



- увеличение масштаба выбранной части текущего вида;



- установка масштаба от 10% до 200%.;

Интерактивное меню включает:



- режим постоянного тока;



- режим переменного тока;



- непрерывный переходной режим;



- однократный переходной режим с установленным временем анали-

за;



- цифровой режим;



- режим работы с цифровыми компонентами и вычислительными

операциями;



- выбор интерактивного режима;



- интерактивная проба.

Далее следуют кнопки:



- выбор цели оптимизации или изменения установок;



- выбор объекта управления;



-включение погрешностей компонентов;

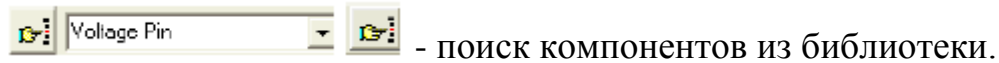


- позволяет показать трёхмерное изображение компонента.



- вызывает диалог, который инициирует модель проектирования пе-

чатных плат.



- поиск компонентов из библиотеки.

5. Панель компонентов

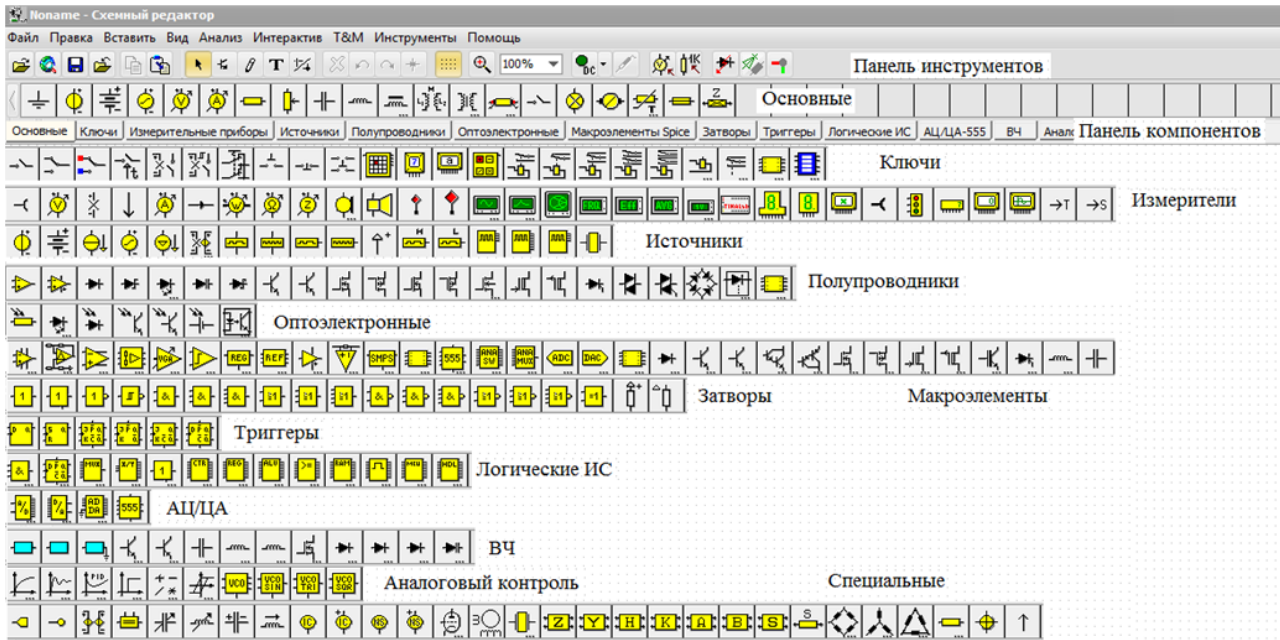


Рис. 3.4. Наборы компонентов

Компоненты расположены в группах, поименованных на кнопках панели компонентов (рис. 3.4). Сначала выбирают группу, а затем требуемый компонент. Щёлкнув на выбранном компоненте, курсором переместите компонент на рабочее поле окна.

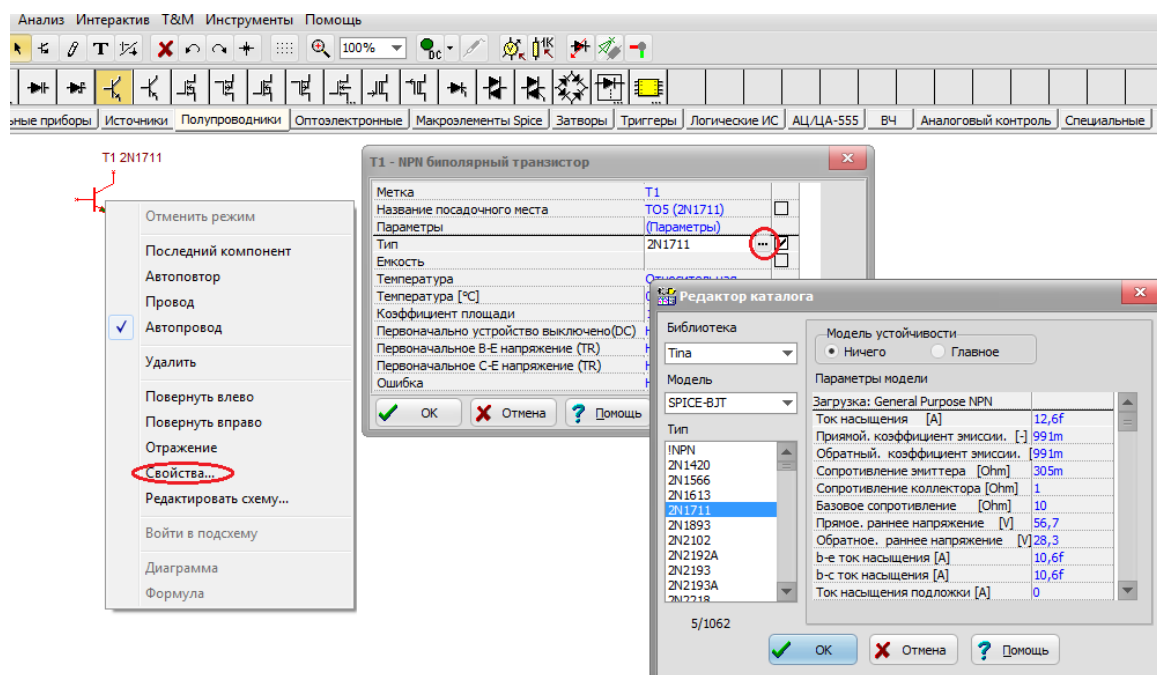


Рис. 3.5. Установка типа компонента

На рабочем поле выделите компонент, правой кнопкой мыши откройте выпадающее окно, выберите «Свойства». Затем выберите тип транзистора и посмотрите параметры модели (рис. 3.5).

Библиотеки компонентов программы TINA весьма обширные и постоянно обновляются.

6. Поиск компонентов

Этот инструмент позволяет найти по имени любой компонент из каталога.

7. Открытие таблицы файлов

Можно иметь несколько разных файлов и выбрать нужный файл из таблицы.

8. Панель задач TINA

Панель задач TINA появляется в нижней части экрана и предоставляет кнопки включения для различных инструментов или приборов T&M, которые в настоящее время используются. Каждый инструмент или прибор работает в своём собственном окне и его можно активировать, нажав на кнопку (значок инструмента). Обратите внимание, что первая кнопка (крайняя слева) (замок) имеет специальную функцию блокировки. Когда окно схемы не заблокировано, и выбраны приборы, вы будете видеть все схематическое окно с любыми другими окнами приборов и инструментов.

9. Линия помощи

Дает краткое описание выделенного компонента.

3.1.2. Редактирование схемы с помощью мыши

Вот несколько основных приёмов мыши, которые помогут вам редактировать схемы:

Использование правой кнопки мыши

Если вы нажмёте правую кнопку мыши в любое время, появится всплывающее окно меню. Используя это меню, вы можете следующие операции:

- *Режим отмены*: выход из последней операции (например, перемещение компонента, рисунок проволоки).
- *Последний компонент*: возврат к последнему компоненту и изменение его положения.
- *Провод*: переключение в режим волочения проволоки. В этом режиме курсор поворачивается в ручку, и вы можете нарисовать провод. Для получения более подробной информации см. абзац «Провод» ниже.
- *Удалить*: удалить выбранные компоненты.

- *Повернуть влево, Повернуть вправо, Отражение*: повернуть или отразить компонент, который в данный момент выбран или перемещается. Вы также можете повернуть выделенный компонент нажатием клавиш Ctrl-L или Ctrl-R

- *Свойства*: используйте эту команду для редактирования свойств (значение, метка) компонента, выбранного в данный момент или перемещаемого. Из меню «Свойства» можно установить все параметры компонента (до его размещения). Это позволяет вам разместить несколько копий компонента с только что введенными свойствами. Пока вы находитесь в Редакторе свойств компонента, правая кнопка мыши имеет ещё одну функцию. Когда вы редактируете поле любого параметра компонента, кроме поля метки, вы можете скопировать это поле рядом с компонентом и его меткой, нажав правой кнопкой мыши и затем выберите команду «Копировать». Вы можете сделать то же самое, нажав [F9].

Использование левой кнопки мыши

В описаниях ниже, термин «щелчок» всегда относится к левой мыши кнопка.левой кнопкой вы можете сделать следующие действия:

- *Выбор*: при нажатии на объект будет выбран нужный объект, а все другие объекты выбраны не будут.

- *Множественный выбор*: нажатие, удерживая клавишу [Ctrl], добавит объект под курсором в группу текущих выбранных объектов. Если объект под курсором уже находится в настоящее время в выбранной группе, нажав [Ctrl] удалите его из группы.

- *Выбор блока*: чтобы выбрать блок объектов одновременно, сначала убедитесь, что под курсором нет объекта. Затем нажмите и удерживайте левую

кнопку при перемещении мыши (перетаскивание). Это создаст прямоугольный блок, и все объекты внутри блока будут выделены.


- *Выбор всех объектов*: нажмите Ctrl + A, чтобы выделить все объекты.
- *Перемещение объектов*: один объект можно переместить, перетаскив его. Наведите курсор на объект, нажмите и удерживайте левую кнопку и переместите мышь. Несколько объектов могут быть перемещены, сначала выбрав их в блок. Затем нажмите левую кнопку, когда курсор будет над одним из выделенных объектов, и, удерживая левую кнопку, перетащите.

- *Изменение параметра*: двойной щелчок по объекту приведёт к его меню параметров, так что вы можете изменить его параметры.

- *Пересечение проводов*: пересечение двух проводов не приводит к соединению на пересечении, если вы сознательно не выбрали, что это один провод. Используйте Edit.Hide / Reconnect для размещения или удаления соединительной точки. Тем не менее, лучше никогда не создавать связи в пересечении проводов, поскольку это позволяет избежать двусмысленности в отношении присутствия или отсутствия точки.

- *Копирование блока или символа*: после того, как блок или символ был выбран, Вы можете скопировать его, нажав Ctrl + C. Затем нажмите левую кнопку за пределами блок или символа, чтобы освободить его, и нажмите Ctrl + V. Вы увидите копию блока, который вы можете разместить, как вы хотите. Если для схемы в окне недостаточно места для копии, нажмите Alt – O или Alt - I, чтобы изменить масштаб. Как только вы нашли блок, щёлкните левой кнопкой мыши один раз, чтобы закрепить его и во второй раз отменить выбор перемещённого блока.

3.1.3. Размещение компонентов

Компоненты выбирают из строки компонентов редактора схем. Их символы перемещают с помощью мыши в требуемое положение. При нажатии на левую кнопку мыши, программа устанавливает контакты символа компонента на ближайшей точке сетки точек. Компоненты могут быть расположены вертикально или горизонтально, их можно поворачивать с шагом в 90 градусов с помощью кнопок . Кроме того, некоторые компоненты (например, транзисторы) можно зеркально отобразить вокруг их вертикальной оси, нажав кнопку на панели инструментов.

На сенсорных экранах при размещении компонентов коснитесь символа компонента на панели инструментов. Затем нажмите приблизительное местоположение на поверхности редактирования, чтобы поместить туда символ. Компонент будет помещён в области редактирования. После этого Вы можете коснуться и перетащить символ на окончательное место с помощью пальца или с помощью позиционирующего устройства, если оно имеется.

После того, как символ компонента был выбран и позиционирован, Вы можете дважды щёлкнуть на нем, чтобы включить диалоговое окно, в котором можно ввести значения параметров и метки. При вводе числовых значений используют аббревиатуры дольных и кратных величин в пределах от 10^{-12} до 10^{12} .

Таблица 3.1


p=пико= 10^{-12}	T=тера= 10^{12}
n=нано= 10^{-9}	G=гига= 10^9
u=микро= 10^{-6}	M=мега= 10^6
m=милли= 10^{-3}	k=кило= 10^3

Соединение компонентов

Провод устанавливает простую перемычку (нулевое омическое соединение) между двумя контактами компонента. Для того, чтобы поместить провод, переместите курсор к контактной точке компонента, откуда Вы хотите начать проводник. Курсор превратится в крестик. Вы можете нарисовать провод двумя различными способами:

1) Выберите начальную точку провода левой кнопкой мыши, а затем переместите крестик с помощью мыши до контакта другого компонента. TINA рисует провод вдоль пути перемещения. Двигаться можно в любом направлении и провод будет отслеживать путь с прямоугольными поворотами. В конечной точке провода, снова нажмите левую кнопку мышь.

2) Можно, удерживая нажатой левую кнопку мыши, провести крест до конечного контакта и выпустить кнопку в конечной точке. Протягивая провод, можно удалить предыдущие участки, перемещая крест назад по той же дорожке. При нажатии на клавиши Ctrl+C, Ctrl+V можно копировать и вставлять участки проводников. После небольшой тренировки Вы научитесь это делать.

Для коротких сечений проводов, возможно, потребуется удерживать клавишу переключения во время рисования. Можно также вызвать инструмент рисования провода или шины  и начать рисовать провод в любом месте, нажав левую кнопку мыши. Когда Вы завершили подключение, используйте в всплывающем меню *отменить* или нажмите правую кнопку мыши, Правая кнопка или клавиша Esc прекратит режим проводки. Будьте уверены, чтобы Вы не оставили никаких не подключённых контактов. Если есть несвязанные компоненты или терминалы, TINA проверит электрические правила соединения

(ERC) и выдаст предупреждение. Проводники с помощью инструмента Wire всегда расположены вертикально или горизонтально. Тем не менее, вы можете добавить угловые сегменты проводов с помощью компонент, изготовленных для мостов, схем треугольников и звёзд из панели специальные компоненты.

Входы и выходы

Некоторые виды анализа (переходные характеристики на постоянном токе, Боде - диаграммы и т.д.) не могут выполняться, пока не определены входы и выходы цепи. Они определяют, где прилагается воздействие и откуда снимать реакцию цепи. Выходы также определяют, какие графики будут отображаться в выбранном режиме анализа. Источники и генераторы должны быть подключены к входам, а измерители – к выходам. Однако измерители могут также служить для определения количественного значения входных сигналов, которые будут использоваться при вычислении переходных характеристик и функций в режиме переменного тока. Так у вольтметров и амперметров есть два режима: ввод / вывод. Это надо учитывать при моделировании, когда должен быть только один ввод.

3.1.4. Основные режимы работы TINA

Возможности и достоинства TINA не уступают программам других ведущих разработчиков САПР (Cadence, National Instruments) и состоят в следующем:

1. Результаты на постоянном и переменном токе легко получаются в виде таблиц напряжения в узлах, на всех элементах, токов во всех элементах, других напряжений. Причём на переменном токе вычисляются амплитуды и фазы.
2. Во всех узлах можно померить напряжении специальным щупом.
3. Программа TINA имеет режим многовариантного анализа (*Шаг параметра*), в котором значение параметров выбранных компонентов

варьируется на каждом шаге вычислений. В результате вычисляется и строится набор графиков, который иллюстрирует чувствительность цепи к изменению параметров компонентов. Изменяемое значение параметра компонента может быть любым численным параметром.

4. Легко получаются амплитудно-частотные и фазо-частотные характеристики цепей. Анализ передаточных характеристик позволяет в режиме *Анализ переменного тока – Переходные характеристики переменного тока* получить амплитудно-частотные характеристики (АЧХ) и фазо-частотные характеристики (ФЧХ), а также, используя *Символический анализ*, получить аналитическое выражение передаточной функции

5. Удобно исследовать переходные характеристики цепей и получать их в виде графиков. Выбрав в главном меню *Анализ-Анализ переходных процессов*, можно провести анализ переходных характеристик, регистрировать графики и получить в редакторе уравнений аналитическое выражение отклика.

6. *Анализ Фурье* вычисляет спектры сложных сигналов и рисует их на диаграммах

7. Передаточные функции и переходные характеристики можно получить в виде аналитических выражений в режиме *Символический анализ*.

8. Предусмотрен режим *Оптимизация*, позволяющий выбрать оптимальные параметры цепи для достижения поставленной цели. Целевой отклик цепи (напряжение, ток, сопротивление или мощность) должны наблюдаться измерителями, предварительно установленными на нужных позициях. Неизвестные параметры цепи будут определены автоматически тогда, когда цепь выдаст целевой выходной результат.

9. TINA включает в себя очень быстрый и мощный симулятор для цифровых схем. В режиме *Цифровой анализ* можно следить за операциями в

цепи шаг за шагом, вперёд и назад, или наблюдать полную временную диаграмму в специальном окне логического анализатора.

10. TINA включает интегрированный VHDL симулятор для проверки VHDL – устройств (спроектированных с помощью высокоуровневого языка описания аппаратуры «*Very High Speed Integrated Circuits Hardware Description Language*»), как в цифровых, так и в аналого-цифровых реализациях.

TINA содержит в библиотеке большое число микроконтроллеров (PIC, AVR, 8051), которые можно тестировать, программировать и запускать в интерактивном режиме. Встроенный программатор позволяет модифицировать программы и наблюдать результаты.

11. *Анализ шумов* определяет шумовой спектр, который соответствует каждому входу или выходу. Может быть вычислена мощность шума и соотношение сигнал-шум.

12. *Статистический анализ и анализ наихудших случаев* позволяет установить требуемую точность элементов цепи и получить статистические характеристики.

Для обучения школьников и студентов работе с программой TINA и предназначен этот лабораторный практикум по электротехнике, электронике и схемотехнике, который подтверждает, что программа TINA может быть успешно использована в научных исследованиях и в образовании, как одна из лучших систем автоматизированного проектирования электронных устройств.

3.2. Моделирование простых цифровых схем

Начнем с исследования простой цифровой схемы. Откройте файл HALF_ADD.TSC из папки ПРИМЕРЫ. В меню Анализ выберите Цифровой - Пошаговый. Появится панель управления, и вы сможете проверить поведение пошаговой схема, нажав кнопку «шаг вперед». После четырех шагов

моделирование остановится, так как в генераторах PSG1 и PSG0 установлены шаблоны на 4 шага.

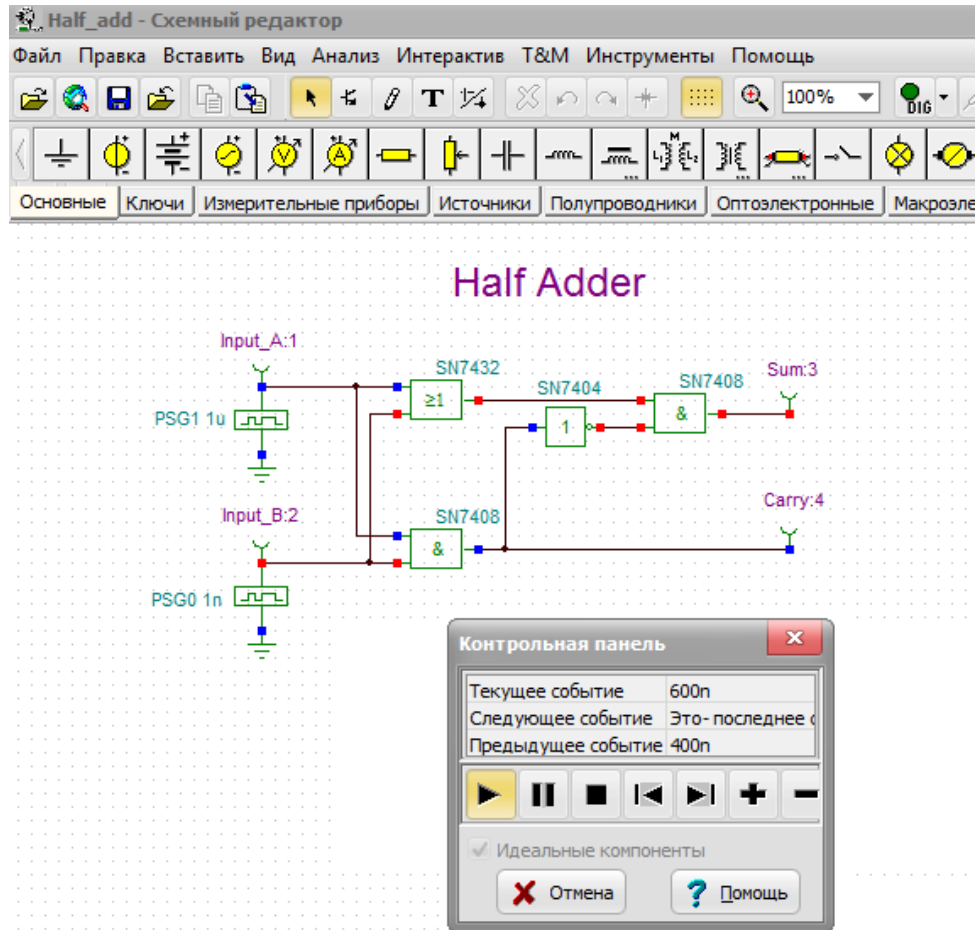


Рис. 3.6. Модель полусумматора

Нажмите кнопку Стоп, а затем Воспроизведение. Начнется автоматическое моделирование четырех шагов.

На каждом шаге маленькие квадратики в узлах будут показывать логический уровень (красный для высокого, синий для низкого, зеленый для высокого Z, черный для неопределенного), поскольку схема синхронизирована.

На рис. 3.6 показано состояние дисплея после четвертого шага.

Теперь давайте рассмотрим переходное поведение схемы. Выберите Анализ – Цифровой временной анализ. Эта команда вызывает меню Цифровое

VHDL моделирование (рис. 3.7). Установим время моделирования 1 мкс и нажмем ОК.

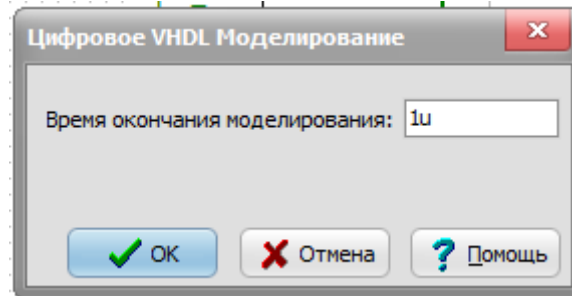


Рис. 3.7. Меню цифрового VHDL моделирования

Получим временную диаграмму сигналов на входах и выходах полусумматора (рис. 3.8).

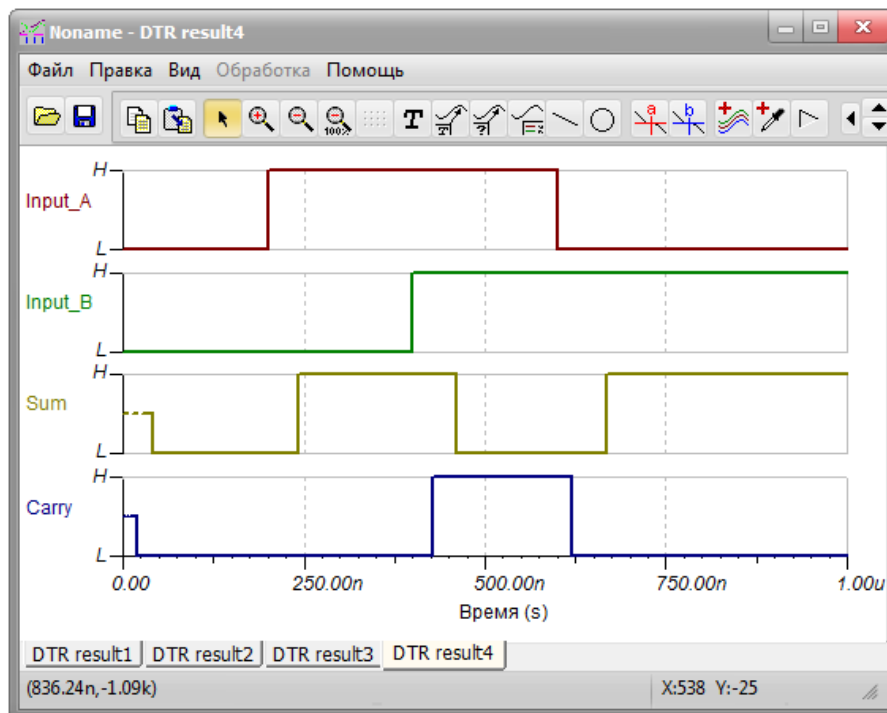


Рис. 3.8. Временная диаграммы сигналов на выходах полусумматора

Вы также можете выбрать режим Анализ переходных процессов вместо Цифрового временного анализа. В этом случае программа будет проводить

аналоговый анализ, давая детальные непрерывные формы волны и напряжения вместо идеализированных логических уровней (рис. 3.9).

Обратите внимание, что схемы, которые содержат только цифровые компоненты могут быть проанализированы как цифровыми, так и аналоговыми методами.

Еще один способ – это использование логического анализатора. В главном меню выберите T&M – Логический анализатор. Установите параметры моделирования, показанные на рис. 3.10 и нажмите Run.

Временная диаграммы сигналов показана на рис. 3.10.

Вы можете установить порядок кривых, просто добавив двоеточие (:) символ и число к имени выхода. Это особенно важно при представлении результатов цифрового анализа, где каждый вывод отображается в виде отдельной диаграммы. Например, если у вас есть выходы с именами OutA, OutB, Carry и Sum, вы можете убедиться, что они будут отображаться в порядке, указанном с помощью меток OutA: 1, OutB: 2, Carry: 3, а Sum: 4.

Результаты чисто аналогового анализа обычно отображаются на одной диаграмме. Однако вы можете заставить TINA отображать результаты в отдельных диаграммах, в порядке, который вы хотите, используя метод маркировки, описанный выше.

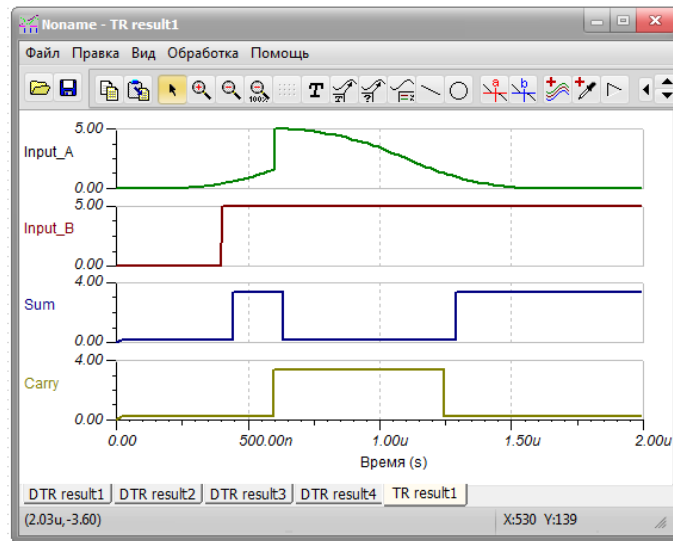


Рис. 3.9. Временные диаграммы при аналоговом моделировании

На временной диаграмме рис. 3.9 в меню Вид Вы можете использовать команда Отдельные кривые или Показать/Скрыть кривые, чтобы отделить кривые в окне (рис. 3.11). Если вы не используете этот метод маркировки, TINA представляет кривые в алфавитном порядке.

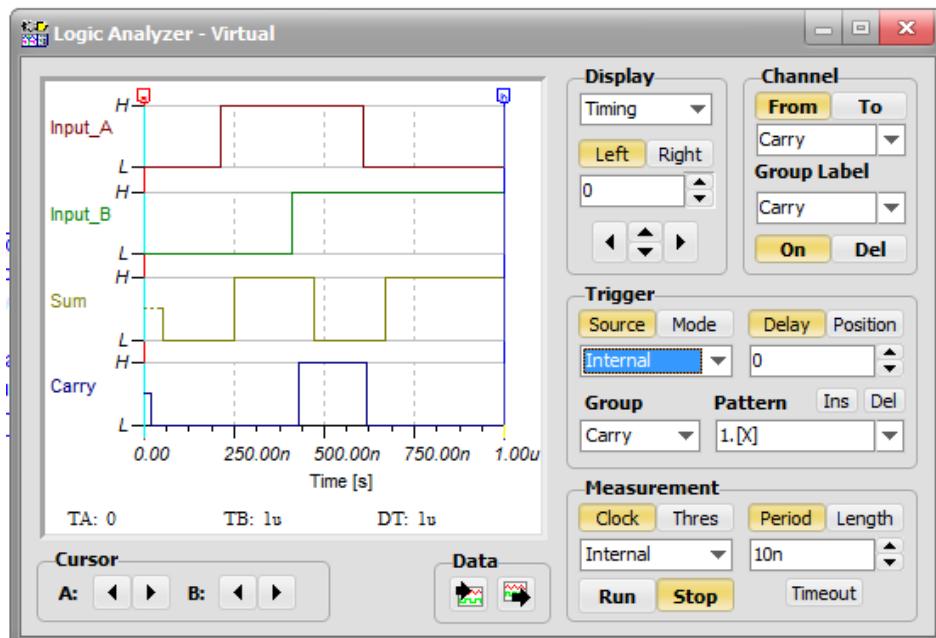


Рис. 3.10. Временная диаграмма в логическом анализаторе

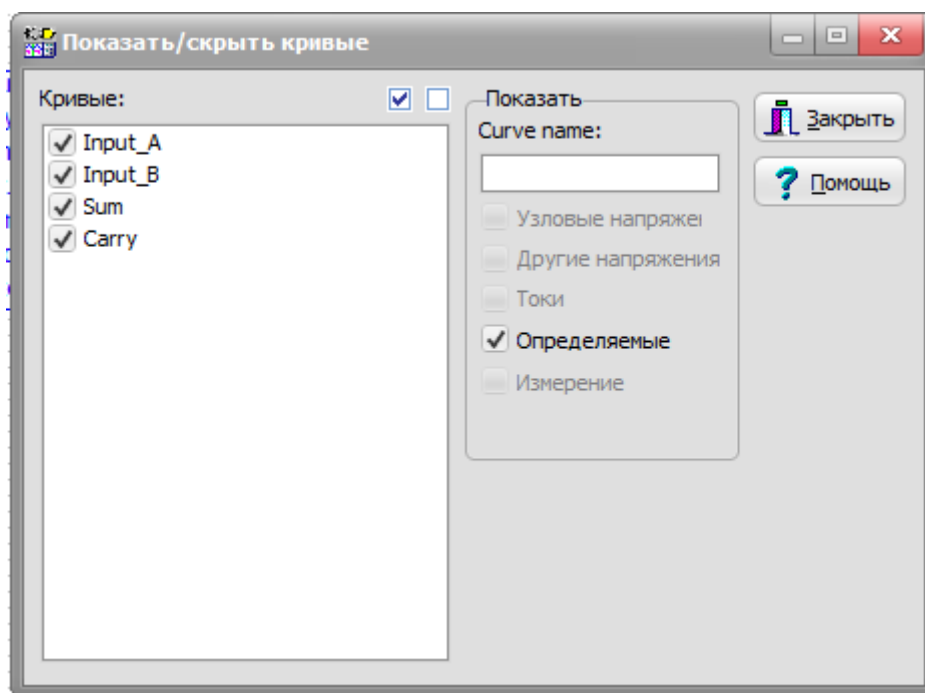


Рис. 3.11. Выбор кривых для отображения

3.3. Анализ цифровых цепей с использованием цифровых HDL имитационных моделей

Языки описания оборудования (HDL) являются стандартными текстовыми языками моделирования, которые используют разработчики электроники для описания и моделирования их чипов и систем до изготовления.

ТИНА-12 включает в себя четыре наиболее широко используемых языка описания аппаратуры (HDL - HardwareDescription Language), определенные стандартами IEEE: VHDL, Verilog, Verilog -A и Verilog -AMC.

VHDL и Verilog используются для моделирования цифровых схем. Два языка сопоставимы при моделировании цифрового оборудования. Тем не менее поведенческие возможности VHDL являются более мощными, в то время как Verilog легче учить и понимать. В TINA вы можете использовать и смешивать модели обоих языков.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Verilog-A - это легко читаемый язык высокого уровня для моделирования аналоговых электронных схем и устройств (например, биполярных и МОП транзисторов).

Verilog-AMS является расширением Verilog для моделирования аналоговых и смешанных сигнальных цепей, позволяющий выполнять инструкции как Verilog, так и Verilog-A, подключить модули и правила.

Полное представление языков HDL в TINA выходит за рамки этого руководства. Мы отсылаем заинтересованного читателя к подробным стандартам, руководствам и информации в Интернете: www.vhdl.org и www.verilog.org.

В следующих разделах мы продемонстрируем использование этих языки через примеры.

3.3.1. Анализ цифровой схемы с использованием цифрового VHDL моделирования

TINA включает в себя мощный цифровой механизм моделирования VHDL. Любая цифровая схема в TINA может быть автоматически преобразована в VHDL код и анализируется как VHDL дизайн. Кроме того, вы можете проанализировать широкий спектр оборудования, описанного в VHDL, и описать ваши собственные цифровые компоненты и оборудование в VHDL. Большое преимущество VHDL - это не только описание оборудования на языке стандарта IEEE, но и то, что это может быть реализовано автоматически в программируемые логические устройства, такие как FPGA (field-programmable gate array- программируемая пользователем вентильная матрица, ППВМ) и CPLD (Complex Programmable Logic Device - программируемая логическая интегральная схема (ПЛИС) с внутренней энергонезависимой конфигурационной памятью, ПЛИС).

TINA может генерировать синтезируемый код VHDL вместе с соответствующим файлом UCF (файл пользовательских ограничений для назначения выводов в FPGA), если в поле «Установить синтезируемый код» установлен флажок Анализ / Параметры меню. Вы можете сохранить созданные VHD и UCF файлы с помощью команды «Создать файл VHD & UCF» в T & M меню. Вы можете читать файлы с помощью бесплатного Xilinx Webpack и генерировать файл потока битов, описывающий реализацию дизайна, а затем загрузить его на чипы Xilinx FPGA.

Перед реализацией дизайна VHDL с дискретными компонентами или с FPGA, проверьте это с помощью симуляции, используя в меню TINA Анализ – Цифровой временной анализ.

Давайте рассмотрим некоторые аспекты VHDL моделирования.

Чтобы сделать наш первый анализ VHDL, откройте схему FULL_ADD.TSC из папки EXAMPLES \ VHDL. Появится следующая схема (рис. 3.12).

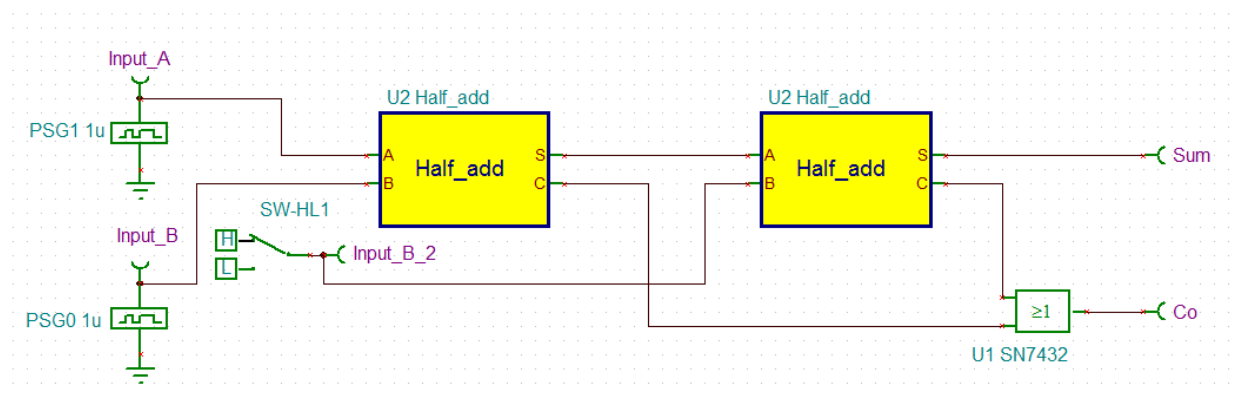


Рис. 3.12. Схема полного сумматора VHDL

Эта схема является комбинацией двух VHDL половинных блоков сумматора (макросов) и дискретный логический элемент ИЛИ.

Если вы дважды щелкнете по любому из блоков полусумматора, а затем в окне (рис. 3.13) нажмете кнопку Вывести подсхему (Enter Macro), появится следующее окно редактора VHDL (рис. 3.14):

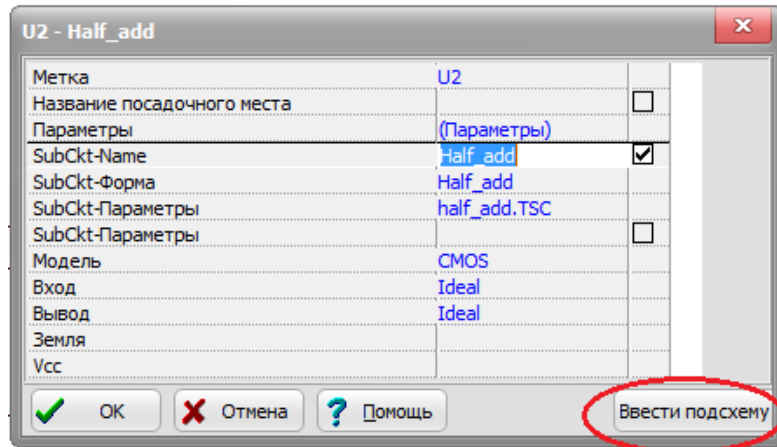


Рис. 3.13. Окно свойств полусумматора

Обратите внимание, что основной код VHDL половинного сумматора находится внизу и это только:

```
S <= ( N5
AND N6 );
N6 <= NOT (
C );
C <= ( A
AND B );
N5 <= ( A
OR B );
```

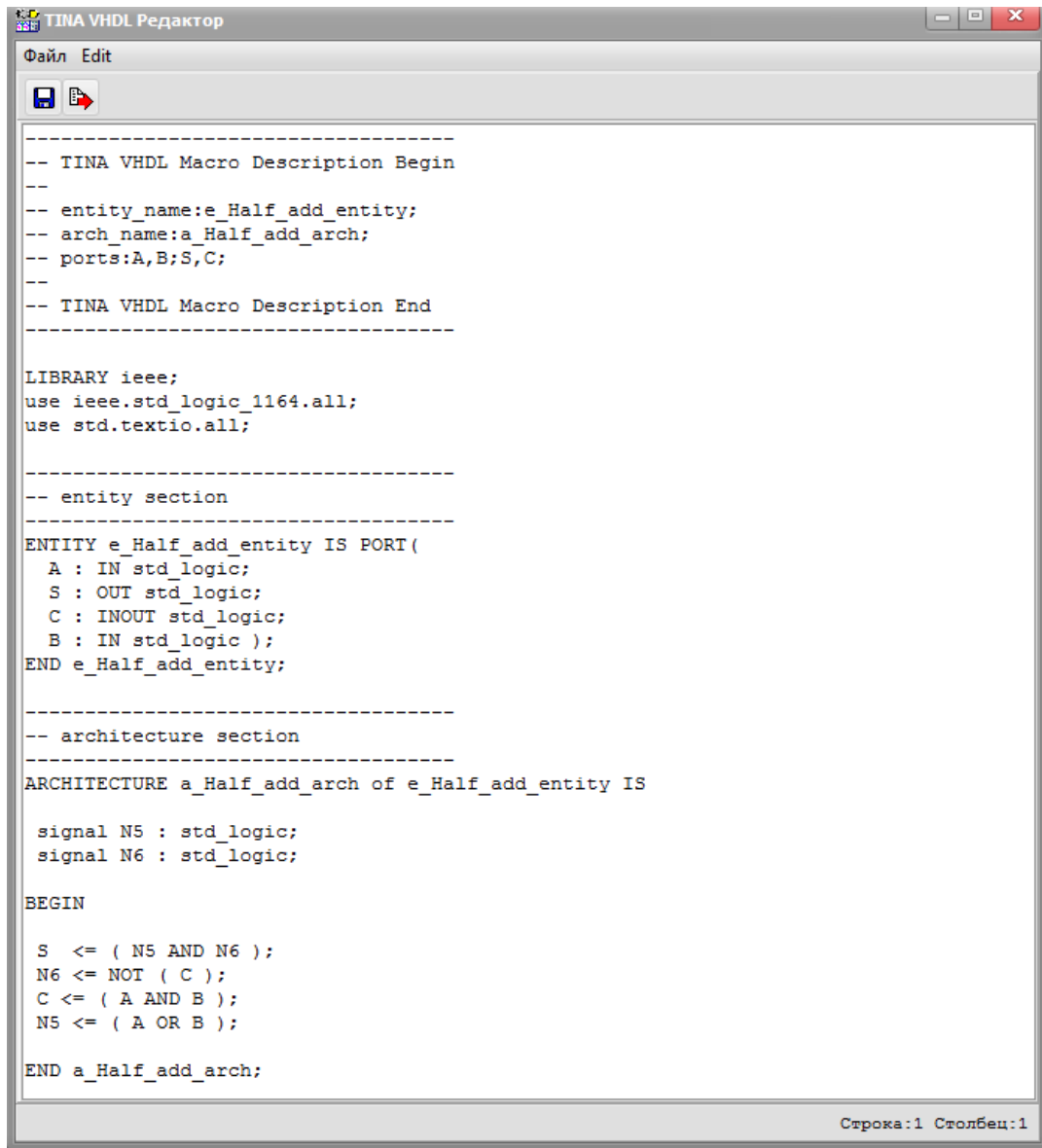


Рис. 3.14. Окно редактора VHDL

На первый взгляд, код может выглядеть немного странно, но на самом деле это машинный перевод нашей половины сумматора, собранного из вентилях (рис. 3.12).

Представляем имена узлов N5 и N6, как показано на рис. 3.15.

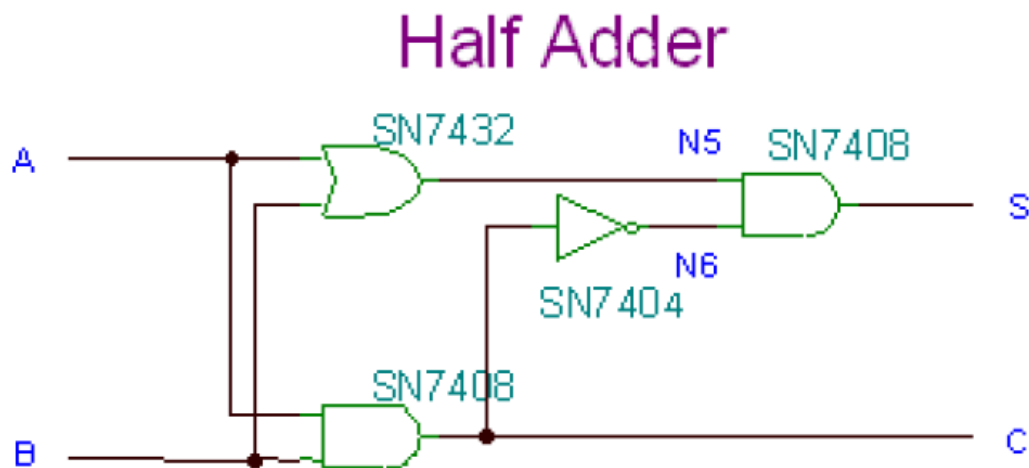


Рис. 3.15. Нумерация узлов полусумматора

Теперь стало ясно, что:

```

C <= ( A
AND B );
N6 <= NOT
( C );
N5 <= ( A OR B );
и поэтому
S <= ( N5 AND N6 );
  
```

Может показаться странным, что в коде VHDL S отображается, чтобы быть рассчитанным по N5 и N6 еще до того, как N5 и N6 были рассчитаны. Это верно, однако, потому что VHDL является параллельным языком, и порядок строк в программе не означает порядок выполнения.

Задержки взяты из заданных дискретных значений. Но если цель аппаратной части - FPGA, программа синтезатора будет использовать задержку значения паспорта ПЛИС.

Теперь выберите Анализ – Цифровой временной анализ из меню анализа и нажмите ОК. Появится следующая диаграмма (рис. 3.16):

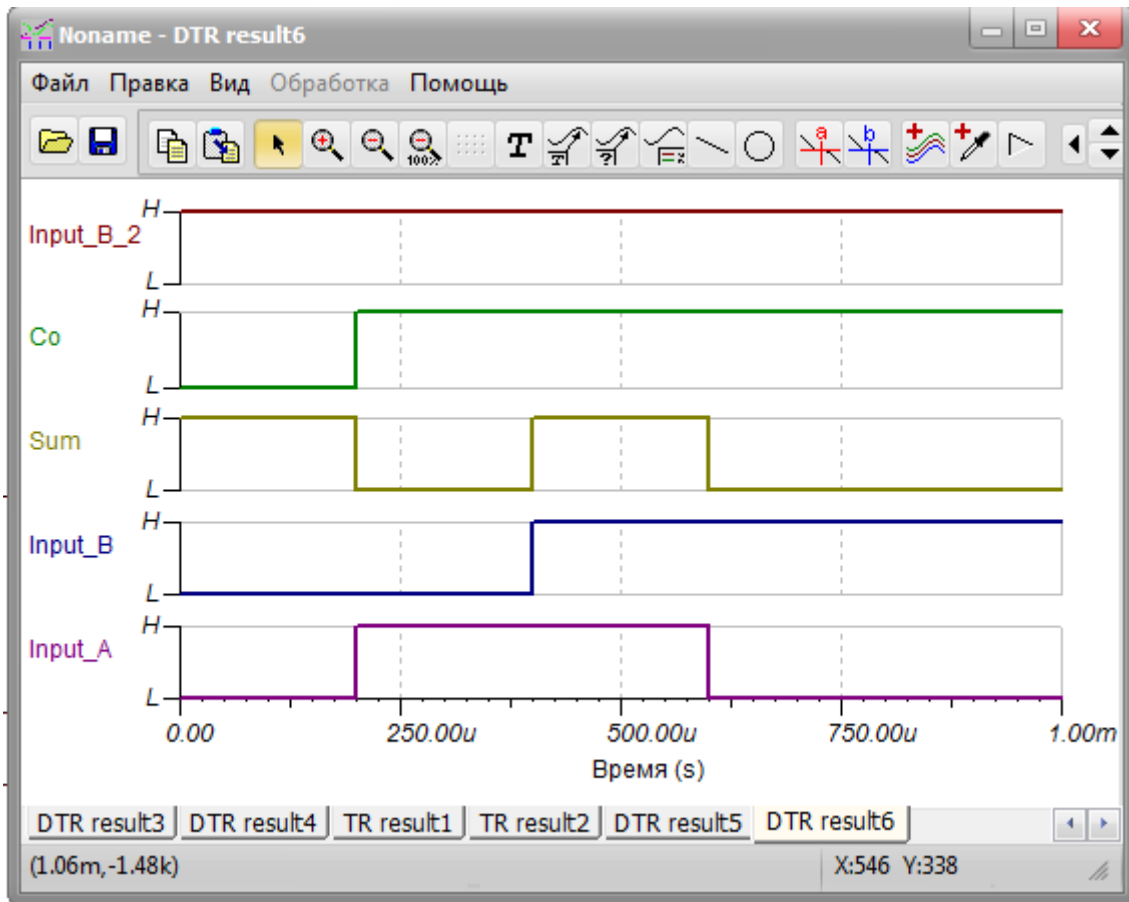


Рис. 3.16. Временная диаграмма VHDL – моделирования

Отличительной особенностью VHDL TINA является то, что вы можете не только просматривать VHDL-код каждого компонента, но вы можете редактировать и запускать их немедленно. Заменим 4-строчный код VHDL –

```
S <= ( N5 AND N6 );
```

```
N6 <= NOT ( C );
```

```
C <= ( A AND B );
```

```
N5 <= ( A OR B );
```

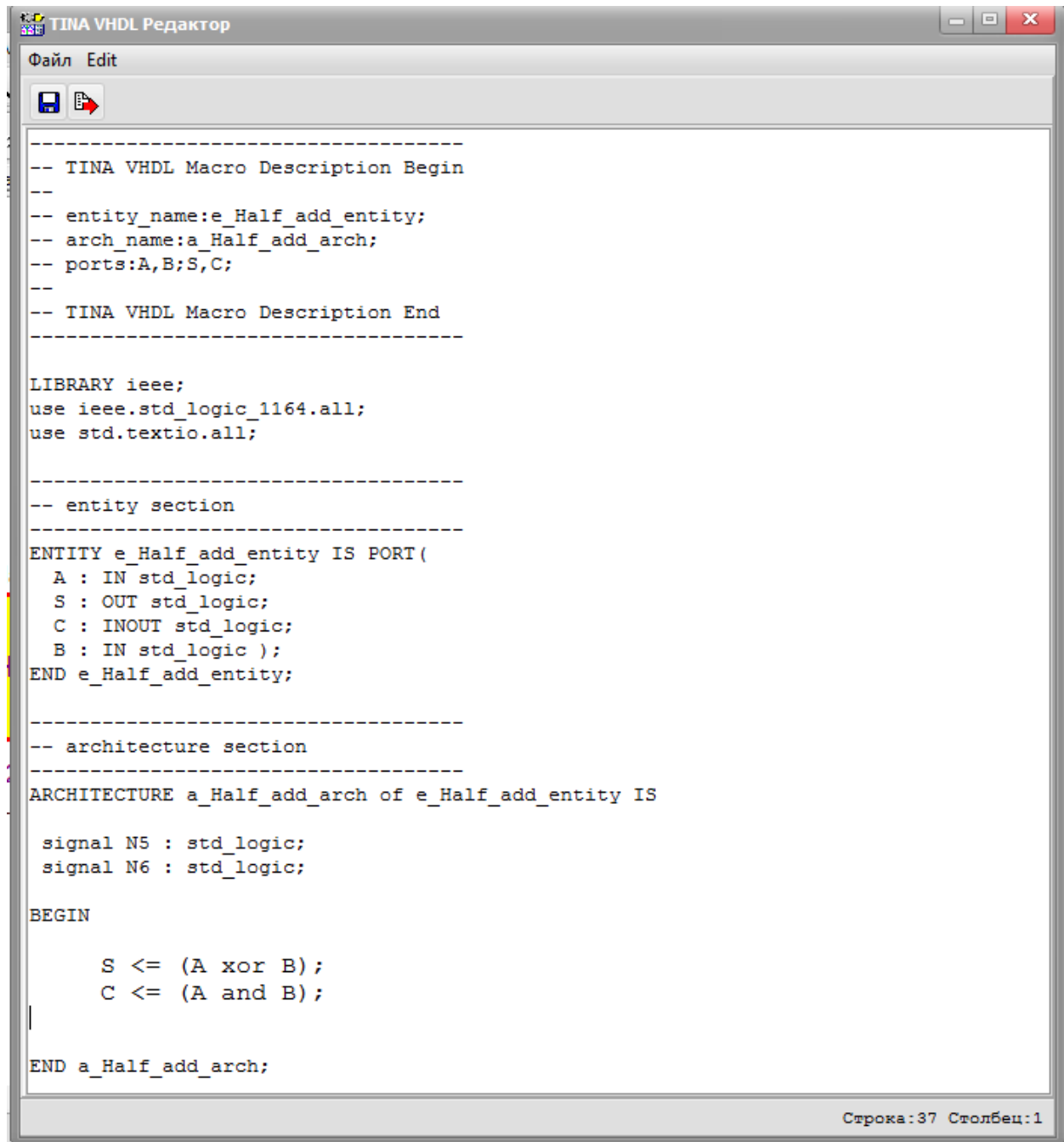
этим более простым двухстрочным кодом:

```
S <= (A xor B);
```

```
C <= (A and B);
```


Это легче понять. На самом деле, если один из входов А или В есть true, S это true. (А и В). Мы распознаем это как функцию Хор.

После редактирования содержимого блоков VHDL они должны выглядеть следующим образом (рис. 3.17):



```

-----
-- TINA VHDL Macro Description Begin
--
-- entity_name:e_Half_add_entity;
-- arch_name:a_Half_add_arch;
-- ports:A,B;S,C;
--
-- TINA VHDL Macro Description End
-----

LIBRARY ieee;
use ieee.std_logic_1164.all;
use std.textio.all;

-----
-- entity section
-----
ENTITY e_Half_add_entity IS PORT(
  A : IN std_logic;
  S : OUT std_logic;
  C : INOUT std_logic;
  B : IN std_logic );
END e_Half_add_entity;

-----
-- architecture section
-----
ARCHITECTURE a_Half_add_arch of e_Half_add_entity IS

  signal N5 : std_logic;
  signal N6 : std_logic;

BEGIN


    S <= (A xor B);
    C <= (A and B);

END a_Half_add_arch;
  
```

Строка:37 Столбец:1

Рис. 3.17. Новый VHDL – код



Теперь закройте окно редактирования, нажав на  на панели инструментов редактора. Выберите Цифровой временной анализ из меню анализа и нажмите ОК. Нарисованная диаграмма (рис. 3.18) будет практически идентична предыдущему графику.

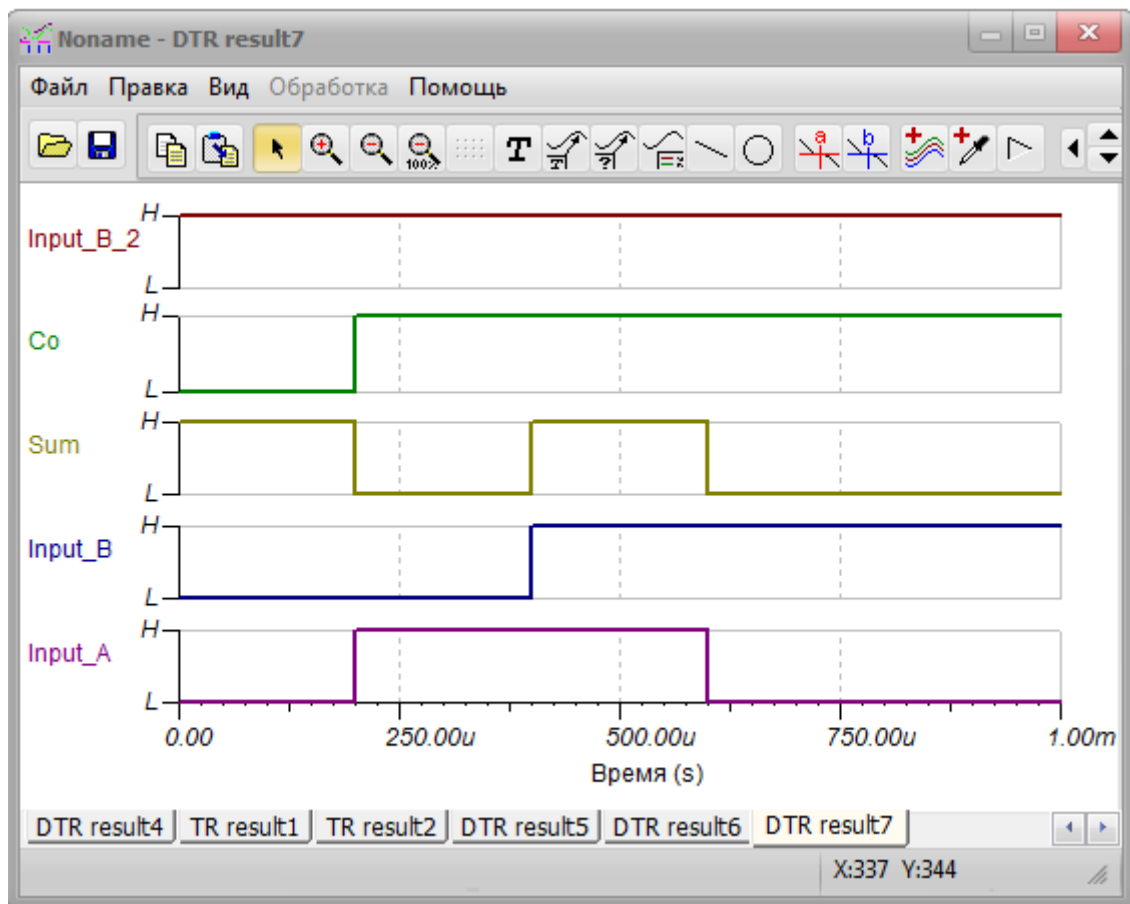


Рис. 3.18. Новая диаграмма после редактирования VHDL – кода.

3.3.2. Отладчик HDL: отладка VHDL кодов

Отладка HDL-программ особенно трудна из-за параллельных процессов в этих языках.

Отличной особенностью TINA является то, что отладчик HDL теперь интегрирован.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Вы можете:

- Выполнять VHDL и Verilog коды оператор за оператором.
- Выполнять подпрограммы как один оператор (Step Over).
- Добавить точки останова (Toggle Breakpoint), работающие непрерывно (Пуск) и останавливая программу на контрольных точках.
- Поместить переменные, сигналы и другие объекты на вкладку Watches и увидеть их значение во время отладки.
- Выполнить просмотр всех точек останова и объектов под точками останова и локальными вкладками в нижней части окна отладчика HDL.

Чтобы попрактиковаться в использовании отладчика HDL в TINA, откройте файл `vhdl_counter.TSC` из папки `EXAMPLES \ VHDL` с командой в меню Открыть Файл. Далее щелкните меню «Анализ» и включите отладчик, щелкнув строку «Включить HDL Debugger».

Наконец, нажмите кнопку DIG на панели инструментов в верхней части экрана и нажмите Пуск в Интерактивном меню. Появится Отладчик HDL (рис. 3.20). Перейдите на вкладку `counter.vhd` внизу кода.

Вы должны увидеть следующий экран.

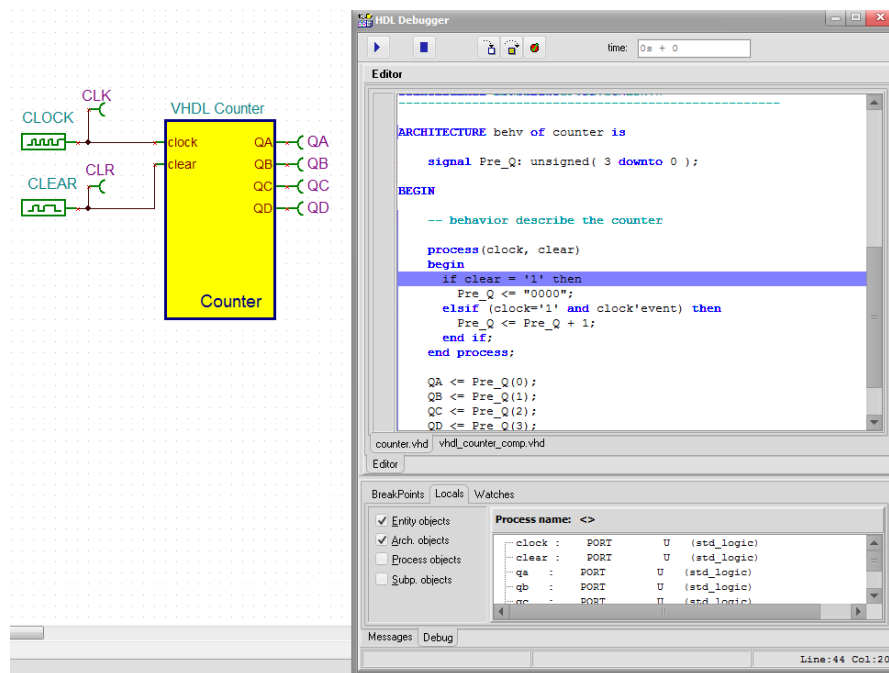



Рис. 3.20. Окно отладчика VHDL

Вы увидите два модуля под вкладками counter.vhd и vhd_counter_comp.vhd. Первым является содержимое макроса счетчика VHDL, в то время как второй файл является преобразованием VHDL всей схемы, включающим источники igiti.

Этот макрос реализует счетчик. Сущность (entity) объекта состоит из пяти процессы, и все процессы работают параллельно. Первый процесс чувствителен к такту и четкому сигналу. Так, когда один из сигналов меняется, этот процесс запускается и выполняется. Другие процессы чувствительны к сигналу Pre_Q. Когда Pre_Q (i) изменяется, i + 1-й процесс запускается и выполняется.

Чтобы выполнить некоторые важные шаги в программе, давайте добавим четыре точки останова, нажав на интересующие линии и нажав кнопку точки останова Toggle BreakPoint . Линии с точкой останова будут помечены красным фоном (рис. 3.21).

Обратите внимание, что на вкладке Точки останова вы можете увидеть все точки останова и удалить любую из них с помощью кнопки Toggle Breakpoint.

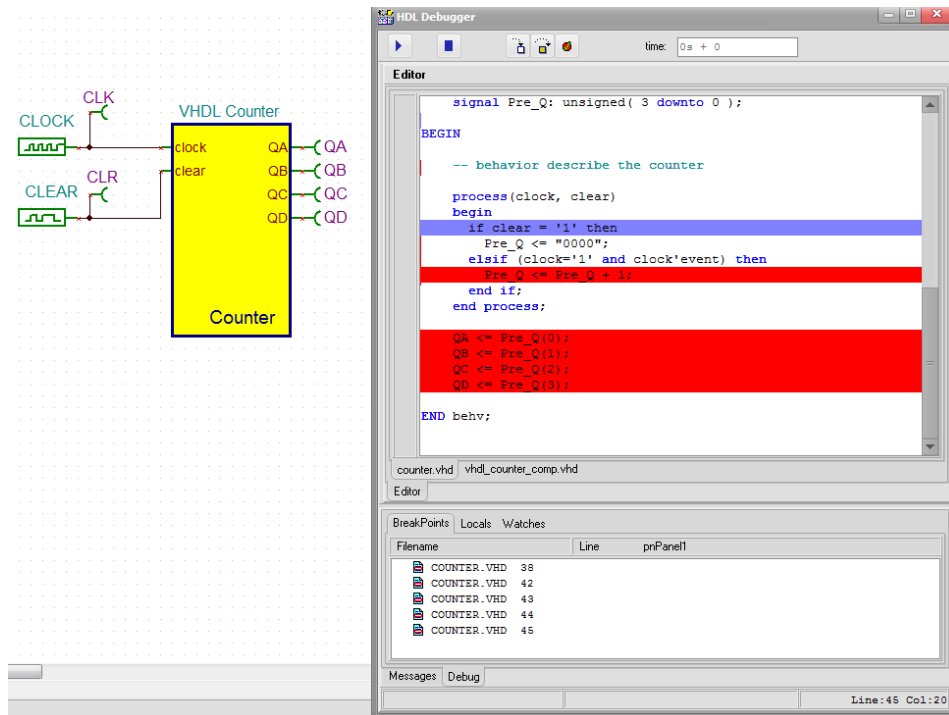


Рис. 3.22. Окно отладчика с точками останова

Теперь перейдите на вкладку Watches, нажмите кнопку Add Object и один за другим добавьте сигнал Pre_Q и порты QA, QB, QC, QD.

Они будут связаны с 4 выходами макроса.

В окне отладчика HDL вы увидите следующее (рис. 3.23).

Теперь давайте начнем отладку, нажав кнопку Run.

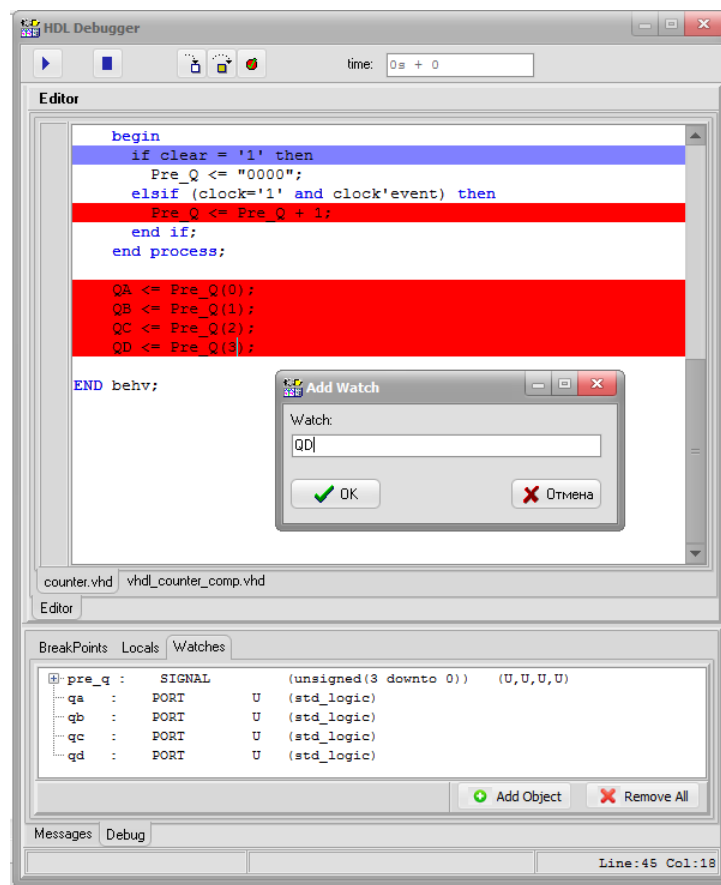


Рис. 3.23. Окно отладчика с точками наблюдения

В моделировании VHDL каждый процесс запускается один раз в начале моделирования. Нажмите кнопку Run несколько раз, пока отладчик не покажет время $500 \text{ нс} + 1$ в поле времени. Это означает, что симулятор достигает 500 нс и 1 дельта-цикла (дельта-цикл представляет собой особый период времени VHDL с бесконечно малой продолжительностью). В это время, равное $1 (= 1)$, Pre_Q был инициализирован с "0" в свободном состоянии. Линия $\text{Pre_Q} \leq \text{Pre_Q} + 1$ запланирует транзакцию по сигналу Pre_Q со значением «0001» для времени $500 \text{ нс} + 2$.

Нажмите Run еще раз, и отладчик остановится на точке останова $\text{QA} \leq \text{Pre_Q}(0)$. В этот момент время = $500 \text{ нс} + 2$, потому что ближайшим событием был ранее описанный запланированный случай. Теперь симулятор

запланирует событие, которое назначит значение «1» на время = 500 нс + 3 для порта QA.

Нажмите Run еще раз, теперь время = 1.5us + 1 и QA = '1'. Обратите внимание что, последние три процесса теперь не запускаются, потому что не было никаких изменений в их список чувствительности.

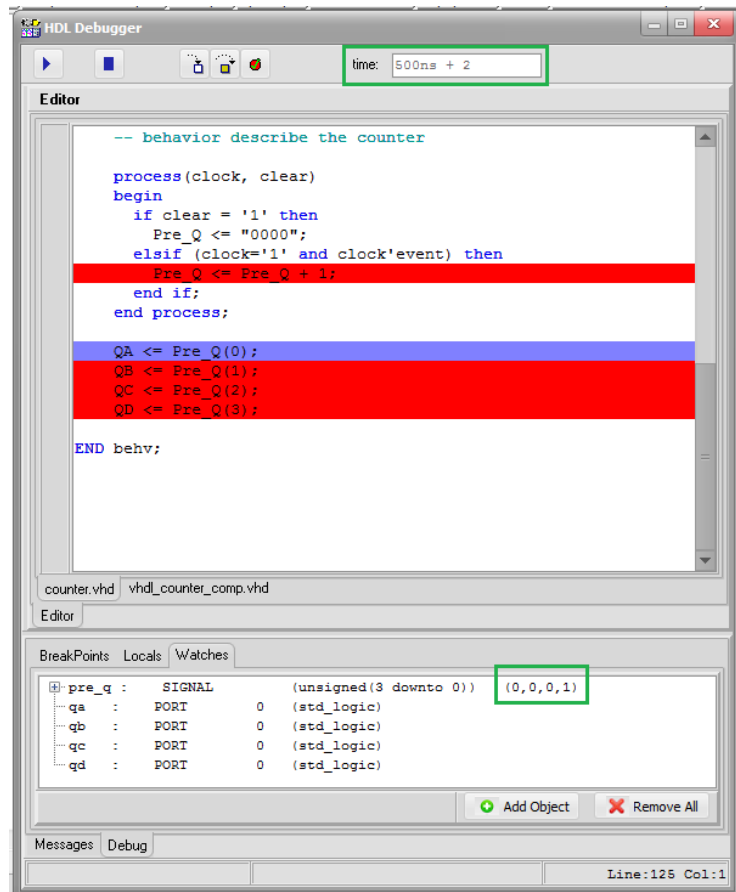


Рис. 3.24. Первая транзакция по сигналу Pre_Q

Вы можете изучить обновление других портов аналогичным образом.

Листинг программы vhd_counter.TSC

```
-- TINA VHDL Macro Description Begin
--
-- entity_name: counter;
-- arch_name: behv;
```

```
-- ports: clock,clear;QA,QB,QC,QD;
```

```
--
```

```
-- TINA VHDL Macro Description End
```

```
-----
```

```
library ieee;
```

```
use ieee.std_logic_1164.all;
```

```
use ieee.std_logic_arith.all;
```

```
-----
```

```
ENTITY counter is port(
```

```
    clock:      in std_logic;
```

```
    clear:      in std_logic;
```

```
    QA, QB, QC, QD: out std_logic);
```

```
END counter;
```

```
-----
```

```
ARCHITECTURE behv of counter is
```

```
    signal Pre_Q: unsigned( 3 downto 0 );
```

```
BEGIN
```

```
    -- behavior describe the counter
```

```
    process(clock, clear)
```



```

begin
    if clear = '1' then
        Pre_Q <= "0000";
    elsif (clock='1' and clock'event) then
        Pre_Q <= Pre_Q + 1;
    end if;
end process;

QA <= Pre_Q(0);
QB <= Pre_Q(1);
QC <= Pre_Q(2);
QD <= Pre_Q(3);

```

```
END behv;
```

3.4. Анализ цифровой схемы с использованием цифрового Verilog моделирования

Рассмотрим подобный пример в Verilog. Файлы находятся в папке Примеры \ Verilog.

TINA также включает в себя мощный цифровой механизм моделирования Verilog.

Преимущество Verilog по сравнению с VHDL в том, что его легче учить и понимать, однако в VHDL есть больше возможностей.

Verilog - аналогично VHDL - также может быть реализован автоматически в программируемых логических устройствах, такие как ПЛИС и CPLD.

TINA переводит модели Verilog и другие цифровые компоненты для синтезируемого кода VHDL вместе с соответствующим файлом UCF (User Constraints File) для назначения выводов в ПЛИС, если установлен флажок «Создать синтезируемый код» в меню «Анализ / Опции».

Вы можете сохранить созданный VHD и файлы UCF с помощью команды «Создать файл VHD & UCF» в Меню Т & М и, используя бесплатное программное обеспечение Xilinx Webpack, которое генерируют файл потока битов, описывающий реализацию проекта, и затем загрузить его в чипы Xilinx FPGA (Зайлинка FPGA).

Перед реализацией Verilog или любого другого цифрового дизайна HDL, либо с дискретными компонентами или ПЛИС, вам нужно проверить это с помощью симуляции, выбрав команду TINA Analysis | Digital Timing Analysis.

Давайте запустим предыдущую схему VHDL вместе с ее моделью Verilog. В папке примеров Verilog откроем файл Full adder Verilog and VHDL comparison time diagram.

Появится следующая схема (рис. 3.25).

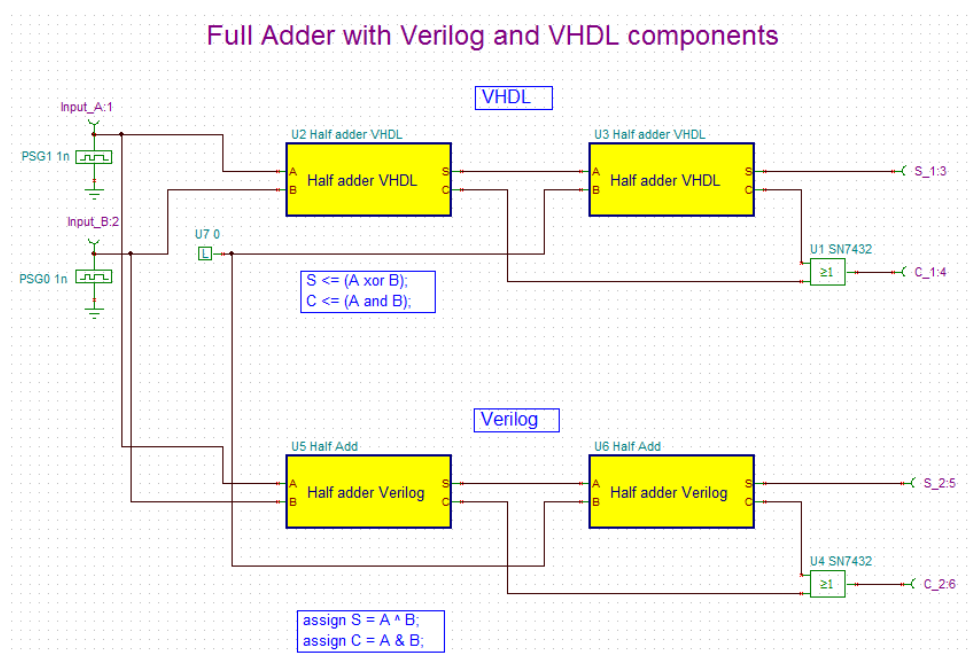
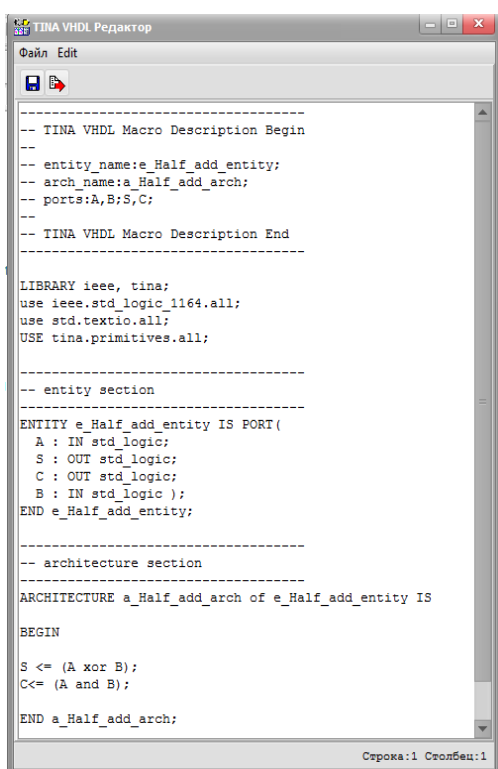


Рис. 3.25. Схема полного сумматора с компонентами Verilog и VHDL

Вы можете увидеть реализацию функции половинного сумматора на обоих языках, они очень похожи. Вы можете дважды щелкнуть VHDL или



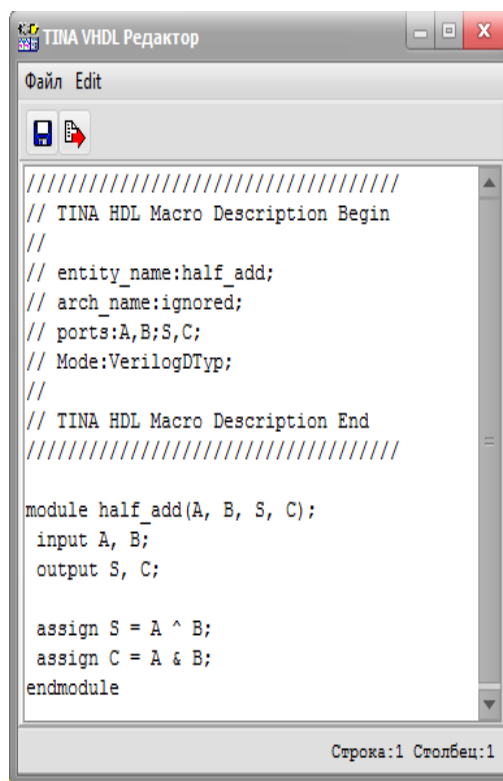
```

-- TINA VHDL Macro Description Begin
--
-- entity_name:e_Half_add_entity;
-- arch_name:a_Half_add_arch;
-- ports:A,B,S,C;
--
-- TINA VHDL Macro Description End
--
LIBRARY ieee, tina;
use ieee.std_logic_1164.all;
use std.textio.all;
USE tina.primitives.all;

-- entity section
--
ENTITY e_Half_add_entity IS PORT(
  A : IN std_logic;
  S : OUT std_logic;
  C : OUT std_logic;
  B : IN std_logic );
END e_Half_add_entity;

-- architecture section
--
ARCHITECTURE a_Half_add_arch of e_Half_add_entity IS
BEGIN
  S <= (A xor B);
  C <= (A and B);
END a_Half_add_arch;

```



```

////////////////////////////////////
// TINA HDL Macro Description Begin
//
// entity_name:half_add;
// arch_name:ignored;
// ports:A,B,S,C;
// Mode:VerilogDType;
//
// TINA HDL Macro Description End
////////////////////////////////////

module half_add(A, B, S, C);
  input A, B;
  output S, C;

  assign S = A ^ B;
  assign C = A & B;
endmodule

```

Verilog макросы и нажмите Enter Макро, чтобы увидеть все детали программ (рис. 3.26).

Verilog

VHDL

Рис. 3.26. Макросы VHDL и Verilog полусумматора

Теперь запустите Цифровой временной анализ из меню Анализ. Появится следующая диаграмма (рис. 3.27):

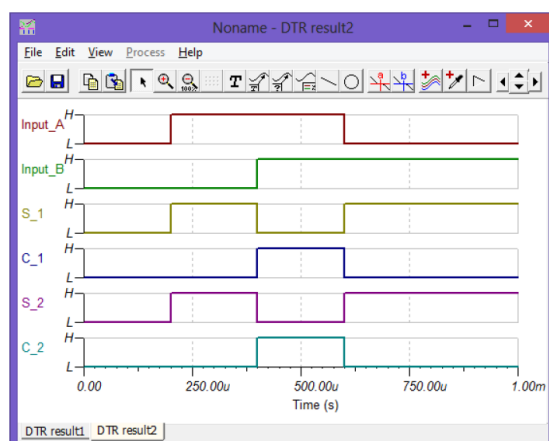


Рис. 3.27. Временные диаграммы двух моделей

Можно видеть, что выходные сигналы от обеих моделей точно совпадают.

3.5. Анализ цепей с использованием моделей Verilog-A

На сегодняшний день наиболее широко используется язык для описания электронных схем и моделей устройств Spice netlist format (1973). Тем не менее Spice описания часто трудно читать и понимать, и им не хватает многих функциональных возможностей языков программирования, которые понадобятся инженерам при создании моделей и симуляций.

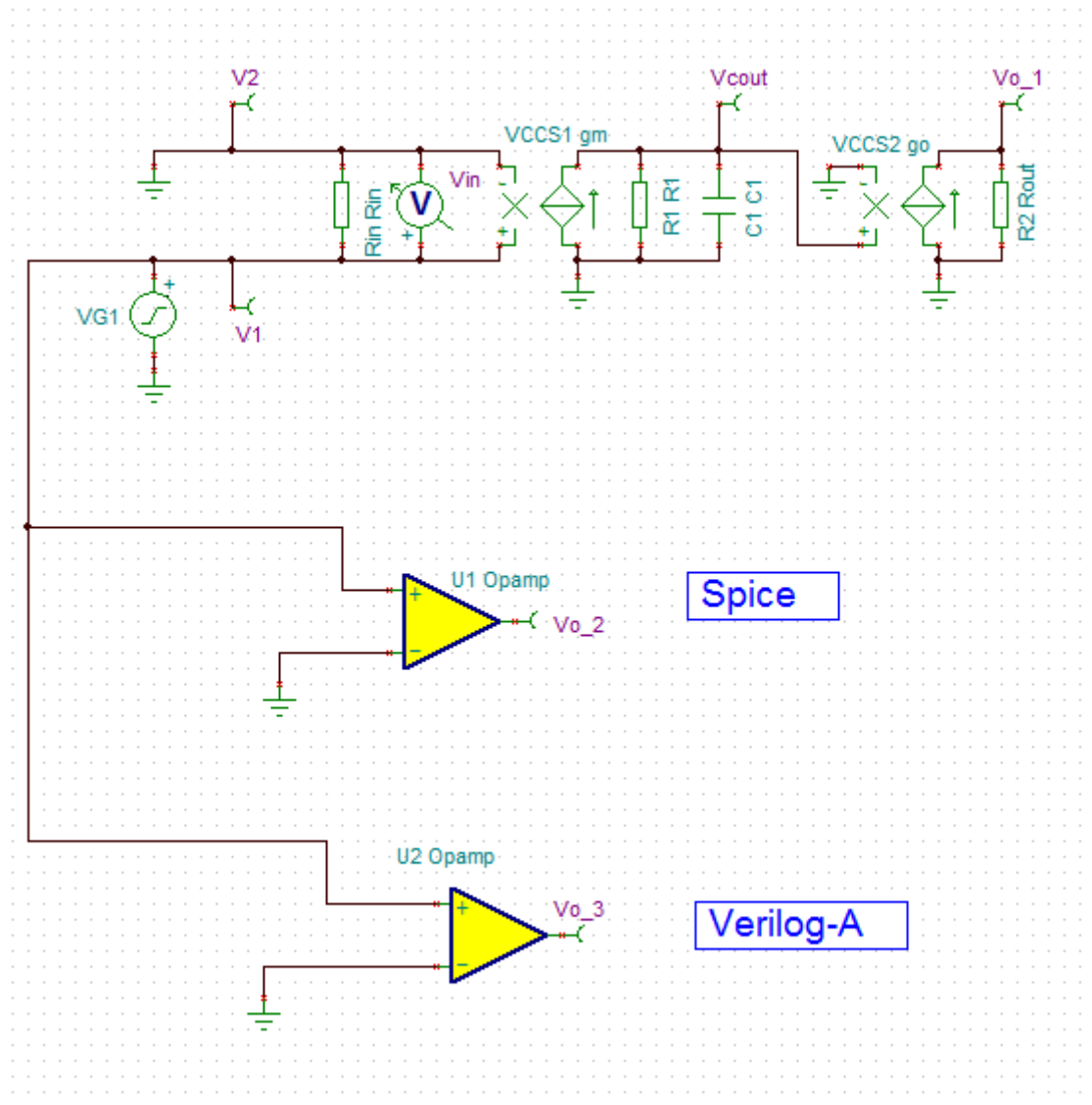


Рис. 3.28. Схема с операционными усилителями.

