

# Школа схемотехнического проектирования устройств обработки сигналов

## Занятие 2.

**Некоторые полезные мелочи, о которых почти никто никогда не пишет, опасаясь прослыть любителем банальных фактов**

*Дьявол кроется в деталях.*

*Известное правило советского инженера*

*Но мы-то с тобой не ученые. Ошибка ученого — это, в конечном счете, его личное дело. А мы ошибаться не должны. Нам разрешается прослыть невеждами, мистиками, суеверными дураками. Нам одного не простят: если мы недооценили опасность. И если в нашем доме вдруг завоняло серой, мы просто не имеем права пускаться в рассуждения о молекулярных флуктуациях — мы обязаны предположить, что где-то рядом объявился черт с рогами, и принять соответствующие меры, вплоть до организации производства святой воды в промышленных масштабах. И слава богу, если окажется, что это была всего лишь флуктуация, и над нами будет хохотать весь мировой совет и все школяры в придачу...*

*Аркадий и Борис Стругацкие. «Жук в муравейнике»*

**Владимир Стешенко**

steshenk@sm.bmstu.ru

### Введение

Когда автор приступал к этому циклу статей, у него уже имелся определенный опыт подобных работ. Близится к завершению «Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС», публикуемая журналом Chip News [1], по ее мотивам написана книга «ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов» [2], которая выходит в конце мая этого года, в это же время выходит учебное пособие автора «ACCEL EDA: технология проектирования печатных плат» [3]. Казалось бы, уж как, чему и в какой последовательности излагать материал — ну совсем очевидное и простое мероприятие. Тем не менее автора подспудно преследует мысль, что очень часто в учебной литературе не рассматриваются подробно простые и очевидные на первый взгляд решения и приемы. Причиной такого положения дел, наверное, является опасение, что «над нами будет хохотать весь мировой совет и все школяры в придачу». В этом занятии хотелось бы осветить несколько очевидных на первый взгляд моментов, о которых как-то принято думать как о простых и понятных без дополнительных комментариев. Рассмотрим далее несколько деталей, в которых кроется тот самый дьявол.

### Деталь 1. Тактовые сетки, синхронизация, задержки

В замечательной книге И. С. Потемкина [4] красной нитью проводится исключительно здравая и мудрая мысль о необходимости проектировать синхронные схемы с двухфазной синхронизацией. Ка-

залось бы, данный подход очевиден, и тем не менее автору регулярно доводилось слышать в разных аудиториях слова о реализации задержек на асинхронных узлах, таких как логические элементы или одновибраторы. На последнем научно-техническом семинаре «Проектирование устройств цифровой обработки сигналов на ПЛИС», который каждый год проводится кафедрой СМ-5 «Автономные информационные и управляющие системы» МГТУ им. Н. Э. Баумана, в кулуарах неоднократно звучал вопрос: почему при реализации на ПЛИС задержка на нескольких последовательно соединенных логических элементах (вентиллях), введенных в проект с помощью схемного редактора, не равна сумме задержек на каждом из них? Казалось бы, даже начинающему пользователю систем проектирования на ПЛИС должно быть известно: компилятор при синтезе произведет оптимизацию и просто выбросит излишние элементы — нет, находятся «энтузиасты», пытающиеся отключить данную опцию в компиляторе, забывая при этом, что это можно сделать только глобально, для всего проекта, в результате чего увеличиваются задержки распространения сигналов практически во всех цепях, что приводит к естественному снижению как быстродействия, так и надежности работы всей разрабатываемой системы. Совет здесь может быть только один: если используете в схемах задержки, привязывайте их к главной тактовой частоте — дешевле обойдется при регулировке и настройке.

Что касается собственно вопросов тактирования — не жалеете времени и сил на формирование

как минимум двух тактовых сеток при проектировании арифметических устройств — не прогадаете. Как правило, удобно использовать две тактовые сетки, сдвинутые друг относительно друга на четверть периода (рис. 1).

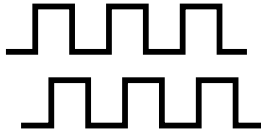


Рис. 1

По первой сетке производится фиксация данных в регистре, затем — обработка (например, выполнение арифметических операций — умножения или сложения), по второй — считывание результата. Такой подход был реализован в схеме демодулятора, описанной в предыдущей статье цикла [5], что позволило избежать сбоев («глитчей», glitches), связанных с конечным временем выполнения арифметических операций на комбинационных схемах. Формирование такой сетки легко обеспечивается известной схемой на двух триггерах D- или T-типа. В последнем случае не стоит забывать о предварительной установке начального состояния триггеров во избежание сбоя взаимного положения сеток.

