

Инструменты проектирования быстродействующих печатных плат на примере САПР Altium Designer Часть 2

Александр Фень (Москва)

Статья продолжает описание инструментов САПР Altium Designer 16.0 для решения задачи синхронизации сигналов при проектировании топологии печатных плат. Описывается xSignal-мастер, используемый для формирования групп xSignals. Также даётся представление о расчёте длины маршрута сигнала и расчёте суммарной длины трассы маршрута сигнала в САПР Altium Designer 16.

xSIGNAL-МАСТЕР

Для формирования групп xSignals целесообразнее применять xSignal Wizard (xSignals-мастер, далее Мастер). Он доступен по командам меню Design > xSignals > Run xSignals Wizard и в пошаговом режиме позволяет определить группы сигналов, объединить их в классы и назначить правила проектирования. В Altium Designer 16.0 в xSignals Wizard доступны два режима:

1. Custom Multi-Component Interconnect – определяемое пользователем множество компонентов для формирования xSignals. Применение этого режима описано на сайте разработчика [1].
2. On-Board DDR3/DDR4 – формирование xSignals для интерфейса памяти DDR3/DDR4. Этот режим был введён одним из первых в дополнение к Custom Multi-Component

Interconnect, так как интерфейс является наиболее трудоёмким с точки зрения разнообразия xSignals.

Рассмотрим формирование xSignals для DDR3/DDR4. После запуска Мастера на первом шаге выбираем режим работы On-Board DDR3/DDR4 (см. рис. 7) для создания маршрутов xSignals для интерфейсов памяти DDR3/DDR4. В этом режиме Мастер автоматически создаёт маршруты xSignals, классы маршрутов xSignals (xSignals Classes), правила для группы соответствующих по длине цепей (Matched Length Groups), правила для группы соответствующих по длине дифференциальных пар (Diff Pair Matched Lengths). Также он устанавливает тип топологии Fly-By (предполагается, что при трассировке будет использоваться эта топология) для интерфейсов DDR3/DDR4, присутствующих на плате.

В диалоговом окне содержатся настройки разрядности шины данных для каждой из линий. Также содержатся настройки допусков для правила соответствия длин цепей (Matched Net Length), создаваемых для:

- адресных / командных / управляющих цепей;
- шин данных;
- тактовых линий.

В следующем диалоговом окне Мастер идентифицирует все возможные компоненты-источники (Source) и компоненты-приёмники (Target) сигнала (см. рис. 8), исходя из префиксов указателя и количества выводов. Здесь требуется:

- Выбрать префикс Контроллера (Controller) и Модуля памяти (Memory Devices). Для фильтрации компонентов необходимо указать минимальное количество выводов (Min Pin Count) как для источника (Source Component), так и для приёмников (Target Components).
- Выбрать один компонент-источник.
- Выбрать один или несколько компонентов-приёмников.

Теперь можно перейти к идентификации всех цепей, входящих в адресную шину (см. рис. 9). Последовательность действий следующая:

