

Altium Designer 6.

Новые возможности в версии 6.8

На примерах схем и топологических решений, опубликованных автором в предыдущих номерах журнала, в статье рассматриваются новые возможности пакета Altium Designer 6.8, в том числе показаны новые параметры и их свойства, а также указаны действия, которые нужно производить для достижения требуемого результата.

Владимир Пранович,
к. т. н.

pranovich@bsu.by

Здесь мы не будем рассматривать изменения, которые касаются улучшения графического и визуального представления информации на экране монитора, так как интереса с точки зрения схемотехнических и топологических возможностей они не представляют. Не будем затрагивать и тех изменений (операции выравнивания длин, работа с дифференциальными линиями и т. п.), которые имеют отношение к вопросам, не рассмотренным автором в предыдущих статьях. Эти проблемы мы раскроем в будущем. В данной статье мы расскажем только о тех новшествах, которые легко продемонстрировать с использованием примеров, опубликованных ранее.

Итак, вернемся к проекту, описанному в статье [1], и на его базе отметим наиболее значимые изменения новой версии пакета в той последовательности, в которой они представлены в документе “AR0143 Whats New in Altium Designer 6.8”.

Вставка метафайла

Это одно из замечательных приобретений новой версии. Оно особенно актуально, так как пакет не обладает широким инструментарием, предназначенным для работы со сложным текстом и фигурами, что заметно осложняет подготовку текстовой и гра-

фической информации для печатной платы. Теперь стало значительно проще вставлять текст и простые рисунки, подготовленные в других пакетах, включая MS Office, в файл топологии.

Наберите в текстовом редакторе WORD сложный текст (см. рис.1), включающий различные объекты редактора. Там представлена рамка таблицы, стрелки, простые фигуры, различное форматирование текста. Скопируйте текст и все фигуры в буфер и просто вставьте на нужный слой в файле топологии. На рис. 1, где представлен результат вставки на слой **Note**, указаны:

1. Граница рамки таблицы.
2. Текст.
3. Текст с форматированием (двойное зачеркивание).
4. Список.
5. Простая фигура — двунаправленная стрелка.
6. Сложные фигуры — типа «звезда» и широкая стрелка с текстом.
7. Диаграмма.

Примечание. Не используйте в автофигурах сплошную заливку: вставка происходит на один слой, а заливка конвертируется в полигон со сплошной заливкой, которая закроет все изображение под ним на печатной плате, и его все равно необходимо будет удалить.

Надо сказать, что со сложными фигурами и рисунками могут возникнуть проблемы. Однако и представленные возможности существенно помогут вам в оформлении технической документации.

Параметры инвертированного текста

На рис. 52 [1] был приведен пример нанесения маркировки, где для отображения использовалась инверсия текста. При этом для настройки параметров инверсного текста могла быть использована только одна величина — значение зазора между границей закрашенной области и введенным текстом. Для получения просто закрашенного места приходилось конец текста заполнять пробелами. Это очень неудобно, так как размер закрашенной области определяется не только длиной текста, но и теми символами, которые используются в нем. И при внесении изменения в текст нужно постоянно следить, чтобы надпись не закрывала контактные площадки.

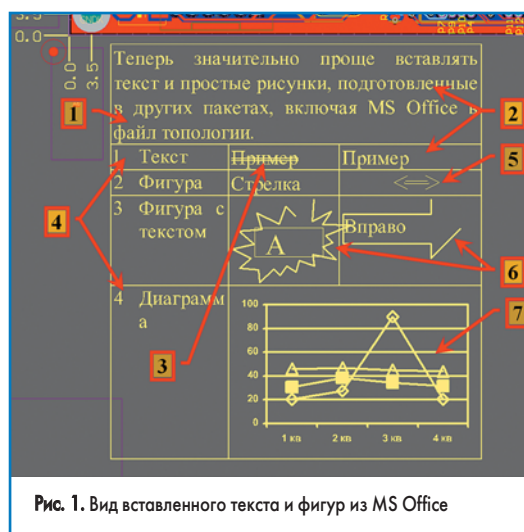


Рис. 1. Вид вставленного текста и фигур из MS Office

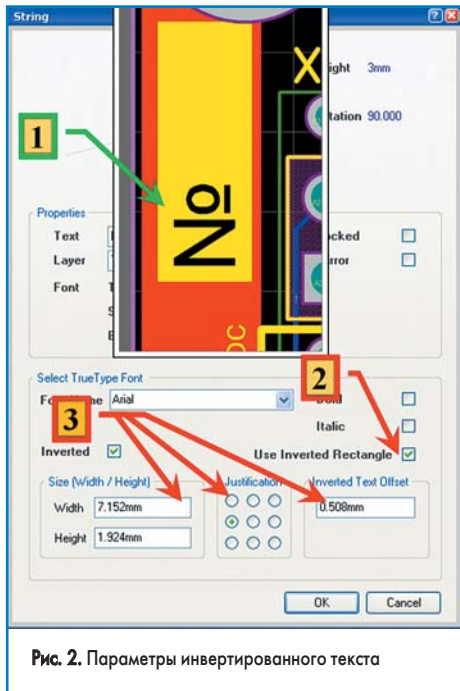


Рис. 2. Параметры инвертированного текста

На рис. 2 представлены прежний и новый вариант представления текстовой надписи. Новый вариант эквивалентен старому по отображению, однако имеет фиксированные размеры. Ниже приведена последовательность действий для получения нового варианта:

1. Вид старой надписи. Параметр **Inverted Border** = 0,508 мм. Именно такой зазор установлен относительно границы текста. Видимый сдвиг надписи вниз обеспечен вводом пробелов в тексте.
2. Флаг **Use Inverted Rectangle**. Это новый параметр. При его установке можно задавать и другие новые параметры, указанные ниже.
3. Параметры текста в **Inverted Rectangle**:
 - **Width, Height** — ширина и высота **Inverted Rectangle**. По умолчанию определяется размерами введенного текста. Если мы хотим оставить размеры закраски прежними, их следует увеличить на удвоенное значение прежнего параметра **Inverted Border** — 0,508 мм;
 - **Justification** — параметр направления «прижатия» текста;
 - **Inverted Text Offset** — параметр зазора между текстом и границей, к которой он будет «прижат». Чтобы оставить внешний вид прежним, следует установить этот параметр равным значению прежнего параметра **Inverted Border** — 0,508 мм.