### Деталь 2. Как же описать проект на ПЛИС?..

Другая большая мозоль в вопросах проектирования устройств цифровой обработки сигналов — переход от использования дискретных элементов к реализации устройства на ПЛИС. Казалось бы, в своих «руководящих материалах» фирмы-производители ПЛИС черным по белому на хорошем английском (понятном даже обычному студенту или вы-

пускнику технического вуза) пишут, что не рекомендуют использовать стандартные библиотеки устаревших макросов (old-style macrofunctions), содержащих элементы 74 серии — нет, всегда находятся охотники перетащить свою схему, разработанную в стандартном ТТЛ-базисе, напрямую, не приложив никаких усилий к ее реализации на языках описания аппаратуры, учитывающих особенности архитектуры ПЛИС (таких как AHDL, Abel HDL). Не буду повторять известные факты, но практически все логические элементы ПЛИС, независимо от производителя, архитектуры и логической емкости, состоят из последовательно соединенных комбинационной части и триггера (регистра) (рис. 2, а), в то время как традиционная ТТЛ и КМОП схемотехника с использованием ИС малой и средней степени интеграции предусматривает изначальное сохранение обрабатываемых данных в регистре (триггере), а затем их обработку и преобразование на комбинационной логике (рис. 2, б). Очевидно, что использование библиотек 74 серии приводит к худшим результатам, чем описание схемы на языках описания аппаратуры.

Конечно, в ряде случаев разработчику бывает трудно освоить методологию проектирования устройств с использованием языков описания аппаратуры, особенно если до этого он не имел особого опыта в написании программ на языках высокого уровня типа С или Паскаль, однако никто и не заставляет создавать весь проект только на HDL. Наиболее удобным и популярным является смешанный режим. В этом случае отдельные узлы описываются на языках описания аппаратуры, для них легко создать специальный символ, отредактировать его для пущей наглядности, а затем спокойно произвести соединения с использованием графического схемного редактора, как это показано на рис. 3.

В этом случае сохраняется наглядность, свойственная графическому вводу, и облегчается работа компилятора. Автор заметил в связи с этим некоторую закономерность.

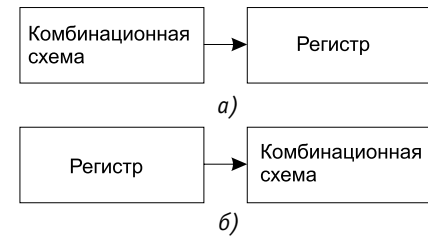


Рис. 2

Мой коллега, отличный программист (как по призванию, так и по образованию), освоил проектирование на ПЛИС. Для него нет особых проблем описать весь проект, вплоть до назначения выводов, целиком на AHDL или VHDL, при этом он гораздо сложнее воспринимает проект, заданный в виде схемы. В свою очередь, я, как человек, умеющий писать только простенькие программки на Фортране или Паскале в объеме стандартного вузовского курса образца середины 80-х годов, привык работать именно в смешанном режиме ввода — описывать ключевые узлы в виде макросов на языках описания аппаратуры, а соединения между ними и назначения выводов производить в графическом редакторе. Очевидно, что каждый из разработчиков в данном случае использовал ту технику, которая ему ближе и к которой он привык на основе предшествующего опыта работы.

Следует отметить, что в последнее время стали распространяться такие способы описания проекта, как ввод его в виде временных диаграмм или граф-схемы автомата.

Так, в пакете Foundation фирмы Xilinx обеспечивается ввод описания цифрового автомата с помощью его графа переходов (State Editor). Данный способ описания проекта позволяет весьма просто и наглядно задать поведение автомата и весьма удобен при разработке различных устройств управления. На рис. 4 приведен пример задания автомата его графом.

В дальнейшем возможно создать для созданного автомата символ и использовать его в редакторе схем.

Подобными возможностями обладает также и пакет StatCAD, позволяющий сгенерировать по введенному графу автомата его описание на языке описания аппаратуры по выбору пользователя.