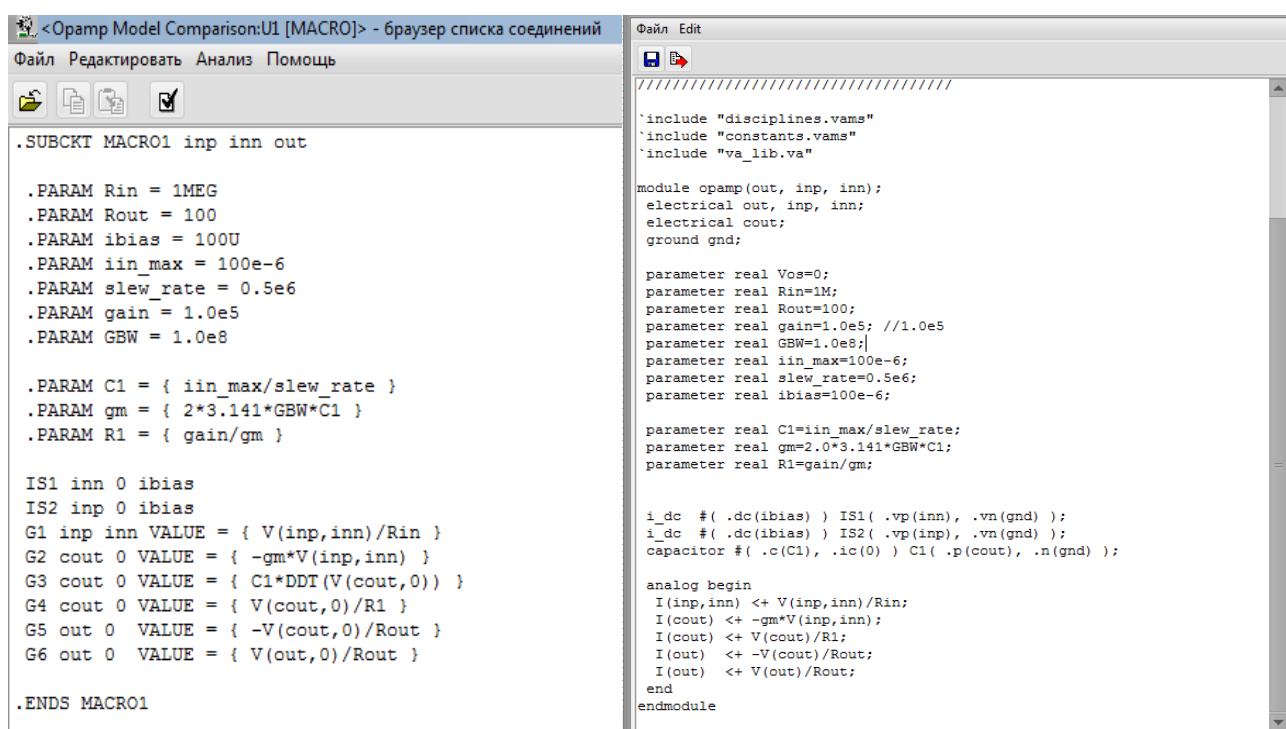
Относительно новый язык Verilog-A (1995) предлагает альтернативный метод с легко читаемым языком программирования в стиле синтаксиса C. Таким образом, Verilog-A является подходящим преемником списков SPICE для описания схемных топологий.

Большинство библиотек устройств TINA имеют формат списка соединений Spice. Однако вы уже можете создавать и импортировать модели и разместить TINA - макросы в формате Verilog. Вы можете найти несколько

примеров языка, модели устройств и схемы в папке examples \ Verilog A программы ТИНА.

Для демонстрации Verilog-A в программе ТИНА загрузите файл из папки ПРИМЕРЫ \ Verilog-A \Opamp Model Comparison.TSC, в котором простая модель операционного усилителя реализуется тремя различными способами: Verilog-A, Spice и принципиальная схема (рис. 3.28).

На рис. 3.29 показаны макросы модели Spice и Verilog-A



Макрос Spice

Макрос Verilog-A

Рис. 3.29. Макросы Spice и Verilog-A

Вы также можете изучить нелинейные модели устройств в Verilog-A и их характеристики в других примерах: DIOD.TSC, JFET.TSC и т. д.

Есть примеры в папке Examples \ Verilog-A \Language Examples, в которых вы сможете посмотреть тексты программ на языке Verilog-A.

Однако, отметим, что моделировать эти схемы можно в программе ТИНА, имеющей дополнительное расширение HDL, за которое надо платить.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

3.6. Анализ цепей с использованием моделей Verilog-AMS

Еще более сложный метод описания электронных схем, содержащих как аналоговые, так и цифровые компоненты, это язык Verilog-AMS. Как мы заметили ранее, Verilog-AMS является производным расширением от чисто цифрового Verilog и чисто аналоговым Verilog A и имеет интерфейс для подключения аналоговых и цифровых частей.

В TINA вы также можете создавать или импортировать макросы Verilog AMS для моделирование устройств со смешанным сигналом.

Давайте посмотрим на структуру такой модели. Откройте ЦАП - схема DAC VAMS.TSC из папки EXAMPLES \ Verilog AMS. Появится следующая схема (рис. 3.30).

Эта схема содержит макрос цифрового аналогового преобразователя (ЦАП) с последовательный периферийный интерфейс (SPI) и макрос тестового стенда, генерирующего цифровой сигнал SPI. Модель ЦАП определена в Verilog AMS.

Интересно, что тестовый стенд на левой стороне написан на VHDL, что является примером смешивания разных HDLs, но здесь мы сосредоточимся на макросе Verilog AMS справа.

Чтобы увидеть код Verilog AMS модели, дважды щелкните макрос ЦАП макрос и нажмите кнопку ввода макроса. Появится следующее окно (рис. 3.31).

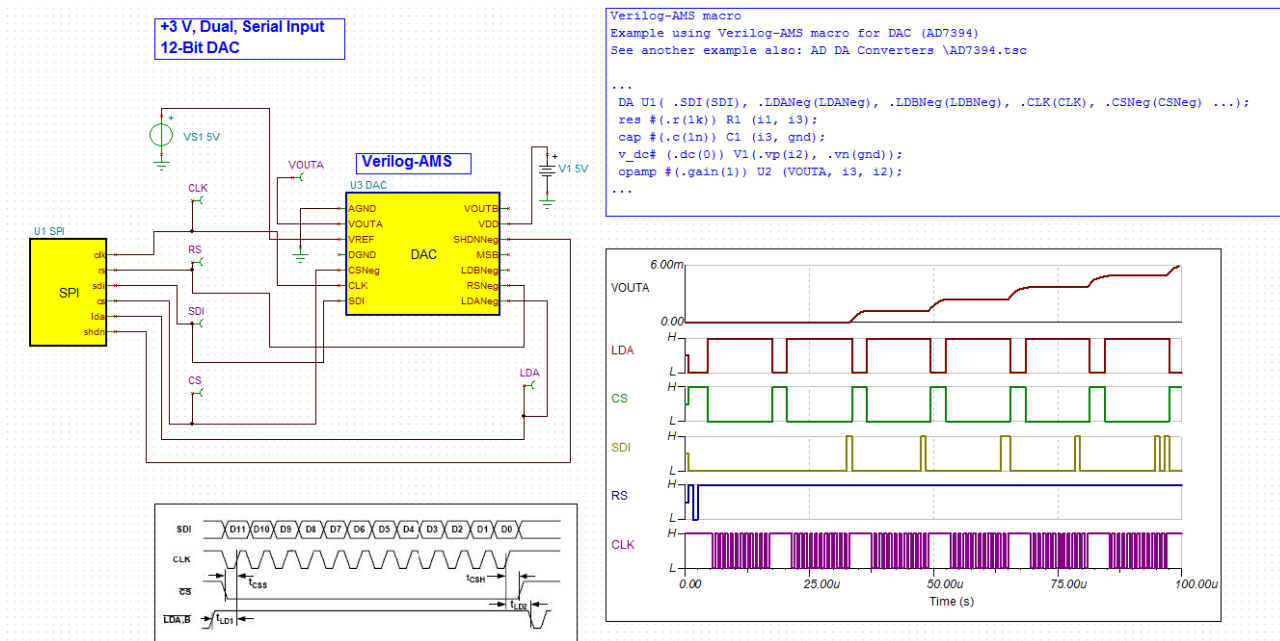


Рис. 3.30. Схема цифроаналогового преобразователя

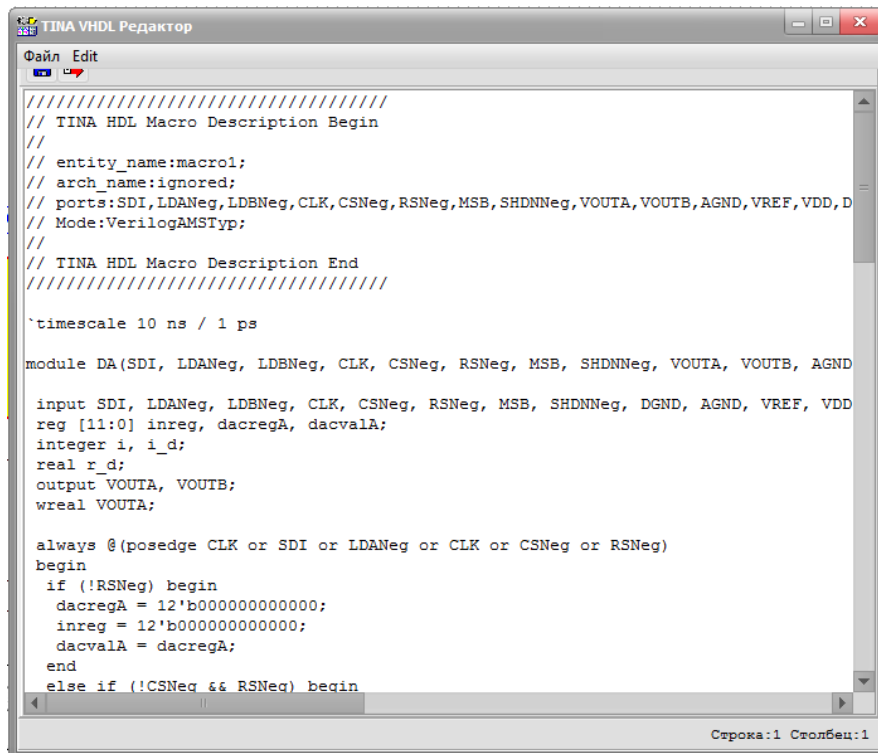
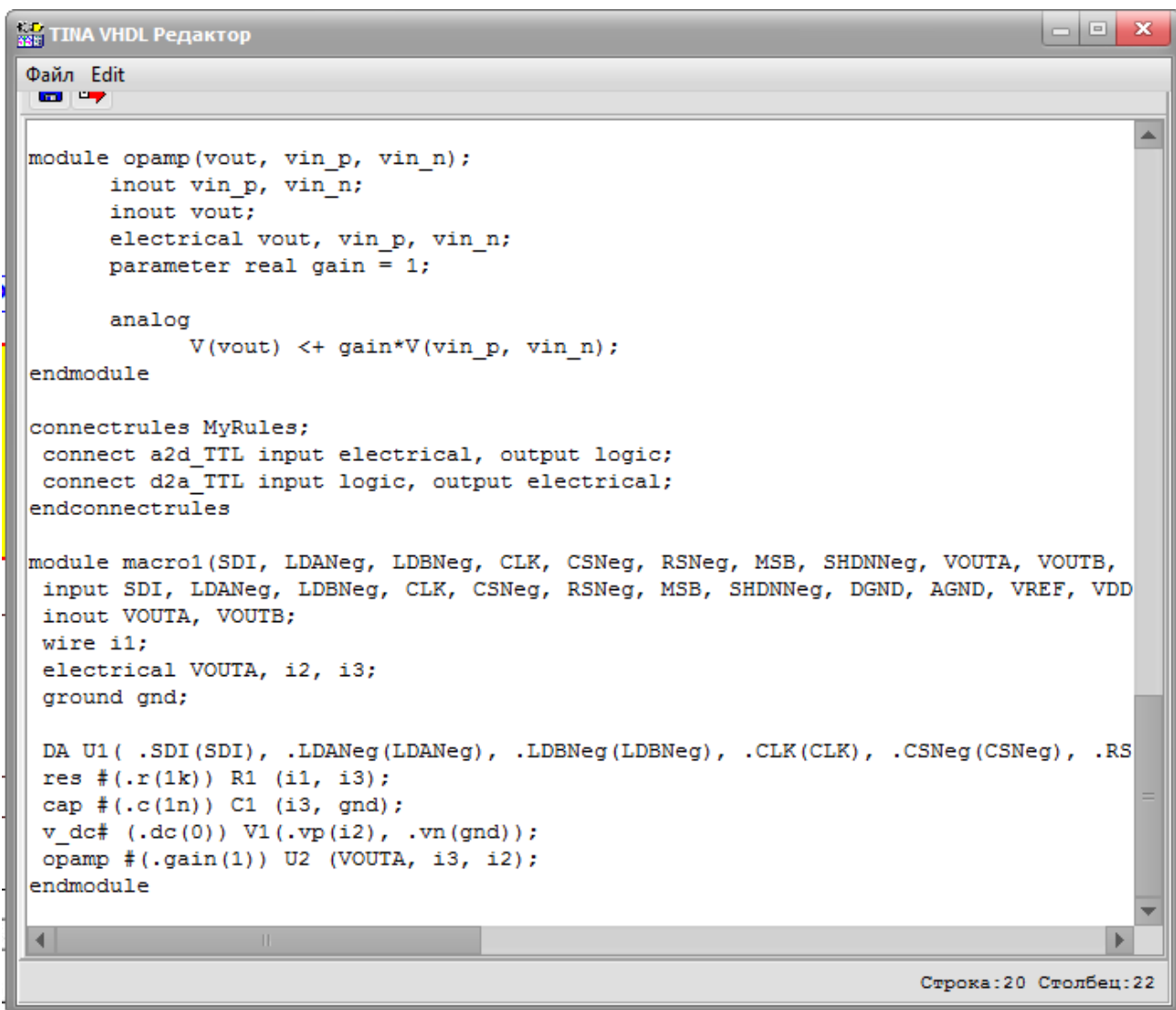


Рис. 3.31. Начало макроса ЦАП

Мы не будем вдаваться в подробный анализ кода. Мы просто хотим показать, что в первой части, показанной выше, модуль DA Verilog преобразует последовательный сигнал в аналоговый сигнал (VOUTA).

В конце макроса, показанного ниже (рис. 3.32), модуль DA вызывается и сигнал сглаживается простым операционным усилителем и RC-фильтром, используя инструкции Verilog A.



```

TINA VHDL Редактор
Файл Edit

module opamp(vout, vin_p, vin_n);
    inout vin_p, vin_n;
    inout vout;
    electrical vout, vin_p, vin_n;
    parameter real gain = 1;

    analog
        V(vout) <+ gain*V(vin_p, vin_n);
endmodule

connectrules MyRules;
    connect a2d_TTL input electrical, output logic;
    connect d2a_TTL input logic, output electrical;
endconnectrules

module macro1(SDI, LDANeg, LDBNeg, CLK, CSNeg, RSNeg, MSB, SHDNNeg, VOUTA, VOUTB,
    input SDI, LDANeg, LDBNeg, CLK, CSNeg, RSNeg, MSB, SHDNNeg, DGND, AGND, VREF, VDD
    inout VOUTA, VOUTB;
    wire i1;
    electrical VOUTA, i2, i3;
    ground gnd;

    DA U1( .SDI(SDI), .LDANeg(LDANeg), .LDBNeg(LDBNeg), .CLK(CLK), .CSNeg(CSNeg), .RS
    res #(.r(1k)) R1 (i1, i3);
    cap #(.c(1n)) C1 (i3, gnd);
    v_dc# (.dc(0)) V1(.vp(i2), .vn(gnd));
    opamp #(.gain(1)) U2 (VOUTA, i3, i2);
endmodule
    
```

Строка:20 Столбец:22

Рис. 3.32. Конец макроса ЦАП

Вы также можете увидеть описание конденсатора во нижнем фрагменте кода.

3.7. Анализ цепей с использованием SystemC

SystemC - еще один отличный инструмент для моделирования оборудования. Включает в себя все возможности C++, используемые во всем мире, и библиотеку классов C++, специально разработанную для проектирования электронных (и неэлектронных) систем на разных уровнях абстракции. SystemC имеет открытый исходный код, бесплатную реализацию, и вы можете скомпилировать его в очень эффективный исполняемый двоичный код с также бесплатным C++ компилятором Microsoft Visual Studio. В SystemC вы можете моделировать оборудование на более высоком уровне абстракции, чем в других HDL, и поэтому для моделирования некоторых очень сложных аппаратных средств, например микроконтроллеров, этот язык проще и эффективнее в использовании, чем другие HDL, такие как VHDL или Verilog.

В версии TINA 11 и более поздних версиях TINA вы также можете создавать и использовать компоненты, смоделированные в SystemC, как в TINA, так и в TINACloud.

Ниже приведены требования для использования SystemC с TINA.

3.7.1. Требования к компилятору

Используйте Microsoft Visual Studio для компиляции моделей SystemC. В нашем примере мы используем Microsoft Visual Studio 2015.

Дистрибутив SystemC

Используйте версию systemc-2.3.1 дистрибутива SystemC.

Компиляция дистрибутива SystemC (SystemC.lib)

Используйте файл проекта MSVC в дистрибутиве:

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

(<Sc_home> \ msvc80 \ SystemC).

Установите в MSVC:

- C ++ / Генерация кода: многопоточная отладочная DLL (/ MDd)
- Дополнительная опция строки cmd: / vmg
- Удалите эту строку из <sc_home> \ src \ systemc.h: используя std::gets ;
- Создайте проект, и результат будет в:

<Sc_home> \ msvc80 \ SystemC \ Debug \ SystemC.lib

3.7.2. Компиляция модели

Используйте шаблон проекта в <TINADir> \ examples \ SystemC \ systemc_model.zip (systemc \ systemc_model.vcxproj).

Откройте Visual Studio и откройте файл проекта. Откройте менеджер свойств (View/Other windows/Property manager) и выберите запись «Macros», выберите Common properties/User macros. Измените на макрос SC_HOME, в который вы распаковали systemc_model.zip файл.

Скомпилируйте конфигурацию Debug. Если вы хотите проверить свою модель на TINACloud вы должны скомпилировать дистрибутив SystemC и проект systemc_model с (/ MTd) (многопоточная отладка).

Создание макроса SystemC в TINA

Подобно другим компонентам HDL и Spice, вы должны преобразовать вашу модель SystemC в макрос TINA.

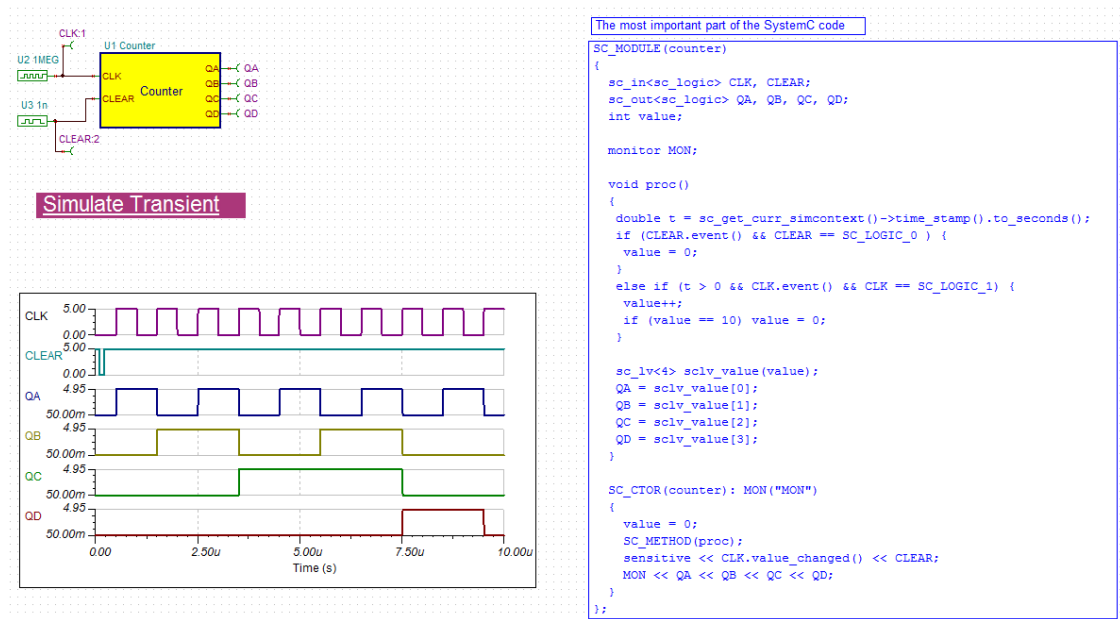


Рис. 3.33. Схема примера счётчика и исполняемый код в SystemC

Вот как это сделать.

Загрузите пример Examples/SystemC/sc_counter.tsc. На рис.3.33 показана схема счетчика и исполняемый код в SystemC.

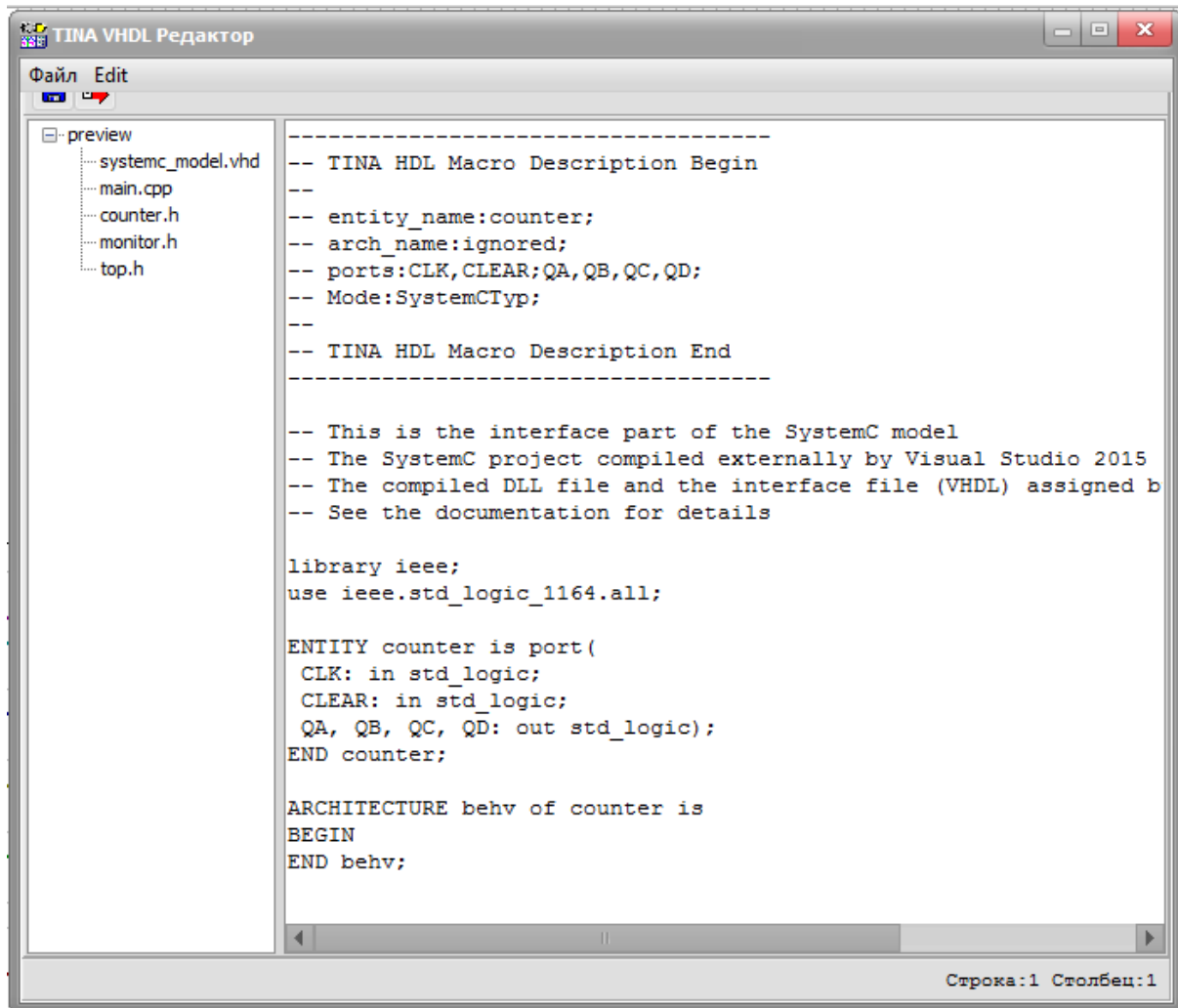


Рис. 3.34. VHDL макрос счетчика

3.7.3. Требования к моделированию

В `sc_main` используйте динамическое выделение для модуля верхнего уровня. Установите `return 0` после создания верхнего экземпляра.

```

int sc_main(int argc, char* argv[])
{
  top* TOP = new top("TOP");
  return 0;
}

```

Макрос SystemC состоит из файла VHDL и файла SystemC dll.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Файл VHDL является только файлом интерфейса. Модуль верхнего уровня SystemC должен содержать те же сигналы, что и в интерфейсе модуля VHDL и в том же порядке.

Пример схемы счетчика в SytemC:

```
SC_MODULE(top)
{
    sc_signal<sc_logic> CLK, CLEAR, QA, QB, QC, QD;
    counter U1;
    SC_CTOR(top): U1("U1")
    {
        U1.CLK(CLK); U1.CLEAR(CLEAR); U1.QA(QA);
        U1.QB(QB); U1.QC(QC); U1.QD(QD);
    }
};
```

Наиболее важная часть модели счётчика SystemC.

```
#ifndef counterH
#define counterH
#include "monitor.h"
SC_MODULE(counter)
{
    sc_in<sc_logic> CLK, CLEAR;
    sc_out<sc_logic> QA, QB, QC, QD;
    int value;
    monitor MON;
    void proc()
    {
        double t = sc_get_curr_simcontext() -
        >time_stamp().to_seconds();
```

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

```

if (CLEAR.event() && CLEAR == SC_LOGIC_0 ) {
value = 0;
}
else if (t > 0 && CLK.event() && CLK ==
SC_LOGIC_1) {
value++;
if (value == 10) value = 0;
}
sc_lv<4> sclv_value(value);
QA = sclv_value[0];
QB = sclv_value[1];
QC = sclv_value[2];
QD = sclv_value[3];
}
SC_CTOR(counter) : MON("MON")
{
value = 0;
SC_METHOD(proc);
sensitive << CLK.value_changed() << CLEAR;
MON << QA << QB << QC << QD;
}
};
#endif

```

VHDL interface file

```

library ieee;
use ieee.std_logic_1164.all;

ENTITY counter is port(

```

```

CLK: in std_logic;
CLEAR: in std_logic;
QA, QB, QC, QD: out std_logic);
END counter;
ARCHITECTURE behv of counter is
BEGIN
END behv;

```

TINA должна приостанавливать симуляцию SystemC в смешанном режиме после каждого изменение выходного порта верхнего уровня. Для этого создайте модуль монитора, вызовите наш `sc_pli_set_node_changed ()`, затем вызовите `sc_pause ()`.

Ниже приведен пример кода для счетчика.

```

#ifndef monitorH
#define monitorH
#include "systemc.h"
#include "C_SCPLI.h"
SC_MODULE(monitor)
{
  sc_in<sc_logic> QA, QB, QC, QD;
  SC_CTOR(monitor)
  {
    SC_METHOD(proc);
    sensitive << QA << QB << QC << QD;
  }
  void proc()
  {
    sc_pli_set_node_changed(true);
  }
}

```



```

sc_pause();
}
};
#endif

```

Поддерживаемые типы портов верхнего уровня:

```
sc_bit, sc_logic, bool, double.
```

3.7.4. Настройка анализа в TINA

Выберите правильный временной шаг в Анализ / Установка параметров анализа / TR максимальный шаг по времени.

Запуск модели

Вы должны установить «Visual C++ Redistributable for Visual Studio 2015», если у вас не установлена Visual Studio 2015.

Пример счетчика

Используйте шаблон проекта в <TINADir> \ examples \ SystemC \ systemc_model.zip (systemc \ systemc_model.vcxproj).

Распакуйте этот zip-файл (<sc_model>).

Откройте Visual Studio и откройте файл проекта. Откройте Менеджер свойств проекта (View/Other windows/Property manager) и выберите «Макросы». Измените макрос на SC_HOME, в который вы извлекли файл systemc_model.zip.

Скопируйте <sc_model> \ systemc_model \ examples \ counter \ top.h в <sc_model> \ systemc_model

и <sc_model>\systemc_model\Examples\counter\monitor.h в

<sc_model>\systemc_model.

Перестройте (rebuild) проект.

Скопируйте `<sc_model> \ systemc_model \ examples \ counter \ systemc_model.vhd` и `<sc_model> \ Debug \ systemc_model.dll` в каталог (например, `d: \ Temp`).

В TINA откройте `<TINADir> \ examples \ SystemC \ counter.tsc`, удалите макрос счетчика.

Теперь выберите Tools / New Macro Wizard ... Введите Counter в макрос в поле «Name», выберите «From file», нажмите значок папки. В диалоговом окне выберите файлы типа «SystemC executable» и найдите ранее скопированный `systemc_model.dll`.

Вставить новый макрос на место ранее удаленного макроса. Запустите Transient. Результат следующий (рис. 3.35).

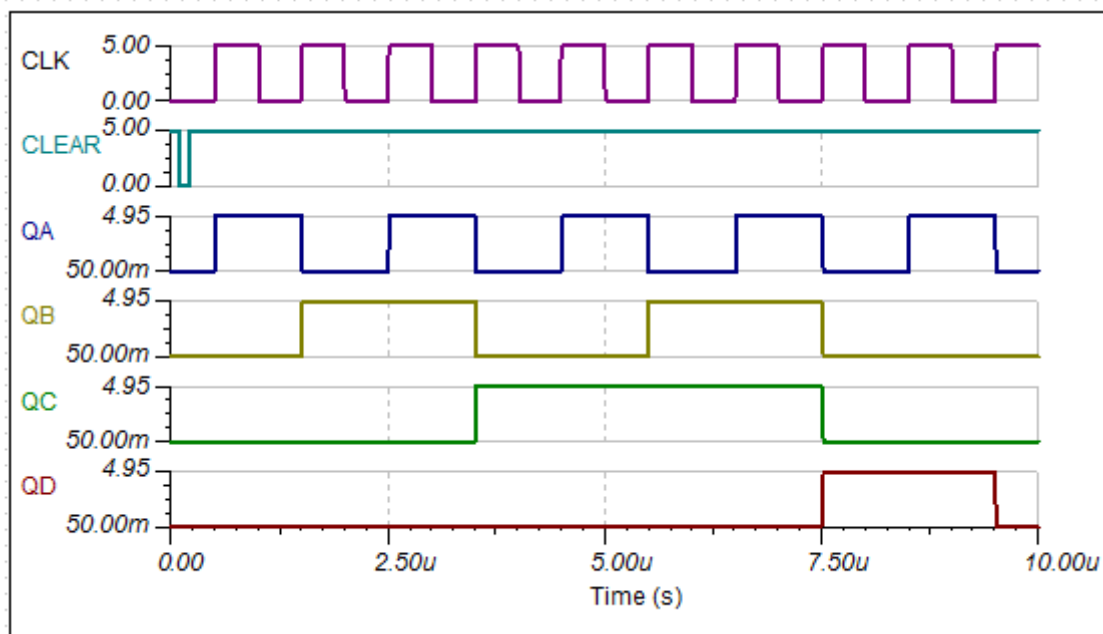


Рис. 3.35. Результат моделирования счетчика в SystemC

3.8. Схемы с микроконтроллерами (MCU)

Для проверки цепей с программируемыми устройствами требуются специальная разработка программного обеспечения, обеспечивающего высокую степень интерактивности. Это приводит к отладке программного обеспечения, которое может проверить код, пошагово работающий в устройстве.

Вы можете видеть, изменять и отлаживать программу, запущенную в любом из поддерживаемых процессоров, и, конечно же, вы можете сделать и запустить свой собственный код

Существует 4 способа предоставления программы для микроконтроллеров в ТИНА.

Вы можете:

1) использовать двоичный код и файл отладки, выполненные любым стандартным компилятором (например, MPLAB для PIC),

2) загрузить свой код сборки для запуска и отладки непосредственно в ТИНА, используя встроенный ассемблер-отладчик,

3) написать свой код MCU на C, установить компилятор C, генерирующий код для MCU, который вы хотите смоделировать, (ТИНА автоматически будет интегрировать его в отладчик кода C),

4) или, наконец, используйте встроенный редактор потоковых диаграмм в ТИНА для генерации и отладки кода MCU.

Все эти методы работы описаны в [13].

Теперь давайте запустим приложение микроконтроллера и посмотрим, как тестировать и изменить его код. Загрузите схему PICFlasher.TSC из папки Examples\ Microcontrollers \ PIC\PICFlasher.

Появляется следующая схема с микроконтроллером 16F73 PIC (рис. 3.36).

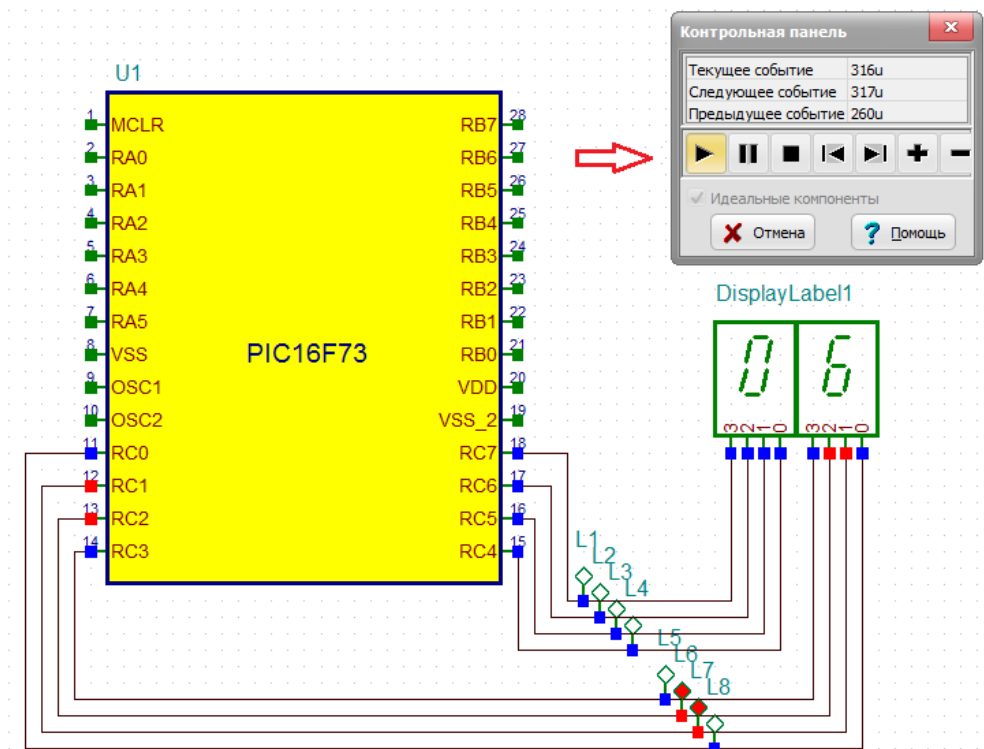




Рис. 3.36. Схема PICFlasher

Эта схема просто считает каждый раз увеличивает счетчик на единицу.

Установить режим  - «Цифровой», в меню выберите Анализ-Цифровой пошаговый и нажмите кнопку «Воспроизведение», чтобы увидеть, как это работает. Дисплей должен изменяться на единицу на каждом шаге.

Чтобы посмотреть или загрузить код в MCU, дважды щелкните на схеме и в окне свойств MCU нажмите условное обозначение  (рис. 3.37).

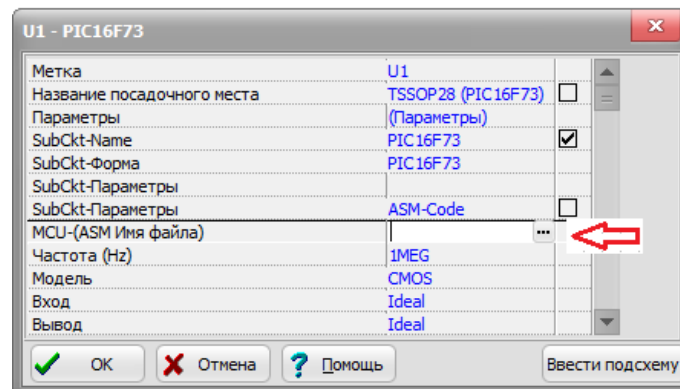


Рис. 3.37. Окно свойств микроконтроллера

Откроется новое окно Выделение входного файла MCU (рис. 3.38).

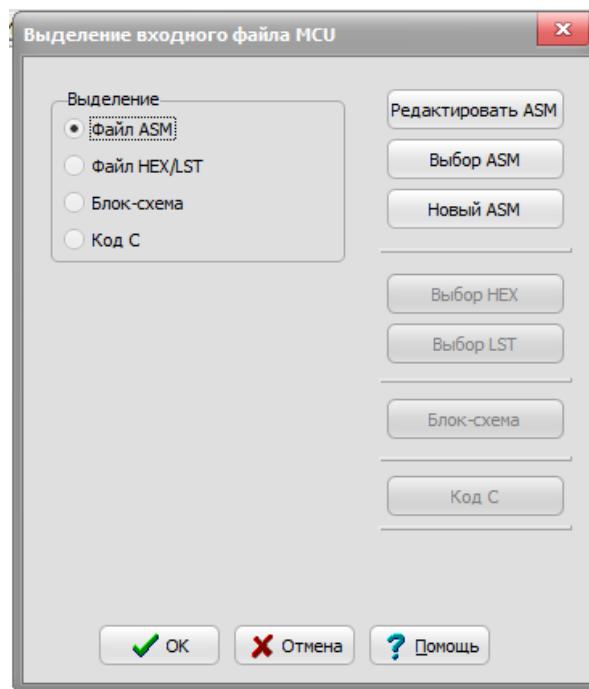


Рис. 3.38. Окно выделения входного файла

Здесь вы можете увидеть и отредактировать код ASM ассемблера в MCU, выбрать другой файл кода ASM или создайте новый ASM прямо в редакторе, который будет появляться при нажатии кнопки New ASM.

Однако, если вы переключитесь на опцию использовать Файл HEX / Lst, вы можете выбрать двоичный (HEX) файл, который вы хотите запустить, и

файл LST, который вы хотите использовать для отладки, как показано в диалоговом окне ниже.


Файлы HEX и LST должны быть сгенерированы соответствующим компилятором (обычно предоставляется производителем MCU бесплатно). Однако TINA имеет встроенный компилятор для всех поддерживаемых MCU, так что вы можете напрямую использовать свой исходный код ASM.

3.8.1. Отладка кода ASM

Выберем файл ассемблера ASM и нажмем «Редактировать ASM».

В новом окне Редактора исходного кода мы увидим программный код в ассемблере (рис. 3.39.).

Теперь давайте внесём следующие изменения в код. Измените инструкцию в строке 25 (вы можете увидеть номер строки в правом нижнем углу окна редактора кода) из `addlw 01H` в `addlw 02H`

Сохраните изменённый код, нажав на значок , и закройте открытое Окно MCU. Если вы теперь выполните запуск режима Цифровой пошаговый, приращение на каждом шаге будет равно 2.

Обратите внимание, что изменённый код будет автоматически сохранен в файле TINA.TSC.

Важное замечание: большим достоинством программы TINA является то, что все данные о проекте хранятся в одном файле *TSC.

3.8.2. Пример обработки прерываний PIC

Теперь давайте посмотрим на другое приложение с большей интерактивностью. Загрузим пример PIC16F84_interrupt_rb4_rb7.TSC из папки Examples\ Microcontrollers \ PIC.

```

processor 16f73      ;Set the processor
radix hex           ;Set the radix
#include <p16f73.inc> ;Include header file

title "flash" ; Program title   June 2002

TEMP1      equ      20H
TEMP2      equ      21H

port      equ      PORTC
tris_port equ      TRISC

;

org 00H
main_start
    clrf      port
    bsf       STATUS, 5      ;bank 1
    clrf      tris_port      ;set port to o/p
    movlw     080H
    movwf     OPTION_REG
    bcf       STATUS, 5      ;bank 0

Loop
    movf      port, 0
    addlw     01H
    movwf     port

    movlw     001H
    movwf     TEMP1
    movwf     TEMP2

delay
    decfsz    TEMP1, F
    goto     delay
    movlw     001H
    movwf     TEMP1
    decfsz    TEMP2, F
    goto     delay
    goto     Loop

end

```

Строка:25 Столбец:13

Рис. 3.39. Исходный код в ассемблере

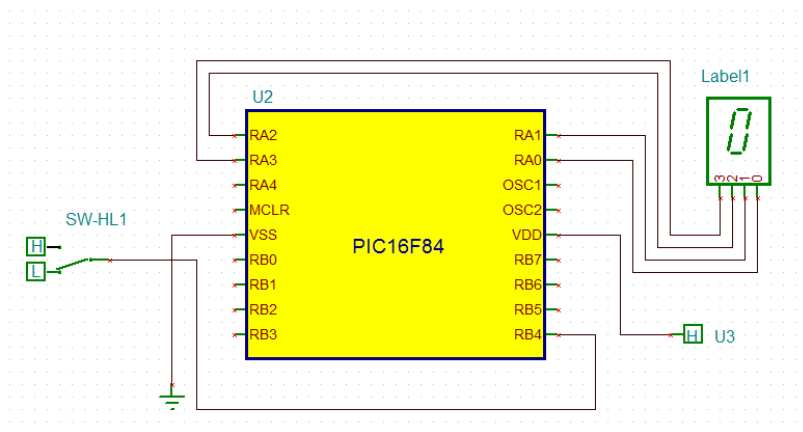


Рис. 3.40. Работа микроконтроллера с прерываниями

Нажми на кнопку. На первый взгляд кажется, что ничего происходит.

Однако, если вы нажмете на переключатель SW-HL1, на дисплее будет появляться 1 (4,95 В) каждый раз, когда переключатель меняется с низкого на высокий.

Это реализуется с помощью функции обработки прерываний PIC16F84.

Теперь давайте посмотрим на операцию более подробно с помощью интерактивного ASM отладчика.

Чтобы активировать отладчик, выберите меню «Анализ» и установите флажок «Включить отладчик кода MCU», как показано ниже в диалоговом окне параметров анализа (рис. 3.41).

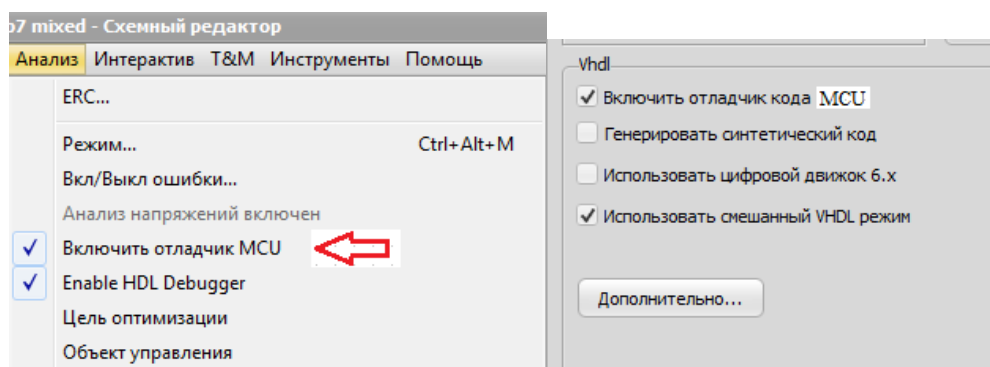


Рис. 3.41. Включение отладчика MCU в меню Анализ и Опции

Устанавливаем режим Цифровой и в меню Интерактив нажимаем Старт.

Открывается диалоговое окно отладчика MCU (рис. 3.42).

Вот краткое описание диалога отладчика MCU.

В верхней строке находятся следующие значки управления:



- Переключить точку останова: вставляет или удаляет точки останова в выбранная строка. Нажмите на строку, где вы хотите разместить или удалить точку останова до нажатия на значок.



- Запустить код в отладчике непрерывно. Линии, являющиеся выполняемыми будет выделены, а код будет прокручиваться.



- Шаг вперед. Пошаговое исполнение. Каждый раз, когда вы нажимаете эту кнопку, выполняется одна команда программы.



- Стоп. Останавливает выполнение программы.

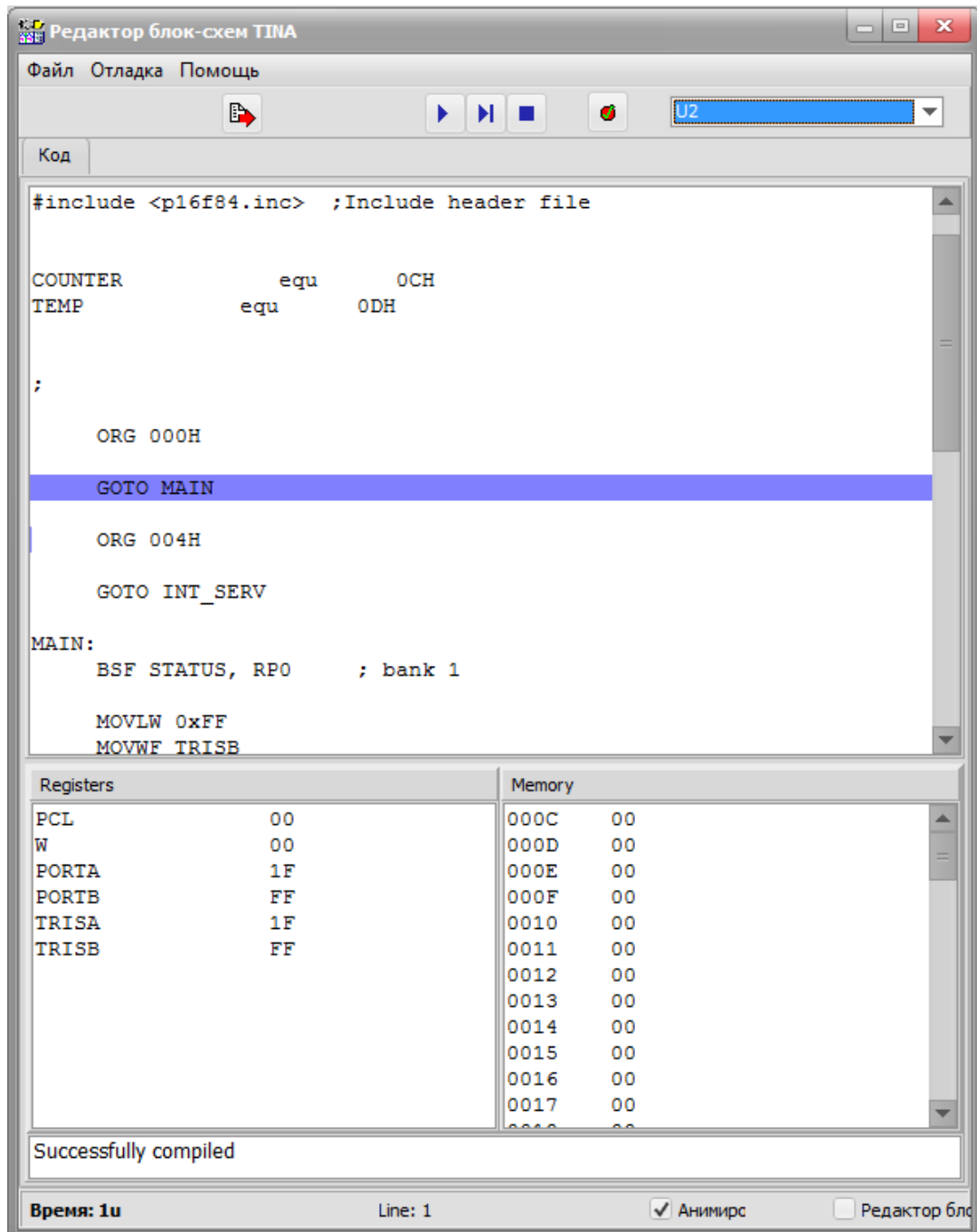


Рис. 3.42. Диалоговое окно отладчика MCU

В окне «Код» (под значками управления) отображается код ASM.

Следующая актуальная команда выделена синим цветом.

Фактическое содержание регистров и ячеек памяти MCU показаны в нижней части экрана.

Давайте шаг за шагом проследим за выполнением программы, нажав кнопка Шаг вперед. Приблизительно после 14 щелчков мы приходим к метке PT1: где программа, кажется, находится в бесконечном цикле.

```
PT1: INCF TEMP, F
      GOTO PT1
```

Теперь нажмите на переключатель SW-HL1 и измените его на Высокий уровень. (Вам следует щелкнуть, когда курсор изменится на стрелку, направленную вверх).

Вернитесь к отладчику и дважды нажмите кнопку «Шаг вперед».

Программа распознает прерывание и перейдет в обслуживание прерываний:

```
INT_SERV: label.
INT_SERV:
INCF COUNTER, F
MOVF COUNTER, 0
MOVWF PORTA
```

Программа увеличит СЧЕТЧИК и скопирует его в ПОРТ А. На выходе теперь будет 1. После этого программа вернется в «бесконечный цикл» на метке PT1.

Переключение SW-HL1 на низкий уровень снова вызовет прерывание и изменение значения на выходе.

3.8.3. Редактирование кода ASM в отладчике

Теперь давайте посмотрим, как сделать небольшое изменение в программе, используя отладчик. Дублируйте оператор `INCF COUNTER, F`, используя `Сору` и вставьте вот так:

```
INT_SERV:
INCF COUNTER, F
INCF COUNTER, F
MOVF COUNTER, 0
MOVWF PORTA
```

Теперь, если вы нажмёте , программа спросит:

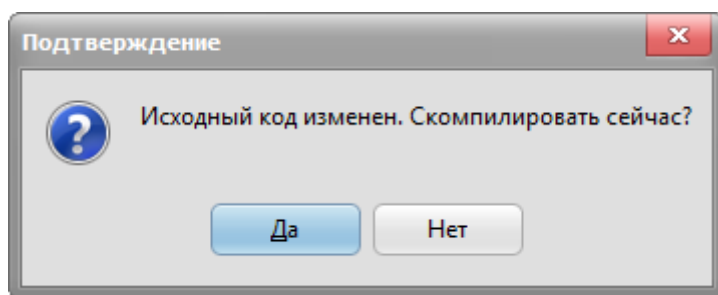






Рис. 3.43. Запрос на компиляцию кода

Нажмите «Да», а затем нажмите кнопку  ещё раз. Теперь прирост значения на выходе будет 2 при каждой смене переключателя.

Вы также можете проверить схему в режиме непрерывной работы отладчика, вызвав режим нажатием кнопки . Даже если отладчик будет работать быстро, вы все еще можете увидеть «бесконечный цикл» и прыжок к серверной процедуре прерывания (`INT_SERV:`) при смене коммутатора.

3.8.4. Создание точки останова в ASM



Часто по существу невозможно добраться до определенного места в программе, так как вам придется сделать один шаг тысячу раз (чтобы найти, когда программа приходит туда впервые). Чтобы заставить программу запустить конкретный оператор и остановиться там, вы можете пометить оператор так называемой точкой останова. Теперь запустите программу в отладчике непрерывный режим с помощью команды  «Выполнить», и программа остановится на отмеченном месте перед выполнением отмеченной команды.

Чтобы продемонстрировать это, выделите инструкцию увеличения в нашем примере после метки INT_SERV: подпрограмма обработки прерываний и нажмите кнопку  останова.

Теперь нажмите кнопку Run. Программа начинает работать и попадает в «бесконечный цикл».

Даже если вы установили точку останова, код не остановится с оно не проходит точку останова. Тем не менее, когда вы меняете переключатель от низкого до высокого, программа остановится на команде

```
INT_SERV:
    INCF COUNTER, F
```

Теперь вы можете возобновить выполнение либо шаг за шагом , либо снова с командой  Run.

3.8.5. Программирование микроконтроллеров с использованием C

Писать ассемблерный код непросто и программисты настольных компьютеров обратились к языкам программирования высокого уровня.

Языки высокого уровня становятся все более популярными, и, возможно, C - самый используемый и самый полезный язык для программирования MCU.

На рынке доступно много компиляторов Си, многие из них бесплатны или имеют бесплатную версию. Вы должны установить компилятор C, который генерирует код для MCU в вашей модели, и тогда TINA автоматически интегрирует его в свой отладчик C-кода.

Вот компиляторы C, совместимые с TINA:

(1) Для PIC: установите компиляторы HI-TECH PIC с

<http://www.htsoft.com/>

Установите его в облегченном режиме Lite. Выберите «Add to environment path»/

(2) Для PIC18: установите компилятор MPLAB C18 с

<http://www.microchip.com>.

(3) Для AVR: установите WinAVR с

<http://winavr.sourceforge.net/>

(4) Для 8051: установить SDCC и GPUTILS SDCC:

<http://sdcc.sourceforge.net/>,

GPUTILS: <http://gputils.sourceforge.net/>

(5) Для ARM: установите пакеты yagarto-tools- *, yagarto-bu- * из

<http://www.yagarto.de>

Не устанавливайте набор инструментов AVR, ARM в каталог, который содержит области типа «C: / Program Files /»

Возможно, вам придется перезагрузить компьютер после установки этих инструментов.

Вы также можете отлаживать C-код и выполнять пошаговые инструкции для большинства микроконтроллеров в TINA. Однако это не работает для 8051 и PIC процессоров старше PIC 18.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Теперь давайте посмотрим, как загрузить и запустить С-код в MCU в TINA.

Мы предположим, что компилятор Winavr уже установлен на вашем компьютере.

Создайте новый файл схемы в TINA с помощью File / New

Найдите и разместите ATTiny26 MCU, используя инструмент Find component в правом верхнем углу экрана. Вы также можете сделать то же самое с помощью вкладки MCU логических микросхем панели инструментов компонента и выбора MCU, затем AVR в качестве производителя.

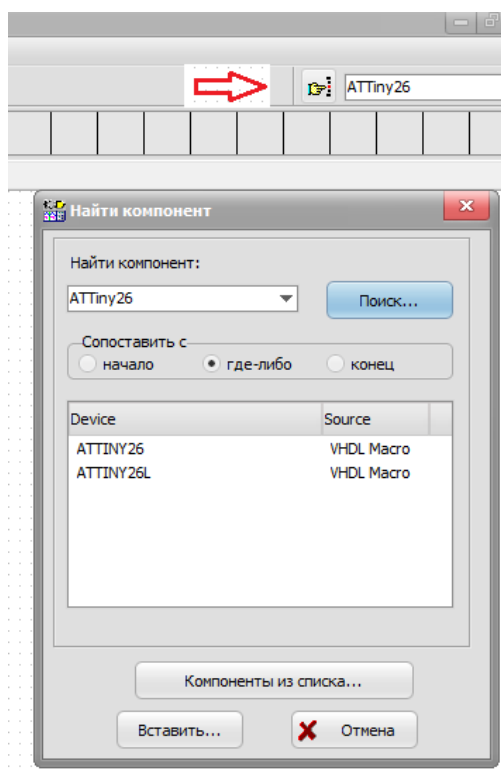



Рис. 3.44. Поиск микроконтроллера

Дважды щёлкните MCU, и появляется окно свойств MCU. Нажмите кнопку , затем на строку «код С», и затем в окне «Выделение входного файла» нажмите на кнопку «Код С» (рис. 3.45).

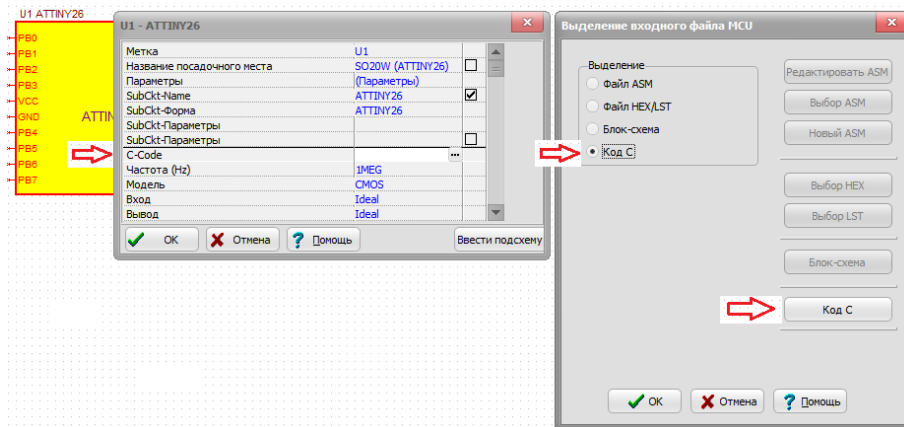


Рис. 3.45. Выделение входного файла кода C.

Вы увидите следующее диалоговое окно (рис. 3.46) .

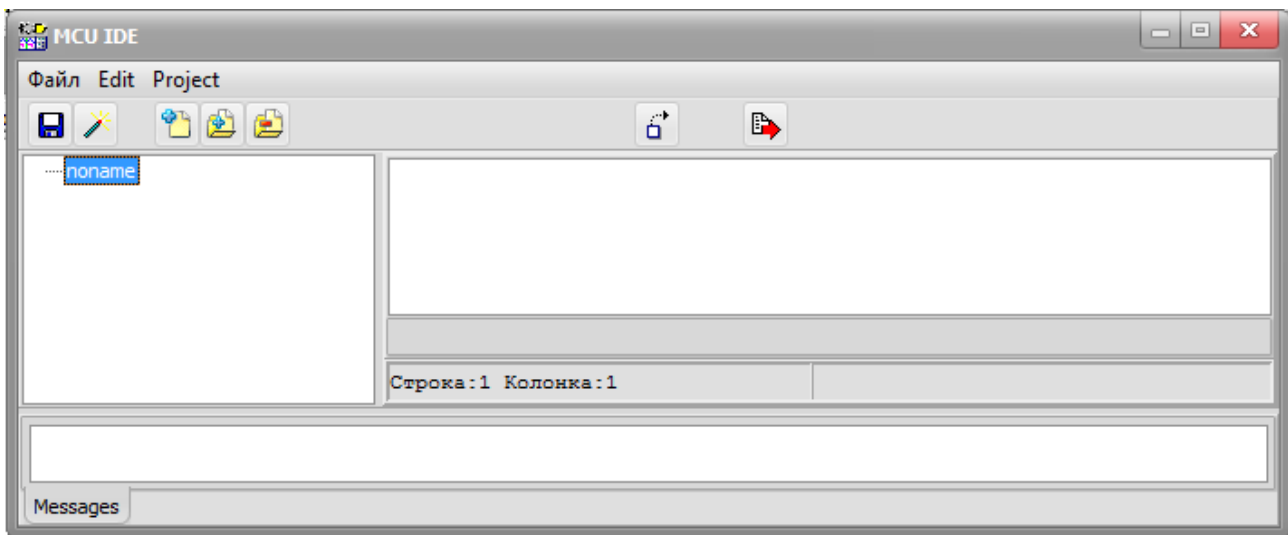


Рис. 3.46. Окно ввода кода C.

Щелкните правой кнопкой мыши синюю метку noname в верхнем левом углу экрана.

Появится всплывающее диалоговое окно, нажмите «Добавить существующий файл».

Откроется диалоговое окно Открыть: перейдите к папке Examples\MicrocontrollerПРИМЕРЫ \C Compiler \AVR (или где находится ваш

собственный файл) и откройте AVRflasher.C. Появится следующее диалоговое окно (рис. 3.47):

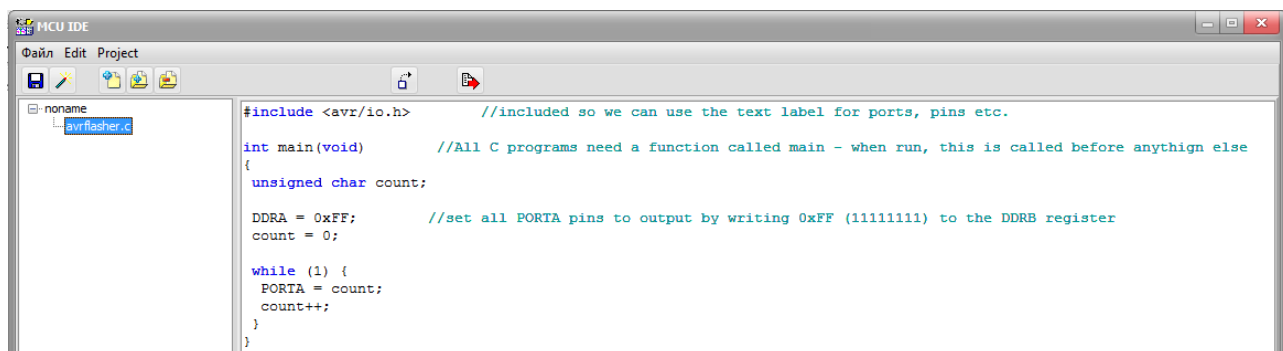






Рис. 3.47. Диалоговое окно AVR с кодом C

Нажмите кнопку  «Make project», а затем кнопку  « Save project » и выйти, нажав  в верхнем правом углу диалогового окна.

Нажмите ОК в диалоговом окне выбора входного файла MCU и в диалоге свойств ATTINY26.

Выберите Цифровой переходной анализ в Интерактивном меню или кнопку режима DIG.

В меню «Интерактив» нажмите «Старт». Теперь, если вы нажмете кнопку  в окне Редактора блок – схем TINA, ваш код C начнет работать. Вы можете видеть, что по изменению красного и синего логических состояний на выводах из MCU. Обратите внимание, что моделирование микроконтроллеров в TINA работает даже без блока питания, чтобы упростить схемы, но, конечно, вы все равно должны подключить источник питания и все необходимые контакты, когда вы разрабатываете печатную плату.

Теперь добавьте источник питания, вывод напряжения и светодиод в схему как показано на рис. 3.48.

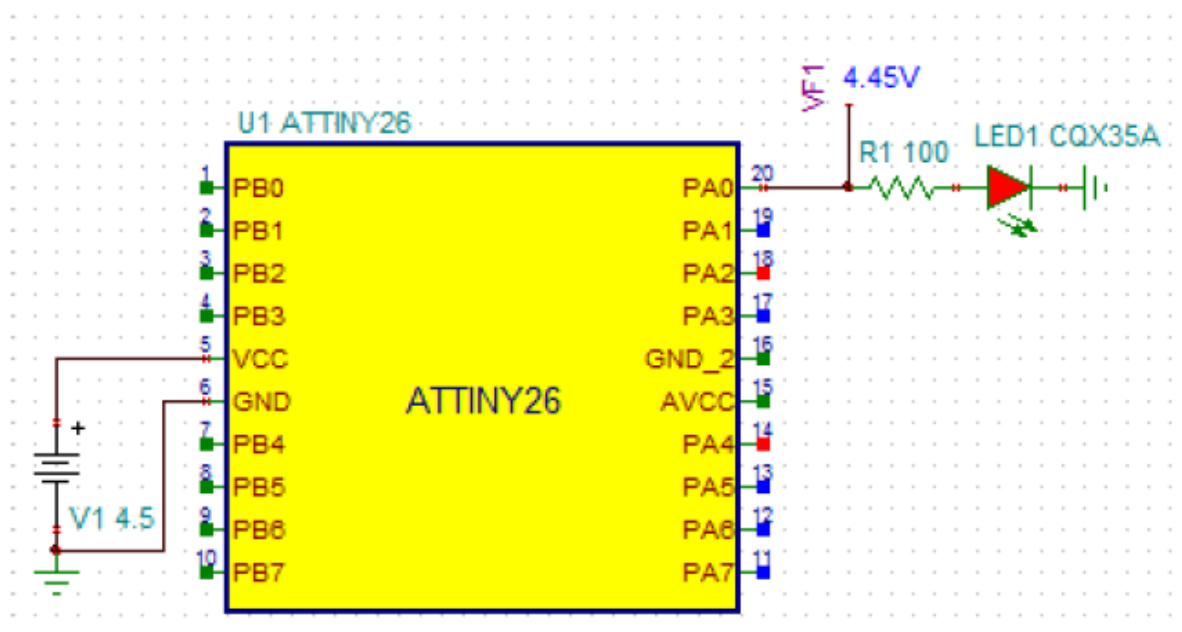


Рис. 3.48. Мигалка на микроконтроллере

Если вы запустите симуляцию, светодиод загорится, когда логический уровень на PA0 будет высокий, и на выводе напряжения будет отображаться аналоговое напряжение. Заметьте, что это напряжение соответствует напряжению источника питания. Другие контакты заполняются логическими значениями в соответствии с правилами Mixed Mode Simulation. В этом большое различие по сравнению с внешним моделированием микроконтроллеров по сравнению с TINA. Вы также можете увидеть все напряжения и отображение вашей полной схемы во время моделирования.

Вы можете сделать это даже шаг за шагом с помощью C - отладчика TINA

3.8.6. Отладка кода C в MCU

Как и с ASM и HEX кодом, вы можете следить за выполнением программой кода C и даже следите за значениями необходимых переменных.

Чтобы продемонстрировать это, давайте откроем файл «check_prime.TSC» (Проверить простое число) в папке Microcontrollers\C compiler\AVR. Появится следующая схема (рис. 3.49).

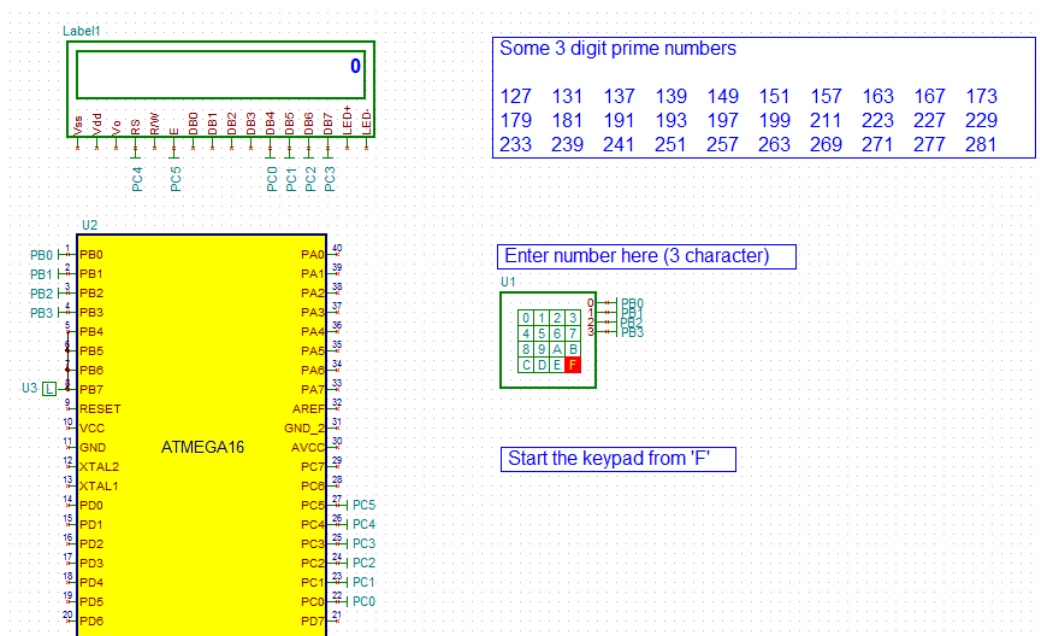


Рис. 3.49. Схема проверки простых чисел

Чтобы проверить эту цепь, нажмите кнопку DIG и введите трехзначный номер (каждая цифра должна быть разной). На дисплее появится «Prime number» или «Not prime».

Теперь, чтобы отладить эту программу на C, отпустите кнопку DIG и затем нажмите «Включить отладчик кода MCU» в меню «Анализ», а затем нажмите кнопку DIG еще раз.

В меню «Интерактив» нажмите «Старт». Откроется окно отладчика кода C (MCU IDE) (рис. 3.50).

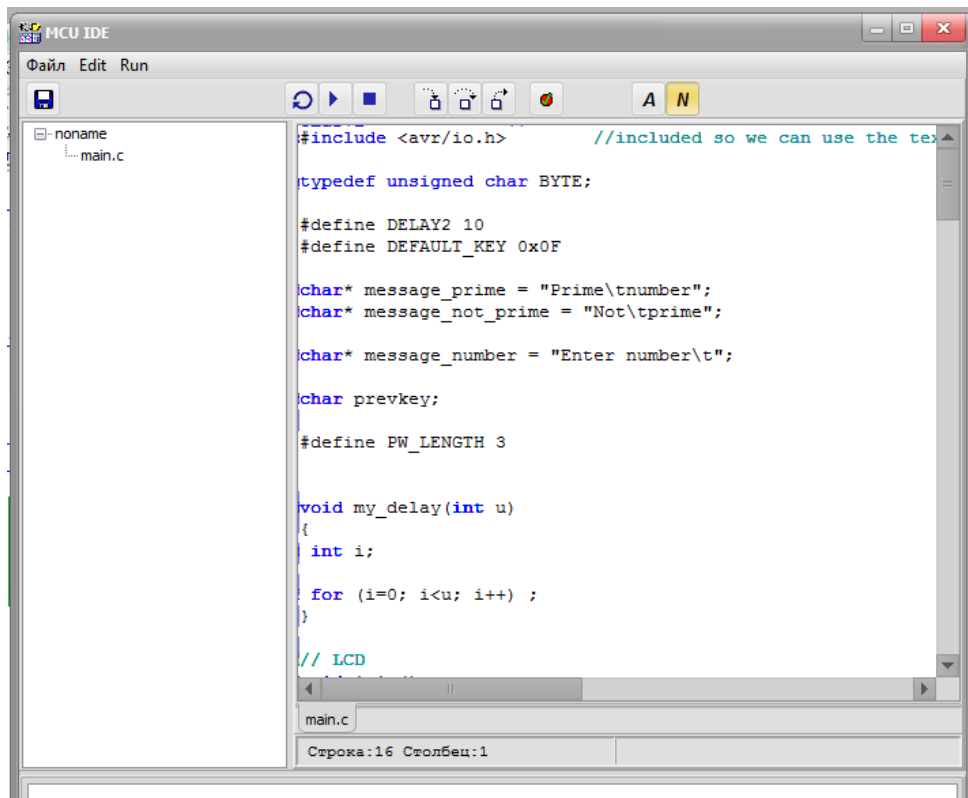


Рис. 1.41. Окно отладчика кода С

Прокрутите код С, пока не появится следующая функция (строка 128):

```

int check_prime(int a)
{
    int c,mod;
    for ( c = 2 ; c <= a - 1 ; c++ )
    {
        mod = a%c;
        if ( mod == 0 )
            return 0;
    }
    if ( c == a )
        return 1;
}

```

```
}

```

Щелкните правой кнопкой мыши переменную «с» и нажмите « Add Watch at cursor ». Нажмите вкладка «Watch List» в нижней части окна IDE MCU.

Переменная «с» должна быть уже в списке. Добавьте так же переменную «а».

Теперь нажмите на строку `if (c == a)` и нажмите Toggle на кнопке точки останова (Add Breakpoint) в верхней части окна MCU IDE. Линия с точкой останова станет красной.

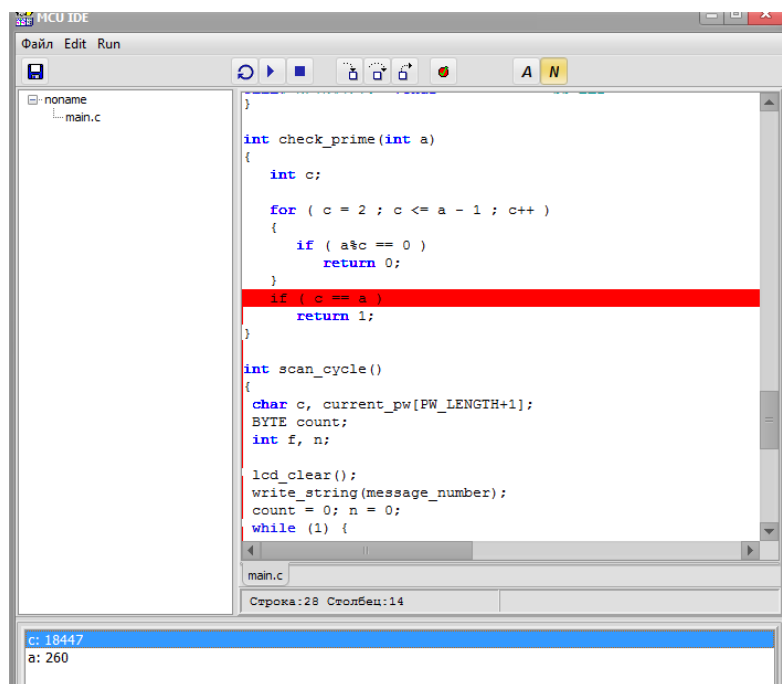


Рис. 3.51. Точка останова в отладчике

Нажмите кнопку  Start.

Программа начнет работать. Вы не будете видеть изменения в окне отладчика, если только вы не нажмете кнопка «А» в верхней части экрана, которая покажет активные инструкции и прокручивайте экран при необходимости. Однако, если Вы вводите простое трехзначное число

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

на маленькой клавиатуре на экране (например, 239), программа остановится на линии `if (c==a)` и вы увидите `a`, с переменные в точке останова (рис. 3.52).

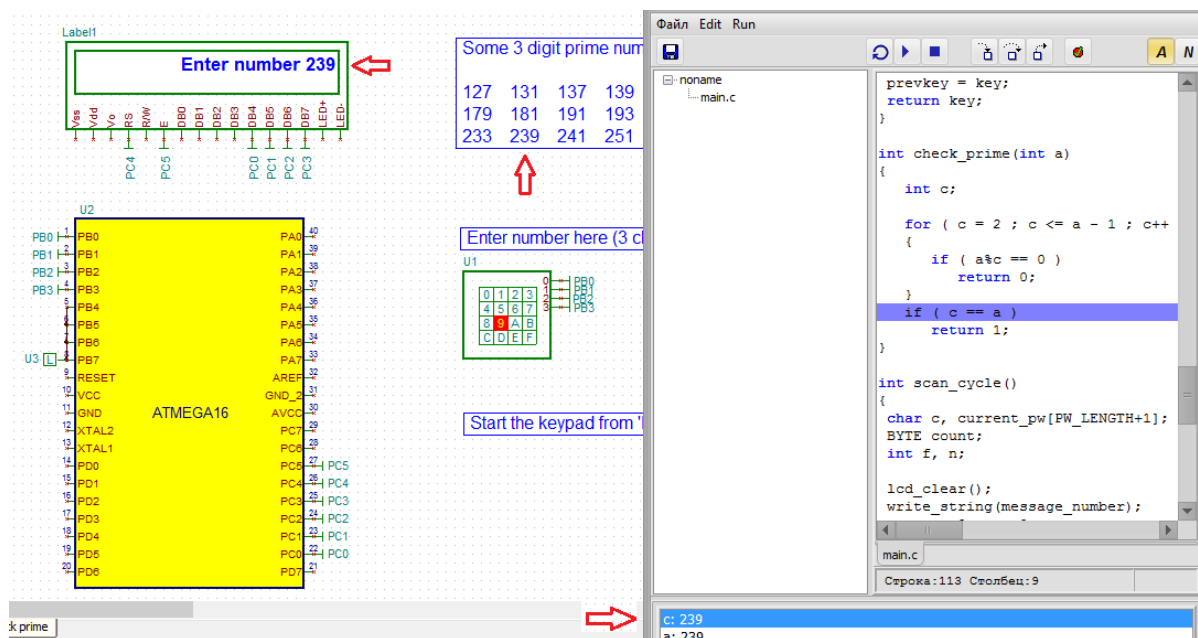


Рис. 3.52. Проверка простого числа 239

После остановки на точке останова вы можете продолжить выполнение

пошагово, нажимая кнопку Step  для каждого шага, или сделать запуск непрерывным, нажав снова кнопку Пуск ..

3.8.7. Использование редактора блок-схем и отладчика в ТИНА

Написание ассемблерного кода MCU часто является сложной и утомительной задачей. Вы можете упростить разработку программного обеспечения и получить больше времени для разработки электронного оборудования, если вместо ручного кодирования вы используете в TINA Редактор блок-схем и отладчик для генерации и отладки кода MCU.

Этот простой в использовании инструмент работает с символами и потоками управления инструкциями, с которыми вы можете представить

алгоритм своей программы. Редактор блок-схем открывается через устройство MCU, как описано ниже. Вы можете найти подробные описания условных обозначений и их параметров в меню «Справка» редактора потоковой диаграммы.

3.8.8. Редактор блок-схем

В следующем примере мы создадим блок-схему для управления MCU, встроенного в простую схему. Блок-схема добавляет два числа, которые считываются с двух портов микроконтроллера PIC16F73. (Вы можете найти полную схему в EXAMPLES\Microcontrollers\PIC\PIC Adder.TSC в TINA.

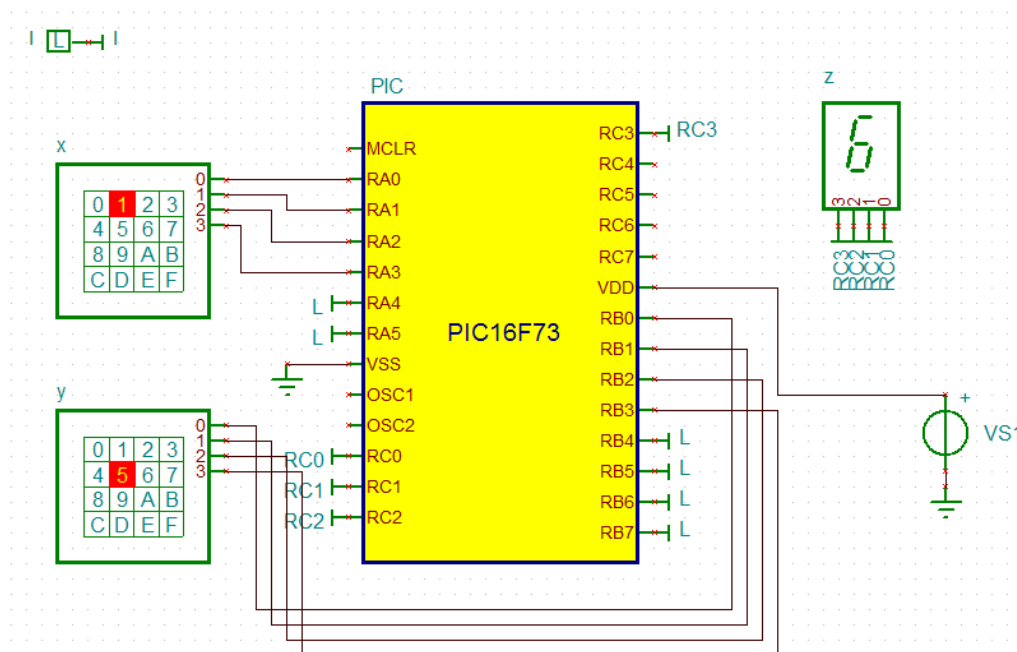



Рис. 3.53. Схема сумматора на микроконтроллере

Если вы собираете схему сами, то сначала выберите MCU на панели инструментов компонента и вставьте его в схематический редактор. Компоненты MCU расположены под вкладкой «Логические схемы-MCU».

Нажмите значок MCU  на панели инструментов и выберите PIC16 в строке

«Изготовитель» диалогового окна MCU. Появится список PIC16 MCU. Выберите PIC16F73 и нажмите ОК. Выбранный MCU будет размещен в редакторе схем.

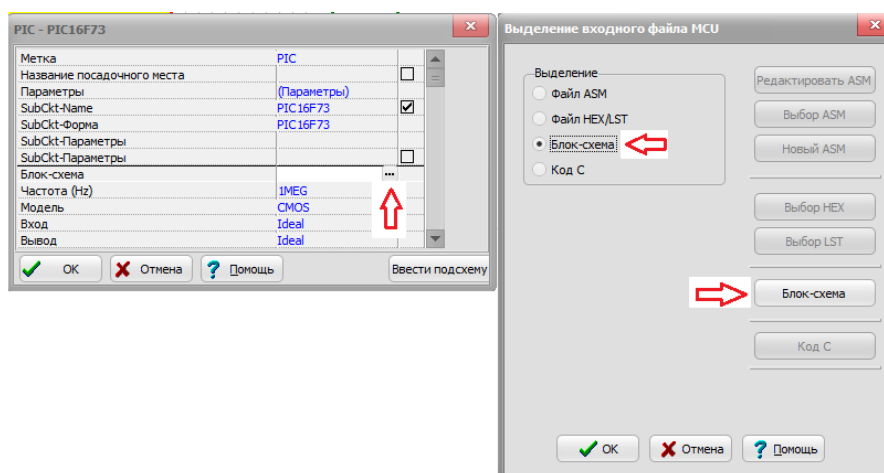



Рис. 3.54. Выбор Блок- схемы как входного файла

Теперь дважды щелкните компонент PIC16 в редакторе, щелкните в поле MCU-(ASM File Name), а затем символ . Откроется диалоговое окно выбора входного файла (рис. 3.54). Выделите «Блок – схема» и нажмите кнопку «Блок – схема».