После данной операции текст надписи можно отредактировать, убрав лишние пробелы. С помощью новых параметров настройки отображения инверсного текста можно размещать надписи на печатной плате с большей гибкостью.

Новые окна задания параметров для отображения печатных плат

В новой версии стало удобнее устанавливать параметры настройки отображения слоев печатной платы, к тому же появились новые функции. На рис. 3 представлен новый вид панелей управления слоями, где указаны основные изменения, а именно:

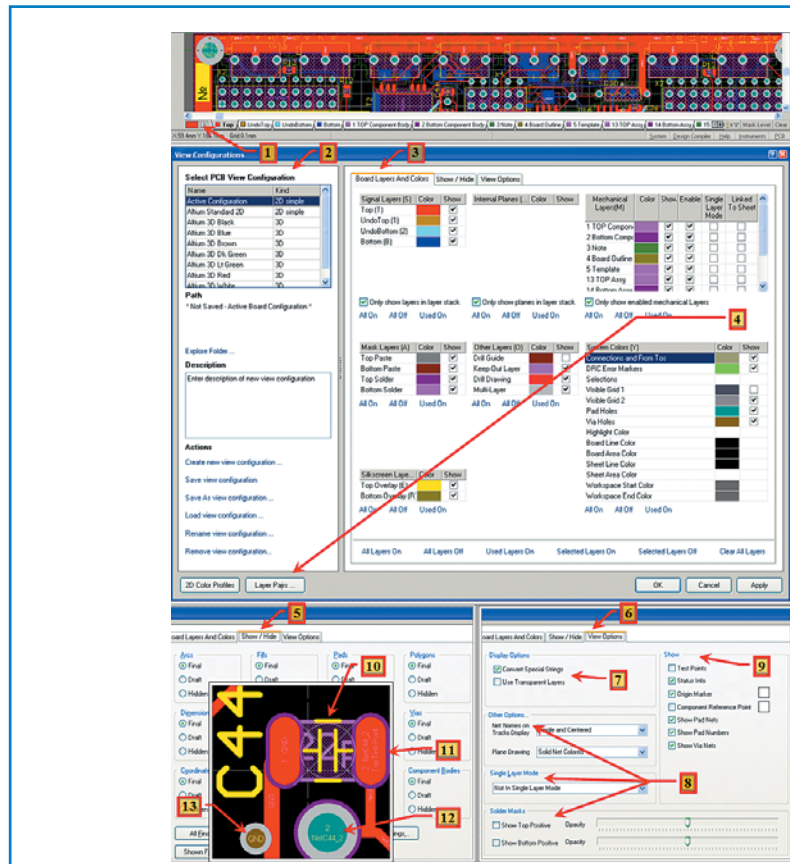


Рис. 3. Окно и вкладка View Configuration

1. Наиболее простой и быстрый вызов окна **View Configuration** можно произвести двойным нажатием на данную метку.
2. В панели появилась новая колонка, в которой приведены стандартные (прежние) настройки вида слоев, и в которую теперь можно добавить пользовательские настройки. Мы не будем рассматривать новые типы настроек. Они, как правило, затрагивают представление вида печатной платы в 3D-отображении, что в старой версии не поддерживалось. Рассмотрим здесь только ту, которая соответствует ранним версиям Altium Designer 6. Это **Active Configuration: 2D simple** (на рисунке она отмечена выделенной строкой).
3. Это вкладка знакомого окна из прежних версий. Здесь нет существенных изменений (разве что там, где на рисунке стоит сноска 4, нет кнопки настройки пар слоев), и мы не будем здесь повторять описание данных настроек.
4. Кнопка назначения пар слоев переместилась на новое место. И это естественно: данные параметры должны быть едины для всех настроек — и стандартных, и пользовательских.
5. Новая вкладка **Show/Hide**. Однако она без изменений пришла из общих настроек **DXP/Preference**. И следует отметить, что теперь выход на установку данных параметров стал более удобным, и более того, данные стандартных и пользовательских настроек можно определить независимо.
6. Новая вкладка **View/Option**. Сюда перешел целый ряд из общих настроек **DXP/Preference**, которые требуют индивидуальных значений параметров для различных настроек из левой колонки (см. пункт 1). Однако и здесь есть новшества.
 7. **Display Option**. Указание опций отображения:
 - специальных текстовых подстановок (текст которых начинается с точки, и в проекте существует параметр с одноименным названием). В зависимости от флага отображается либо сам текст, либо текстовое значение одноименного параметра;
 - установка флага «прозрачности» слоев.
 8. Целый ряд настроек. Затрагивают тип подписи сигналов на элементах топологии и способы отображения слоев типа **Plane**. Вид отображения масок и т. п. Здесь добавлены новые варианты и параметры отображения.
 9. **Show**. Указаны флаги:
 - **Test Point** — отображение подписей тестовых точек (специальных или назначенных из существующих контактных площадок или переходных отверстий);
 - **Status Info** — указание в статусной строке информации о выделенном объекте;
 - **Origin Market** — отображение точки привязки отсчета координат;
 - **Component Reference Point** — это новая возможность, позволяющая отображать точку привязки компонента. Эта функция может быть удобной при проектировании сложных плат;
 - **Show Pad Net, Show Pad Number, Show Via Net** — отображение имен цепей для контактных площадок и переходных отверстий, а также номера контактной площадки.
 10. Для конденсатора C44 указана точка привязки посадочного места — перекрестие с окружностью в центре компонента.