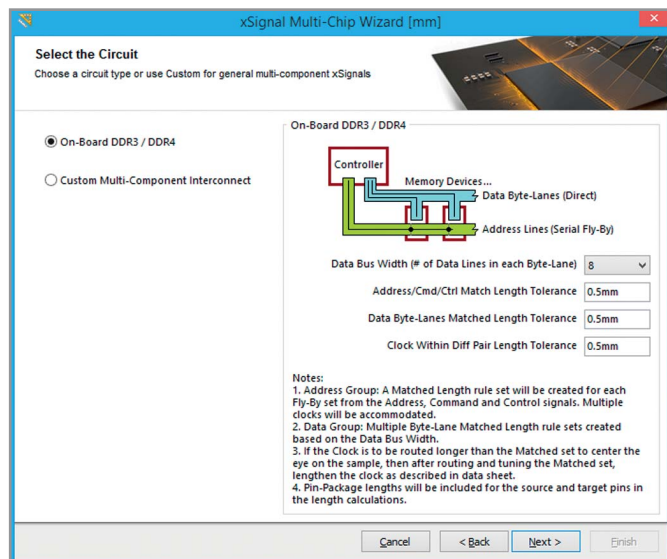


Рис. 7. Начальное окно xSignal Wizard

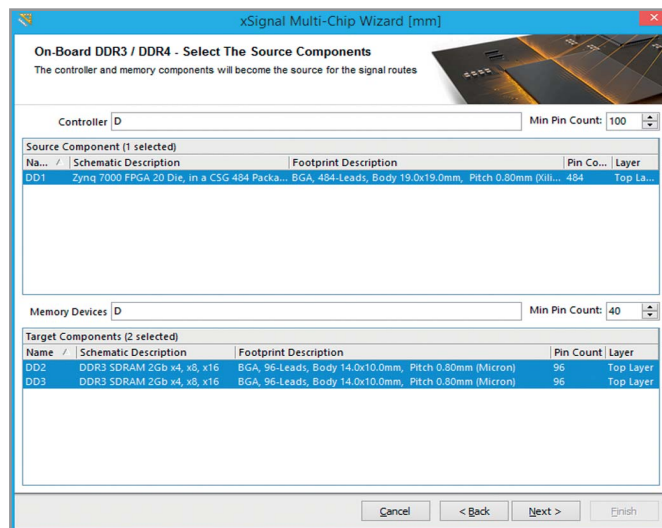


Рис. 8. Идентификация всех возможных исходных и конечных компонентов в xSignal Wizard

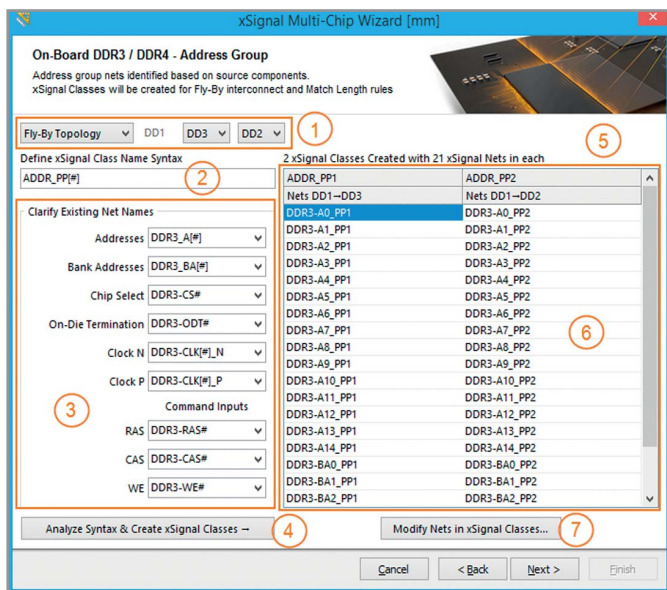


Рис. 9. Идентификация всех цепей, входящих в адресную шину в xSignal Wizard:

Обозначения: 1 – поле выбора топологии; 2 – поле задания имени класса; 3 – маска определения имён цепей шины адреса; 4 – кнопка запуска анализа синтаксиса и создания xSignal классов; 5 – количество классов и маршрутов; 6 – перечень классов и маршрутов xSignal; 7 – кнопка для ручного изменения цепей и классов маршрутов.

1. Необходимо выбрать требуемую топологию из выпадающего списка. Поддерживаются варианты топологии Fly-By Topology и T-Branch Topology. Если выбран вариант топологии Fly-By, то компоненты-приёмники входят в заданный диапазон обозначений (Point-to-Point) и программа попытается установить порядок компонентов-приёмников автоматически. Если Мастер запускается до того как была произведена расстановка компонентов, то порядок устройств должен быть указан вручную, с помощью выпадающих списков. Если выбран вариант топологии T-Branch, то половина компонентов-приёмников должна быть показана до компонента-источника, а половина – после. В данном случае можно использовать выпадающие списки (см. рис. 10) для установки требуемого порядка компонентов-приёмников.
2. Пользователем задаётся имя класса. По умолчанию используется имя ADDR_PP[#]. Символ [#] отражает количество устройств памяти. При необходимости суффикс PP может быть изменён.
3. Мастер анализирует компоненты, смотрит на их суффиксы и отображает синтаксис имени полностью на основании приведённых далее принципов:

