

## Между ISE и ViewDraw

Эта статья предназначена, в первую очередь, для тех инженеров, которые разрабатывают свои проекты для FPGA в ISE и ViewDraw, однако первая часть статьи будет полезна тем, кто использует только ISE, а схемы плат «рисует» во ViewDraw. В материале описана небольшая программа Pin Manager, выполненная автором, опыт ее разработки и использования. Программа Pin Manager позволяет ликвидировать брешь в автоматической разработке проектов, сокращает ручной труд и устраняет ошибки, которые могут появиться при ручном назначении параметров в выполняемом проекте. Статья не имеет цели рекламировать Pin Manager, а только описывает заложенные в нее методы работы и методики разработки самой программы.

Иосиф КАРШЕНБОЙМ  
iosifk@narod.ru

### Небольшое историческое отступление

Кажется, еще совсем недавно были те времена, когда над столом каждого уважающего себя инженера висела фотография или нарисованная от руки схема, на которой было изображено десятка два или три микросхем с нарисованными квадратиками корпусов и с номерами выводов. И каждый инженер, и даже его начальник знал, куда надо подключить вывод № 7 и вывод № 14. Можно сказать, что перестройка в нашей отрасли и в стране в целом и началась с резкого увеличения числа выводов у микросхем. Вот вам факты (да простит меня редактор за небольшое отступление):

- 14 выводов — социализм;
- 16 выводов — развитой социализм;
- 40 выводов — перестройка и гласность;
- 68 выводов — новая Россия;
- 204 вывода — дефолт;
- 1517 выводов — сейчас, это вы сами знаете что.

### Есть ISE, есть ViewDraw, и несколько раз по 1517 выводов

Итак, есть ISE. Кажется, с помощью этой программы можно творить чудеса. Но... Как только дело доходит до назначения выводов и стыковки со схемой, тут как раз и возникает возможность для появления ошибок. Надо назначить тип выводов, указать их нагрузочную способность, скорость нарастания, расположение на корпусе и т. д. А если проект выполнен в ISE и во ViewDraw, то трудоемкость возрастает многократно, ибо надо выполнить часть проекта в ISE, затем добавить часть во ViewDraw, и уже там не только перечислить параметры выводов, как в USF-файле для ISE, но и графически изобразить каж-

дый вывод, указать его параметры путем «кликанья» по графическому символу. Причем каждый параметр надо устанавливать отдельно. «Ну и что тут такого — скажет иной читатель, — сел, пару дней поколотил по «клаве», и все дело». С одной стороны, это так, но ведь при конструировании PCB разработчик платы имеет возможность поменять выводы на FPGA для тех сигналов, которые допускают эту возможность. После этого процесс «кликанья» и переназначения параметров выводов приходится повторять. А ведь именно там, где много ручной работы, много и ошибок. Поиск такой ошибки довольно трудоемок. Неправильно же назначенный вывод может привести к выходу из строя других компонентов и микросхем на плате.

Именно для того, чтобы уменьшить число ошибок при разработке устройств с FPGA, и была сделана автором небольшая программа Pin Manager. Программа была ориентирована на применение микросхем серии virtex2, но опыт ее использования может быть полезен тем разработчикам, которые используют микросхемы других типов.

Далее будет приведено краткое описание выполняемых функций и описание форматов файлов программы ViewDraw.

### Программа Pin Manager

Программа Pin Manager выполняет следующие функции:

1. Позволяет создать базу данных для выводов FPGA в данном проекте. Для создания базы данных для конкретного типа корпуса микросхемы импортируется файл «пустого проекта», который получается в результате работы ISE.

В базу данных заносятся все параметры выводов:

1.1. XY coord — контакт на корпусе.

1.2. User name — название вывода.

1.3. Pin Usage — назначение вывода, например вход IO, вход опорного напряжения и т.д.

1.4. Pin description — импортируется из описания выводов Xilinx.

1.5. Direction — вход или выход.

1.6. IO Standard.

1.7. IO Bank.

1.8. Drive (mA).

1.9. Slew Rate.

1.10. Termination.

1.11. IOB — должен ли этот сигнал иметь триггер в выходной ячейке микросхемы.

1.12. Delay.

1.13. Voltage.

1.14. Constraint.

1.15. User group.

1.16. TNM.

1.17. Comment.

Назначения параметров можно производить как по отдельности, так и для всей колонки параметров одновременно. Кроме того, есть способ заполнения параметров в колонке по шаблону, накладываемому на данные из другой колонки. Например, требуется установить параметр Drive=24 для тех сигналов, в названии которых есть «I\_FORCE\_». Таким образом, производится назначение параметров на все провода для шины «I\_FORCE\_» одновременно. При назначении параметров «старое» значение заносится в промежуточный буфер и может быть возвращено обратно.