В нашем примере блок-схема уже готова и показана на рис. 3.55. Слева вы видите блоки, из которых собирают блок- схемы. Эти блоки выполняют засылки в порты, арифметические и логические операции, ожидание, запуск и останов программы и т.п.

Нажмите «Код» на верхней панели и вы увидите ASM код программы (рис. 3.56), соответствующей данной блок-схеме.

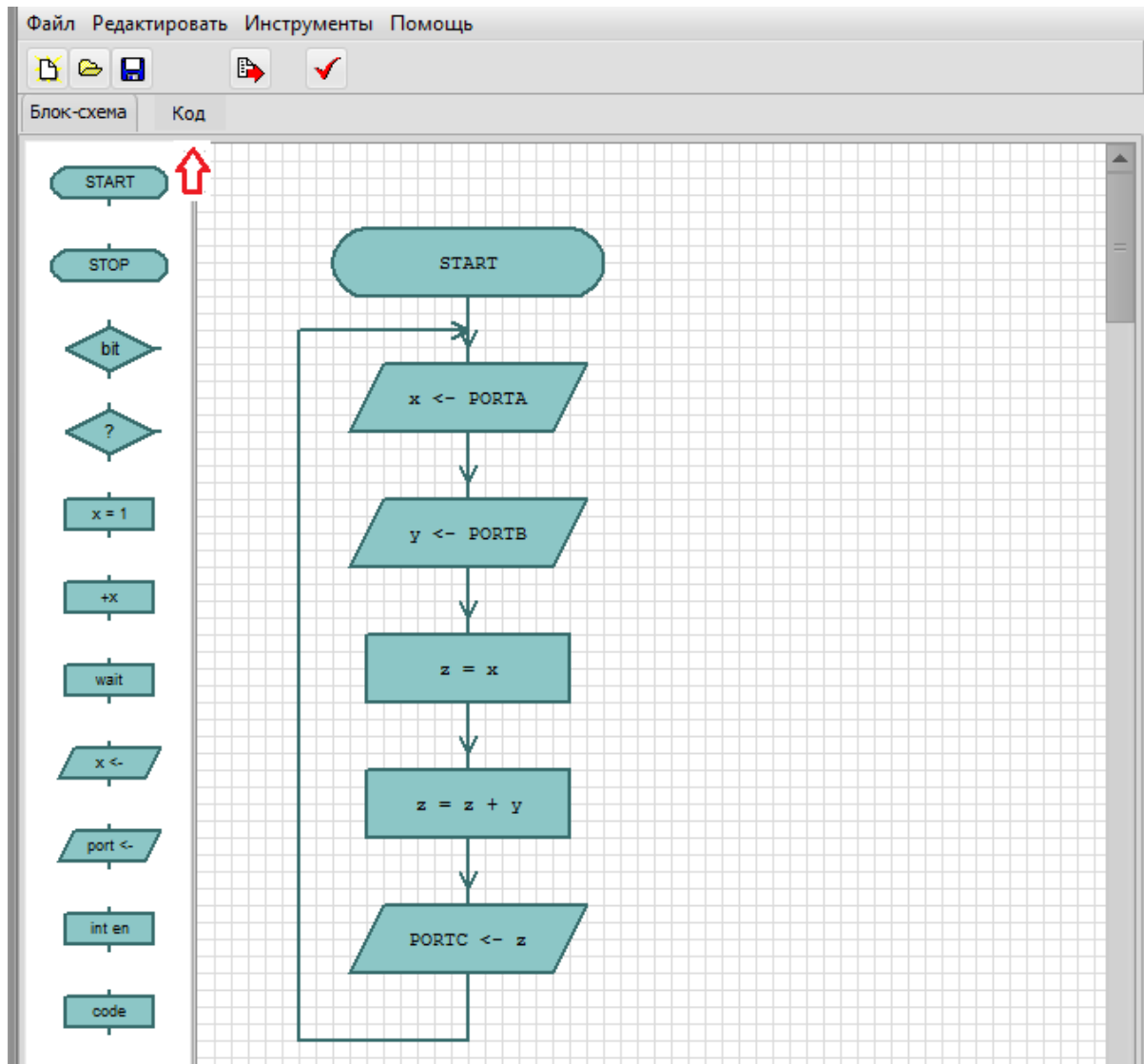



Рис. 3.55. Блок- схема программы

Чтобы сохранить блок-схему в макросе MCU, нажмите на панели инструментов , затем дважды нажмите ОК (при выборе входного файла MCU и на MCU диалоговое окно свойства), чтобы вернуться в редактор схем.

Вы также можете сохранять потоковые диаграммы в файлах .tfc с помощью команды Save и Save as...редактора блок-схем. Затем вы можете открыть их и связать их с другими MCU.

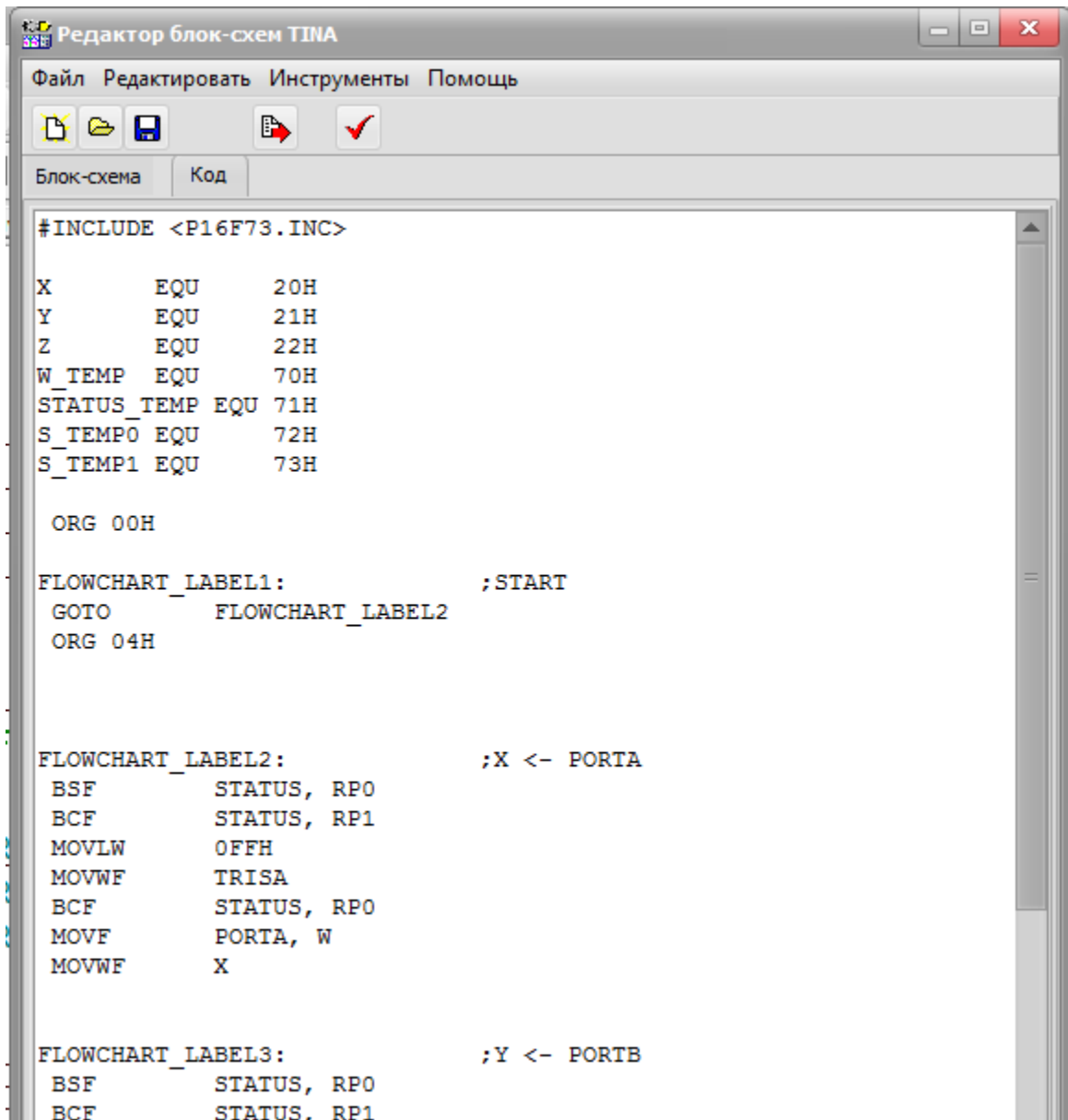


Рис. 3.56. Программа ASM для блок-схемы

3.9. Моделирование в смешанном режиме (Spice - VHDL – MCU совместное моделирование)

TINA версии 8 и выше включают новый мощный движок симулятора смешанного режима. Он основан на алгоритме смешанного режима XSPICE, расширен с помощью компонентов MCU и VHDL. В своих цепях вы можете

свободно смешивать любые аналоговые или цифровые компоненты TINA, включая микроконтроллеры (MCU) и макросы с содержанием Spice или VHDL.

Вы можете легко изменить эти компоненты вместе с кодом в MCU. TINA проанализирует аналоговые части в аналоговой виде, цифровые части в цифровом, и автоматически создаст интерфейсы между компонентами. Это обеспечивает синхронизацию и быструю сходимость моделирования.

Давайте рассмотрим некоторые из применений этого режима на нескольких примерах.

3.9.1. Генерация сигналов с использованием VHDL и Spice подсхемы

Следующая схема (EXAMPLES\VHDL\Mixed\Wavegenerator.TSC) генерирует аналоговый синусоидальный или пилообразный сигнал в зависимости от состояния левого переключателя SW-MODE (рис. 3.57).

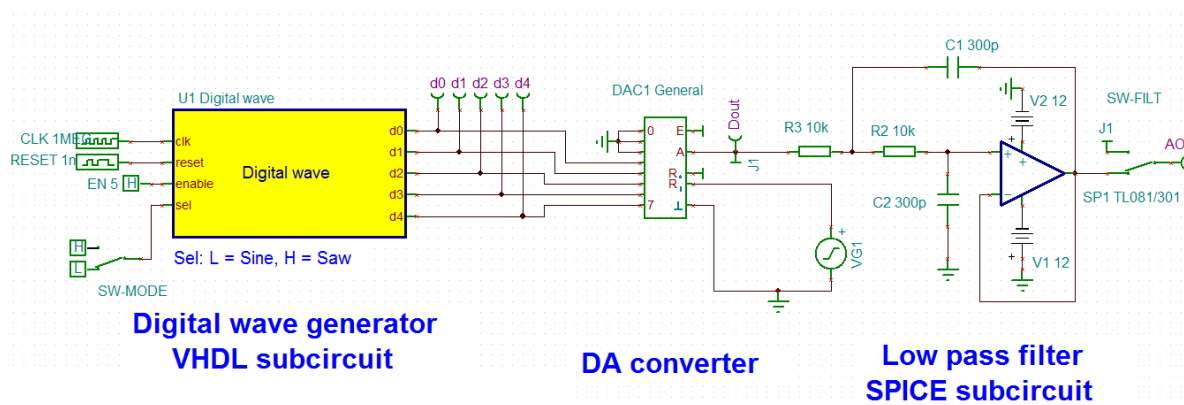


Рис. 3.57. Цифро-аналоговый генератор сигналов

Блок Digital Wave слева от схемы включает в себя VHDL код с таблицей поиска Sine_LUT для синусоиды и счетчик для пилообразного сигнала.

Выделим этот блок и в окне «Свойства» нажмем «Ввести подсхему».

Откроется редактор VHDL с полным текстом кода.

Важной частью кода VHDL является:

```

process(Reset, Clk)
begin
    if (Reset = '1') then
        Wave <= (others => '0');
        LUT_index <= 0;
    elsif rising_edge(Clk) then
        if (Enable = '0') then
            Wave <= (others => '0');
        elsif (Sel = '0') then
            Wave <= Sine_LUT(LUT_index);
        else
            Wave <=
conv_std_logic_vector(LUT_index, 5);
            end if;

            if (LUT_index = LUT_index_max) then
                LUT_index <= 0;
            else
                LUT_index <= LUT_index + 1;
            end if;
        end if;
    end process;

    d0 <= Wave(0);
    D1 <= Wave(1);
    D2 <= Wave(2);
    D3 <= Wave(3);

```

```
D4 <= Wave(4);
```

Вы можете увидеть все детали кода и изменить его при необходимости, дважды щелкнув поле Digital Wave и нажав кнопку Enter Macro в диалоговом окне его свойств (рис. 3.58).

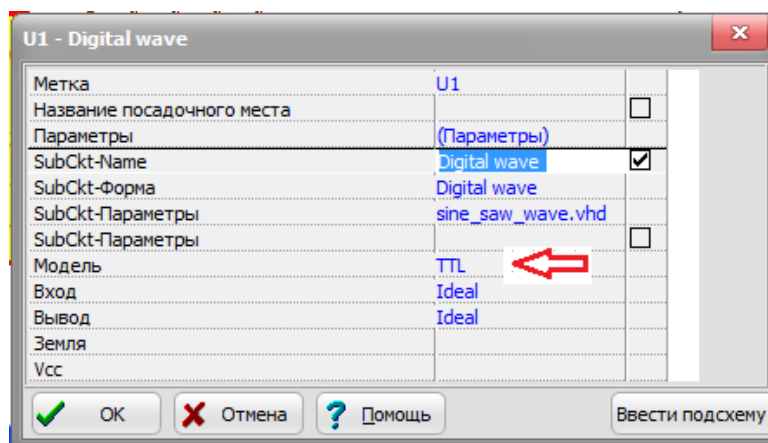


Рис. 3.58. Диалоговое окно свойств Digital Wave

Обратите внимание, что в этом диалоговом окне модель установлена на TTL, но вы можете выбрать из различных других моделей (CMOS, LS, HC, HCT и т. д.).

Цифровой выход счетчика преобразуется в аналоговый сигнал в 5-битном преобразователе DA TINA, показанном в середине цепи.

Выход синусоидальной формы ЦАП должен быть очищен с помощью фильтра низких частот. Мы будем использовать модель операционный усилитель Spice Opamp TL081 в фильтре нижних частот. Нажмите кнопку ввода макроса на диалоговое окно свойств и TINA откроет макрос (рис. 3.59). Код содержит LUT – таблицу для формирования синусоидального сигнала.

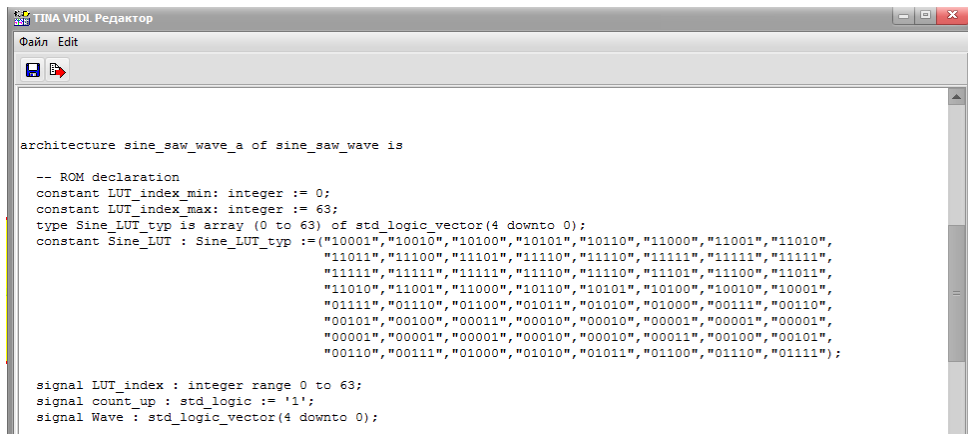


Рис. 3.59. LUT – таблица для синусоидального сигнала.

Важной частью кода является следующий. Вы можете просмотреть и, если необходимо, измените код Spice внутри макроса (рис. 3.60).

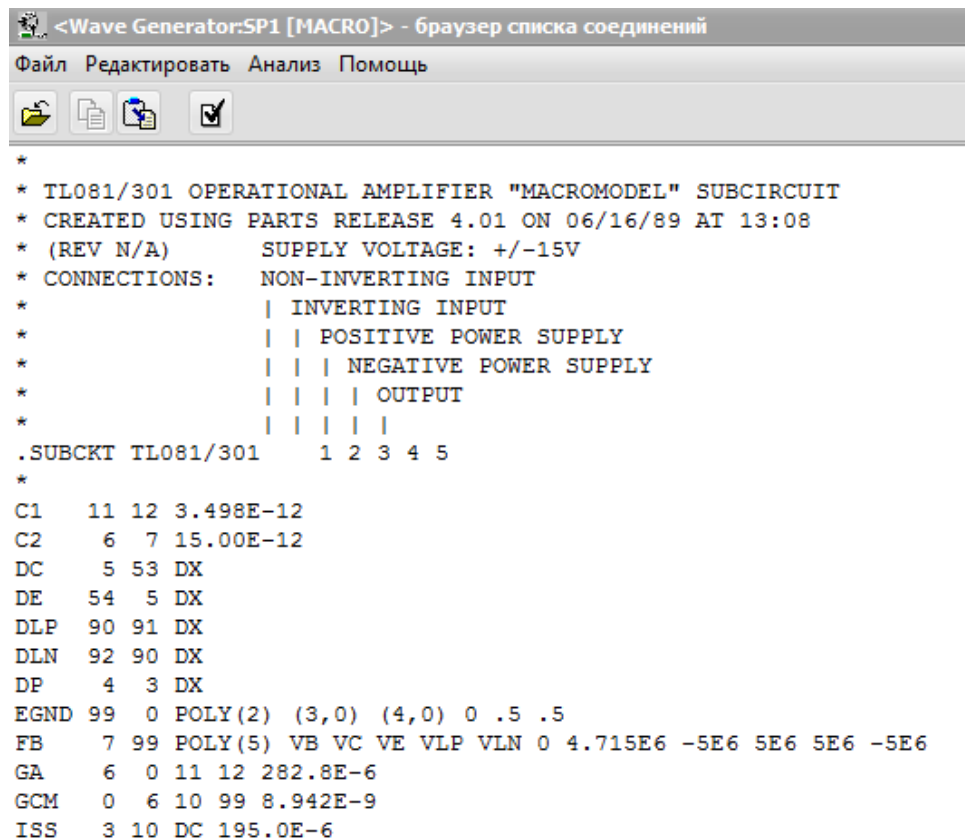


Рис. 3.60. Код макроса операционного усилителя

Пилообразный сигнал с выхода счетчика (на выводе J1) не требуется фильтровать, поэтому мы подключим его непосредственно к одному терминалу переключить SW_FILT.

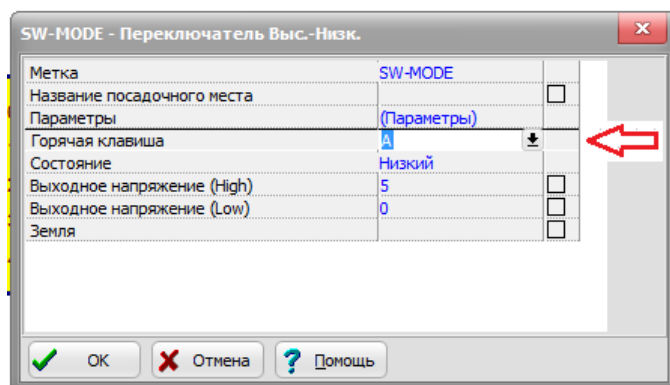


Рис. 3.61. Диалоговое окно переключателя

Синусоида формируется на выходе ЦАП (DAEX) и фактически требует фильтрации, поэтому мы пропустим ее через фильтр низких частот и подключим отфильтрованный аналоговый выход Aout к другому терминалу SW_FILT. Перемычка (J1) соединяет DAEX к вывод на коммутатор. Хотя это не очевидно на схеме, переключатели SW_FILT и SW_MODE синхронизируются, как будто они были переключателем DPDT. Мы заставляем их синхронизироваться путем назначения обоим переключателям для управления одной горячей клавишей A. (См. диалог свойств для SW_FILT и SW_MODE, где была назначена горячая клавиша A) (Рис. 3.61).

Вот окончательные формы сигналов всей цепи, включая пять выходных сигналов счетчика (рис. 3.62). SW_MODE находится в состоянии High, выбрав пилообразный сигнал. Моделирование проведено в режиме Переходные процессы.

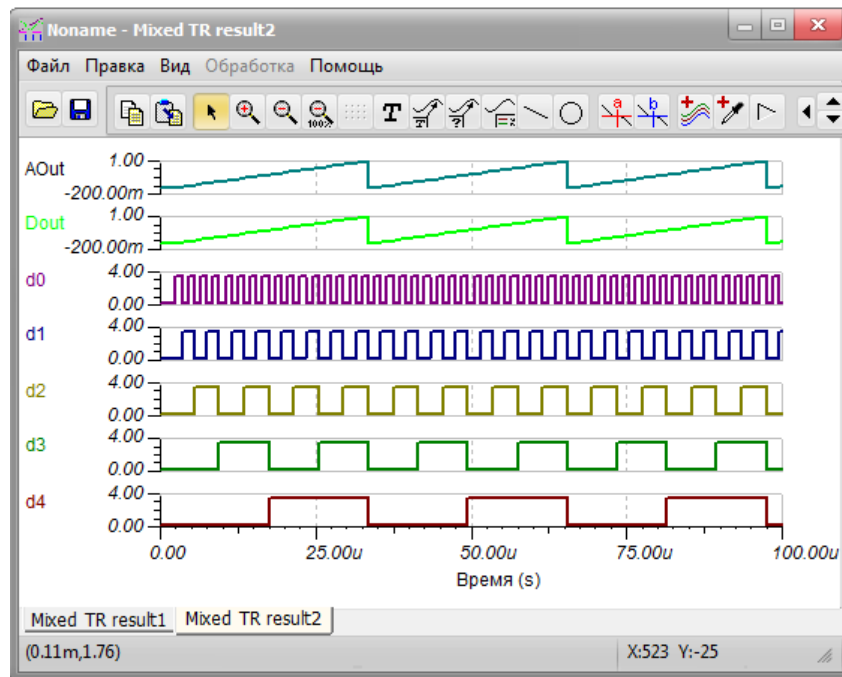


Рис. 3.62. Формирование пилообразного сигнала

Если мы изменим переключатель SW-MODE на Low и запустим Transient анализ снова, получим такие формы волны (рис. 3.63).

Чтобы увидеть эффект аналогового фильтра, удалите кривые с d0 по d4 из диаграмма, нажав на кривые и нажав клавишу Del. Кроме того, вы можете временно удалить выходы от d0 до d4 и запустить Анализ переходных процессов снова (рис. 3.64).

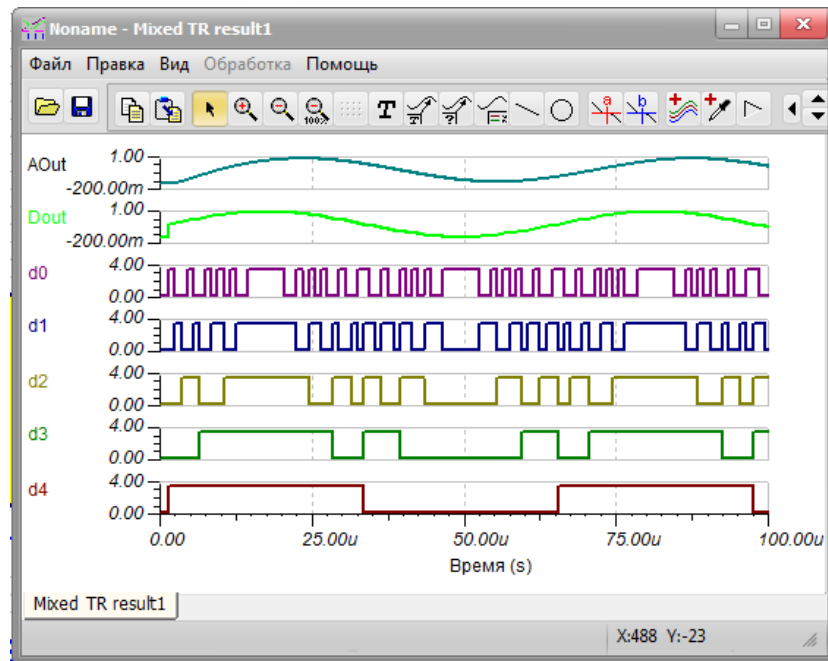


Рис. 3.63. Формирование синусоидального сигнала

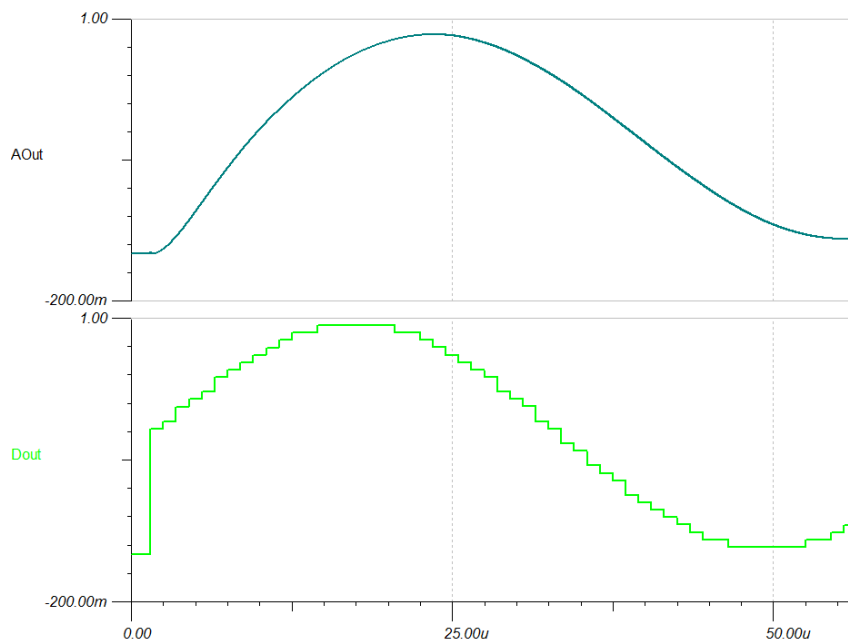


Рис. 3.64. Эффект от сглаживающего фильтра

Чтобы продемонстрировать гибкость VHDL-функций в TINA, мы изменим код VHDL для генерации прямоугольной волны вместо пилообразной формы. Просто установите команды с Wave (0) до Wave (3) на ноль в коде

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

VHDL. Дважды щелкните макрос Digital Wave и нажмите кнопку Enter Macro кнопка. Найдите команду Wave <= conv_std_logic_vector(LUT_index,5) <= и вставьте следующие операторы:

```
Wave(0) <= '0';
Wave(1) <= '0';
Wave(2) <= '0';
Wave(3) <= '0';
```

Вы можете обновить макрос, просто закрыв редактор (нажмите кнопку в правом верхнем углу окна). Моделирование покажет прямоугольную волну (рис. 3.65).

Вы можете проверить более сложную версию этой схемы в EXAMPLES\VHDL\Mixed\Wave generator dipsw.TSC. Там вы можете выбрать все три формы волны, которые мы обсуждали, используя DIP-переключатель.

Обратите внимание, что вы можете загрузить часть кода VHDL в ПЛИС и использовать аппаратную форму.

3.9.2. Генерация сигналов с использованием SystemC и Spice подсхемы

В полной индустриальной версии TINA-12 есть пример моделирования генератора сигналов, в котором подсхема цифрового генератора описана на языке SystemC. Исследование такой схемы возможно в программе TINA-12 с установленным расширением HDL, которое требует дополнительной оплаты.

На рис. 3.66 показана схема модели с описанием в SystemC.

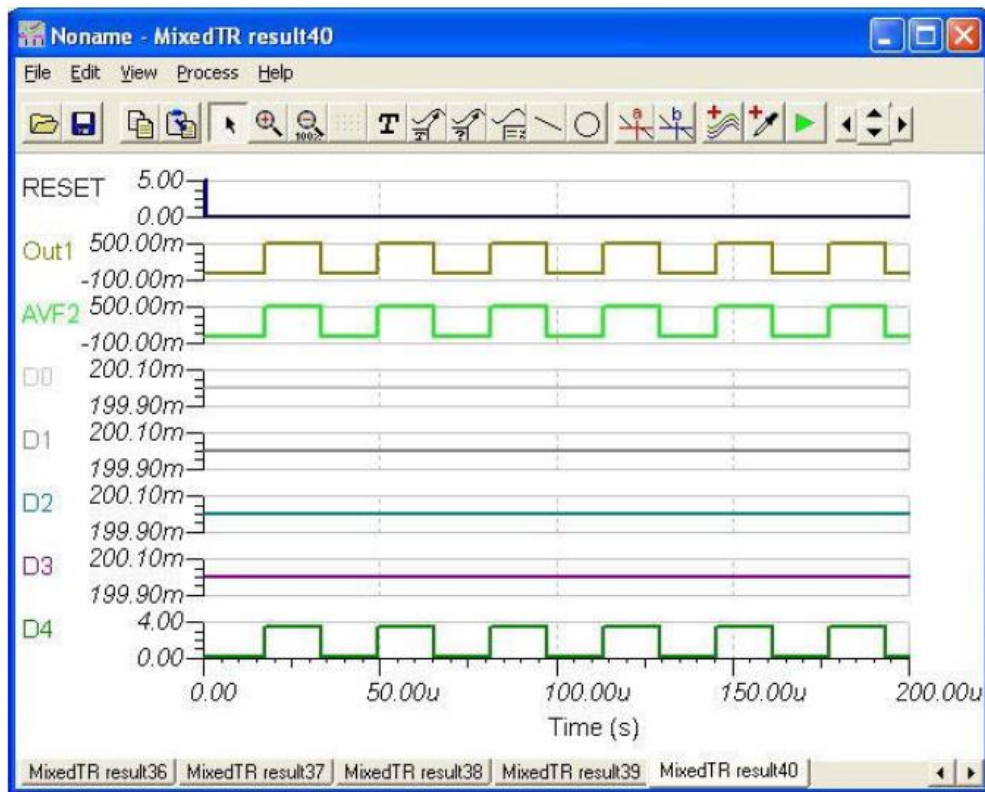


Рис. 3.65. Формирование прямоугольной волны

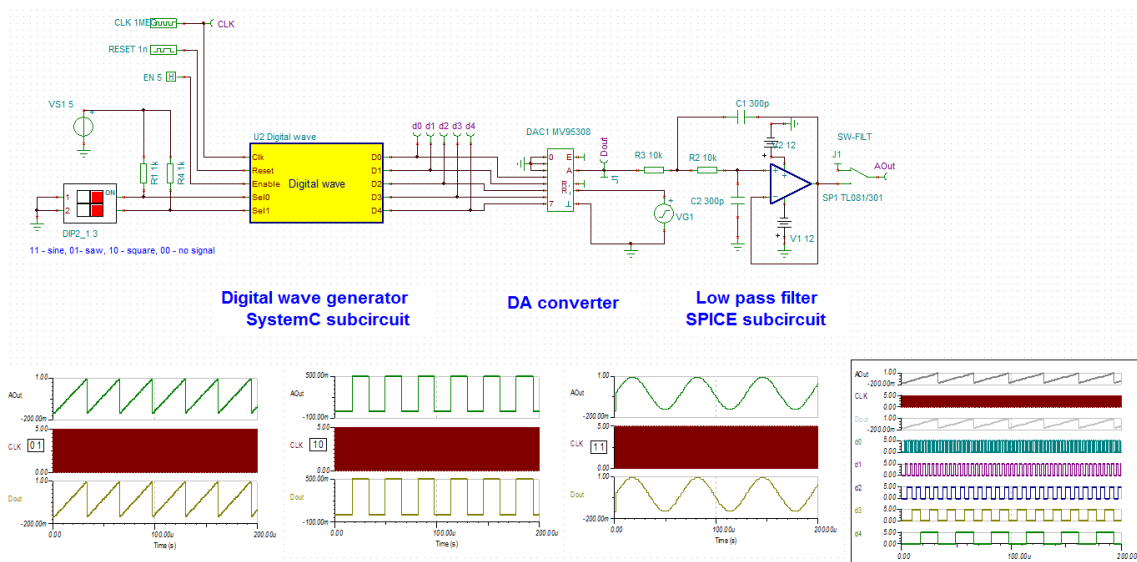


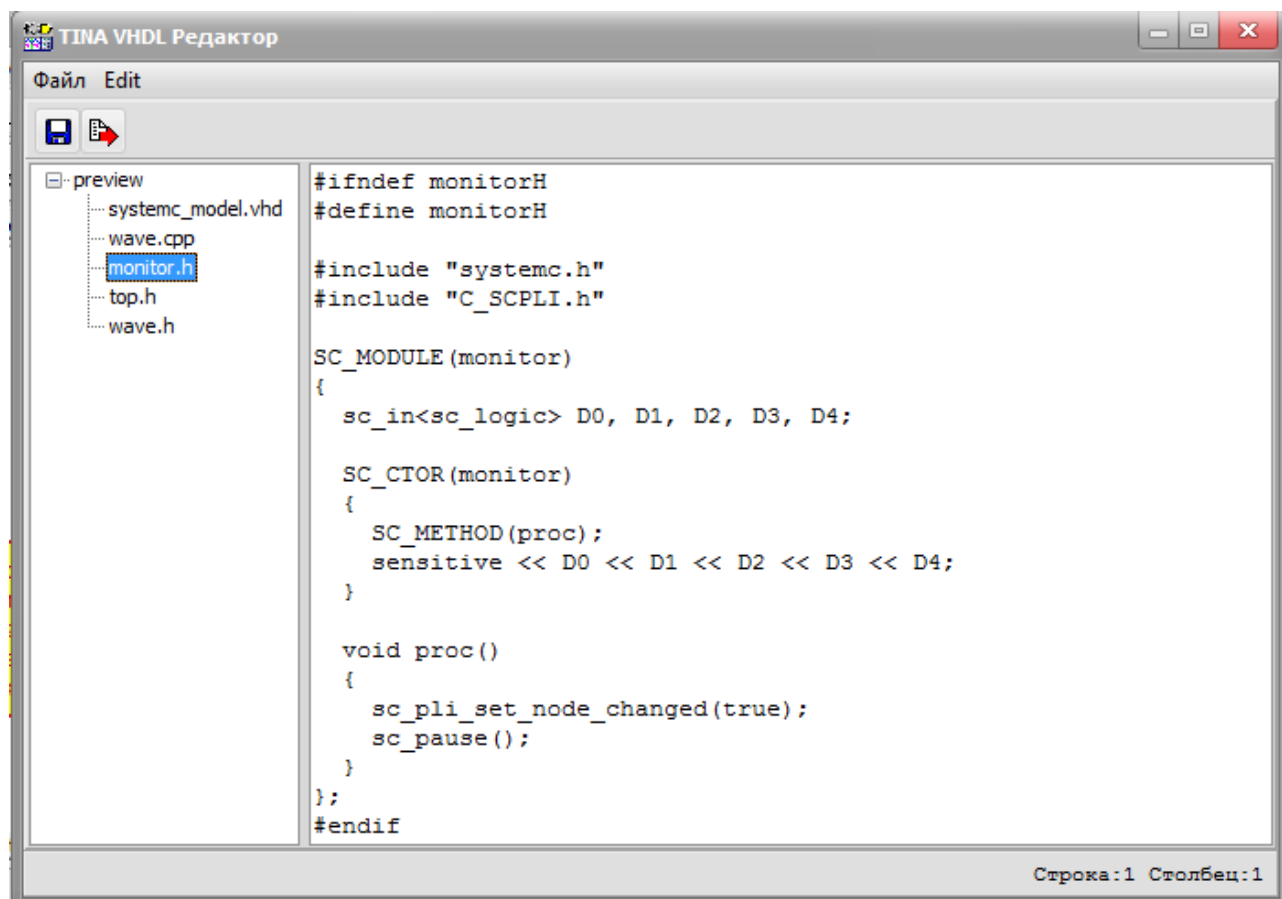
Рис. 3.66. Схема модели с описанием в SystemC

Для примера один из файлов SystemC_model VHDL показан на листинге 3.1.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Файл monitor.h


Листинг 3.1



3.9.3. Импульсный источник питания на микроконтроллере

Импульсный источник питания SMPS (Switched Mode PowerSupply) в программе TINA смоделирован на микроконтроллере PIC16F88 (рис. 3.67).

Микроконтроллер PIC в цепи вырабатывает выходной сигнал ШИМ на выводе RB0, который контролирует переключение FET T1. Процедура прерывания кода в PIC сравнивает напряжение обратной связи на VFB (подключен к контакту RA1 PIC), со встроенным пороговым напряжением. Если напряжение ниже чем порог, определенный в коде, рабочий цикл выходного сигнала ШИМ увеличивается. Вы можете изучить код ASM в PIC,

дважды щелкнув по PIC, щелкнув по строке MCU, нажав кнопки  (маленькая кнопка с тремя точками подряд), и, наконец, нажав кнопку «Редактировать ASM».

Вы можете увидеть и отладить код [здесь](#).

Нажмите на строку «Включить код отладчика MCU» в меню «Анализ», нажмите TR, «Интерактив» в меню «Анализ», и, наконец, нажмите на Старт.

На выходе источника установится напряжение около 9 В.

Выполните анализ переходных процессов. Формы сигналов (рис. 3.68) демонстрируют, как аналоговые части и MCU взаимодействуют в TINA.

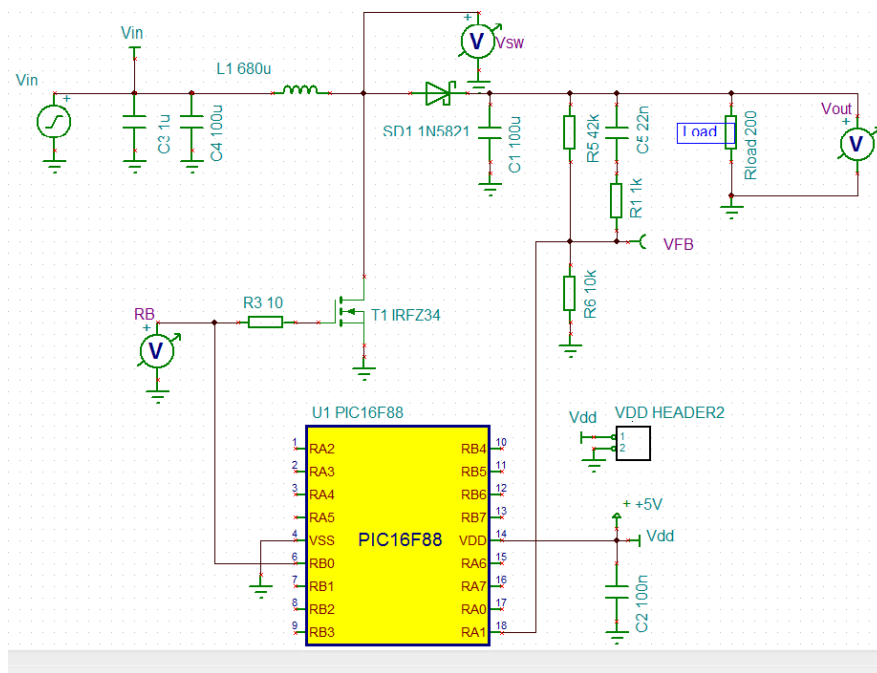


Рис. 3.67. Импульсный источник питания на микроконтроллере

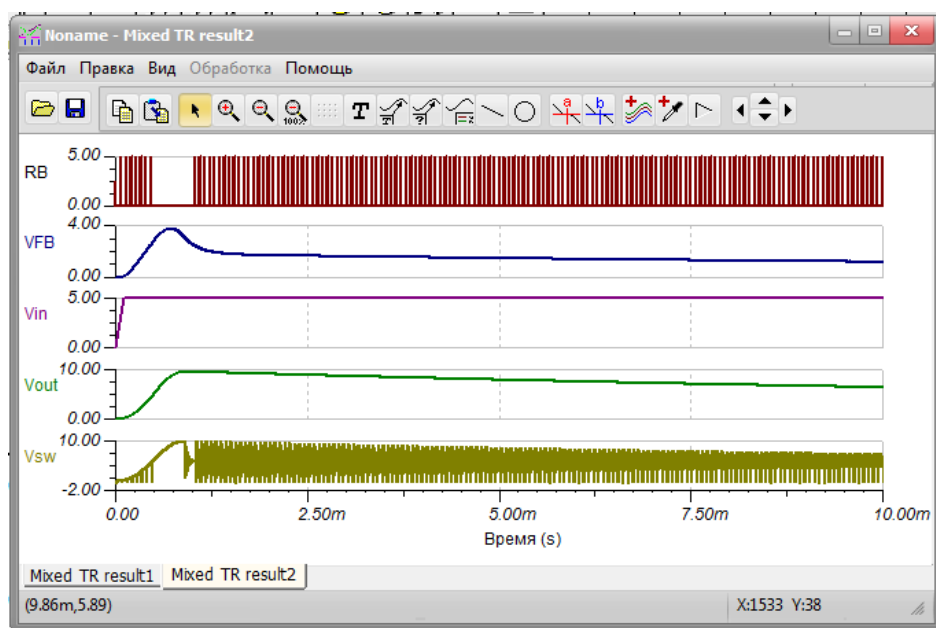


Рис. 3.68. Сигналы в импульсном источнике питания

3.9.4. Схемы для формирования сигналов в инфокоммуникационных системах

В папке Examples\Communication вы можете найти и исследовать важные устройства, применяемые в инфокоммуникационных системах: амплитудный модулятор и демодулятор, амплитудный двоичный модулятор (манипулятор) (ASK – Amplitude shift keying) и демодулятор, частотный двоичный модулятор (манипулятор) и демодулятор (FSK – Frequency shift keying), фазовый двоичный модулятор (манипулятор) (Phase shift keying) и демодулятор (Phase shift keying (Фазовая двоичная модуляция)- манипуляция), и др. схемы (рис. 3.69)

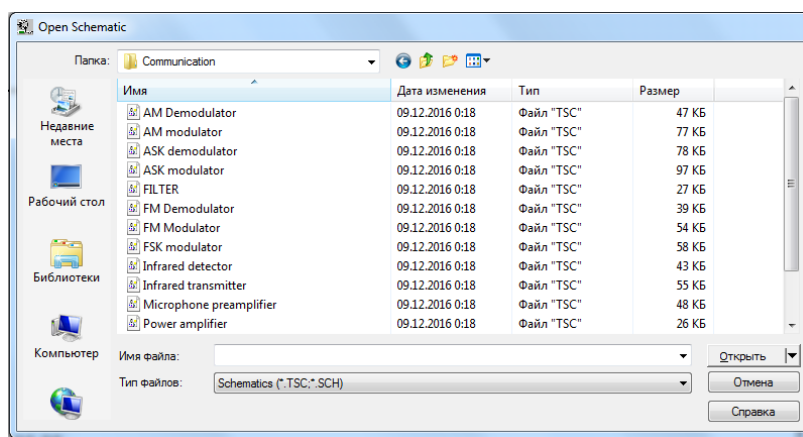


Рис. 3.69. Каталог схемы устройств в папке Communication

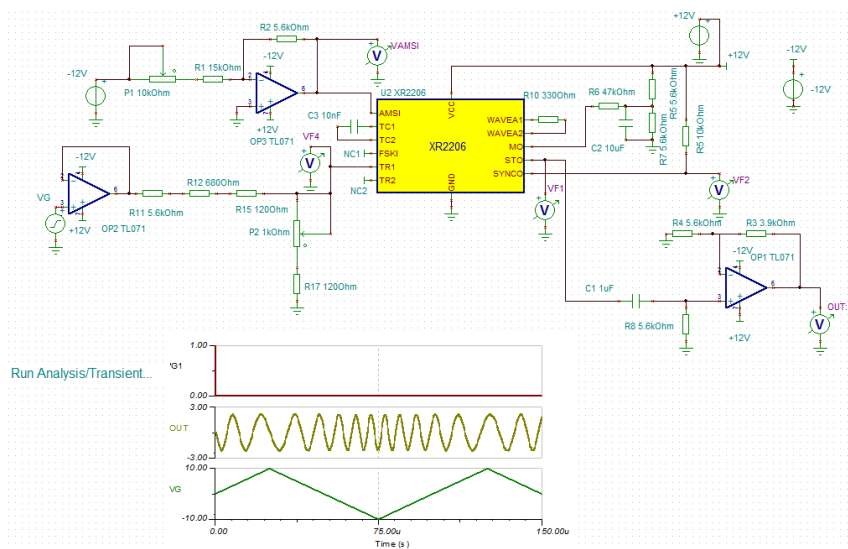


Рис. 3.70. Модель частотного модулятора

На рис. 3.70 показана модель частотного модулятора и диаграммы сигналов.

3.9.5. Модели устройств компании Infineon

Разработчик САПР TINA компания DesignSoft тесно сотрудничает с компанией Infineon (ФРГ) - крупным производителем разнообразной электроники. На рис. 3.71 показана модель промышленного импульсного источника питания AC/DC (220V/5V-12W).

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

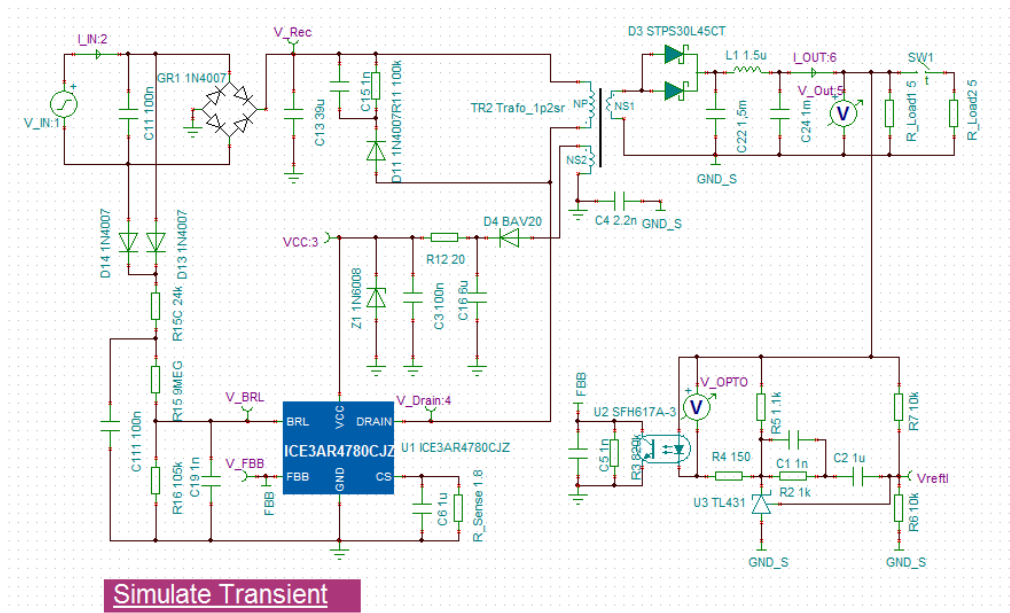


Рис. 3.71. Модель импульсного источника питания AC/DC

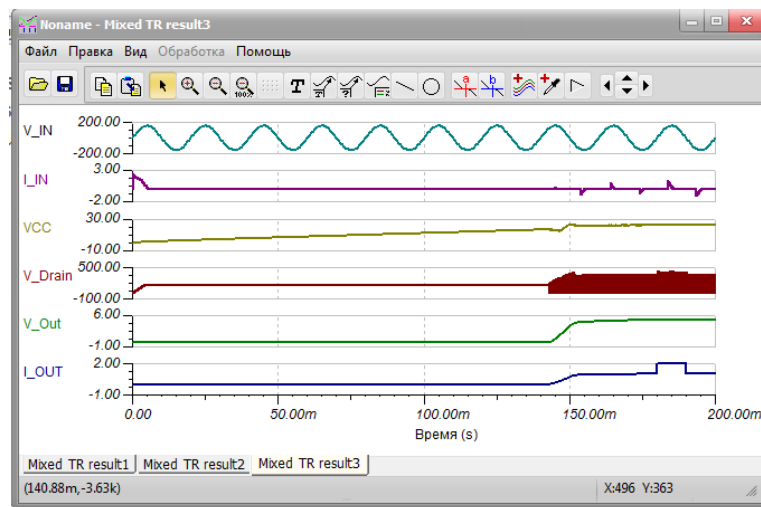


Рис. 3.72. Диаграммы сигналов импульсного источника питания

Диаграммы напряжений показаны на рис. 3.72.

Такие модели компания Infineon использует в производственном процессе для проверки и наладки аппаратуры.

3.10. Проектирование печатных плат в среде TINA

В этом разделе описана последовательность операций в среде TINA PCB Design Suite (конструкторский комплект для печатных плат) на примере проектирования печатной платы активного фильтра (рис. 3.73).

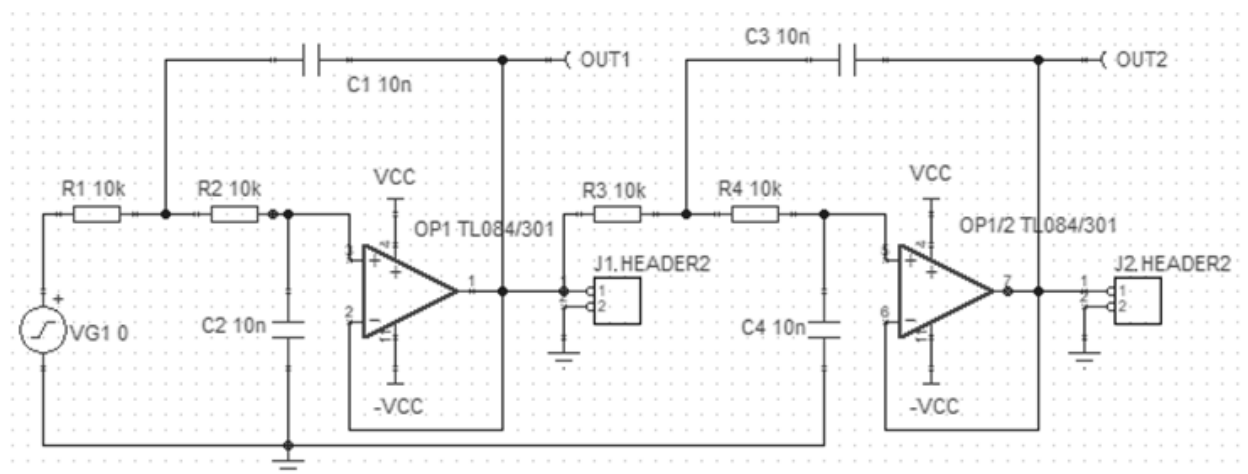


Рис. 3.73. Схема активного фильтра

В PCB-дизайне главное, чтобы каждый элемент схемы имел свое физическое представление с точными физическими размерами. Для этого в TINA есть компоненты Footprints (футпринты) - отпечатки (следы) - специальные чертежи, которые показывают контуры и выводы этих элементов. Можно использовать "Footprint name editor", который находится на панели инструментов во вкладке Tools. В этом окне можно увидеть все элементы и соответствующие им футпринты, которые присутствуют в схеме (рис. 3.74).

Чтобы увидеть конкретную информацию о каждом элементе схемы отдельно, то необходимо выбрать свойства этого элемента и в открывшемся окне выбрать Footprint name. В новом окне (PCB information) будет вся информация об этом элементе и даже его вид в 3D (рис. 3.75).

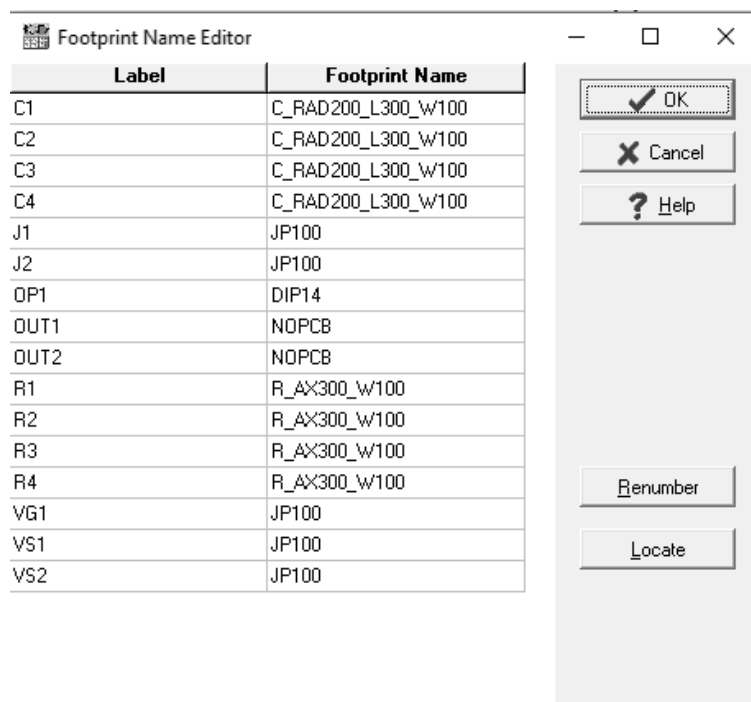


Рис. 3.74. Редактор футпринтов для схемы активного фильтра

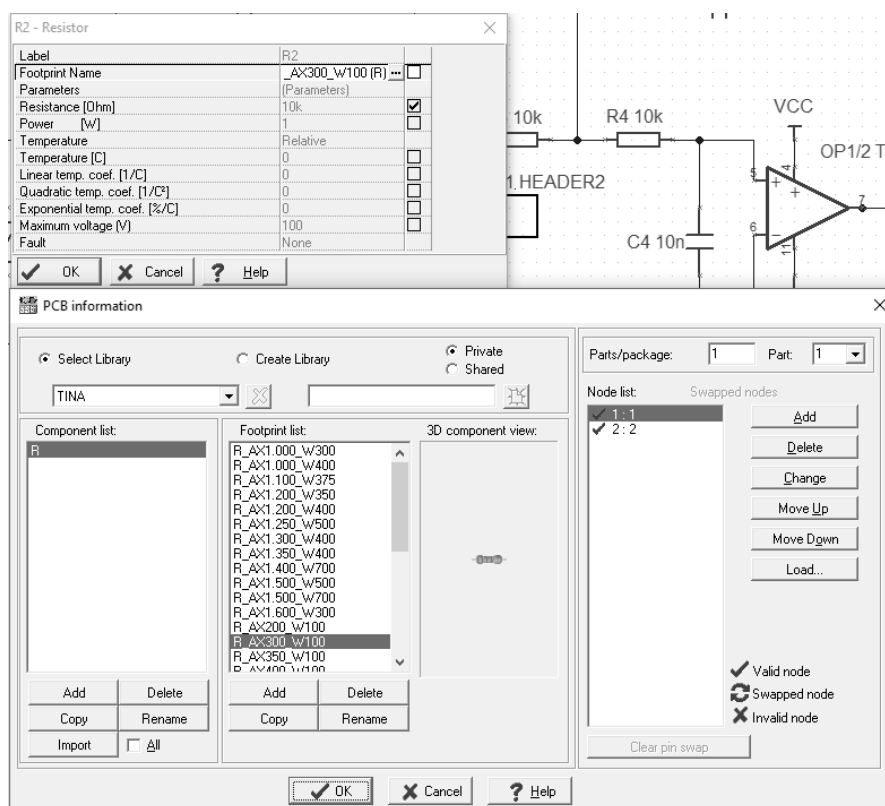


Рис. 3.75. Информация об элементе схемы R

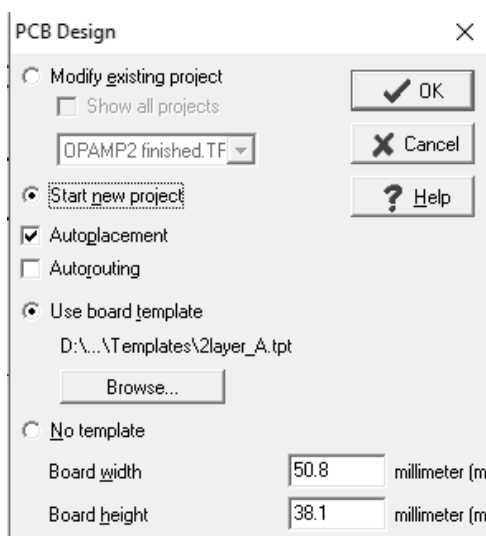


Рис. 3.76. Задание начальных параметров платы

Когда мы утвердили все необходимые параметры, нажимаем ОК и, если схема уже готова к РСВ-проектированию, то нужно нажать на правую крайнюю кнопку на панели инструментов - PCB design, или можно выбрать эту опцию на вкладке Tools. Откроется новое окно, в котором вводятся исходные данные для настройки печатной платы.

Выбираем Start new project, затем ставим галочку Autoplacement, чтобы элементы были автоматически размещены на схеме, но Autorouting пока не используем, так как будет удобнее это сделать в конце непосредственно после корректировки элементов, если вдруг нужно будет их переместить на схеме. Будем использовать шаблон для платы "2layer_A" – двуслойная плата типа А, которая относится к общей (средней) сложности проектирования. Выбираем размеры платы оставляем по стандарту (50.8 мм на 38.1 мм) (рис. 3.76).

Нажимаем ОК и открывается PCB designer с нашей схемой (рис. 3.77).

Как видно по рисунку, элементы расположились не совсем так, как нам это нужно, поэтому можно вручную разместить их с помощью мышки. Нажимаем на элемент и переносим его в то место, где он должен быть. При необходимости его можно вращать, нажав на него правой кнопкой мыши.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

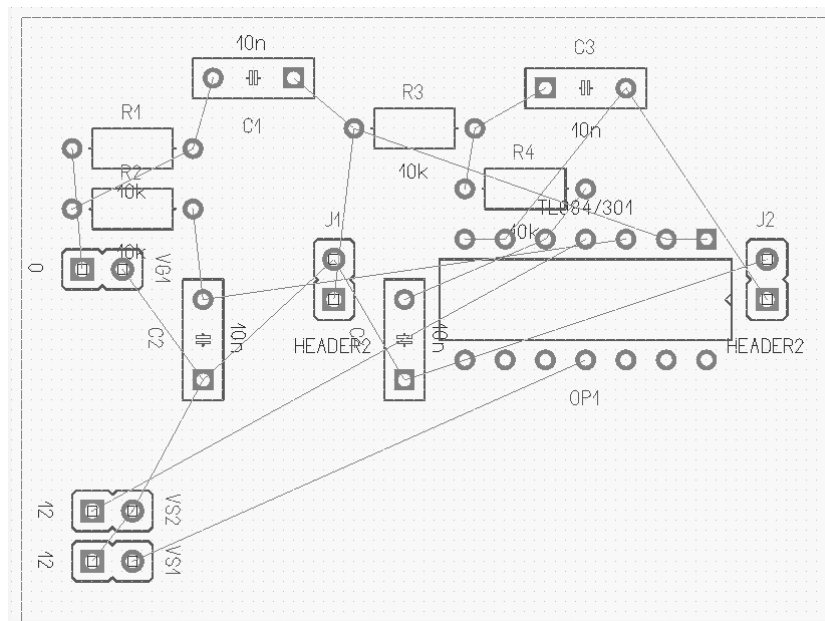


Рис. 3.77. Первоначальный макет печатной платы

После ручного размещения у нас будет более удобное расположение для проведения каналов связи между элементами, о которых будет подробнее сказано далее. После этого выбираем на вкладке Tools → Net editor, который откроет окно с настройками маршрутизации. Нажмем Modify All и в новом окне в поле Track width зададим ширину 12,5. Затем, нужно, зажав клавишу shift, выделить первые три элемента (GND, -VCC, VCC), чтобы задать им ширину 25 (рис. 3.78).

Нажимаем ОК, затем Tools → Autorotate board и мы видим, что у нас автоматически произошла маршрутизация (трассировка). Теперь на нашей плате появились соединительные дорожки, необходимые для взаимодействия элементов (в компьютерных науках их называют шины). Их также можно подправить мышкой, если что-то пошло не так. Теперь, осталось проверить нашу плату на ошибки. Для этого нажимаем F7 → DRC (Design Rule Check) и программа выдаст информацию о наличии или отсутствии ошибок. Если всё

нормально и программа не нашла ошибок, то можно добавить область с текстом на плату, выбрав во вкладке Insert → Text и разместить текст на плате (рис. 3.79).

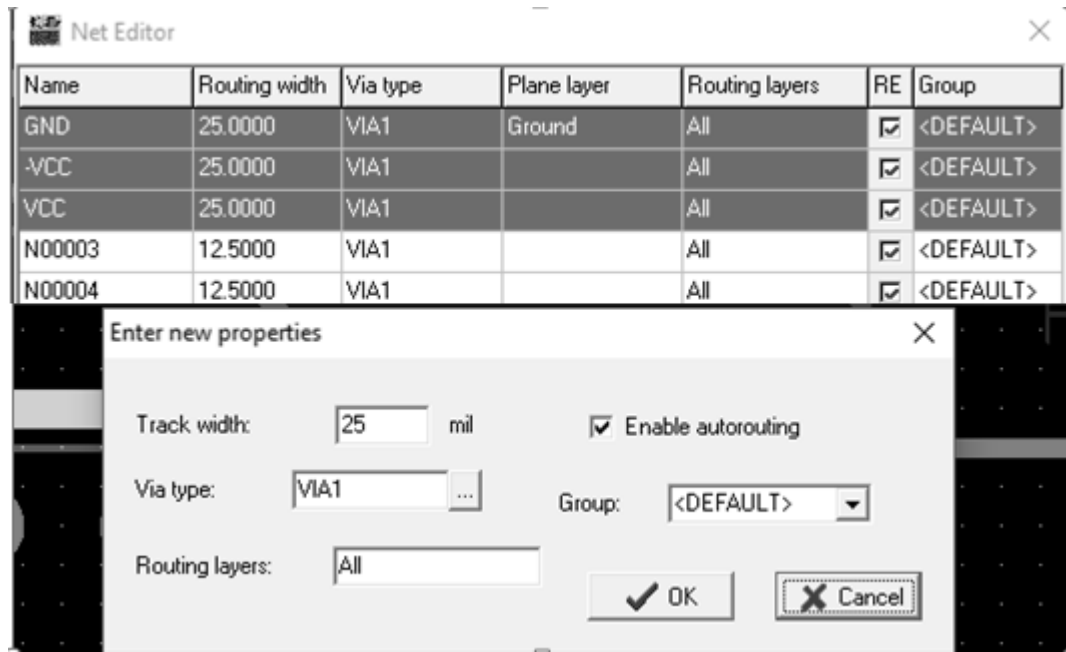


Рис. 3.78. Задание ширины дорожек печатной платы

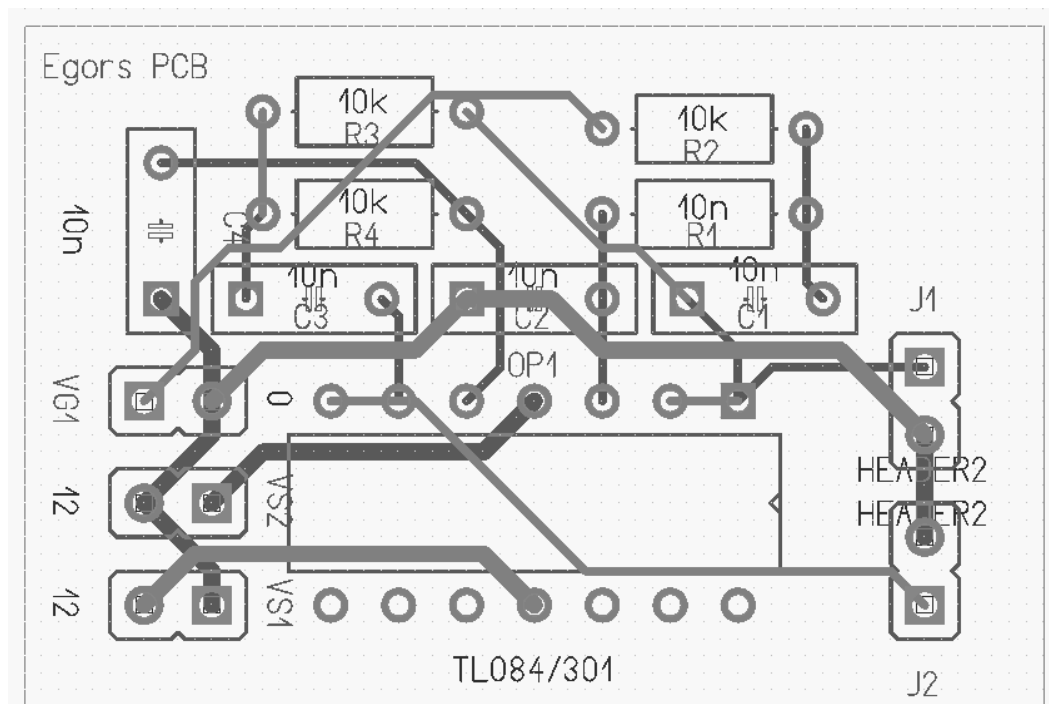


Рис. 3.79. Плата после трассировки

Когда все сделано, можно нажать View → 3D view и рассмотреть плату в 3D режиме (рис. 3.80).

Если есть проблемы с маршрутизации, то всегда можно использовать проверку DRC, нажав на клавишу F7. Появится окошко, в котором будет сказано, есть или нет ошибки. Теперь у нас есть схема, и если что-то нужно подвинуть, то можно выбрать "Select/Move components/tracks" и переместить все компоненты и дорожки как нам удобно. Чтобы выделить всю дорожку, необходимо зажать Shift и кликнуть на нужную нам. Чтобы удалить, нужно нажать клавишу Del. Чтобы удалить и не использовать какую-то дорожку, нажмите F4 и снимите галочку в столбце RE.

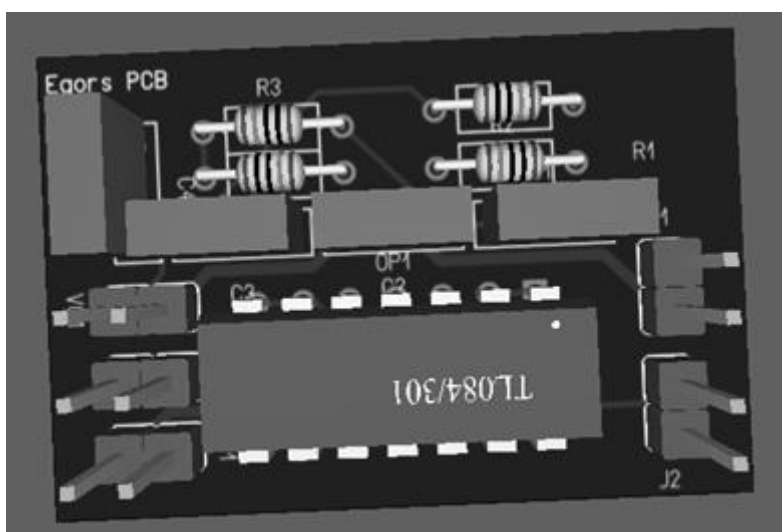


Рис. 3.80. 3D вид готовой платы

Чтобы посмотреть на исходную схему с нашей платой, достаточно сохранить результат, который мы проделали в PCB Design и затем, в редакторе схем выбрать Tools → Backannotate и выбрать сохраненный файл. Затем программа Tina обновит исходную схему, и все пины (места соединений) которые были изменены, будет отображены на схеме.

После этого можно распечатать плату или сделать расширение файла для фотопринтера Gerber. Необходимое можно выбрать в меню File (если нет ограничений в демоверсии программы). Чтобы получить файл формата Gerber (RS – 274X format) напрямую для "фотоплоттера", нужно выбрать Export Gerber file из вкладки File.

Таким образом, показано, что методика проектирования печатных плат в среде TINA вполне эффективна, понятна и может быть применена для обучения студентов основам САПР в курсах электроники и микроконтроллеров.

Глава 4. Проектирование цифровых устройств в среде OrCAD 17.2

4.1. Цифровое моделирование в среде OrCAD 17.2

Система автоматизированного проектирования OrCAD 17.2 компании Cadence - это одна из лучших программ сквозного проектирования электронной аппаратуры, предоставляющая дизайнерам широкие возможности разработки и моделирования электронных схем и создания печатных плат.

Последние версии OrCAD (16.6, 17.2) имеют всё необходимое для выполнения различных этапов процесса разработки: входное проектирование, функциональное моделирование, синтез, размещение, маршрутизация, моделирование задержек, генерация элемента. Основным недостатком OrCAD является высокая стоимость профессиональных версий. Однако, компания Cadence предлагает бесплатные облегченные учебные версии программы OrCAD Capture CIS Lite, которые мы используем в нашем учебном пособии.

Подробно изучить моделирование в САПР OrCAD 17.2 можно в [15].

Реальные электронные устройства в настоящее время разрабатывают с использованием компьютерных программ сквозного проектирования, в которых электронные компоненты моделируются PSpice – моделями.

PSpice использует один и тот же механизм моделирования как для аналоговых, так и для цифровых компонентов. Цифровая транзисторно-транзисторная логика (ТТЛ) и комплементарные полевые транзисторы (CMOS) моделируются как подсхемы и включают в себя общие цифровые функции, такие как вентили, регистры, триггеры и инверторы. В каждой подсхеме цифровой примитив составляет функцию вентилей (AND, OR и т. д.) и определяет спецификацию времени и интерфейса для функции вентилей. Другие цифровые устройства включают в себя линии задержки, аналого-цифровые преобразователи АЦП (ADC или AtoD), цифроаналоговые

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

преобразователи ЦАП (DAC или DtoA), память (RAM, ROM) и программируемые логические массивы.

4.1. Модели цифровых устройств

Библиотеки стандартных компонентов содержат полный набор цифровых компонентов, выполненных по разным технологиям. Каждый цифровой компонент описан электрически с помощью модели цифрового устройства в виде подсхемы и определение хранится в библиотеке моделей. Имя соответствующей подсхемы определяется значением атрибута MODEL компонента. Другие атрибуты - MNTYMXDLY, IO_LEVEL и набор PSPICEDEFAULTNET - передается в подсхему, таким образом обеспечивая высокоуровневое средство для воздействия на поведение цифровой модели устройства.

Как правило, цифровые компоненты, представленные в библиотеках, являются достаточными для большинства схем. Однако, если ваш дизайн требует цифровые компоненты, которые еще не были представлены в PSpice и библиотеках моделей, вам необходимо самостоятельно определить модели цифровых устройств, соответствующих новым цифровым компонентам.

Полная цифровая модель устройства имеет три основные характеристики:

- Функциональное поведение: описывается уровнем вентилях и поведением цифровых примитивов, содержащихся в подсхеме.
- Поведение ввода - вывода: описывается моделью ввода-вывода, интерфейсом и источниками питания, связанными с логическими устройствами.
- Временное поведение: описывается одной или несколькими моделями времени с примитивными задержками передачи от выводов к выводам (pin-to-pin) или примитивной проверкой ограничений.

4.1.1. Функциональное поведение

Функциональное поведение модели цифрового устройства определяется одним или несколькими взаимосвязанными цифровыми примитивами. Как правило, логическая диаграмма цифрового устройства может быть реализована непосредственно с использованием примитивов, предоставляемых в PSpice. В приведенной ниже таблице приведен краткий обзор цифровых примитивов.

Таблица 4-1

Сводка цифровых примитивов

Тип	Описание	Тип	Описание
BUF	Буфер	Двунаправленные шлюзы передачи	
INV	Инвертор	NBTG	N-канальный шлюз передачи
AND	И	PBTG	P-канальный шлюз передачи
NAND	И-НЕ	Триггеры и защелки	
OR	ИЛИ	JKFF	J-K-триггер, срабатывание с отрицательным фронтом
NOR	ИЛИ-НЕ	DFF	D-триггер, срабатывание положительным фронтом
XOR	Исключающее ИЛИ	SRFF	RS-триггер
NXOR	Исключающее ИЛИ-НЕ	DLTCH	D-триггер
BUFA	Буферный массив	PULLUP	Подтягивающая рези-

			сторная матрица
INVA	Инверторный массив	PULLDN	Стягивающая резисторная матрица
ANDA	Массив И	Линии задержки	
NANDA	Массив И-НЕ	DLYLINE	Линия задержки
ORA	Массив ИЛИ	Программируемые логические массивы	
NORA	Массив ИЛИ-НЕ	PLAND	Массив И
XORA	Массив исключающего ИЛИ	PLOR	Массив ИЛИ
NXORA	Массив исключающего ИЛИ-НЕ	PLXOR	Массив Исключающее ИЛИ
AO	Составной элемент И-ИЛИ	PLNAND	Массив И-НЕ
OA	Составной элемент ИЛИ-И	PLNOR	Массив ИЛИ-НЕ
AOI	Составной элемент И-(ИЛИ-НЕ)	PLNXOR	Массив исключающего ИЛИ-НЕ
Примечание: вышеперечисленные элементы с добавлением цифры 3 (AND3, OR3A и т.п.) имеют третье состояние с		PLANDC	И массив, истина и дополнение

большим импедансом.		
---------------------	--	--

Таблица 4.1 (Продолжение)

Тип	Описание	Тип	Описание
Память		Многобитовые преобразователи аналого-цифровые (ЦАП) и цифроаналоговые (АЦП)	
ROM	ПЗУ	ADC	Многобитовый АЦП
RAM	ОЗУ	DAC	Многобитовый ЦАП
Поведенческие модели			
LOGICEXP	Логическое выражение	PINDLY	Задержка от пина к пину
CONSTRAINT	Проверка ограничений		

Формат цифровых примитивов аналогичен формату аналоговых устройств. Одно отличие состоит в том, что большинство цифровых примитивов требуют двух моделей вместо одной:

- Модель времени, которая определяет задержки распространения и временные ограничения, такие как время установки и удержания.

- Модель ввода-вывода, которая определяет информацию, относящуюся к входным / выходным характеристики устройства.

Причина наличия двух моделей заключается в том, что, хотя информация о времени специфическая для устройства, характеристики ввода/вывода специфичны для целой группы логических устройств. Таким образом, многие устройства в одной и той же групповой ссылке имеют одну и ту же модель ввода-вывода, но каждое устройство имеет свою собственную модель синхронизации.

На рис. 4.1 представлен обзор определения цифрового устройства в терминах его примитивов и основных атрибутов модели. Это модель элемента И-НЕ, выполненного по технологии CMOS с двумя входами.

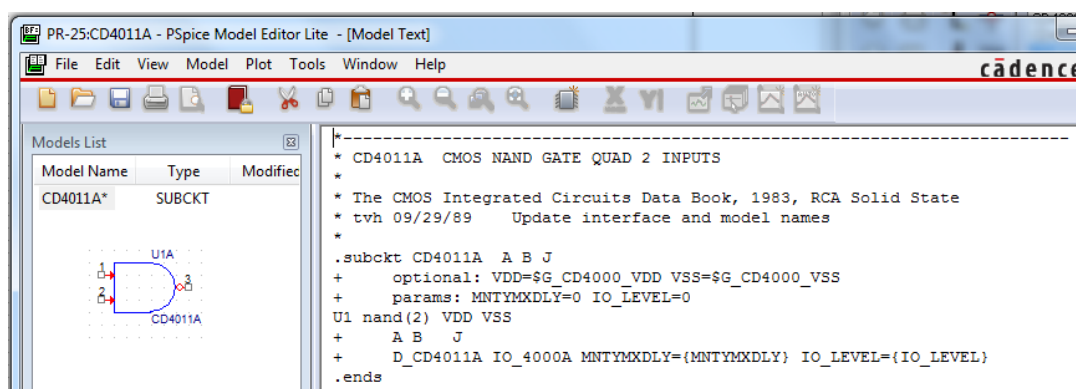


Рис. 4.1. Описание элемента И-НЕ

Первые пять строк - это комментарии, дающие описание компонента и ссылку на источники данных. В строке 6 приведено определение подсхемы CD4011B с тремя контактами A, B и J. Глобальный источник питания определяется $VDD=\$G_CD4000_VDD$ и $VSS=\$G_CD4000_VSS$. Необязательными параметрами являются $MNTYMXDLY=0$, который определяет минимальную, типичную и максимальную задержку и IO_LEVEL , который определяет одну из четырех аналого-цифровых (AtoD) или (DtoA) интерфейсных подсхем, если цифровое устройство подключается к аналоговому устройству.

Компонент U1 определяет примитив с двумя входами `nand(2)`, который имеет входные терминалы `VDD`, `VSS`, `A`, `B` и `J`. Знак «+» означает продолжение на следующей строке. Следующая строка (строка 11) объявляет две модели: временную модель `D_CD4011B`, которая определяет временные параметры, такие как задержка распространения, время установки и удержания, а также модель входа/вывода (I/O), `IO_4000B`, которая определяет характеристики нагрузки и переключения для вентиля. Подсхемы всегда заканчиваются выражением «`ends`», как в строке 12.

Модель `D_CD4011B` может быть найдена в `CD4000.lib` и модели `IO_4000B` в `dig_io.lib`. Более подробную информацию можно найти в Справочное руководство по PSpice.

4.2. Цифровые цепи

Цифровые вентили по умолчанию не показывают свои контакты питания, потому что потребуется относительно большое количество проводов для соединения всех вентилях с питанием, что может привести к чрезмерному усложнению схемы. Вместо этого устройства TTL и CMOS подключены к глобальным узлам электропитания, которые не отображаются, и по умолчанию в них установлено значение 5 В. Различные источники питания могут быть установлены для диапазона напряжения питания от 3 до 18 В для устройств CMOS. Это не повлияет на ввод пороговых значений и выходные драйверы для CMOS-устройств, но задержки распространения будут все еще определяться для источника питания 5 В. Для точной установки задержки распространения временные модели должны быть изменены.

Чтобы установить цифровые логические уровни на выводах интегральной схемы (IC), рекомендуется использовать цифровые символы HI и LO в меню `Place> Power>`(выбрать библиотеку `Source.olb`) и использовать цифровые

подтягивающие (pullup) или стягивающие (pulldown) резисторы из библиотеки dig_misc, чтобы связать контакт с высоким или низким уровнем через резистор.

Символы No Connect из меню Place можно использовать для идентификации несвязанных выводов. На рисунке 4.2 показаны соответствующие символы и компоненты Capture.

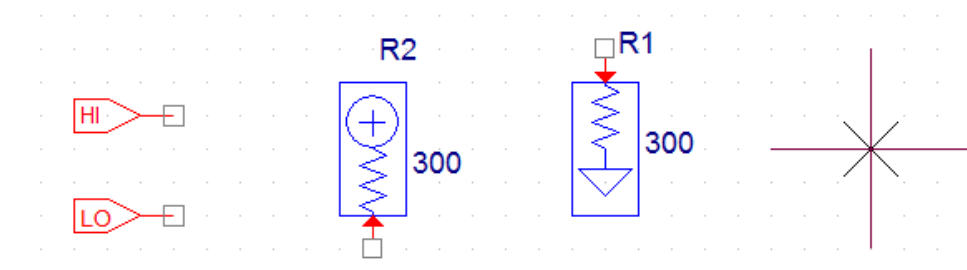


Рис. 4.2. Цифровые символы: HI, LO, pullup, pulldown, No Connect

4.2.1. Моделирование цифрового счетчика

Создадим новый проект pr-25 для моделирования цифрового счетчика. На рисунке 4.3 цифровой сигнал синхронизации подается ко входу 8-битного двоичного счетчик (U1A и U1B). Чтобы включить счетчик, ввод CLR привязан к минимуму, используя цифровой символ LO.

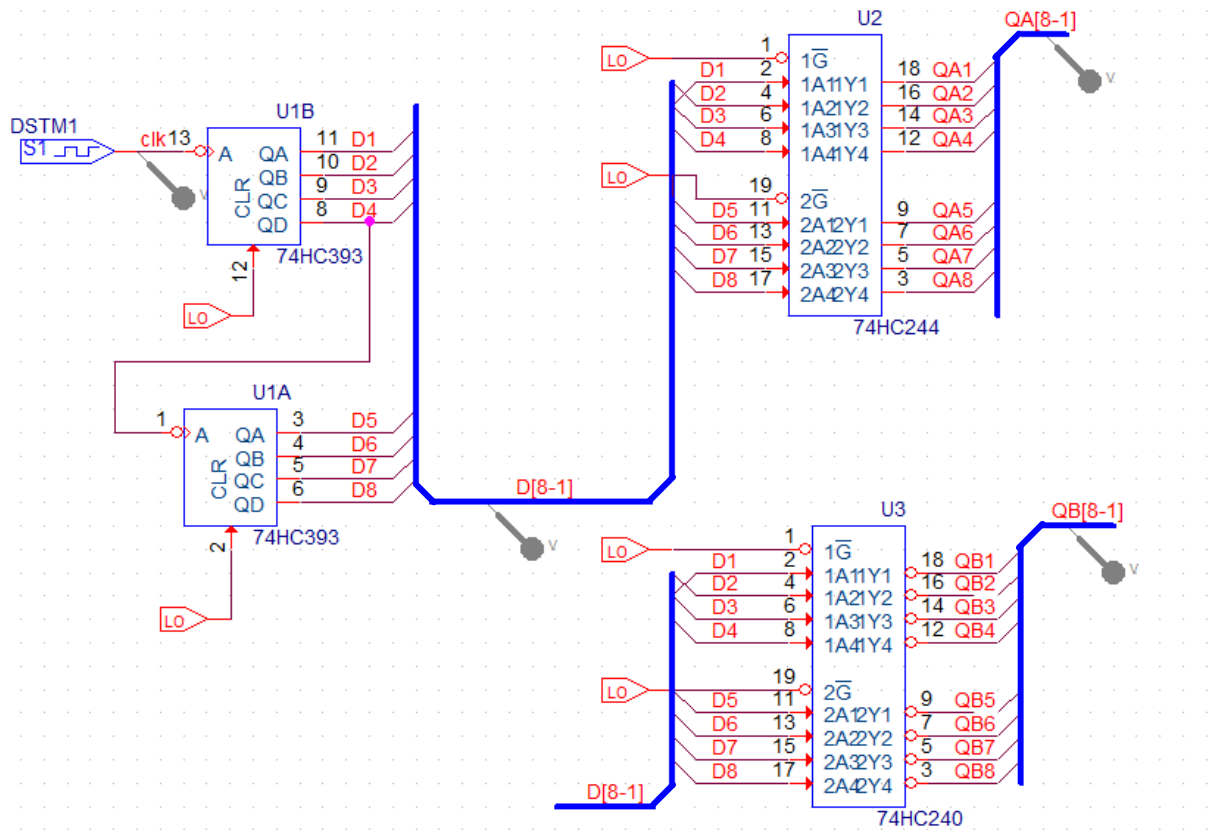


Рис. 4.3. Схема модели цифрового счетчика

Микросхемы соединены 8-разрядными шинами. Чтобы нарисовать шину, выберите Place>Bus или нажмите «В».

