

Программное обеспечение для перевода проектов ПЛИС из САПР Maxplus II в базис БМК

Алексей Королёв (Москва)

В статье описаны принципы разработки проектов для базовых матричных кристаллов (БМК) в среде разработки для ПЛИС с последующей трансляцией в базис БМК при помощи разработанной автором программы. Уделено внимание характерным особенностям и возможным затруднениям, возникающим в процессе трансляции, и способам их преодоления.

ВВЕДЕНИЕ

В последнее время широкое применение получили программируемые логические устройства (ПЛУ). Эти устройства позволяют конечному пользователю задавать операции с помощью процесса, который называется программированием структуры связей схемы. Применение ПЛУ увеличивает эффективность проектирования и упрощает разработку, поэтому многие современные цифровые системы разрабатываются именно на программируемых логических интегральных схемах (ПЛИС). Задача разработчика схем заключается при этом в обозначении выводов, расстановке логических элементов, связей между ними и написании тестов.

Микросхема ПЛУ поставляется с соединительными звеньями, которые находятся внутри интегральной схемы. Чтобы получить к ним доступ, микросхему надо перевести в особый режим, в котором на её определённые выходы подаются специальные (более высокие) напряжения. Этот процесс называется программированием ПЛУ. При этом микросхема помещается в специальный прибор – программатор. В свою очередь, большая часть современных программаторов присоединяется к персональному компьютеру с программным обеспечением, содержащим библиотеки с заложенной информацией о множестве типов программируемых устройств [1].

В микросхемах на базовых матричных кристаллах (БМК) возможность такого программирования отсутству-

ет. Это существенно затрудняет возможность использования БМК для быстрого создания опытных образцов. С другой стороны, при среднесерийном производстве использование БМК вместо ПЛИС обходится дешевле. Поэтому для того, чтобы быстрее получить опытный образец, схему разрабатывают в системе проектирования ПЛИС, а затем файлы проекта преобразуют в систему проектирования для БМК. Именно о таком процессе преобразования пойдёт речь.

В настоящее время широкое распространение получило программное обеспечение Max+Plus II для проектирования ПЛИС фирмы Altera. Программа ConvChip, разработанная автором данной статьи, позволяет преобразовать проект схемы из системы проектирования Max+Plus II в систему моделирования «Невод» для микросхем на базе БМК. В процессе преобразования программа ConvChip осуществляет перевод структуры схемы, тестовых векторов и сопутствующей технической информации.

При переводе тестов преобразователь оставляет только те тестовые вектора, на которых реакция схемы полностью определена. Происходит определение значений не задействованных в тесте выводов и объединение в один тест нескольких файлов временных диаграмм. Поскольку в Max+Plus II время поступления данных на входы микросхемы на протяжении всего теста постоянно изменяется, происходит вычисление оптимального периода, одинакового

для всех тестовых воздействий. Вычисление данного периода необходимо, если контрольное и измерительное оборудование (КИО), на котором происходит тестирование схемы БМК, не в состоянии менять период на протяжении теста. Коррекция тестовых воздействий осуществляется преобразователем при помощи множества опций и настроек, учитывающих специфику КИО и выбранной библиотеки БМК в системе «Невод».

Структура экспортного файла из Max+Plus II записана на языке EDIF-2.0, который широко распространён на сегодняшний день. В процессе работы преобразователь переводит данный язык во внутренний формат STR для системы моделирования. Базисы ПЛИС и БМК существенно различаются, поэтому в процессе преобразования структура схемы сильно модифицируется. Рассмотрим различные способы обработки структуры схемы.

ОРГАНИЗАЦИЯ СИНХРОНИЗАЦИИ ДЛЯ ТРИГГЕРОВ РАЗЛИЧНЫХ ТИПОВ

Сразу после включения ПЛИС происходит сброс всех триггеров. Таким образом, перед подачей первого тестового вектора содержимое каждого триггера известно [2]. После включения БМК состояние триггеров не определено, поэтому после подачи тестовых векторов реакция схемы становится непредсказуемой на протяжении нескольких тестов. Триггер может сохранять неопределённость на выходе на протяжении всего теста. Для устранения данной проблемы преобразователь устанавливает цепи сброса для всех триггеров. На рисунке 1 показана исходная схема подключения синхронного D-триггера из стандартной библиотеки Max+Plus II [3].