Внешний вид окна программы с основной таблицей сигналов приведен на рис. 1.

2. Программа позволяет создать базу данных для блоков DCM в данном проекте.

3. Программа позволяет выделять дифференциальные сигналы стандарта LVDS, собирать их в пары и помещать в отдельную таблицу (рис. 3).

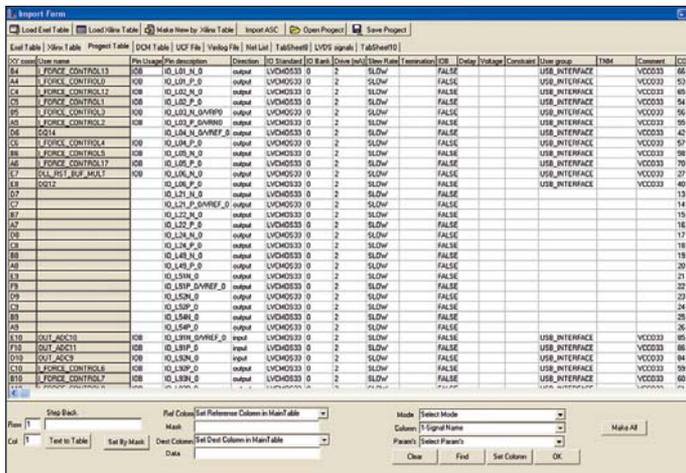


Рис. 1. Внешний вид окна программы с основной таблицей сигналов

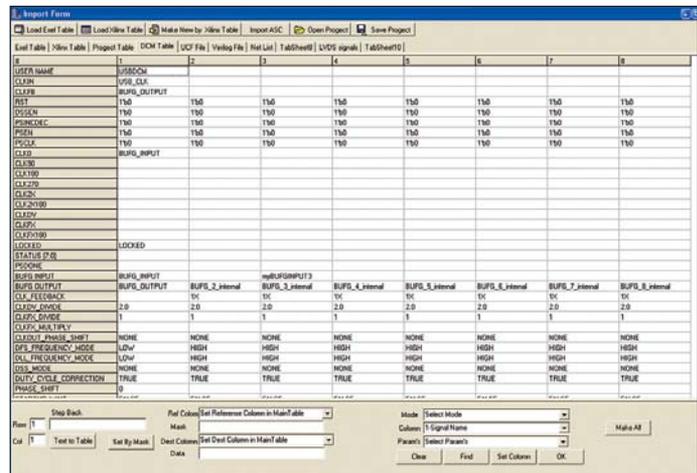


Рис. 2. Внешний вид окна программы с таблицей данных для блоков DCM

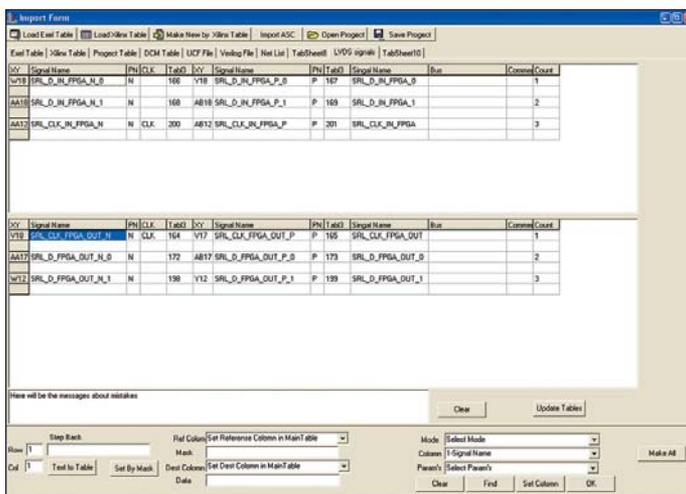


Рис. 3. Внешний вид окна программы с таблицей дифференциальных сигналов стандарта LVDS



Рис. 4. Внешний вид окна программы с изображением выводов микросхемы в графическом виде

4. Программа формирует изображение выводов микросхемы в графическом виде.