Чтобы нарисовать угловую шину, удерживайте Shift и нажмите левую кнопку мыши, чтобы определить угол, а затем нарисуйте шину.

Каждый выход счетчика подключается к 8-разрядной шине, используя точки входа в шину через Place> Bus Entry, выбрав значок или нажав E на клавиатуре.

Примечание:

Начиная с версии 16.3, соединительные контакты могут автоматически наноситься на шину. Нарисуйте шину, а затем выберите Place> Auto Wire > Connect to Bus. Нажмите на соединительный контакт и затем щелкните по шине

(вам будет предложено ввести имя сети). Вход в точку шины и провод будут нарисованы автоматически.

Каждый провод, подключенный к точке входа шины, обозначен D1, D2 и т.д., а сама шина имеет сетевое имя D[8-1], порядок которого, определяется msb-lsb (старший значащий бит – младший значащий бит). Шина на ввод данных в U3 также называется D[8-1] и поэтому будет подключена как 8-битная шина. Шина также может быть обозначена как D [7-0] или D[7..0], согласно вашему предпочтению, если крайний правый бит вы обозначаете нулевым.

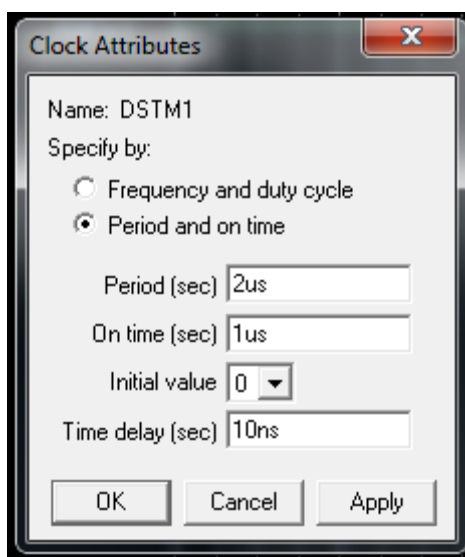


Рис. 4.4. Установка тактового генератора

Только сигналы одного типа могут быть сгруппированы вместе на шине, смешанные шины не могут быть определены в Capture. Однако в Probe сигналы различных типов могут собираться вместе и отображаться в виде сигнала шины.

Маркеры могут быть размещены как на шинах, так и на проводах.

В тактовом генераторе DSTM1 установим период 2 мкс, время включенного состояния 1 мкс, задержка 10 нс (рис. 4.4).

4.2.2. Профиль цифрового моделирования

Сначала установим режим Transient с временем моделирования 16 мкс. Затем на вкладке Options выбираем Gate Level Simulation>General и устанавливаем типичную синхронизацию (рис. 4.5).

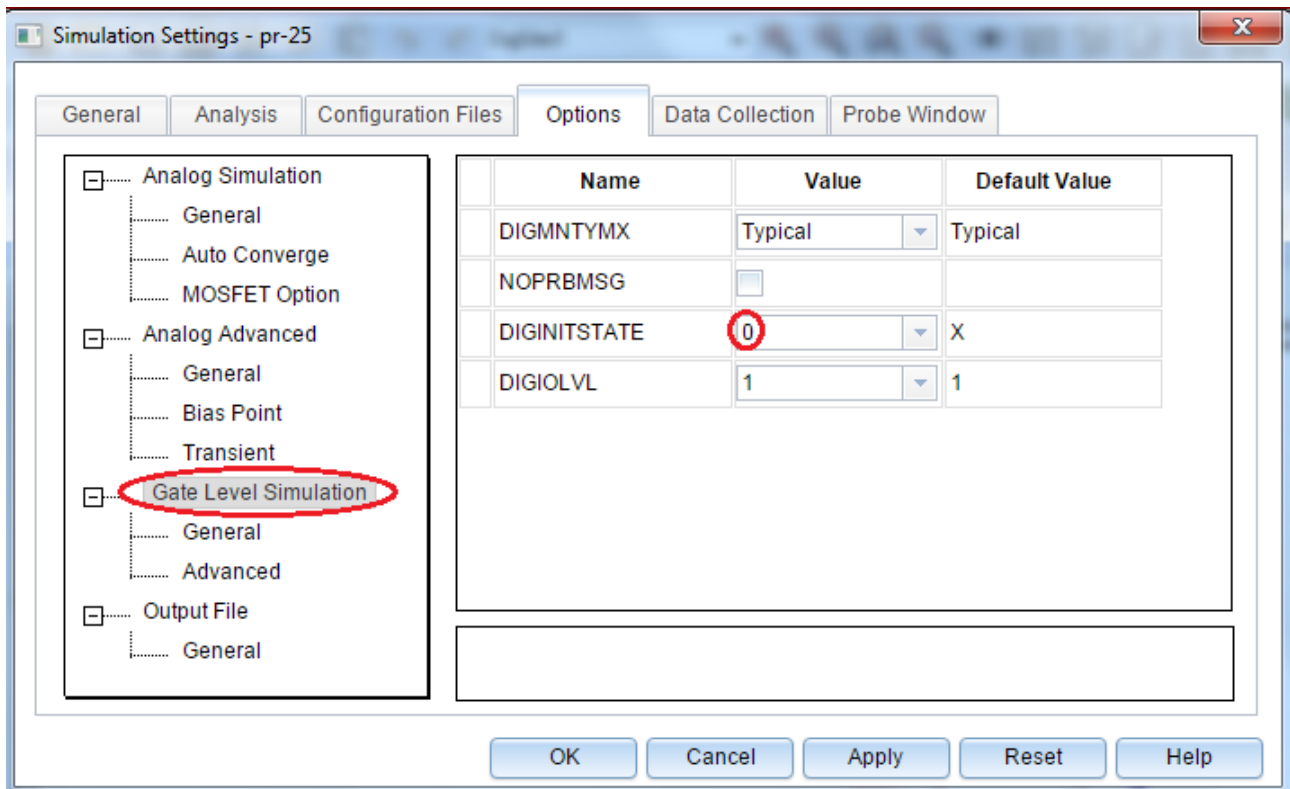


Рис. 4.5. Установка профиля цифрового моделирования

Параметр «Режим синхронизации» позволяет вам выбрать минимальные, максимальные или наихудшие временные характеристики для цифровых устройств. Есть четыре интерфейса ввода/вывода AtoD и DtoA, которые вы можете выбрать, и, что наиболее важно, вы можете инициализировать все триггеры либо на X (бесконечный импеданс), либо в состояние логического 0 или 1, установив параметр DIGINITSTATE.

4.2.3. Отображение цифровых сигналов

Цифровые сигналы отображаются как с высоким, так и с низким уровнем логики. Однако для областей двусмысленности, когда время перехода точно не известно, возрастающие и падающие переходы будут показаны желтым цветом. Неизвестные состояния отображаются как две красные линии и высокоимпедансные состояния будут показаны как три синие линии.

Примечание:

Одна из распространенных ошибок заключается в неправильной инициализации регистров (триггеров) в цепи, так что в результате появляются две красные линии, представляющие неизвестное состояние. Убедись, что вы инициализируете триггеры, как показано на рис. 4.5.

Вы можете группировать цифровые сигналы вместе и отображать их как шину в окне Probe. Имя шины можно создать в поле Trace Expression окна Add Trace. До 32 цифровых сигналов могут быть указаны в перечне msb to lsb, с основанием шестнадцатеричного (по умолчанию), десятичного, восьмеричным или двоичного представления.

Для примера:

{D4 D3 D2 D1}; myBus; d отобразит D4 - D1 (msb-lsb), обозначенные как myBus с десятичными числами;

{WR RD CE}; control; b будет отображать управление шиной в двоичном формате.

На рисунке 2.6 после моделирования показаны сигналы на разных шинах счетчика в разных исчислениях. Вывод сделан последовательным выполнением команды Trace>Add Ttace для следующих выражений Trace Expression:

{D[8:1]} {QA[8:1]} {QB[8:1]} {QA[8:1]};myBus; b {QB[8:1]}; myBus; d
{QA[8:1]}; myBus; h QA[8:1]

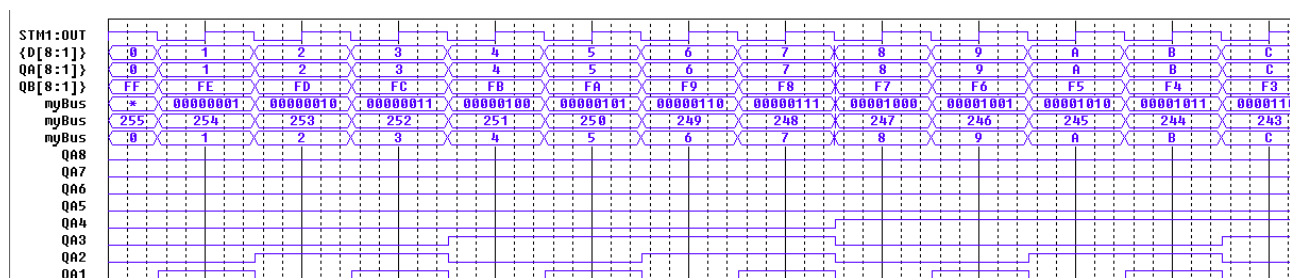


Рис. 4.6. Вывод данных в разных форматах

Примечание: в строке Trace Expression после записи очередного выражения не следует ставить какой-либо знак.

Данные на рис. 4.6 показывают увеличение счета на шинах D[8:1], QA[8:1], так как исходное состояние триггеров было DIGINITSTATE=0.

Установим в опциях профиля моделирования (рис. 4.5) DIGINITSTATE=1. В этом случае результаты моделирования на шинах D[8:1], QA[8:1] показывают счет в сторону уменьшения (рис. 4.7)

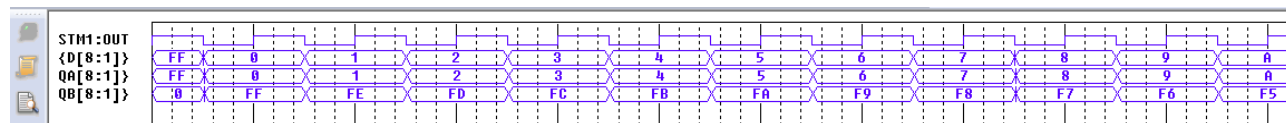


Рис. 4.7. Вывод данных для счета с уменьшением

Примитив MNTYMXDLY является необязательным параметром устройства, который указывает, следует ли использовать минимальные, типичные, максимальные или наихудшие цифровые временные значения задержки из модели устройства.

В свойствах компонента 74HC393 установлено PARAMS: MNTYMXDLY=0 (рис. 4.8). Это означает, что он принимает текущее значение параметра DIGMNTYMX. DIGMNTYMX в нашем профиле моделирования установлен как «типичный».

* 74HC393 COUNTER BINARY 4-BIT, ASYNCHRONOUS

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

*

* HIGH-SPEED CMOS LOGIC DATA BOOK, AUG 1989, TI

* JLS 6-30-92 REMODELED USING LOGICEXP, PINDLY, AND CONSTRAINT
DEVICES

*

.SUBCKT 74HC393 A_I CLR_I QA_O QB_O QC_O QD_O

+ OPTIONAL: DPWR=\$G_DPWR DGND=\$G_DGND

+ PARAMS: MNTYMXDLY=0 IO_LEVEL=0

Рис. 4.8. Фрагмент свойств счетчика 74HC393

Подробнее с основными и расширенными опциями цифрового моделирования можно познакомиться в PSpice 17.2 User Guide.

4.3. Контрольные вопросы

1. Какие основные цифровые устройства можно моделировать в PSpice?
2. Какие основные характеристики имеют цифровые модели ?
3. Какие цифровые примитивы использует PSpice ?
4. Какие две модели должны иметь цифровые примитивы ?
5. Поясните описание элемента И – НЕ.
6. Какие типовые цифровые символы используют в цифровых цепях ?
7. Как выполняют соединение цифровых микросхем многозарядными шинами ?
8. Как обозначают цифровые шиги ?
9. Как устанавливают профиль цифрового моделирования ?
10. Как можно отображать цифровые сигналы в окне Probe ?
11. Как записывают выражения Trace Expression для отображения цифровых сигналов в разных исчислениях и в разных форматах ?
12. Как можно изменять параметры цифрового компонента в списке его свойств ?

Глава 5. Смешанное моделирование

PSpice использует один и тот же механизм моделирования для аналоговых и цифровых схем. Результаты моделирования в окне Probe распределены на одной и той же оси времени, но разделяются на отдельные окна аналоговых и цифровых графиков. Аналоговые и цифровые компоненты в цепи соединены вместе в узлах. В PSpice существует три типа соединительных узлов:

- аналоговый, где все подключенные части являются аналоговыми;
- цифровой, где все подключенные части цифровые;
- и интерфейс, где есть смесь аналоговых и цифровых частей.

Узлы интерфейса автоматически разделяются на один аналоговый узел и один или больше цифровых узлов, вставляя аналоговые и цифровые интерфейсные подсхемы, которые являются аналого-цифровыми (ADC, AtoD) или цифро-аналоговыми (DAC, DtoA) интерфейсами.

Эти подсхемы также будут иметь собственный источник питания. Поскольку этот процесс происходит автоматически и работает незаметно, нам обычно не приходится беспокоиться о интерфейсных подсхемах, хотя они доступны в виде графиков в окне Probe.

5.1. Исследование аналогового компаратора с цифровым выходом

На рис. 5.1 показана схема аналогового компаратора. Выходной транзистор с открытым коллектором подключен к цифровому затвору. Подтягивающий резистор подключен к цифровому источнику питания и выходное заземление компаратора подключено к цифровому заземлению. Установки источника сигнала V1 показаны на схеме. Цифровой генератор стимулов DSTM1 имеет период 2 мкс.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

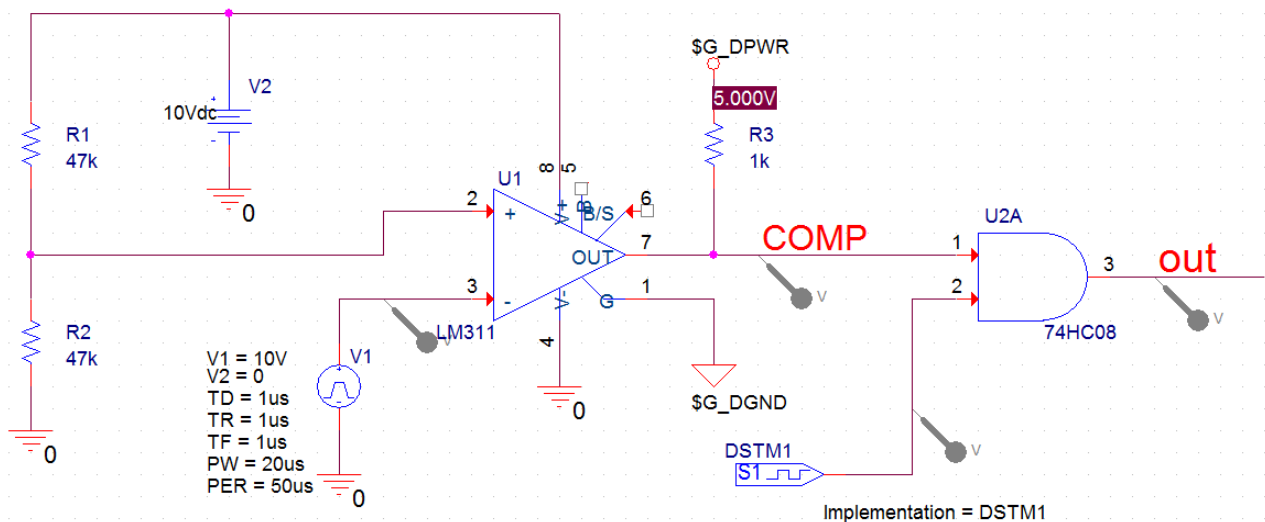


Рис. 5.1. Схема аналогового компаратора с цифровым выходом

Для смешанных аналоговых и цифровых схем выполняют те же процедуры для размещения деталей, создания профиля моделирования и моделирования.

Для упрощения цепей питания используют свойство PSPICEDEFAULTNET. Это свойство определяет сетевое имя, на которое подается питание или земля (невидимый) контакт.

Например, если контакты питания и заземления на цифровой части подключены к цифровым цепям \$ G_DPWR и \$ G_DGND соответственно, то свойства определяются следующим образом:

PSPICEDEFAULTNET = \$ G_DPWR

PSPICEDEFAULTNET = \$ G_DGND

Так для микросхемы 74HC08 в редакторе свойств модели (рис. 5.2) указаны обозначения цифровых цепей питания и земли. Описание определения подсхемы, которое начинается с:

.SUBCKT 7400 A B Y

+ optional: DPWR=\$G_DPWR DGND=\$G_DGND

```
+ params: MNTYMXDLY=0 IO_LEVEL=0.
```

Затем указываются соответствующие свойства компонента:

```
IMPLEMENTATION = 7400
MNTYMXDLY = 0
IO_LEVEL = 0
PSPICETEMPLATE = X^@REFDES %A %B %Y %PWR %GND
@MODEL PARAMS:IO_LEVEL=@IO_LEVEL
MNTYMXDLY=@MNTYMXDLY
```

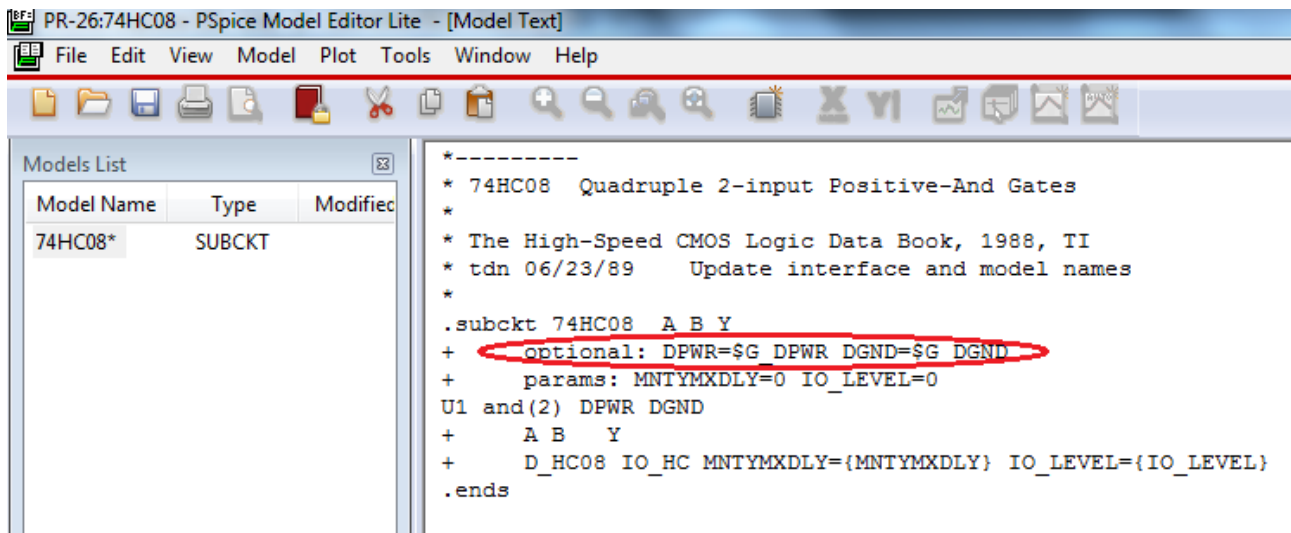


Рис. 5.2. Обозначения цифровых цепей питания и земли

После вызова подсхемы интерфейса PSpice A/D вставляет один или несколько интерфейсных блоков питания. Например, PSpice A/D вставил DIGIFPWR, который является источником питания подсхем, используемых всеми TTL-моделями в библиотеке моделей. DIGIFPWR создает глобальные узлы \$ G_DPWR и \$ G_DGND, которые используются по умолчанию для каждого компонента части TTL. Для проверки на рис. 5.1 показано измеренное напряжение в узле \$ G_DPWR.

Выполним моделирование, используя профиль Transient с длительностью 100 мкс и максимальным шагом 10 нс.

На рис. 5.3 показаны цифровые формы сигналов, изображаемые в верхней области Probe и аналоговые сигналы, изображенные в нижней области.

В схеме (рис. 5.1) напряжение на неинвертирующем входе компаратора составляет $\sim 5\text{В}$. Если $V1 > 5\text{В}$, компаратор имеет на выходе низкий уровень и на выходе элемента «И» (74НС08) тактовые импульсы отсутствуют.

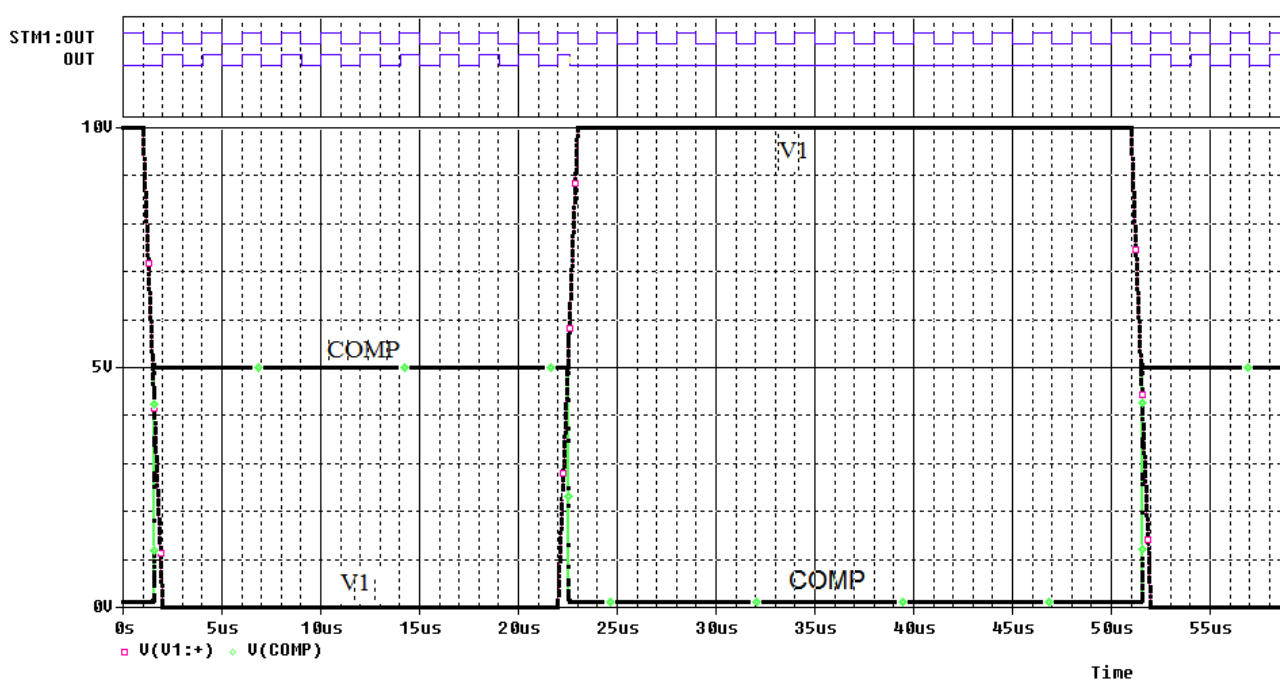


Рис. 5.3. Цифровые и аналоговые сигналы компаратора

5.2. Исследование цифро-аналогового преобразователя

На рисунке 5.4 показана функциональная схема цифро-аналогового преобразователя AD7224 (ЦАП).

ЦАП AD7224 - это высокоточный 8-разрядный цифро-аналоговый преобразователь с выходным усилителем и двойным буфером интерфейсной логики на монолитном CMOS-чипе. Не требуется внешних подстроек для достижения полной заданной производительности компонента. Логика

интерфейса с двойной буферизацией состоит из двух 8-битных регистров - входной регистр и регистр ЦАП. Только данные, содержащиеся в регистре ЦАП, определяют аналоговый выход преобразователя. Двойная буферизация позволяет одновременно обновлять систему, содержащую несколько AD7224. Оба регистра могут быть доступны под управлением трех внешних линий CS, WR и LDAC. Когда оба регистра доступны, линия RESET выполняет функцию установки нуля. Эта функция полезна для циклов калибровки системы. Все логические входы - это уровни совместимые с TTL и CMOS (5 В), а логика управления по скорости совместима с большинством 8-разрядных микропроцессоров.

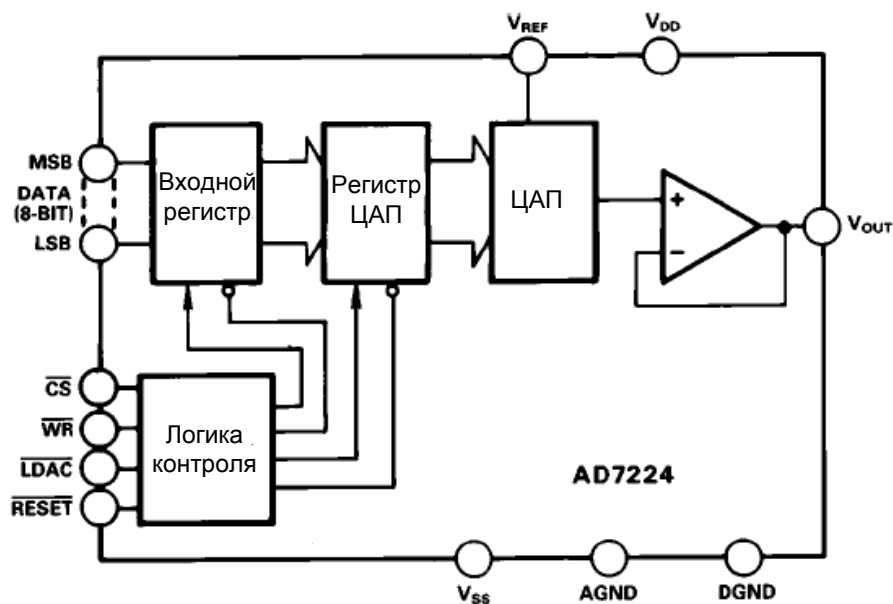


Рис. 5.4. Функциональная схема ЦАП AD7224

Схема модели цифро-аналогового преобразователя показана на рис. 5.5.

На вход подано цифровое информационное слово 0111 1111. Из спецификации производителя напряжение определяется:

$$V_0 = V_{REF} \cdot \frac{127}{256} = 4,96 B$$

Циклы синхронизации ЦАП должны быть настроены в соответствии с технической спецификацией.

1. Настройте переходный анализ со временем моделирования 5 мс и максимальным шагом 10 нс.

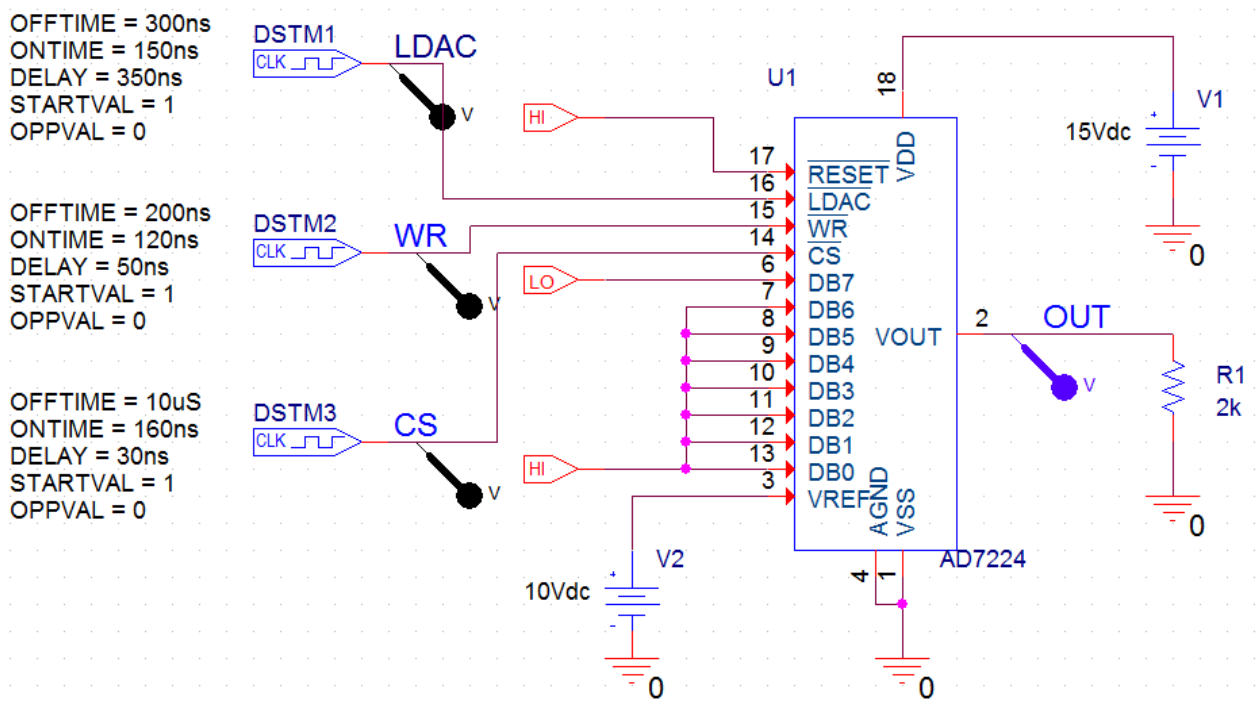


Рис. 5.5. Схема модели ЦАП

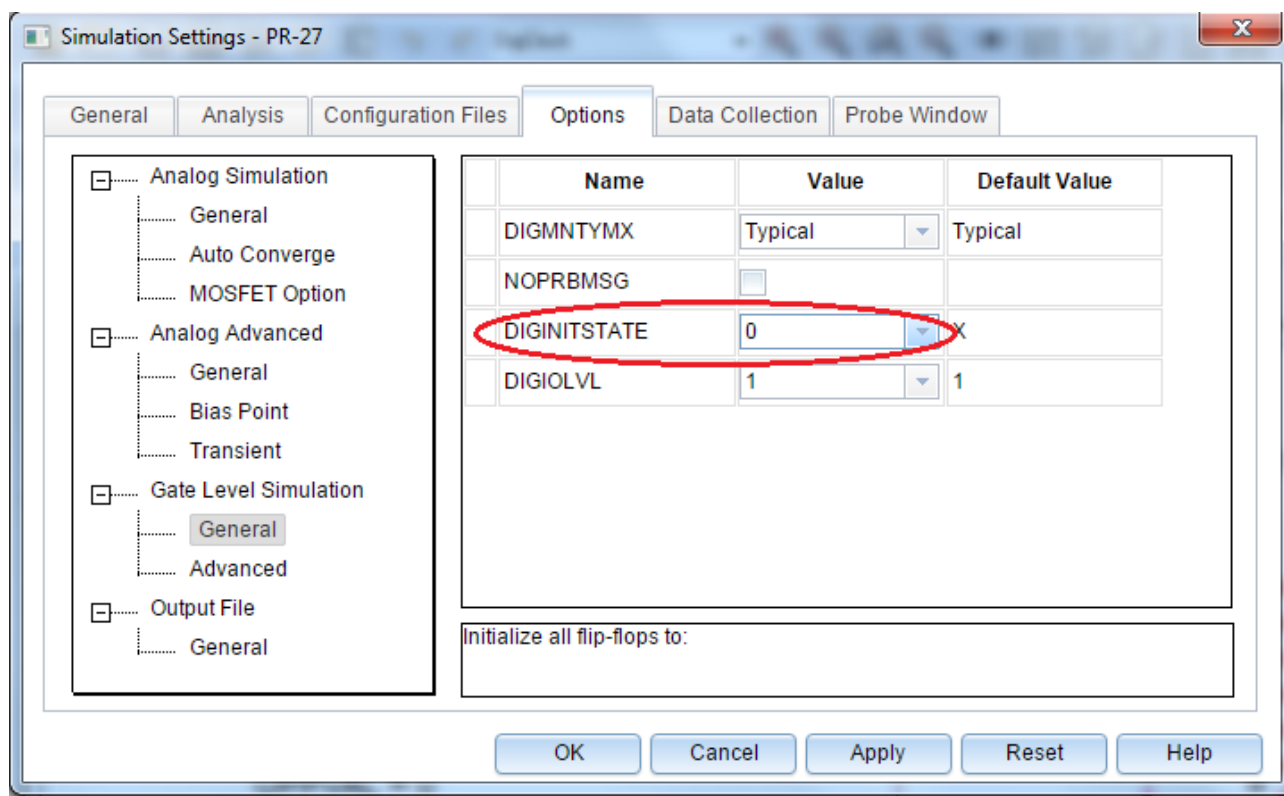


Рис. 5.6. Установка нулевого состояния цифровых схем

2. Перейдите на вкладку Options и выберите категорию Gate Level Simulation> General. Установите начальное значение цифровых схем нулевым: DIGINITSTATE = 0 (рис. 5.6). Закройте профиль моделирования.
 3. Поместите маркеры напряжения на цепях LDAC, WR, CS и OUT.
 4. Выполните моделирование.
- Результаты показаны на рис. 5.7.

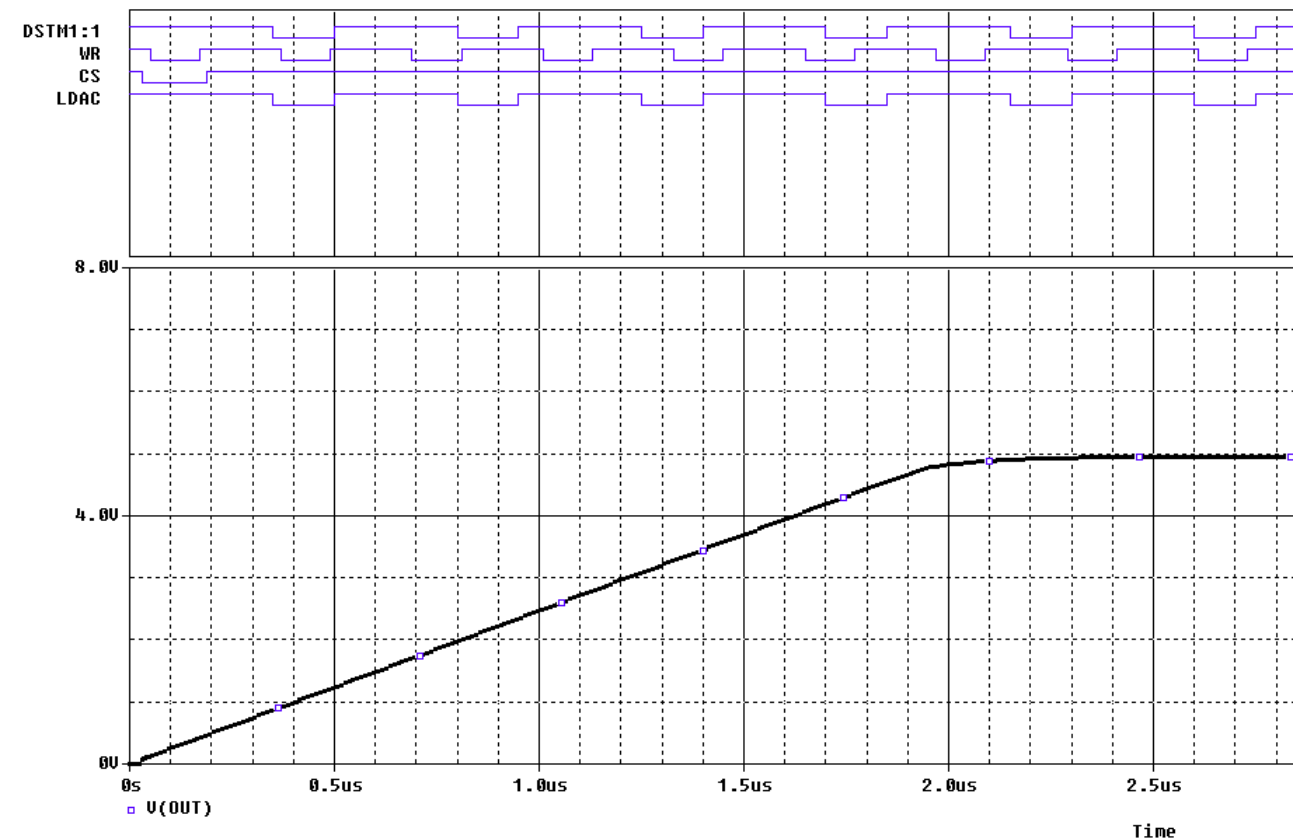


Рис. 5.7. Результаты моделирования ЦАП

В окне Probe вы увидите, что верхние графики показывают цифровые сигналы, а нижний график показывает аналогового выходной сигнал, достигающий значения ~ 5 В.

Установка программного обеспечения Cadence \ OrCAD включает в себя хороший выбор примеров аналоговых, цифровых и смешанных схем в каталогах anasim, digsim и mixsim. Их можно найти в установочном каталоге, например:

< install path> \ Cadence \ SPB_17.2\ tools \ pspice \capture_samples \

< install path> \ Cadence \ OrCAD_17.2 \ tools \ pspice \capture_samples \

5.3. Контрольные вопросы

1. Как представляются результаты смешанного моделирования аналоговых и цифровых схем в окне Probe ?
2. Какие типы соединительных узлов использует PSpice ?
3. Поясните работу модели аналогового компаратора с цифровым выходом.
4. Как подводят питание к цифровым компонентам и что следует указать в свойствах модели компонента ?
5. Поясните форму цифровых и аналоговых сигналов в модели компаратора.
6. Поясните назначение управляющих сигналов ЦАП AD7224.
7. Как вычисляют выходное напряжение ЦАП для известного цифрового сигнала ?
8. Поясните настройку модели ЦАП.
9. Поясните форму графиков результатов моделирования ЦАП.

Глава 6. Создание иерархических проектов

Вводимая для моделирования принципиальная схема может быть либо плоской, в которой сигналы соединены между страницами схемы, или иерархической, в которой схема разделена на блоки и сигналы соединены поперечно вверх и вниз по иерархии.

Создадим новый проект FullAdd (полный сумматор) в папке PR-28. Плоские схемы представлены в Менеджере проекта как имеющие одну схемную папку с рядом ассоциированных страниц, в то время как иерархические схемы будут иметь более чем одну схемную папку (Рис. 6.1а).

6.1. Создание иерархического проекта

В окне менеджера проекта создаются файл fulladd.dsn. Ниже дизайн-файла, создаётся папка с именем SCHEMATIC1. Эта папка имеет страницу схемы названную PAGE1.

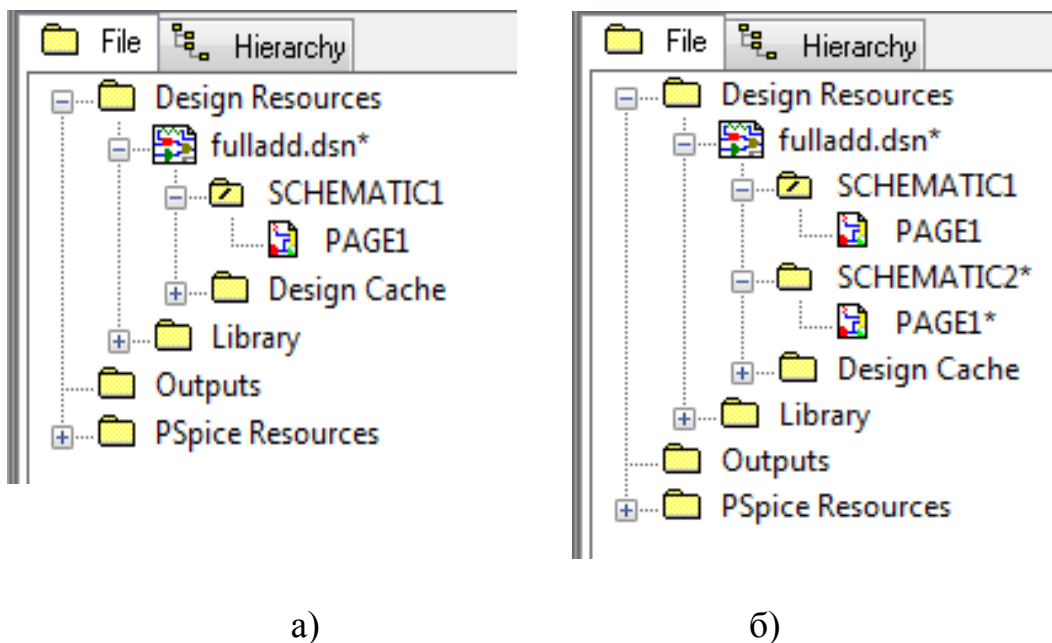


Рис. 6.1. Структура плоского проекта (а) и иерархического проекта (б)

Электрические схемы большинства проектов размещаются на нескольких страницах не самого большого формата. Имеются два способа организации схем большого объема:

- плоские обычные многостраничные структуры;
- иерархические структуры.

Электрические цепи, расположенные на разных страницах многостраничной схемы, соединяются друг с другом с помощью так называемых межстраничных соединителей (off-page connectors), имеющих одинаковые имена. Все страницы таких схем содержатся в одной папке на одном и том же уровне. Их структура показывается в менеджере проектов при нажатии клавиши File. Например, на рис. 6.1 в папке SCHEMATIC 1 могут быть помещены страницы схемы PAGE1 и PAGE2.

В иерархических проектах каждая схемная папка в иерархии будет представлена иерархическим блоком в схемной папке. Выбрав иерархический блок, вы выбираете основную схему и эффективно спускаетесь вниз по иерархии. Для плоской схемы (рис. 6.1а), есть одна схемная папка и одна или несколько страниц. Для иерархической схемы (рис. 6.1б) может быть две и более схемные папки в иерархии и каждая со своей собственной схемной страницей или страницами. Каждую папку мы создаем, выбрав в меню проекта FullAdd опцию New Shematic и задав название новой папки.

На схемах **иерархических** проектов размещаются специальные символы, называемые иерархическими блоками (hierarchical block). Принципиальная электрическая схема каждого такого блока размещается в виде отдельной схемы, помещаемой в папку на том же уровне иерархии, что и основная схема. Иерархическая структура показывается в менеджере проектов при нажатии клавиши **Hierarchy**.

6.1.1. Создание плоского проекта полусумматора HalfAdd

Создадим сначала плоский проект для моделирования цифрового полусумматора HalfAdd. Для полного сумматора полусумматор является проектом нижнего уровня.

Переименуем папку SHEMATIC1 и PAGE1, назвав их HALFADD (рис. 6.2).

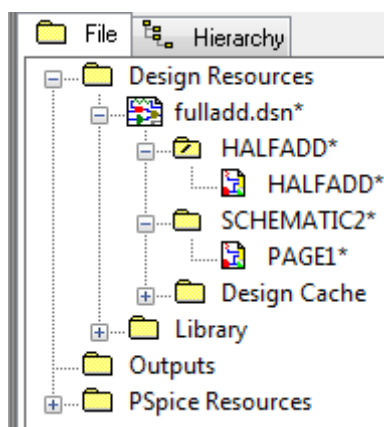


Рис. 6.2. Структура проекта с полусумматором

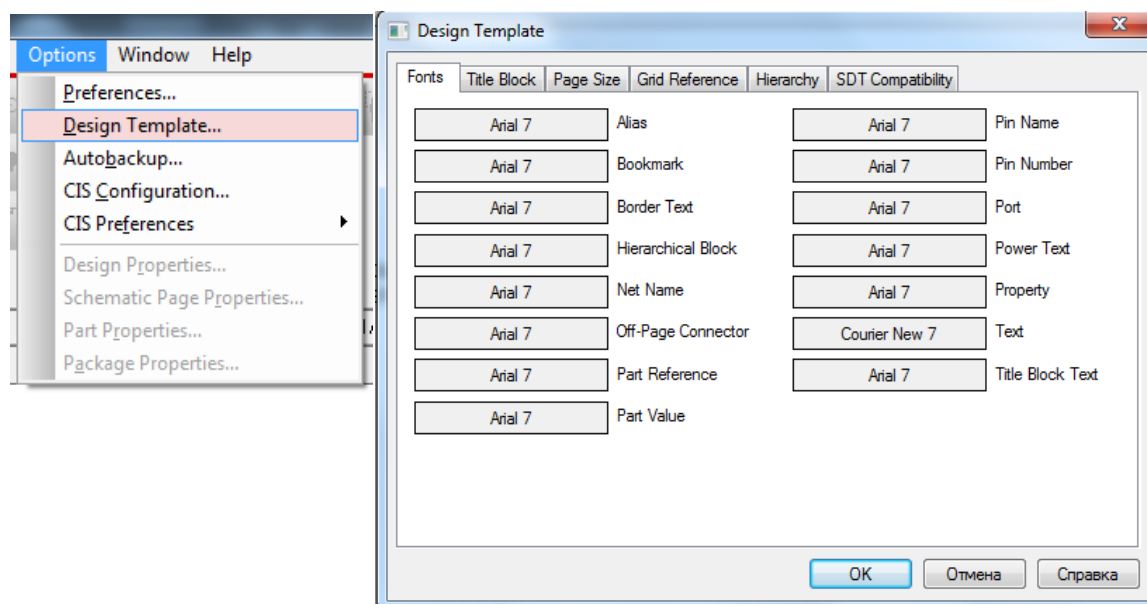


Рис. 6.3. Настройка шаблона дизайна

Напомним, что перед тем, как начать создания дизайна в OrCAD Capture, вы можете указать по умолчанию характеристики для вашего проекта с использованием шаблона дизайна. Шаблон дизайна может использоваться для указания шрифтов по умолчанию, размера страницы, заголовков блока, сетки и так далее. Чтобы настроить шаблон дизайна в OrCAD Capture, используйте диалоговое окно Design Template.

Для того, чтобы открыть диалоговое окно Design Template, из меню Options выберите Design Template (рис. 6.3).

6.1.2. Иерархические порты и off-page разъемы

Так как полусумматор будет входить в более сложный иерархический проект, рассмотрим новые компоненты для различных соединений.

В иерархических схемах и в случае плоских схем обычно имеется одна папка и одна или более страниц. Для подключения сигналов между страницами, используются внестраничные разъемы: Place > Off-Page Connectors (рис 6.4). Два типа разъемов используют для указания направления потока данных, то есть от входа к выходу. Когда провод подключается к внестраничному разъему, схемное название провода наследует имя соединителя.

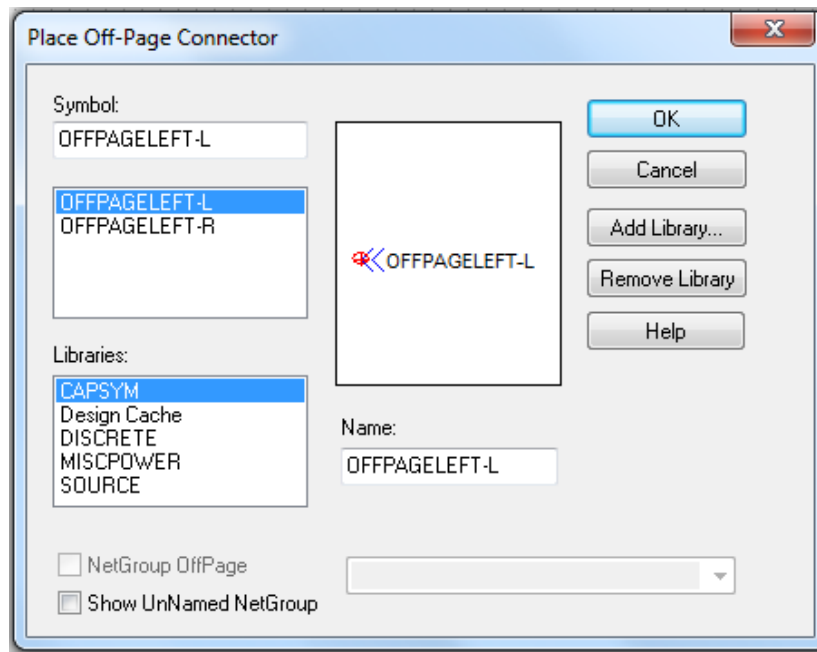


Рис. 6.4. Внестраничные порты

Иерархические порты подключают сигналы между уровнями иерархии. Для установки таких портов надо выбрать Place > Hierarchical Ports (Рис. 6.5). Как и с внестраничными разъемами, провод подключенный к иерархическому порту наследует имя порта.

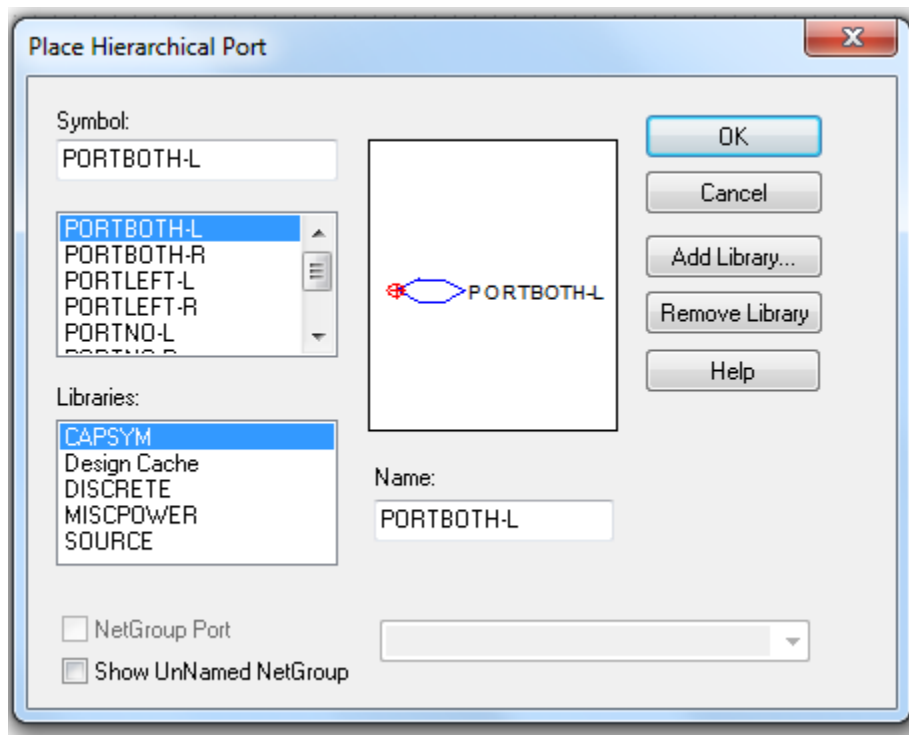


Рис. 6.5. Размещение иерархических портов

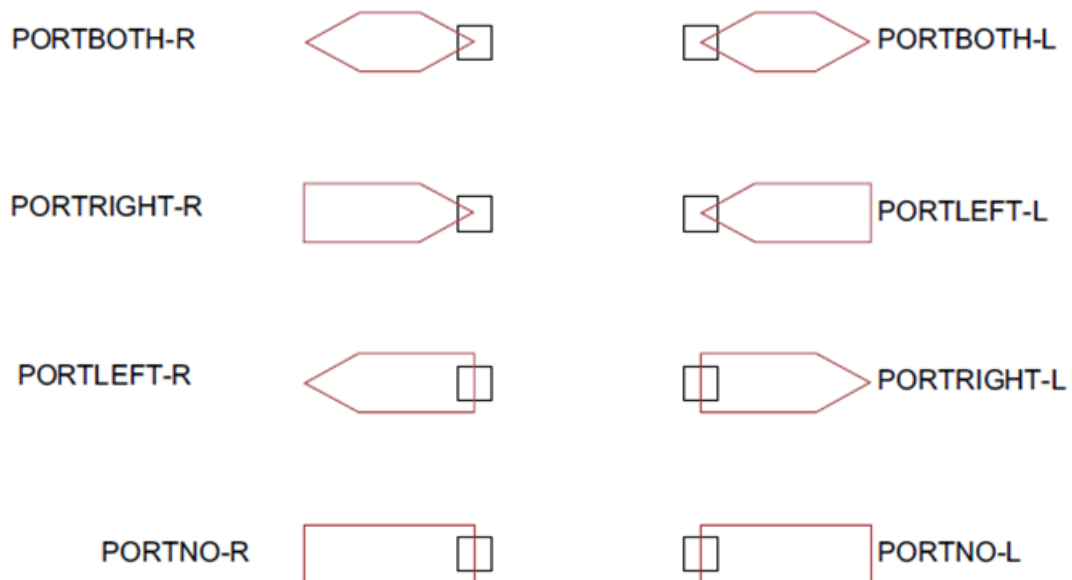


Рис. 6.6. Различные виды иерархических портов

Доступны различные иерархические порты, отличающиеся типом порта и направлением потока данных. Рис. 6.6 показывает различные типы иерархических портов.

ских портов. Для примера: PORTRIGHT-R представляет собой порт, который имеет направление передачи вправо и имеет соединение на правой стороне.

В этом разделе мы создадим простой плоский полусумматор с портами X и Y в качестве входов и SUM и CARRY в качестве выходов.

1. Требуется создать схему (рис. 6.7) из библиотеки Parts.

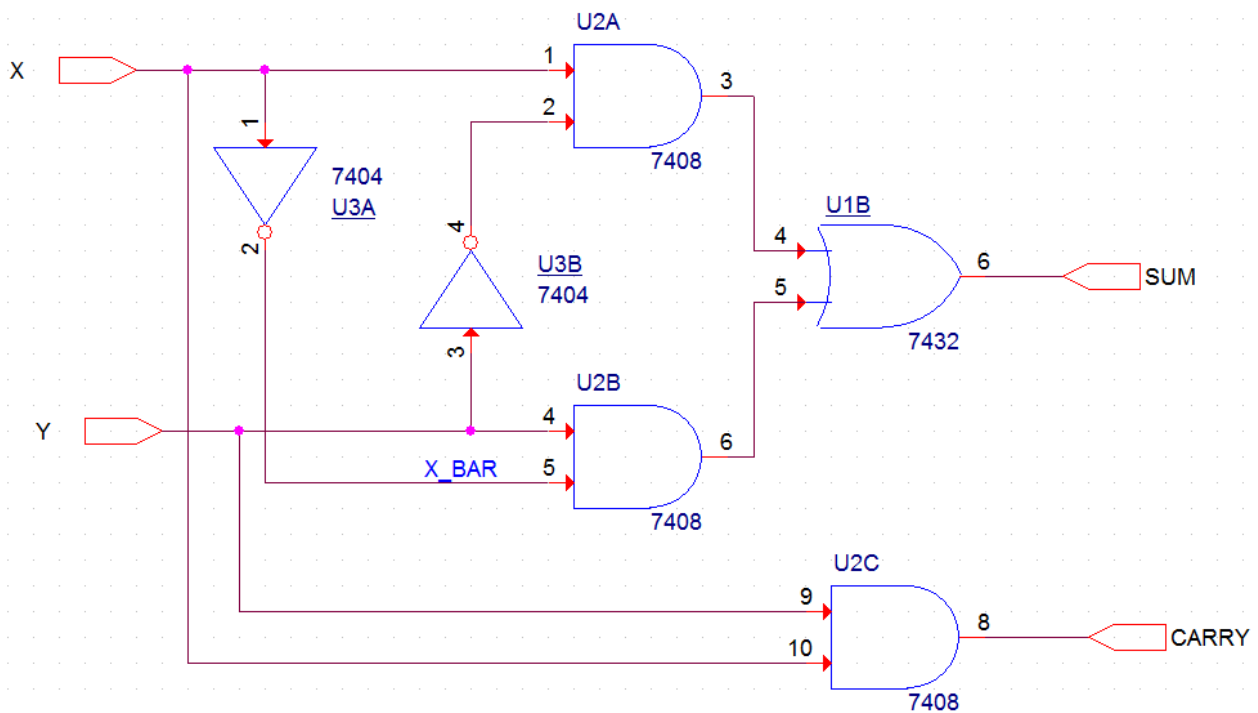


Рис. 6.7. Схема простого полусумматора

2. В меню Place в Capture, выберите Part. В диалоговом окне Place Part, сначала выберите библиотеку вентилях 7400.OLB, из которой должны быть добавлены компоненты, а затем добавьте их на страницу схемы.

3. Из меню Place выбираем иерархические порты:

входные порты из библиотеки CAPSUM : PORTRIGHT-R;

выходные порты из библиотеки CAPSUM: PORTLIGHT-L (рис. 6.6).

3. Выполняем соединения и переименовываем порты. Получаем нужную схему полусумматора (рис. 6.7).

6.1.3. Создание иерархического проекта Full Adder

В Capture можно создать иерархические проекты, используя один из следующих способов:

- восходящий метод (Bottom-up method);
- нисходящий метод (Top-down method).

Другой метод создания иерархического дизайна состоит в том, чтобы создавать компоненты или символы для дизайна на самом низком уровне и сохранять символы в определённой пользователем библиотеке. вы можете позже добавить определённую пользователем библиотеку в ваши проекты и использовать эти символы на схеме. Например, вы можете создать компонент для дизайна полусумматора, а затем вместо иерархических блоков, использовать эту часть в новой схеме.

Более подробно мы рассмотрим этот подход в разделе «Создание компонентов для схем».

В этом разделе мы создадим иерархическую структуру полного сумматора. Конструкция с половинным сумматором, созданная в разделе «Создание плоского проекта полусумматора HalfAdd» будет использоваться в качестве дизайна самого низкого уровня.

6.1.4. Восходящий метод

При создании иерархической конструкции с использованием восходящей методологии, необходимо выполнить следующие действия.

- Создание схемы низшего уровня.
- Создание схем более высокого уровня, которые будут содержать низкоуровневыми схемы в виде иерархических блоков.

Выполняем следующие шаги:

1. Создание проекта в Capture по известной нам методике.

2. Создание схемы низшего уровня. В схеме полного сумматора, например, дизайном нижнего уровня является полусумматор.

Для того, чтобы пройти шаги по созданию дизайна полусумматора, надо повторить действия из разделов 6.1.2, 6.1.3. Можно вместо этого создать новый проект fulladd-2 на основе ранее созданного проекта fulladd.

3 Создание конструкции более высокого уровня. Надо создать схему для полного сумматора, который использует полусумматор, созданный на предыдущем шаге. Нужные шаги приведены в следующем разделе.

6.1.5. Создание схемы полного сумматора

1. В окне менеджера проекта, щёлкните правой кнопкой мыши на fulladd.dsn и выберите New Schematic.

2. В диалоговом окне Schematic укажите имя новой схематической папки как FULLADD и нажмите OK (рис. 6.8).

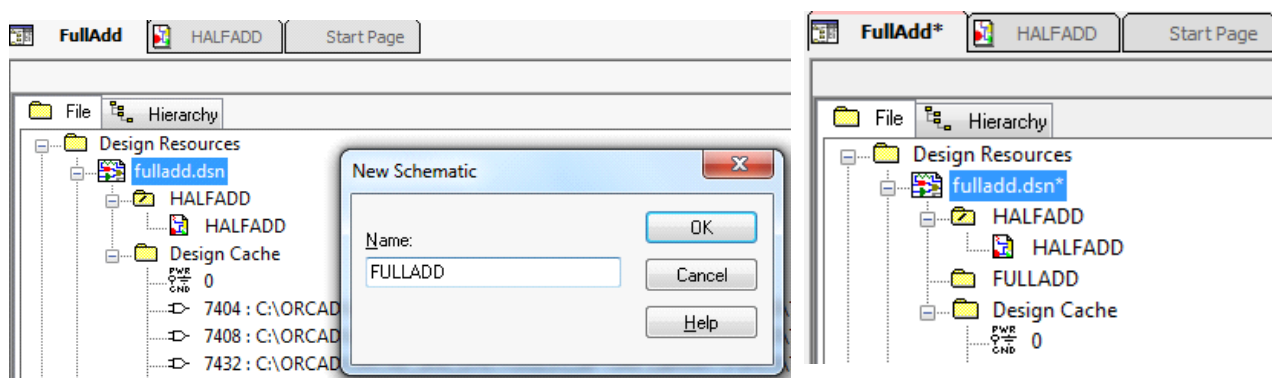


Рис. 6.8. Создание папки FULLADD

В окне диспетчера проекта, папка FULLADD появляется ниже fulladd.dsn.

3. Сохраните дизайн.

4. Для того, чтобы сделать схему полного сумматор в качестве корневого дизайна (проект верхнего уровня), щёлкните правой кнопкой мыши на FULLADD и из всплывающего меню выберите Make Root.

Папка FULLADD движется вверх и в папке появляется передний слэш (рис. 6.9).

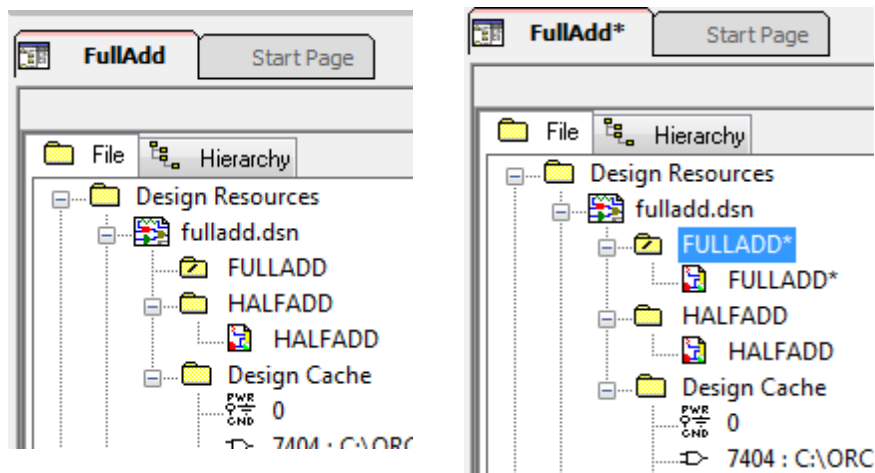


Рис. 6.9. Перемещение папки FULLADD вверх

5. Щелкните правой кнопкой мыши на FULLADD и выберите New Page.

6. В новой странице в схеме: в окне диалога FULLADD укажите название страницы, как FULLADD и нажмите OK. Новая страница FULLADD добавляется ниже схемной папки FULLADD.

7. Дважды щёлкните страницу FULLADD, чтобы открыть её для редактирования.

8. В меню Place выберите Hierarchical Block.

9. В диалоговом окне Place Hierarchical Block, укажите ссылку на HALFADD_A1 (рис. 6.10).

10. Укажите тип осуществления как Schematic View.

11. Укажите имя осуществления как HALFADD и нажмите кнопку OK.

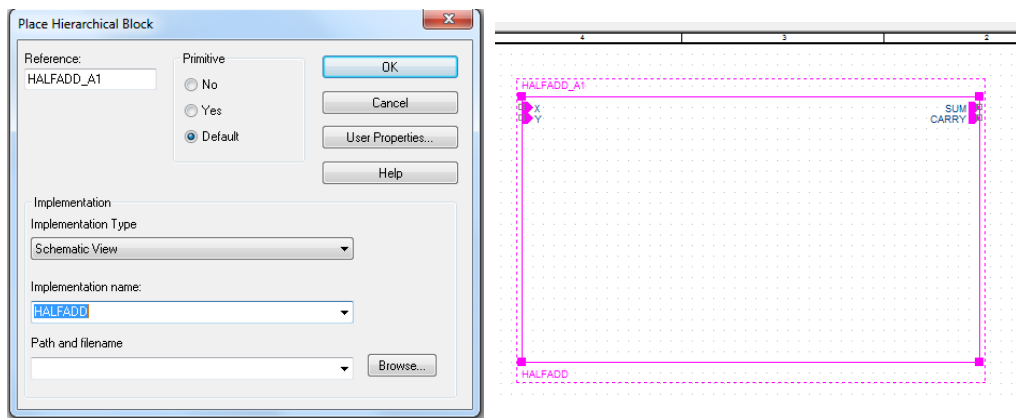


Рис. 6.10. Создание иерархического блока

Курсор изменится на перекрестие.

12. Нарисуйте прямоугольник на странице схемы.

Иерархический блок с входными и выходными портами будет нарисован на странице.

13. При необходимости измените размер блока. Кроме того, переместите входные и выходные порты на блоке путем перетаскивания.

Примечание: Для того, чтобы проверить, является ли иерархический блок правильным, выделите его, щёлкните правой кнопкой мыши на блоке и выберите Descend Hierarchy. Должна появиться ранее созданная схема полусумматора (рис. 6.11).

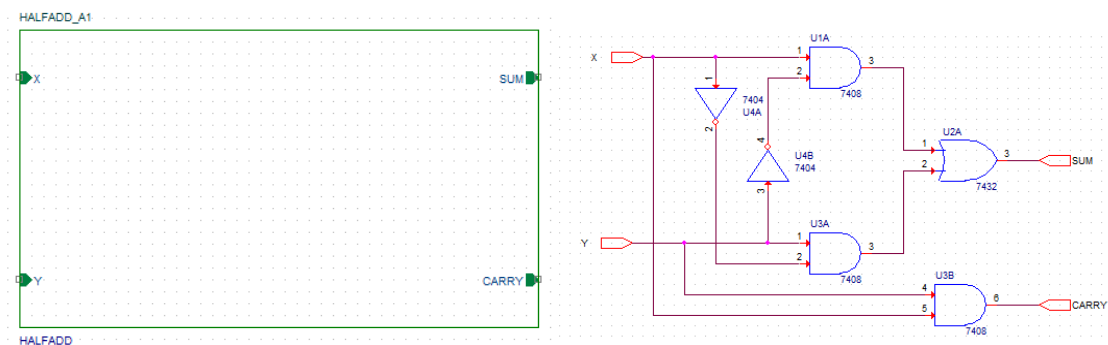


Рис. 6.11. Проверка правильности иерархического блока

14. Поместите другой экземпляр иерархического блока на схематическую страницу. Для этого :

- Выделите иерархический блок.
- В меню «Правка» выберите команду «Копировать».
- В меню «Правка» выберите команду «Вставить».
- Поместите экземпляр блока в желаемое место нахождения.

Примечание: В качестве альтернативы, вы можете использовать <Ctrl> + <C> и <Ctrl> + <V> для копирования и вставки блока.

15 По умолчанию позиционное обозначение для второго иерархического блока будет HALFADD_A2. Дважды щёлкните на позиционное обозначение, чтобы изменить ссылку на HALFADD_B1 (рис. 6.12).

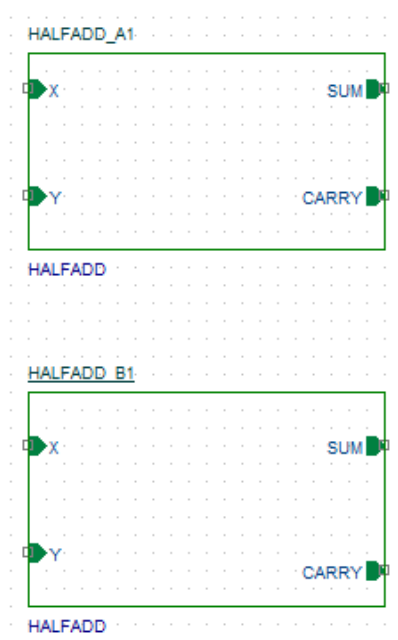


Рис. 6.12. Установка второго иерархического блока

Двойной щелчок на блоке раскроет его содержимое.

2. Используя Place Part, добавьте вентили ИЛИ (7432) к схеме и соедините блоки проводниками, как показано на рис. 6.13.

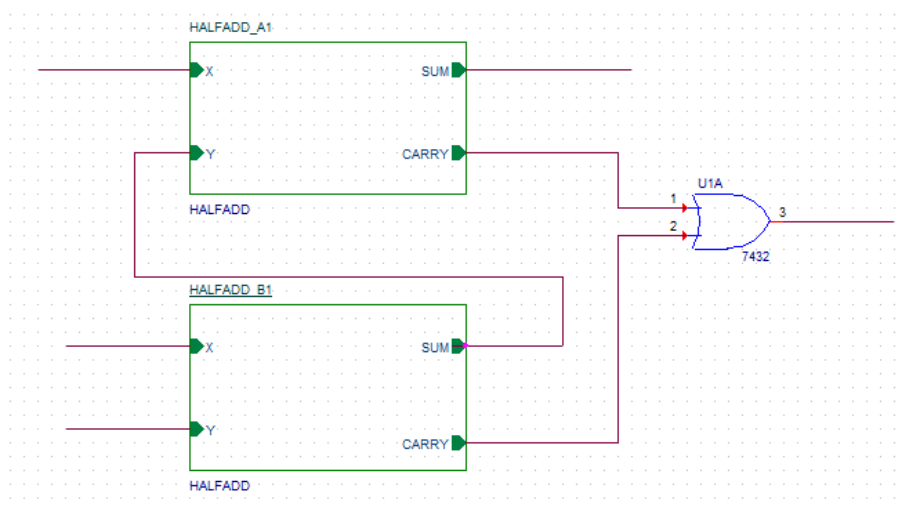


Рис. 6.13. Соединения иерархических блоков

17. Добавьте стимул к дизайну. В диалоговом окне Place Part, используйте кнопки Add Library для добавления к дизайну библиотеки SOURCSTM. OLB.

Эта библиотека находится в <install_dir>/tools/capture/library/pspice.

18. Из Part List выберите DigStim1 и нажмите кнопку OK. Символ прикрепляется к курсору.

19. Поместите символ на трех входных портах: порт X на HALFADD_A1, порт X и Y на HALFADD_B1.

20. Щёлкните правой кнопкой мыши на схеме и выберите End Mode.

21. Укажите значение свойства по осуществлению как Carry, X и Y, соответственно (рис. 6.14).

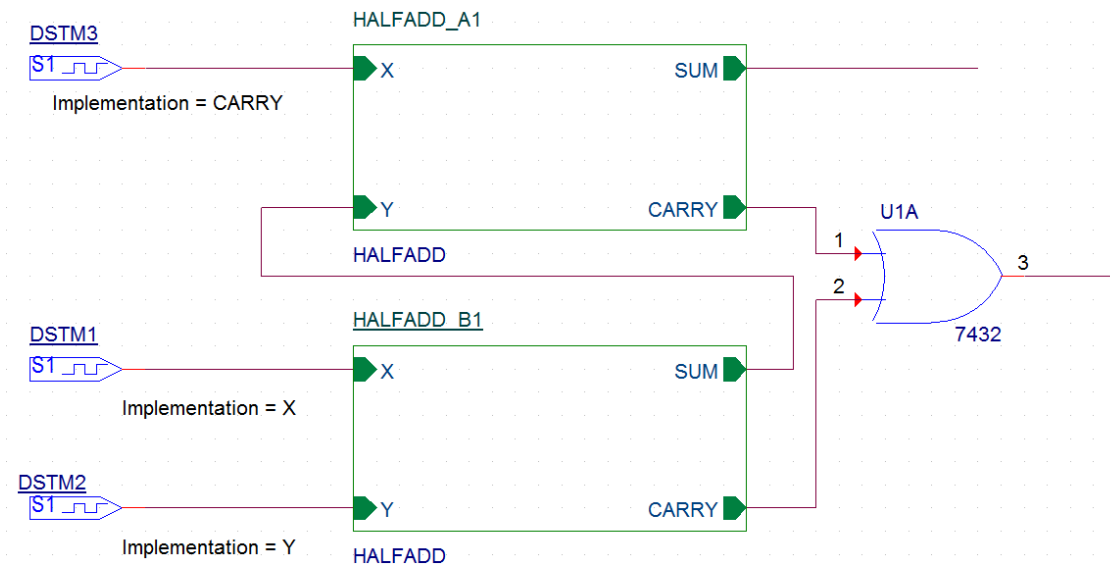


Рис. 6.14. Установка стимулов в проект

22. Выберите Place Part, чтобы добавить выходной порт CARRY_OUT на выходе логического элемента ИЛИ (рис. 6.15).

Для этого:

- Из списка библиотек выберите CAPSYM.
- Из списка символов выберите PORTLEFT-L и нажмите кнопку ОК.
- Поместите выходной порт, как показано на рис. 6.15.
- Дважды щёлкните имя порта и измените название порта на CARRY_OUT.

30. Сохраните дизайн.

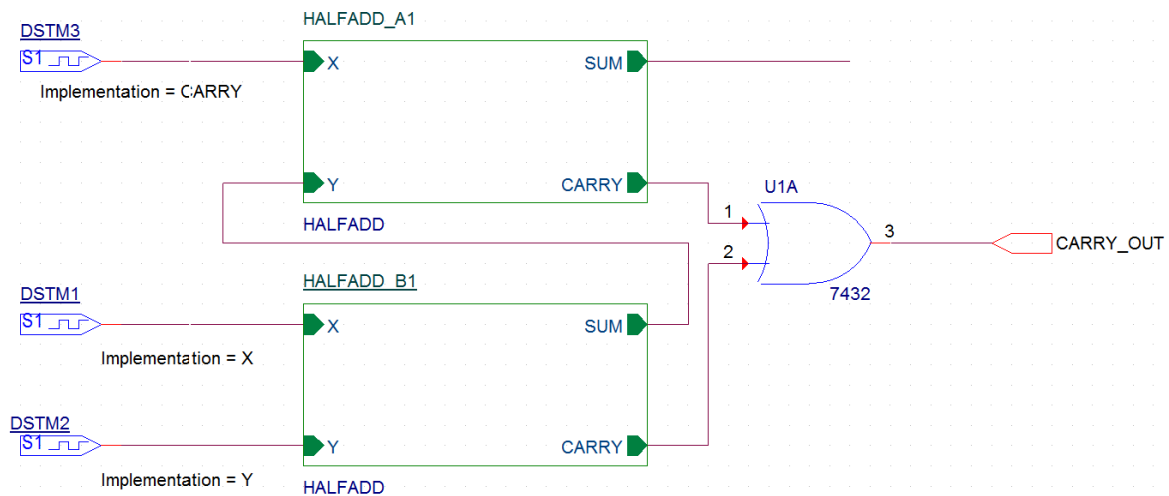


Рис. 6.15. Добавление выходного порта CARRY_OUT

6.1.6. Добавление в проект аналоговых компонентов

Мы только что добавили цифровые компоненты к конструкции. Полная схема проекта показана на рис. 6.16.

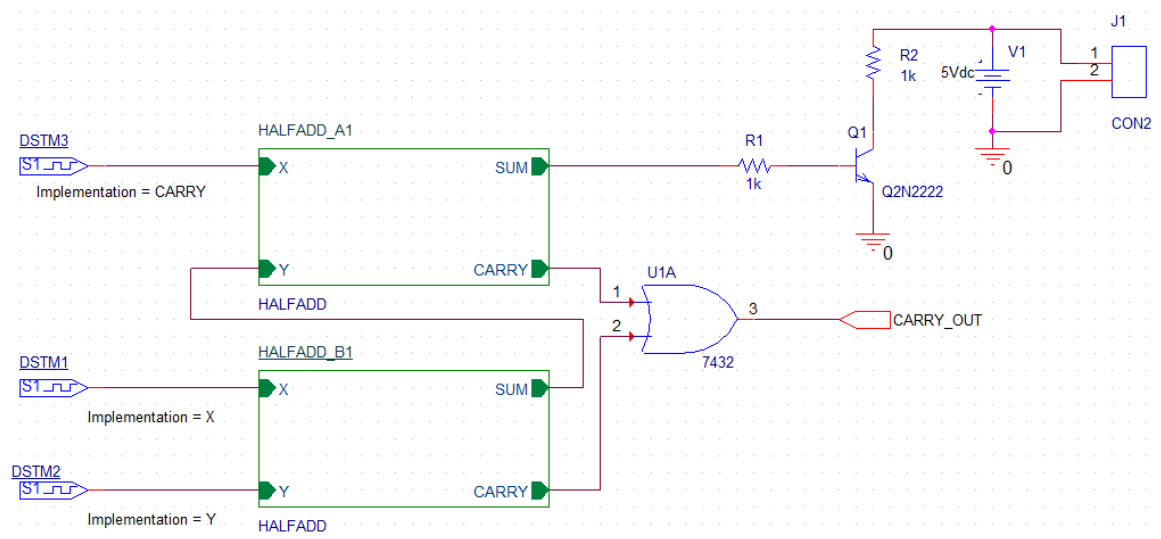


Рис. 6.16. Полная схема проекта FULLADD с аналоговыми компонентами

Резисторы можно взять из библиотеки PSpice Components или ANALOG.OLB.

Транзистор Q2N2222 находим в библиотеке EVAL, добавляем эту библиотеку в проект и помещаем транзистор в схему.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Источник напряжение 5Vdc можно взять из библиотеки PSpice Components.

Из библиотеки CONNECTOR надо добавить в проект коннектор CON2 (рис. 6.17).

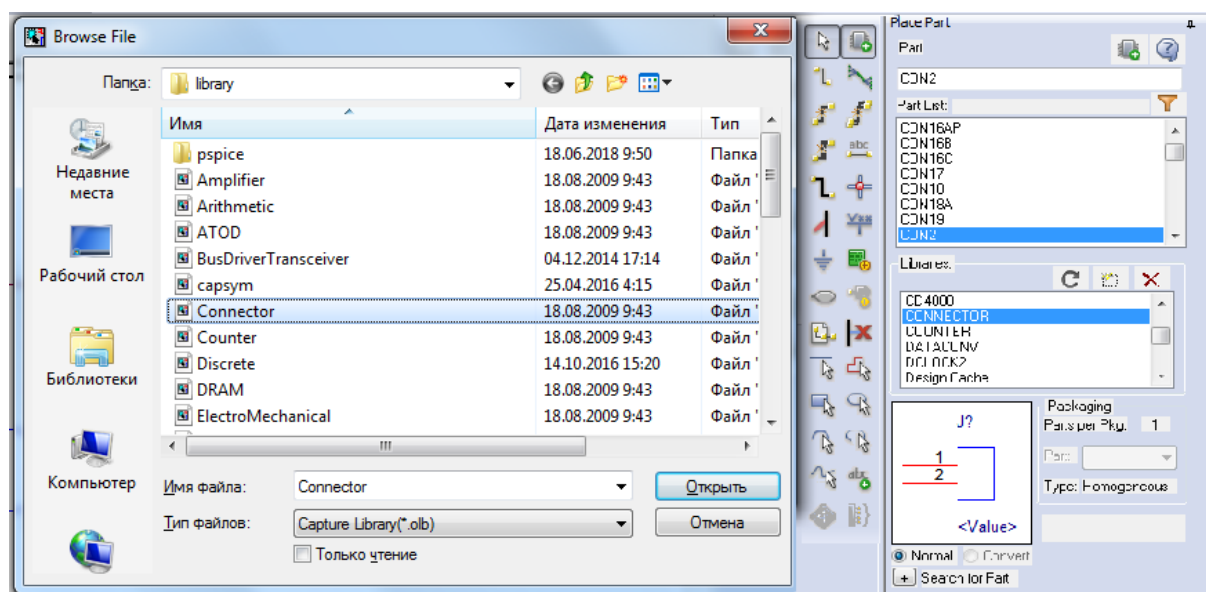


Рис. 6.17. Добавление коннектора CON2

Итак, вы успешно создали иерархический дизайн полного сумматора с использованием методологии снизу вверх. Все компоненты, используемые в этой конструкции, кроме коннектора были взяты из PSpice Library. Временно исключив коннектор, вы можете моделировать эту конструкцию с помощью PSpice.

6.1.7. Создание и сохранение компонентов для новых проектов

В предыдущем проекте FULLADD были созданы схемы полусумматоров, которые могут быть полезны в новых проектах.

Вместо того, чтобы создавая иерархический блок, каждый раз собирать схему полусумматора, вы можете, используя уже готовую схему, создать

компонент «Полусумматор», сохранить его в библиотеке, а затем повторно использовать компонент в любой конструкции, когда потребуется.

В этом разделе, мы создадим компонент для схемы простого полусумматора (рис. 6.7), который вы создали в разделе плоской схемы этой главы. Такие созданные компоненты называют ещё Hierarchical symbol (иерархический символ).

Чтобы сформировать компонент из схемы, выполните следующие шаги.

1. В окне менеджера проекта, выберите папку HALFADD.
2. В меню Tools выберите Generate Part.
3. В диалоговом окне Generate Part (рис. 6.18), укажите местоположение дизайн-файла, содержащего схему, для которой этот компонент должен быть сделан.

Для этого примера надо указать местоположение fulladd.dsn.

4. В раскрывающемся списке Netlist/source, укажите тип источника как Capture/Shematic/Design.

5. В текстовом поле Part name, укажите имя компонента, который должен быть создан как HALFADD.

6. Укажите имя и расположение библиотеки, которая будет содержать этот новый компонент. Для текущего примера дизайна, укажите имя библиотеки как fulladd.olb.

7. Если вы хотите, чтобы вместе с новым компонентом был создан схемный источник, установите флажок Copy Schematic в Library. Для этой конструкции выберите Check box.

8. Убедитесь, что выбрана опция Create new part.

9. Чтобы указать схемную папку, в которой содержится дизайн, для которого должен быть создан этот компонент, выберите HALFADD из имен раскрывающегося списка.

10. Нажмите кнопку ОК, чтобы сгенерировать компонент HalfAdd.

Откроется окно Split Part Section Input Spreadsheet с описанием выводов компонента (рис. 6.19).