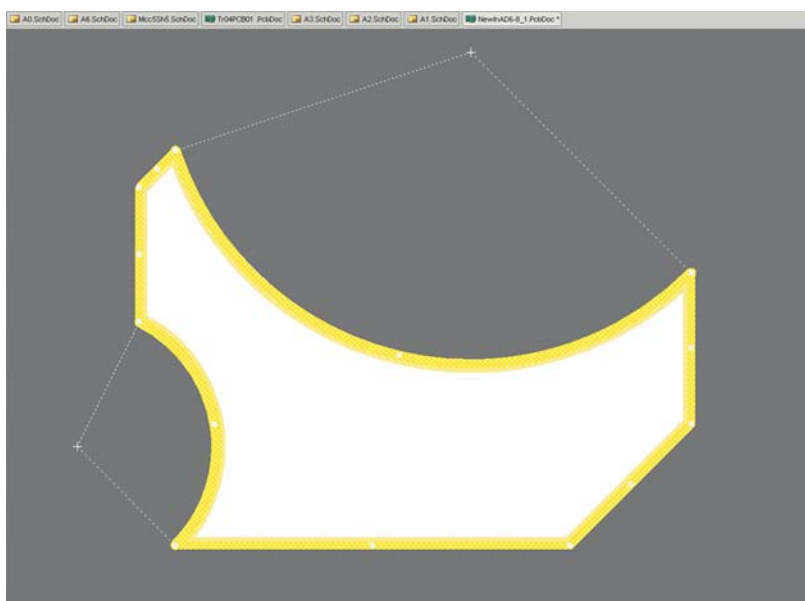


Рис. 4. Создание полигона из выделенных объектов

11. Контактная площадка конденсатора, на которой отображены номер площадки, имя электрической цепи, подпись контрольной площадки.
12. Контактная площадка, которой не присвоен тип контрольной.
13. Переходное отверстие, на котором отображено имя электрической цепи.

Расстановка компонентов

Появилась новая операция при расстановке компонентов: **Tools/Component Placement/Reposition Selected Components**. При этом достаточно сначала последовательно выделить ряд компонентов, затем применить данную команду, и эти компоненты можно расставить в такой же последовательности. Это упростило процесс расстановки, поскольку прежде необходимо было находить и ставить на место каждый элемент.

В то же время данная последовательность, как правило, определяется электрической схемой, и, зная ее, пользователь теперь легко может указать последовательность предварительной расстановки и в топологическом редакторе.

Работа с полигонами

Ранее границы полигона определялись непосредственно при его создании. В случае необходимости редактирования его границы при создании топологии возникали некоторые неудобства в виду сложности операции по коррекции узлов границы полигона. Теперь последняя операция упрощена.

Также добавлена новая операция по созданию контура полигона. Теперь можно нарисовать линиями, дугами и т. п. замкнутый контур, а затем выделить данные объекты и объединить их в контур будущего полигона. Более того, кон-

тур можно сделать в любых других пакетах, затем импортировать и на их основе создать контур границы полигона. Это особенно удобно, так как позволяет создавать сложные формы полигона (например, в логотипе фирмы) из уже имеющегося изображения.

Рассмотрим этот процесс подробнее. Мы не будем производить экспорт из других пакетов, так как он все равно приведет к созданию границ полигона через его контур. Такой контур мы создадим сами (рис. 4).

1. Итак, операцией **Place/Line** или **Place/Arc** создадим замкнутый контур (желтый контур на рисунке из двух полуокружностей, соединенных ломаной линией).
2. Применим операцию **Tools/Polygon Pours/Define from selected objects**.
3. Полигон будет создан (на рисунке он белого цвета). Теперь достаточно выделить его и указать, как обычно, остальные параметры полигона.

Выделение объектов

Дополнительно к прежней функции выделения **Alt + click** (замаскировать все, кроме указанного компонента) добавлены еще:

- **Ctrl + double click** — выделить все электрические связи, принадлежащие данному классу (рис. 5);
- **Alt + double click** — выделить все посадочные места, принадлежащие данному классу.

Жгут (шина) разнородных электрических цепей

Это одно из самых значимых новшеств, и потому рассмотрим его подробнее, так как в исходном документе ему уделено недостаточно внимания. Итак, на схеме появились три новых типа элементов. Эти элементы и их главные свойства показаны на рис. 6:

1. **Harness Connector** — элемент для объединения разнородных электрических сигналов, включая шины, в один жгут.
2. **Signal Harness** — изображение жгута на схеме с разнородными электрическими сигналами. На рисунке данному жгуту присвоено имя SPI с помощью метки **NetLabel**.
3. **Harness Entry** — вводы **Harness Connector**, назначение и свойства схожи с **Sheet Entry**.
4. **Harness Definitions** (на рисунке не показаны) — текстовые определения **Harness Connector**. Хранятся отдельными файлами

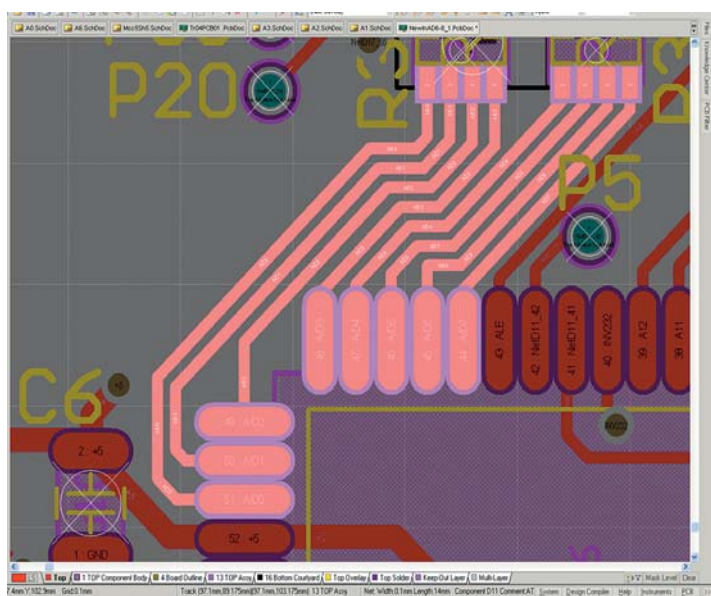


Рис. 5. Выделение класса цепей

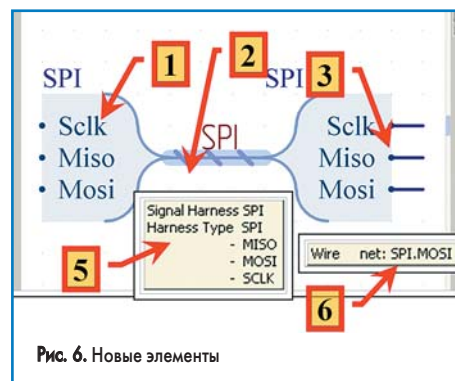


Рис. 6. Новые элементы

в проекте, в соответствии с названием **Harness Connector**.