Кнопки позволяют увеличивать или уменьшать масштаб изображения. В каждой клетке, соответствующей определенному типу вывода, помещена информация о координате вывода, названии сигнала, типе вывода. Цвет клетки определяет тип вывода — IO, питание, земля, неподсоединенные выводы и т. д. На рис. 4 представлено изображение для xc2v250\_fg456.

5. Импортирует ASC-файл, полученный при конструировании PCB с помощью программы ViewDraw, и заносит импортированный файл в базу. При этом меняются только названия сигналов на соответствующих контактах, а все остальные параметры типа Constraint или TNM остаются. Соответственно корректируется таблица для дифференциальных сигналов.

6. После того как базы данных созданы, программа позволяет выполнить различные настройки, необходимые для генерации выходных файлов. Внешний вид окон программы, предназначенных для установки параметров, приведен на рис. 5.

7. Программа выполняет генерацию Verilog-файлов. В случае использования линий связи стандарта LVDS весь проект становится удобно разбивать на две части. Одна часть —

внешняя, содержит описание буферов и портов и является проектом верхнего уровня. Вторая часть — внутренняя логика, она представляет собой проект нижнего уровня.

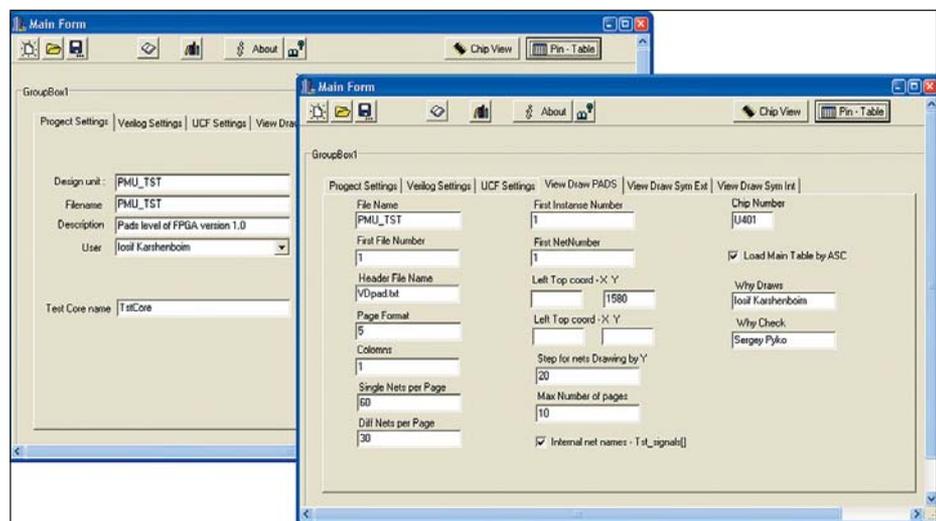


Рис. 5. Внешний вид окон программы, предназначенных для установки параметров

Соответственно, сначала программа производит генерацию Verilog-файлов верхнего проекта. В этом файле программа автоматически вставляет буферы нужного типа и делает переход от дифференциальных сигналов в верхнем проекте к недифференциальным сигналам в нижнем проекте, описывает входы и выходы и провода, поступающие к ним.

Для блоков DCM в проект вставляются фрагменты текста, и если названия сигналов в таблице данных для блоков DCM соответствуют принятым соглашениям о названии промежуточных сигналов в проекте верхнего уровня, то никакой ручной доработки не требуется. В противном же случае надо только исправить имена сигналов, поступающих на блоки DCM.

Для проекта нижнего уровня программа так же выполняет генерацию Verilog-файлов. В этом файле программа описывает порты нижнего уровня, входы и выходы проекта и провода, поступающие к ним.

Поскольку программа первоначально создавалась для быстрой разработки проектов, предназначенных для проверки межсоединений на плате, программа позволяет в проекте нижнего уровня подключить к выводам выходы узла тестирования. Узел тестирования — это счетчик и сдвиговый регистр, он представляет собой файл, написанный на Verilog. Этот файл подгружается к проекту нижнего уровня.