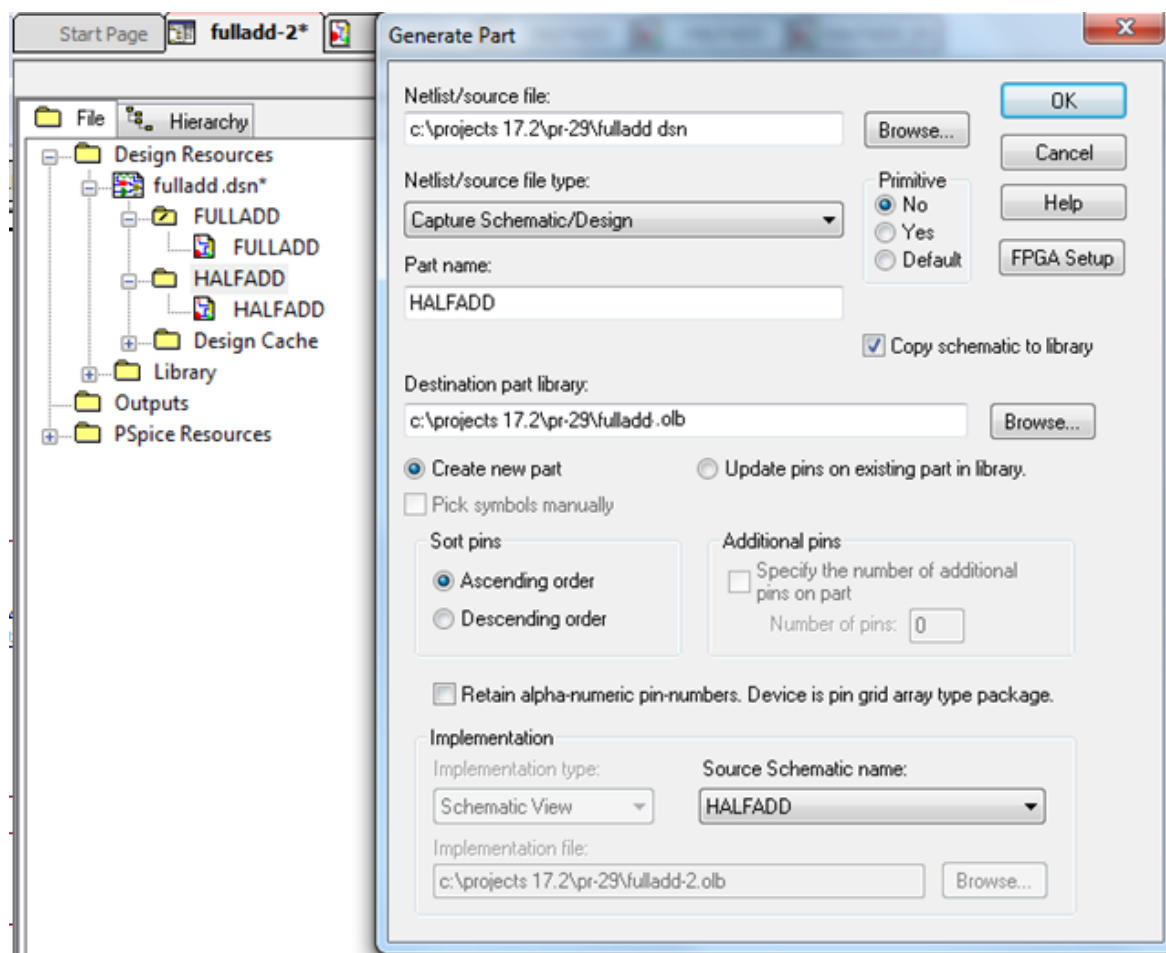


Рис. 6.18. Создание компонента HALFADD

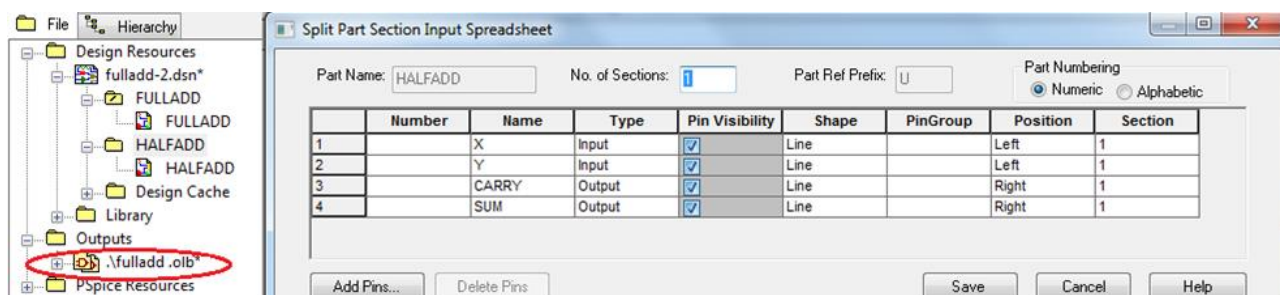


Рис. 6.19. Описание созданного компонента

Новая библиотека fulladd.olb будет сгенерирована и будет видна в папке Outputs в окне диспетчера проекта (рис. 6.19). Новая библиотека также добавляется в окно Place Part. Теперь вы можете использовать диалоговое окно Place Part, чтобы добавить компонент «Полусумматор» в любой новый дизайн.

Для этого в библиотеки нового проекта надо добавить библиотеку fulladd.olb из предыдущего проекта со схемой полусумматора из папки библиотеки FULLADD. После этого поместите на страницу компонент HALFADD (рис. 6.20). Посмотрите его схему.

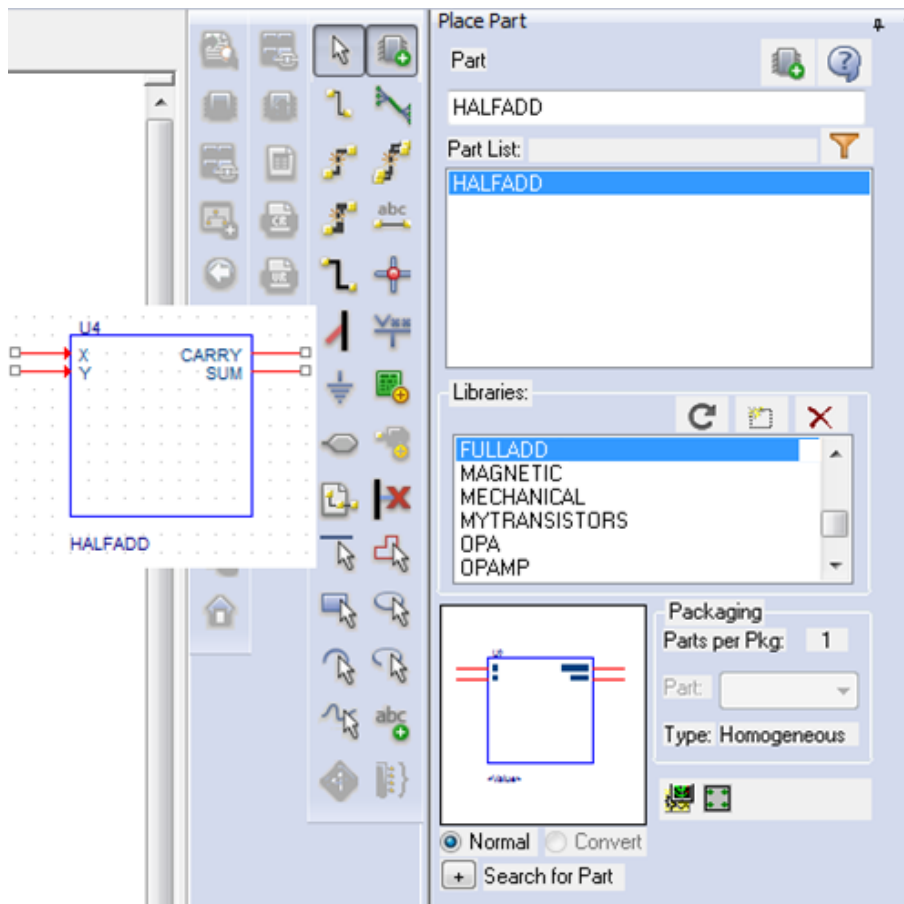


Рис. 6.20. Открытие сохраненного компонента полусумматора

6.1.8. Нисходящий метод

При создании иерархической конструкции, используя нисходящий метод, надо применять следующую последовательность шагов:

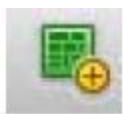
- Создать дизайн верхнего уровня с использованием функциональных блоков, входов и выходов, которые известны.
- Создать схематический дизайн для функционального блока, используемого в конструкции верхнего уровня.

В этом разделе представлен обзор шагов, которым необходимо следовать для создания полного сумматора, используя нисходящий метод.

1. Создайте проект FullAdd-TD.
2. Создайте дизайн верхнего уровня, используя следующие шаги:
 - 2.1. В меню Place выберите иерархический блок.

Примечание: В качестве альтернативы, вы можете выбрать кнопку на

панели инструментов



- 2.2. В диалоговом окне Place выберите Hierarchical Block, укажите ссылку как HALFADD_A1, Тип реализации, имя реализации, как HALFADD, и нажмите ОК (рис. 6.21).

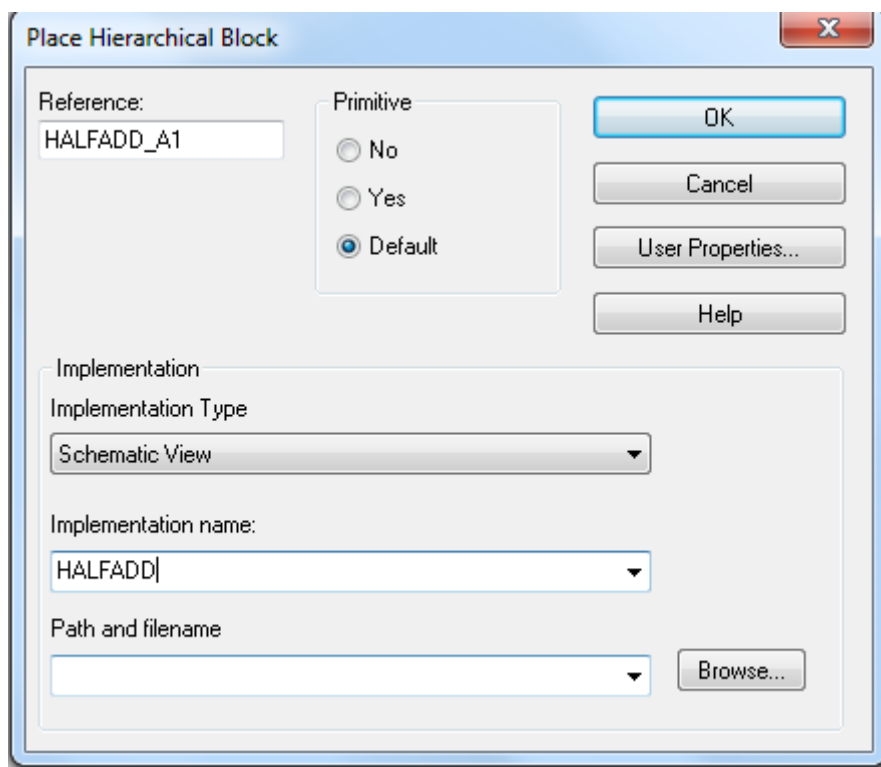


Рис. 6.21. Создание первого иерархического блока

2.3. Нарисуйте иерархический блок с нужными размерами (рис. 6.22).

Обратите внимание, что в отличие от иерархического блока, созданного в восходящем методе, иерархический блок в нисходящем методе не имеет приложенной информации о портах.

2.4. Выберите иерархический блок и затем из меню Place, выберите Hierarchical Pins (рис. 6.23).

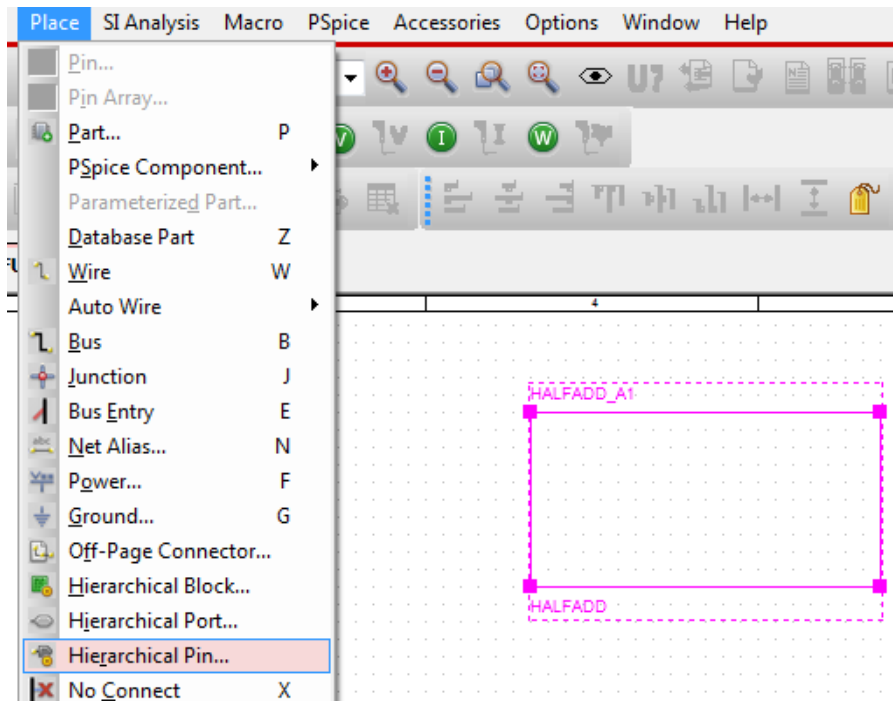


Рис. 6.22. Изображение иерархического блока

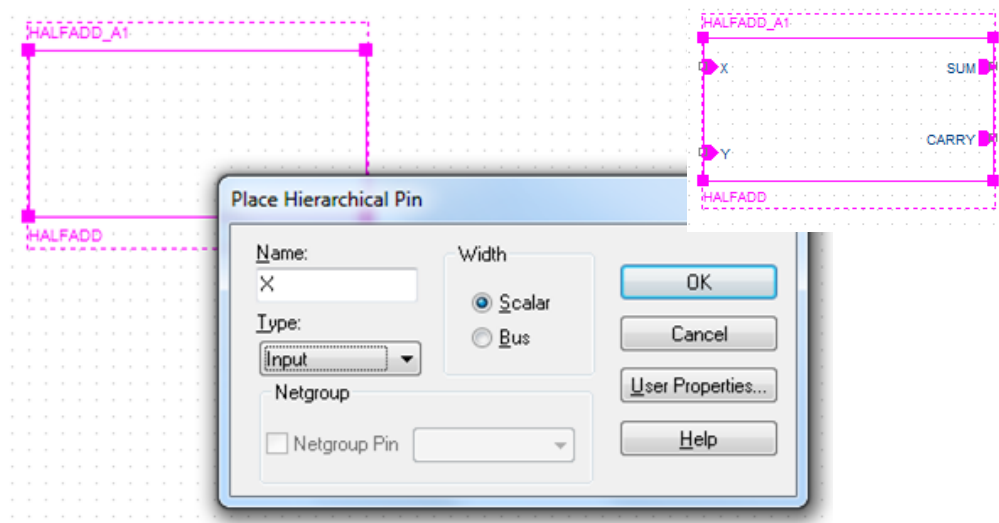


Рис. 6.23. Установка контактов блока

2.5. В диалоговом окне Place Hierarchical Pin укажите имя контакта как X, тип как вход, Width как Scalar и нажмите кнопку OK.

2.6. Поместите пин, как показано на рис. 6.23. Аналогично добавьте пин Y и два выходных пина SUM и CARRY.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

2.7. Поместите другой иерархический блок с Implementation Type как HALFADD. Простейший способ сделать это - скопировать существующий иерархический блок и вставить его на страницу схемы. По умолчанию опорное название второго иерархического блока HALFADD_A2. Измените этого название на HALFADD_B1 (рис. 6.24).

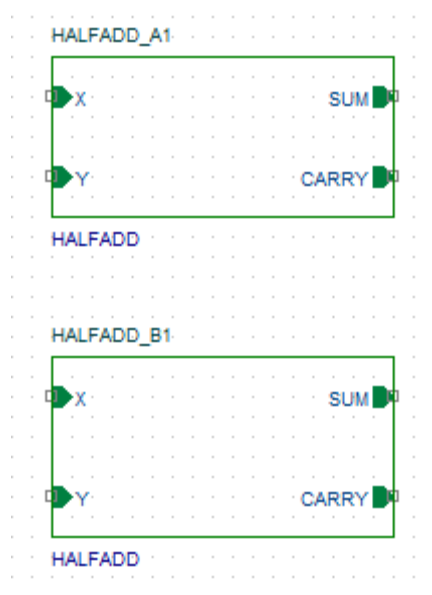


Рис. 6.24. Установка второго иерархического блока

2.8. Завершите создание схемы полного сумматора путём добавления портов, проводов и стимулов. Получится полный сумматор (рис. 6.25).

2.9. Сохраните проект.

3. Нарисуйте дизайн низшего уровня, используя шаги, перечисленные ниже.

Например, для полного сумматора, самым низким уровнем является полусумматор.

3.1. Чтобы нарисовать схему полусумматора, щёлкните правой кнопкой мыши на любом из иерархических блоков HALFADD.

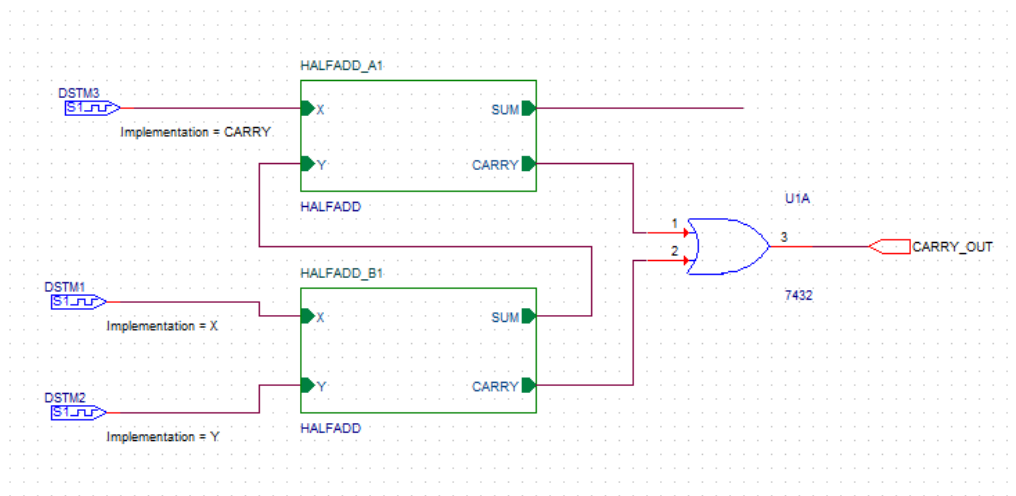


Рис. 6.25. Схема структуры полного сумматора без внутренних схем

3.2. Из всплывающего меню выберите Descend Hierarchy.

3.3. Появится новая страница в Schematic: 'HALFADD' . Укажите имя страницы как HALFADD и нажмите кнопку OK.

Новая страница схемы появится с двумя входными портами X и Y, и двумя выходными портами SUM и CARRY (рис. 6.26).

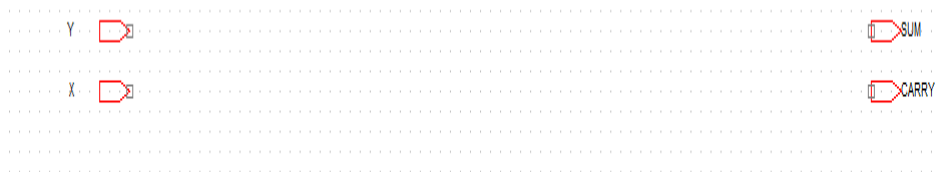


Рис. 6.26. Заготовка для схемы полусумматора

Теперь вы можете нарисовать схему полусумматора на этой странице схемы, используя шаги, описанные ранее в создании плоской конструкции (рис.18.27).

4. Вместо того, чтобы повторно создавать схему, загрузим сохраненный ранее компонент полусумматора.

Для этого в схеме (рис. 6.25) двойным щелчком откройте пустую страницу полусумматора HALFADD_A1.

Выберите Place Part и добавьте в список библиотек fulladd.olb из папки предыдущего проекта, в котором был создан и сохранен компонент «Полусумматор» (рис. 6.20).

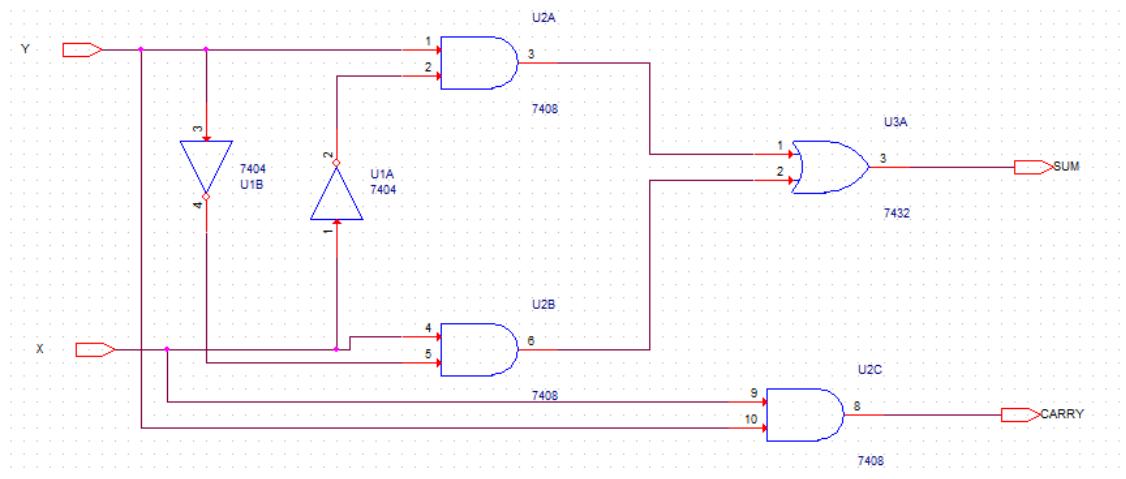


Рис. 6.27. Повторный набор схемы полусумматора

Откройте библиотеку fulladd.olb и поместите на страницу компонент HALFADD (рис. 6.28).

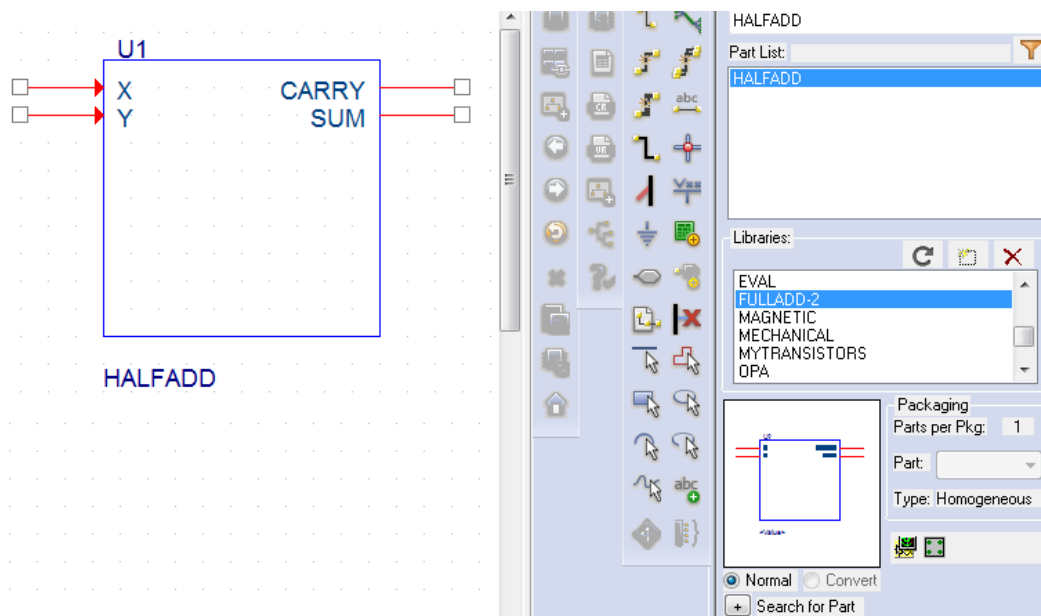


Рис. 6.28. Добавление в проект ранее созданного компонента

Сохраните проект. После этого иерархические блоки будут иметь внутри схемы полусумматоров.

В окне диспетчера проекта, добавляется новая схематичная папка HALFADD ниже полной fulladd-td.dsn (рис. 6.29).

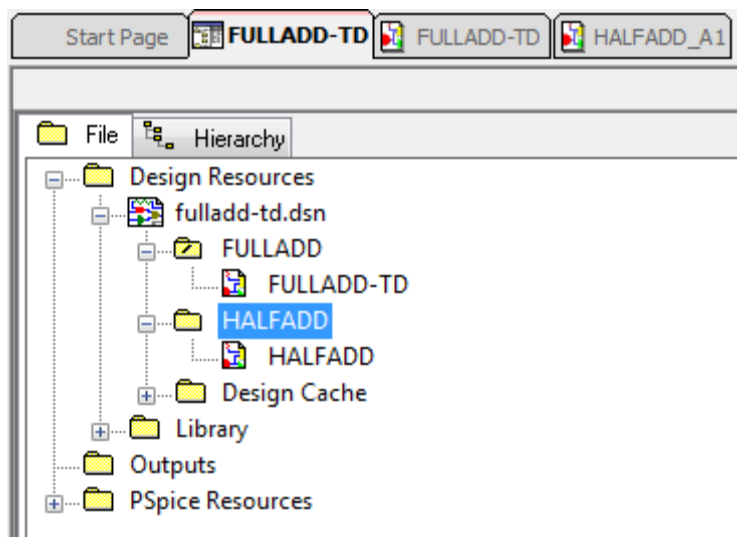


Рис. 6.29. Структура диспетчера иерархического проекта

Двойной щелчок на иерархическом блоке открывает его внутреннюю схему. Это же можно сделать так: выделите блок, щёлкните правой кнопкой и в меню выберите Descend Hierarchy.

6.2. Перемещение по иерархической конструкции

Для перехода на более низкие уровни иерархии, щёлкните правой кнопкой мыши иерархический блок и выберите Descend Hierarchy.

Точно так же, чтобы двигаться вверх по иерархии, щелкните правой кнопкой мыши на схеме полусумматора и выберите Ascend Hierarchy (рис. 6.30).

Опции меню Ascend Hierarchy и Descend Hierarchy также доступны в окне в раскрывающемся меню окна View.

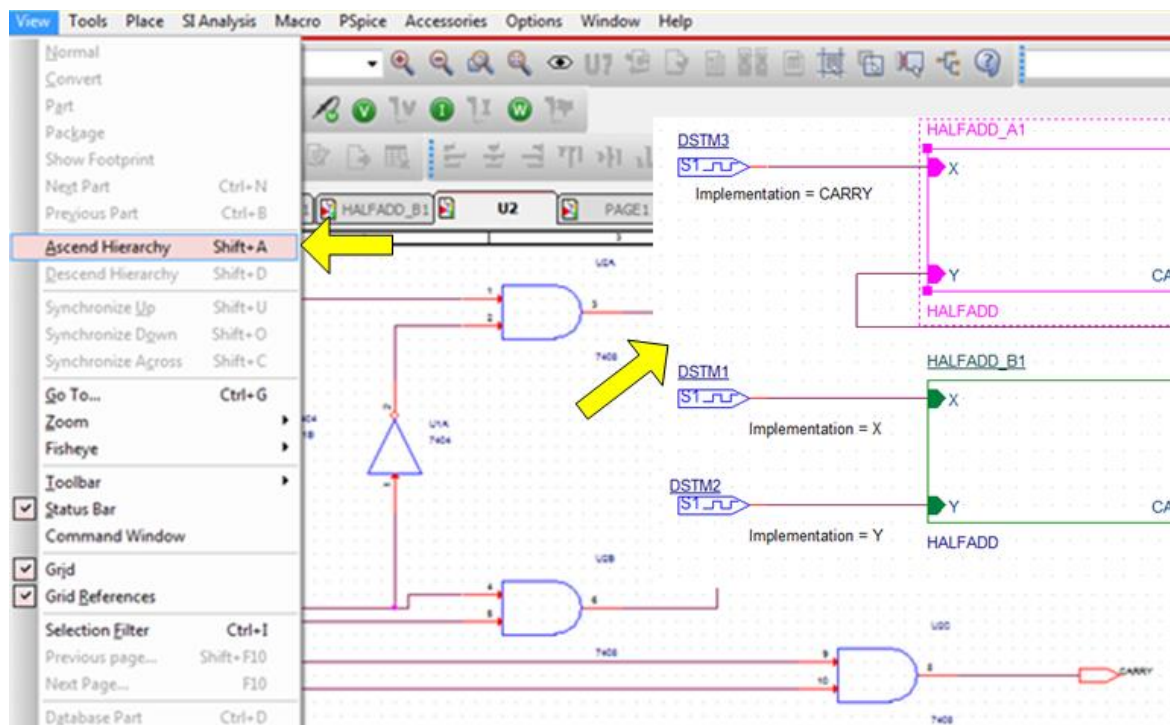


Рис. 6.30. Перемещение вверх по иерархии

Во время работы с иерархическими конструкциями, вы можете сделать изменения в иерархических блоках, а также в конструкции на самом низком уровне. Для того, чтобы сохранить обновления с изменениями различных уровней иерархии, вы можете использовать Synchronize options, которые доступны в меню View.

Выберите Synchronize Up, когда вы внесли изменения в конструкции самого нижнего уровня и хотите, чтобы эти изменения были отражены выше в иерархии.

Выберите Synchronize Across, когда после внесения изменений в иерархическом блоке необходимо, чтобы изменения были отражены во всех экземплярах блока.

Выберите Synchronize Down, когда вы внесли изменения в иерархическом блоке и хотите, чтобы эти изменения были отражены в конструкции самого нижнего уровня.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

6.3. Моделирование полного сумматора

Выполним моделирование полного сумматора, чтобы получить временные диаграммы его функционирования.

Вернемся к схеме полного сумматора с цифровыми сигналами (рис. 6.31). В этой схеме временно исключен коннектор CON2, так как для него нет PSpice модели.

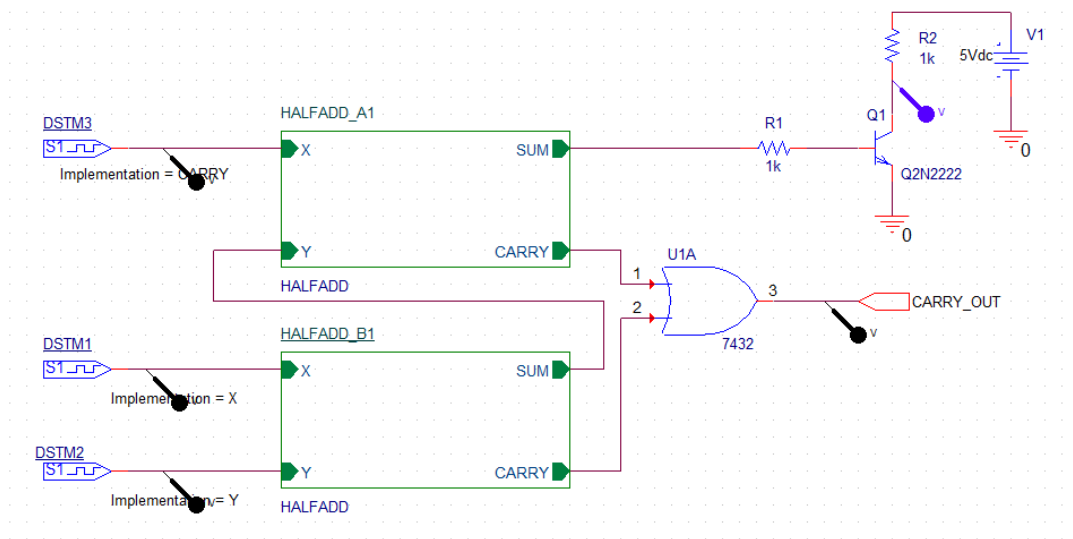


Рис. 6.31. Схема полного сумматора с цифровыми сигналами

Выполним установку цифровых стимулов (рис. 6.32).

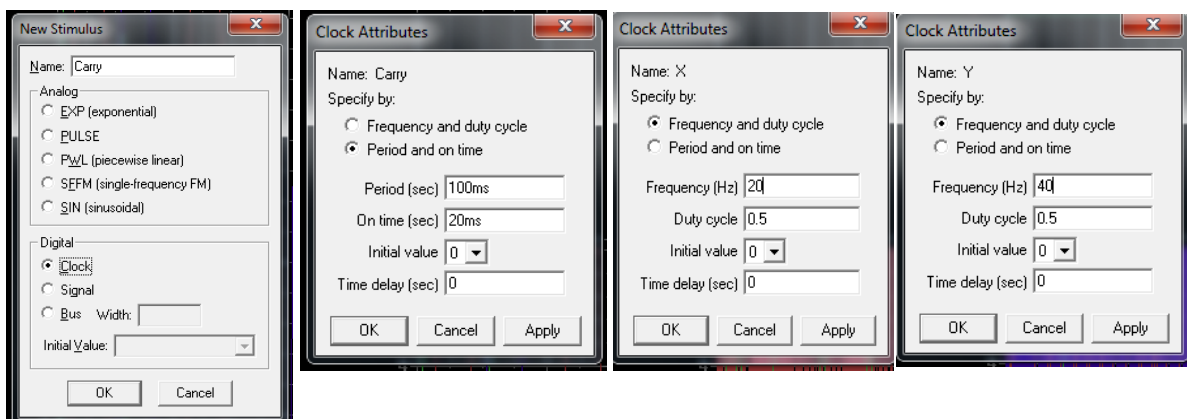


Рис.

6.32. Установка цифровых стимулов

В профиле моделирования Transient установим время 400 мс, шаг 100 мкс (рис. 6.33), а в опциях установим Gate Level Simulation.

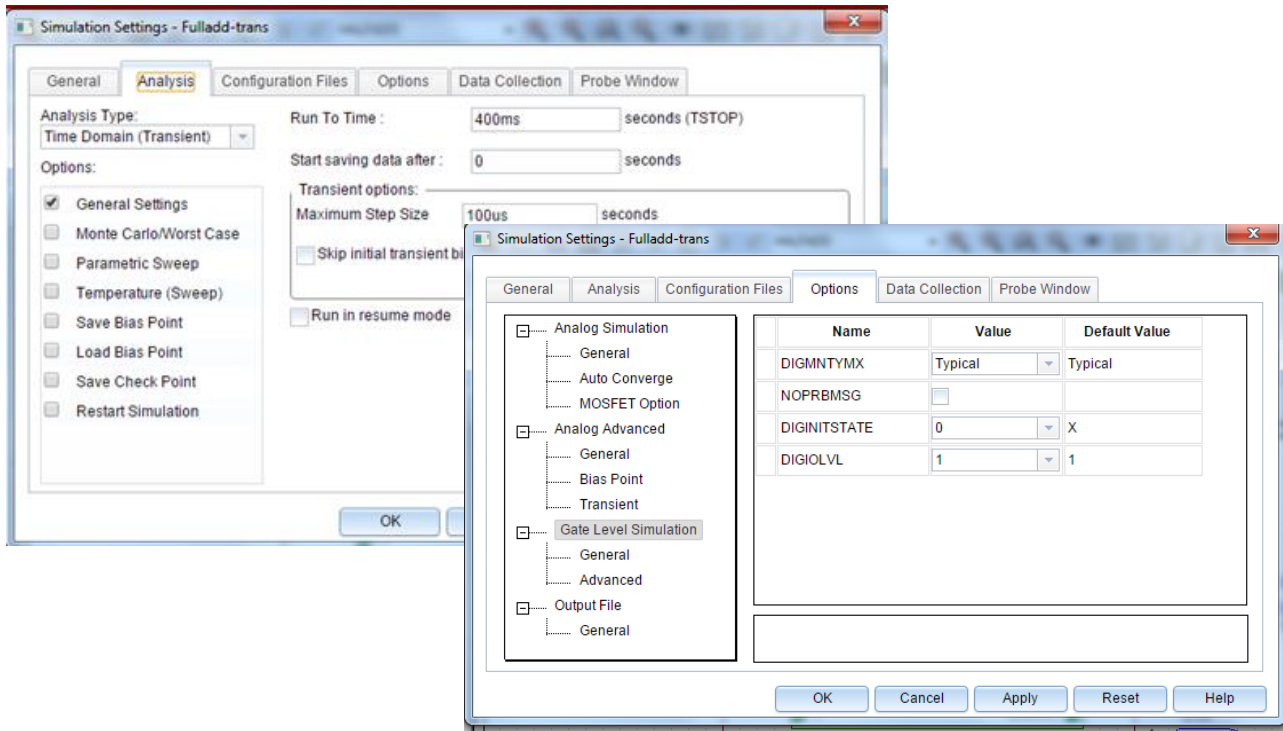


Рис. 6.33. Установка профиля моделирования

Результаты моделирования показаны на рис. 6.34.

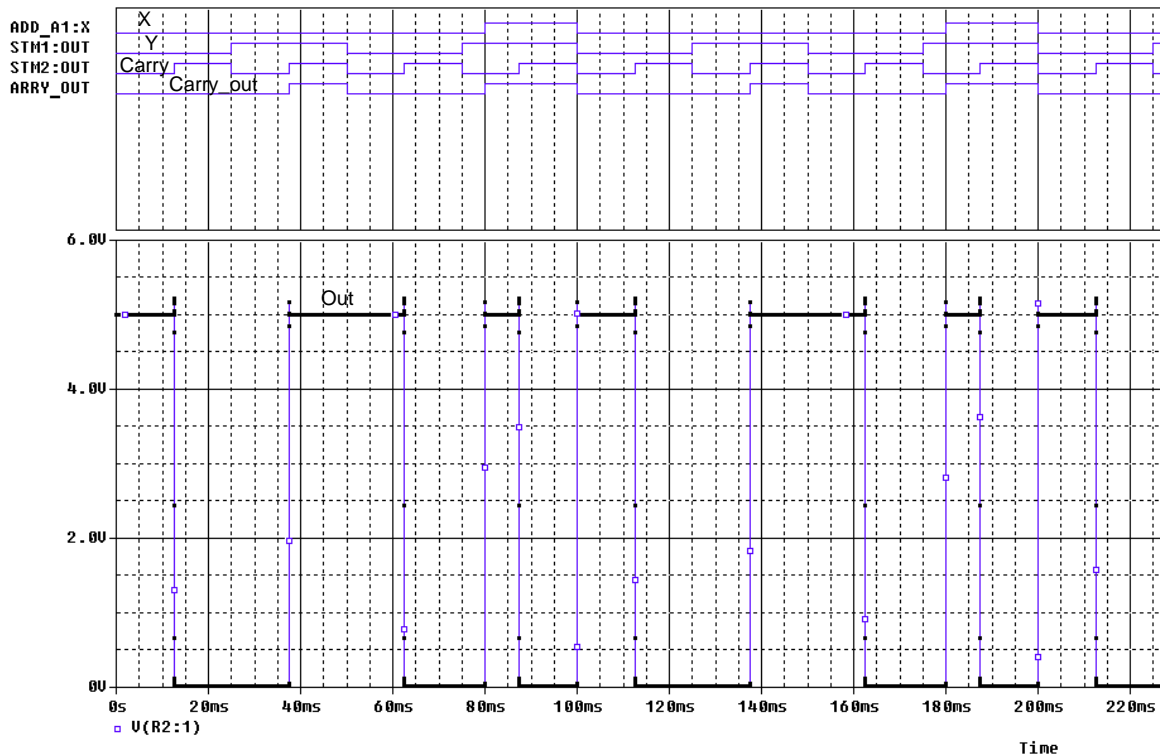


Рис.

6.34. Результаты моделирования полного сумматора

6.4. Контрольные вопросы

1. В чем разница между плоской и иерархической схемой ?
2. Какую структуру имеют иерархические проекты ?
3. Какие внестраничные и иерархические порты используют в PSpice ?
4. Расскажите о методологии создания иерархического проекта восходящим методом.
5. Как проверить правильность схемы иерархического блока ?
6. Как создать несколько аналогичных иерархических блоков ?
7. Как создать и сохранить новый компонент, например, полусумматор ?
8. Где будет сохранена новая библиотека для созданного компонента ?
9. Расскажите о методологии создания иерархического проекта нисходящим методом.
10. Как создать дизайн низшего уровня в нисходящем методе ?

11. Как использовать сохраненный ранее компонент для создания дизайна низшего уровня ?
12. Какие опции используют для перемещения по иерархической конструкции и контроля схем ?
13. Как можно вносить изменения в иерархические проекты и сохранять их ?
14. Почему при моделировании полного сумматора приходится временно удалять коннектор ?

Глава 7. Испытательные стенды

Как правило, при запуске пробной симуляции, чтобы проверить цепь на схеме можно добавить, например, источники напряжения и резисторы нагрузки. Вы можете даже удалить некоторые компоненты из схемы. Однако, как только пробное моделирование выполнено, все добавленные элементы должны быть удалены, а любой удаленный компонент надо восстановить.

До версии 16.5 можно было добавить свойства PSpiceOnly для компонентов, которые используются только для моделирования, и поэтому эти свойства не будут включены, например, в списке соединений для печатной платы (PCB). Начиная с версии 16.5, вы можете использовать опцию Partial Design Feature, которая использует испытательные стенды, позволяющие Вам определить те компоненты, которые используются только для моделирования. Вы можете также выборочно разделить схемы для различных профилей моделирования и создавать проекты с использованием схемы из других проектов. Использовать испытательные стенды очень полезно, когда у вас есть схема, которая была составлена из набора схем из других проектов. Это позволит вам проверить работоспособность каждой отдельной цепи, которую вы встроите позже в полную схему.

При создании тестового стенда, папка Test Bench, которая содержит всю схемную документацию, добавляется в нижней части Менеджера проекта. Все компоненты во всех схемах в папке Test Bench будут выделены серым цветом. Тогда выборочно можно «активировать» те части, которые необходимы для моделирования и добавить детали, такие как источники напряжения и нагрузочные сопротивления. Компоненты могут быть выбраны и исключены из основной схемы или из созданных испытательных стендов.

При создании тестового стенда в проекте создается другая схемная папка. Папка проекта будет содержать две папки:

<project name>- PSpiceFiles

<project name>- TBFiles

Схема со схемной утилитой (SVS) будет сравнивать схемы испытательного стенда с основным проектом таким образом, что основной проект может быть обновлён с модифицированными значениями компонентов.

7.1. Использование частичного моделирования проекта

Используя функцию частичного моделирования, вы можете:

Определить отдельные компоненты любого проекта и моделировать только выбранные части;

Моделировать различные схемы в проекте с различными профилями моделирования;

Создать список соединений только для определенной части проекта;

Сравнивать и быстро объединять части дизайна. Чтобы использовать эту функцию, вы выбираете часть, называемую тестовым стендом основного проекта. Вы создаете один или более тестовых стендов с помощью меню OrCAD Capture Tools> Test Bench>Create Test Bench. Испытательные стенды перечислены в окне диспетчера проектов главного проекта.

Вы можете добавить компоненты из проекта в тестовый стенд, выбрав их из основного, а затем можно добавить профили и смоделировать тестовый стенд. Вы также можете синхронизировать основной проект с тестовым стендом, чтобы распространить любые изменения, внесенные в дизайн стенда.

Вы можете использовать частичное моделирование проекта в потоке, показанном на рис. 7.1.

Для этого:

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

1. Создайте тестовый стенд.
2. Выберите детали в главной схеме.
3. Завершите соединение плавающих цепей в тестовом стенде.
4. Моделируйте схему тестирования.
5. Просмотрите различия свойств между стендом и основным проектом.
6. Обновите основной проект с измененными значениями.

Остальные разделы данной главы подробно объясняют эти шаги.

Для использования этой функции требуется лицензия OrCAD Capture CIS.

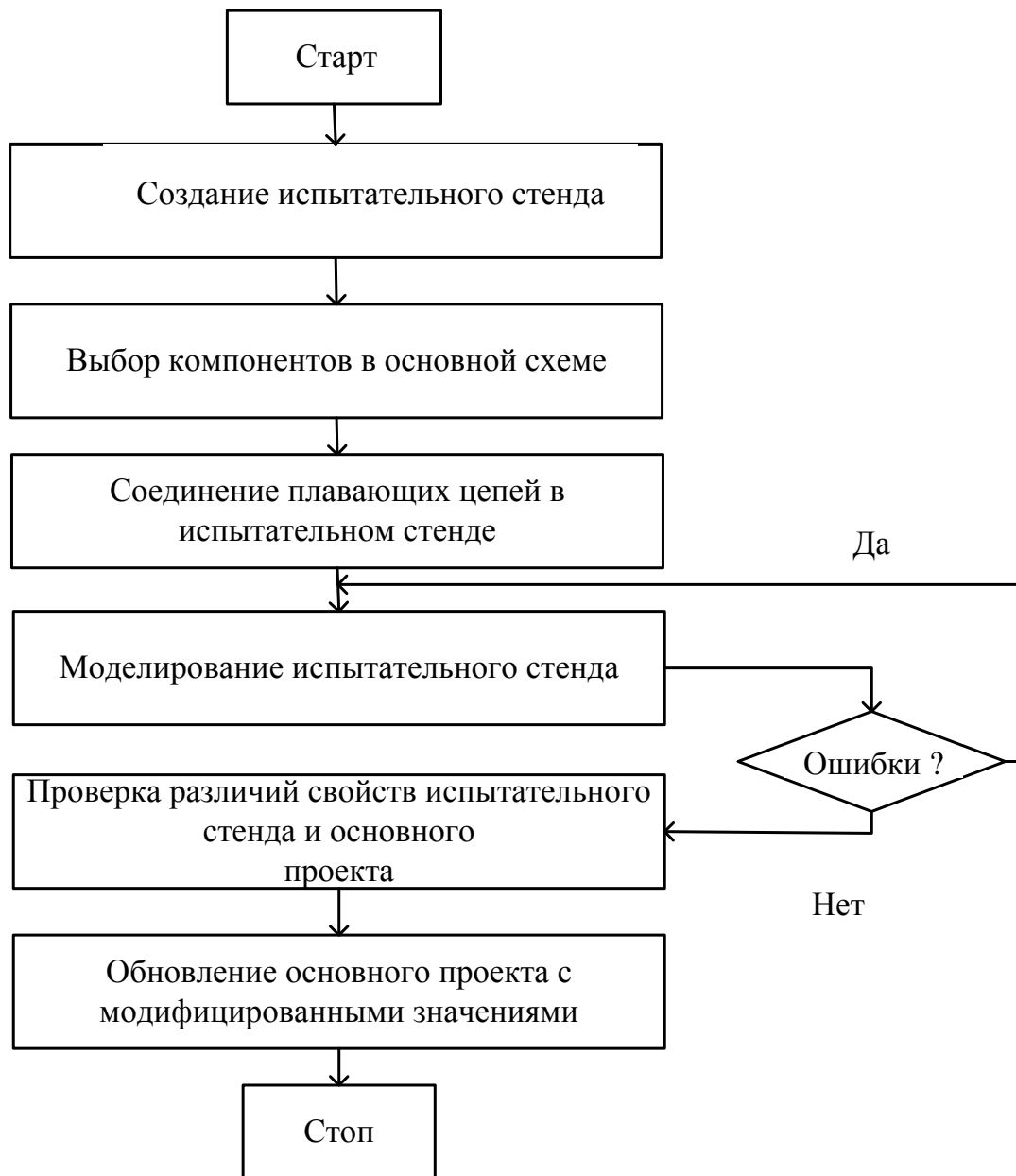


Рис. 7.1. Поток частичного моделирования тестового стенда

7.2. Работа с тестовым стендом

Тестовый стенд подобен любому другому новому проекту, созданному в Capture. Когда вы создаете тестовый стенд, он перечисляется под узлом TestBenches в Менеджере главного проекта. Все профили моделирования, па-

параметры или переменные в основном проекте копируются по умолчанию в тестовый стенд.

Компоненты в разных схемах выделены серым цветом. Вы можете активировать компоненты для создания частичного проекта.

Возможно, вам придется добавить окончания соединений и другие компоненты в частичный проект стенда, потому что проект стенда должен быть полным сам по себе.

Вы также можете внести изменения на свой стенд, чтобы подготовить его для моделирования, добавив профили стимулов или моделирования. Вы можете моделировать тестовый стенд даже, если основной проект не является проектом PSpice. Если главный проект является проектом PSpice, Test Bench может наследовать профили моделирования из основного проекта.

7.2.1. Создание тестового стенда

Создайте новый проект TestBench на основе ранее исследованного проекта аналогового компаратора с цифровым выходом (глава 5).

1. Выберите файл DSN в диспетчере проектов
 2. Выберите Tools>Test Bench>Create Test Bench.
- Появится поле Test Bench.
3. Введите имя в поле Enter Test Bench Name.

Вы можете установить имя тестового стенда по умолчанию, добавив свойство Default Test Bench Name в [TEST BENCH] раздела capture.ini. Например, чтобы установить имя тестового стенда по умолчанию для MyTestBench, добавьте следующий раздел в capture.ini:

```
[TEST BENCH]
Default Test Bench Name=MyTestBench
```

Нажмите «ОК».

Тест-стенд добавляется в TestBench в диспетчере проектов. Созданный стенд содержит все конструкции из основного проекта (рис. 7.3).

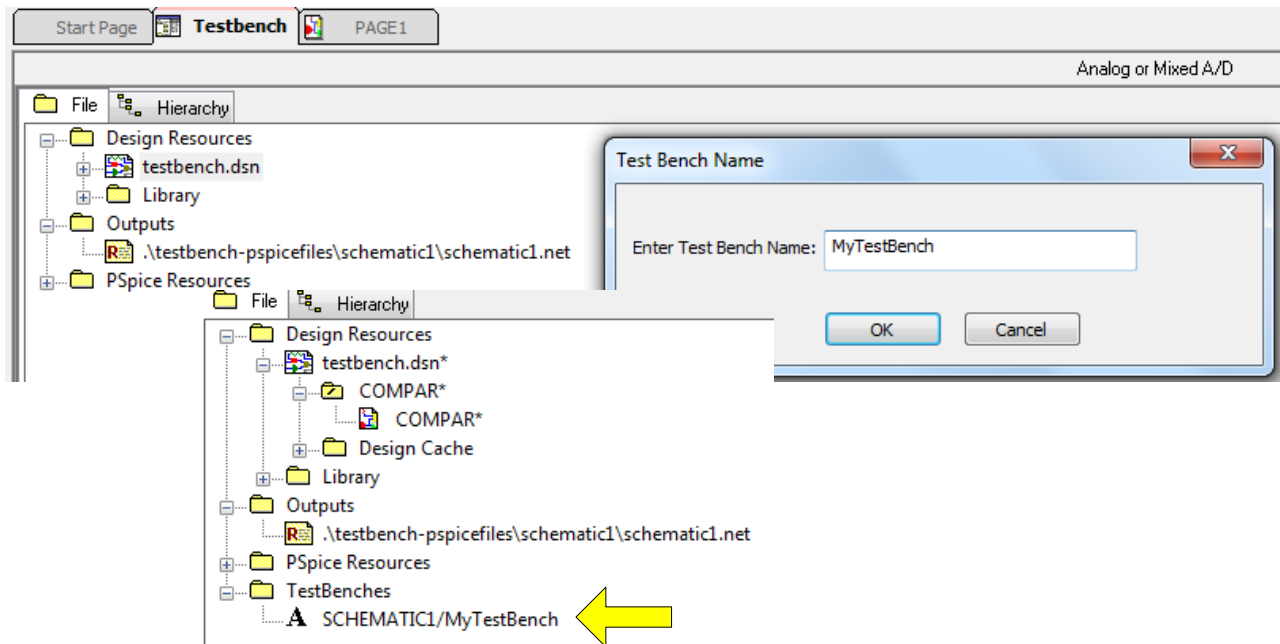


Рис. 7.3. Добавление в проект тестового стенда

В главном проекте переименуем папку SCHEMATIC и назовем ее COMPAR.

Теперь в главном меню теперь имеется две закладки (рис. 7.4):

COMPAR – схема основного проекта;

[TB]PAGE1 – схема тестового стенда.

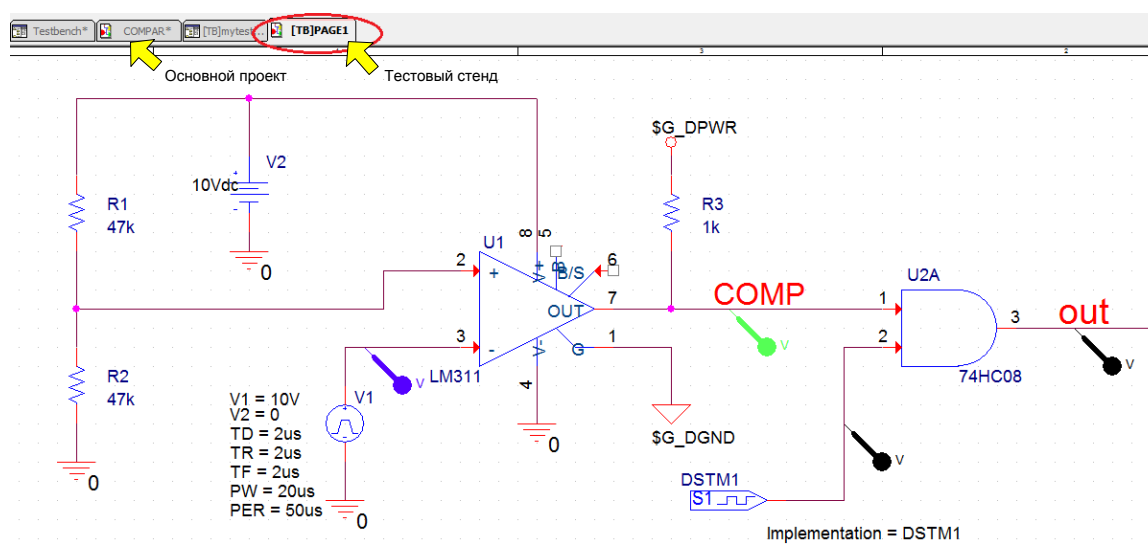


Рис. 7.4. Схема испытательного стенда

Компоненты на страницах схемы тестового стенда неактивны. Вы должны добавить компоненты для стенда, чтобы иметь возможность работать над частичным проектом.

Примечание. Вы можете активировать тестовый стенд, щелкнув правой кнопкой мыши на стенде в диспетчере проектов в разделе TestBenches и выбрав MakeActive.

7.2.2. Активация компонентов

Вы можете активировать компоненты на стенде, используя любой из параметров: контекстное меню для выбранной части в главном проекте, контекстное меню для выбранных частей в дизайне стенда или из редактора иерархических блоков.

Чтобы активировать компоненты из главного проекта:

1. Выберите компоненты в главном проекте.
2. Щелкните правой кнопкой мыши и выберите TestBench>Add Part(s) To Active TestBench (рис. 7.5).

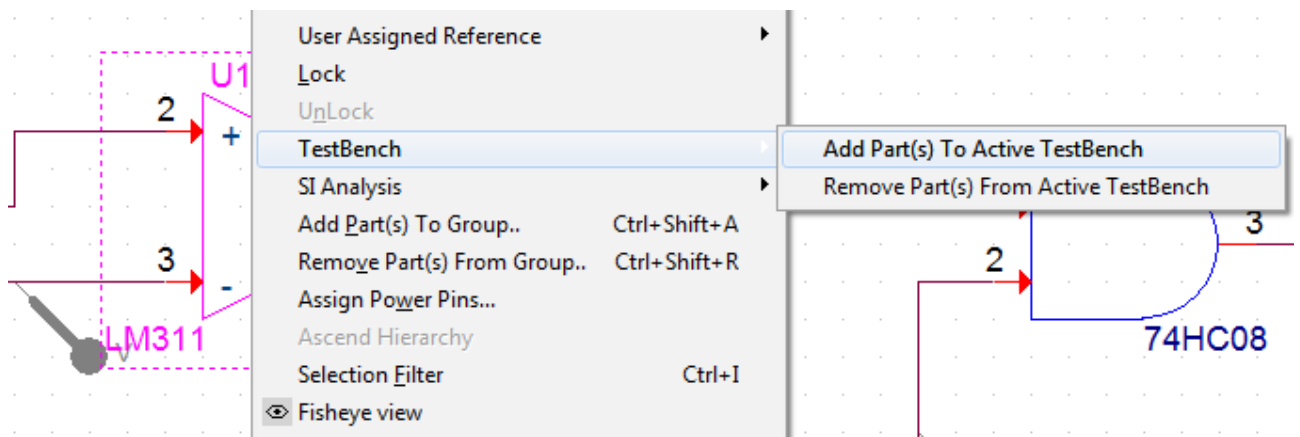


Рис. 7.5. Активизация компонентов Test Bench из основного проекта

Для активации компонентов из стенда:

Выберите компоненты в дизайне стенда.

Щелкните правой кнопкой мыши и выберите TestBench>Add Part(s) To Self (рис. 7.6).

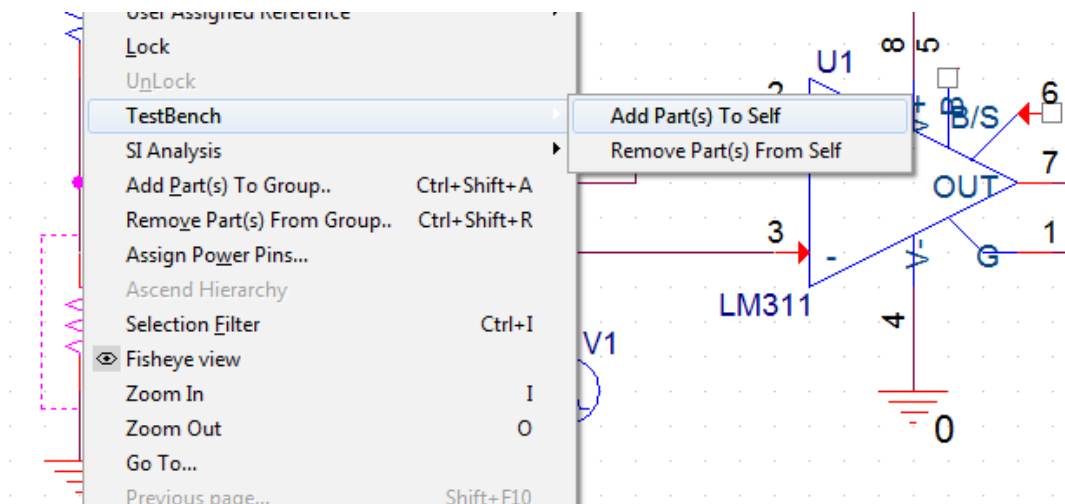


Рис. 7.6. Активизация компонентов из испытательного стенда

Чтобы активировать компоненты с помощью редактора иерархии, проверьте компоненты, которые необходимо добавить в редактор иерархии основного проекта, как показано на рис.7.7.

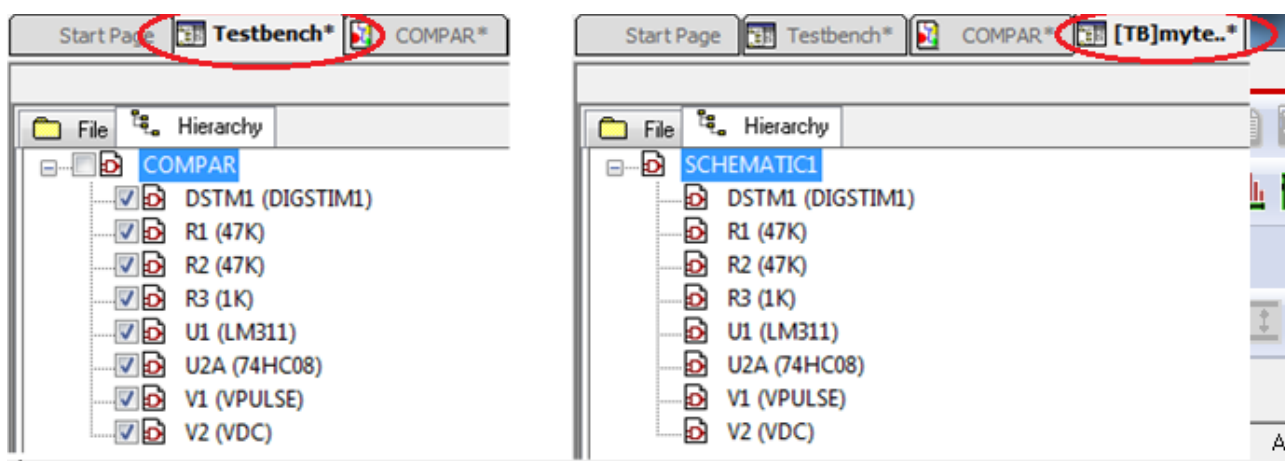


Рис. 7.7. Проверка активации компонентов в иерархии

Точно так же вы можете инактивировать компонент из тестового стенда из контекстного меню или редактора иерархии. Capture игнорирует все неактивные компоненты. В результате эти неактивные компоненты не будут обрабатываться, например, для моделирования.

Когда вы активируете только часть дизайна, многие сети могут стать плавающими, потому что они не соединены. Вы можете легко решить эту проблему, выполнив плавающий сетевой поиск.

Для этого:

- Выберите дизайн тестового стенда в Capture;
- В меню поиска выберите Floating Nets (плавающие цепи), как показано на рис. 7.8.
- Нажмите кнопку «Find».

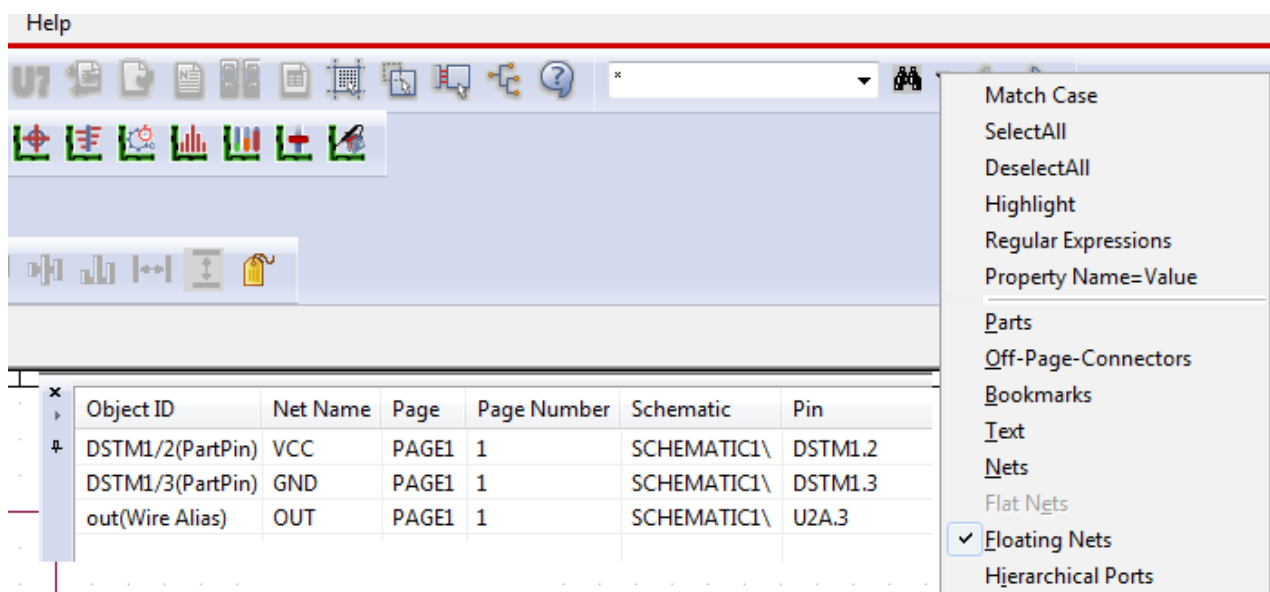


Рис. 7.8. Поиск «плавающих цепей»

Все плавающие сети, требующие терминаторов, перечислены на вкладке «Плавающие цепи» в Find.

Дважды щелкните строку в окне поиска, чтобы выбрать ее в проекте.

7.3. Сравнение и обновление основного проекта

После того, как выполнена активация компонентов и правильное соединение всех компонентов, можно проверить функционирование испытательного стенда и сравнить результаты с основным проектом. Для этого генераторе стимула DSTM1 установим период 2 мкс и время включения 1 мкс. В профиле моделирования установим режим Transient на время 100 мкс с минимальным шагом 10 нс. Выполним моделирование. Результаты (рис. 7.9) совпадают с полученными ранее для основного проекта (рис. 5.3).

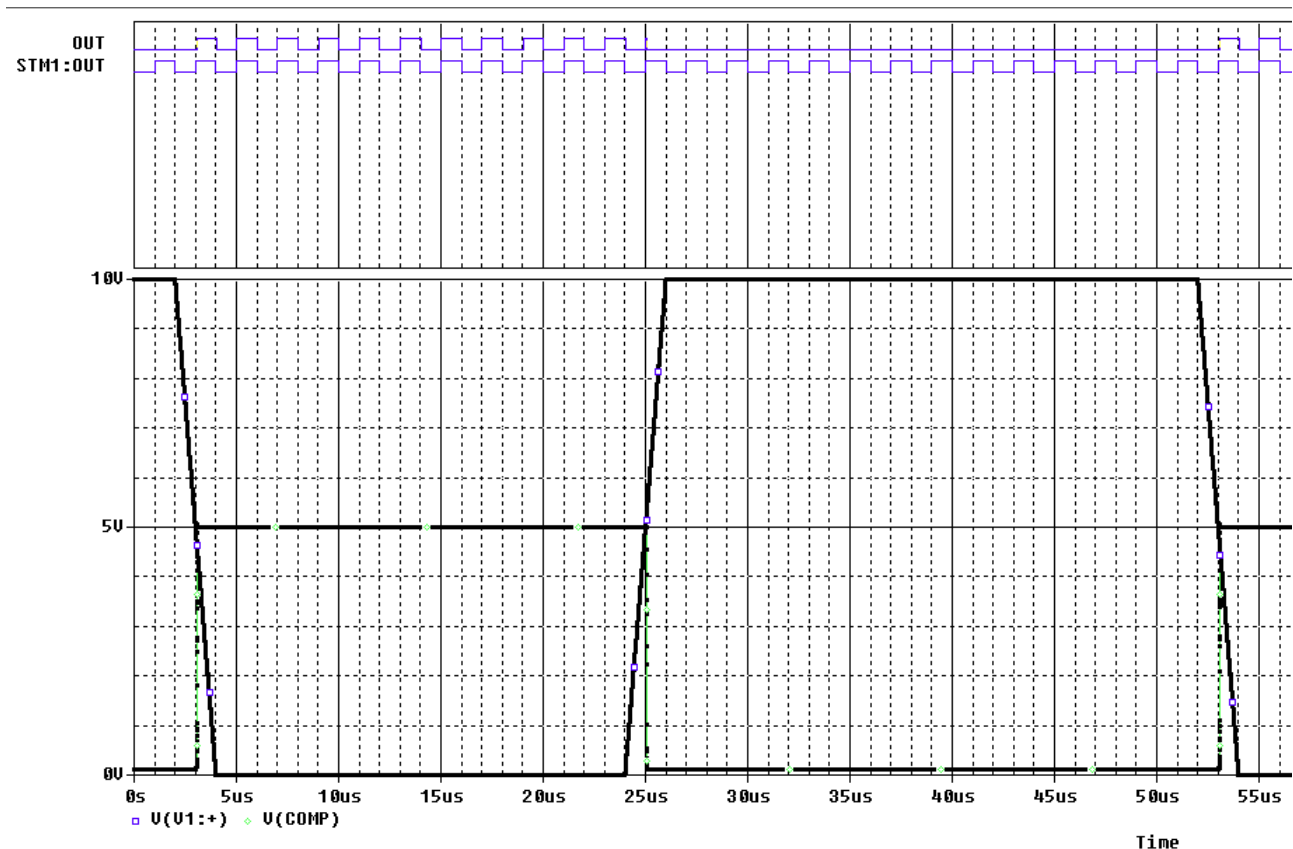


Рис.7.9. Результаты моделирования тестового стенда

Вы можете сравнить схемы в главном проекте и стенде, чтобы выделить расхождения с использованием утилиты SVS. Эта утилита отображает различия и использует цветовой код для выделения различных типов изменений. Окно результата имеет две панели, левая панель представляет испытательный стенд. Различия перечислены для категорий:

- несовпадающие объекты (unmatched object) (желтый по умолчанию);
- отсутствующие объекты (missing objects) (по умолчанию красные);
- совпадающие объекты (matching objects) (белый цвет).

Вы можете проверить любое из перечисленных различий на панели тестового стенда и распространять изменения в основной проект.

Однако вы не можете обновить основной проект для отсутствующих объектов.

Вы можете нажать «Настройки» (), чтобы открыть диалоговое окно «Параметры» и изменить на вкладке цвета по умолчанию. Вы также можете фильтровать разные объекты, если вы не хотите, чтобы они были перечислены.

Для сравнения и распространения изменений:

1. Выберите основной DSN файл в диспетчере проектов.
2. Выберите Tools>Test Bench>Compare Test Bench.

В окне SVS отображаются различия между основным проектом и проектом тестового стенда. В тестовом стенде мы добавили резисторы R4 и R5. Их отсутствие в основном проекте отмечено в окне SVS (рис. 7.10).

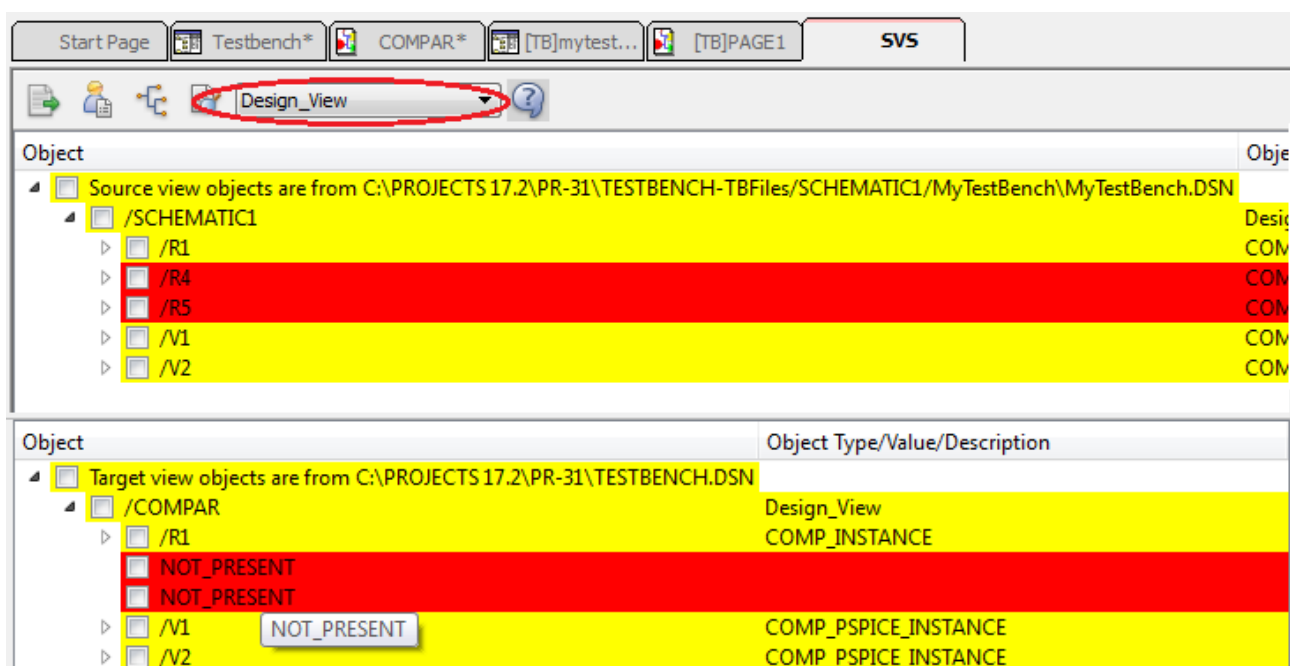


Рис.7.10. Окно сравнения проектов

Чтобы обновить основной проект с помощью различий в тестовом стенде, проверьте различия, которые вы хотите использовать для обновления на вкладке SVS и нажмите Accept Left ().

7.4. Контрольные вопросы

1. Для чего применяют и как используют испытательные стенды ?

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

2. Как отображается тестовый стенд в Менеджере проекта ?
3. Как используют функцию частичного моделирования проекта ?
4. Как организован поток частичного моделирования ?
5. Расскажите о порядке создания тестового стенда.
6. Как можно активировать компоненты на тестовом стенде из основного проекта ?
7. Как выполнить активацию компонентов из испытательного стенда ?
8. Как проверить наличие плавающих цепей ?
9. Как провести сравнение основного проекта и тестового стенда в окне SVS ?

Глава 8. Обработка схемы

После того как вы создали свой эскизный проект, вам может потребоваться обработка вашей конструкции путём добавления дополнительной информации для таких задач, как моделирование, синтез и разработка топологии печатной платы.

В этом разделе описываются некоторые из задач, которые можно выполнять в OrCAD Capture для обработки вашего проекта.

Мы будем изучать это на примере проекта полного сумматора с выходным транзистором и коннектором из главы 6 (рис. 6.16 и 8.1). Для этого создадим новый проект в папке PR-33-Refer на основе проекта PR-29-FullAdd-2.

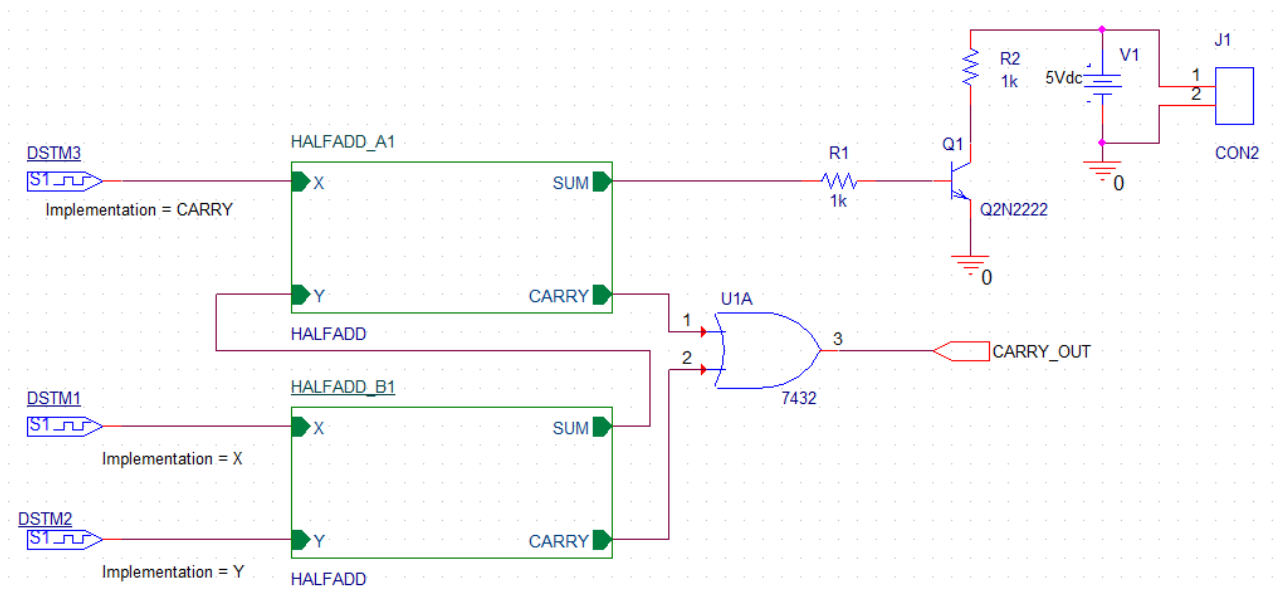


Рис. 8.1. Схема полного сумматора с выходным транзистором

8.1. Добавление ссылок для компонентов

Для того, чтобы быть в состоянии передать ваш эскизный проект в Редактор печатных плат для компоновки и трассировки, вам необходимо убедиться, что все компоненты в конструкции однозначно идентифицируются со ссылками компонентов.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

В OrCAD Capture можно назначить ссылки либо вручную или с помощью команды Annotate.

По умолчанию Capture добавляет ссылки на все компоненты, размещённые на странице схемы. При необходимости эту функцию можно отключить, выполнив действия, перечисленные ниже.

1. В меню Options выберите Preferences.
2. В диалоговом окне Preferences выберите Miscellaneous tab.
3. В разделе Auto Reference снимите флажок Automatically reference.
4. Нажмите кнопку ОК, чтобы сохранить эти настройки.

Последовательность действий показана на рис. 8.2.

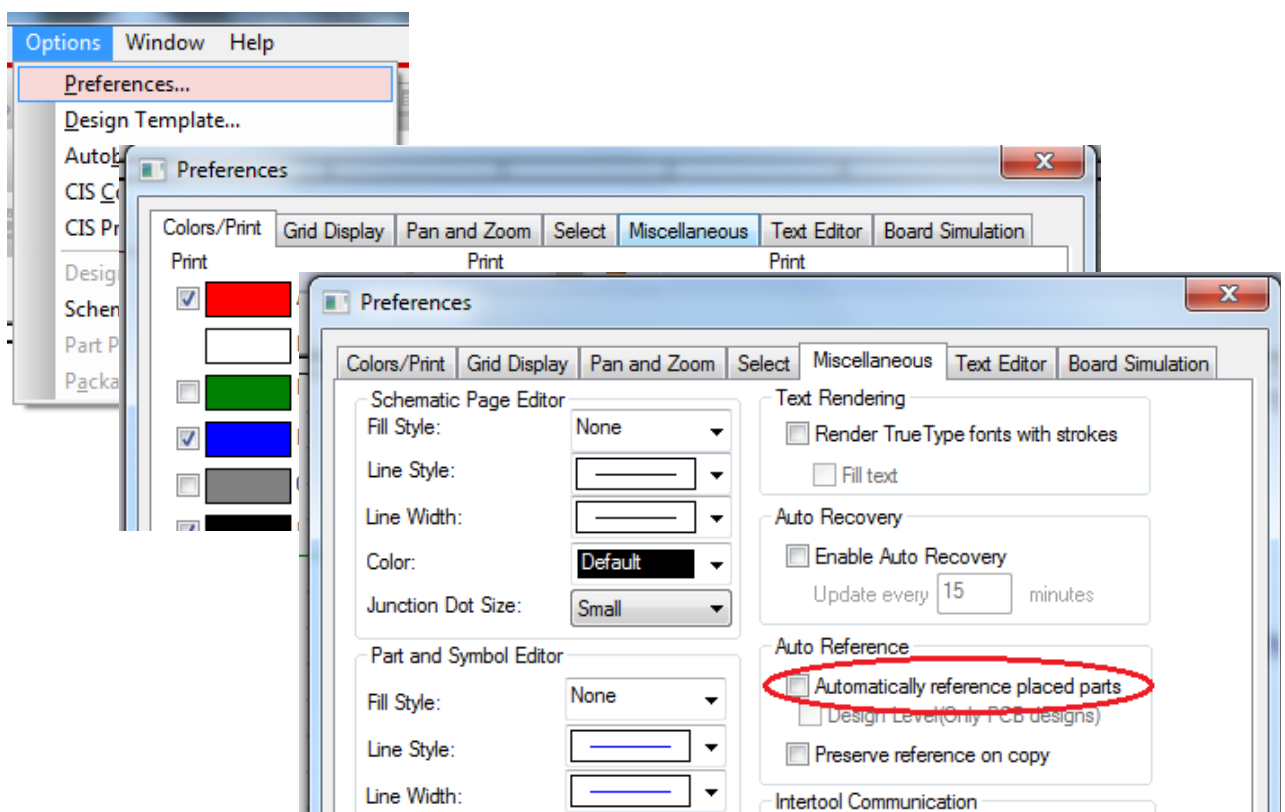


Рис. 8.2. Отключение автоматических ссылок


Поскольку существуют две копии иерархического HALFADD блока в дизайне FULLADD, оба дизайна должны быть аннотированы. Для этого

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

выполним команду Annotate. (Хотя уникальные ссылки были назначены по умолчанию, когда компоненты были размещены, копирование иерархических блоков должно скопировать содержимое иерархического блока буквально и повторное аннотирование ссылок компонентов не требуется).

Чтобы присвоить уникальные ссылки на компоненты в проекте FULLADD с помощью команды Annotate, выполните следующие шаги:

1. В окне менеджера проекта, выберите файл pr-33-refer.dsn.
2. Из меню Tools выберите Annotate.

Примечание: В качестве альтернативы, вы можете нажать кнопку Annotate на панели инструментов .

