

УДК 658.512.011.56:681.325.65

ОСНОВНЫЕ ПОДХОДЫ К АВТОМАТИЗИРОВАННОМУ ПЕРЕПРОЕКТИРОВАНИЮ ЛОГИЧЕСКИХ СХЕМ

Калюта А.Г.

Институт Технической Кибернетики НАН Беларуси

Введение

Средства автоматизации в настоящее время находят применение практически на всех этапах проектирования, контроля и моделирования электронных систем. Конкурентная борьба, а также большая сложность современных СБИС сделало необходимым применение развитых средств автоматизации проектирования. Сегодня проблематика проектирования интегральных схем стала настолько широка, что общая задача проектирования разбивается на отдельные подзадачи, решение которых постоянно исследуется и совершенствуется. Одной из таких проблем является и проблема перепроектирования. В данной статье рассматриваются основные подходы к задаче перепроектирования.

1. Суть проблемы перепроектирования

Достаточно формализованным и проработанным при проектировании является подход, когда на входе системы имеется высокоуровневое исходное описание проектируемой схемы, описания используемой технологии, описание проектных ограничений и стоимостных функций. Таким образом, имеет место преобразование типа «описание - схема». Однако на практике такой подход применяется лишь к построению схем небольшой сложности.

При проектировании схем большой сложности, проектируемая схема разбивается с помощью специальных методов декомпозиции на достаточно простые составляющие, при проектировании которых и применяются преобразования типа «описание - схема». Однако возникает задача оптимального объединения спроектированных составляющих, при решении которой может потребоваться перепроектирование отдельных частей схемы. При перепроектировании осуществляется преобразование типа «схема - новая схема», где новая схема должна обладать некоторыми новыми качествами, определяемыми критериями перепроектирования. В качестве критериев перепроектирования могут выступать, например, ограничения по производительности, потребляемой мощности[3], площади занимаемого кристалла или требований удоботестируемости.

2. Перепроектирование программно-аппаратных систем

Кроме перепроектирования логических схем, термин «перепроектирования» употребляется и в отношении сложных аппаратно-программных систем, жизненный цикл которых в настоящее время весьма ограничен. Разработка столь сложных аппаратно-программных комплексов занимает большой промежуток времени, поэтому перед разработчиками стоит задача скорейшей модернизации ранее выпущенных систем, а так же оперативное исправление недостатков, обнаруженных уже при эксплуатации системы или на поздних этапах проектирования [4]. Перепроектирование аппаратно-программных систем предполагает использование аппаратно-программного исполнения предыдущей версии системы для получения новой версии посредством изменения программной части [1]. Определение возможности перепроектирования таких систем предполагает последовательное выявление изменений в спецификации, определение временных ограничений из предыдущего аппаратного исполнения, которые должны быть согласованы с новым программным исполнением. Далее производятся изменения в программной части с

учетом временных ограничений, накладываемых аппаратной частью и проверяется соответствуют ли новые характеристики производительности тем, которые были заложены в спецификации. При неудовлетворительном результате производится либо корректировка исходных данных перепроектирования, либо параметров перепроектирующей системы.

3. Оптимизация и изменение базиса

Частным случаем перепроектирования является задача оптимизации логических схем.

Глобальная оптимизация[2] предполагает упрощение логических уравнений на основе глобального подхода, который позволяет производить значительное преобразование схемы в соответствии с выбранными критериями оптимизации и не зависит от базовой технологии. После глобальной оптимизации необходимо провести технологически-ориентированные преобразования схемы, которые должны обеспечивать наиболее компактную и эффективную реализацию полученной логической схемы на основе базовой технологии.

Подход, основанный на правилах[2], состоит в осуществлении иерархически организованной последовательности локальных преобразований над логической схемой в соответствии с выбранным критерием оптимизации. На этапе локальных преобразований осуществляется перевод описания схемы на уровень вентилей и при помощи некоторого набора правил обеспечивается основная оптимизация схемы. Производится учет задержек, и преобразования выполняются в том случае, если они обеспечивают улучшение технических или габаритных показателей, которые являются критериями оптимизации. Качество оптимизации существенно зависит от порядка применения правил.

Перепроектирование так же применяется при переводе логических схем из одной базовой технологии в другую, обеспечивающую большую степень интеграции или производительность. Естественно, что прямое

замещение вентилях одного базиса вентилями другого базиса является весьма неэффективным. Поэтому здесь возможно применение как методов глобальной, так и локальной оптимизации. 5.

Заключение

На сегодняшний день наиболее исследованной областью применения перепроектирования является глобальный подход к оптимизации логических схем при помощи аппарата булевых функций. Проблематика оптимизации с использованием локального подхода исследована несколько хуже и поэтому представляет определенный интерес. И совсем малоизученной областью является задача обратного синтеза, когда новая схема реализует функции исходной схемы, но построена на другом технологическом базисе и использует схемную реализацию элементов нового базиса. Задача обратного синтеза находит свое применение в системах, осуществляющих автоматическое восстановление логической схемы непосредственно по изображениям кристалла. В таких системах она решает проблему перепроектирования логической схемы с учетом новых и существовавших прежде схемных решений на основе некоторого технологического базиса.

Литература

1. C.Coelho, J.Yang, V.Mooney and G.De Micheli. Redesigning Hardware-Software Systems. Third International Workshop on Hardware/Software Co-Design, September 1994.
2. A.Newton, A.Sangiovanni-Vinsentelli. CAD Tools for ASIC Design, Малый тематический выпуск по специализированным интегральным схемам. Август 1996.
3. R.Rahar, F.Somenzi. Boolean Techniques for Low-Power Driven Re-Synthesis, DECE, University of Colorado, Boulder.

4. H.J. Eikerling, W.Hardt, J.Gerlach, W.Rosenstiel. A Methodology for Rapid Analysis and Optimization of Embedded Systems.

ТЕХНОЛОГИЯ СОЗДАНИЯ АДАПТИРУЕМЫХ АРМ.

Косинов Г.П., Шибeko В.Н

Гомельский политехнический институт им. Сухого П.О.

Внедрение новой компьютерной техники и программного обеспечения в нашей стране происходит в больших масштабах. Причем на момент внедрения программное обеспечение полностью отвечает требованиям законодательства и конечных пользователей и поставляется в виде набора баз данных, файлов инструкций и исполняемого кода программы. Руководство предприятия и пользователи зачастую не подозревают о том, что даже небольшие изменения должны производиться над исходным текстом программы, который часто просто отсутствует. Горячая телефонная линия в данной ситуации не поможет, а фирма, выпустившая программное обеспечение, реагирует на такие изменения крайне неохотно, так как программное обеспечение разрабатывается универсальное и в небольших клиентах они просто не заинтересованы. Пользователь просто отказывается от такого программного продукта. Если же разработчик отдает текст программы, то при нынешней ситуации с авторскими правами, его АРМ просто никто не купит.

Поэтому наряду с общеизвестными требованиями к внедряемым программным продуктам - надежность, эффективность, сопровождаемость, мы выделяем новое требование - модифицируемость.

Модифицируемость - это способность пользователя программного обеспечения без участия разработчика и изменения исходного текста программы изменять поток и структуру входных данных, алгоритм расчета и организацию выходных потоков информации.