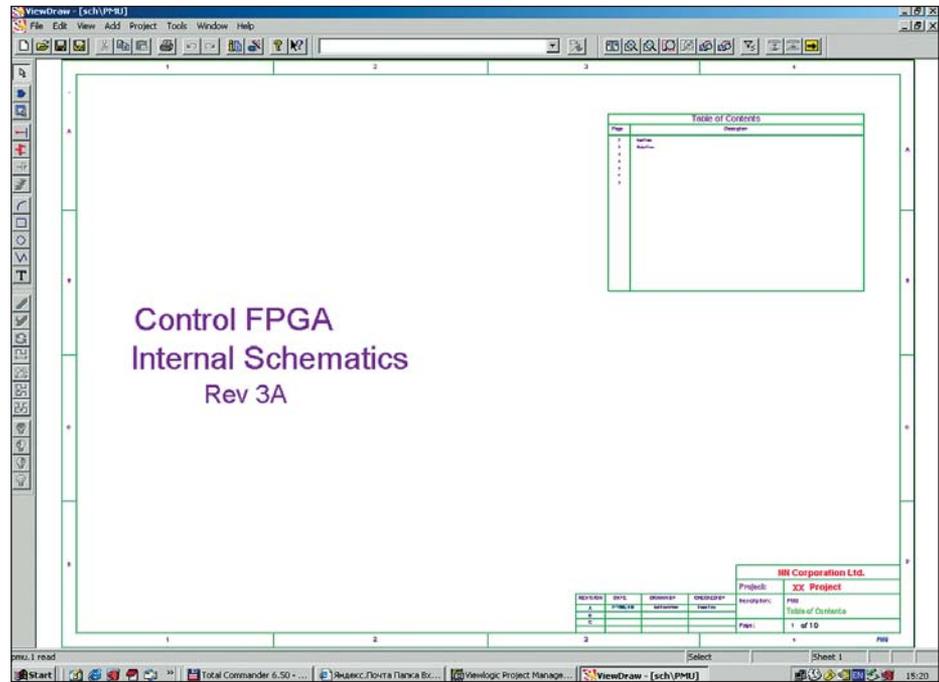


Рис. 6. Заглавная страница с названием проекта, сгенерированная программой

8. Программа выполняет генерацию USF-файлов. На основании параметров, введенных в основную таблицу сигналов, производится генерация USF-файла для данного проекта.

9. Программа выполняет генерацию страниц проекта в формате ViewDraw. К ним относятся:

9.1. Заглавная страница с названием проекта (рис. 6).