По опыту автора такой подход к описанию цифровых автоматов безусловно облегчает их синтез с использованием формальной теории, однако следует быть осторожным при практической реализации схем — зачастую «автоматная модель» не позволяет получить самое лучшее решение с точки зрения компактности и быстродействия.

### Деталь 3. Практические рекомендации по разработке печатных плат

Современные электронные узлы значительно отличаются от устройств разработки конца 80-х — начала 90-х годов. Во-первых, новые технологии поверхностного монтажа привели

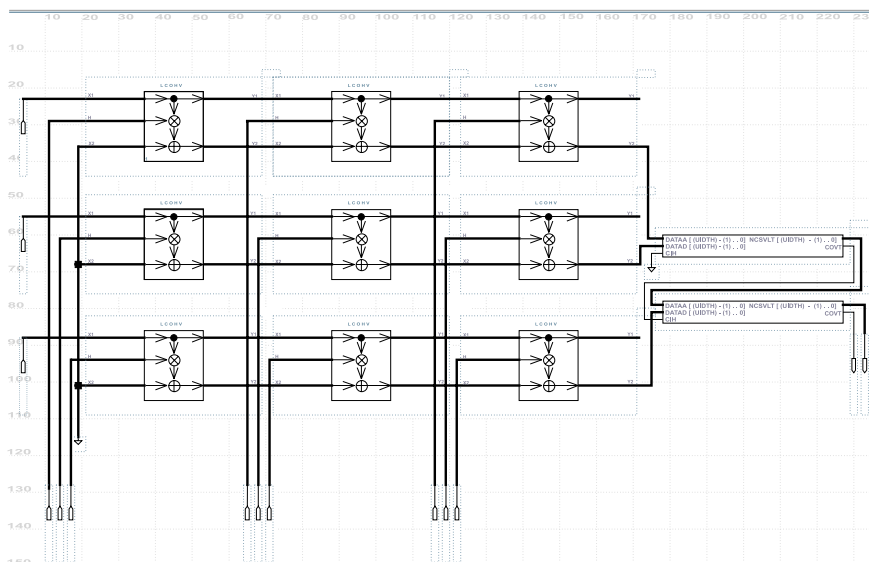


Рис. 3

как минимум двух тактовых сеток при проектировании арифметических устройств — не прогадаете. Как правило, удобно использовать две тактовые сетки, сдвинутые друг относительно друга на четверть периода (рис. 1).

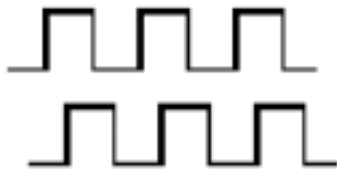


Рис. 1

По первой сетке производится фиксация данных в регистре, затем — обработка (например, выполнение арифметических операций — умножения или сложения), по второй — считывание результата. Такой подход был реализован в схеме демодулятора, описанной в предыдущей статье цикла [5], что позволило избежать сбоев («глитчей», glitches), связанных с конечным временем выполнения арифметических операций на комбинационных схемах. Формирование такой сетки легко обеспечивается известной схемой на двух триггерах D- или T-типа. В последнем случае не стоит забывать о предварительной установке начального состояния триггеров во избежание сбоя взаимного положения сеток.

### Деталь 2. Как же описать проект на ПЛИС?..

Другая большая мозоль в вопросах проектирования устройств цифровой обработки сигналов — переход от использования дискретных элементов к реализации устройства на ПЛИС. Казалось бы, в своих «руководящих материалах» фирмы-производители ПЛИС черным по белому на хорошем английском (понятном даже обычному студенту или вы-

пускнику технического вуза) пишут, что не рекомендуют использовать стандартные библиотеки устаревших макросов (old-style macrofunctions), содержащих элементы 74 серии — нет, всегда находятся охотники перетащить свою схему, разработанную в стандартном ТТЛ-базисе, напрямую, не приложив никаких усилий к ее реализации на языках описания аппаратуры, учитывающих особенности архитектуры ПЛИС (таких как AHDL, Abel HDL). Не буду повторять известные факты, но практически все логические элементы ПЛИС, независимо от производителя, архитектуры и логической емкости, состоят из последовательно соединенных комбинационной части и триггера (регистра) (рис. 2, а), в то время как традиционная ТТЛ и КМОП схемотехника с использованием ИС малой и средней степени интеграции предусматривает изначальное сохранение обрабатываемых данных в регистре (триггере), а затем их обработку и преобразование на комбинационной логике (рис. 2, б). Очевидно, что использование библиотек 74 серии приводит к худшим результатам, чем описание схемы на языках описания аппаратуры.