Преобразователь устанавливает цепь сброса для обнуления содержи-

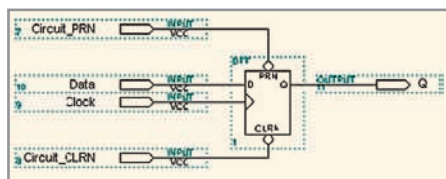


Рис. 1. Исходное подключение триггера

мого всех триггеров, которая включается только перед началом тестирования. На рисунке 2 показано подключение цепи сброса для данного D-триггера. На схеме обозначены следующие выводы:

1. Q – выход триггера;
2. Circuit_CLRN и Circuit_PRN – входы для сброса и установки триггера;
3. Data – вход данных триггера;
4. Clock – вход синхронизации, по которому работает триггер;
5. Reset – вход сброса, единый для всех триггеров.

Если Reset = 1, то CLRN = 0, PRN = 1 и сигналы Circuit_CLRN, Circuit_PRN игнорируются, т.е. триггер сбрасывается в нулевое значение. При Reset = 0 цепь синхронизации отключается, и сигналы Circuit_CLRN и Circuit_PRN проходят на триггер.

УСТРАНЕНИЕ КОНФЛИКТОВ ИСТОЧНИКОВ ДЛЯ ДВУНАПРАВЛЕННЫХ ВЫХОДОВ

На рисунке 3 показан фрагмент схемы с двунаправленными выводами.

Элемент с тремя состояниями TRI при сигнале OE = 1 пропускает сигнал с входа Input на двунаправленный вывод Inout и выход Output. При сигнале OE = 0 цепь входного сигнала Input отключена и сигнал с двунаправленного вывода Inout поступает на выход Output. Таким образом, с помощью элемента TRI в Max+Plus II осуществляется управление двунаправленным выводом. В Max+Plus II такое подсоединение не вызывает особых проблем, но в схеме, полученной после преобразования в базис БМК, в точке A могут происходить конфликты источников сигналов. В одной связи может быть несколько выводов Input, Output и Inout, поэтому и точек, где появляются конфликты сигналов, может быть очень много. При этом выводы могут быть разделены большим числом различных вентилях и функциональных блоков, число которых может достигать нескольких тысяч. На рисунке 4 показан исходный фрагмент схемы в базисе ПЛИС, в котором преобразователь