- При наведении указателя на **Signal Harness** во всплывающей подсказке окажутся:
 - имя жгута **Signal Harness**;
 - название соединителя **Harness Type**;
 - список вводов **Harness Entry**.
- При наведении указателя на электрическую связь (на рисунке вывод от **Harness Entry=Mosi**) указывается имя цепи — **SPI.MOSI**. То есть в данном примере имена цепям по умолчанию (если имя не дано иным способом) присваиваются по названию жгута (**Signal Harness**) и названию ввода, которые разделены точкой.

Harness Connector

Рассмотрим подробней работу с **Harness Connector** (рис. 7).

Окно **Harness Connector** содержит две вкладки:

- Вкладка **Harness Entry** — список вводов. Здесь вы можете добавлять и редактировать вводы.
- Вкладка **Property** — настройка внешнего вида и других свойств **Harness Connector**.
- Обратите внимание: чтобы скрыть название **Harness Connector** (как правило, это ненужная информация, загромождающая схему), следует установить флаг **Hide Harness Type**.

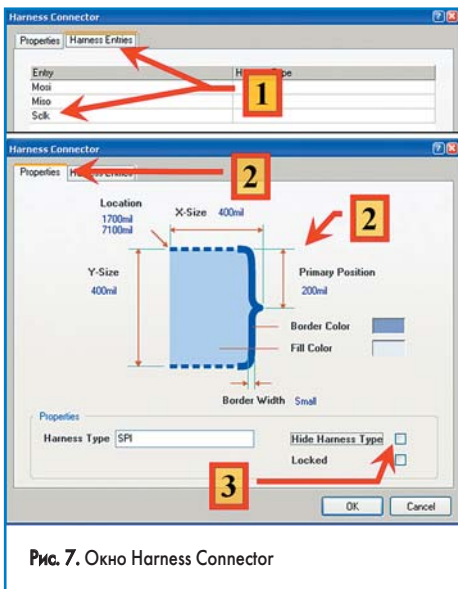


Рис. 7. Окно Harness Connector

Harness Entry

Harness Entry имеет одну особенность, на которой остановимся подробней (рис. 8). В настройках можно указать полное имя **Harness Entry** или сокращенное (для шин сигналов):

- Выбор вида отображения.
- Вид схемы с полным изображением названия **Harness Entry**.
- Вид схемы с кратким изображением названия **Harness Entry**.

Применение Signal Harness

На рис. 9 представлены типовые способы использования **Harness**.

- Подключение жгута (**Signal Harness**) к вводам (**Sheet Entry**) ссылки (**Sheet Symbol**) на лист. Имена **Sheet Entry** присваиваются

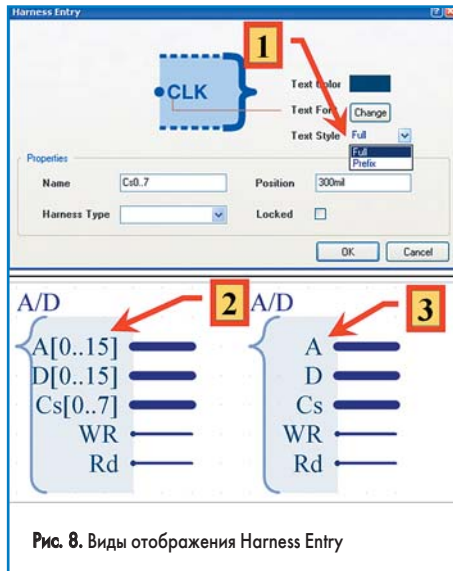


Рис. 8. Виды отображения Harness Entry

также, как и обычным электрическим цепям. Цвет отображения **Sheet Entry** при подключении к **Signal Harness** автоматически изменится и станет таким, как установлено по умолчанию в разделе **Preference** для такого типа подключений.

- В местах пересечения **Signal Harness** с одинаковыми именами образуется **DOT** (утолщение).
- Имя локального (принадлежащего только данному листу схемы) **Signal Harness**.
- Такое же имя (**SPI_2**). Однако заметьте: оно не обязано совпадать с именем (**SPI**) **Harness Entry**, к которому он подключен.
- Присвоение имени **Signal Harness** с помощью **Port**. Такой способ применяется для соединений жгута с идентичными жгутами на других листах схемы, или при передаче к подчиненному листу в иерархических проектах. Цвет **Port** при подключении к **Signal Harness** автоматически изменяется на значение, установленное по умолчанию в разделе **Preference**.
- Показаны имена цепей, присвоенные в этом примере:
 - Сигнал на выводе «**Harness Entry = RD**», который принадлежит «**Harness Connector=A/D**». Имя сигнала присвоено по имени «**Signal Harness= A/D**», к которому под-

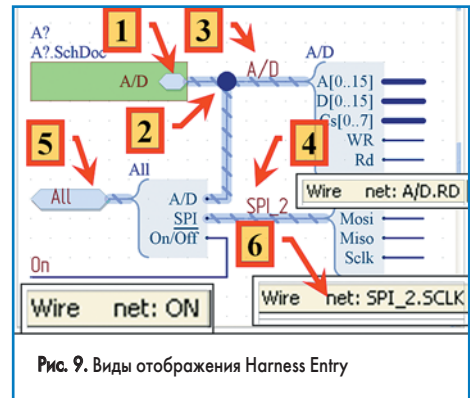


Рис. 9. Виды отображения Harness Entry

ключен **Harness Connector**, и имени **Harness Entry** и имеет значение «**A/D.RD**».

- Аналогично присвоено имя сигналу «**SPI_2.Sclk**». Заметьте: имя «**Harness Entry = SPI**», к которому подключен «**Signal Harness SPI_2**», не участвует в процедуре присвоения названия.
- Цепь, обозначенная «**NetLabel=On**», имеет такое же имя, однако в жгуте она будет связана по названию соответствующего **Harness Entry**. Это не должно нас смущать. Целостность передачи электрических связей здесь обеспечивается так же, как и при использовании в **Sheet Entry** в иерархических проектах.

Примеры использования Harness в проектах

Обратимся к схемам, описанным в статье [1], и на их примере покажем использование **Signal Harness** и их особенности в конкретном применении.

Схема АЦП

На рис. 2 [1] была представлена схема аналого-цифрового преобразователя на основе ADS1255. Здесь и далее в примерах не следует обращать внимание на частичное несовпадение внешнего вида схем. Все они модифицированы для журнальной статьи.

