

А. А. Можейко

(Новосибирск)

**ОБ ОДНОМ ПОДХОДЕ К ПРОБЛЕМЕ РЕИНЖЕНИРИНГА
ЦИФРОВЫХ СХЕМ НА ПРИМЕРЕ ПЕРЕПРОЕКТИРОВАНИЯ
ТЕЛЕКОММУНИКАЦИОННОЙ СХЕМЫ**

Введение. В Институте систем информатики им. А. П. Ершова СО РАН выполнена экспериментальная работа по перепроектированию некоторой цифровой телекоммуникационной схемы, реализованной на базовом матричном кристалле (БМК), в полностью заказную интегральную схему.

В данной работе подробно рассматривается процесс перепроектирования от момента постановки задачи до синтеза вентиляционной схемы включительно.

Постановка задачи. Требовалось перепроектировать некоторую телекоммуникационную цифровую схему, реализованную на БМК, в заказную интегральную схему и изготовить ее по одномикронной КМОП-технологии.

Исходная схема. Исходная схема спроектирована при помощи системы P-CAD 4.5 и представлена в виде трехуровневой иерархической логической схемы, состоящей из 331 элемента БМК (тактируемые триггеры с асинхронным сбросом или установкой (60); RS-триггеры (6); мультиплексоры двухвходовые, логические элементы (159); усилители, инверторы (106)).

Инструментальные средства. Перепроектирование предполагалось осуществить при помощи системы автоматизированного проектирования цифровых интегральных схем ALLIANCE v3 [1], разработанной в лаборатории MASI Парижского университета им. Пьера и Марии Кюри.

Система ALLIANCE v3 была установлена на сервере DEC Alpha AXP PC133 с операционной системой DEC OSF/1.

Метод перепроектирования. Вследствие ряда причин: 1) вынужденное использование учебной системы ALLIANCE v3 со всеми ее недостатками, присущими некоммерческой системе; 2) несоответствие библиотеки элементов БМК библиотекам стандартных ячеек системы ALLIANCE v3; 3) нерегулярная структура исходной схемы – выбран следующий подход к перепроектированию.

Сначала описать исходную схему на языке VHDL [2] (некотором подмножестве языка VHDL стандарта IEEE 1076, воспринимаемом автоматическим синтезатором логических схем системы ALLIANCE v3). Затем стандартными средствами системы ALLIANCE v3 реализовать схему в виде заказного кристалла, используя VHDL-описание перепроектируемой схемы в качестве входной информации для подсистемы автоматического синтеза логической схемы.

Таким образом, подход заключался в извлечении из исходной схемы синтезательного поведенческого описания и реализации перепроектируемой схемы на другой технологической платформе при помощи САПР СБИС путем автоматического синтеза по поведенческому описанию.

Преимуществом данного подхода является его универсальность, так как язык VHDL воспринимается всеми САПР СБИС. Возможное некоторое увеличение числа

логических элементов (как мы увидим ниже) в результате переноса на другую платформу следует отнести к недостаткам метода.

Альтернативным методом можно считать следующий: спроектировать вручную такой набор стандартных ячеек для системы ALLIANCE v3, который бы полностью соответствовал набору элементов БМК, затем разместить стандартные ячейки и произвести трассировку средствами системы ALLIANCE v3. Число элементов при этом не увеличивается, однако в целом данный метод является более дорогостоящим из-за ручного проектирования стандартных ячеек и перестает быть универсальным (его нельзя применить для переноса схем, содержащих, например, микросхемы средней степени интеграции или другие более крупные блоки).

Автоматизация процесса создания VHDL-описания схемы. Описание вручную на языке VHDL сложной мелкозернистой схемы с нерегулярной структурой представляет собой довольно трудоемкий процесс, кроме того, высока вероятность внесения ошибок.

Поэтому была разработана следующая технология автоматизации процесса создания VHDL-описания исходной схемы:

– стандартными средствами системы P-CAD (PC-NODES, PC-LINK) исходная иерархическая схема преобразуется в плоскую, состоящую только из элементов БМК;

– средством PC-FORM получается схема в формате P-CAD COMPONENT LIST;

– специально разработанный нами конвертор преобразует схему в формате P-CAD COMPONENT LIST в поведенческое VHDL-описание, воспринимаемое автоматическим синтезатором логических схем. (Это описание можно использовать как для моделирования работы схемы, так и в качестве входного описания для подсистемы автоматического синтеза логических схем системы ALLIANCE v3.)

Пример конвертации из P-CAD COMPONENT LIST в VHDL:

```
P-CAD COMPONENT LIST : XOR : UC000016 (Q0, Q1, UN000037)
VHDL : un000037 <= q0 XOR q1;
```

Проблема. Основная проблема состояла в том, что синтезатор системы ALLIANCE v3 использует только тактируемый триггер без какой-либо установки или сброса. Исходная же схема содержала триггеры с асинхронным сбросом или установкой.

Проблема усугублялась тем, что триггеры схемы синхронизировались от разных синхроимпульсов; схема содержала асинхронные двоичные счетчики и логические цепи через асинхронные входы триггеров.

Трансформация схемы. Анализ схемы показал, что есть возможность преобразования схемы в эквивалентный по поведению чисто синхронный вид без нарушения функции схемы, автоматического преобразования, при помощи набора некоторых правил локального преобразования.

Реализацию этих правил преобразования было решено выполнить с помощью языка Turbo Prolog 2.0, позволяющего записывать алгоритмы правил локального преобразования в виде, максимально приближенном к естественному (скорость исполнения программы для данных размеров схемы нас устраивала).