- Выполняется автоматический поиск цепей, соответствующих указанным функциям. Как только подходящая цепь будет обнаружена, синтаксис её имени вводится в соответствующее поле.
- После этого цепи между компонентами пересматриваются, и, как только будет найден суффикс, происходит идентификация префикса. Например, для определения адресных линий Мастер ищет имя цепи следующего формата `_A[#]`.
- Если не удаётся обнаружить цепь с суффиксом, начинающимся на символ подчёркивания «_» и содержащим далее цифры, то производится поиск простого текста после символа подчёркивания «_». Также проверяется использование в качестве разделителей дефиса или точки.
- Если не удаётся в автоматическом режиме определить синтаксис, то эти поля должны быть заполнены проектировщиком. Чтобы удобнее было указывать существующие цепи, можно воспользоваться выпадающими списками.
- 4. Как только порядок устройств и синтаксис имён определён, необходимо нажать кнопку Analyze Syntax and Create xSignal Classes (см. рис. 9) для создания списка маршрутов xSignal. Мастер просмотрит синтаксис и то,

- как соединены компоненты, а также сформирует перечень классов маршрутов xSignal (xSignal Classes). Количество создаваемых классов совпадает с количеством модулей памяти.
- 5. Определяется количество создаваемых классов маршрутов xSignal (на рисунке 9 – два класса) и количество маршрутов xSignal в каждом классе (на рисунке 9 их 21).
- 6. Маршруты xSignal сгруппированы в столбцы по количеству классов, по одному столбцу на каждый класс. Подписи DD1-DD3 и DD1-DD2 в подзаголовке указывают на то, между какими устройствами показаны в данном столбце маршруты xSignal.
- 7. Если автоматически сгенерированный список неполный или неточный, можно нажать кнопку для ручного добавления цепей в класс или удаления из класса. Однако необходимо учесть, что, если повторно нажать на кнопку Analyze Syntax and Create xSignal Classes, то все изменения, сделанные вручную, будут утрачены.

Идентификация шины данных

Последний этап работы Мастера – это идентификация всех цепей, входящих в шину данных – Data Group Nets (см. рис. 11). Последовательность

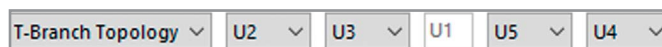


Рис. 10. Выпадающие списки для установки требуемого порядка конечных устройств в топологии T-Branch

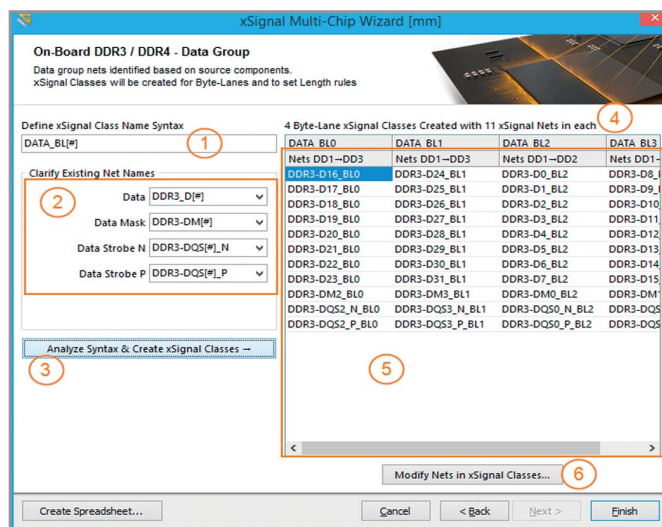


Рис. 11. Идентификация всех цепей, входящих в шину данных в xSignal Wizard

Обозначения: 1 – поле определения синтаксиса имени класса маршрутов xSignal; 2 – маска определения имён цепей шины данных; 3 – кнопка запуска анализа синтаксиса и создания xSignal классов; 4 – количество классов и маршрутов; 5 – перечень классов и маршрутов xSignal; 6 – кнопка для ручного изменения цепей и классов маршрутов.