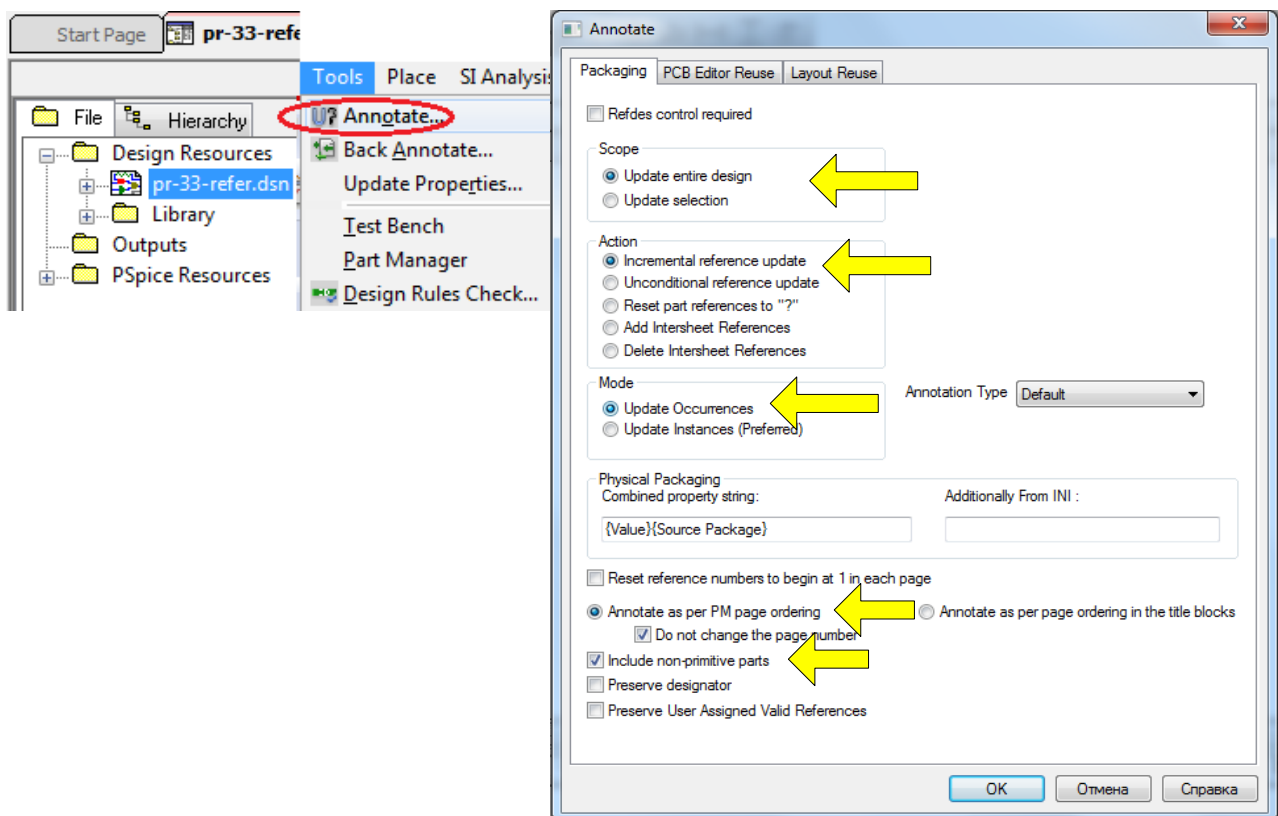


Рис. 8.3. Аннотирование проекта

3. На вкладке Packaging диалогового окна Annotate укажите, хотите ли вы обновить полный дизайн или только часть конструкции. Выберите кнопку Update entire design.

4. В разделе Action выберите кнопку опции обновления Incremental reference.

Примечание: Чтобы узнать о других доступных опций, см. диалоговое окно справки.

5. Схема полного сумматора представляет собой сложную иерархическую конструкцию. Поэтому выбирайте опцию Update Occurrences.

(Это должно быть установлено по умолчанию).

Примечание: При выборе параметра Update Occurrences может появиться предупреждающее сообщение. Игнорируйте это сообщение, потому что для всех сложных иерархических конструкций режим Occurrences является предпочтительным режимом.

6. Для остальных параметров примите значения по умолчанию и нажмите кнопку ОК, чтобы сохранить настройки.

Появляется предупреждающее окно Undo Warning.

7. Нажмите кнопку Yes. Появится окно с сообщением о том, что проводится аннотация.

8. Нажмите кнопку ОК.

Ваша конструкция аннотирована и сохраняется. Вы можете просмотреть значение обновлённых указателей ссылок на странице Schematic. Для этого в главном меню на вкладке Windows откройте окно Session Log (рис. 8.4).

Предупреждение:

При выборе команды Annotate после генерации списка соединений для редактора плат PCB Editor, вы получите сообщение об ошибке, при которой аннотирование на данном этапе может привести к тому, что плата выйдет из

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

синхронизации с эскизным проектом. Это может привести к дальнейшим проблемам повторной аннотации (backannotation).

8.2. Создание отчёта перекрёстных ссылок

Используя Capture, можно создать перекрёстные справочные отчёты для всех компонентов в вашей схеме. Отчёт о перекрёстных ссылках содержит информацию, такую как название компонента, ссылку на компонент и библиотеку, из которой компонент был выбран.

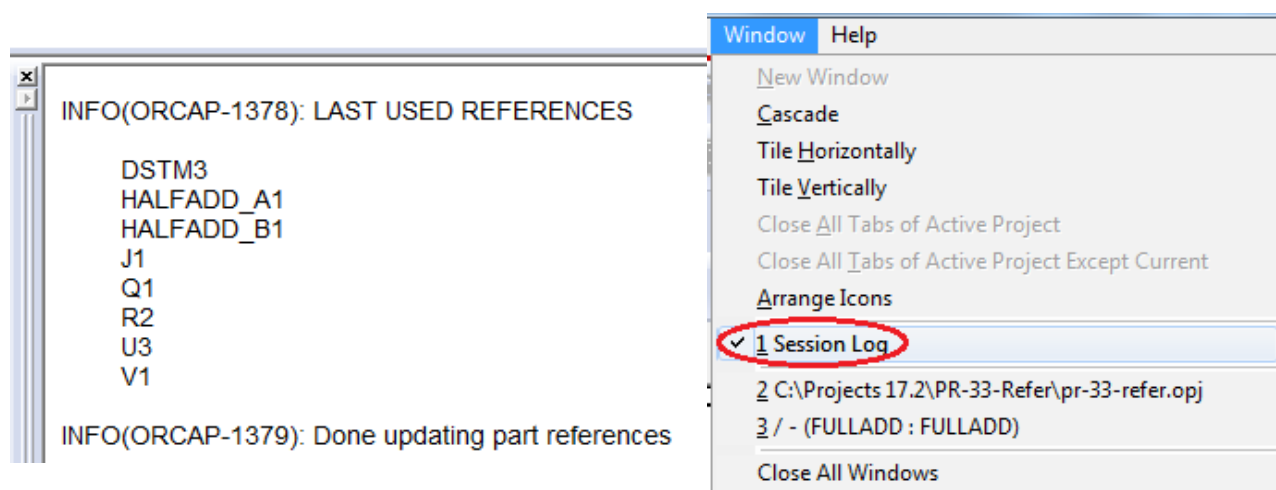


Рис. 8.4. Просмотр указателей ссылок

Для создания отчета о перекрёстных ссылках с помощью Capture сделайте следующее:

1. В Менеджере проектов выделите файл pr-33-refer.dsn.
2. В меню Tools выберите Cross References.

В качестве альтернативы, вы можете выбрать кнопку перекрёстных

ссылок на панели: 

2. В диалоговом окне Cross Reference Parts убедитесь, что выбрана кнопка опции Cross reference entire design.

Примечание: Если вы хотите сгенерировать отчет перекрестных ссылок для конкретной схемной папки, выберите схемную папку перед открытием

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

диалогового окна Cross Reference Parts, а затем выберите опцию Cross References.

3. В разделе Mode выберите кнопку Use Occurrences option .

Примечание: Не обращайте внимания на предупреждение, которое отображается при выборе режим Use Occurrences. Для сложной иерархической конструкции, вы должны всегда использовать этот режим.

4. Укажите отчёт, который вы хотите сгенерировать.

5. В случае, если вы хотите, чтобы отчёт отображался автоматически, установите флажок View Output.

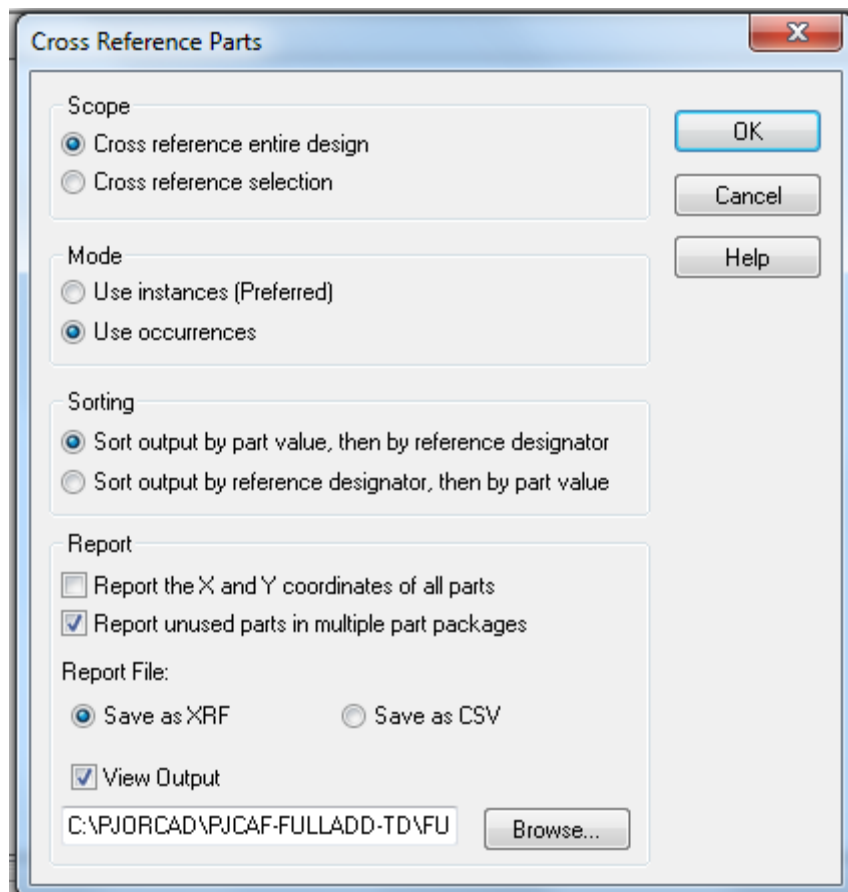


Рис. 8.5. Установка параметров отчета

6. Нажмите кнопку ОК для создания отчёта.

Образец выходного отчета показан на рис. 8.6.

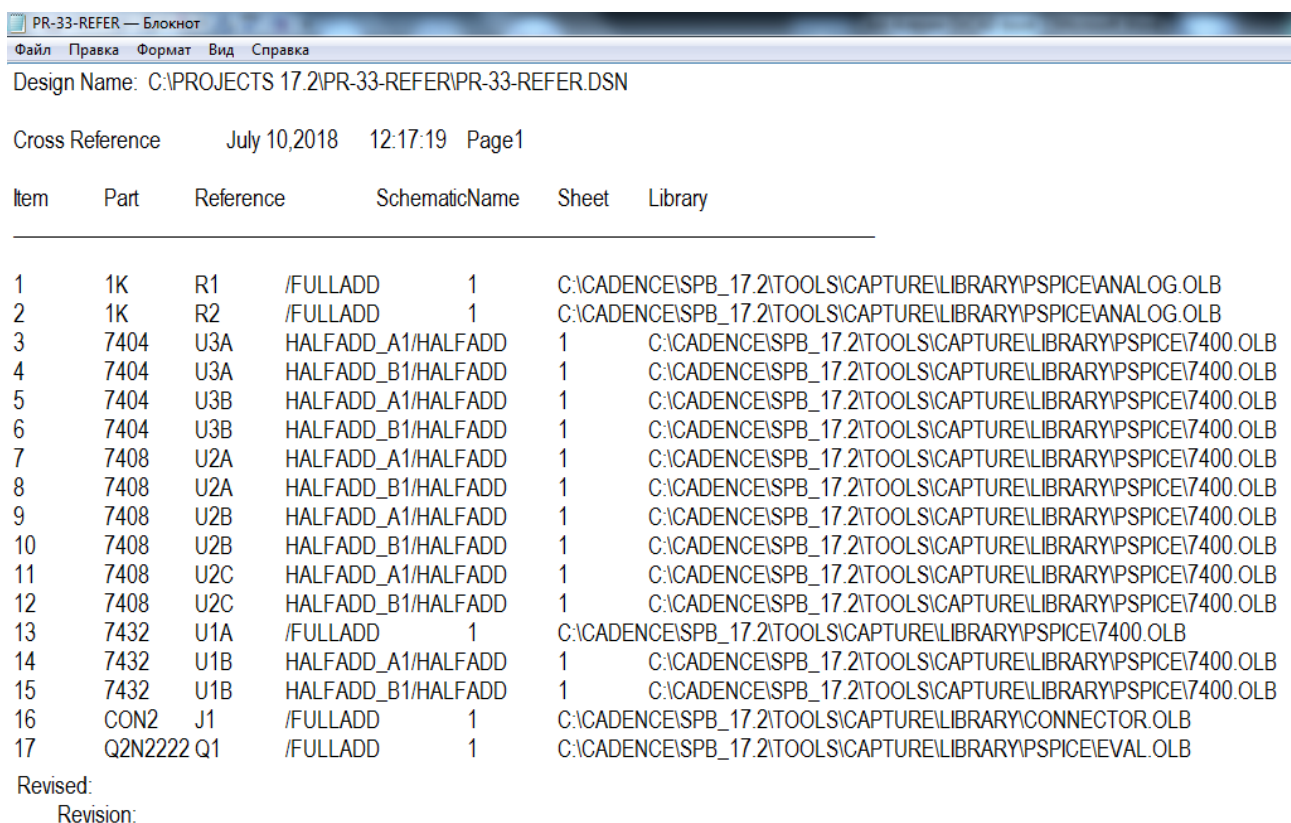
Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

8.3. Создание списка материалов

После того, как вы завершили свой проект, вы можете использовать Capture, чтобы создать спецификацию материалов Bill of Materials (BOM). Спецификация материалов является составным списком всех элементов, которые необходимы для разработки печатных плат.

Используя Capture, можно создать BOM для электрических, а также неэлектрических компонентов, таких как винты. Стандартный отчет BOM включает в себя наименование, количество, ссылки на компонент и значение компонента.

Создадим спецификацию для проекта FULLADD. Откроем этот проект.



PR-33-REFER — Блокнот

Файл Правка Формат Вид Справка

Design Name: C:\PROJECTS 17.2\PR-33-REFER\PR-33-REFER.DSN

Cross Reference July 10, 2018 12:17:19 Page 1

Item	Part	Reference	SchematicName	Sheet	Library
1	1K	R1	/FULLADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\ANALOG.OLB
2	1K	R2	/FULLADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\ANALOG.OLB
3	7404	U3A	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
4	7404	U3A	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
5	7404	U3B	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
6	7404	U3B	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
7	7408	U2A	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
8	7408	U2A	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
9	7408	U2B	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
10	7408	U2B	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
11	7408	U2C	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
12	7408	U2C	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
13	7432	U1A	/FULLADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
14	7432	U1B	HALFADD_A1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
15	7432	U1B	HALFADD_B1/HALFADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\7400.OLB
16	CON2	J1	/FULLADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\CONNECTOR.OLB
17	Q2N2222	Q1	/FULLADD	1	C:\CADENCE\SPB_17.2\TOOLS\CAPTURE\LIBRARY\SPICE\EVAL.OLB

Revised:
Revision:

Рис. 8.6. Образец выходного отчета перекрестных ссылок

Для создания отчёта спецификации:

1. В окне диспетчера проекта, выберите файл проекта pr-33-refer.dsn.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

2. В меню Tools выберите Bill of Materials.

3. Для создания отчёта спецификации для всего дизайна, убедитесь, что выбрана кнопка Process entire design.

4. Для сложных иерархических конструкций, предпочтительным режимом является режим возникновения. Поэтому используйте кнопку Use Occurrences.

Примечание: В случае, если вы получаете предупреждение о том, что это не предпочтительный режим, игнорируйте предупреждение.

5. Укажите имя отчета BOM, которое будет генерироваться. Для нашего дизайна, надо принять имя FULLADD.BOM (рис. 8.7).

Примечание: По умолчанию отчет будет назван designname.BOM.

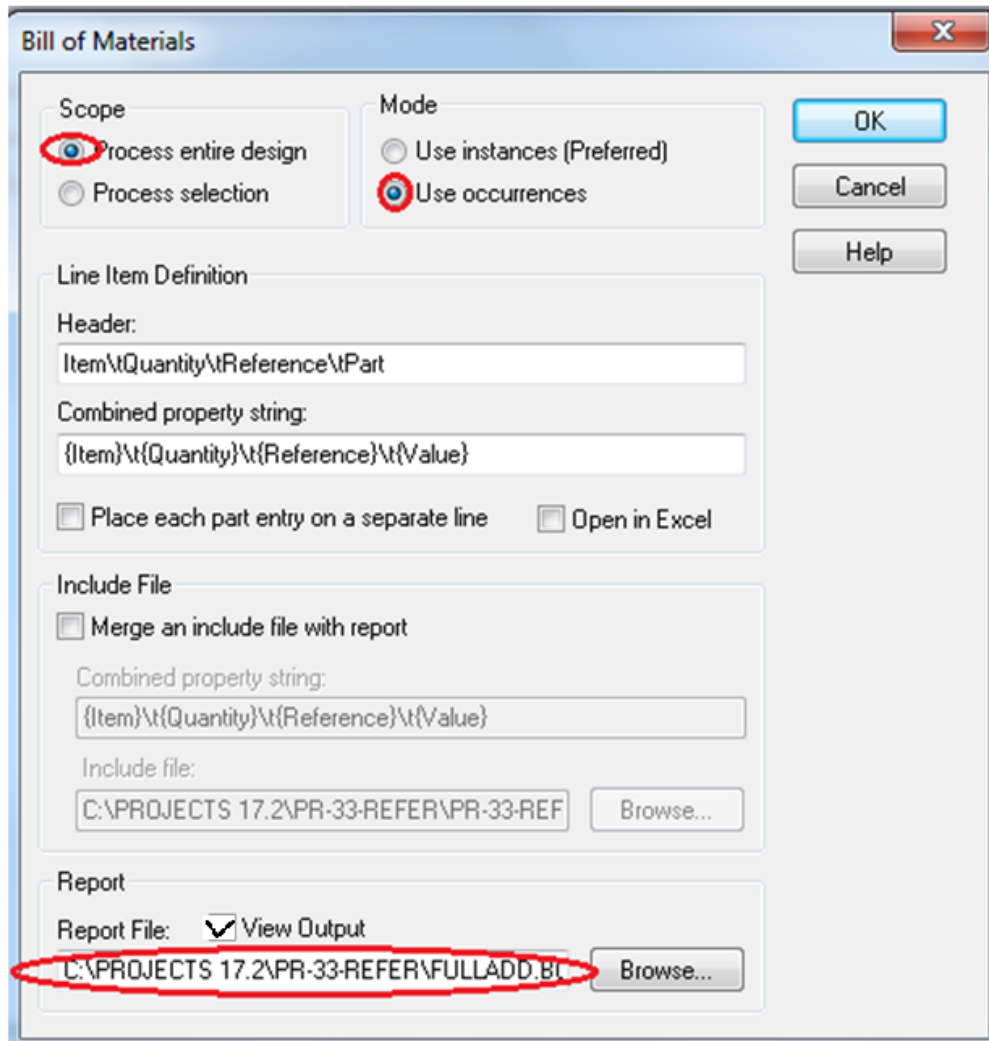


Рис. 8.7. Установки для создания списка материалов

6. Нажмите кнопку ОК.

Отчёт BOM генерируется. Пример отчёта показан на рис. 8.8.

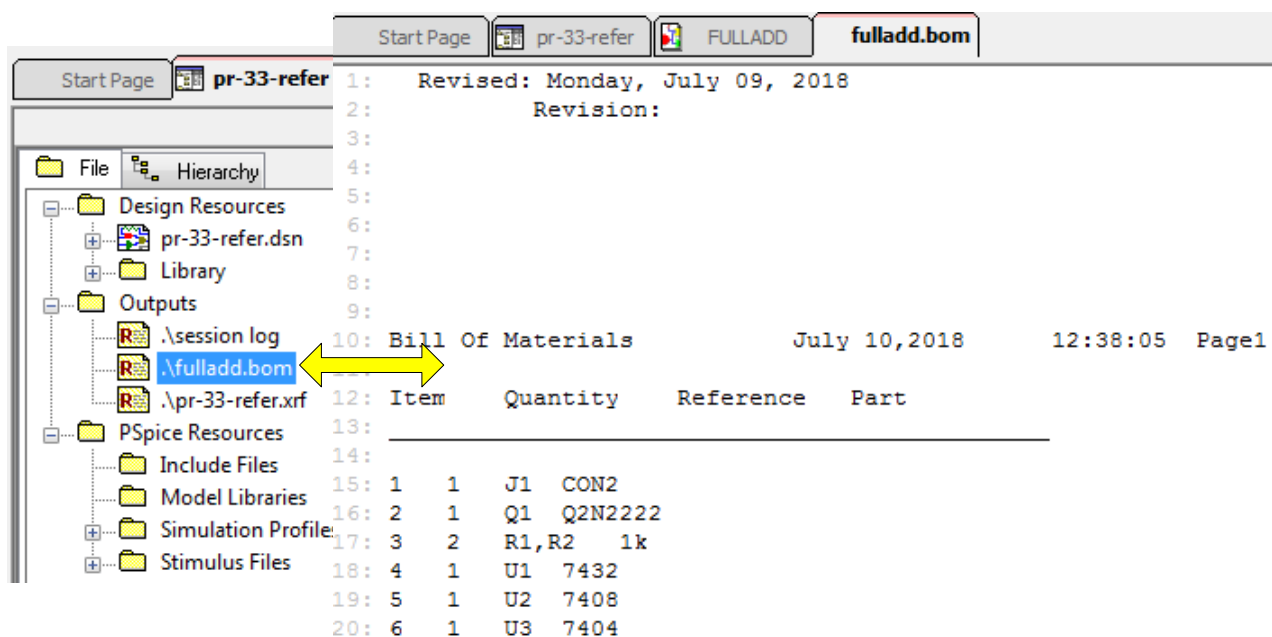


Рис. 8.8. Пример отчета по спецификации материалов

8.4. Добавление специфических свойств редактора PCB

Для того, чтобы быть в состоянии передать ваш проект в OrCAD PCB Editor для размещения компонентов и разводки, вам нужно добавить информация о корпусе (footprint – футпринт) для каждого из компонентов в проекте. По умолчанию некоторые футпринты, доступные для всех компонентов из Pspice - совместимых библиотек, находятся в <install_dir>/tools/capture/library/pspice.

Тем не менее, эти футпринты могут оказаться не действительными. Вам нужно будет изменить их до допустимых значений футпринта корпуса.

Вы можете добавить информацию о футпринте на этапе проектирования схемы в OrCAD Capture или на этапе проектирования платы в инструментах компоновки печатной платы. В этом разделе вы научитесь добавлять информацию о футпринтах к компонентам во время этапа проектирования схемы.

Для добавления информации о корпусе к вентилю ИЛИ 7432 на странице схемы FULLADD выполните следующие действия:

1. Щёлкните правой кнопкой мыши на логическом элементе ИЛИ и выберите Edit Properties .

Появится окно редактора свойств.

2. На вкладке Filter из раскрывающегося списка выберите Allegro-PCB Designer (Рис. 8.9).

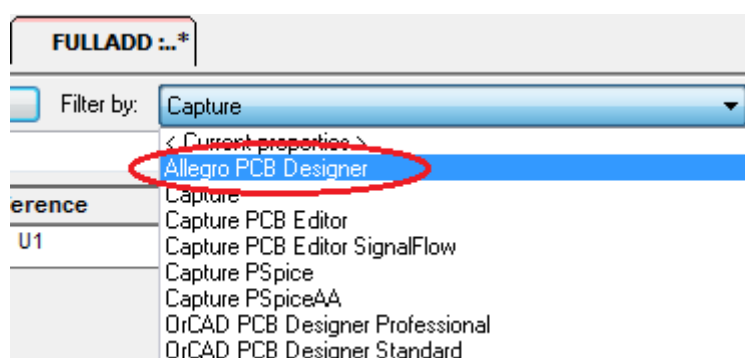


Рис. 8.9. Выбор программы Allegro PCB Designer

Столбцы в таблице отображают свойства корпуса для PCB Editor.

3. Чтобы изменить значение свойства PCB Footprint, нажмите на соответствующую ячейку и введите значение SOIC14 (рис. 8.10).

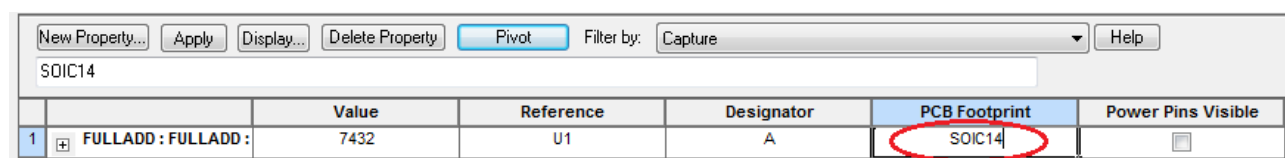


Рис. 8.10. Выбор корпуса элемента ИЛИ

4. Нажмите кнопку Apply или кнопку ENTER.

5. Сохраните изменения и закройте окно Редактора свойств.

6. Добавьте в PCB информацию о корпусах для всех компонентов в схеме.

Для резисторов введите RES500. Для коннектора- JUMPER2. Для транзистора TO18.

Для компонентов полусумматоров надо выбрать футпринт SOIC14.

Ваша схема теперь готова быть к передаче в OrCAD PCB Editor для размещения компонентов и разводки.

8.5. Проверка правил проектирования

После того, как вы завершили свой проект, рекомендуется выполнить проверку правил проектирования (Design rules check - DRC), чтобы изолировать любые нежелательные ошибки проектирования, которые могут быть в схеме.

Чтобы запустить DRC для схемы полного сумматора, выполните следующие шаги:

1. В окне менеджера проекта, выберите файл проекта.
2. В меню Tools выберите команду Design Rule Checks (рис. 8.11).

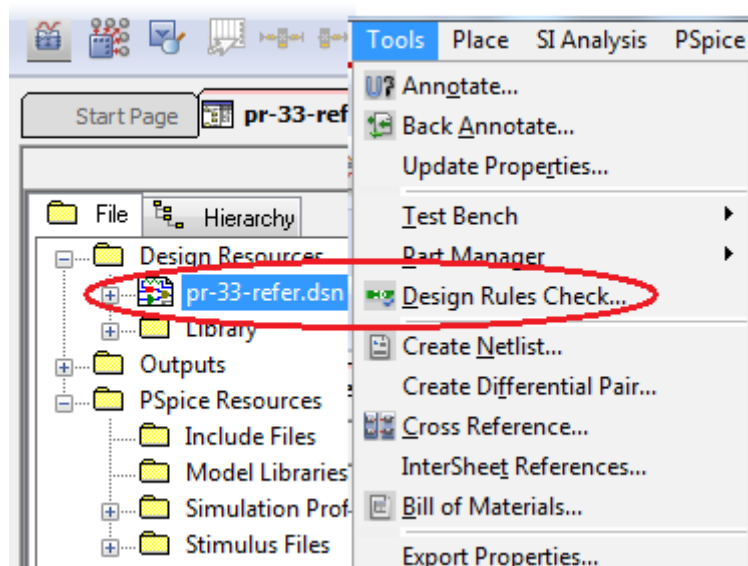


Рис. 8.11.

Примечание: В качестве альтернативы, вы можете выбрать DRC кнопкой на панели инструментов.

3. В диалоговом окне DRC вкладка Design Rules Check выбирается по умолчанию. Укажите свои предпочтения (рис. 8.12).

По умолчанию выбирается кнопка Check entire design (проверить весь вариант дизайна). Чтобы запустить DRC на всю разработку, примите выбор по умолчанию.

4. Выберите кнопку опции Use Occurrences.

Примечание: Для сложных иерархических конструкций Use Occurrences режим является предпочтительным режимом. Поэтому игнорируйте предупреждение, которое отображается при выборе этого режима.

5. Для запуска DRC, выберите кнопку опции Check design rules под Action.

6. В разделе Report выберите то, что требуется указать в DRC отчете.

7. Установите флажок View Output.

Если этот флажок установлен, отчет DRC автоматически открывается для просмотра после полной проверки.

8. В текстовом поле Report File, укажите имя и расположение DRC файла, который будет создан.

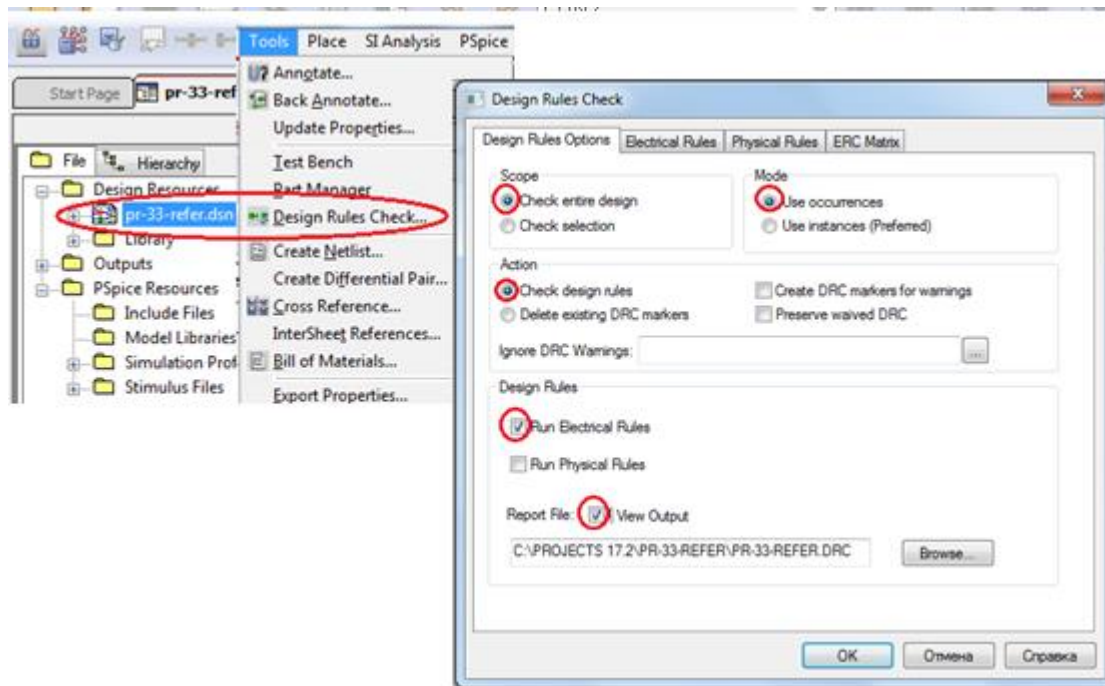


Рис. 8.12. Установка проверки правил проектирования

9. На вкладке Electrical Rules установите флажки на нужных вам проверках (рис. 8.13).

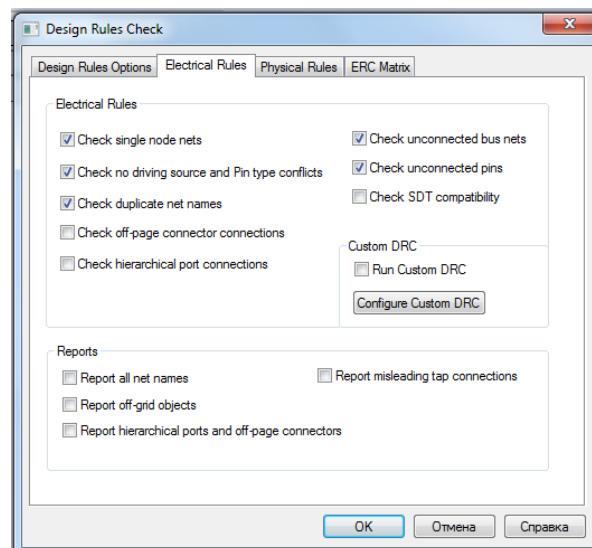


Рис. 8.13. Установка проверок электрических правил

9. Нажмите кнопку ОК.

После выполнения проверки отчет DRC отображается в формате, показанном ниже.

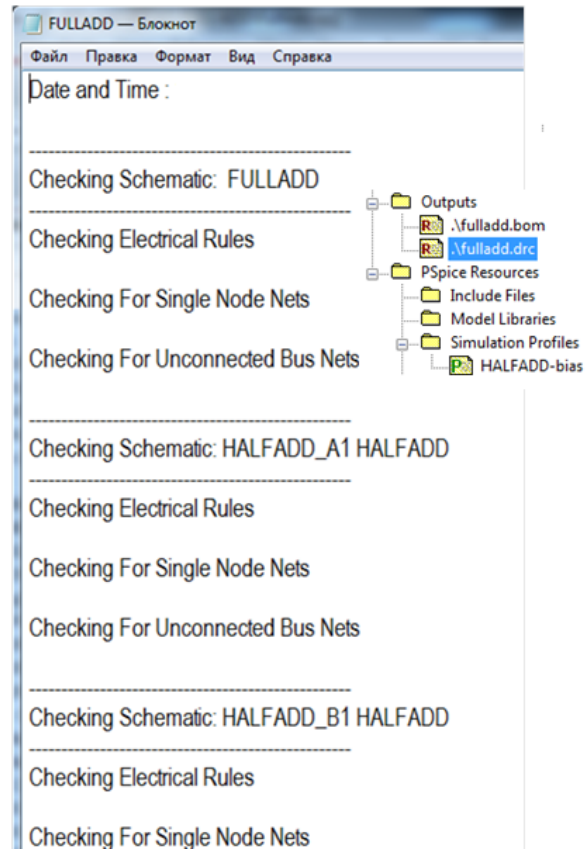


Рис. 8.14. Отчет проверки правил проектирования

На рис. 8.15 показана проверка матрицы ERC.

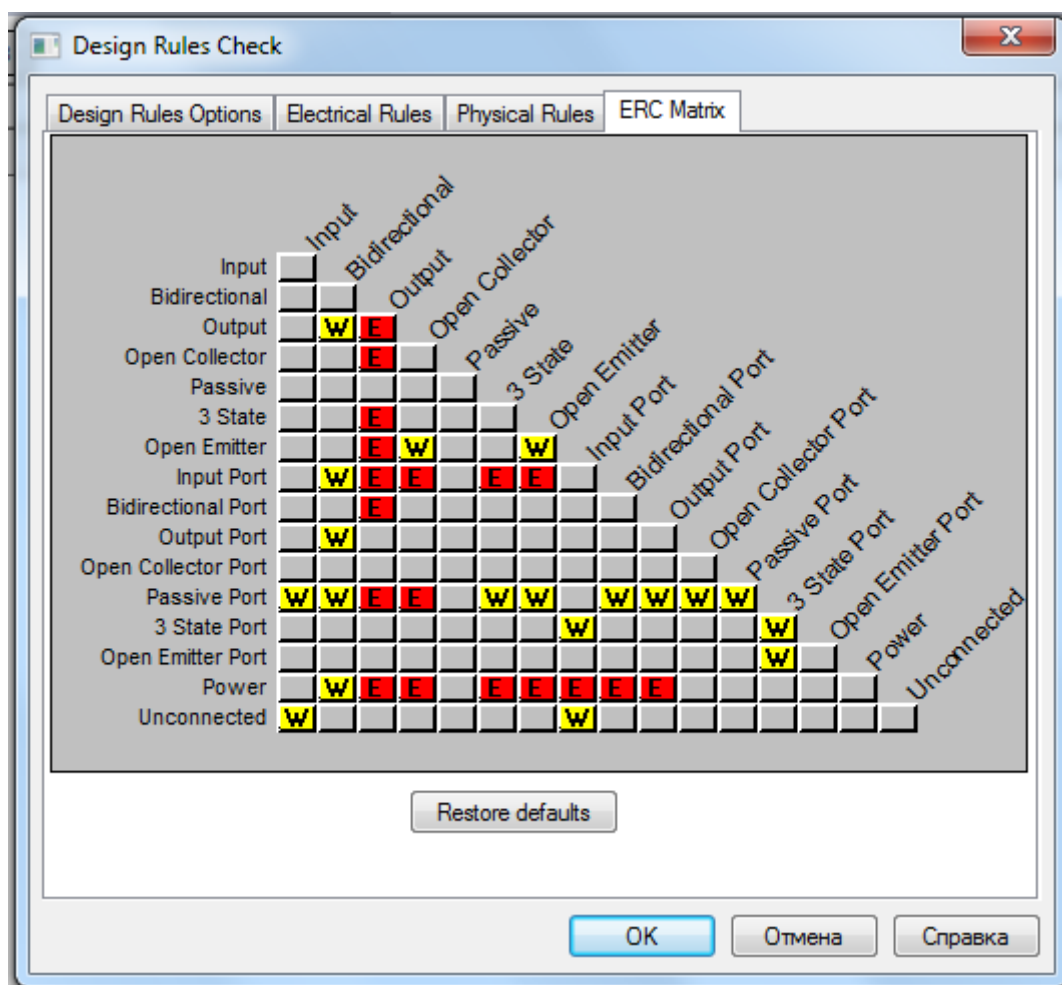


Рис. 8.15. Проверка матрицы электрических правил (ERC)

В этом разделе были рассмотрены шаги для создания плоских и иерархических проектов с использованием OrCAD Capture. Вы были ознакомлены с основными задачами создания проекта, такими как создание схем, добавление библиотеки в проект, размещение компонентов и редактирование свойств.

Далее познакомимся с циклом проектирования печатных плат.

8.6. Контрольные вопросы

1. Какие задачи выполняются для обработки проекта в OrCAD Capture ?
2. Для чего требуется назначать ссылки на компоненты схемы ?

3. Как выполняют ручное аннотирование проектов ?
4. Как просмотреть значения обновленных указателей ссылок компонентов ?
5. Какую информацию содержит отчет о перекрестных ссылках ?
6. Как можно создать отчет о перекрестных ссылках ?
7. Как создают спецификацию материалов (Bill of Materials) ?
8. Как добавить в проект информацию о корпусах компонентов ?
9. Как выполняют проверку правил проектирования (DRC) ?
10. Какие правила проверяют включает DRC ?

Глава 9. Проектирование печатных плат с использованием

OrCAD PCB Editor

9.1. Обзор

Редактор печатных плат OrCAD (на основе Allegro® PCB) – этот мощный и гибкий инструмент для компоновки и трассировки печатных плат позволяет PCB (PCB-printed circuit board) дизайнерам создавать и использовать данные для сквозного проектирования электронных устройств.

Это интерактивная среда для создания и редактирования сложных, многослойных печатных плат. Набор функций, который предоставляет, предназначен для широкого спектра современных разработок и технологических задач.

В этой главе вы будете использовать OrCAD PCB Editor, чтобы на основе проекта полного сумматора, созданного ранее, довести эскизное проектирование до печатной платы. Здесь рассмотрены некоторые из общих задач, решаемых в редакторе печатных плат. В процессе изучения вы также можете использовать кросс-проверки между Capture и PCB Editor.

Для того, чтобы пройти все шаги, описанные в этой главе, вы должны иметь готовый дизайн полного сумматора. Полный сумматор, используемый
Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

нами, представляет собой иерархическую конструкцию. Она состоит из двух экземпляров иерархического блока HALFADD.

Вы можете использовать конструкцию, созданную ранее. Однако в учебной программе Orcad 17.2 Lite установлено ограничение на максимальное количество компонентов в схеме и некоторые важные подготовительные операции (например, DRC – проверку правил проектирования для полного сумматора) не удастся выполнить. Поэтому мы будем использовать файлы дизайна, которые поставляется с обучающей программой.

Файлы дизайна полного сумматора доступны как файл flowtut.zip, который расположен в <Install_dir> / DOC / flowtut / tutorial_example.

Распакуйте файл flowtut.zip и извлеките его в пустой каталог, скажем orcad_flow. После извлечения файла flowtut.zip вы найдёте два папки - частичные (Partial) и полные (Complete), созданные в каталоге orcad_flow.

Каталог *partial* содержит файлы, созданные в предыдущих разделах. Используйте файлы этого каталога, только если вы хотите пропустить шаги по созданию дизайна, выполненные ранее и непосредственно перейти к этой главе.

Полный каталог *complete* содержит все файлы, созданные ранее в руководстве [3].

Вы можете использовать файлы полного каталога, чтобы проверить свои результаты.

Важное замечание !

Проектирование печатных плат – это сложный и достаточно трудоемкий процесс, требующий многих навыков и опыта, который приходит не сразу. Компания Cadence представила в Интернете полезные обучающие видеоролики. В этой главе Вы изучите основы работы в учебных версиях OrCAD PCB Editor Lite и OrCAD PCB Router Lite, которые помогут вам овладеть искусством проектирования.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

9.2. Подготовка в Capture

Для того, чтобы быть в состоянии использовать в PCB Editor проект, созданный в Capture, необходимо выполнить некоторые задачи. Некоторые из этих задач выполняются в Capture, в то время как остальные выполняются в среде редактора печатных плат.

Задачи, которые должны быть выполнены в Capture:

- Запуск проверки правил проектирования – DRC.
- Создание списка соединений PCB Editor netlist.

Запуск DRC был выполнен в предыдущем разделе. Полезно повторить это для проекта, запущенного из каталога *complete* и убедиться в совпадении результатов.

9.2.1. Создание списка соединений для редактора печатных плат

После запуска проверки правил проектирования вы создаёте в Capture список соединений для PCB Editor. Для этого:

1. В окне менеджера проекта, выберите файл `fulladd.dsn`.

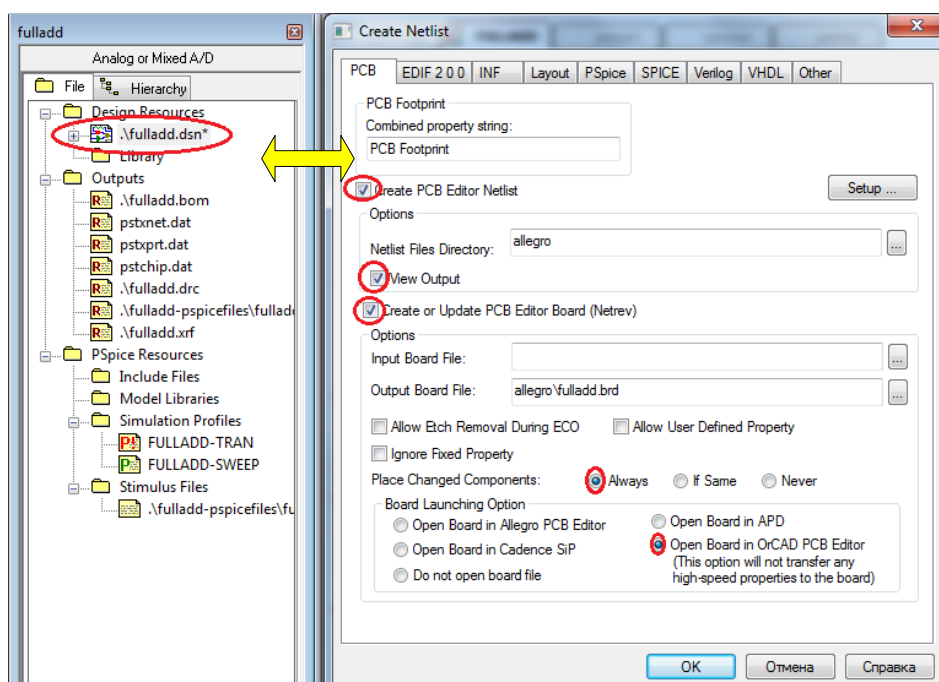


Рис. 9.1. Создание списка соединений

2. В меню Tools в Capture выберите Create Netlist. Появится диалоговое окно Create Netlist.

3. Выберите вкладку PCB (если она ещё не выбрана).

Флажок Create PCB Editor Netlist выбирается по умолчанию. При установке этого флажка будет сгенерирован Список соединений в формате редактора печатных плат, который состоит из следующих трёх файлов: PSTCHIP.DAT, PSTXNET.DAT и PSTXPRT.DAT.

PSTCHIP.DAT: этот файл содержит описание для каждого отличного типа компонентов, которые используются в дизайне.

- PSTXNET.DAT: этот файл подключения, называемый также плоский список или расширенный список соединений, содержит каждую сеть, её свойства, подключённые к ней узлы и свойства узлов.

- PSTXPRT.DAT: этот файл, также упоминается как расширенный список компонентов, содержит список физических компонентов и перечисляет каждое

позиционное обозначение и секции, связанные с ним, указанные через позиционное обозначение и номер секции.

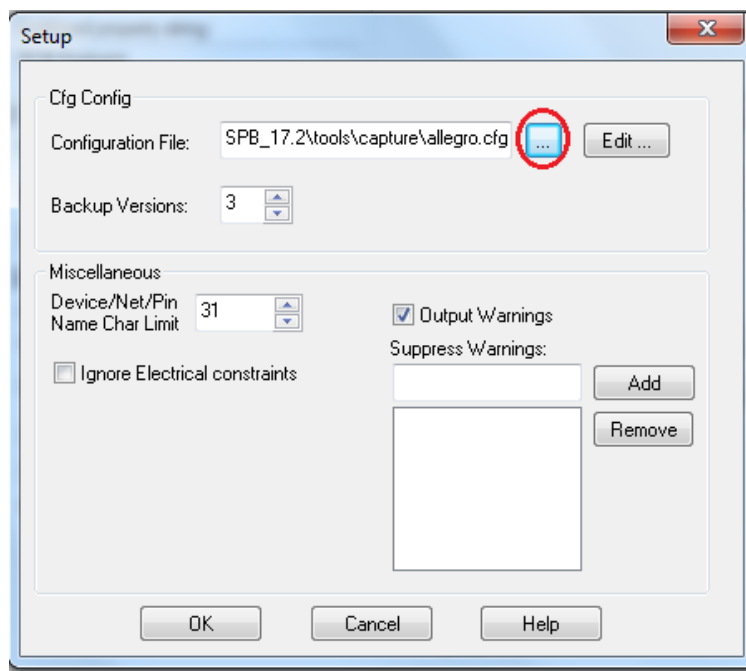


Рис. 9.3. Установка файла конфигурации

Примечание: Убедитесь, что правильный файл конфигурации (Allegro.cfg) задаётся в диалоговом окне настройки. Для просмотра файла конфигурации нажмите кнопку Setup. Путь к файлу конфигурации должен быть выбран, используя кнопку Browse: <install_dir> \ Tools \ capture \ allegro.cfg, где <Install_dir> это место установки (рис. 9.2) .

Примечание: Текстовое поле Netlist Files Directory содержит местоположение каталога, в котором будут сохранены файлы PST * .DAT. Расположение по умолчанию является Allegro subdirectory в каталоге вашего дизайна.

4. Установите флажок View Output, чтобы автоматически открыть три PST * .DAT файла списка соединений в отдельных окнах Capture для просмотра и редактирования после того, как netlisting завершится.

5. Установите флажок **Create or Update PCB Editor Board (Netrev)**, чтобы создать в PCB Editor плату, которая соответствует списку соединений, который вы генерируете.

Примечание: Текстовое окно **Output Board File** содержит имя платы, которое в данном случае является **fulladd.brd** и расположение директории, где будет создан файл платы, которая в этом случае будет **\allegro**.

6. Выберите **Open Board** в опции **OrCAD PCB Editor**, чтобы открывать файл **Output Board File** в OrCAD PCB Editor автоматически после того, как завершается netlisting.

7. Нажмите кнопку **OK** в диалоговом окне **Create Netlist**.

Появится сообщение с запросом, чтобы сохранить ваш дизайн до создания списка соединений. Нажмите кнопку **OK**.

Если имеются указания об ошибках, их следует найти и устранить.

Происходит генерация файлов (рис.9.4

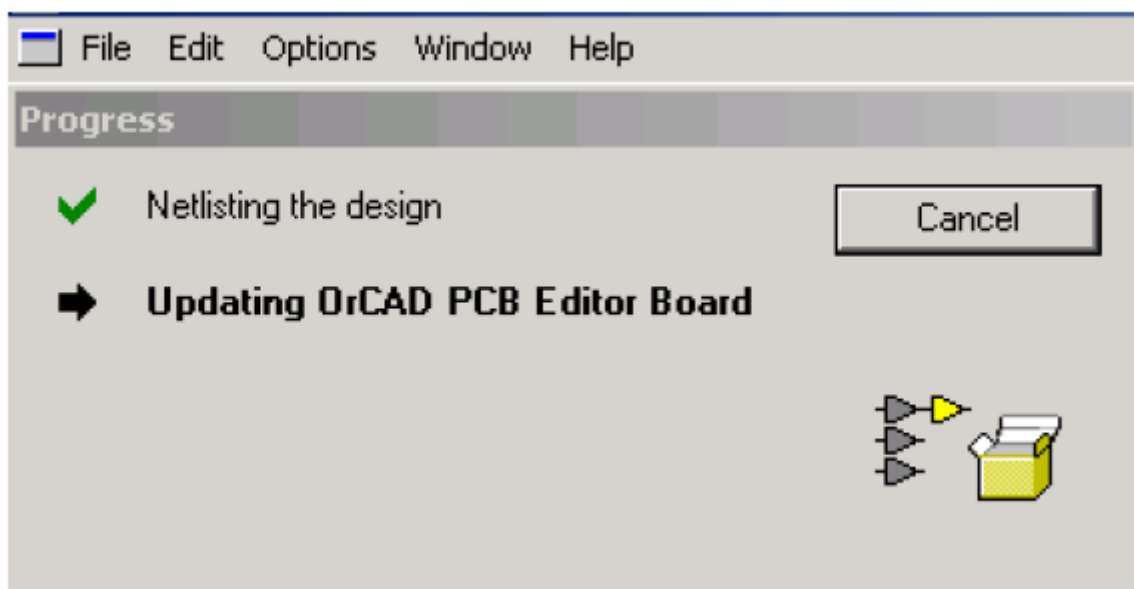


Рис. 9.4. Генерация файлов списка соединений

После этого Capture генерирует файлы списка соединений (PSTCHIP.DAT, PSTXPRT.DAT и PSTXNET.DAT) и файл платы (Fulladd.brd) в указанном месте каталога, который в этом случае будет: \complete\allegro. Кроме того, открываются файлы списка соединений в отдельных окнах Capture и они появляются под Outputs directory в окне менеджера проекта (рис. 9.5).

Фрагменты содержания файлов показаны на рис. 9.5.

Файл платы fulladd.brd откроется в редакторе OrCAD PCB Editor.

8. После выполнения команды Create Netlist открывается окно редактора печатных плат OrCAD PCB Design (рис. 9.6). Выбрав Display>Windows, можно открыть нужные окна и панели.

Окно редактора OrCAD PCB Design содержит большое количество вспомогательных панелей и инструментов, с которыми мы будем знакомиться по мере изучения процесса проектирования печатных плат. Некоторые панели и окна показаны на рис. 9.6. Перемещение объекта в окне проектирования выполняют, используя окно позиционирования.

Примечание: Если окно не открылось, надо проверить в каталоге Cadence «Все программы» загружается или нет программа PCB Design Lite. Если загрузки нет, возможно закончился срок демоверсии. Тогда переустановите программу OrCAD ещё раз.

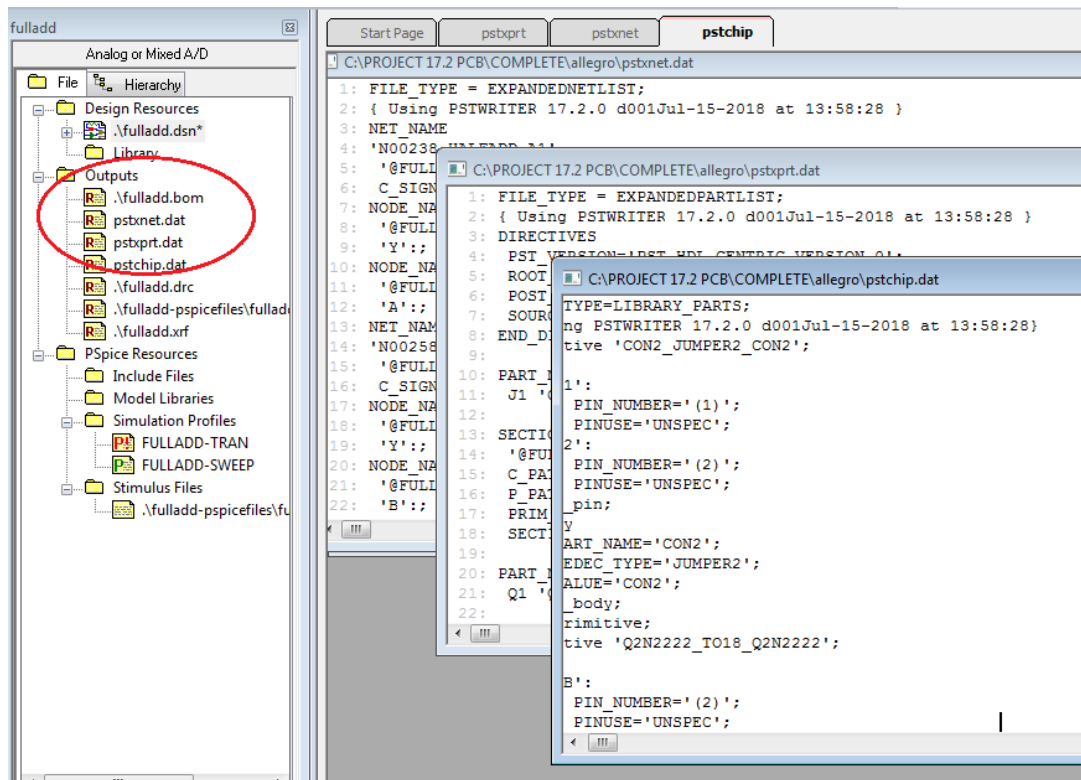


Рис. 9.5. Фрагменты файлов списка соединений

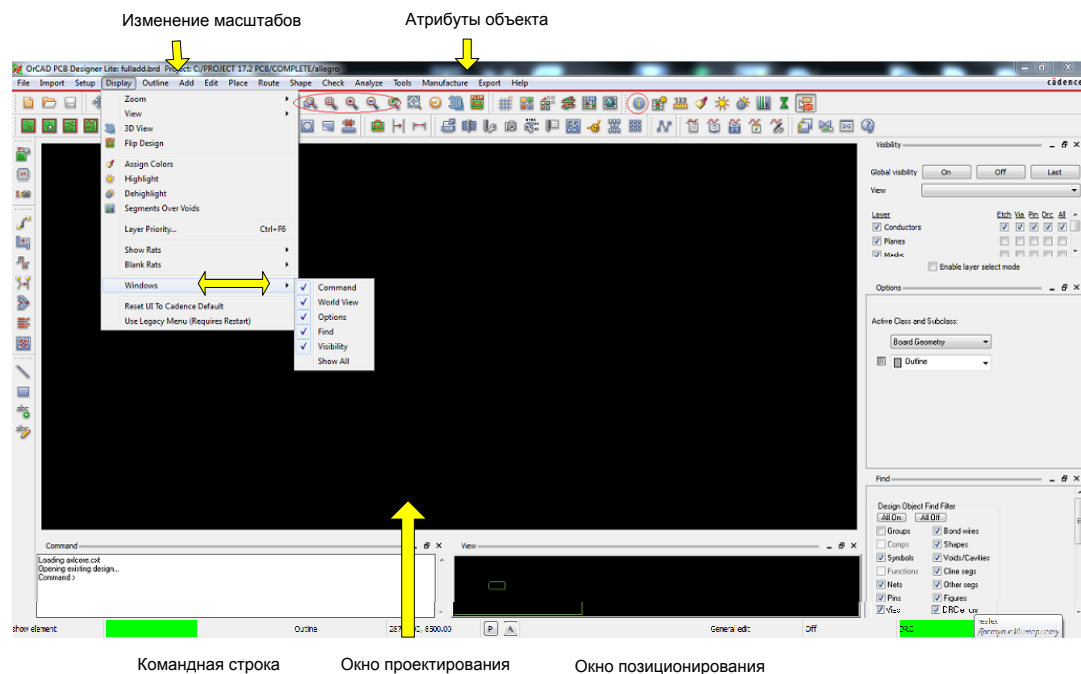


Рис. 9.6. Окно редактора печатных плат OrCAD PCB Design

9.3. Начало работы в PCB Editor

Режимы работы приложения показаны на рис. 9.7.

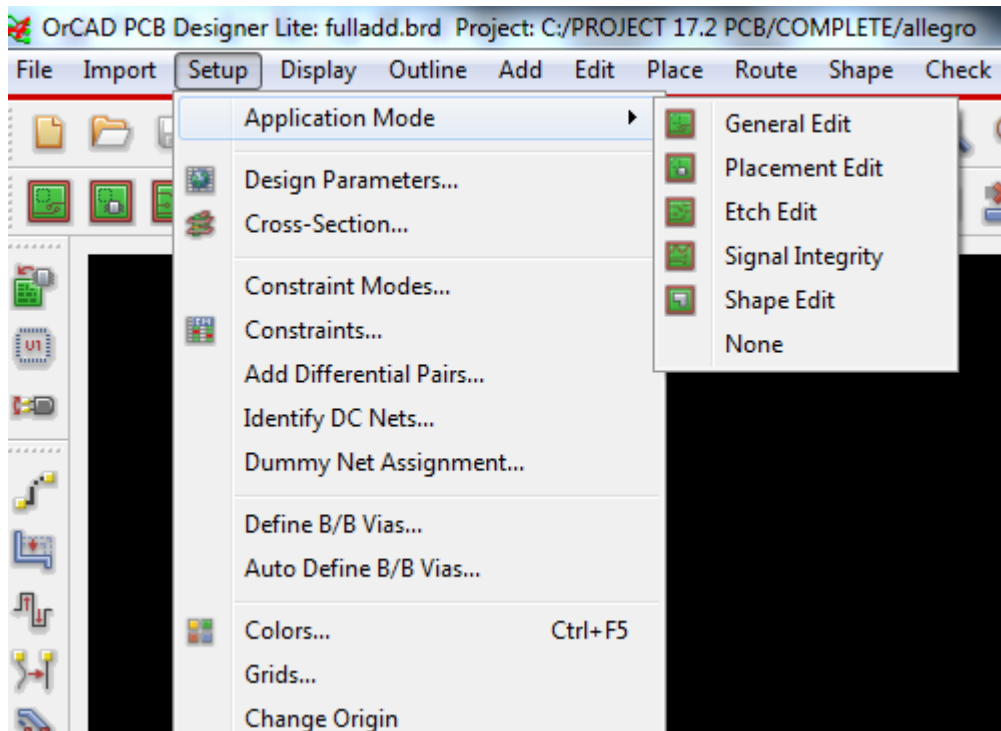


Рис. 9.7. Режимы работы PCB Editor

Режим *General edit* позволяет выполнять операции редактирования, такие как размещение, разводка, перемещение, копирование, зеркальное отображение.

Режим *Placement edit* позволяет редактировать размещение компонентов.

Режим *Etch edit* адаптирует среду для выполнения задач травления, таких как добавление разъемных соединений, настройка задержки, сглаживание клиньев или острых углов сегмента.

Режим *Signal Integrity* обеспечивает быстрый и легкий доступ к часто используемым командам SI.

Shape Edit обеспечивает быстрый и легкий доступ для редактирования границы формы, такие как сглаживание краев формы с углами или без углов, многосегментное сглаживание и добавление надрезов.

Режим *None* позволяет выйти из любого режима и выбрать новый.

Текущий режим работы отражается в строке состояния (рис. 9.8).

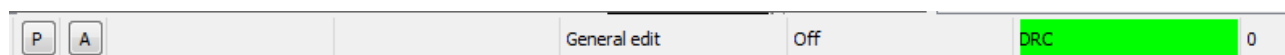


Рис. 9.8. Отображение текущего режима

Параметры проектирования устанавливаются на вкладках *Design Parameter Editor* из меню *Setup* (рис.9.9).

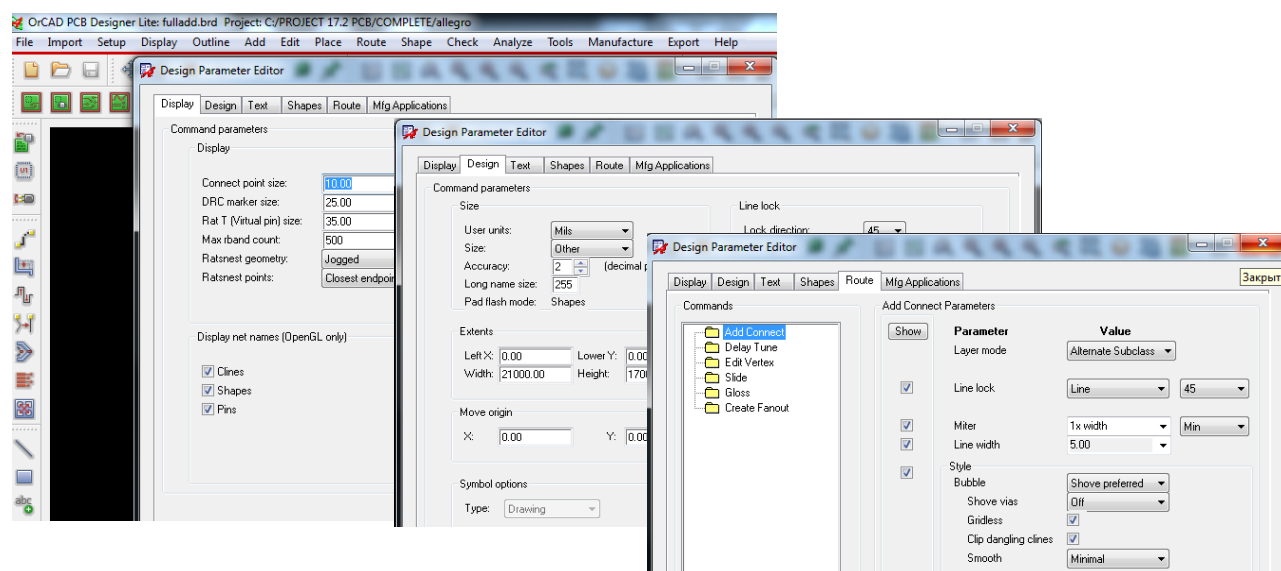


Рис. 9.9. Установка параметров проектирования

9.4. Создание печатной платы

После создания списка соединений для редактора РСВ следующим шагом будет создание новой платы в редакторе печатных плат. *Capture netlist* генерирует файл платы и три файла списка соединений, совместимых с РСВ. Дополнительную информацию см. в разделе «Создание списка соединений для редактора печатных плат».

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

9.4.1. Создание контура платы

Контур платы определяет границы платы. Чтобы создать схему платы в редакторе печатной платы:

1. В меню Add выберите Line. Панель Options изменится, как показано на рис. 9.10.

Примечание. Убедитесь, что панель Options в правой части окна редактора печатной платы отображает Active Class (активный класс) как Board Geometry и подкласс как Outline (контур).

2. Задайте следующие параметры в окне Options:

- а. Окончание (замок) линии Line Lock: Line, 90°
- б. Ширина линии Line Width: 20,0
- с. Строковый шрифт Line Font: сплошной (Solid)

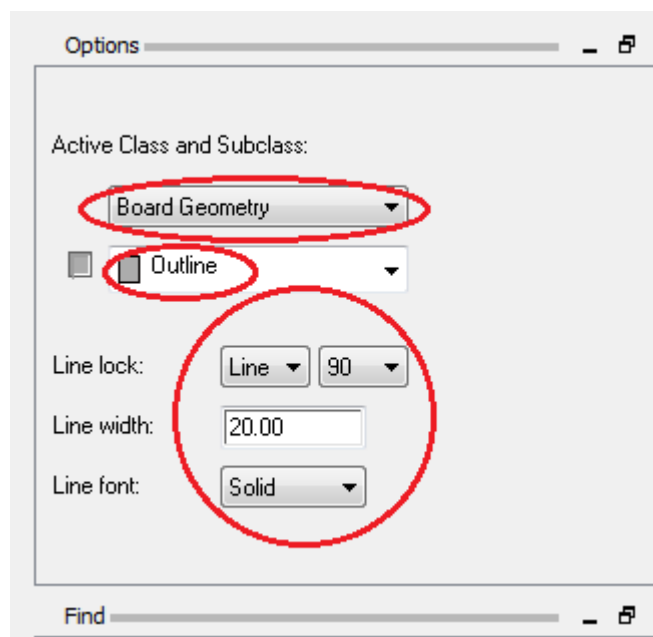


Рис. 9.10. Установка параметров линий

Примечание. Пользовательские единицы по умолчанию в редакторе печатных плат – mils (0,001 дюйма). Чтобы просмотреть пользовательские единицы, выберите Design Parameters в меню Setup. В диалоговом окне Design Param-
Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

eter Editor перейдите на вкладку Design. Пользовательские единицы определены в разделе Size (рис. 9.11).

Примечание. Расстояние по размеру сетки по умолчанию для координат X и Y в редакторе печатной платы составляет 25 мил каждый. Чтобы просмотреть интервал сетки, выберите Setup> Grids. Можно также на вкладке Design Parameter Editor нажать кнопку Setup Grids. Появится диалоговое окно Define Grid. Установите нужные значения и нажмите Ok (рис. 9.12).

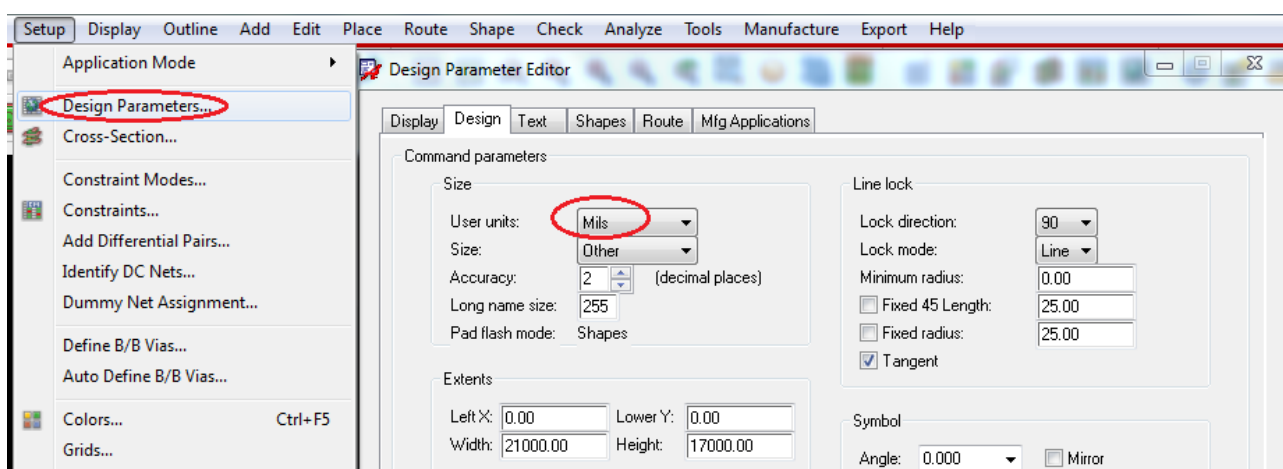


Рис. 9.11. Установка единиц измерения

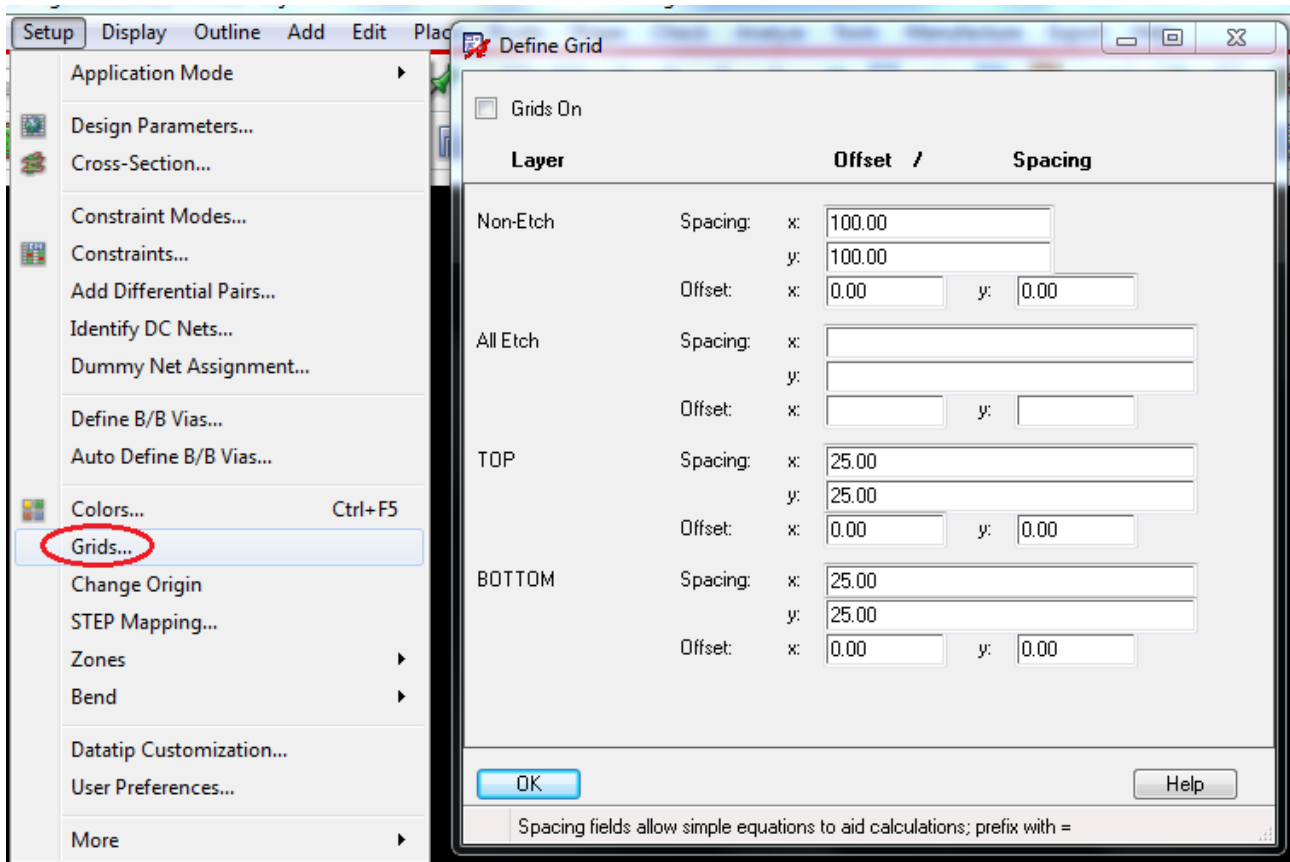


Рис. 9.12. Установка параметров сеток

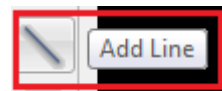
3. Создадим контур платы с координатами:

☐ 3000, 3000

☐ 3000, 5000

☐ 1000, 5000

☐ 1000, 3000



Сначала надо включить режим Add Line на панели инструментов.

Чтобы вставить первый угол контура платы, поместите курсор в координаты: 1000, 3000 и нажмите левую кнопку мыши.

Примечание: При перемещении курсора в окне в характеристиках конструкции координаты будут постоянно меняться. Вы можете просматривать координаты в правом нижнем углу окна редактора печатных плат.

Совет. Можно также использовать команду `pic` в консоли PCB Editor, чтобы указать координаты. Например, чтобы указать отправную точку, введите `1000 3000`.

На рис. 9.13 показан запуск команды `Pick`, выбор типа координат (абсолютные или относительные), позиционирование контура на поле. На рис. 9.14 показан ввод координат и их отображение в окне команд.

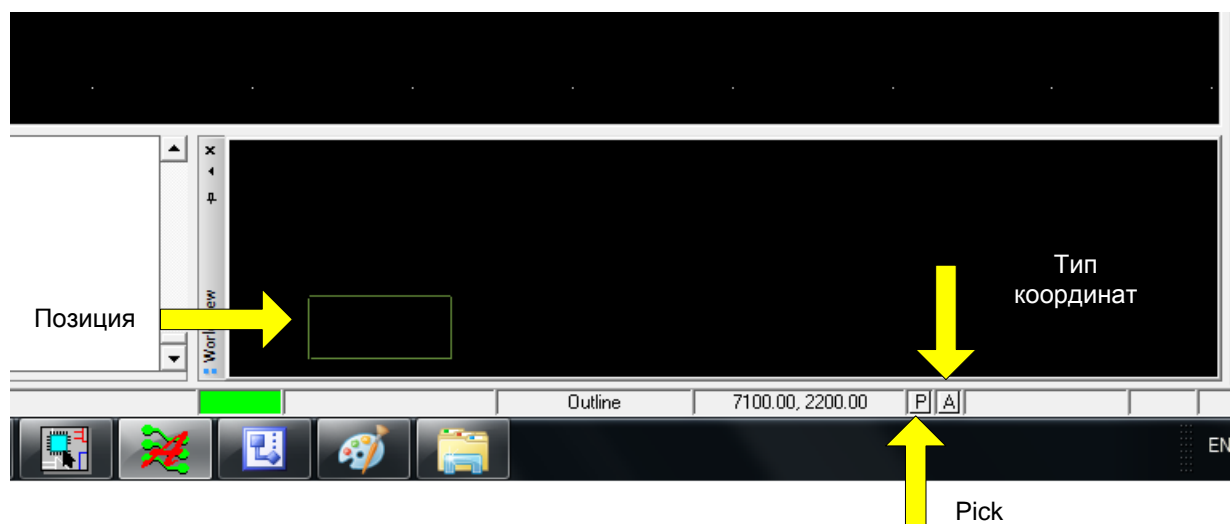


Рис. 9.13. Начало ввода координат платы

4. Заполните оставшийся контур платы, используя следующие координаты (рис. 9.14)

- ☐ 3000, 3000
- ☐ 3000, 5000
- ☐ 1000, 5000
- ☐ 1000, 3000

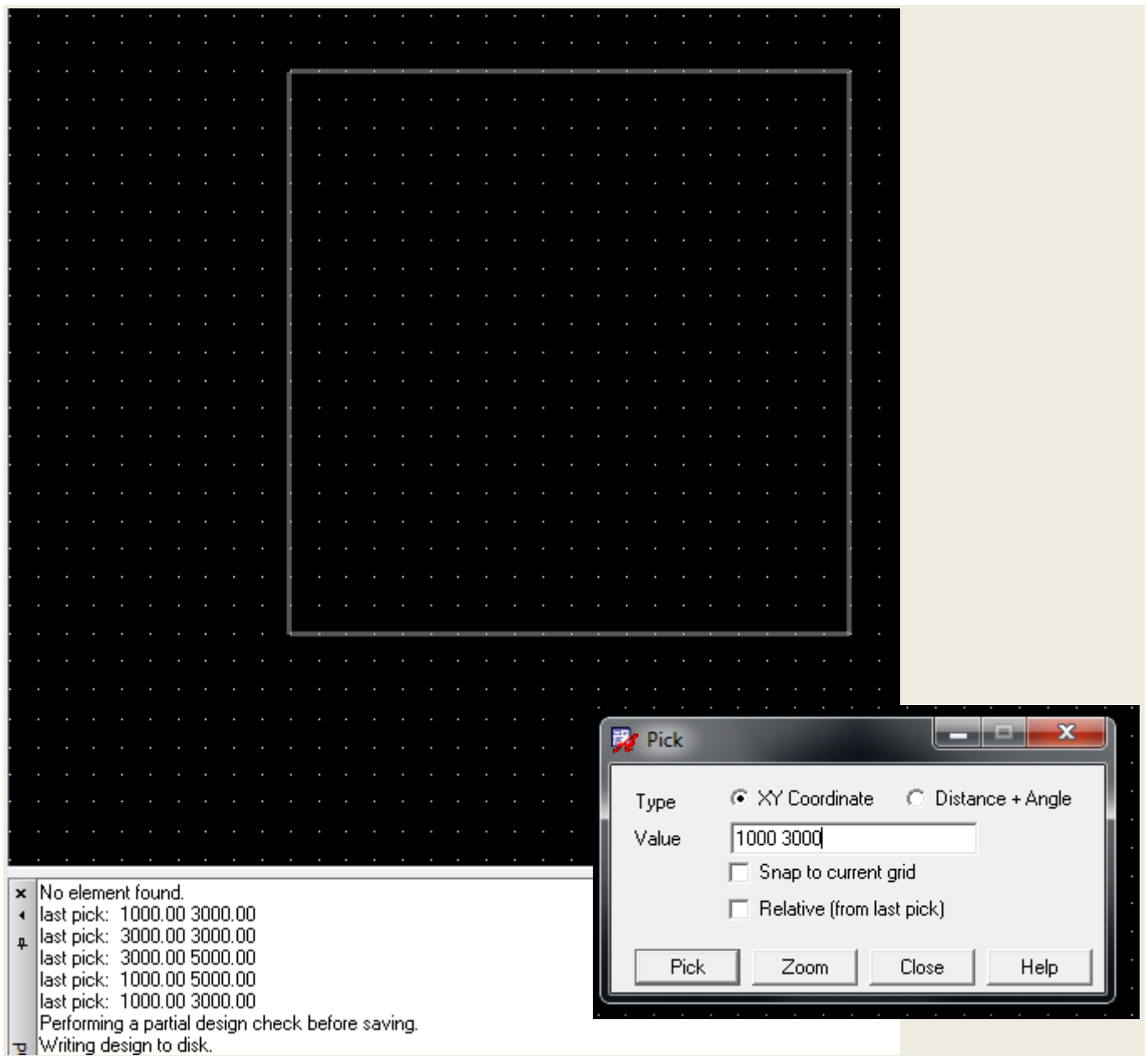


Рис. 9.14. Ввод координат контура платы

5. Когда вы находитесь на последнем повороте и замкнёте контур платы, щёлкните правой кнопкой мыши и выберите Done. Контур платы будет создан.

Примечание: Убедитесь, что контур платы является замкнутым прямоугольником. В нашем примере контур квадратный.

Совет:

Чтобы удалить контур:

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

- ☐ Выберите Delite из меню Edit.
- ☐ В окне Options выберите флажок Cline под группой Delete Net Options.
- ☐ Щелкните левой кнопкой мыши по контуру, чтобы выбрать его.
- ☐ Щелкните правой кнопкой мыши и выберите пункт Done. Контур бу-


дет полностью удалён.

Сохраните файл платы и сделайте выход из PCB Editor.

6. Повторно войдите в PCB Editor из меню <Все программы>CaDENCE< PCB Editor Lite. Откроется файл fulladd.brd с контуром платы.

7. Выберите Zoom Fit из меню View, чтобы отобразить целиком контур вашей платы в окне дизайна, как показано на рис. 9.15

Совет: В качестве альтернативы, вы можете использовать любой из этих методов масштабирования контура платы в окне дизайна:

- ☐ Напечатайте zoom fit в командной строке;
- ☐ -или нажмите на значок .

8. Для включения сетки нажмите в главном меню значок .

9. Для изменения толщины линии контура выделите линию и выберите Line > Change Width.

9.4.2. Добавление монтажных отверстий

После создания контура платы добавим в плату монтажные отверстия.

Чтобы добавить монтажные отверстия в вашей плате, выполните:

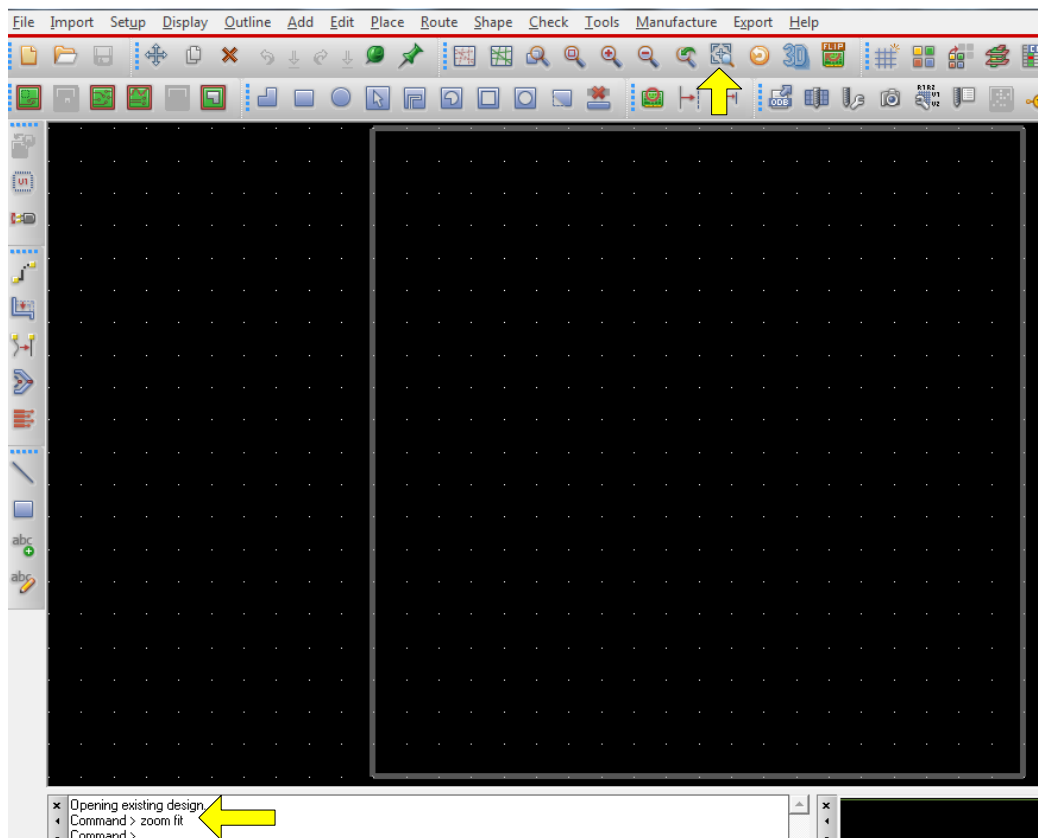


Рис. 9.15. Масштабирование контура платы

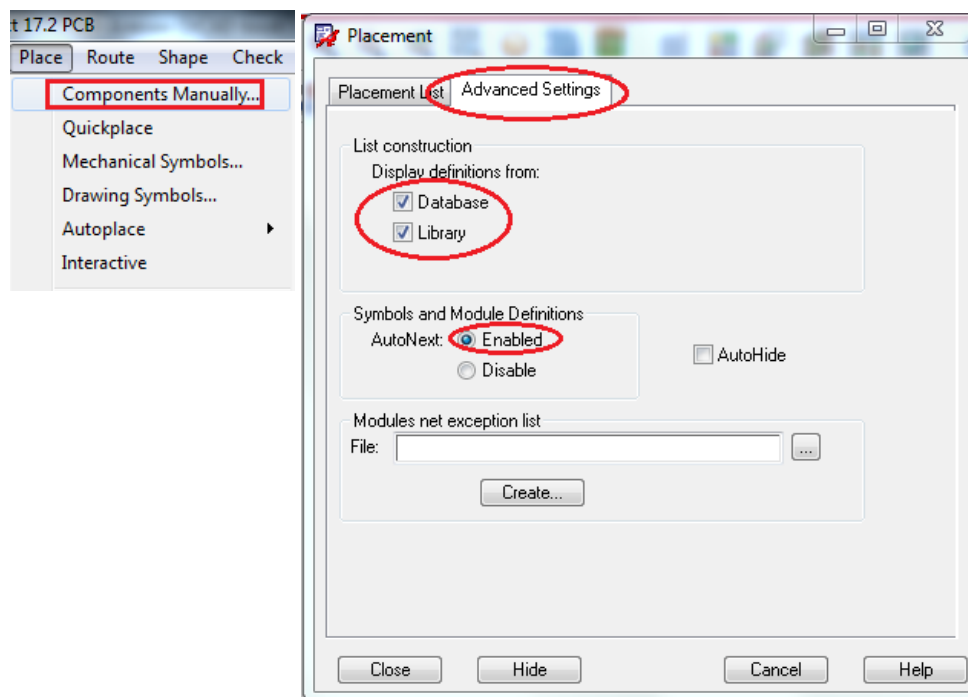


Рис. 9.16. Установка библиотек для компонентов

1. В меню Place выберите Components Manully. Появится окно Placement.
2. Выберите вкладку Advanced Settings.
3. Установите флажок Library и Database под разделом List construction (рис. 9.16).
4. Нажмите кнопку Close, чтобы закрыть диалоговое окно Placement.
5. Снова выберите Components Manully из меню Place.
6. В диалоговом окне Placement выберите вариант Mechanical symbols из выпадающего меню.
7. Выберите нужный механический символ. Для нашего проекта механический символ MTG125 (рис. 9.17).