Конечно, в ряде случаев разработчику бывает трудно освоить методологию проектирования устройств с использованием языков описания аппаратуры, особенно если до этого он не имел особого опыта в написании программ на языках высокого уровня типа С или Паскаль, однако никто и не заставляет создавать весь проект только на HDL. Наиболее удобным и популярным является смешанный режим. В этом случае отдельные узлы описываются на языках описания аппаратуры, для них легко создать специальный символ, отредактировать его для пущей наглядности, а затем спокойно произвести соединения с использованием графического схемного редактора, как это показано на рис. 3.

В этом случае сохраняется наглядность, свойственная графическому вводу, и облегчается работа компилятора. Автор заметил в связи с этим некоторую закономерность.

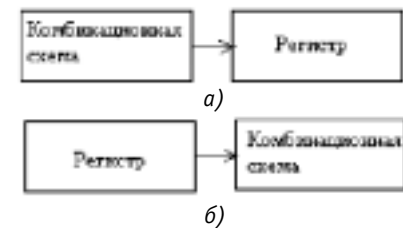


Рис. 2

Мой коллега, отличный программист (как по призванию, так и по образованию), освоил проектирование на ПЛИС. Для него нет особых проблем описать весь проект, вплоть до назначения выводов, целиком на AHDL или VHDL, при этом он гораздо сложнее воспринимает проект, заданный в виде схемы. В свою очередь, я, как человек, умеющий писать только простенькие программки на Фортране или Паскале в объеме стандартного вузовского курса образца середины 80-х годов, привык работать именно в смешанном режиме ввода — описывать ключевые узлы в виде макросов на языках описания аппаратуры, а соединения между ними и назначения выводов производить в графическом редакторе. Очевидно, что каждый из разработчиков в данном случае использовал ту технику, которая ему ближе и к которой он привык на основе предшествующего опыта работы.

Следует отметить, что в последнее время стали распространяться такие способы описания проекта, как ввод его в виде временных диаграмм или граф-схемы автомата.

Так, в пакете Foundation фирмы Xilinx обеспечивается ввод описания цифрового автомата с помощью его графа переходов (State Editor). Данный способ описания проекта позволяет весьма просто и наглядно задать поведение автомата и весьма удобен при разработке различных устройств управления. На рис. 4 приведен пример задания автомата его графом.

В дальнейшем возможно создать для созданного автомата символ и использовать его в редакторе схем.

Подобными возможностями обладает также и пакет StatCAD, позволяющий сгенерировать по введенному графу автомата его описание на языке описания аппаратуры по выбору пользователя.

По опыту автора такой подход к описанию цифровых автоматов безусловно облегчает их синтез с использованием формальной теории, однако следует быть осторожным при практической реализации схем — зачастую «автоматная модель» не позволяет получить самое лучшее решение с точки зрения компактности и быстродействия.

### Деталь 3. Практические рекомендации по разработке печатных плат

Современные электронные узлы значительно отличаются от устройств разработки конца 80-х — начала 90-х годов. Во-первых, новые технологии поверхностного монтажа привели