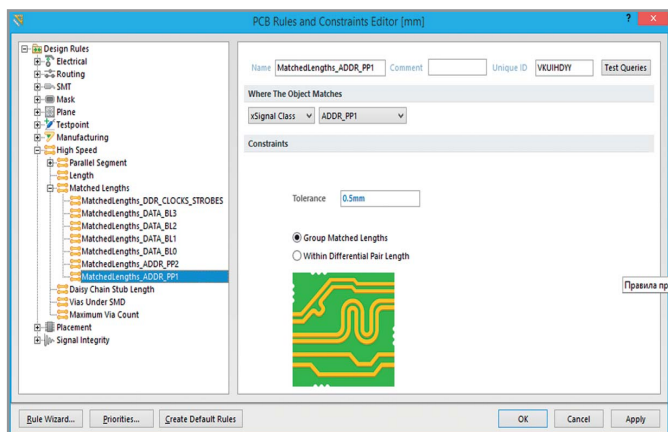


Рис. 12. Вид правила проектирования Matched Length в редакторе правил

действий в этом диалоговом окне следующая:

1. Определение синтаксиса имени класса маршрутов xSignal. По умолчанию используется имя ADDR_BL[#]. Символ [#] отражает количество шин данных, которое определяется как общее количество линий данных, делённое на ширину шины данных, указанную ранее. При необходимости суффикс BL может быть изменён.
2. Мастер анализирует компоненты и просматривает их суффиксы в проекте, затем отображает синтаксис полного имени. Их можно поправить с помощью выпадающих списков, если они определены неточно.
3. Как только синтаксис имён определён, нужно нажать кнопку Analyze Syntax and Create xSignal Classes (см. рис. 11) для создания списка маршрутов xSignal. Мастер маршрутов xSignal просмотрит синтаксис и то, как соединены компоненты, а также сформирует перечень классов маршрутов xSignal (xSignal Classes). Количество создаваемых классов совпадает с количеством шин данных, соединённых с модулями памяти.
4. Определяется количество создаваемых классов и количество маршрутов xSignal в каждом классе. На рисунке 11 их четыре и одиннадцать соответственно.
5. Для созданных классов маршрутов xSignal создаётся правило проектирования Matched Lengths (Совпадение длин). Подписи DD1-DD3 и DD1-DD2 в подзаголовке показывают маршруты xSignal, между какими устройствами (начальным и конечным) они проложены.
6. Для создания таблицы в формате XLS со списком маршрутов xSignal, созданных Мастером, нужно нажать кнопку Create Spreadsheet.

По итогам работы Мастера автоматически создаются следующие маршруты xSignal и классы маршрутов xSignal Classes:

- *Маршруты xSignal для адресных линий, описанных на странице Address Group.* В примере создано два класса маршрутов xSignal, каждый из которых содержит по 21 маршруту xSignal (15 адресных линий плюс выбор устройства, тактирующая дифференциальная пара, разрешение и стробирующая цепь, как видно на рисунке 11).
- *Маршруты xSignal для линий данных, описанных на странице Data Group.* В примере создано четыре класса маршрутов xSignal, каждый из которых содержит по 11 маршрутов xSignal (как видно на рисунке 11, 8 линий данных, маска, стробирующая дифференциальная пара).

В процессе работы Мастер может автоматически создать правила Matched Lengths (Совпадение длин). Например, когда каждый из классов маршрутов xSignal создан Мастером и идентифицирован. В правилах для Matched Lengths используется ограничение Group Match Lengths (Совпадение длин в группе). То есть длина каждого маршрута сравнивается с длиной самого большого маршрута в группе. Расхождение между ними должно быть не более допуска (Tolerance), указанного для этого правила, только тогда правило считается выполненным.

Мастер создаёт правила Matched Lengths и для всех дифференциальных пар, входящих в созданные Мастером классы маршрутов xSignals. Например, на рисунке 11 в создаваемые классы маршрутов xSignals добавляются дифференциальные пары тактирующих импульсов для адресных сигналов и стробирующих импульсов линий данных. В этих правилах использу-

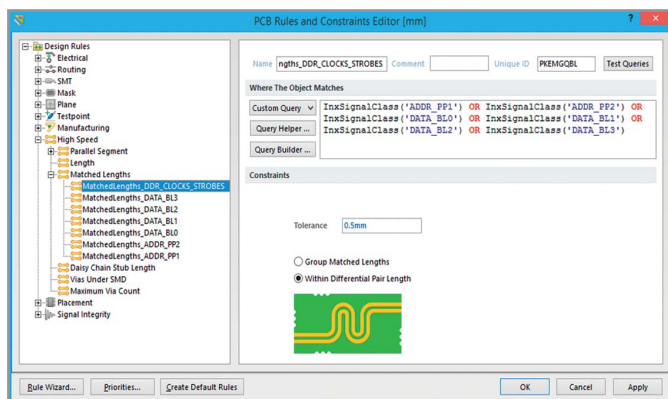


Рис. 13. Вид правила Matched Length для дифференциальных пар в редакторе правил

ется ограничение Within Differential Pair Length (Совпадение длин внутри дифференциальной пары). То есть длина каждого проводника дифференциальной пары сравнивается друг с другом. Для выполнения правила расхождение между ними должно быть не более допуска (Tolerance), указанного в правиле.