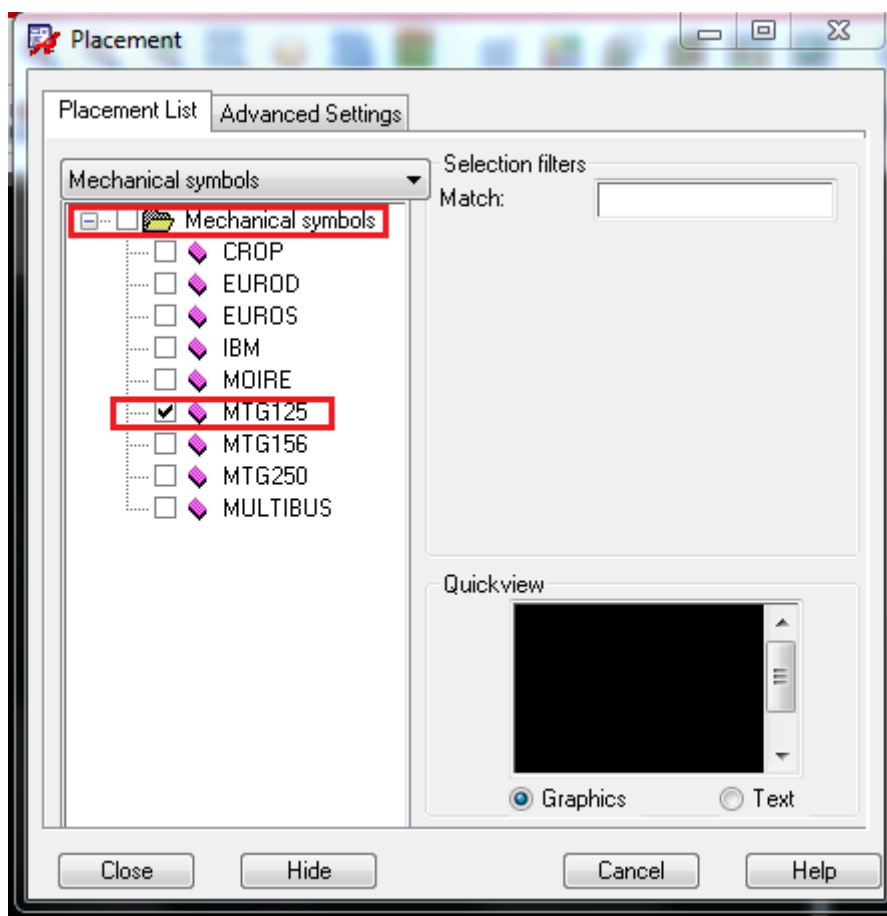


Рис. 9.17. Выбор монтажного отверстия

8. Нажмите кнопку Hide.

9. Диалоговое окно Placement закрывается и механический символ MTG125 прикрепляется к курсору.

10. Переместите механический символ в верхней левый угол окна дизайна платы и щёлкнуть левой кнопкой мыши, чтобы освободить символ.

11. Щёлкните правой кнопкой мыши и выберите пункт Done. Механический символ помещается на плату.

12. Повторите шаги с 5 по 11, чтобы поместить механические символы на остальные три угла окна дизайна. Смотрите рисунок ниже (рис. 9.18).

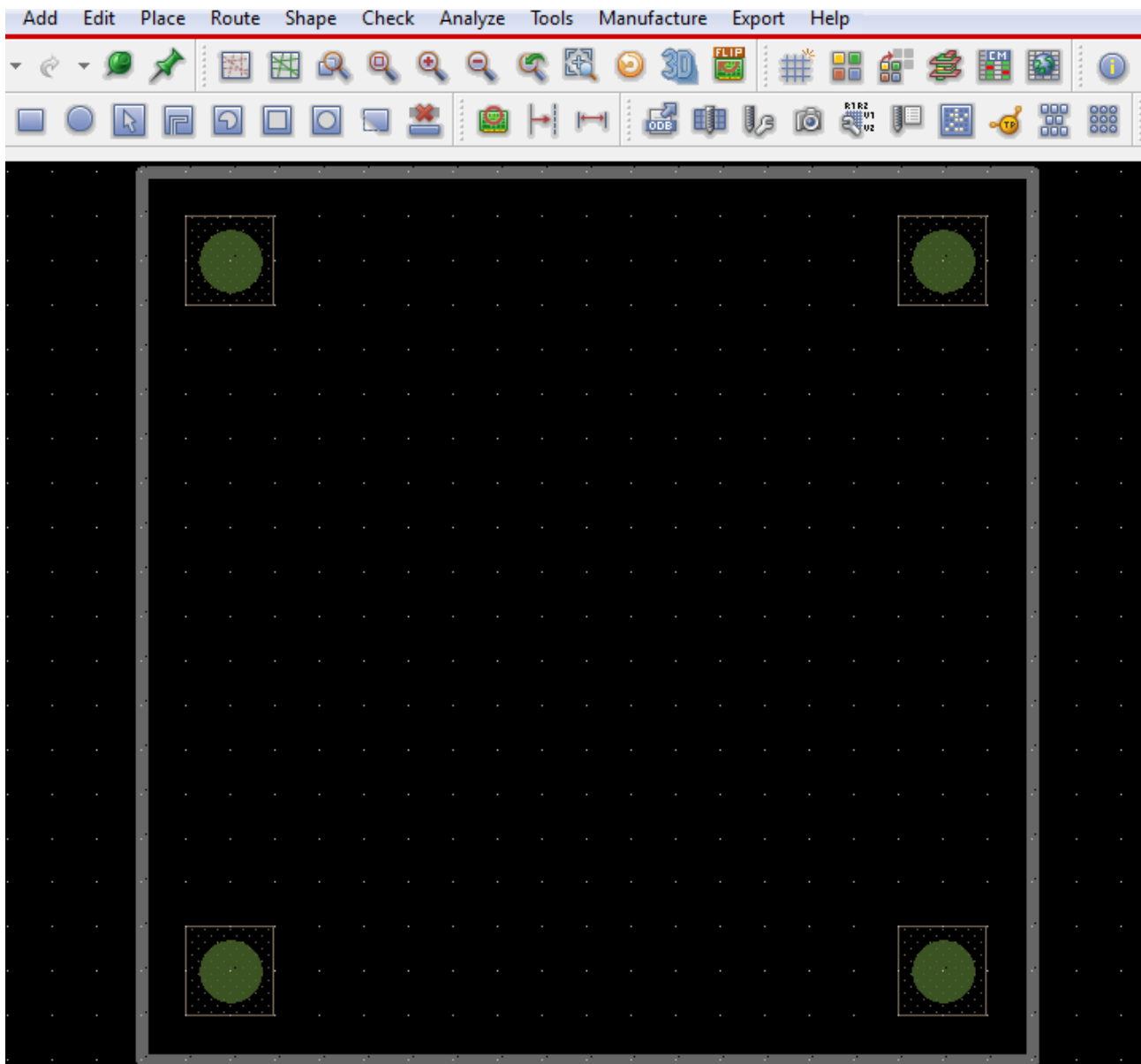


Рис. 9.18. Плата с монтажными отверстиями

Файл наброска образца платы, `fulladd_outline.brd` является доступным по адресу: `/ complete/ Allegro`.

Примечание: В качестве альтернативы, выберите `Copy` из меню `Edit` и, щёлкнув левой кнопкой мыши, поместите механический символ в окне дизайна. Выбранный механический символ прикрепляется к курсору. Передвигая символ в нужное место в окне дизайна, щёлкните левой кнопкой мыши, чтобы

освободить символ. Теперь, щёлкните правой кнопкой мыши и выберите пункт Done.

Для удаления компонента из дизайна надо выделить компонент левой кнопкой мыши, нажать правую кнопку и на вкладке Symbol выбрать Unplace component (рис. 9.19).

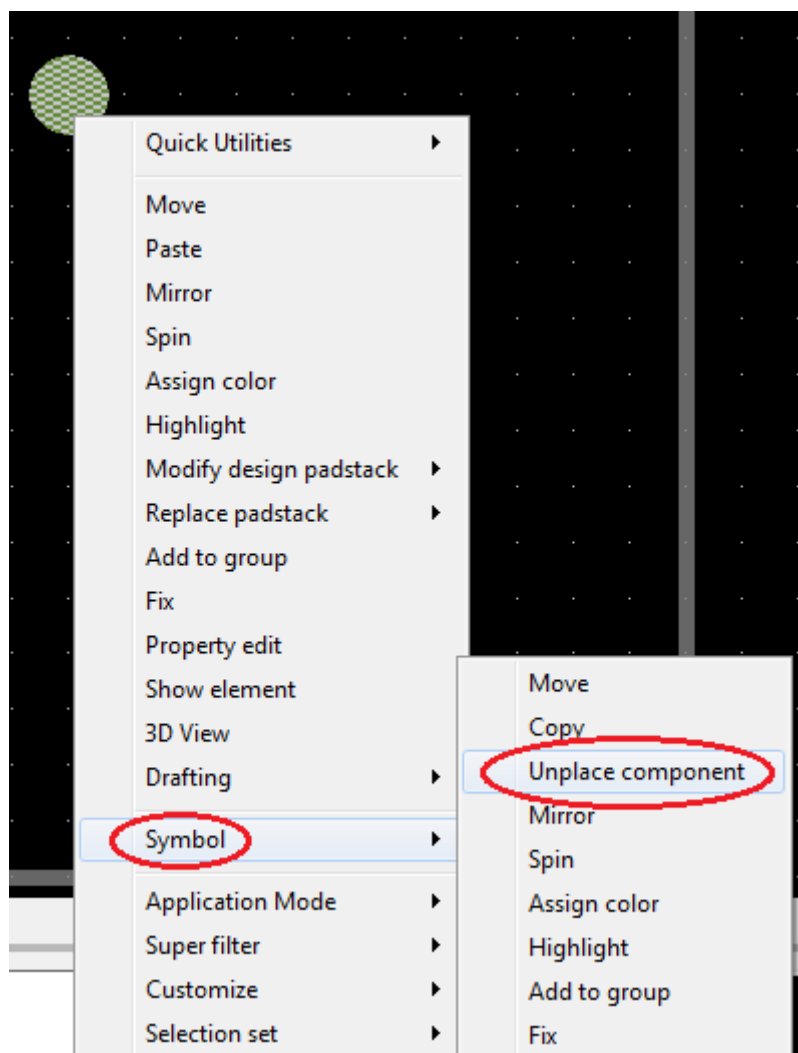


Рис. 9.19. Удаление компонента

9.5. Размещение компонентов

После того, как вы создали контур платы, вы можете начать размещение ваших компонентов на плате. OrCAD PCB Editor поддерживает как размещение вручную, так и автоматическое размещение компонентов.

В этом разделе мы будем использовать размещение вручную, чтобы создать печатную плату для дизайна полного сумматора. Существуют различные способы, которыми вы можете выбрать компонент для размещения. В этом проекте вы научитесь размещать компоненты лишь с помощью RefDes.

9.5.1. Выбор компонентов с помощью RefDes

1. В меню Place выберите Manually. Появится диалоговое окно Placement, отображающее в структурном дереве все компоненты, которые вы можете разместить в вашем дизайне. Например, для нашего случая папка Components by refdes содержит компоненты: J1, Q1, R1, R2, U1, U2, U3 и U4.

Примечание: только неразмещенные компоненты отображаются в окне Placement.

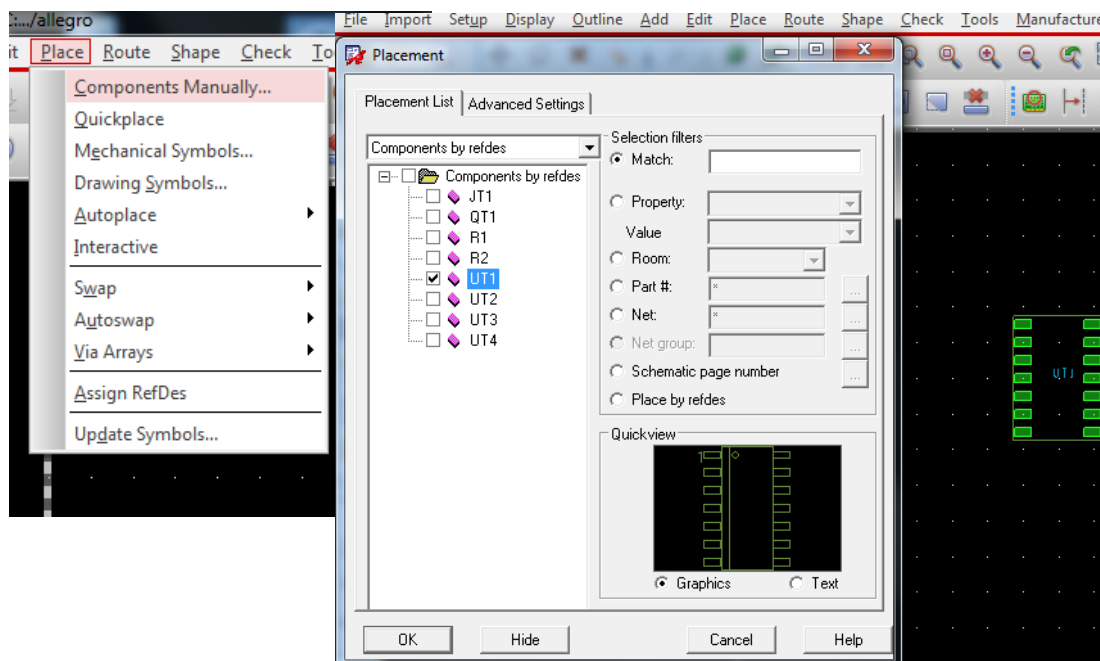


Рис. 9.20. Размещение компонента U1

Примечание: Вы также можете выбрать все компоненты одного типа, поставив галочку рядом со значком папки.

Примечание: Окно Quickview отображает корпус для выбранного компонента в графике и текстовом режиме.

2. Выберите компонент U1, установив флажок, как показано на рис. 9.20.

3. Нажмите Hide. Диалоговое окно Placement закроется, и имя (имена) компонента, в данном случае, U1, который вы выбрали, прикрепится к курсору.

4. Переместите компонент в нужное место, щёлкните правой кнопкой мыши и выберите Done.

Чтобы повернуть компонент, щелкните правой кнопкой и выберите Rotate (Spin) из всплывающего меню.

Примечание: Убедитесь, что угол поворота определяется в панели Options. Для нашего проекта угол поворота 90 (рис. 9.21).

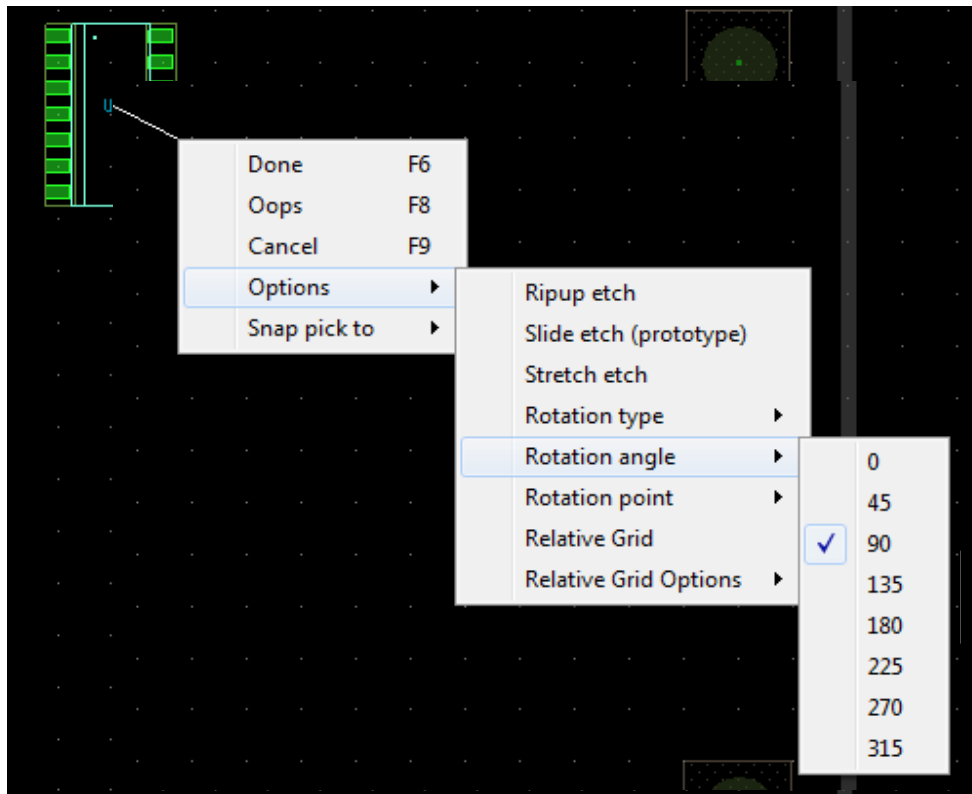


Рис. 9.21. Установка параметров поворота компонента

5. Поверните компонент в направлении против часовой стрелки и щелкнуть левой кнопкой мыши, чтобы освободить компонент (Рис. 9.22).

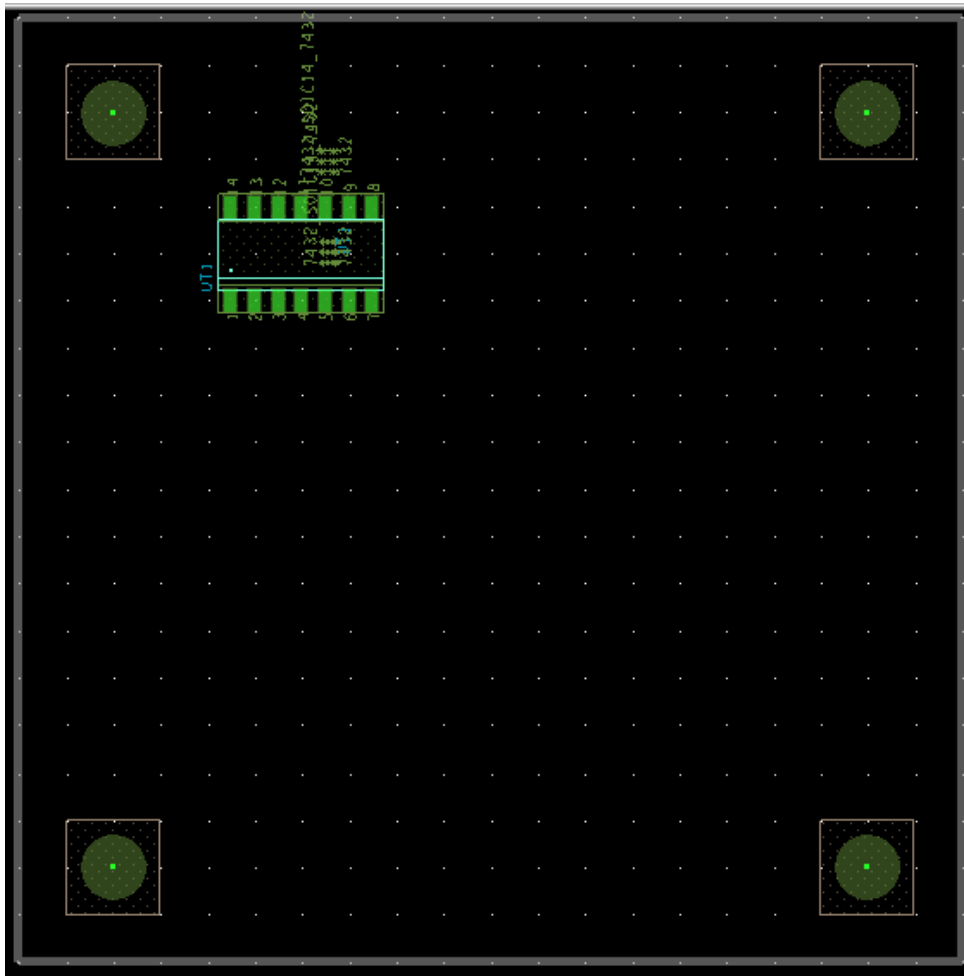


Рис. 9.22. Положение компонента U1 после поворота

6. Повторите шаги 2 - 5, пока все компоненты, доступные в диалоговом окне Placement, не будут размещены в окне дизайна, как показано на рис. 9.23. Обращаем ваше внимание на то, что показанное расположение компонентов на плате будет соответствовать итоговому виду разводки. Рекомендуем вам выполнить такое же расположение с учетом положения первых выводов, обозначенное точками на корпусах и показанное стрелками на рис. 9.23.

Файл образца платы с размещёнными компонентами, fulladd_placed.brd доступен по адресу: /complete/allegro.

7. Выберите Refresh из меню View, чтобы обновить экран (рис. 9.24).

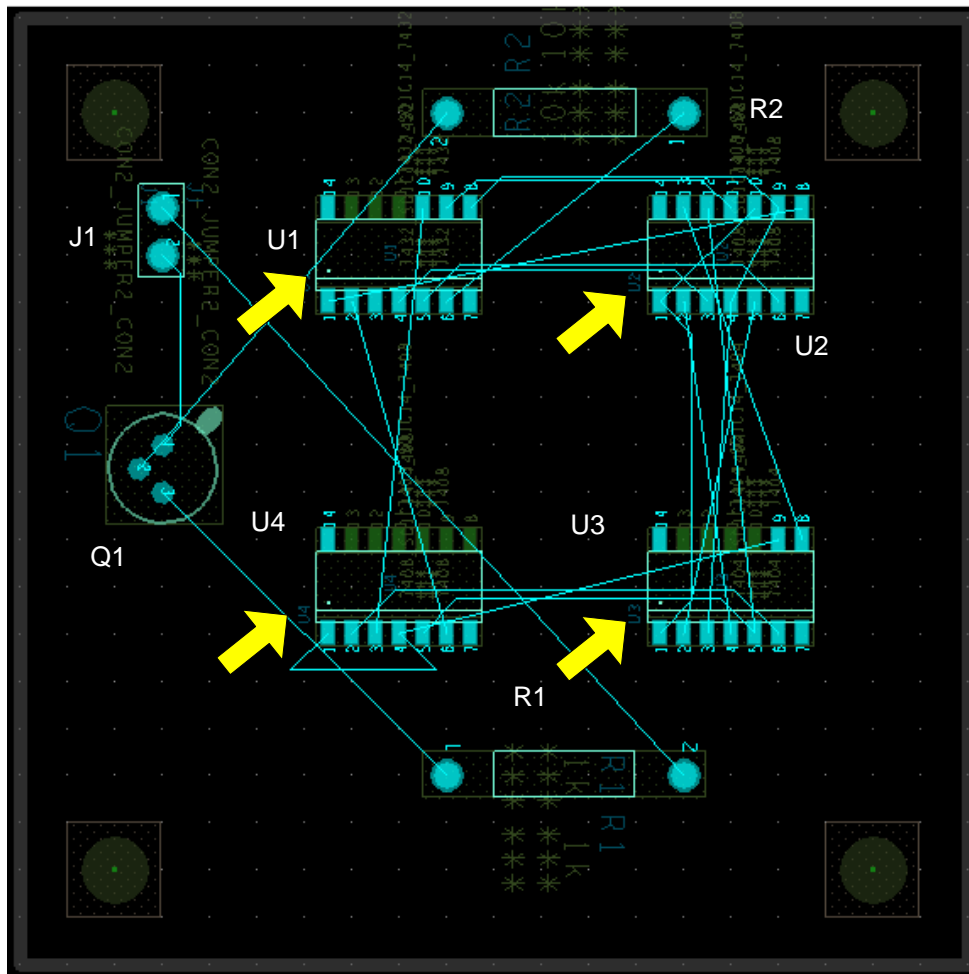
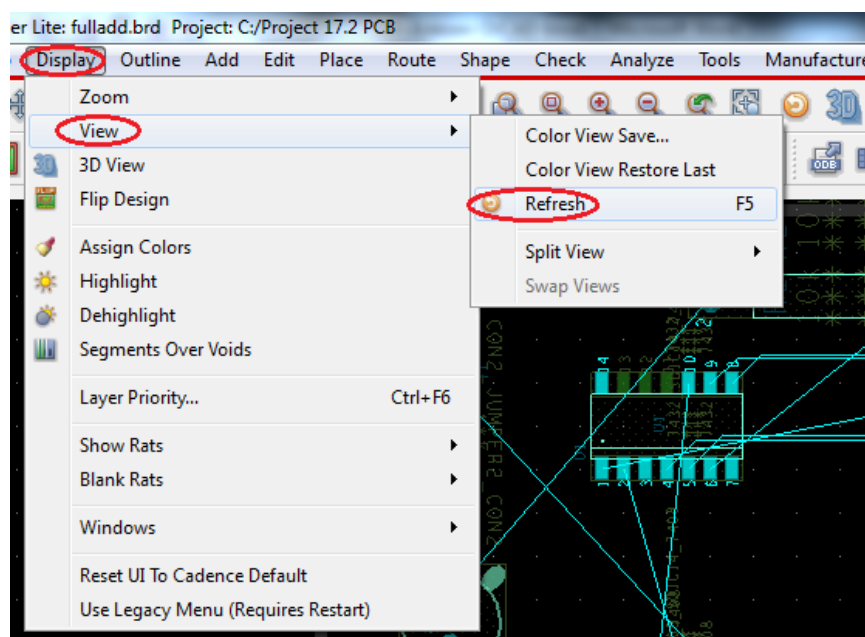


Рис. 9.23. Плата с размещенными компонентами



Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Рис. 9.24. Обновление экрана

9.5.2. Поиск компонентов на плате

Чтобы найти компонент в редакторе печатных плат, выполните следующее:

1. Откройте панель Find (рис. 9.25);

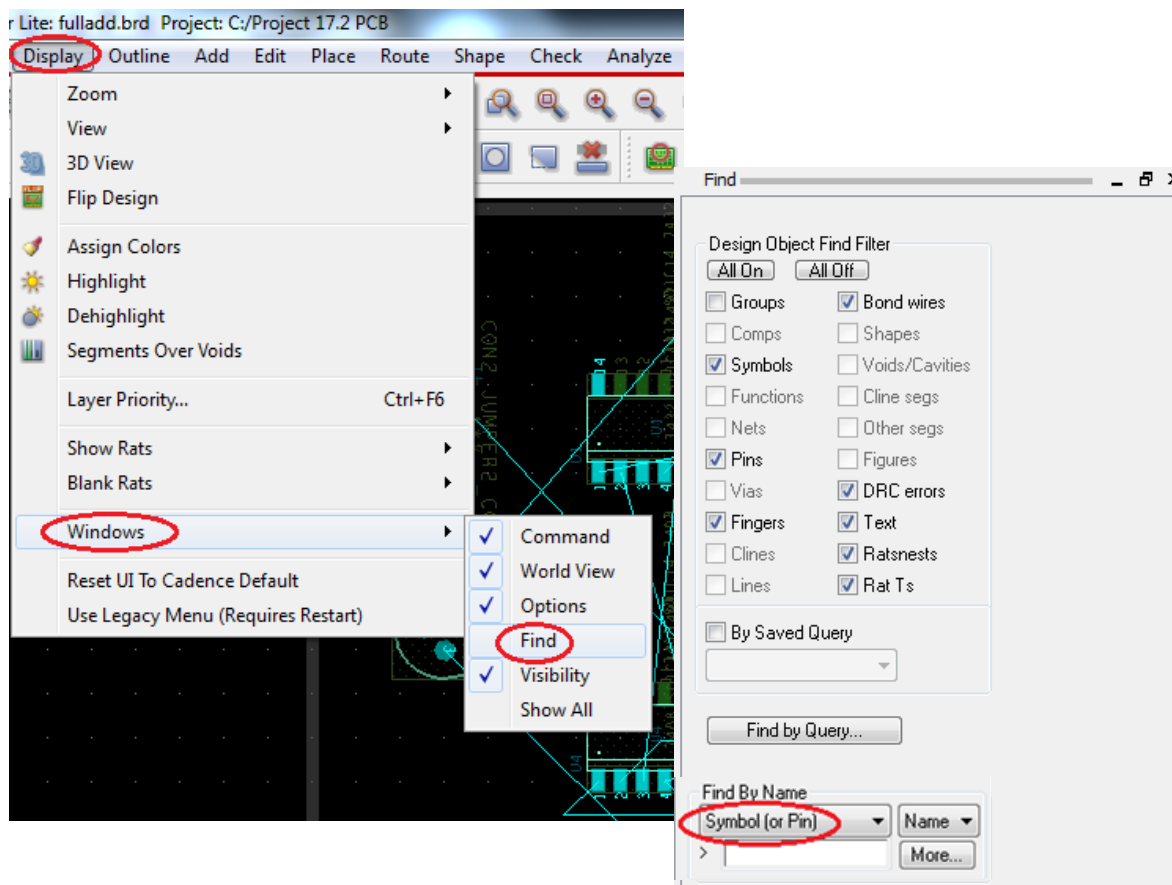


Рис. 9.25. Открытие панели поиска Find

2. В меню Find панели окна редактора печатных плат:

- Выберите опцию Symbol(or Pin) в раскрывающемся списке Find By Name.
- Нажмите кнопку More (дополнительно). Диалог Find by Name or Property появится в раскрывающемся меню, отображающий все доступные компоненты.

- Выберите компонент, который вы хотите найти. Выбранный компонент появится в окне Selected objects.
- Нажмите Ok. Компонент высветится в окне проектирования (рис. 9.26).

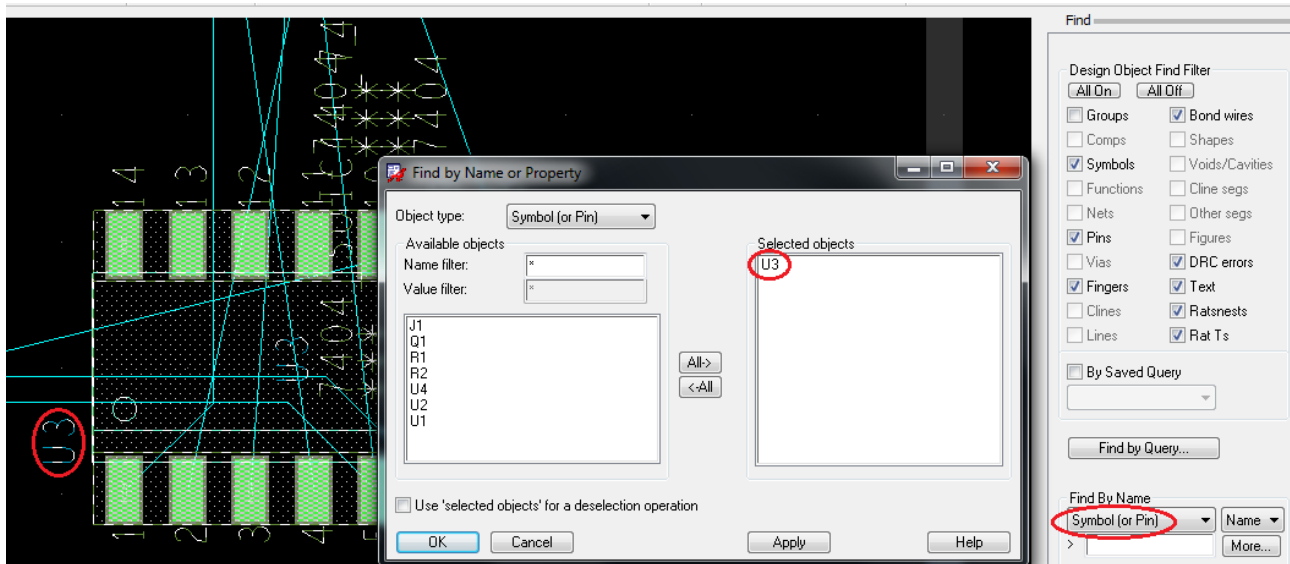


Рис. 9.26. Поиск компонента U3

Вы можете найти цепь в редакторе печатных плат. Чтобы найти цепь, в меню Find нажмите All On в Design Object Find Filter и установите стрелку напротив Net. В окне Find By Name выберите Net и нажмите More. В открывшемся окне Find by Name or Property установите Net. Откроется список цепей и Вы можете найти нужную цепь, например, N00061 (рис. 9.27). На печатной плате эта цепь будет выделена пунктиром.

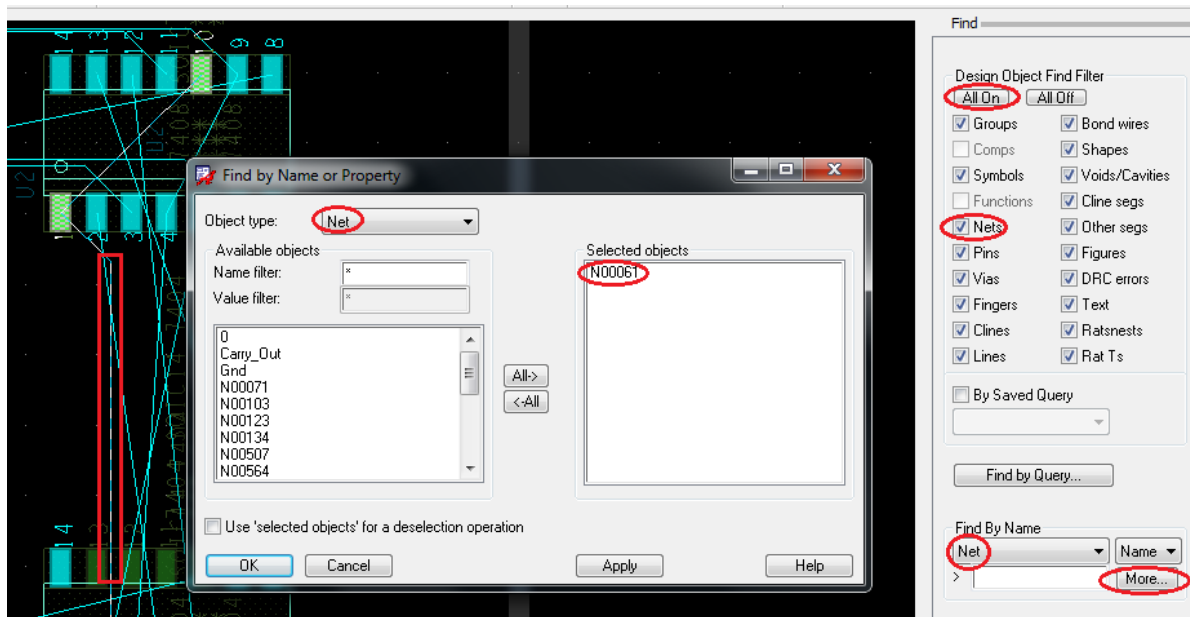


Рис. 9.27. Выделение нужной цепи N00061

Можно выполнить поиск цепи или объекта по запросу. Для этого нажмите Find by Query, выберите Net и нажмите More. В открывшемся окне выберите из объектов Net. В полях выберите Net Name, а затем нужную цепь, например, N00103. Эта цепь отобразится пунктиром (рис. 9.28). Результаты поиска будут сохранены в папке проекта печатной платы.

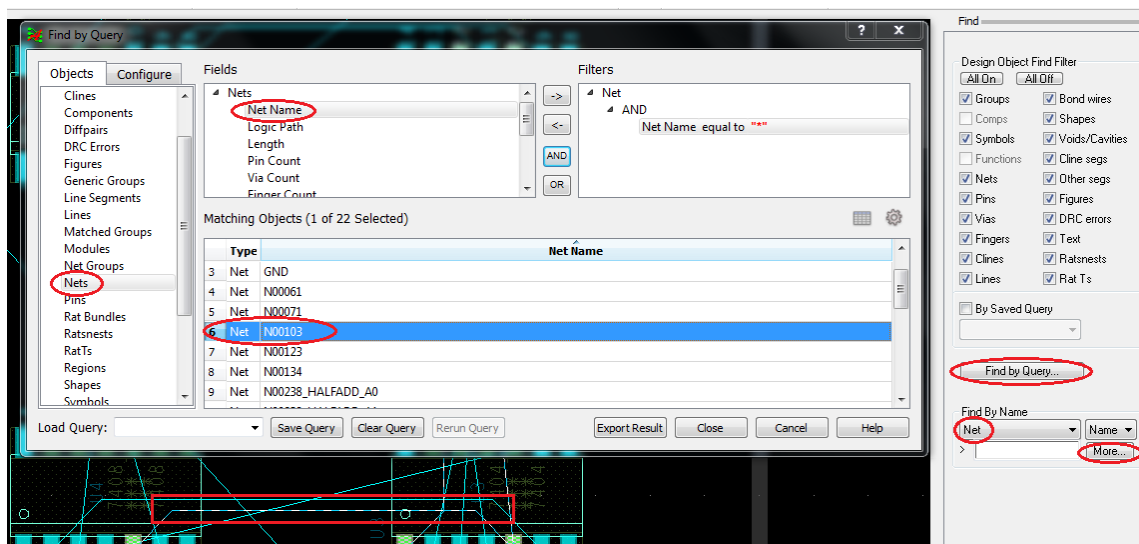


Рис. 9.28. Поиск цепи по запросу

Если потребуется, можно изменить цвет каждой цепи, выполнив Setup>Colors>Nets (рис. 9.29).

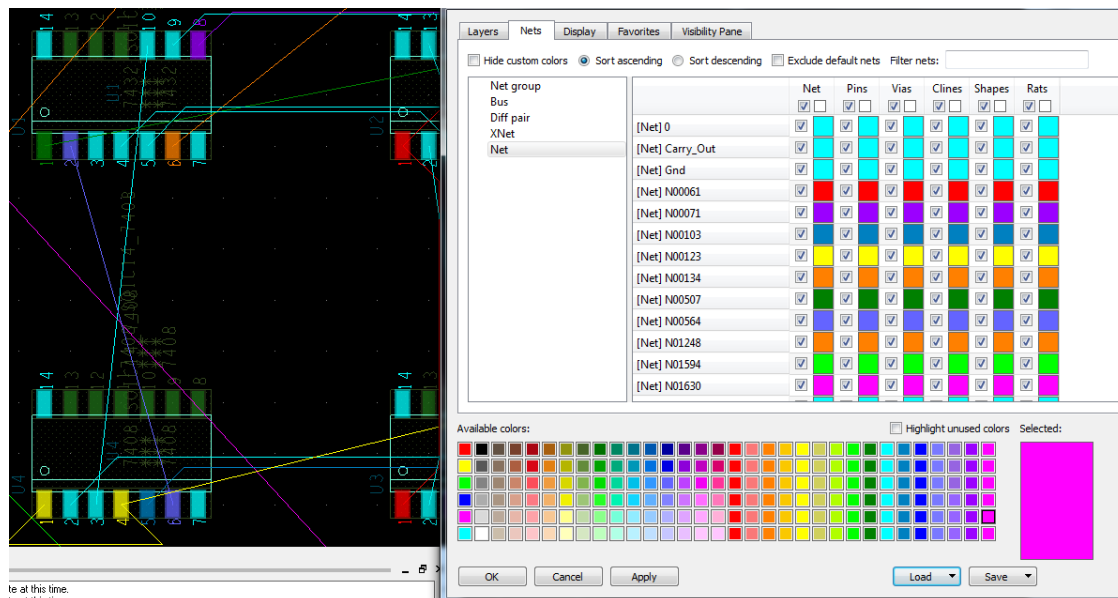


Рис. 9.29. Изменение цветов цепей

9.5.3. Проверка правил разработки

PCB Editor позволяет запускать проверку правил разработки DRC онлайн (On) или в пакетном режиме (Off). По умолчанию включено On. При размещении компонентов, если есть какие-либо нарушения правил проектирования, маркеры ошибок отображаются на плате.

Примечание. Чтобы запустить DRC в режиме онлайн, выберите *Enable On-Line DRC* в меню Setup (рис. 9.30).

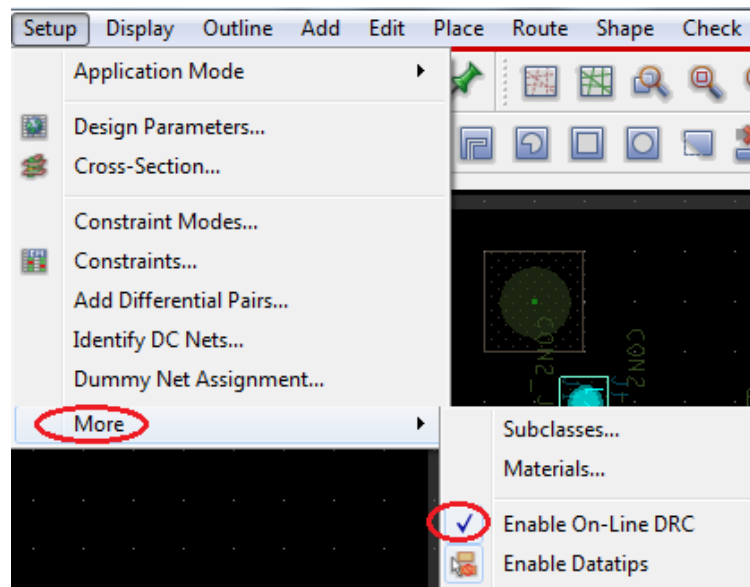


Рис. 9.30. Запуск проверки правил разработки

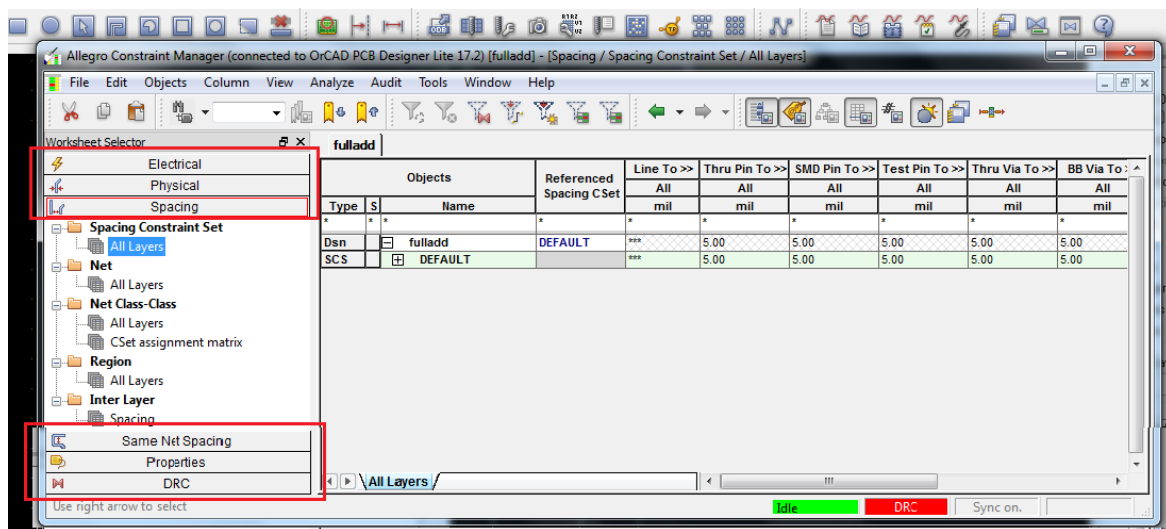


Рис. 9.31. Проверка правил разработки печатной платы

Чтобы проверить основные расстояния и физические ограничения для дизайна вашей платы, выберите *Setup - Constraints - Spacing*. Появится окно с указанием настроек по умолчанию. Для нашего проекта мы примем значения по умолчанию. Allegro Constraint Manager позволяет выполнить разнообразные проверки дизайна-электрические, физические и пр. (рис. 9.31).

9.6. Использование категории DataTip

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Конструкции становятся более плотными и различить определённый элемент в плотной конструкции может быть сложно. Чтобы помочь вам выбрать правильный элемент, наведя курсор, над выделенным элементом появляется контекстно-зависимая информация DataTip, которая идентифицирует элемент. DataTip будет появляться над панелью команд консоли окна, если установить переменную `datatips_fixedpos` в меню **Setup>User Preferences** (рис. 9.32).

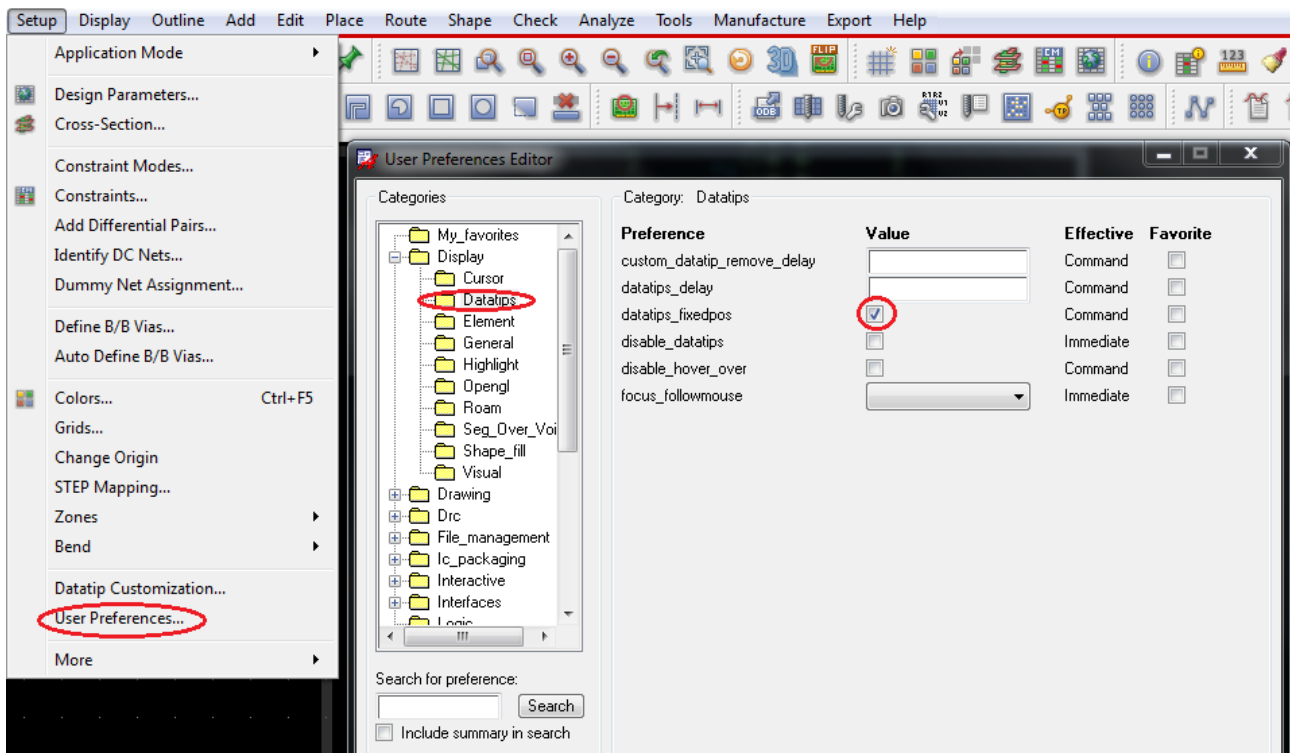


Рис. 9.32. Установка переменной `datatips_fixedpos`

Можно выводить дополнительные параметры, установив **Setup>DataTips Customization** (рис. 9.33).

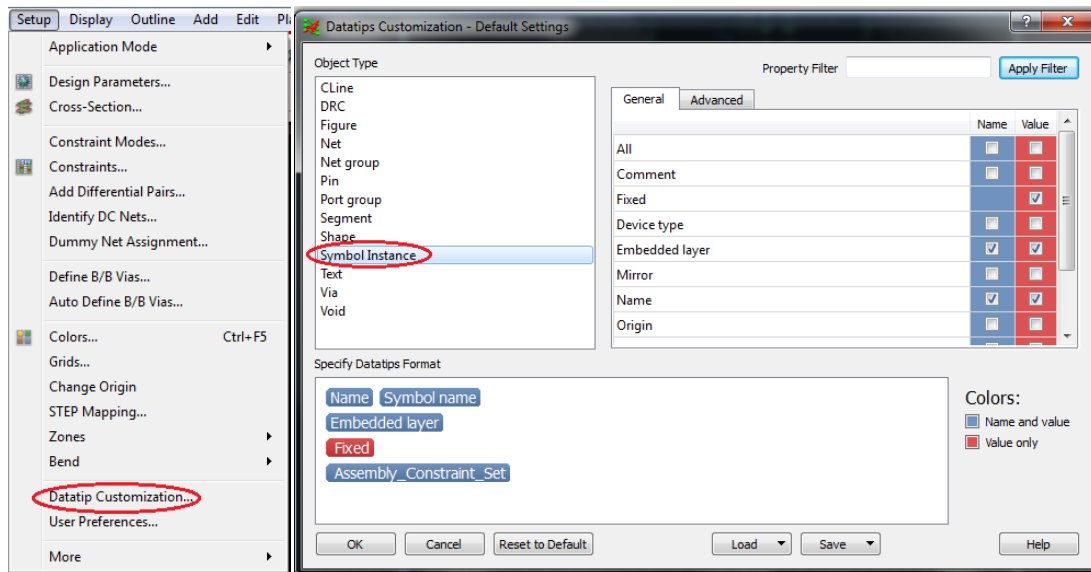


Рис. 9.33. Дополнительные параметры образца символа

9.7. Выделение области

Этот инструмент позволяет выделять элементы конструкции одним из выбранных инструментов набора выделения. Команды, которые работают при этом выборе набора, затем появляются на правой кнопке мыши всплывающего окна меню (рис.9.34). Чтобы снять выбор, нажмите Clear all Selections. При нажатии правой кнопкой Selection Set появляются варианты выделения области. Вы можете выбрать подходящий (Polygon) и он станет постоянным (Persistent) до сброса.

Для переключения режима надо выполнить Persistent Off.

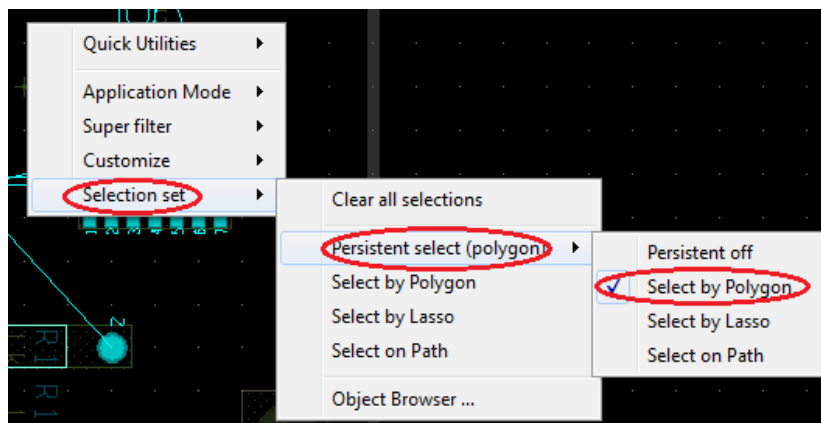


Рис. 9.34. Выбор области выделения

На рис. 9.35 показано выделение области, используя Polygon.

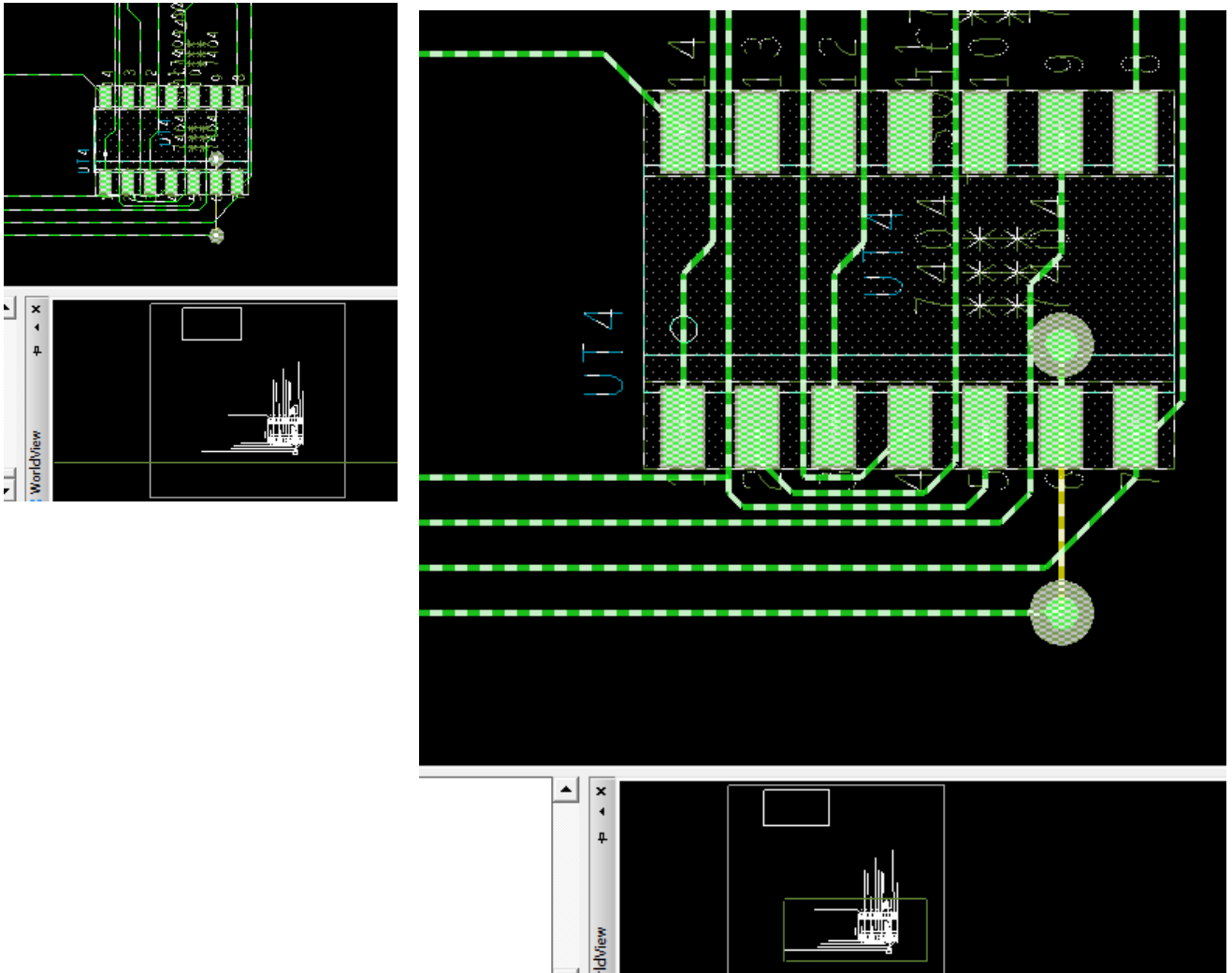


Рис. 9.35. Выделение области, используя Polygon

9.8. Выбор элементов дизайна с помощью Superfilter

Superfilter позволяет выбрать конкретный тип элемента для более точного выбора и временно отключить все остальные элементы. Для этого правой кнопкой мыши надо открыть всплывающее меню и отметить нужный для поиска тип объекта. На рис. 9.36 выбираются будут только цепи, а на рис. 9.37 будут

выбираться символы из всплывающего меню правой кнопкой мыши, а не через окно Find.

По умолчанию для Superfilter установлено значение Off (Выкл.). Это означает, что выбираются все объекты в конструкции (нефильтрованный выбор).

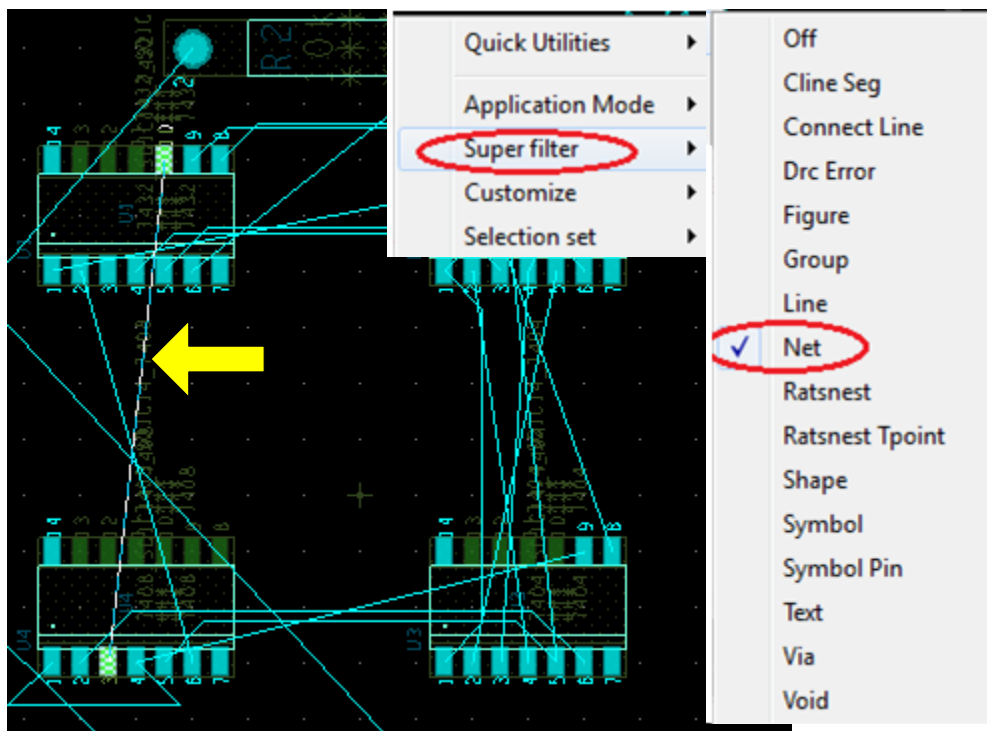


Рис. 9.36. Выбор только цепей суперфильтром

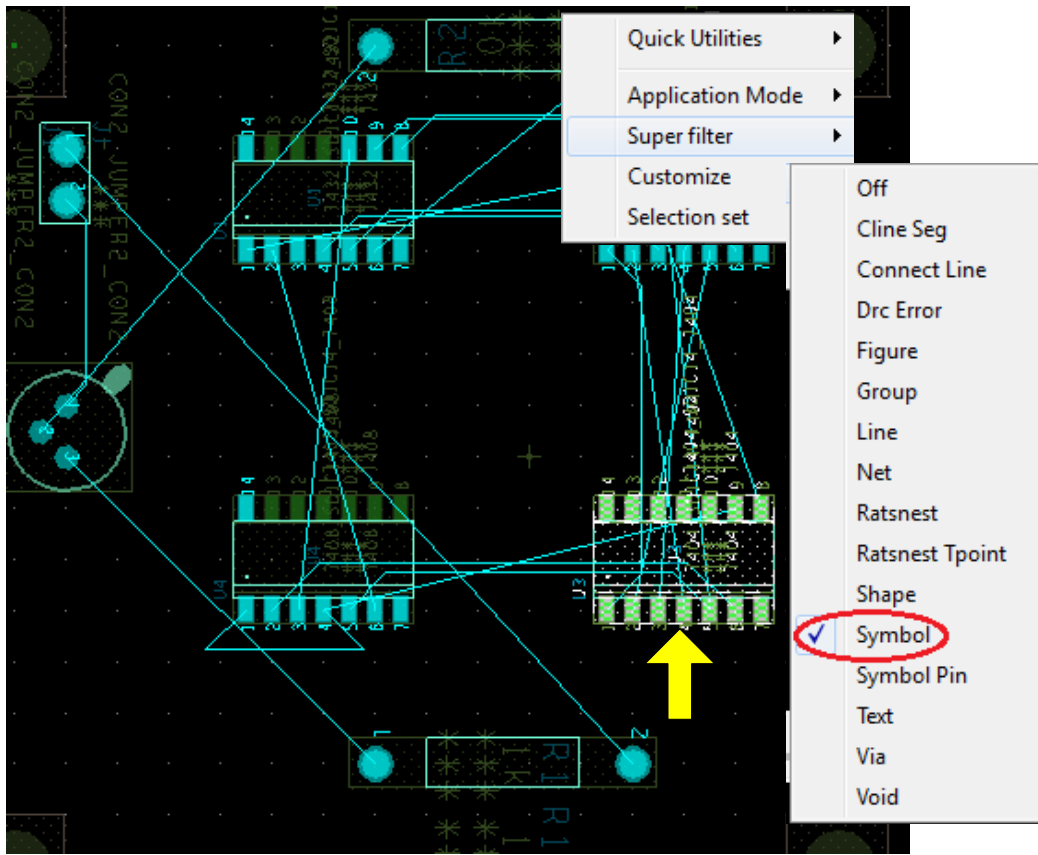


Рис. 9.37. Выбор только символов суперфильтром

9.9. Общие параметры на всплывающих меню

Во всплывающем меню после нажатия правой кнопки мыши можно выполнять дополнительные функции, а также будут доступны различные варианты.

Quick Utilities позволяет получить доступ к часто используемым функциям, таким как Undo, Design Parameters, Grids, Change active subclass (рис. 9.38).

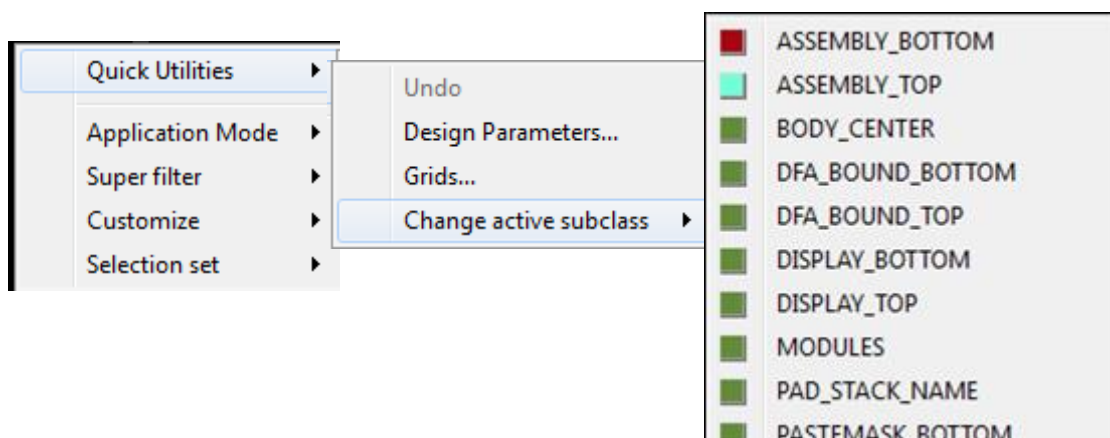


Рис. 9.38. Всплывающее меню из Quick Utilities

Щелчок правой кнопкой мыши на элементе конструкции открывает более подробное содержание всплывающих меню (рис. 9.39). Рекомендуем ознакомиться с опциями этих меню.

9.10. Описание режима привязки

Если команда выполняется на подмножестве набора компонентов или на иерархических блоках, то соответствующие элементы надо добавить к набору, а остальные части проекта игнорируются.

Для этого выполняют привязку компонентов.

Режим привязки доступен, когда даётся интерактивная команда редактирования, например, *Move* или *Copy*. При этом правой кнопкой открывается меню, содержащее команды *Snap pick to* (Привязать к ..), *Persistent Snap* (Shape Center) и прочие (рис. 9.40). Можно нарисовать дорожку проводника близко к контакту, щелкнуть правой кнопкой мыши и выбрать *Snap pick to>Pin* (привязать к пину). Это удобно для других целей, таких как сквозные отверстия (Via).

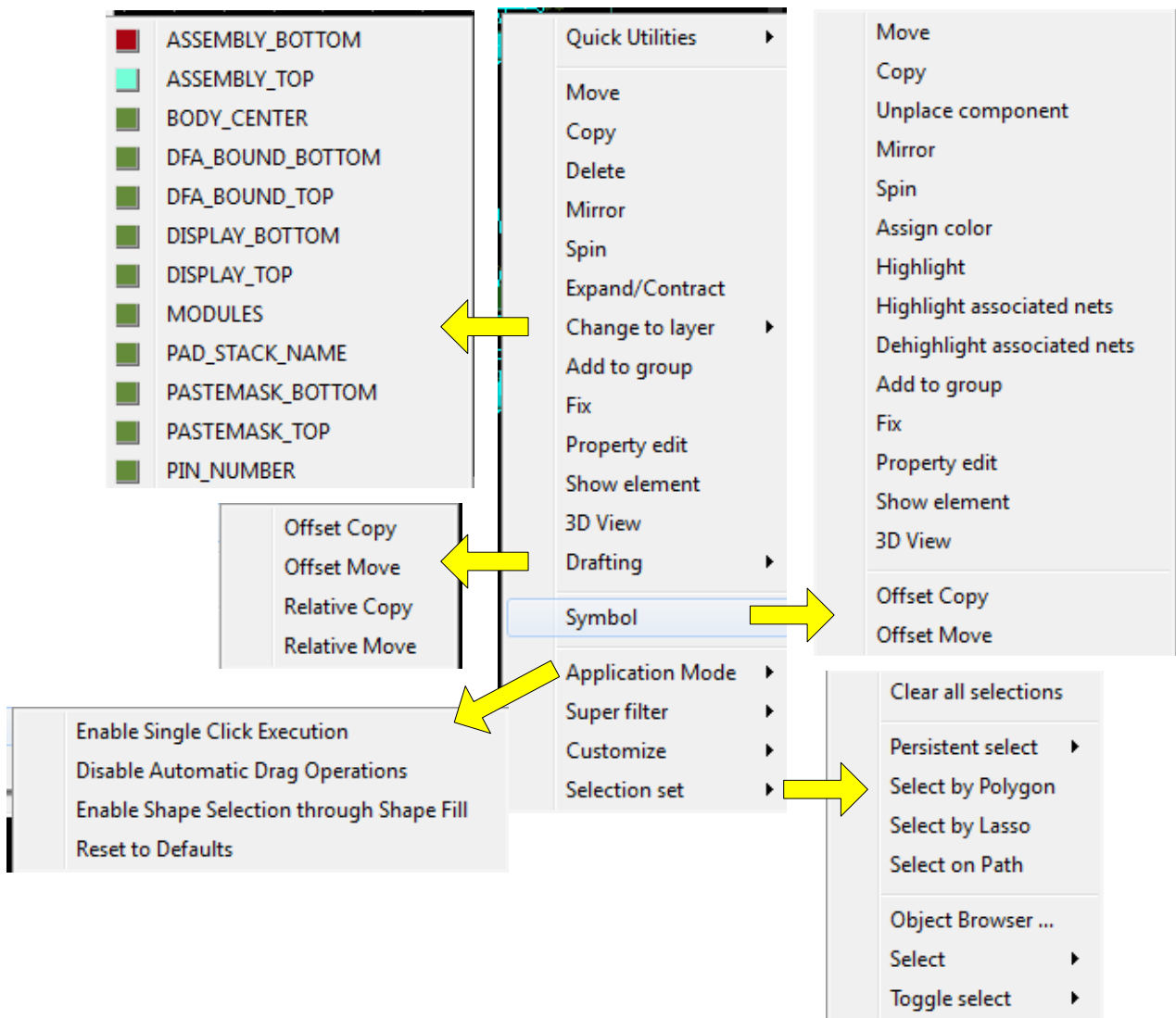


Рис. 9.39. Всплывающее меню после щелчка на элементе конструкции

По желанию можно контролировать как отправную точку выбора, так и точку назначения. Привязка к точке достигается от текущего положение мыши к выбранному режиму привязки и зависит от режима привязки и типа объекта привязки. Если объекты не доступны, привязка будет неудачна. Появится сообщение в окне командной строки, указывающее, что привязка оказалась не-

удачной. Подробнее об этих режимах следует читать в полном руководстве по OrCAD 17.2 от компании Cadence.

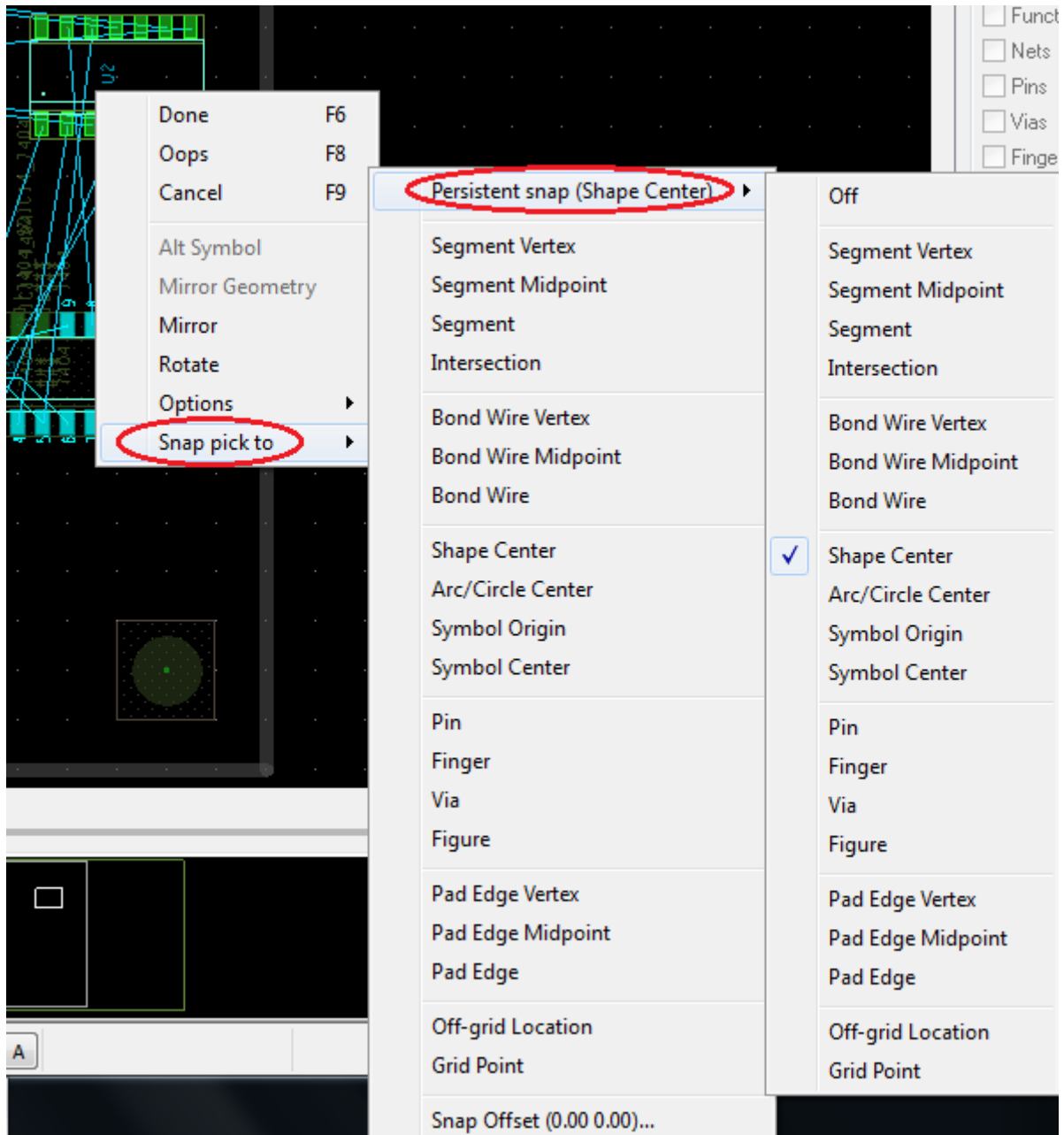


Рис. 9.40. Установки режима привязки

9.11. Использование панели окна WorldView

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Есть три способа, которыми вы можете контролировать вид конструкции с помощью окна WorldView:

- Для отображения конкретных областей дизайна
- Для прокрутки дизайна
- Для увеличения или уменьшения масштаба конструкции

В окне просмотра (позиционирования) открывается дополнительное меню (рис. 9.41).

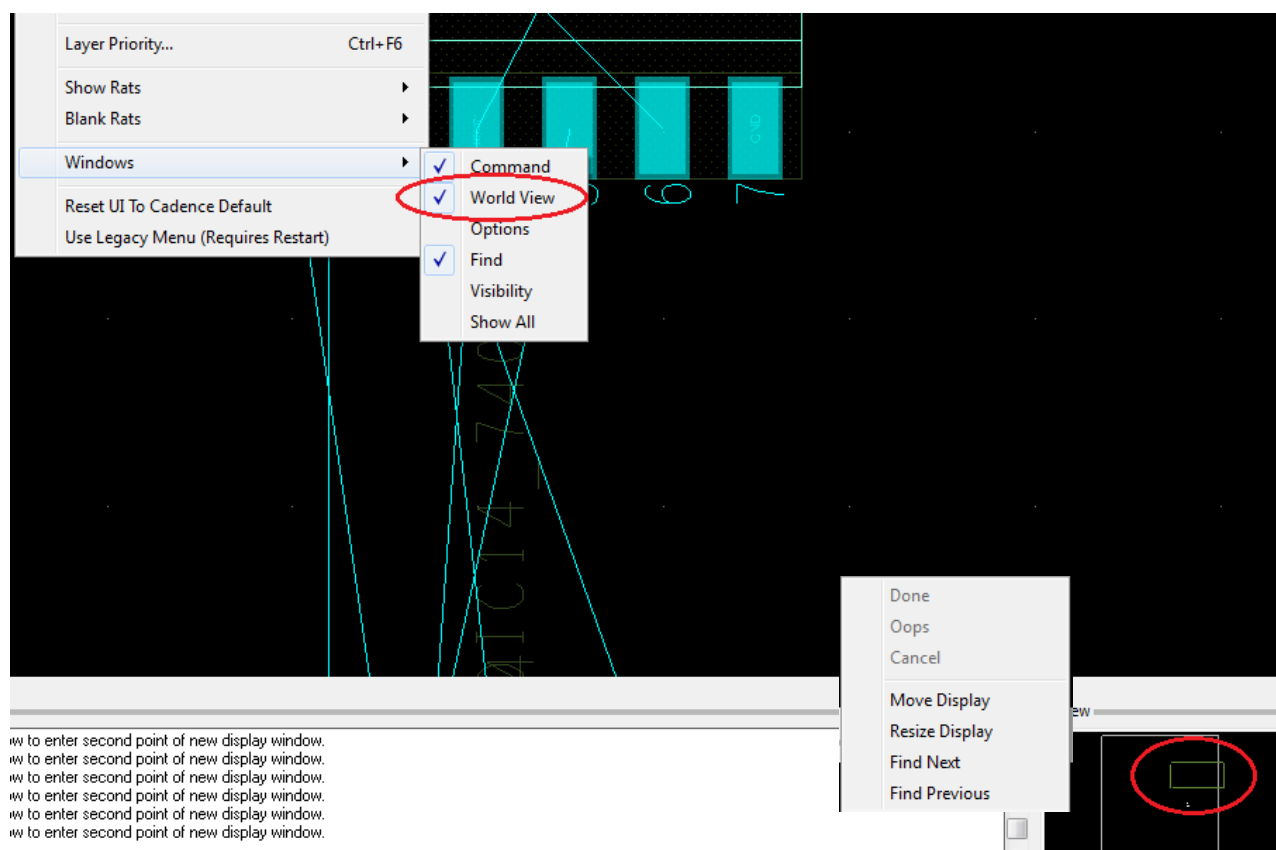


Рис. 9.41. Работа с окном WorldView

Для отображения платы в полном размере надо выделить границу и нажать кнопку Zoom Fit или в командной строке набрать zoom fit (рис. 9.42).

9.12. Маршрутизация

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

После завершения размещения компонентов на плате, вы можете сделать маршрутизацию (трассировку) платы полного сумматора для выполнения электрических соединений между компонентами. OrCAD PCB Editor поддерживает как ручную маршрутизацию так и автомаршрутизацию. Обычно наиболее важные цепи сначала разводят вручную, закрепляют их, а затем выполняют автоматическую маршрутизацию (разводку) остальной части платы.

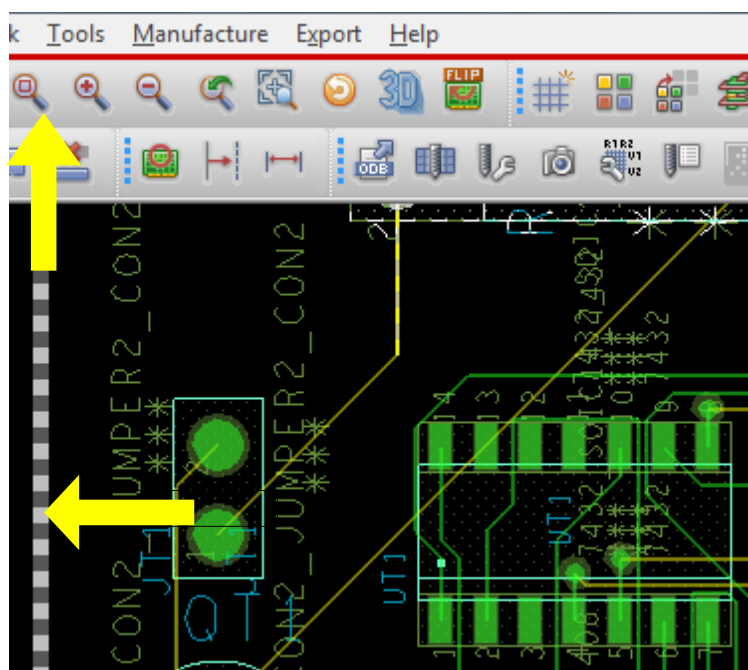


Рис. 9.42. Возврат к исходному масштабу платы

9.12.1. Руководство по маршрутизации

Шаги, используемые в процесс ручной маршрутизации, следующие:

- Проверьте контур платы с помощью определений, маршрутизации и с помощью сеток.
- Выполните разводку питания и земли.
- Просмотрите поверхностный монтаж устройств и проверьте подключение к сети питания и заземлению.
- Разведите остальные сигналы с помощью ручной маршрутизации.
- Выполните оптимизацию маршрутизации с использованием ручных команд.

- Проверьте наличие нарушений промежутков трасс и проверьте статистику маршрутизации.

Примечание: Чтобы узнать больше о каждом из этих этапов, см. документацию PCB Editor.

9.12.2. Ручная маршрутизация цепей VCC и GND.

Перед тем, как начать маршрутизацию цепей VCC и GND, убедитесь, что вы удалили свойство NO_RAT, прикрепленное к этим цепям. Чтобы удалить это свойство:

1. Выберите Object Properties в меню Edit или нажмите клавишу F12.
2. В панели Find выберите опцию Net (если она ещё не выбрана) из раскрывающегося списка Find By Name.
3. Введите VCC и нажмите кнопку More (рис. 9.43)

На плате высветятся проводники VCC. Выделите проводник и выберите Properties Edit.

Появится диалоговое окно Edit Property, отображающее все свойства, относящиеся к цепи VCC.

4. Выберите свойство NO_RAT в списке Available Properties. Установленные свойства отображаются в панели на правой части диалогового окна (рис. 9.43).