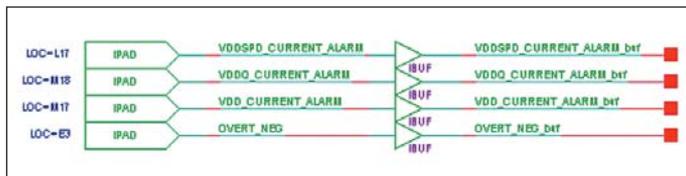


Рис. 7. Массив входных буферов и PAD, сгенерированных программой

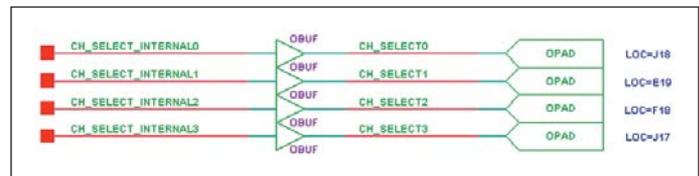


Рис. 8. Массив выходных буферов и PAD, сгенерированных программой

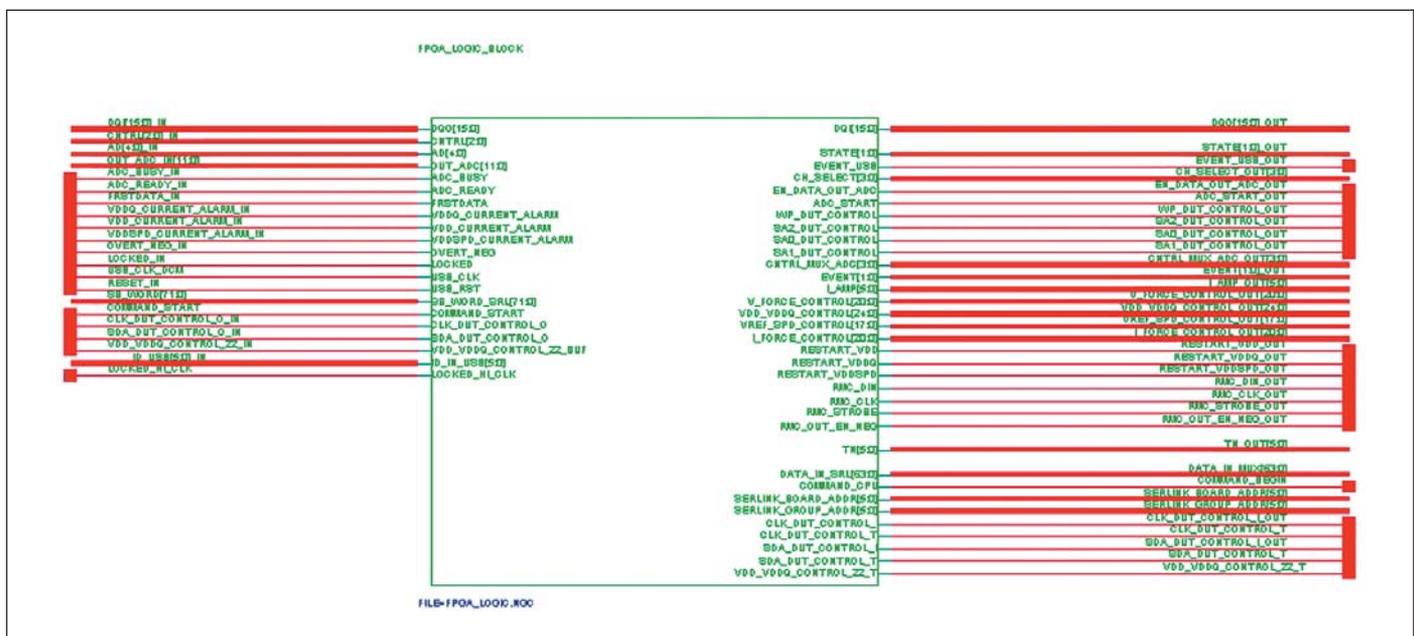


Рис. 9. Функциональный блок проекта нижнего уровня, сгенерированный программой

9.2. Страницы, на которых помещены буферы ввода-вывода и PAD с установленными параметрами по скорости нарастания сигнала, расположения вывода, ограничениями для данного сигнала, стандарту сигнала и другими параметрами, перечисленными в п. 1. Программа «умеет» работать с наиболее распространенными типами буферов, а именно с буферами ввода, вывода, вывода с третьим состоянием, дифференциальными входами и выходами и т. д. (рис. 7, 8).

9.3. Выполняется генерация символа проекта нижнего уровня как библиотечного элемента — «квадратика».

9.4. Выполняется генерация страницы проекта с установленным блоком — «квадратиком» проекта нижнего уровня, к которому уже подведены и названы провода, идущие к буферам ввода-вывода (рис. 9).

## Секрет файлов ViewDraw

Теперь расскажем о том, как надо формировать файлы для программы ViewDraw. При изучении формата файлов программы ViewDraw была использована статья Стюарта Брорсона [1].

Пример файла программы ViewDraw (часть строк этого и других файлов примеров удалены для того, чтобы не загружать статью):

```
V 51
K 144004269600 FL_se
Y 0
D 0 0 2338 1653
Z 7
i 75
U 2000 50 20 0 3 3 @SHEET=1
U 1540 110 10 0 5 3 @DATETIME=07.08.04 20:20:24
U 2270 20 18 0 8 3 @NAME=FL_se
I 32 A2SHEET 1 0 0 1 '
T 1510 1450 15 0 3 Page
L 2200 1470 1500 1470
Q 0 5 0
T 190 760 100 0 3 Internal Schematics
T 200 870 100 0 3 Control FPGA
E
```

Мы видим, что файлы — текстовые, причем строго структурированные, что значительно облегчает нам задачу.