Параметр Tolerance (Допуск) настраивается во второй диалоговой форме Мастера, корректировка сформированного правила доступна в редакторе правил PCB Rules and Constraints Editor (см. рис. 12).

Если в создаваемых маршрутах присутствуют дифференциальные пары, то дополнительно создаются правила для каждой из этих пар, задающие допустимое расстояние между проводниками одной пары. При необходимости можно настроить допуск в редакторе правил (см. рис. 13). Общее правило Matched Length создаётся для дифференциальных пар. При этом указывается область действия каждого класса маршрутов xSignals, содержащих дифференциальную пару.

В результате своей работы Мастер создаёт маршруты xSignal между одним компонентом-источником и несколькими компонентами-приёмниками. Для этого используется компонентно-ориентированный подход для идентификации потенциальных маршрутов xSignal: выбирается один из компонентов-источников, все интересующие цепи и конечные компоненты, затем производится анализ всех возможных маршрутов от этого компонента к компонентам-приёмникам, через последовательные пассивные компоненты с учётом всех возможных ветвлений. После этого проектировщик выбирает, какие маршруты xSignal нужно создать и, при необ-

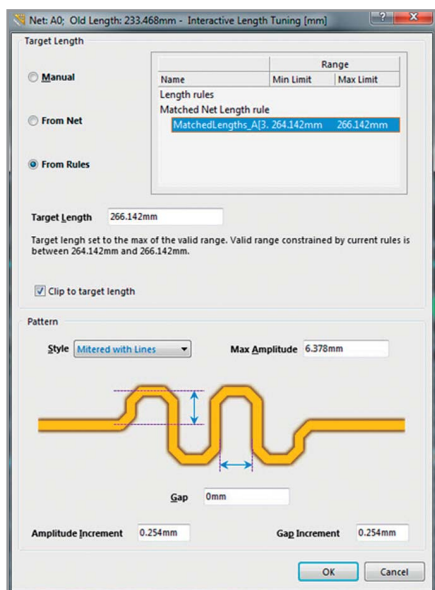


Рис. 14. Настройка конструктивных параметров формы меандра

ходимости, создаёт для них ручную правило Matched Lengths.

Изначально Мастер поддерживал создание маршрутов xSignal для нескольких компонентов только по указанию пользователя. Сейчас он получил расширение, позволяющее автоматически создавать маршруты xSignal и классы маршрутов xSignal для некоторых из популярных коммуникационных интерфейсов и интерфейсов памяти. Первым получил поддержку интерфейс памяти DDR3/DDR4.

УЧЁТ КОНСТРУКТИВНОЙ ДЛИНЫ ВЫВОДОВ В КОРПУСАХ УСТРОЙСТВ

В расчёте Signal Length (Длина маршрута сигнала) учитывается Pin/Pkg Length (Конструктивная длина вывода корпуса), что важно для цифровых интерфейсов, работающих на частотах более 500 МГц, где длина вывода микросхемы или проволоки внутри корпуса, соединяющего вывод с кристаллом, оказывает ощутимое влияние на время прохождения сигналов [2]. Это задержка, обусловленная аппаратными особенностями микросхемы, называется Pin-Package Delay. Производители микросхем предоставляют информацию о вносимых задержках, указывая время в пикосекундах, или длину от кристалла до конца вывода микросхемы. В Altium Designer используется способ учёта с указанием длины, поскольку в этом случае вносимую задержку легко прибавить к полной длине проводника или маршруту xSignal во время проектирования. Информация о значе-