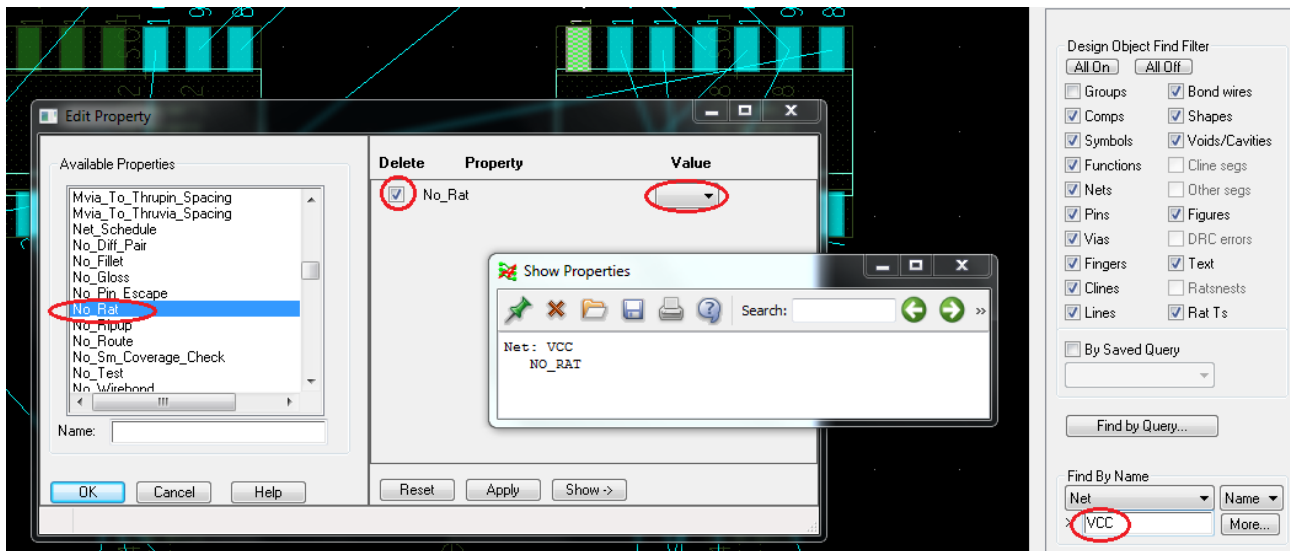


Рис. 9.43. Установка свойства No_Rat для цепей VCC

Дополнительные сведения о свойствах редактора печатных плат см. в документации PCB Editor.

5. Установите флажок Delete рядом с именем свойства No_Rat.
6. Сделайте пустым меню Value.
7. Нажмите кнопку Apply.
8. Нажмите кнопку ОК, чтобы закрыть диалоговое окно Edit Property.

Аналогичные действия выполните для цепей GND (Рис. 9.44).

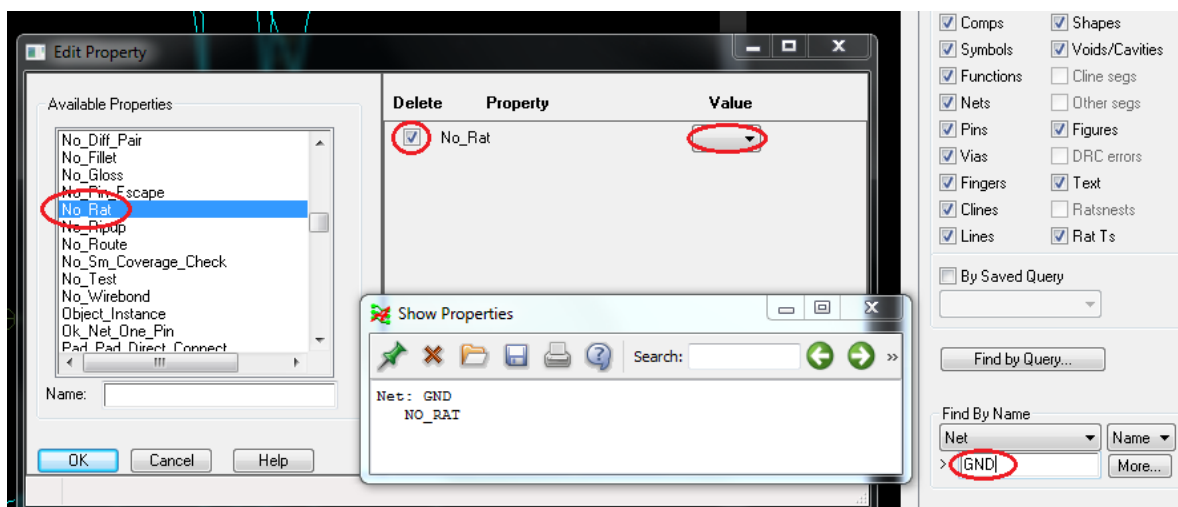


Рис. 9.44. Установка свойства No_Rat для цепей GND

Примечание: Вы можете использовать диалоговое окно Edit Property для добавления или удаления свойств компонента или цепи.

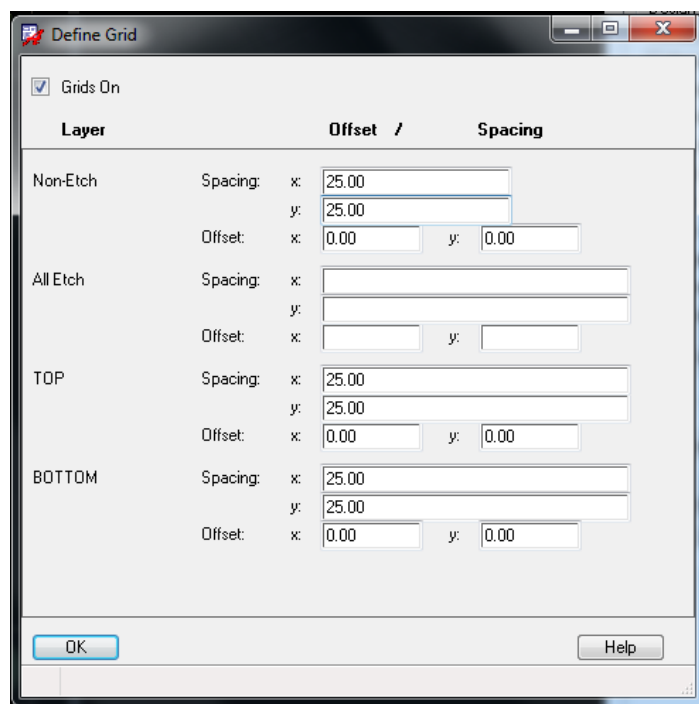


Рис. 9.45. Установка шага сетки

Перед ручной разводкой убедитесь, что шаг сетки соответствует или меньше шага выводов микросхем.

Установим шаг сетки 25.0. Для этого выполним Setup>Grids (рис. 9.45).

Чтобы вручную развести VCC и GND цепи:

1. Выберите вкладку Find в правой части окна редактора печатных плат. Появится панель Find.
2. Выберите опцию Net из списка Find By Name .
3. Нажмите кнопку More. Появится окно Find by Name или Property, отображающие все доступные цепи.
4. Выберите VCC. Элемент цепи VCC отображается в сетке Selected objects (рис. 9.46).

5. Нажмите кнопку ОК. Все сети VCC будут выделены в окне проекта.

6. Выберите Connect из меню Route.

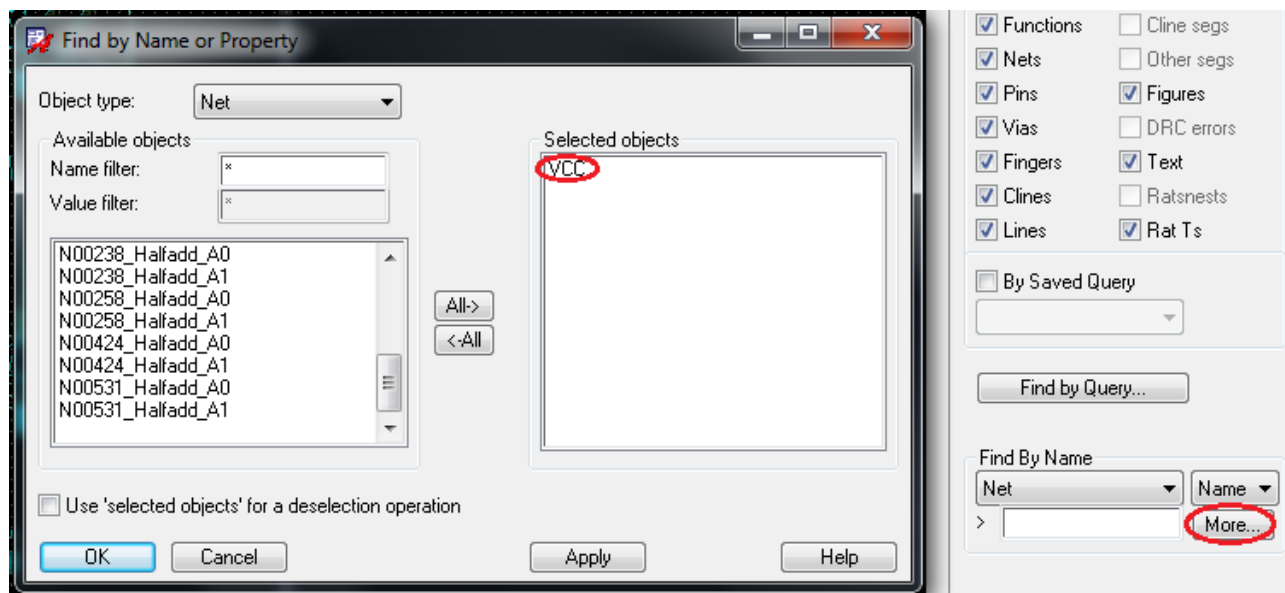


Рис. 9.46. Выделение цепей VCC

Примечание: В качестве альтернативы, вы можете щёлкните значок Add

Connect .

7. Измените ширину линии до 20.00 в панели Options.

8. Теперь нажмите на цепь, которую надо проложить.

9. Нарисуйте цепь по нужному пути.

10. После завершения разводки, щёлкните правой кнопкой мыши на цепи и выберите Done.

Аналогичным образом, выполните описанные выше действия для ручной маршрутизации цепи GND.

Плата с разведенными цепями VCC и GND показана на рис. 9.45.

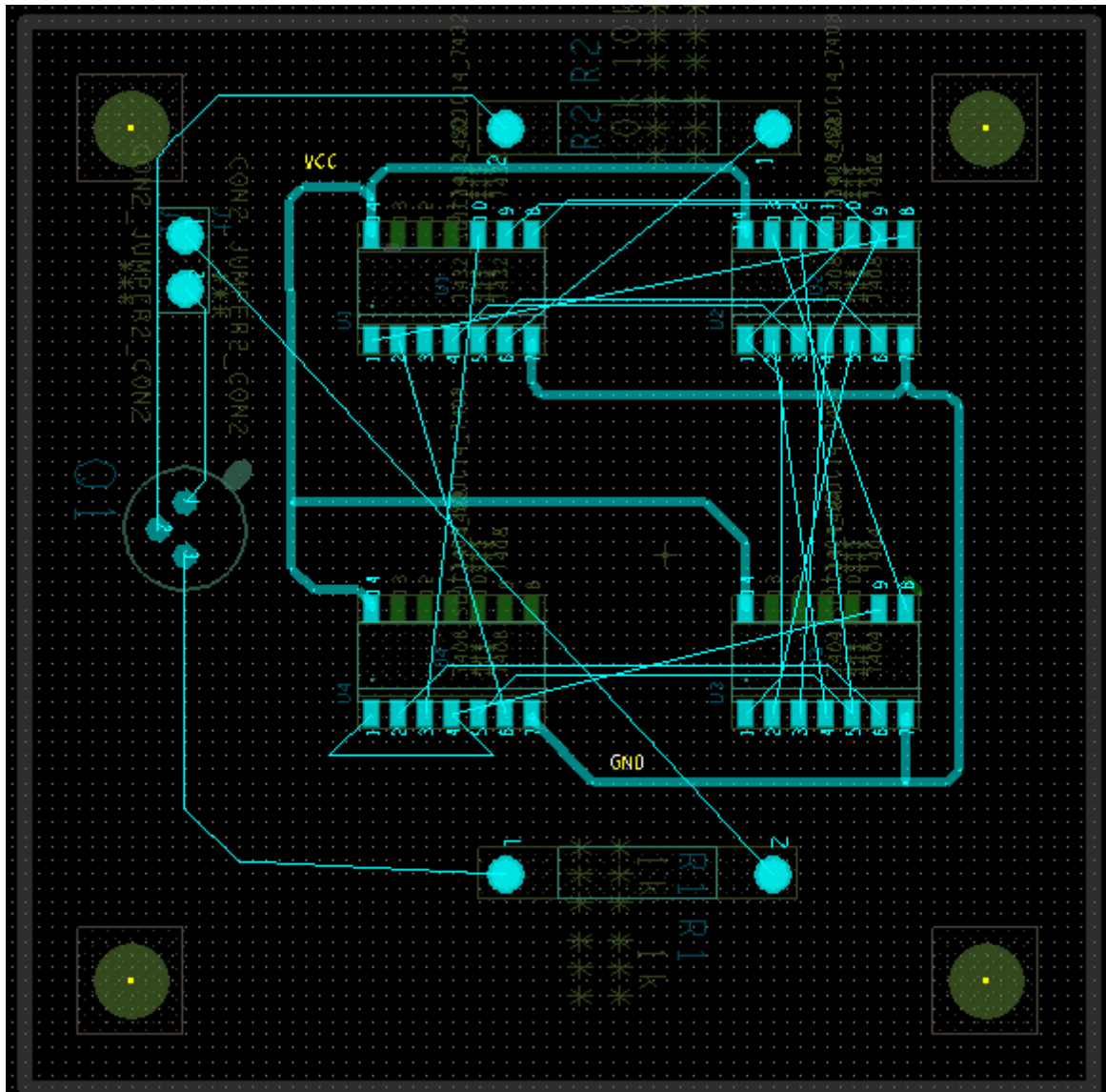


Рис. 9.45. Плата после разводки цепей VCC и GND

9.12.3. Маршрутизация остальных сетей вручную

Чтобы вручную трассировать остальные сети, выполните следующее:

1. Введите команду zoom fit, чтобы ваша плата соответствовала окну дизайна.
2. Поместите курсор на цепь, которую надо проложить и увеличьте масштаб.

3. Выберите Connect из меню Route или выделите цепь, нажмите правую кнопку мыши и выберите Add connect (рис. 9.46).

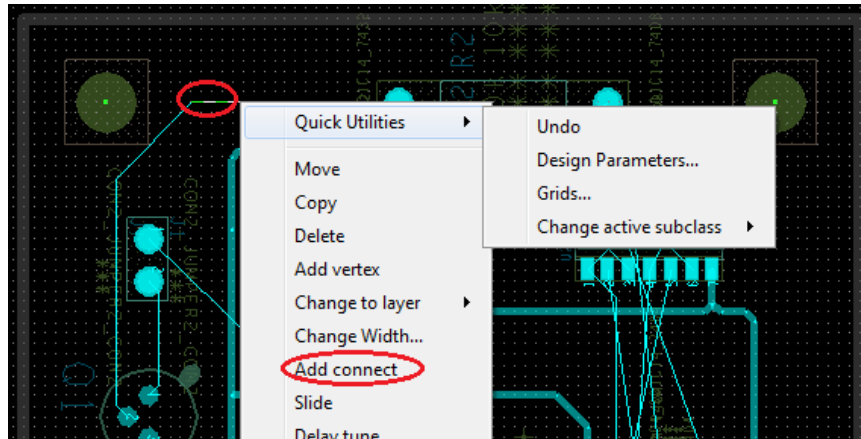


Рис. 9.46. Выбор цепи для разводки

4. Нажмите на цепь, которую разводите. Панель Options изменится, как показано на рис. 9.47.

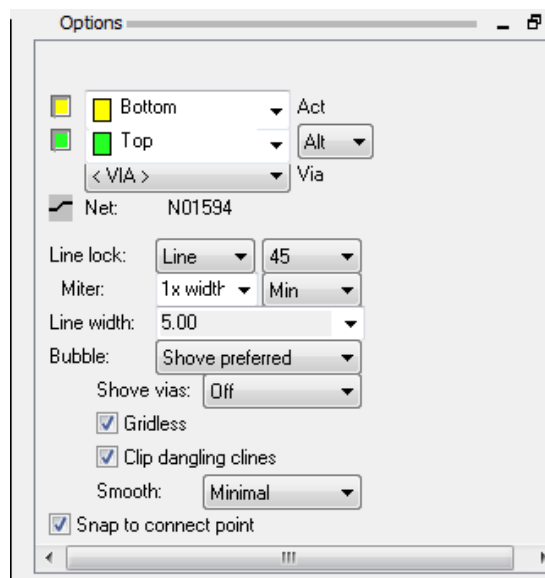


Рис. 9.47. Панель Options для разводки остальных цепей

Примечание: Убедитесь, что настройка Line Lock является Line, 45. Ширина линии 5.00.

5. Нарисуйте цепь по нужному пути.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

6. После завершения маршрутизации, щёлкните правой кнопкой мыши на цепи и выберите Done.

Для изменения слоев во время маршрутизации выполняют добавление переходных отверстий Add Vias:

1. Нажмите на цепь, которую проводите.

2. Щёлкните правой кнопкой мыши на цепь и выберите Add Via. Добавится переходное отверстие Via. Текущий активный слой становится Alternate layer и Alternate layer становится Active layer и наоборот. Например, если у вас есть верхний и нижний слой, где текущий активный слой верхний, то затем, когда вы добавляете Via, нижний слой станет активным слоем и верхний слой становится Alternate layer (рис. 9.48).

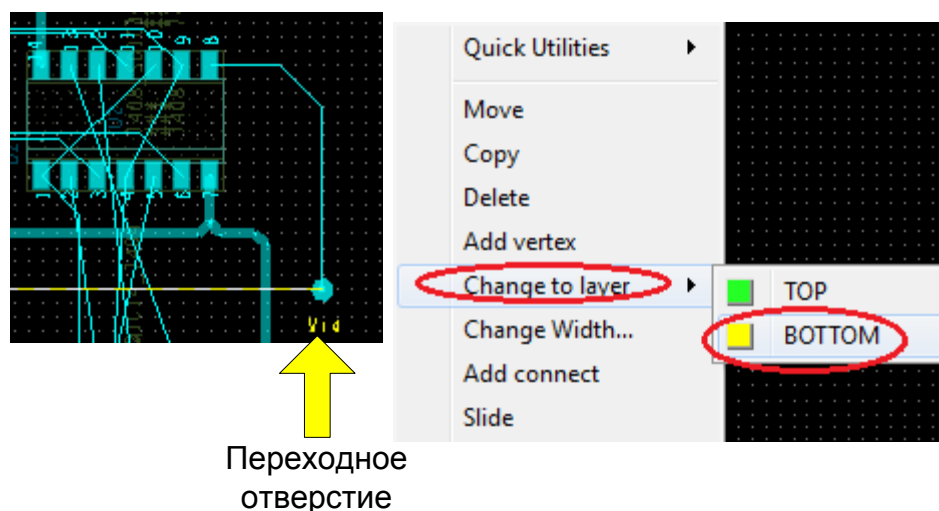


Рис. 9.48. Добавление переходного отверстия и изменение слоя проводника

3. Нарисуйте цепь по нужному пути.

4. После завершения маршрутизации, щёлкните правой кнопкой мыши на цепи и выберите Done.

В целом, получение навыков ручной маршрутизации потребует от вас терпения и определенных усилий. Если вы справитесь с трудностями, то получите эталонный образец разведенной платы, показанный на рис. 9.49.

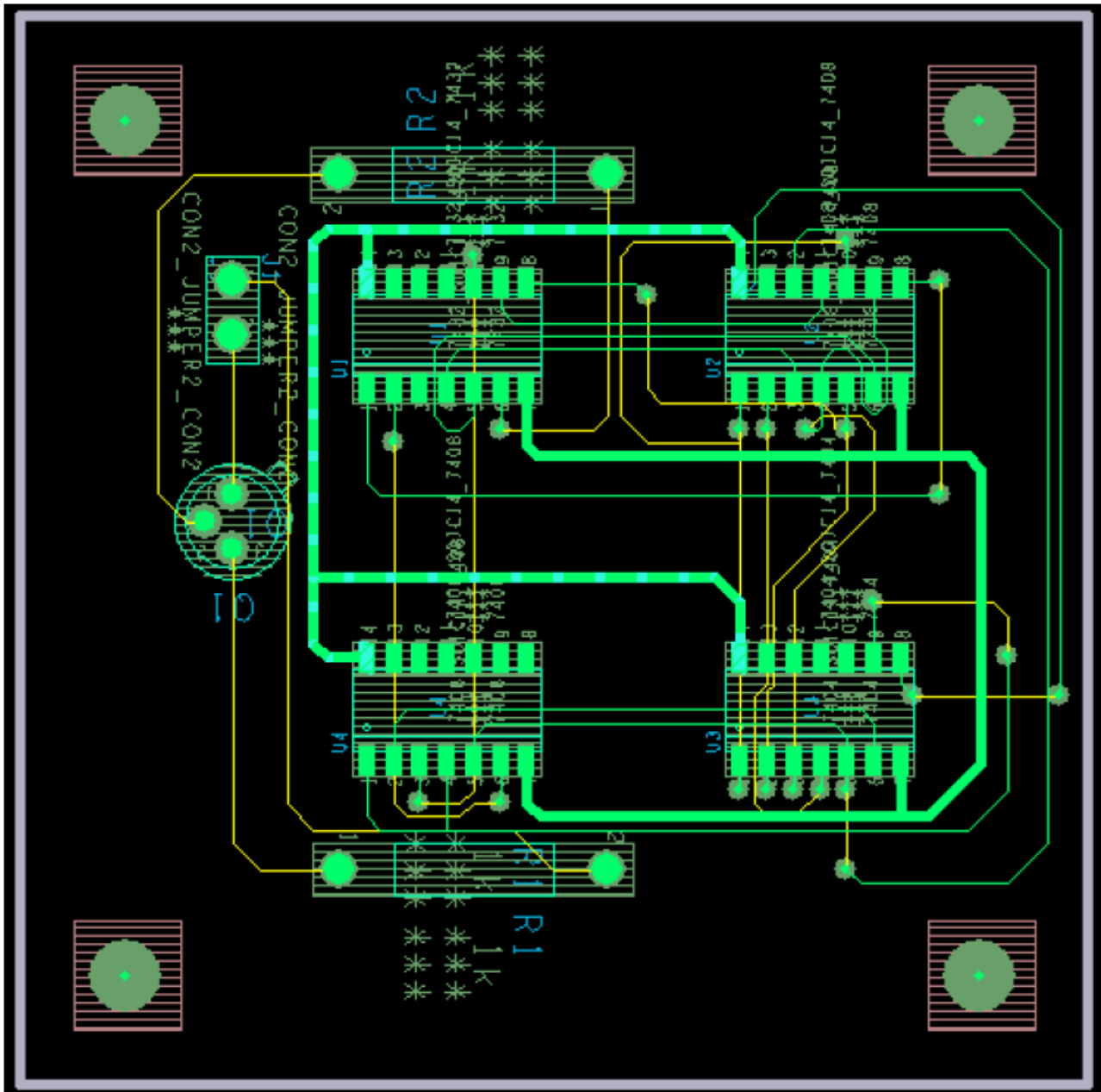


Рис. 9.49. Эталонный образец платы

Файл образца платы fulladd.brd доступен по адресу: /complite / allegro

Более подробную информацию о маршрутизации плат вы найдете в специальных руководствах компании Cadence.

9.13. Автоматическая маршрутизация с помощью PCB Editor

OrCAD PCB Editor поддерживает автомаршрутизацию платы, компонентов и DRC.

Автомаршрутизация предполагает, что цепи на законченной плате проложены. Разводка компонента выполняется только для цепей, прикрепленных к выбранному компоненту.

Для разводки платы:

1. Выберите Route - PCB Router - Route Automatic. Появится диалоговое окно Automatic Router.
2. Нажмите Route. Плата будет разведена.

Для получения дополнительной информации см. документации PCB Editor.

Примечание: Эта опция может быть недоступна, если вы используете программу PCB Editor Lite.

9.14. Автоматическая маршрутизация с помощью OrCAD PCB Router

При выборе автотрассировщика OrCAD PCB Router маршрутизируется вся плата. PCB Router использует маршрутизацию на основе Shape-Based или бессеточной технологии и является более быстрым инструментом маршрутизации.

Согласно ей все объекты печатной платы моделируются в виде совокупности геометрических фигур (прямоугольник, круг, дуга, трасса, полигон). При бессеточной технологии каждый объект моделируется не набором узлов сетки, а геометрически точно, за счет чего достигается более плотный монтаж. Харак-

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

терная особенность бессеточной технологии – меньшие затраты памяти компьютера. Она в основном предназначена для маршрутизации многослойных печатных плат с высокой плотностью расположения компонентов в автоматическом, интерактивном и ручном режимах. Особые преимущества имеет при наличии планарных компонентов, выполненных в разных системах единиц (метрической и английской).

Чтобы использовать автоматический маршрутизатор PCB Router:

1. В меню Пуск, выберите OrCAD PCB Router.
2. Укажите файл дизайна, который должен быть загружен.

Вы можете открыть .dsn файл, созданный в папке allegro вашего проекта, если вы выполнили предыдущие шаги в этой главе. Вы также можете открыть файл FULLADD.ses из complete/allegro.

В полной версии программы PCB Router появится диалоговое окно PCB Router ShapeBased Automation Software отображающее файл дизайна (рис. 9.50).

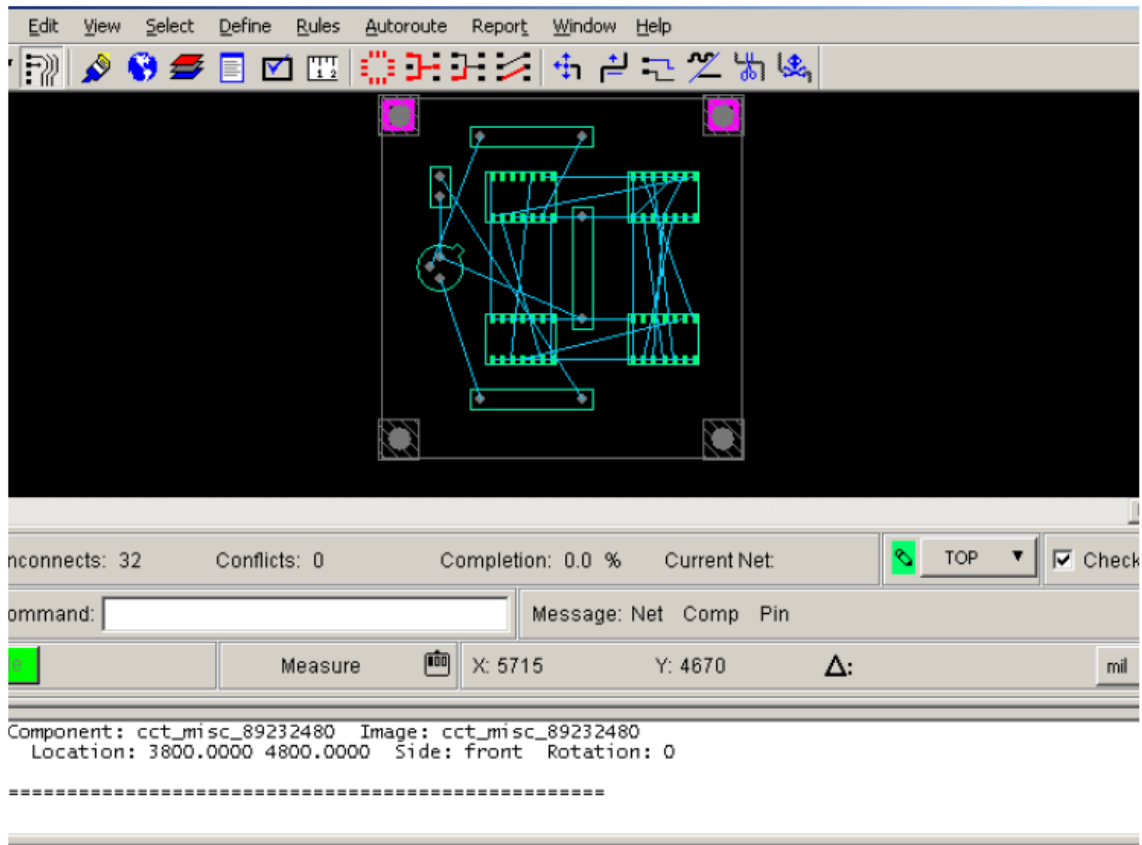


Рис. 9.50. Окно маршрутизатора PCB Router в полной программе

3. Выберите Route из меню AutoRoute. Появится диалоговое окно AutoRoute.

4. Выберите опцию Basic в диалоговом окне AutoRoute.

Для получения дополнительной информации см. *PCB Router User Guide* и *PCB Router Tutorial*.

5. Чтобы начать автоматическую маршрутизацию, нажмите кнопку OK.

Процесс маршрутизации начинается и плата будет разведена.

Примечание. В учебной программе PCB Router Lite файл FULLADD.ses открывается и указываются ошибки в файле.

Поэтому мы будем использовать учебные файлы из библиотеки PCB Router. Путь к файлам:

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

C:/Cadence/SPB_17.2/share/specctra/tutorial/lesson1.dns (рис. 9.51).

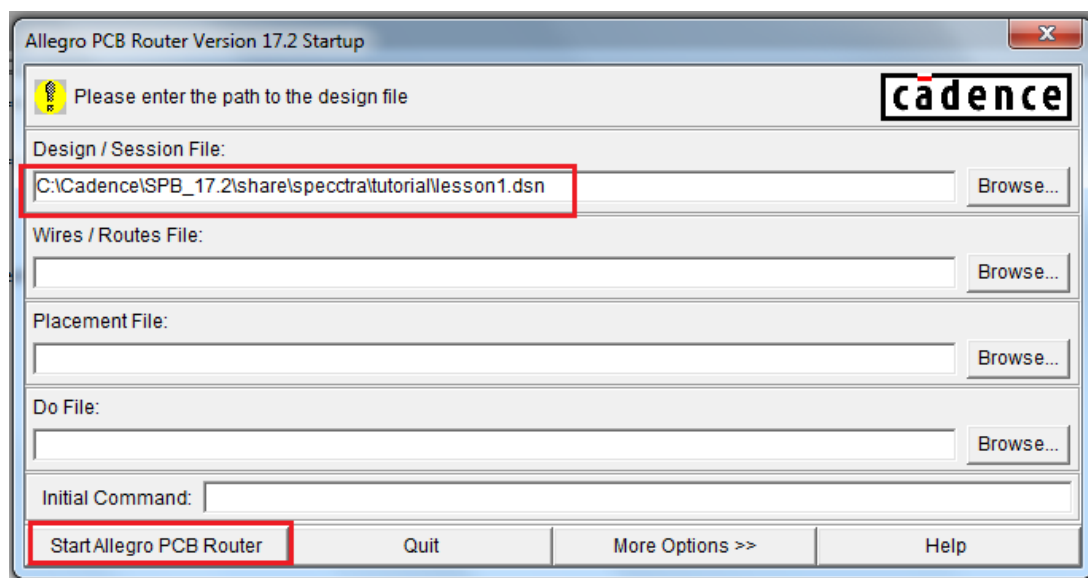


Рис. 9.51. Выбор учебного файла lesson1

Нажмите Start Allegro PCB Router. Соглашаемся работать в Демоверсии без сохранения результатов (рис. 9.52).

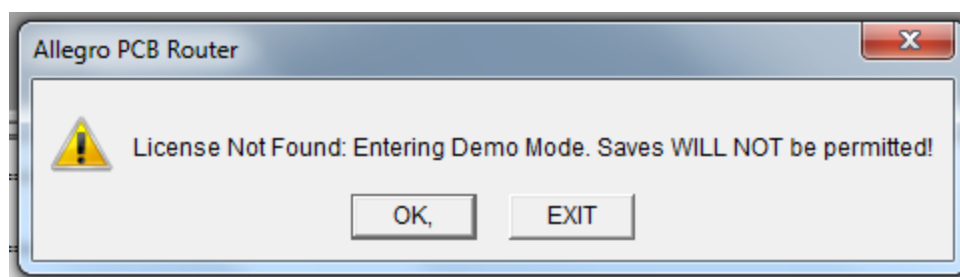


Рис. 9.52. Работа в демоверсии без сохранения результатов

На рабочем поле появится макет платы с установленными компонентами (рис. 9.53).

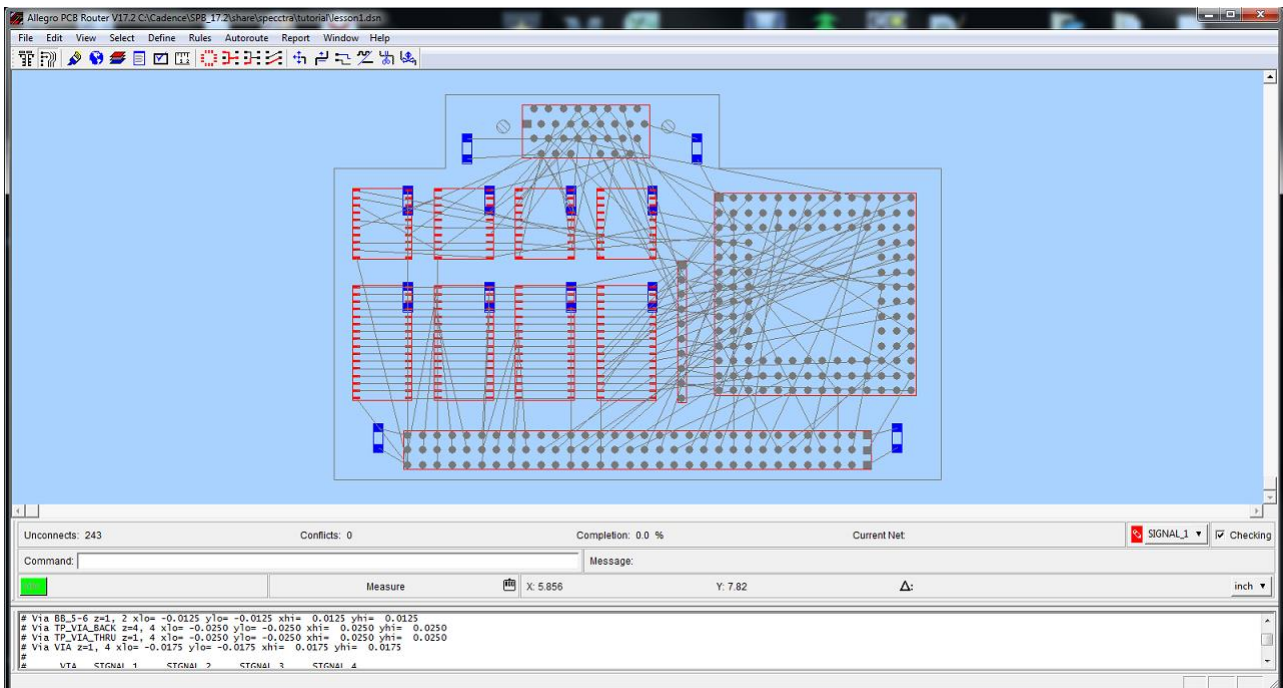


Рис. 9.53. Макет платы с компонентами

Выполняем установки для автомаршрутизации: Autoroute>Route>Basic (рис. 9.54)

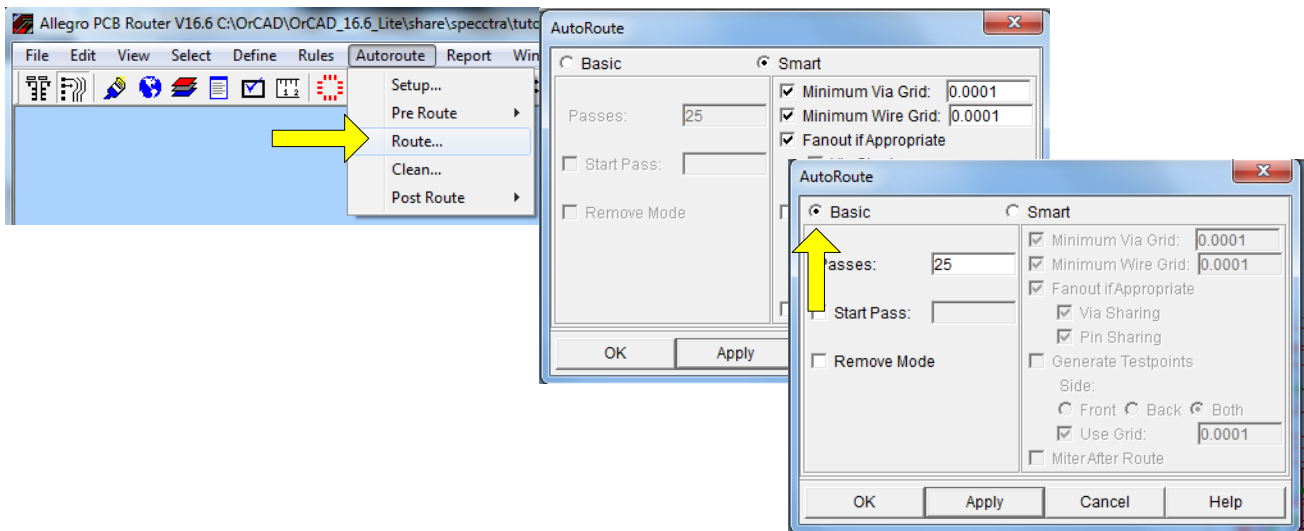


Рис. 9.54. Установки для автоматической маршрутизации

Игнорируем предупреждение об ошибке лицензии (рис. 9.55).

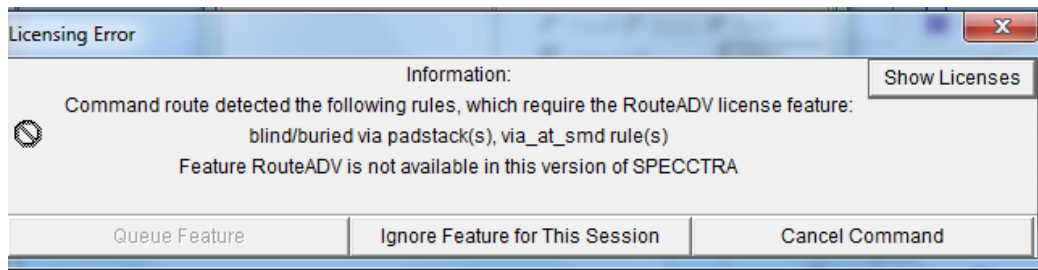


Рис. 9.55. Предупреждение об ошибке лицензии

Несколько секунд выполняется автомаршрутизация и мы получаем разведенную плату (рис. 9.56)

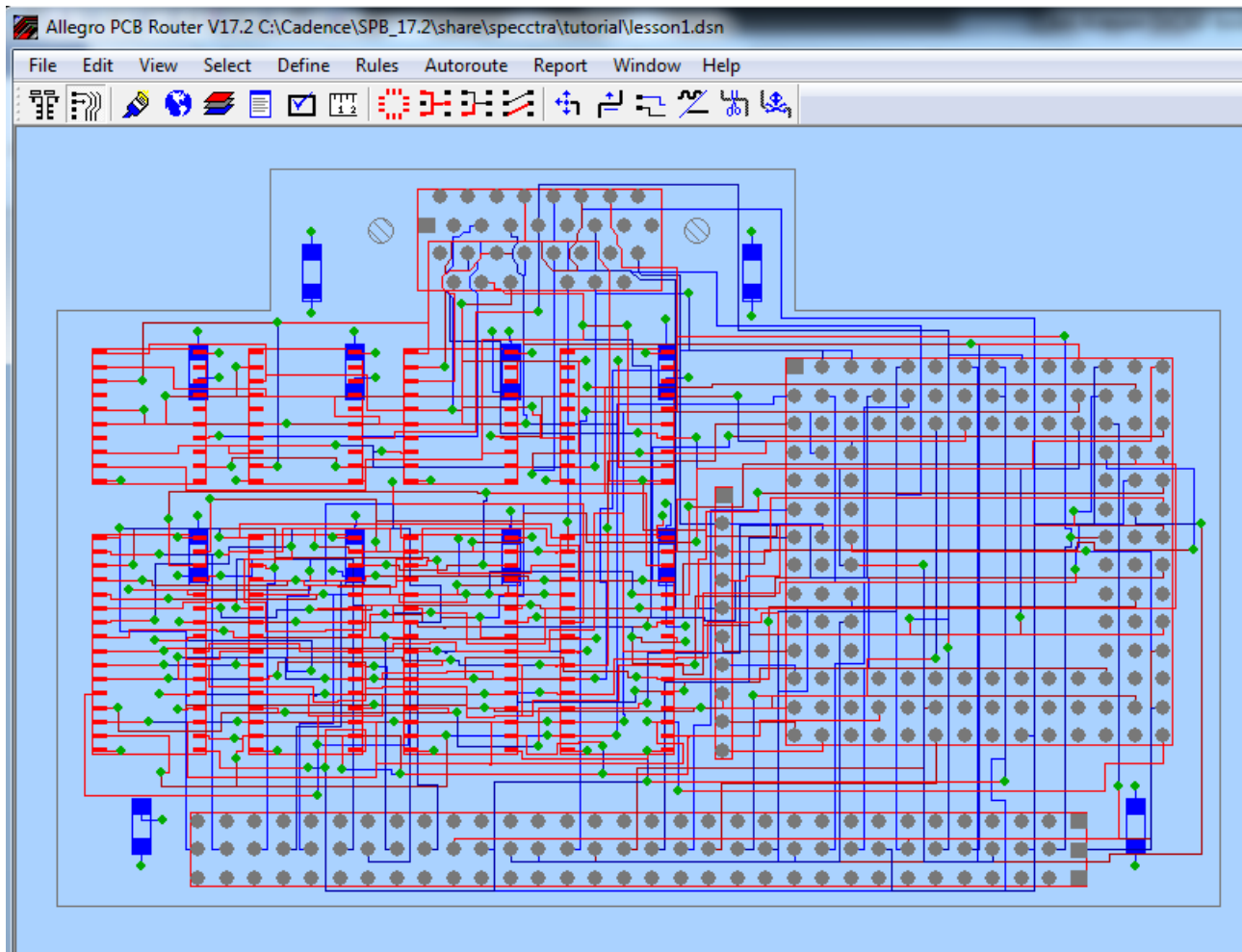


Рис. 9.56. Разведенная плата

9.15. Пост-обработка

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

В данном разделе представлены некоторые из задач, которые не являются частью процесса размещения и маршрутизации, но связаны между собой и могут быть выполнены с помощью редактора OrCAD PCB. Чтобы узнать больше о пост-обработке, обратитесь к документации редактора печатных плат.

9.15.1. Переименование компонентов вручную

После завершения размещения и маршрутизации вашей печатной платы, вы можете переименовать компоненты вручную на печатной плате в определённом порядке.

1. В меню Edit выберите Text.
2. Щелкните левой кнопкой мыши на позиционном обозначении, которое нужно изменить. Выбранное позиционное обозначение появляется в командной строке.
3. Изменение позиционное обозначение как требуется в командной строке и нажмите клавишу Enter.

PCB Editor переименовывает компоненты. Ссылочные обозначения для компонентов на плате изменятся.

4. Сохраните файл платы и закройте PCB Editor.

9.15.2. Автоматическое переименование компонентов

1. Выберите Logic - Auto Rename Refdes. Появится диалоговое окно Rename Refdes.

Примечание: в учебной версии эта опция может отсутствовать.

2. Выберите опцию Use default grid. Эта опция использует сетку по умолчанию, которая представляет встроенный метод переименования компонентов (см рис. 9.57).

3. Нажмите кнопку More. Появится диалоговое окно Rename Ref Des Set Up, на котором вы установили все параметры ссылок названий (рис. 9.58). Для получения дополнительной информации обратитесь к документации редактора печатных плат.

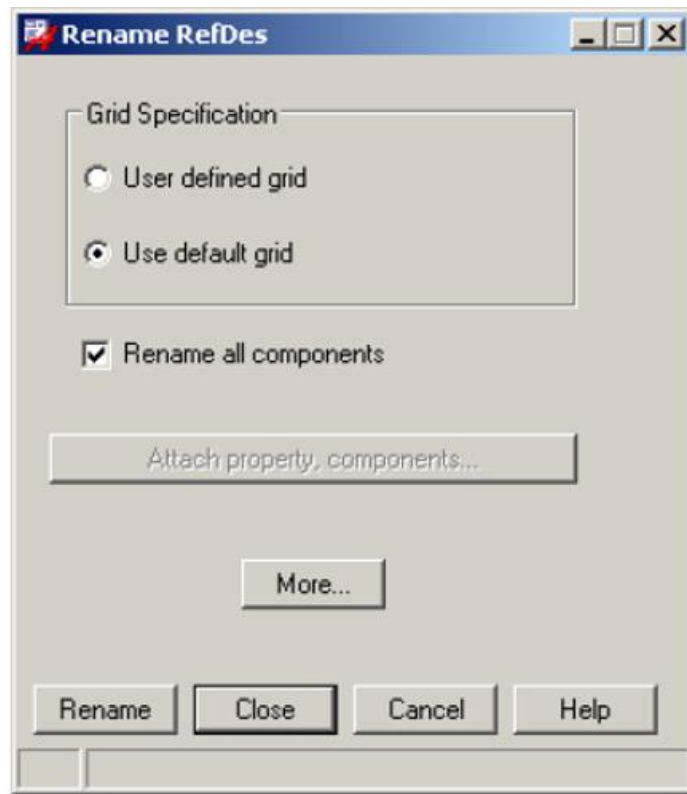


Рис. 9.57. Окно автоматического переименования

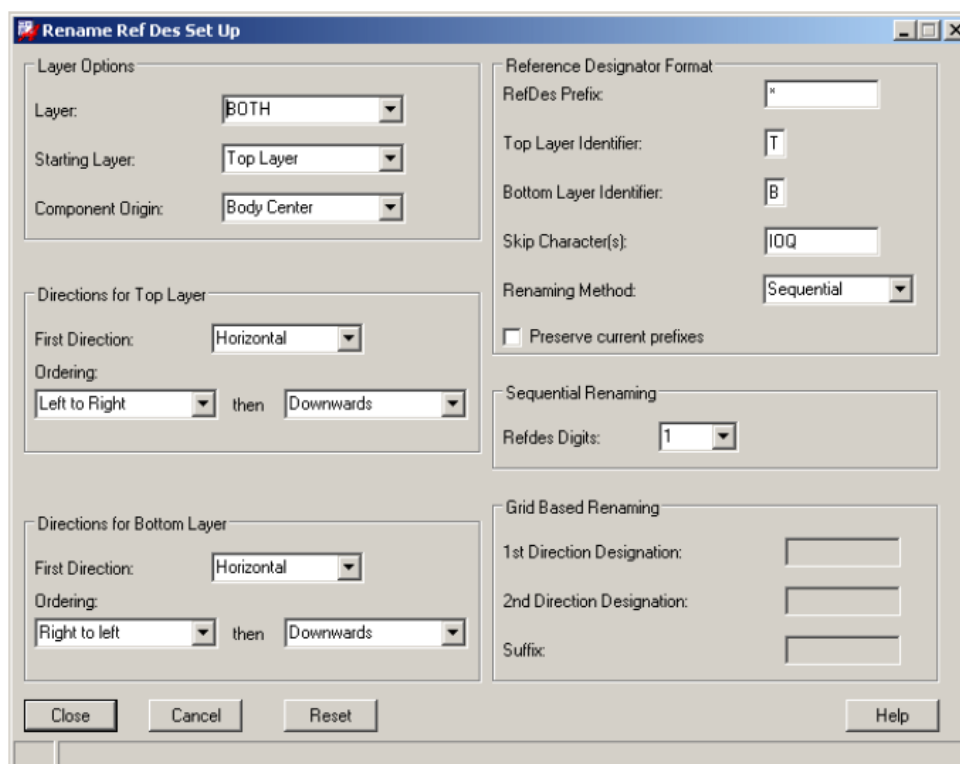


Рис. 9.58. Установки автоматического переименования

4. Примите значения по умолчанию и нажмите кнопку Close в диалоговом окне Rename Ref Des Set Up, чтобы закрыть диалоговое окно и сохранить настройки.

Снова появляется диалоговое окно Rename RefDes.

5. Нажмите Rename в диалоговом окне Rename RefDes .

Редактор PCB автоматически переименовывает каждый компонент в вашей конструкции в одной операции. Статус операции переименования отображается в командной строке.

9.15.3. Обратное аннотирование

При создании печатной платы вы можете внести некоторые изменения в файле редактора PCB платы (.brd). В результате файл платы и файл дизайна в Capture могут быть не синхронизированы. Для того, чтобы убедиться, что оба

эти файла находятся в синхронизации, вы можете выполнить backannotate (обратное аннотирование) изменений в файле печатной платы для Capture.

Когда вы выполняете backannotate, информация, например, расположение компонентов и имена компонентов (изменились в связи с переименованием) добавляется к схеме в Capture.

Для выполнения обратного аннотирования изменений в схеме:

1. Откройте FullAdd.opj в Capture.
2. В окне менеджера проекта, выберите fulladd.dsn.
3. Выберите пункт меню Tools – Back Annotate. Появится диалоговое окно Backannotate (рис. 9.59).
4. Выберите вкладку PCB Editor, если она не выбрана.
5. Выберите кнопку Generate Feedback Files (если она ещё не выбрана).

Примечание: Убедитесь, что правильный конфигурационный файл (allegro.cfg) задается в диалоговом окне настройки. Чтобы просмотреть файл конфигурации, нажмите кнопку Setup. Путь к файлу конфигурации должен быть <install_dir>\tools\capture\allegro.cfg, где <install_dir> расположение установки программы.

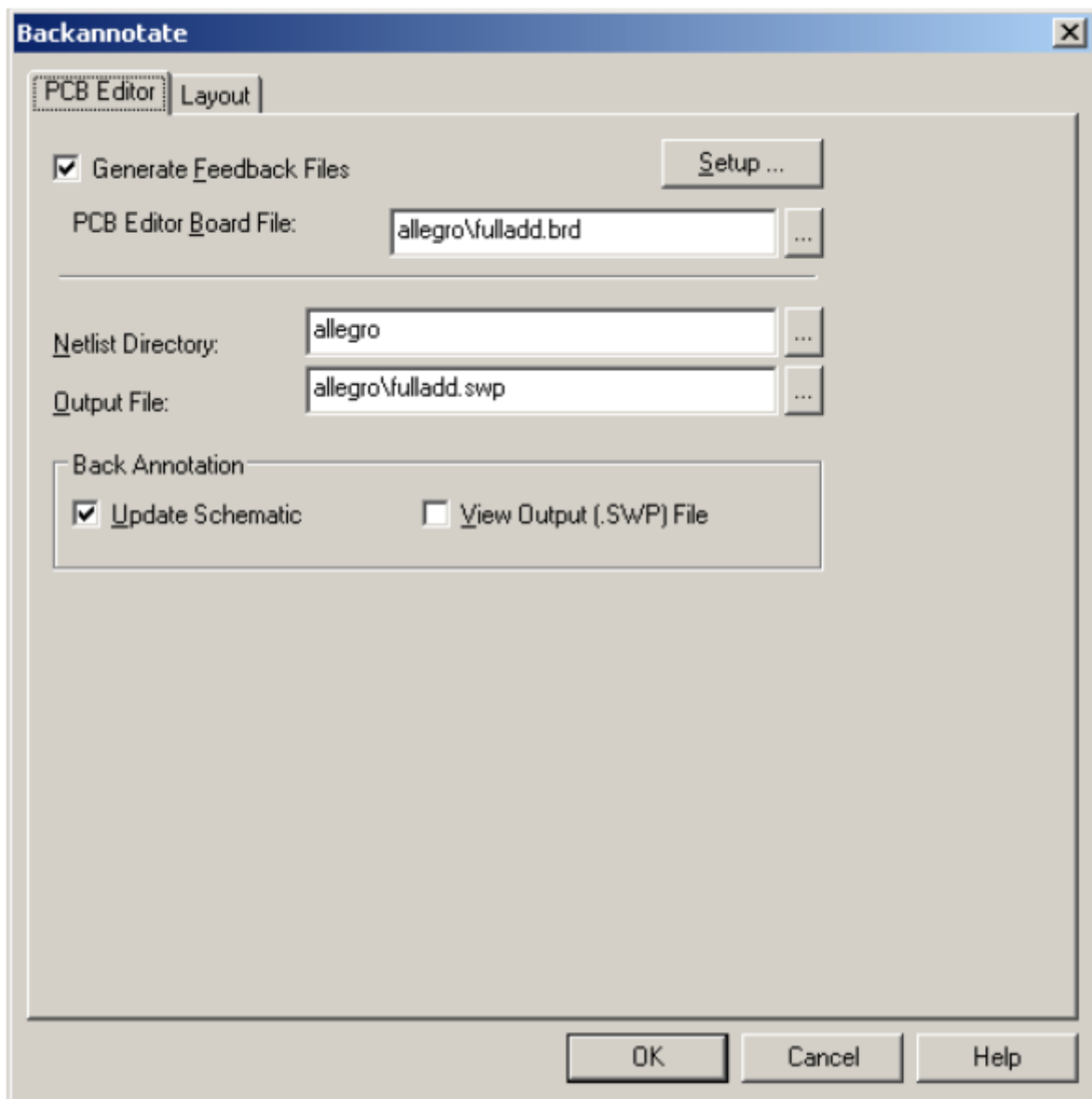


Рис. 9.59. Окно обратного аннотирования

Примечание: Убедитесь, что текстовое поле Netlist Directory содержит каталог, где будут сохранены обновлённые файлы списка соединений (PST * .dat). Расположением по умолчанию является подкаталог Allegro в каталоге проекта.

6. Перейдите в папку, где необходимо сохранить файл .swp. Файл .swp генерируется Capture после внесения изменений в файл платы (.brd). Чтобы узнать больше о файле .swp см. Руководство пользователя OrCAD Capture. Для

нашего учебника .swp файл называется fulladd.swp и каталог, в котором будет сохранен файл, является: / complite / allegro.

7. Установите флажок Update Schematic (если он еще не выбран), если вы хотите, чтобы в Capture эскизное проектирование (fulladd.dsn) было обновлено с последней информацией backannotation из файла .swp.

8. Выберите флажок View Output (.swp) File, чтобы автоматически открыть файл .swp в отдельном окне Capture для просмотра и редактирования после того, как файл .swp генерируется. Этот флажок не выбран по умолчанию.

9. Нажмите кнопку ОК в диалоговом окне Backannotate. Появится сообщение с запросом, чтобы сохранить изменённый дизайн до создания нового списка соединений и файл .swp (см. рис. 9.60).

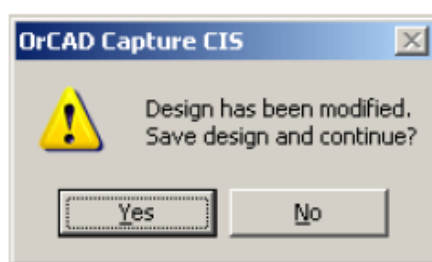


Рис. 9.60. Запрос о сохранении дизайна

10. Нажмите кнопку Yes в окне сообщения.

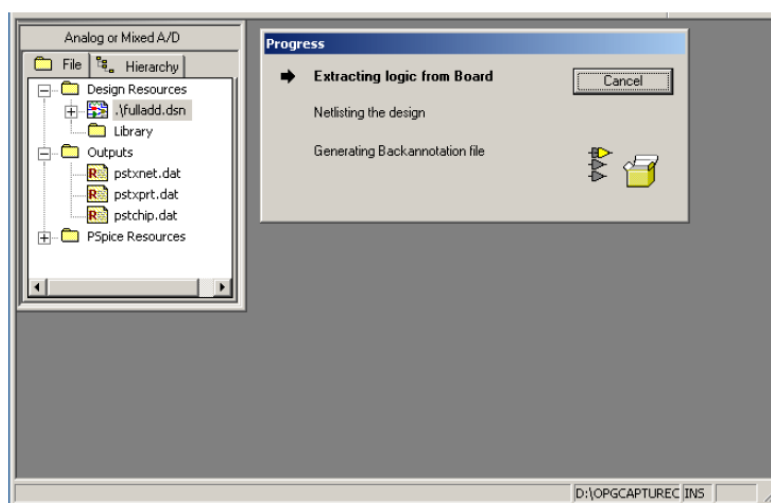


Рис. 9.60. Генерация нового списка соединений

Capture генерирует файлы списка соединений (PSTCHIP.DAT, PSTXPRT.DAT и PSTXNET.DAT) (рис. 9.60) и создаёт файл fulladd.swp в указанном месте каталога, который в данном случае является \complete\allegro.

Файл .swp открывается в отдельном окне Capture, а также появляется в каталоге выходов в окне диспетчера проекта.

Схема обновляется в соответствии с изменениями в файле платы на основе созданного файла .swp.

Точно так же, если файл платы открыт в редакторе печатных плат и вы делаете изменения в эскизном проектировании, вы можете быть уверены, что эти изменения перейдут на плату во время создания списка соединений в Capture.

Чтобы сделать это:

1. В окне менеджера проекта, выберите fulladd.dsn.
2. В меню Tools выберите Create Netlist.
3. На вкладке PCB Editor в диалоговом окне Create Netlist укажите расположение каталога базовой платы. Для нашего учебника, каталог базовой платы /complete/allegro/fulladd.brd.
4. В текстовом поле Output Board File укажите имя платы и расположение каталога, где будет создан обновлённый файл платы.
5. Нажмите кнопку ОК в диалоговом окне Create Netlist.

Capture генерирует файлы списка соединений (PSTCHIP.DAT, PSTXPRT.DAT и PSTXNET.DAT) и обновлённый файл платы создаётся в указанном месте каталога. Изменения в эскизном проекте появляться в файле платы.

9.16. Кросс-зондирование и кросс-выделение между редактором

печатных плат и Capture

OrCAD PCB Editor тесно интегрирован с OrCAD Capture. В результате вы можете использовать кросс-зондирование для проверки потока информации между эскизным проектированием и дизайном платы и наоборот.

Cross probing позволяет выбрать объект в схемы Capture и увидеть соответствующий объект в редакторе печатных плат.

Чтобы включить кросс-зондирование, необходимо включить Intertool (межинструментальную) связь между Capture и редактором печатных плат.

Чтобы сделать это:

1. В окне менеджера проекта в Capture, выберите fulladd.dsn.
2. В меню Options в Capture, выберите Preferences.
3. Выберите вкладку Miscellaneous.
4. Убедитесь, что установлен флажок Enable Intertool Communication в секции Intertool Communication.
5. Нажмите кнопку ОК.

Перед тем, как начать cross probing, откройте окна Capture и редактор PCB. Выберите компонент в Capture. Редактор PCB автоматически отображает соответствующие компоненты.

Например, если вы выбираете R1 в файле FULLADD.DSN, соответствующий резистор R1 будет отображаться в редакторе печатных плат, как показано на рис. 9.61.

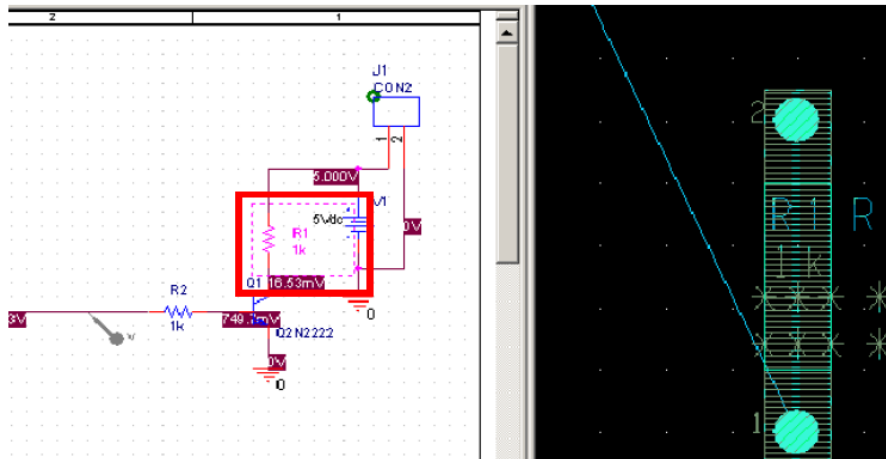


Рис. 9.61. Отображение компонента в редакторе печатных плат

Кросс подсветка позволяет выбрать объект в редакторе печатных плат и увидеть соответствующий объект выделенным в Capture.

В случае перекрестной подсветки между редактором печатных плат и Capture, сначала выберите Highlight из меню Display, а затем выберите компонент в редакторе печатных плат и соответствующий компонент будет выделен Capture.

Например, если вы выбираете R1 в файле FULLADD.BRD, соответствующий резистор R1 будет выделен в Capture, как показано на рис. 9.62.

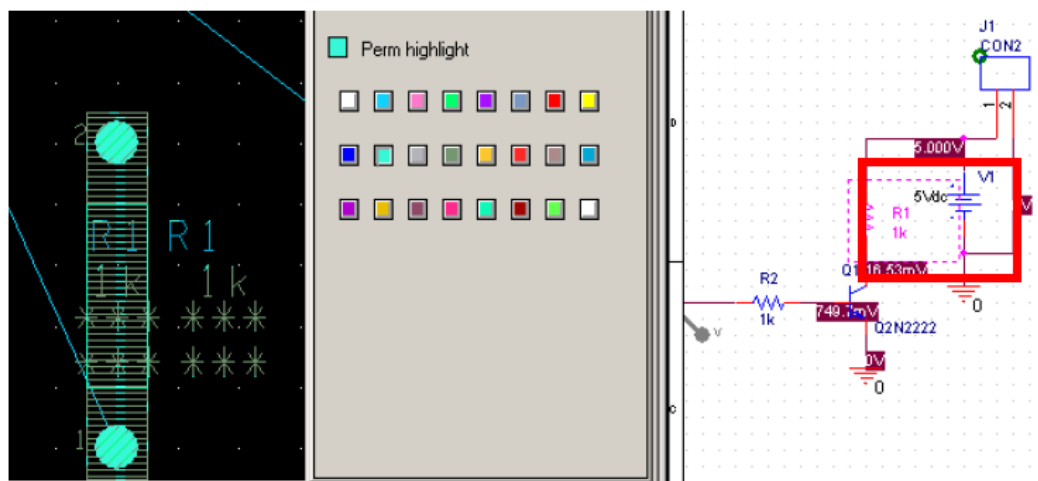


Рис. 9.62. Подсветка компонента в Capture

Примечание: Если вы хотите отключить подсветку, выберите Dehighlight из меню Display.

Примечание: в учебных версиях опции кросс-зондирования и кросс-выделения могут быть отключены.

9.17. Генерация вывода

Последней задачей в создании дизайна платы является создание выходных файлов. Вы можете создать Gerber файлы, файлы сверления, DXF-файлы и принтер / плоттер файлы.

Перед тем, как создавать отчёты и выходные файлы, вы должны сделать резервную копию вашего дизайна и очистить дизайн. Для того, чтобы очистить ваш дизайн:

1. Выберите Route – Gloss – Line Parameters.

Появится диалоговое окно Line Smoothing.

2. Примите значения по умолчанию и нажмите Gloss.

Конструкция очищается. Теперь вы можете генерировать желаемые выходные файлы и отчёты.

Перед созданием выходного файла (artwork), убедитесь, что Вы запустили Update DRC из меню Tools в редакторе печатных плат.

9.18. Выходные файлы

Используя редактор OrCAD PCB, вы можете создавать различные файлы, которые в дальнейшем могут быть использованы с различными инструментами сторонних производителей, таких как GerbTool, VisualCAD, AutoCAD, и так далее.

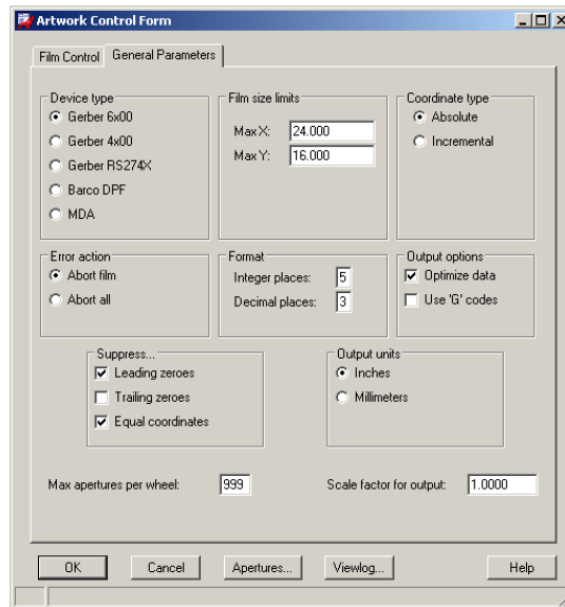


Рис.9.63. Окно Artwork Control Form

Для создания этих выходных файлов, выполните следующие действия:

1. В меню Manufacture выберите Artwork.

Появится диалоговое окно Artwork Control Form (рис. 9.63).

2. На вкладке General Parameters выберите кнопку Gerber RS274X опцию в разделе Device type section.

3. Примите значения по умолчанию и нажмите кнопку ОК, чтобы закрыть диалоговое окно Artwork Control Form.

4. Снова выберите Artwork, из меню Manufacture.

5. Выберите вкладку Film Control.

6. Установите флажки, соответствующие слою (слоям) пленки в диалоговом окне Artwork Control Form. Для нашего случая выбраны как верхний (TOP) так и нижний (BOTTOM) слои (рис. 9.64).

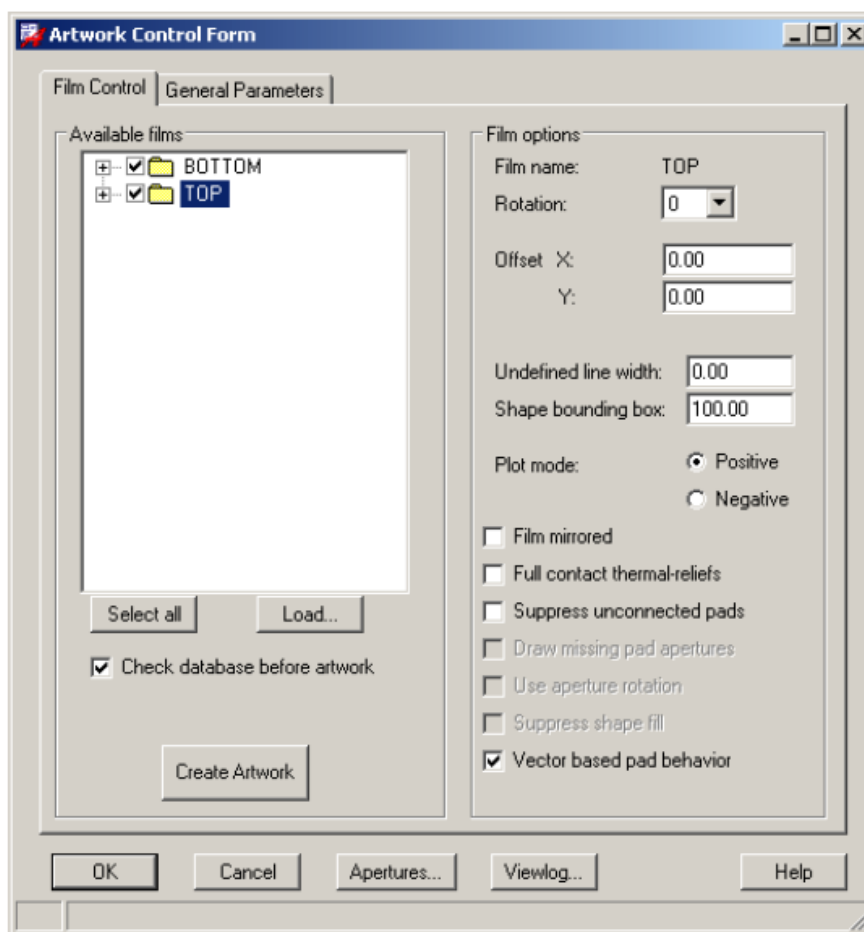


Рис. 9.64. Установка параметров Artwork Control Form

7. Нажмите кнопку **Create Artwork**. Появится окно сообщения, показывающее ход создания Artwork. После этого, в Artwork будут созданы файлы с расширением .ART, которые сохраняются в папке \complete\allegro каталога дизайна (для этого учебника).

8. Нажмите кнопку **OK**, чтобы закрыть диалоговое окно Artwork Control Form.

Вы можете просматривать файлы Artwork , созданные в редакторе печатных плат.

Только Cadence® поддерживает Artwork.

Для просмотра произведения Artwork:

1. В меню File, выберите Import - Artwork. Появится диалоговое окно загрузки Cadence Artwork .

2. Введите или найдите имя файла Artwork (.ART), который вы хотите загрузить в текстовом поле Filename. (рис. 9.65).

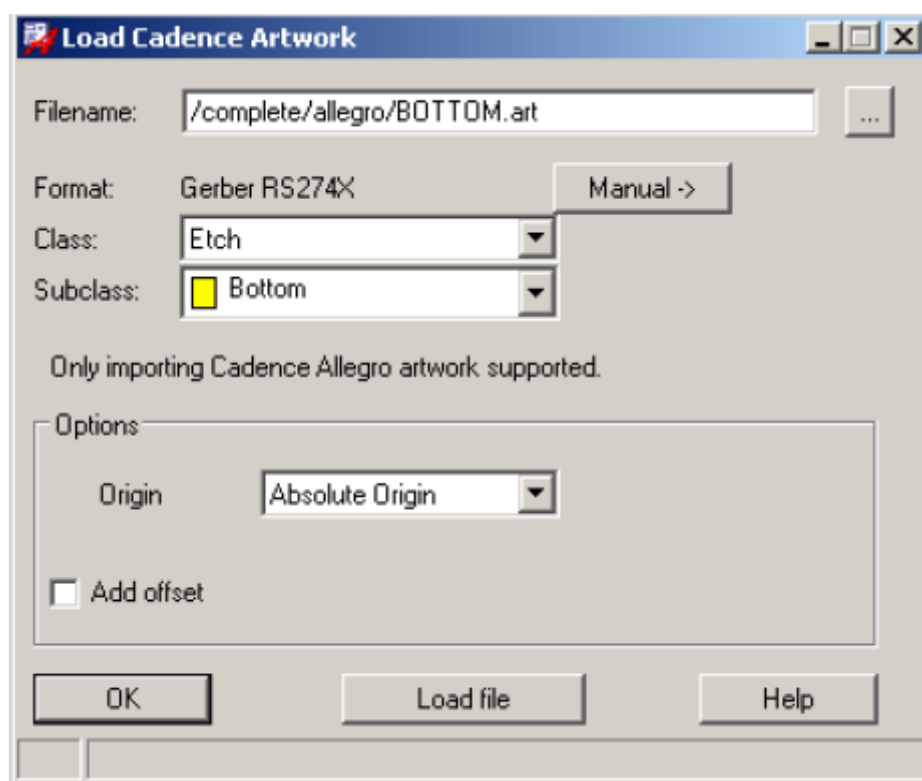


Рис. 9.65. Загрузка файла Artwork

3. Выберите подкласс из выпадающего меню Subclass (рис. 9.65).

4. Нажмите Load File. Динамический прямоугольник, который представляет содержание данных Gerber, появится в рабочей области пользовательского интерфейса.

5. Щелкните левой кнопкой мыши на динамическом прямоугольнике, чтобы поместить в окне дизайна. Artwork помещается в окне дизайна. Образец файла Artwork, BOTTOM.art доступен по адресу: //complete/allegro.

Примечание: Файл образца платы, fulladd_artwork.brd показывающий artworks (TOP.art и BOTTOM.art) доступен по адресу: /complete/allegro.

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

Для переключения между верхней и нижней частями, размещенными на fulladd_artwork.brd, надо выбрать вкладку Visibility (рис. 9.66) и выбрать artwork, который вы хотите просмотреть из представленных в раскрывающемся списке. Выбранная работа отображается в окне дизайна.

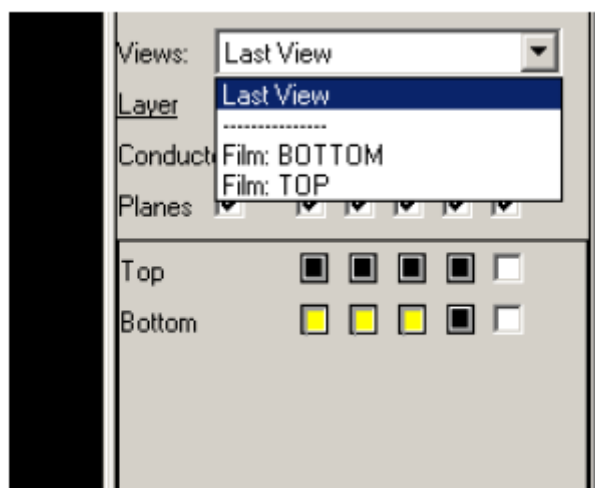


Рис. 9.66. Выбор отображаемой части платы

9.19. Отчеты

Вы можете создавать различные отчеты с использованием редактора OrCAD PCB.

Для создания отчетов, выполните следующие действия:

1. В меню Tools выберите команду Quick Reports.
2. Выберите нужные отчеты, которые надо сгенерировать.

Для дизайна полного сумматор выберите компонент вариант отчета Component Report.

9.20. Резюме

В этой главе вы познакомились с OrCAD PCB Editor, который является инструментом размещения компонентов и маршрутизации печатных плат в

Алехин В.А. Методы проектирования цифровых устройств в составе инфокоммуникационных систем. РТУ – МИРЭА, 2019.

OrCAD. Вы выполнили задачи, необходимые для принятия проекта от OrCAD Capture - инструмента схемного проектирования, чтобы разместить компоненты и сделать маршрутизацию платы в OrCAD PCB Editor. Вы также познакомились с OrCAD PCB Router, который является инструментом, используемым для компоновки и трассировки печатных плат в автоматическом режиме.

Для получения дополнительной информации о OrCAD PCB Editor читайте документацию по PCB Editor:

- Allegro® PCB Editor User Guide

- Allegro PCB and Package Physical Layout Command Reference. Table of Contents

- Allegro Platform Properties Reference

Чтобы узнать больше о OrCAD PCB Router, читайте:

- PCB Router User Guide

9.21. Контрольные вопросы

1. Для чего служит и как используется редактор печатных плат OrCAD PCB Editor ?
2. Какие задачи надо выполнить в Capture для подготовки к работе в OrCAD PCB Editor ?
3. Как выполняют создание списка соединений ?
4. Какие файлы формируются при создании списка соединений ?
5. Какие основные панели содержит окно редактора печатных плат ?
6. Какие режимы работы использует OrCAD PCB Editor ?
7. Как устанавливают параметры проектирования в OrCAD PCB Editor ?
8. Как создают контур печатной платы ?
9. В каких единицах измеряют размеры в OrCAD PCB Editor ?

10. Как использовать команду Pick для создания контура печатной платы ?
11. Как можно удалить неудачный контур печатной платы ?
12. Как можно целиком отобразить контур платы в окне дизайна ?
13. Как выполняют добавление монтажных отверстий на плату ?
14. Как удалить ненужный компонент с платы ?
15. Как размещают компоненты на плате, используя RefDes ?
16. Как можно выполнить поиск компонентов на плате ?
17. Как выполняют проверку правил проектирования и для чего ее используют ?
18. Для чего используют категорию DataTip ?
19. Как можно выделить область на плате ?
20. Как выполняют выбор элементов конструкции с помощью Superfilter ?
21. Для чего и как используют режим привязки компонентов ?
22. Для чего используют окно WorldView ?
23. Как выполняют ручную маршрутизацию цепей VCC и GND ?
24. Как выполняют ручную маршрутизацию остальных цепей ?
25. Для чего служат переходные отверстия и как переключают активный и альтернативный слой ?
26. Как выполняют автоматическую маршрутизацию с помощью OrCAD PCB Router ?
27. Что включает в себя пост-обработка с помощью редактора OrCAD PCB ?
28. Для чего используют кросс-зондирование и кросс-выделение ?
29. Какие выходные файлы для изготовления печатной платы можно создать, используя редактор OrCAD PCB ?

Основная литература

1. Пуговкин А.В. Основы построения инфокоммуникационных систем и сетей: учебное пособие / А.В. Пуговкин. – Томск : Эль Контент, 2014, - 156 с.
2. Пономарев, Д.Ю. Основы построения инфокоммуникационных систем и сетей: учеб. пособие / Д.Ю. Пономарев; Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2014. – 176 с.
3. Гребешков А.Ю. Аппаратные средства телекоммуникационных систем: учебное пособие. – Самара.: ФГБОУ ВО ПГУТИ, 2017. – 296 с.
4. Червенчук И.В. Информационные системы и процессы, моделирование и управление. Моделирование информационных систем с помощью UML: Учебное пособие. – Омск: Омский государственный институт сервиса, 2006. 48с
5. Удовикин В.Л. У 313 Основы построения телекоммуникационных систем и сетей: руководство к лабораторным работам / В.Л. Удовикин. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 49 с.
6. Цифровая связь: методические указания для проведения практических занятий / сост. С.П. Москвитин, А.В. Иванов. – Тамбов: Изд-во Першина Р.В., 2012. - 33с.
7. Зеленин А.С. Проектирование инфокоммуникационных сетей. Курсовое проектирование: учеб.-метод. пособие / А.С. Зеленин, Н.В. Тарченко, В.Н. Урядов.-Минск : БГУИР, 2018.- 84 с.
8. TINA V.12. The Complete Electronics Lab for Windows. User Manual. <http://tina.com>.

9. Алехин В.А. Электротехника и электроника. Компьютерный лабораторный практикум в программной среде TINA-8 : учебное пособие для вузов / В. А. Алехин. — М.: Горячая линия, 2014. — 208 с.
10. Алехин В.А. Электротехника, электроника и схемотехника : Лабораторный практикум в облачной среде схемотехнического проектирования TINACloud: учеб. пособие для вузов /— М.: Горячая линия - Телеком, 2017. — 216 с.
11. Алехин В.А. Электротехника: теория и практика. Моделирование в среде TINA - 8 : учебное пособие для вузов /— М.: Горячая линия - Телеком, 2017. — 308 с.
12. Алехин В.А. Электроника: теория и практика. Моделирование в среде TINA-8: учебное пособие для вузов / — М.: Горячая линия - Телеком, 2017. — 308 с.
13. Алехин В.А. Микроконтроллеры PIC: основы программирования и моделирования в интерактивных средах MPLAB IDE, mikroC, TINA, Proteus : практикум /— М.: Горячая линия-Телеком, 2016. — 248 с.: ил.
14. Алехин В.А. SystemC. Моделирование электронных систем: учебное пособие для вузов / - М.: Горячая линия - Телеком, 2018. — 320 с.
15. Алехин В.А. OrCAD 17.2. Анализ и проектирование электронных устройств: учебное пособие для вузов / - М.: Горячая линия - Телеком, 2019. — 328 с.
16. Махлин А.А. Дельта-сигма модуляция: назад в будущее. //Компоненты и технологии. КОМПОНЕНТЫ И ТЕХНОЛОГИИ • №. 11 '2010. с. 154-158.

17. Дельта-сигма модуляция. [Электронный ресурс]. URL:
<http://microsin.net/programming/dsp/delta-sigma-modulation.html>

Дополнительная литература

1. [PSpice 17.2 User Guide \(pspug\).pdf](#) 800 стр.

URL: [http://ecadtools.com.au/documents/PSpice%2017.2%20User%20Guide%20\(pspug\).pdf](http://ecadtools.com.au/documents/PSpice%2017.2%20User%20Guide%20(pspug).pdf)

2. [OrCAD Capture 17.2 User Guide \(cap_ug\).pdf](#)

URL: [http://ecadtools.com.au/documents/OrCAD%20Capture%2017.2%20User%20Guide%20\(cap_ug\).pdf](http://ecadtools.com.au/documents/OrCAD%20Capture%2017.2%20User%20Guide%20(cap_ug).pdf)

3. [Orcad 17.2. Flow Tutorial](#)

Cadence. Product Documentation : C:/Cadence/SPB_17.2/doc/flowtut/

4. OrCAD Capture User Guide. Product Version 17.2. 2016.

URL: [http://ecadtools.com.au/documents/OrCAD%20Capture%2017.2%20User%20Guide%20\(cap_ug\).pdf](http://ecadtools.com.au/documents/OrCAD%20Capture%2017.2%20User%20Guide%20(cap_ug).pdf)

5. OrCAD Lite Reference Product Version 17.2-2016 April 2016

URL:

<https://www.orcad.com/sites/orcad/files/resources/files/OrCAD%2017.2-2016%20Lite%20Limits.pdf>

6. OrCAD CIS User Guide Product Version 17.2-2016 April 2016

URL:

[http://ecadtools.com.au/documents/OrCAD%20CIS%2017.2%20User%20Guide%20\(cisug\).pdf](http://ecadtools.com.au/documents/OrCAD%20CIS%2017.2%20User%20Guide%20(cisug).pdf)

7. Cadence Allegro and OrCAD: What's New in Release 17.2-2016

http://makerdiary.qiniudn.com/Cadence%20SPB%2017.2%20Release_Notes.pdf

8. Dennis Fitzpatrick. Analog Design and Simulation using OrCAD Capture and PSpice. URL: <https://searchworks.stanford.edu/view/12320475>

9. Разевиг В.Д. Система проектирования OrCAD 9.2. М.: Солон-Р. 2001. С. 519

<http://ru.b-ok.org/book/2723191/16264e>

10. Д. Кеон. OrCAD PSpice. Создание электрических цепей. Пер. с англ. А. Осипова. — М.: Издательский дом ДМК!пресс. 2007. — 628 с.

11. Анатолий Иванов. САПР Cadence как основа сквозного маршрута проектирования электроники и СБИС. Cadence Design Systems, Inc. 2009.- 52 с.

12. Александр Акулин, 10 причин перейти на новый релиз САПР печатных плат Cadence Allegro/OrCAD 17.2-2016// Производство электроники, №5, 2016. с. 100-102.

13. В.Ежов. Расширенные возможности моделирования аналоговых и цифровых схем в OrCAD PSpice. По материалам семинара компании PCB SOFT. //

Электроника; наука | технология | бизнес. №10 (00160) 2016. с. 1-8.

14. Зограф Ф. Г. Основы компьютерного проектирования и моделирования радиоэлектронных средств: лабораторный практикум [Элек-

тронный ресурс] / Ф. Г. Зограф. – Красноярск: Сиб. федерал. ун-т, 2011. – 120 с.

15. Б.В. Гусев, Е.В. Лагунов. Моделирование электронных схем в OrCAD. Учебное электронное текстовое издание. – Екатеринбург: Уральский Федеральный университет имени первого президента России Б.Н. Ельцина, 2016.- 48 с.

16. Романовский М.Н. Моделирование аналоговых схем в OrCAD PSpice. – Томск: ТУСУР, 2016. – 76 с.