- Итак, с учетом применения **Signal Harness**, схема представлена в следующем виде (рис. 10):
 - На схему добавлен новый **Port** (с именем **logic8k**) и **Signal Harness**. Данный порт за-

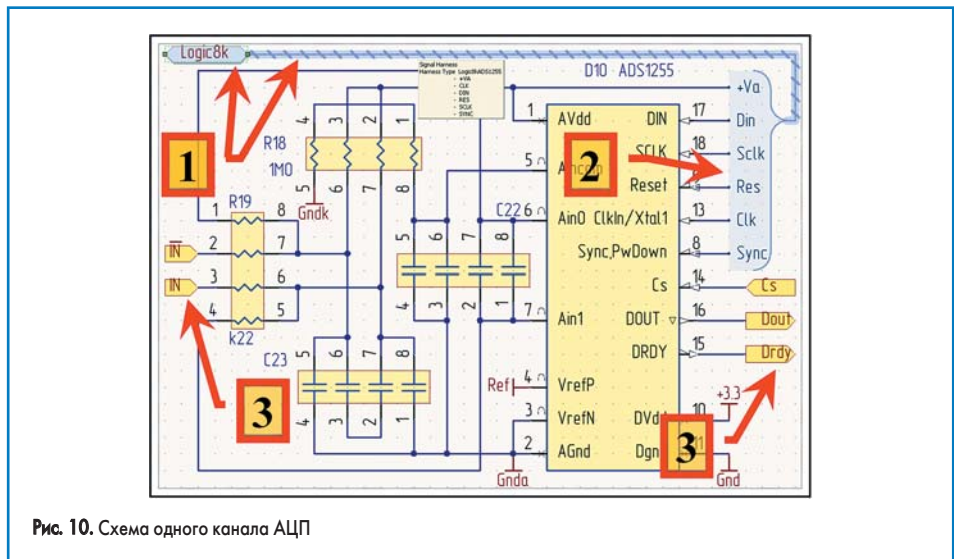


Рис. 10. Схема одного канала АЦП

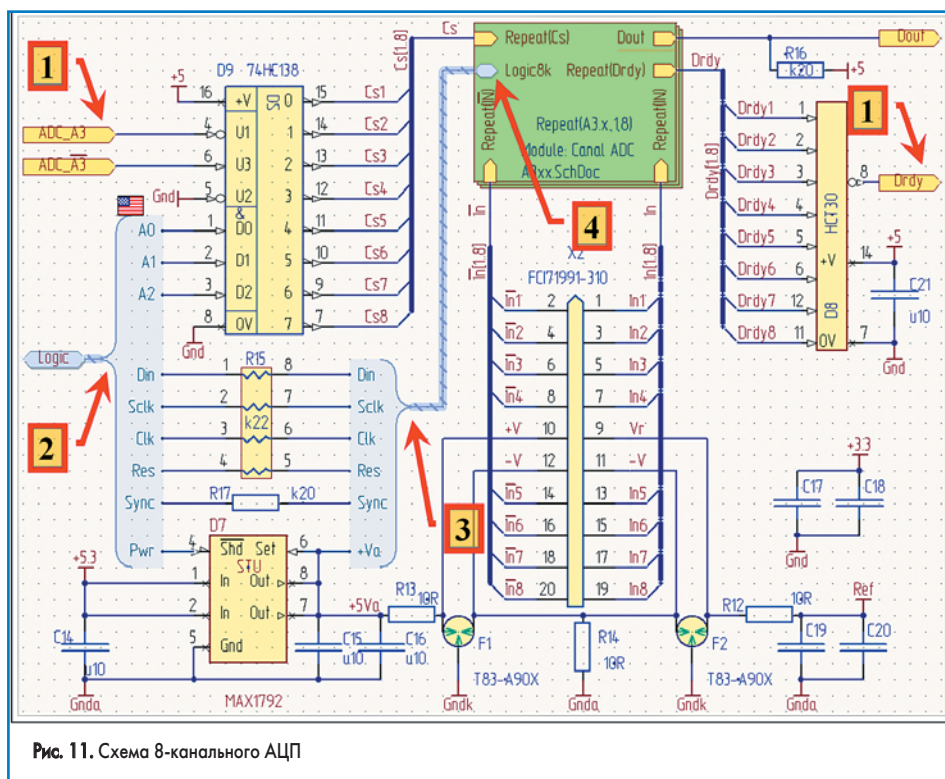


Рис. 11. Схема 8-канального АЦП

меняет все **Port** из прототипа, которые войдут в **Harness Connector**.

- Harness Connector** объединяет все входные сигналы АЦП, которые далее будут общими для всех каналов АЦП.
- Port**, которые есть и в прототипе и которые относятся к каждому каналу АЦП индивидуально.

Схема очень проста, и ее преимущества пока не очевидны, хотя можно отметить следующее:

- количество **Port** сократилось, и это слегка облегчает ориентацию в схеме;
- теперь положение **Port** более соответствует требованиям ГОСТ — **Port**, относящиеся

к входным цепям, находятся слева, а к выходным — справа.

Схема 8-канального АЦП

На рис. 24 [1] была представлена схема 8-канального аналого-цифрового преобразователя с примененной иерархии. На рис. 11 представлен ее новый вариант, где указаны:

- Port**, которые есть и в прототипе и которые относятся к адресации 8-канальных АЦП для более сложных схем.
- Новый **Port** (с именем **logic**), **Signal Harness** и **Harness Connector**. Они заменяют все **Port** из прототипа, которые относятся к входным цепям.