С этой целью исходная схема (P-CAD COMPONENT LIST) при помощи специального конвертора преобразовывалась в некоторое Пролог-представление, где элементы схемы отображались в виде фактов базы данных Пролога.

Пример конвертации из P-CAD COMPONENT LIST в Turbo Prolog:

```
P-CAD COMPONENT LIST : XOR : UC000016 (Q0, Q1, UN000037)
Turbo Prolog : xor ("q0", "q1", "un000037").
```

Затем это промежуточное представление обрабатывалось при помощи некоторых правил локального преобразования, в результате чего получалась эквивалентная по поведению схема, содержащая только синхронные триггеры.

Пример правила локального преобразования:

```
rule : - dff (D, C, Q),
      dff (X, T, C), retract (dff (D, C, Q)),
      assertz (dff (M, T, Q)),
      assertz (mux (Q, D, A, M)),
      assertz (not (X, N)),
      assertz (and (N, C, A)), fall.
```

Далее другой конвертор преобразовывал Пролог-представление трансформированной схемы в синтезируемое поведенческое VHDL-описание, которое можно подавать как входное логическому синтезатору ALLIANCE v3.

Пример конвертации из Turbo Prolog в VHDL:

```
Turbo Prolog : xor ("q0", "q1", "un000037"),
VHDL : un000037 <= q0 XOR q1;
```

Технологическая цепочка перепроектирования. Таким образом, была разработана технологическая цепочка для автоматизированного перепроектирования некоторых цифровых схем, содержащих запоминающие элементы со смешанным синхронно-асинхронным режимом работы:

Исходная схема (P-CAD COMPONENT LIST)
Конвертация в Пролог-представление
Пролог-представление схемы
Трансформация в синхронную схему
Пролог-представление схемы
Конвертация в VHDL-описание
VHDL-описание схемы
САПР СБИС ALLIANCE v3
Результат (формат CIF)

1. Средствами системы P-CAD иерархическая схема преобразуется в плоскую, выделяется формат P-CAD COMPONENT LIST.

2. Данный формат конвертируется в некоторый Пролог-представление схемы, где элементы схемы представлены как набор фактов базы данных Пролога.

3. Средствами языка Пролог при помощи набора правил преобразования схема преобразуется из смешанной синхронно-асинхронной в синхронную, эквивалентную по поведению.

4. Преобразованная эквивалентная схема автоматически конвертируется в синтезируемое поведенческое VHDL-описание, которое можно использовать как входное для автоматического синтеза вентиляционной схемы в соответствии с библиотекой стандартных ячеек выбранной технологии.

5. Далее схема реализуется стандартными средствами системы ALLIANCE v3 в виде файла в формате CIF, используемого для изготовления заказной СБИС.

Верификация схемы. Исходная схема сопровождалась набором тест-векторов (8000), при помощи которого в статическом режиме (без учета задержек) проверялась правильность функционирования схемы.

Этого нам достаточно, так как задержки перепроектированной схемы получались заведомо значительно меньшими, и известно, что исходная схема работоспособна.

Исходный набор тест-векторов был конвертирован в файл соответствующего формата системы ALLIANCE v3 и использовался для верификации поведения схемы на всех этапах перепроектирования.

Верификация осуществлялась с помощью подсистемы ASIMUT системы ALLIANCE v3.

При верификации использовалось поведенческое VHDL-описание схемы в качестве модели.

Проверялись следующие поведенческие VHDL-описания:

экстрагированное из исходной схемы;

поведенческое описание эквивалентной трансформированной схемы;

экстрагированное из топологии перепроектированной схемы.

Результат перепроектирования. Исходная схема (331 элемент БМК) была конвертирована в Пролог-представление (331 элемент), частично оптимизирована (269 элементов, удалены буферы и усилители), трансформирована в чисто синхронную схему (403 элемента), конвертирована в поведенческое VHDL-описание (1119 строк), глобально оптимизирована с помощью подсистемы LOGIC системы ALLIANCE на уровне поведенческого описания (599 строк), синтезирована с помощью подсистемы LOGIC системы ALLIANCE в вентиляционную схему (541 стандартная ячейка) и реализована в виде CIF-файла (566 Мбайт), пригодного для реализации заказного кристалла размером 2,7×2,6 мм по одномикронной КМОП-технологии.

Некоторый неприятный момент (541 стандартная ячейка перепроектированной схемы против 331 элемента БМК исходной схемы) на самом деле не является столь существенным, как кажется на первый взгляд: во-первых, увеличение числа стандартных ячеек по сравнению с числом элементов БМК исходной схемы – это плата за несоответствие библиотек, а во-вторых, это вызвано несовершенством библиотеки стандартных ячеек некоммерческой учебной системы ALLIANCE v3.

Кроме того, следует учитывать тот момент, что стандартные ячейки сильно отличаются друг от друга по площади и их общее число на самом деле не дает точной характеристики площади схемы.

Заключение. Проведенный эксперимент демонстрирует возможность перепроектирования цифровых схем путем экстракции из исходной схемы синтезального поведенческого VHDL-описания и последующего автоматизированного синтеза схемы на другой технологической платформе.

Таким образом, VHDL-описание цифровой схемы выступает не только как универсальное средство абстрактного представления схем, но и как средство переноса схем или даже систем с одной технологической платформы на другую.

СПИСОК ЛИТЕРАТУРЫ

1. ALLIANCE v3 (доступна по ftp на сервере ftp.ibp.fr [132, 227, 60, 2] в директории /ibp/softs/masi/alliance).
2. VHDL Language Reference Manual // Draft Standard 1076/B: CAD Language Systems, Inc., 1987.

Поступило в редакцию 8 декабря 1997.