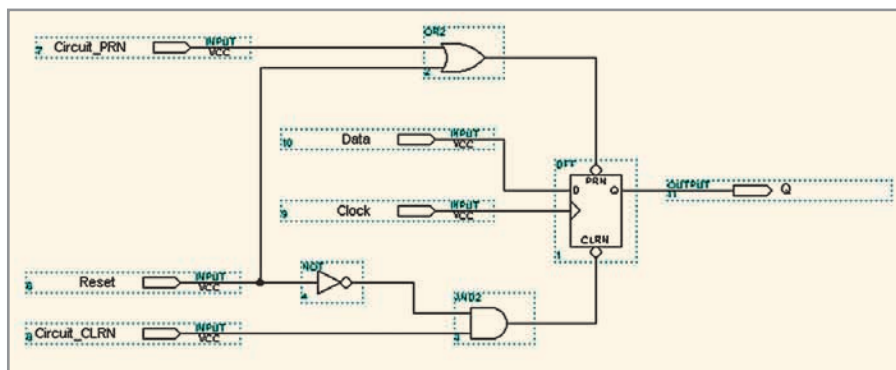


Рис. 2. Подключение триггера после преобразования

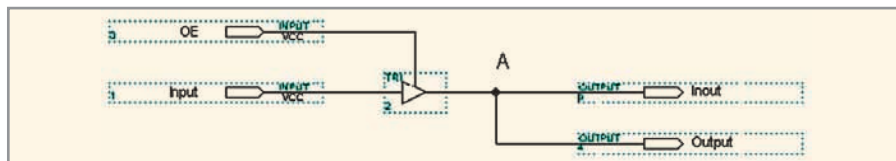


Рис. 3. Схема управления двунаправленным выводом

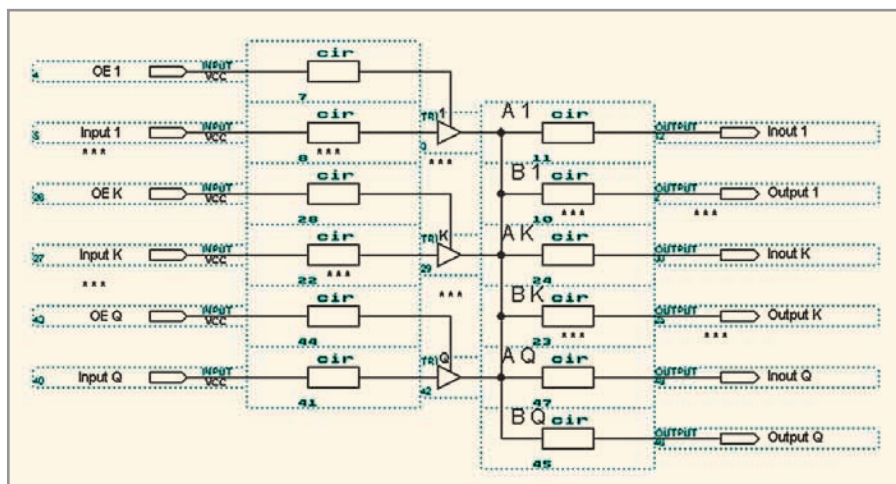


Рис. 4. Исходный фрагмент схемы с двунаправленными выводами в общем виде до преобразования

устраняет возможные конфликты источников при переводе проекта.

Для сокращения размеров рисунка многоточием обозначаются элементы, аналогичные изображенным с индексами 1, K, Q. Cir – произвольный фрагмент схемы, разделяющий рассматриваемые выводы; её устройство не имеет принципиального значения. Эти схемы не рассматриваются преобразователем при замене элементов TRI на схему TRI-BMC. Номера 1, K, Q после имени обозначают, что данные элементы аналогичны друг другу.

Точки A(1..K..Q) показывают места возможных конфликтов источников входа с двунаправленным выводом. Точки B(1..K..Q) показывают места возможных конфликтов источников входа с выходом.

Преобразователь устраняет конфликты источников путём замены

элементов TRI на схему TRI-BMC, которая управляет режимом двунаправленного вывода. Рассмотрим алгоритм модификации схемы:

- поиск всех двунаправленных выводов Inout (1..K..Q) в конкретной связи;
- поиск точек A (1..K..Q) и B (1..K..Q), в которых возможны конфликты;
- вычисление выходных связей, проходящих через точки B (1..K..Q);
- вычисление количества элементов TRI и способа их подключения;
- замена элементов TRI на TRI-BMC с соответствующим подключением выводов.

На рисунке 5 представлен фрагмент схемы после модификации по данному алгоритму.

Элемент TRI-BMC разрабатывается в базисе БМК под конкретную библиотеку системы моделирования «Невод».