Файл состоит из отдельных строк. Каждая строка начинается с ключевой буквы и продолжается набором параметров. Похоже на пакетную передачу данных. От буквы V — начала пакета до буквы E — конца пакета. Буквы K, Y, D и т. д. обозначают начало субкадров, входящих в пакет. Цифры и пр. выглядят как данные, передающиеся в субпакете.

Итак:

- **V** — начало файла.
- **K** — не очень понятный код, который генерируется при создании файла программой ViewDraw, был взят автором из «родных» файлов ViewDraw, которые генерировались у него на машине. Тем не менее, этот код воспринимался и на других машинах.

Таблица 1

Атрибуты	Описание	Комментарий
xpos, ypos	Местоположение текста атрибутов	
text_size	Размер букв текста (Font size)	Обычно 10
rotation	0 = 0 1 = 90 2 = 180 3 = 270	
text_pos	Место привязки текста 1 = lower 2 = middle 3 = upper	
Scope	0 = local 1 = global	
attribute	Текст атрибутов, обычно записываемый как Attribute=value	

P — параметр, описывающий выводы (табл. 2), его формат:  
P pin\_id xend yend xbeg ybeg rotation side invert

Таблица 2

Атрибуты	Описание	Комментарий
pin_id.	нумерация ID выводов, начинается с 1	
xend, yend, xbeg, ybeg	Начало и конец линии, подключенной к выводу	
rotation	0 = 0 1 = 90 2 = 180 3 = 270	
side	неизвестно	Значения от 0 до 4, обычное значение — 2
invert	0 = неинвертирующий вход 1 = инвертирующий вход (рисуетя «кружок», как знак инверсии)	

A — параметр атрибутов, которые связаны с конкретным выводом (табл. 3), его формат:  
A xpos ypos text\_size rotation text\_pos visibility attribute

- **Y** — символ, определяющий параметр «type»:
  - 0 = composite;
  - 1 = module;
  - 2 = annotate;
  - 3 = pin.
- **D** — размер символа «блок». Формат записи — D xmin ymin xmax ymax.
- **Z** — размер листа на экране — параметр «size»:
  - 0 = A;
  - 1 = B;
  - ...
  - 5 = A4;
  - ...
  - 10 = Z (user defined).

- **i** — неизвестный параметр.
- **U** — параметр, который доступен как пункт «properties» из выпадающего меню (см. табл. 1), его формат: U xpos ypos text\_size rotation text\_pos scope attribute.

Длинные строки могут быть разделены на части. Части строки имеют в начале символ «+», затем пробел.

Все эти чудеса дешифрования в духе Эдгара По конечно интересны, но существуют и простые инженерные способы получить требуемую информацию. Поскольку все файлы программы ViewDraw — текстовые, то поступаем следующим образом. Сначала «вручную» делаем шаблон страницы. Открываем

Таблица 3

Атрибуты	Описание	Комментарий
xpos, ypos	Местоположение атрибутов	
text_size	Размер букв текста (Font size)	Обычно 10
rotation	0 = 0 1 = 90 2 = 180 3 = 270	
text_pos	Место привязки текста 1 = lower 2 = middle 3 = upper	
visibility	0 = invisible 1 = visible 2 = name only 3 = value only	
attribute	Текст атрибутов, обычно записываемый как Attribute=value	

L — параметр, описывающий метку, связанную с конкретным выводом (табл. 4), его формат:  
L xpos ypos text\_size rotation text\_pos scope visibility logic\_sense text\_label

Таблица 4

Атрибуты	Описание	Комментарий
xpos, ypos	Местоположение атрибутов	
text_size	Размер букв текста (Font size)	Обычно 10
rotation	0 = 0 1 = 90 2 = 180 3 = 270	
text_pos	Место привязки текста 1 = lower 2 = middle 3 = upper	
scope	0 = local 1 = global	
visibility	0 = invisible 1 = visible 2 = name only 3 = value only	
logic_sense	0 = normal 1 = inverted (bar above label)	Активный уровень сигнала прямой или инверсный, для инверсного подчеркивание над текстом
text_label	Текст метки	

**b** — параметр, описывающий размер прямоугольника, рисуемого как контур символа, его формат:  
b xmin ymin xmax ymax.  
**T** — параметр, описывающий строку текста, его формат: T xpos ypos text\_size rotation text.  
**I** — инстанс, или компонент на поле чертежа, в его формат входят: I instance\_number instance\_name...  
**C** — подключение входов для компонента.  
**N** — цепь, ее формат: N net\_number.  
**J** — точка подключения цепи.  
**S** — параметр, определяющий направление передачи в цепи от входа цепи к ее выходу.  
**E** — конец файла.