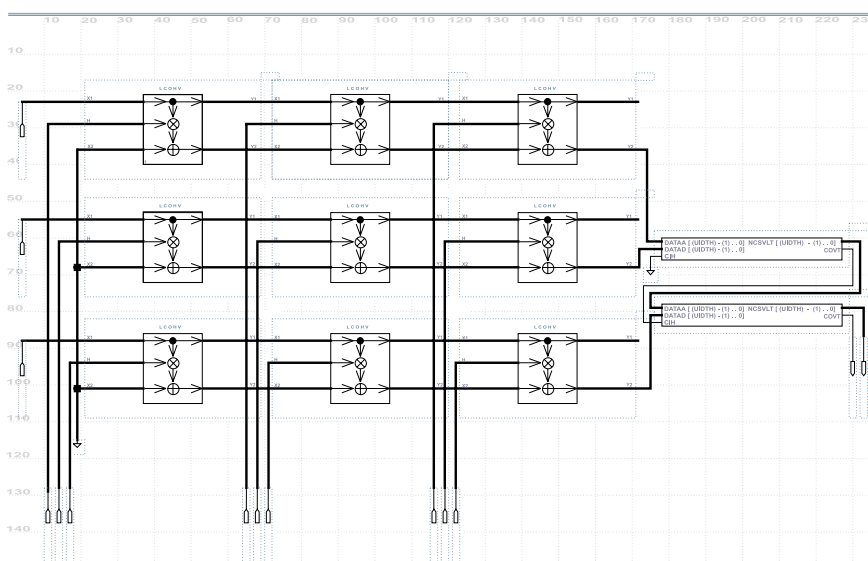


Рис. 3

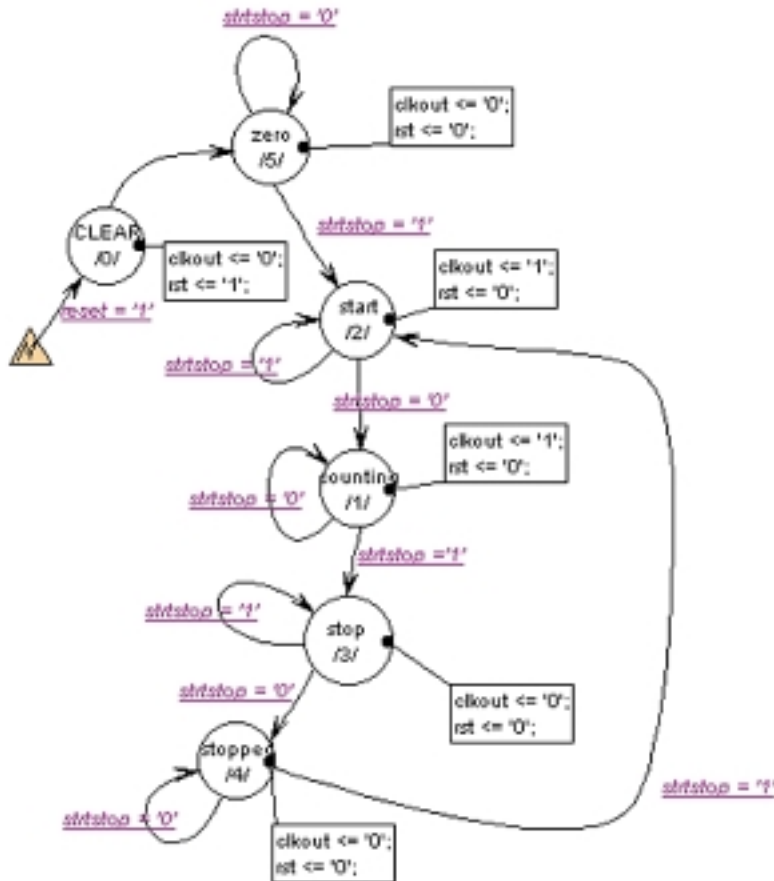


Рис. 4

к уменьшению габаритов компонентов в 3–6 раз. Во-вторых, появились новые корпуса интегральных схем с малым шагом между выводами (0,5–0,65 мм), корпуса с шариковы-

ми выводами (BGA), новые малогабаритные дискретные компоненты и соединители. В-третьих, повысилась точность изготовления печатных плат, добавились возможности для разводки сложных устройств в малых габаритах. Появление новой элементной базы позволяет говорить о возможности воплощения сложных систем на одной плате и даже на одном кристалле (system-on-chip). Это означает, что на одной и той же типичной плате устройства обработки сигналов в малых габаритах размещаются высоко-

чувствительный аналоговый тракт, аналого-цифровой преобразователь, высокоскоростная схема цифровой обработки на процессоре и (или) программируемых логических интегральных схемах, буферные элементы и драйверы линий связи, элементы стабилизаторов напряжения питания и преобразователей уровня и другие узлы. Естественно, это накладывает свой отпечаток на методологию разработки платы.

При проектировании плат с использованием средств САПР необходимо всегда помнить, что сколь бы совершенными ни были алгоритмы автоматической трассировки, они никогда не заменят работу конструктора. В лучшем случае в автоматическом режиме возможна трассировка малочувствительных, медленных (до 3–5 МГц) цифровых цепей. А серьезные платы — ручками, господа, ручками.