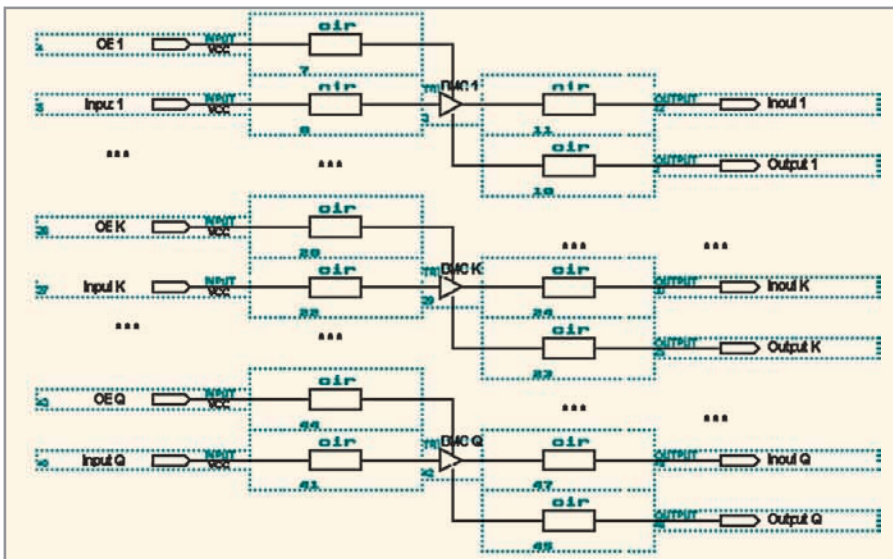


Рис. 5. Конечный фрагмент схемы с двунаправленными выводами в общем виде после преобразования

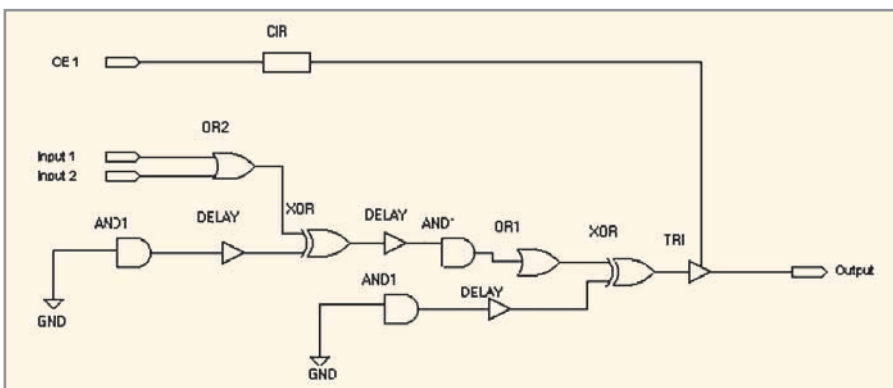


Рис. 6. Фрагмент схемы EDIF после экспорта структуры

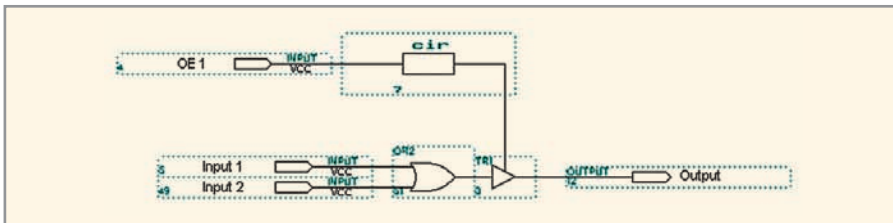


Рис. 7. Фрагмент схемы EDIF, разработанный без программы Max+Plus II

Принцип действия данного элемента состоит в управлении двунаправленным выводом путём разделения входящих потоков от вывода Input и вывода Inout, работающего в режиме входа.

СОКРАЩЕНИЕ КОЛИЧЕСТВА ВЕНТИЛЕЙ В СТРУКТУРЕ СХЕМЫ ПОСЛЕ ПРЕОБРАЗОВАНИЯ

Экспорт схемы в формат EDIF из программы Max+Plus II имеет множество недостатков:

- отсутствует сохранение всей иерархии проекта для оптимизации процесса компиляции в Max+Plus II [4];
- структура экспортного файла в формате EDIF не соответствует

структуре исходного проекта Max+Plus II. Это происходит из-за того, что в Max+Plus II макромодули разбиты до уровня реализации схемы на базисе аппаратуры ПЛИС;

- велик коэффициент избыточности вентилях в экспортной структуре по сравнению с аналогичной схемой, сразу сделанной в базисе БМК. Остановимся на последнем недостатке подробнее.