ViewDraw, определяем формат листа, добавляем рамку и штамп. Выполняем в штампе все требуемые надписи. Сохраняем результат в файл. Получаем шаблон листа, который мы будем формировать программным способом. Теперь мы знаем, куда надо подставлять дату и фамилии разработчиков, а также название проекта, название страницы и другие параметры программным способом в соответствии с заданными для программы Pin Manager параметрами.

Далее делаем шаблон для автоматической генерации страниц с массивами входных буферов и PAD.

Для этого также «вручную» делаем шаблоны страницы. На поле листа с рамкой и штампом помещаем входной буфер

и PAD, соединяем их проводами, устанавливаем все атрибуты буфера и PAD так, как впоследствии мы их хотим видеть. Располагаем этот фрагмент схемы в левом нижнем углу листа. Далее копируем этот фрагмент схемы и помещаем его в правом верхнем углу листа, так чтобы оба фрагмента располагались по вертикали друг над другом. Далее, точно таким же образом, копируем и располагаем третий фрагмент схемы под вторым, так чтобы получить требуемый шаг по вертикали. Сохраняем результат в файл. Теперь рассмотрим полученный файл. Видим, что начало координат в программе ViewDraw — левый нижний угол. По полученному шаблону определяем минимальную нижнюю координату и максимальную верхнюю координату. Определяем требуемый нам шаг по вертикали. Далее вычисляем, сколько компонентов можно разместить на одной странице, если их располагать в одну колонку.

Если необходимо располагать компоненты в две колонки, то все действия повторяем для второй колонки. Когда программа Pin Manager располагает на странице максимально возможное число компонентов, и необходимо продолжить генерировать фрагменты схем, она добавляет в файл символ окончания фай-

ла — «E», записывает результат на диск и начинает формировать следующий файл.

Таким же образом производится формирование шаблонов и для фрагментов схем выхода, двунаправленных входыходов и пр.

Почти таким же способом поступаем и в случае формирования символа элемента. Делаем «вручную» шаблон символа элемента. Добавляем к символу 3 входа и 3 выхода. 2 входа дадут размер по вертикали, а третий — шаг. Так же и с выходами. Сохраняем результат в файл. Анализируем полученный файл и видим, как описываются входы и выходы, какой шаг по вертикали, какой взят размер контура символа элемента и пр.

При формировании листа схемы с нашим библиотечным элементом необходимо к шаблону листа схемы добавить строки, относящиеся к символу элемента и сделать подключения проводов. Названия проводов берутся из названий проводов проекта нижнего уровня и соответствуют названиям проводов, которые соединяют буферные компоненты схемы с проектом нижнего уровня.

### Заключение

Приведенное здесь описание программы Pin Manager позволяет понять, что такой ме-

тод проектирования удобен тем, что позволяет быстро и без ошибок выполнить наиболее трудоемкую (могу даже сказать, наиболее «тупую») часть проектирования. В том случае, когда по итогам конструирования PCB необходимо откорректировать схему, весь процесс исправлений занимает несколько минут.

Сама база данных может быть выполнена различными способами. Она может быть сделана как однопользовательской, так и многопользовательской. Программа Pin Manager выполнена так, что позволяет импортировать и экспортировать свои файлы в формате Exel (текстовый формат с разделителем «;»). Это позволяет производить сортировку и поиск в таблицах стандартными средствами.

Ну а тех читателей, которых заинтересовала связь числа выводов микросхем с процессами в нашем обществе, автор отсылает к своему сайту [2]. ■

### Литература

1. Stuart Brorson. Secrets of the ViewDraw file format. <http://www.brorson.com/ViewDraw/ViewLogicFileFormat.pdf>.
2. И. Каршенбойм. «Новогодние истории». [www.iosifk.narod.ru](http://www.iosifk.narod.ru).