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-MACTEP

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 Intercon-

- песt определяемое пользователем множество компонентов для формирования xSignals. Применение этого режима описано на сайте разработчика [1].
- 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:

💐 xSigna	I Multi-Chip Wizard [mm]	×
Select the Circuit Choose a circuit type or use Custom for general multi-	component xSignals	
On-Board DDR3 / DDR4 Custom Multi-Component Interconnect	On-Board DDR3 / DDR4	ect) Fly-By)
	Data Bus Width (# of Data Lines in each Byte-Lane) Address/Cmd/Ctrl Match Length Tolerance Data Byte-Lanes Matched Length Tolerance Clock Within Diff Pair Length Tolerance	8 V 0.5mm 0.5mm
	Notes: 1. Address Group: A Matched Length rule set will be created PJ-By set from the Address, Command and Control signals. clocks will be accoundedted. 2. Data Group, Multiple Byte-Lane Matched Length rule sets based on the Data Bus Width. 3. If the Clock site for outed to longer than the Matched set I for the Clock site for outed counting and turing the Match engthen the clock as described in data sheet. 4. Pin-Rackage lengths will be included for the source and t the length calculations.	I for each Multiple created o center the ned set, arget pins in
	<u>Cancel</u> < Back Next >	Einish

Рис. 7. Начальное окно xSignal Wizard

Source C	omponent (1 selected)		Min Pin Count:	100
Na /	Schematic Description	Footprint Description	Pin Co	. Layer
Memory	Devices D		Min Pin Count:	40
Memory Target Co	Devices D imponents (2 selected)		Min Pin Count:	40
Memory arget Co Jame /	Devices D mponents (2 selected) Schematic Description	Footprint Description	Min Pin Count:	40 -
Memory arget Co lame / D2 D3	Devices D mponents (2 selected) Schematic Description DDR3 SDRAM ZGb v4, x8, x16 DDR3 SDRAM ZGb v4, x8, x16	Footprint Description BGA 96-Leads, Body 14:0x10.0mm, Pitch 0.80mm (Micror BGA 96-Leads, Body 14:0x10.0mm, Pitch 0.80mm (Micror	Min Pin Count: Pin Count 96	40 Layer Top Laye

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

On-Board DDR3 / D Address group nets ider xSignal Classes will be o	DR4 - Address Grou ntified based on source c created for Fly-By interco	IP omponents. nnect and Match Length rule	5	
ly-By Topology 🛛 🗸	DD1 DD3 V DD	2 ~ 1	(5)
efine xSignal Class Nam	e Syntax	2 xSignal Classes Crea	ated with 21 xSignal Nets in each	2
DDR_PP[#]	(2)	ADDR_PP1	ADDR_PP2	^
		Nets DD1→DD3	Nets DD1→DD2	
Clarify Existing Net Nam	es	DDR3-A0_PP1	DDR3-A0_PP2	
	0002 4(#)	DDR3-A1_PP1	DDR3-A1_PP2	
Addresses	DDR5_A[#]	DDR3-A2_PP1	DDR3-A2_PP2	_
Bank Addresses DDR3 BAI#1	DDR3-A3_PP1	DDR3-A3_PP2		
		DDR3-A4_PP1	DDR3-A4_PP2	
Chip Select	DDR3-CS#	DDR3-A5_PP1	DDR3-A5_PP2	
		DDR3-A6_PP1	DDR3-A6_PP2	
On-Die Termination	DDR3-ODT#	DDR3-A7_PP1	DDR3-A7_PP2 6	
Clock N	DDP3_CIVEE1 N	DDR3-A8_PP1	DDR3-A8_PP2	
CIUCKIN	DDIO-CER[*]_I	DDR3-A9_PP1	DDR3-A9_PP2	
Clock P	DDR3-CLK[#]_P	DDR3-A10_PP1	DDR3-A10_PP2	
0		DDR3-A11_PP1	DDR3-A11_PP2	_
(3)	Command Inputs	DDR3-A12_PP1	DDR3-A12_PP2	
RAS	DDR3-RAS#	DDR3-A13_PP1	DDR3-A13_PP2	_
		DDR3-A14_PP1	DDR3-A14_PP2	
CAS	DDR3-CAS#	DDR3-BA0_PP1	DDR3-BA0_PP2	
	0002 WE#	DDR3-BA1_PP1	DDR3-BA1_PP2	
VVE	DDRJ-WL#	DDR3-BA2_PP1	DDR3-BA2_PP2	~
Analyze Syntax & Cre	eate xSignal Classes →	(4)	Modify Nets in xSignal Classes (7)	

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.

 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 компонентовприёмников 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.