На рисунке 6 показан типичный фрагмент схемы в формате EDIF после экспорта из Max+Plus II, который реализует логику «ИЛИ» на элементе OR2. C_{ir} – любая схема, не представляющая интереса в данном случае. Элементы DELAY, AND1, OR1 и XOR2 не влияют на

логику работы данной схемы, поэтому их можно удалять из схемы, но при этом надо учитывать «гонки» сигналов.

На рисунке 7 показан типичный фрагмент схемы, который сразу выполнен в формате EDIF без использования Max+Plus II. Данная схема также реализует логику «ИЛИ», но при этом число используемых вентилях на порядок меньше. Поэтому и ячеек БМК, необходимых для данного фрагмента схемы, тоже будет на порядок меньше.

В процессе преобразования структуры EDIF происходит упрощение многовыходных вентилях, у которых часть выводов подключается на шину «земли» или питания. Поясним процесс упрощения. Пусть E(K) – исходный многовыходный элемент с K входами, а E(C) – преобразованный элемент с C входами. Если у элемента E(K) существует A входов, подключённых на «землю» или питание, и эти входы не влияют на логику элемента, то $C = K - A$. Если $A \geq K - 1$, то $C = 0$ или $C = 1$, и тогда элемент E(C) удаляется из схемы. Например, имеется элемент ИЛИ(10), у которого три входа подключены на «землю», тогда этот элемент можно преобразовать к ИЛИ(7).

Попутно происходит удаление элементов с одним входом, которые не влияют на логику работы схемы. Иными словами, преобразователь ConvChip производит «чистку» проекта от тех вентилях, которые практически не приводят к «гонкам» сигналов. Затем проект дважды моделируется в базисе БМК. Если преобразователь исправить «гонки» не в состоянии, то разработчик задаёт программе для анализа необходимые ветви схемы и исправляет «гонки» вручную. Как показывает практика, такая «чистка» сокращает исходную схему в полтора-два раза, что существенно экономит ресурсы кристалла БМК.

ОСНОВНЫЕ ОПЦИИ И НАСТРОЙКИ ПРОГРАММ CONVCHIP

Преобразователь ConvChip содержит более двухсот различных опций, которые позволяют настраивать перевод проекта схемы под конкретные требования. Рассмотрим основные группы настроек:

- 1) настройки формата конечных файлов;

- 2) настройки КИО;
- 3) настройки тестовых векторов;
- 4) настройки структуры схемы;
- 5) общие настройки конвертора.

Остановимся подробнее на каждом пункте.

1) При экспорте проекта схемы из Max+Plus II происходит переименование шин, состоящих из нескольких выводов, имён связей, имён элементов и т.д. Иными словами, имена, которые задал разработчик в системе проектирования, могут иметь другие названия в файле экспортных временных диаграмм тестовых векторов и в структуре схемы на языке EDIF. Ситуация усложняется ещё и тем, что формат имён выводов в системе моделирования «Невод», которая использует преобразованный проект в базисе БМК, имеет другие правила построения идентификаторов. Например, одной шине выводов может соответствовать четыре имени:

- исходное имя в Max+Plus II;
- имя в EDIF-файле;
- имя в тесте схемы;
- имя в системе моделирования.

Опции данной группы позволяют настроить тип выводов схемы, имён связей и элементов. В данную группу также входят функции, которые изменяют формат преобразования файлов в базис БМК.

- 2) Формирование временной сетки, обеспечивающей построение тестов с постоянным периодом подачи входных воздействий, т.е. по правилам КИО для БМК. В зависимости от выбранного типа микросхемы указывается напряжение питания, максимально возможное количество выводов, номера выводов «земли» и питания, данные для карты заказа и т.д.
- 3) Для исходной схемы в базисе ПЛИС разработчик может создавать несколько файлов тестов, причём все они могут содержать различные выводы. Настройки данной группы, в зависимости от требований к формату тестов, могут изменять систему исчисления тестовых элементарных проверок, доопределять в тесте незадаанные выводы, разбивать векторы на логически законченные блоки, вводить комментарии и т.д. Если в тесте заданы внутренние точки, то при необходимости преобразователь их

либо устраняет, либо преобразует к определённому виду.