Особое внимание следует обратить на проблему заземления. Каждая фирма-производитель элементной базы уделяет ей огромное внимание в своих Application Notes, однако до сих пор многие разработчики наступают на одни и те же грабли.

Как известно, земляная шина определяется как эквипотенциальная поверхность, потенциал которой служит для схемы уровнем отсчета напряжений.

При проектировании земляных цепей преследуются две цели. Во-первых, следует помнить, что заземление минимизирует напряжение шумов, возникающее при прохождении токов от нескольких схем через общее сопротивление земли. Во-вторых, необходимо исключить образование контуров заземления, чувствительных к электромагнитным полям и разностям потенциалов.

Таким образом, заземление представляет собой обладающую низким импедансом цепь возврата тока. Отсюда ясно, что протекание любого тока в системе заземления приводит к появлению разности потенциалов. Ясно также, что эта разность потенциалов должна быть минимальной. Следовательно, при проектировании топологии земли нужно обеспечить импеданс заземления на как можно более низком уровне и контролировать токи, протекающие между источниками и нагрузками.

Поэтому необходимо использовать несколько цепей заземления, соединенных в

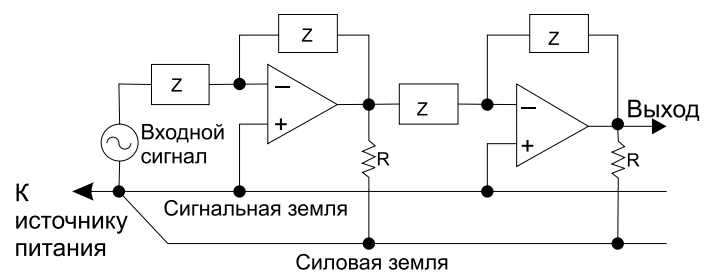


Рис. 5. Схема заземления для двух аналоговых земель

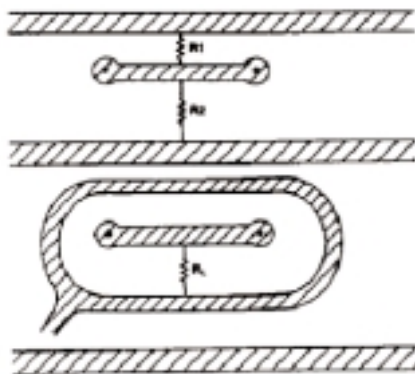


Рис. 6. Использование защитного кольца

одной точке. Причем традиционного деления на аналоговую и цифровую земли может оказаться недостаточно. На рис. 5 представлена схема заземления для двух аналоговых земель, причем одна служит землей для слабого входного сигнала, а другая — для мощного выходного.

Следует избегать заземления, обычно именуемого гирляндой. Рекомендуется использо-

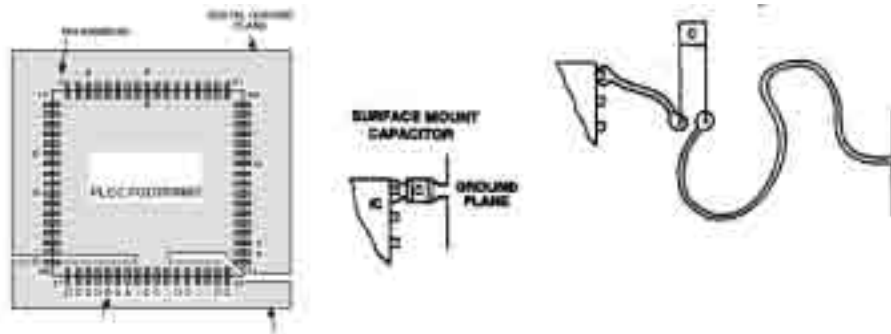


Рис. 7. Установка фильтрующих емкостей

вать одноточечные схемы заземления, когда различные земли соединяются в точке ввода у разъема питания.

Если разрабатываемая плата, содержащая как аналоговые, так и цифровые узлы, должна работать на достаточно высокой тактовой частоте (>1 МГц), то не следует «экономить на спичках» и разрабатывать двухслойную плату. В этом случае необходимо использовать многослойную плату, в которой внутренние

слои представляют собой сплошные плоскости земли и питания.