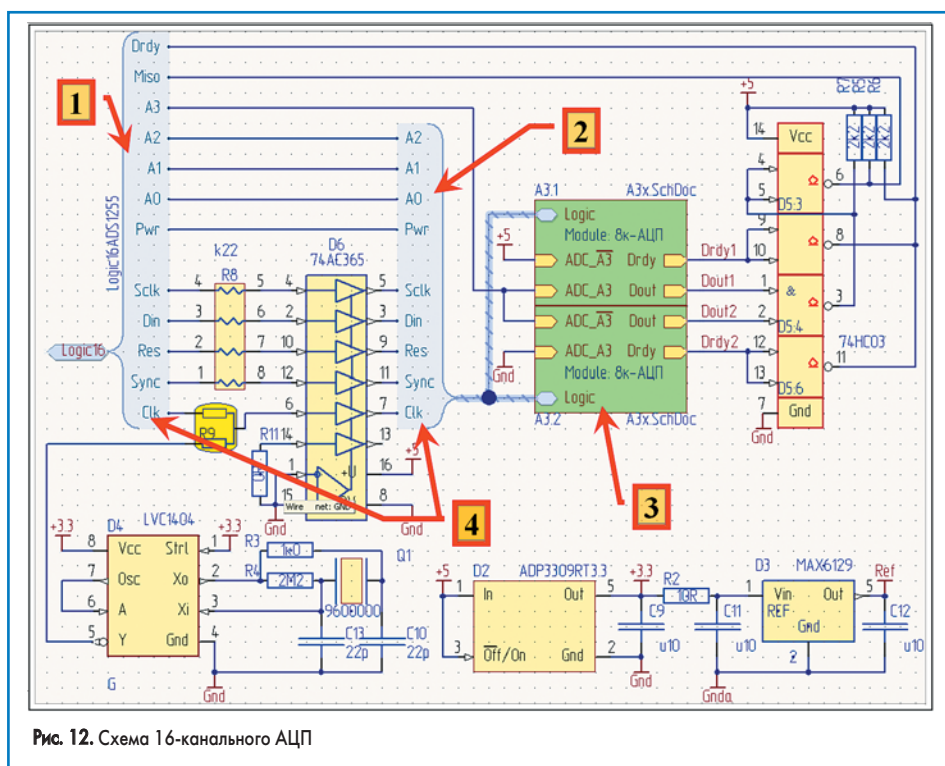


Рис. 12. Схема 16-канального АЦП

3. Новые **Signal Harness** и **Harness Connector**.

Они заменяют внутренние сигналы на схеме. Используются как для обеспечения удобства расположения элементов схемы, так и для связи сигналов с подчиненным листом через **Sheet Entry** ссылки **Sheet Symbol**.

- Sheet Entry**. Заметьте, через данный **Sheet Entry** производится связь всех 6 сигналов, подключенных через **Harness Connector**. Отметим следующее:

- как и в предыдущем случае, количество **Port** сократилось, а схема более соответствует требованиям ГОСТ;
- использование внутреннего **Signal Harness** (ссылка 3 рис. 11) существенно облегчает выбор места для элементов схемы.

Схема 16-канального АЦП

Перейдем теперь к схеме 16-канального АЦП, прототип которого можно найти на рис. 35 и 36 [1]. Как видно на данных рисунках, в прототипе схема «перегружена» дублирующими связями, а подписями к ним, необходимыми для создания сложной иерархии.

Теперь обратимся к новому варианту данного листа схемы, который представлен на рис. 12. На рисунке указаны:

- Новый **Port** (с именем **logic16**), **Harness Connector**. Они заменяют все **Port** из прототипа, которые относятся к входным и выходным цепям. Заметьте, на схеме отсутствует **Signal Harness**, а **Port** подключен непосредственно к **Harness Connector**. Это также разрешено и позволяет сократить место на схеме.
- Новые **Signal Harness** и **Harness Connector**. Они заменяют внутренние сигналы на схеме. Используются как для удобства расположения элементов схемы, так и для связи сигналов с подчиненным листом через **Sheet Entry** ссылки **Sheet Symbol**. Заметьте, что их использование именно на данном листе существенно видоизменило и упростило изображение схемы.

- Два **Sheet Entry**, идентичные прототипу. Заметьте, через данные **Sheet Entry** производится связь всех 9 сигналов, подключенных через **Harness Connector**.

- Обратите внимание: разные **Harness Connector** имеют идентичные по названию **Entry Harness**. Однако в зависимости от подключения эти выходы могут объединять как идентичные, так и различные электрические сигналы.

Отметим следующее:

- как и в предыдущих случаях, количество **Port** сократилось, а схема более соответствует требованиям ГОСТ;
- существенно сократилась площадь, занимаемая **Sheet Entry**;
- сильно сократилось место, занимаемое изображением подвода электрических связей к **Sheet Symbol**;
- значительно улучшилось изображение схемы из-за значительного сокращения множества пересекающихся линий связи;
- сократилась площадь, занимаемая схемой;
- и самое важное — несомненно улучшилась читаемость электрической схемы.

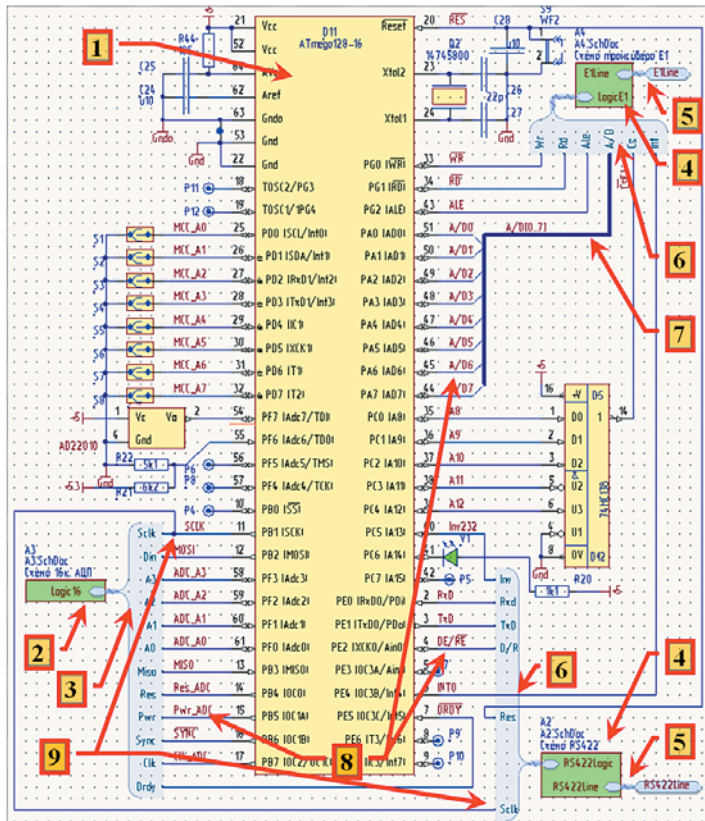


Рис. 13. Схема микроконтроллера

Схема микроконтроллера

Наконец, мы можем полностью представить и схему микроконтроллера даже для журнального варианта. В прототипе [1] дана только краткая ссылка на нее. Сама же схема из-за сложности была представлена отдельными фрагментами. Итак, обратимся к рис. 13, где представлены:

1. Непосредственно изображение схемы микроконтроллера.
2. Ссылка **Sheet Symbol** на схему 16-канального АЦП. Размеры **Sheet Symbol** минимизированы, так как он теперь имеет только один **Sheet Entry** вместо прежних двенадцати.
3. **Signal Harness** и **Harness Connector**. Здесь они используются в обычном назначении. Порядок размещения **Entry Harness** в **Harness Connector** и размер последнего выбирается с учетом удобства подключения связей к микроконтроллеру.
4. **Sheet Symbol** на подчиненные листы схем трансивера и интерфейса. Схемы будут представлены ниже.
5. **Port** и **Signal Harness**. Используются для связи с подчиненными листами схемы трансивера и интерфейса.
6. **Signal Harness** и **Harness Connector**. Здесь они используются в обычном назначении, как и в пункте 3.
7. Обратите внимание: здесь к **Harness Connector** подключены не только одиночные связи, но и шина. При этом в свойствах соответствующего **Entry Harness** указано отображение только префикса (имени шины, без указания ее пределов).
8. В данном примере все имена цепей, передаваемые через **Signal Harness**, определяются выводами микроконтроллера и соответ-

венно имеют метки **NetLabel**. Хотя этим электрическим цепям можно и не давать специальные имена.

9. Обратите особое внимание: один и тот же сигнал может поступать на **Entry Harness** (с одинаковыми или не совпадающими именами) разных **Harness Connector**.

Итак, применение этих новшеств Altium Designer 6 позволило на данном примере схемы микроконтроллера существенно улучшить читаемость схемы. Надеемся, именно это нововведение будет незамедлительно задействовано пользователями Altium Designer 6.

Схема интерфейса

Рассмотрим теперь схему интерфейса (рис. 14):

1. **Port**, **Signal Harness** и **Harness Connector**. Здесь они используются в обычном назначении (как и на предыдущих листах) и интереса не представляют.
2. Здесь применен способ, который упрощает работу с длинными связями на схеме. Стрелками указаны два одинаковых **Entry Harness** с именем **TxD** в одном **Harness Connector**. Соответственно, нет необходимости вести линию связи между двумя входами разных микросхем. Связь обеспечивается за счет объединения через **Harness Connector**.

Схема трансивера и первый лист схемы проекта ничего нового в плане использования **Harness** и **Harness Connector** не содержит.

Итак, в результате использования данного нововведения, все схемы существенно упростились, и самое главное, процесс проектирования их стал намного легче. Нам кажется, именно это новшество является наиболее кардинальным, и оно предоставляет новые под-

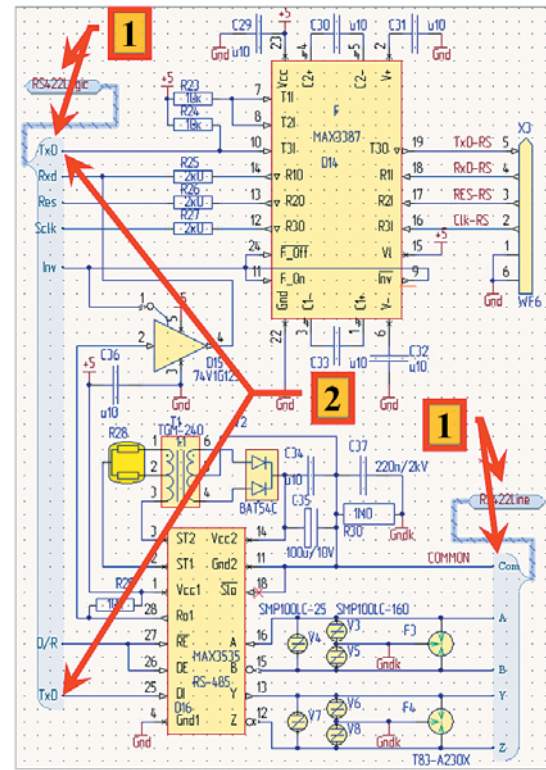


Рис. 14. Схема интерфейса

ходы в проектировании схемотехнической части.

Схема модуля (Device Sheets)

Теперь можно подключать к проекту ранее реализованные и проверенные схемы проектов. Покажем это на примере. Будем считать, что схема канала АЦП у нас отработана, и подключим ее дополнительно на первый лист в качестве демонстрации возможности.

Первым делом файл схемы следует сохранить в специальную папку, например, **.../Altium/Device/SCHDoc/** с именем **ADC1255.SchDoc**. Теперь поступим следующим образом (рис. 15):

1. Выполним команду **Place Device Sheet Symbol**.