4) Настройки данного типа позволяют модифицировать структуру схемы. Виды модификаций зависят от требований системы моделирования к структуре схемы, а также от возможных проблем, появляющихся после перевода проекта в базис БМК. Все модификации можно задать с помощью опций программы либо записать их на встроенном языке микроопераций. Например, требуется ввести цепи сброса триггеров в схеме. Для этого включается соответствующая опция либо для каждого триггера пишутся следующие команды:

```
#CreateElem
RESDFF:RESDFF_1;//объявление элемента сброса
#ReplacePin
DFF_1/PRN=RESDFF_1/R_PRNCIR;//замена выводов триггера DFF_1
#ReplacePin
DFF_1/CLRN=RESDFF_1/R_CLRNCTR;//замена выводов триггера DFF_1
#CreateNet
DFF_1/PRN,RESDFF_1/R_PRN;//создание новой связи для триггера
#CreateNet
DFF_1/CLRN,RESDFF_1/R_CLRN;//создание новой связи для триггера
#CreateElem
AMPLIFIER:AMPLIFIERLev1_1;//создание усилителя сигнала
#CreateNet AMPLIFIERLev1_1/Y,RESDFF_1/R_RES;//подключение усилителя
#CreateNet
PinResForDFF,AMPLIFIERLev1_1/IN1;
//создание вывода схемы для сброса
```

Здесь после знака # пишется соответствующая команда, после чего указываются её операнды.

Если модификация схемы стандартная, т.е. заданы режимы установки сброса триггеров, ввод блоков управления двунаправленными выводами, «чистка» проекта и многие другие, то соответствующие команды преобразователь генерирует автоматически, после чего их можно просмотреть. Если требуется нестандартная модификация, то разработчик пишет код на языке микроопераций, после чего запускается компилятор для создания исполняемого файла, кото-

рый в дальнейшем используется при преобразовании схемы.

5) К данной группе относятся функции поиска, позволяющие отыскивать фрагмент схемы между двумя заданными выводами для последующего анализа. Сюда относятся также и опции, необходимые для создания отчётов о преобразовании, типе выявленных ошибок в схеме и т.д.

Существует категория ошибок, которые не влияют на процесс моделирования преобразованной схемы в базисе БМК, но такие ошибки приведут к её неопределённому поведению после изготовления [5]. Примером таких ошибок является нарушение нагрузочных способностей элементов – на логику процесса моделирования такая ошибка не влияет, но реальная схема может работать некорректно. Для устранения подобных ошибок программа ConvChip автоматически находит «перегрузки» и формирует цепь усиления сигнала.

ЗАКЛЮЧЕНИЕ

Описанный преобразователь обладает разнообразными инструментальными средствами, позволяющими анализировать работу схемы. В настоящее время существуют пакеты программ, позволяющие выполнять рассматриваемый перевод схем, но ни один из них не имеет функций, необходимых для оптимального перевода проектов из базиса ПЛИС в базис БМК. Как правило, данные программы только переводят схему и тесты из одного формата в другой без каких-либо изменений, которые рассматривались в данной статье. К тому же они не имеют гибкой настройки под конкретную библиотеку БМК.

ЛИТЕРАТУРА

1. Точки Р., Уидмер Дж., Нил С. Цифровые системы. Теория и практика, 8-е изд. Пер. с англ. М.: Вильямс, 2004.
2. Bursky D. Advanced CPLD Architectures Challeng FPGA, Gas. Electronic Design. 1998. № 22. P. 78 – 86.
3. Камалов Д.А. Системы автоматизированного проектирования фирмы Altera Max+Plus II 2 М.: РадиоСофт, 2002.
4. Стещенко В.Б. ПЛИС фирмы Altera: проектирование устройств обработки сигналов. М.: ДОДЭКА, 2000.
5. Brown S.D., Fransis R., Rose J., Vranesic Z. Field-Programmable Gate Arrays. Boston: Kluwer Academic, 1992.