- Пользователем задаётся имя класса. По умолчанию используется имя ADDR_PP[#]. Символ [#] отражает количество устройств памяти. При необходимости суффикс PP может быть изменён.
- 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,

T-Branch Topology \checkmark U2 \checkmark U3 \checkmark U1 U5 \checkmark U4 \checkmark Рис. 10. Выпадающие списки для установки требуемого порядка

конечных устройств в топологии T-Branch

💐 xS	igna	al Multi-Chip Wiza	ard [mm]		
On-Board DDR3 / DDR4 - Data Group Data group nets identified based on source com xSignal Classes will be created for Byte-Lanes	poner and to	nts. 9 set Length rules	****		
Define xSignal Class Name Syntax		4 Byte-Lane xSignal G	lasses Created with 1	1 xSignal Nets in each	(4)
DATA_BL[#] (1)		DATA BLO	DATA BL1	DATA BL2	DATA BL3
		Nets DD1→DD3	Nets DD1→DD3	Nets DD1→DD2	Nets DD1-
Clarify Existing Net Names	_	DDR3-D16_BL0	DDR3-D24_BL1	DDR3-D0_BL2	DDR3-D8_I
		DDR3-D17_BL0	DDR3-D25_BL1	DDR3-D1_BL2	DDR3-D9_I
(2) Data DDR5_D[#]	~	DDR3-D18_BL0	DDR3-D26_BL1	DDR3-D2_BL2	DDR3-D10
Data Mask DDR3-DM[#]	~	DDR3-D19_BL0	DDR3-D27_BL1	DDR3-D3_BL2	DDR3-D11
	_	DDR3-D20_BL0	DDR3-D28_BL1	DDR3-D4_BL2	DDR3-D12
Data Strobe N DDR3-DQS[#]_N	~	DDR3-D21_BL0	DDR3-D29_BL1	DDR3-D5_BL2	DDR3-D13
	~	DDR3-D22_BL0	DDR3-D30_BL1	DDR3-D6_BL2	DDR3-D14
Data Strobe P DDRS-DQS[#]_P		DDR3-D23_BL0	DDR3-D31_BL1	DDR3-D7_BL2	DDR3-D15
L	_	DDR3-DM2_BL0	DDR3-DM3_BL1	DDR3-DM0_BL2	DDR3-DM1
		DDR3-DQS2_N_BL0	DDR3-DQS3_N_BL1	DDR3-DQS0_N_BL2	DDR3-DQS
		DDR3-DQS2_P_BL0	DDR3-DQS3_P_BL1	DDR3-DQS0_P_BL2	DDR3-DQS
Analyze Syntax & Create xSignal Classes –		Ċ	Ð		
		<			>
		[Modify Nets in xSig	nal Classes 6	
Create Spreadsheet		2	ancel < Back	Next >	Einish

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.

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

3	PCB Rules and Constraints Editor [mm]	? 🔀	8	PCB Rules and Constraints Editor [mm]
Borkige Rules Construction	Constraints	Unique ID VICUHOW Test Queries	Beign Rules Constrained and a second a second and a second a se	New Index Address Utique ID MEMOGEL Text Querier Where The Object Matches InxSignalClass (*ADDR, FPS1) OR InxSignalClass (*ADDR, FP21) OR InxSignalClass (*ADDR, FP31) OR InxSignalClass (*ADDR, FP21) OR InxSignalClass (*ADTA, 552.0) OR InxSignalClass (*ADTA, 552.0) Georg Builder Constraints Tolerance 0.5mm Oroup Matched Lengths © With Differental Par Length
			Bule Wizard Priorities Create Default Rule	s OK Cancel Apply
Rule Wizard Priorities Create Default Rule:	\$	OK Cancel Apply	Рис. 13. Вил плавила Ма	tched Length для лифференциальных дар

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

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

действий в этом диалоговом окне следующая:

- 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.
- 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.
- 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 xSigna 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 используется 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 олним 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 вручную 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:

$$\begin{split} \mathbf{L}_{\mathrm{xsig}} &= \mathbf{L}_{\mathrm{track}} + \mathbf{L}_{\mathrm{via}} + \mathbf{H}_{\mathrm{cu}} \ / \ 2 + \\ &+ \mathbf{L}_{\mathrm{pin/pkg}\,(s)} + \mathbf{L}_{\mathrm{pin/pkg}\,(l)}, \end{split}$$

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 Н слоёв, заданных в Layer Stack Manager, H_{си} / 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_{pin/pkg (s)} – длина вывода устройства-источника сигнала, L_{pin/pkg (l)} – длина вывода устройстваприёмника сигнала.

Трассировка хSignal

После определения стратегии трассировки и объектов xSignal производится 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.
- Хватынец С.А. Эвристический алгоритм проектирования многослойных коммутационных плат. Сетевой электронный научный журнал «Системотехника». 2008. №6.