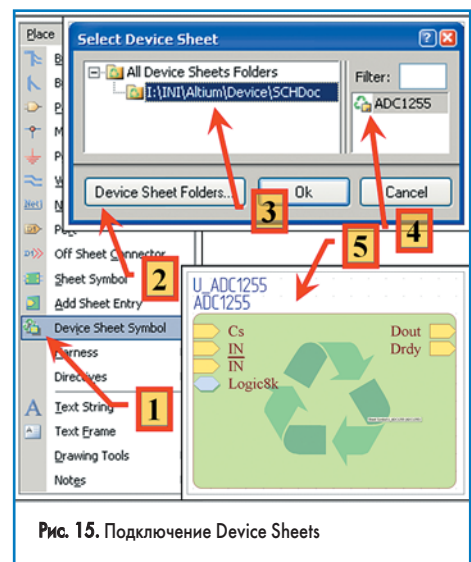


Рис. 15. Подключение Device Sheets

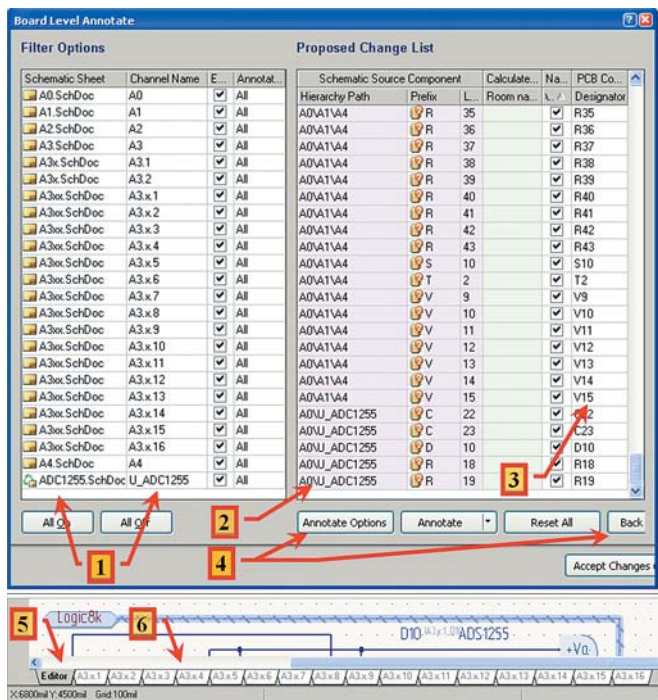


Рис. 16. Окно Board Level Annotate

2. В открывшемся окне нажмем кнопку **Device Sheet Folder**.
3. Добавим ссылку `.../Altium/Device/SCHDoc` на специальную папку.
4. Выберем из списка ссылку на файл с требуемой схемой.
5. Разместим на листе схемы **Device Sheet Symbol**.

Device Sheet Symbol это тоже что и **Sheet Symbol**. Единственное отличие — схема недоступна для редактирования. Это естественно, так как она в неизменном виде может использоваться и в других проектах.

Отметим также, что настройку вышеуказанных специальных папок можно провести и через команду `DXP/Preferences/Schematic/Device Sheets`.

Новый вид автонумерации

Добавлен новый тип (**Board Level Annotate**) автонумерации компонентов схемы. Вызов производится командой `Tolls/Board Level Annotate`. На рис. 16 представлено окно **Board Level Annotate**, где отмечены новые возможности нумерации.

1. В первом столбце указан список имен файлов листов схем, во втором — номер канала. Заметьте: так как схема иерархическая, более того, с повторяющимися блоками, в данном столбце указаны все каналы проекта. Соответственно, появляется возможность нумерации одного из каналов. К тому же, в последней строке указана ссылка на подключенный выше в разделе файл `ADC1255.SchDoc`. Обычный вызов автонумерации не дает такой возможности.
2. В этом столбце указана последовательность иерархии. Назначение информации в остальных столбцах соответствует их названию.

3. Здесь указан столбец с окончательным видом обозначения компонентов.
4. И, наконец, добавлены кнопка для быстрого вызова настроек проекта по способу присвоения обозначений компонентов и кнопка для обратной аннотации из файла топологии печатной платы.
5. Теперь, после компиляции проекта и открытия листа канала АЦП (см. также рис. 10), первой открывается вкладка **Editor** схемы. В ней можно редактировать все элементы схемы.
6. Для просмотра схемы конкретного канала АЦП следует открыть соответствующую вкладку справа. При этом все обозначения компонентов будут менять свое значение на то, которое присвоено в результате нумерации для данного канала.

Параметры Sheet Entry

Добавлена возможность редактирования шрифта **Sheet Entry**. К сожалению, это не коснулось ни **Port**, ни **Pin** для компонентов.

Параметры Schematic Library

Теперь появилась возможность отображения параметров **Designer** и **Comment** в библиотечном редакторе (рис. 17). Раньше всегда возникали проблемы как с установкой месторасположения данных параметров, так и с выбором шрифта для их отображения. Раньше выход на установку данных параметров был не очевиден. Сейчас свойства этих параметров легко установить, а сами параметры отображаются и в редакторе библиотек.

Формирование PDF-документа

Данная функция стала доступна и через **Job**-файл. Это значительно упрощает подготовку

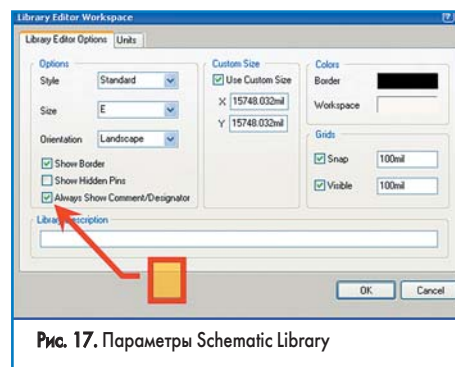


Рис. 17. Параметры Schematic Library

документов, поскольку через **Job**-файл можно настроить разные варианты данных документов с различным наполнением информацией.

Защита компонентов на схеме

Добавлена возможность защиты объектов от перемещения и выделения и на схеме.

К сожалению, данную защиту можно изменить только через инспектор (рис. 18).

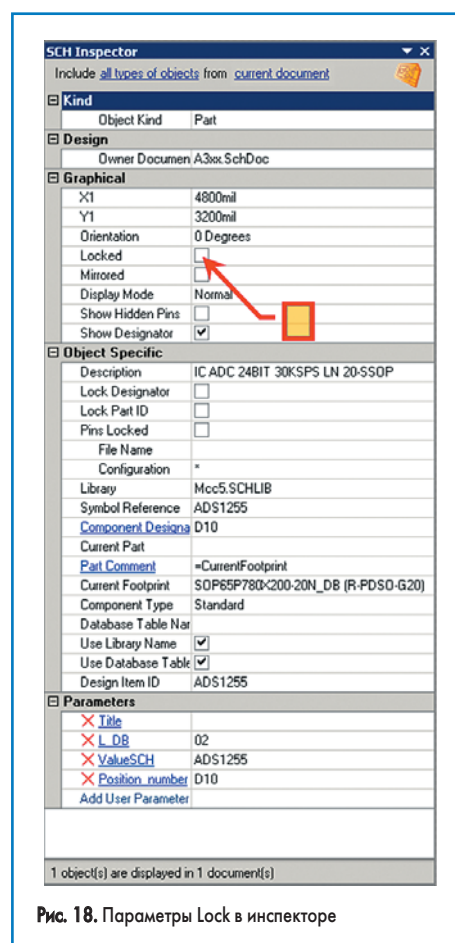


Рис. 18. Параметры Lock в инспекторе

Остальные нововведения не затрагивают схемный и топологический редакторы пакета, и мы их приводить не будем.

Надеемся, что представленная демонстрация возможностей новой версии пакета проектирования позволит быстрее начать их практическое применение в ваших проектах.

Литература

1. Пранович В. Altium Designer 6 в примерах // Технологии в электронной промышленности. 2007. № 5, 6, 7, 8.