Нелишне напомнить об известном всем скин-эффекте и зависимости импеданса от частоты, поэтому область металлизации одного и того же размера будет иметь различный импеданс на низких и высоких частотах. Рекомендуется все свободное пространство платы заполнять сплошной областью металлизации, соединенной с общей шиной, чтобы избежать наводок.

Выбирая шаг сетки трассировки, надо учитывать не только разрешающую способность фотоплоттера и другого технологического оборудования, но и помнить о возможных токах утечки через плату. Особенно это критично к входным высокоомным цепям усилителей, в этом случае неплохой мерой является защитное кольцо (рис. 6)

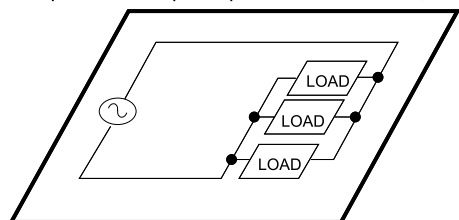
Если передается скоростной сигнал на достаточно большое расстояние, то сделать это можно только по согласованной линии на низкоомную нагрузку.

Несколько слов следует сказать о фильтрующих емкостях. Очень часто начинающий разработчик чисто механистически устанавливает их где придется, руководствуясь толь-

ко нормой установки блокирующих емкостей на число тех или иных микросхем. Следует помнить, что для того, чтобы фильтрующие емкости эффективно работали, длина цепи от вывода микросхемы до емкости должна быть минимальной. Использование планарных компонентов практически полностью позволяет решить эту проблему (рис. 7).

Не следует разносить в пространстве сигнальную и возвратные цепи одного сигнала

Неправильная трассировка сигнальной цепи



Правильное решение

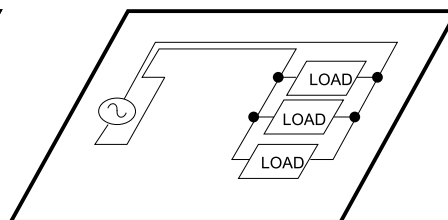


Рис. 8. Разводка сигнальной цепи и цепи возврата

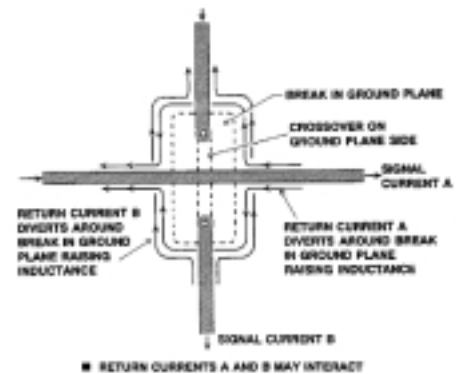


Рис. 9. Взаимодействие возвратных токов на вырезе в заземляющей плоскости

(рис. 8). Это замечание касается также дифференциальных входных цепей, которые должны иметь одинаковую длину.

Следует стремиться выполнить разводку чувствительных аналоговых цепей в одном слое (со стороны установки компонентов) и избегать пересечений проводников, так как нарушение целостности заземляющего слоя вызывает увеличение его индуктивности и, следовательно, возрастает степень взаимного влияния возвратных токов (рис. 9).

При проектировании платы, содержащей скоростные цифровые микросхемы (ПЛИС, сигнальные процессоры и т. п.), следует подумать о теплоотводе. Полезны бывают дополнительные слои для отвода излишнего тепла от микросхем.

Полезно после завершения разработки топологии платы максимально расширить все силовые цепи, земли, ключевые сигнальные цепи.

При разработке ответственных устройств не следует забывать о возможностях современных средств по анализу целостности сигналов, тепловых режимов, прочности и т. д.

### Литература

1. Стешенко В. Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС. // Chip News, 1999, № 8–10, 2000, № 1, 3–5.
2. Стешенко В. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов — М.: «Дододека», 2000
3. Стешенко В. ACCEL EDA: технология проектирования печатных плат. — М.: «Нолидж», 2000. — 512 с., ил.
4. Потемкин И. С. Функциональные узлы цифровой автоматики. М.: Энергоатомиздат, 1986. — 320 с.: ил.
5. Стешенко В. Школа схемотехнического проектирования устройств обработки сигналов. Занятие 1. Алгоритмы, элементная база, системный анализ. // Новые компоненты и технологии, № 3, 2000