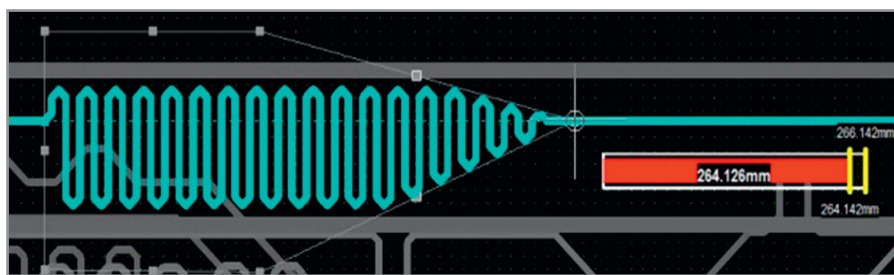


Рис. 15. Интерактивная подстройка меандра проводника на плате

нии параметра Pin/Pkg Length доступна в любом режиме работы с цепями в панели PCB.

Информацию о влиянии конструктивных особенностей микросхемы на время прохождения сигнала необходимо учитывать на этапе назначения входов/выходов или после конфигурирования FPGA. Значение Pin/Pkg Length вносится в Pin Properties (Свойства вывода) в символе компонента в раздел PCB Options. Внесение данных выполняется на этапе формирования библиотеки или на этапе проектирования электрической схемы. После сохранения проекта в файл печатной платы значение длины передаётся в свойства контактной площадки Pad из раздела Properties.

РАСЧЁТ СУММАРНОЙ ДЛИНЫ ТРАССЫ МАРШРУТА СИГНАЛА

Для расчёта суммарной длины маршрута сигнала xSignals в Altium Designer применяется собственный калькулятор длины, который использует следующую формулу:

$$L_{x\text{sig}} = L_{\text{track}} + L_{\text{via}} + H_{\text{cu}} / 2 + L_{\text{pin/pkg (s)}} + L_{\text{pin/pkg (l)}}$$

где L_{track} рассчитывается с учётом кратчайшего пути в участках медного покрытия платы (контактные площадки, регионы и т.п.), L_{via} – расстояние между Layer(start) и Layer(stop) с учётом N слоёв, заданных в Layer Stack Manager, $H_{\text{cu}} / 2$ – половина суммы толщин медного покрытия начального и конечного слоя при переходе между слоями (например, $(0,035 + 0,018) / 2 = 0,0265$, сколько переходов столько дополнительных слагаемых толщины меди), $L_{\text{pin/pkg (s)}}$ – длина вывода устройства-источника сигнала, $L_{\text{pin/pkg (l)}}$ – длина вывода устройства-приёмника сигнала.

ТРАССИРОВКА xSIGNAL

После определения стратегии трассировки и объектов xSignal производит-

ся постепенный сдвиг островов компоновки элементов [3] для получения пространства для прокладки трасс зигзагообразных проводников (меандров). Последнее, собственно, и требуется для решения самой главной задачи – синхронизации сигналов. Формирование меандра производится с помощью следующих инструментов: для одиночной трассы – Tools > Interactive Length Tuning (см. рис. 14), для дифференциальной трассы – Tools > Interactive Diff Pair Length Tuning.

Данный инструмент позволяет выбрать форму меандра и интерактивно изменять параметры меандра, настраивая необходимую точность длины для обеспечения заданной задержки сигнала (см. рис. 15).

ЗАКЛЮЧЕНИЕ

Рассмотренный вариант решения задачи синхронизации высокоскоростных цепей при проектировании топологии печатной платы для высокоскоростного интерфейса является одним из наиболее формализованных. Предлагаемый набор инструментов в САПР Altium Designer позволяет проектировать современные высокоскоростные интерфейсы передачи данных, определяя полный путь прохождения сигнала с учётом конструктивных длин выводов устройств источника и приёмника сигнала, формируя необходимые задержки сигнала путём добавления в проект интерактивно настраиваемого меандра.

ЛИТЕРАТУРА

- Altium. xSignal Wizard. www.techdocs.altium.com/display/ADOH/xSignal+Wizard.
- Altium. Pin-Package Delay Support. [www.techdocs.altium.com/display/ADOH/\(\(Pin+Package+Delay+Support\)\)_AD](http://www.techdocs.altium.com/display/ADOH/((Pin+Package+Delay+Support))_AD).
- Хватынец С.А. Эвристический алгоритм проектирования многослойных коммутационных плат. Сетевой электронный научный журнал «Системотехника». 2008. №6.

