

**Информационные технологии
проектирования электронных
устройств
Часть 1**

Лекция 5

Лекция 5

Алгоритмы компоновки конструктивных модулей различных уровней иерархии

- 1 Компоновка схем электрических
- 2 Компоновка конструктивных элементов по коммутационным платам
- 3 Классификация алгоритмов компоновки

Вопрос 1 Компоновка схем электрических

Типы задач компоновки

Компоновка – процесс перехода от логическо-функционального или схемного описания устройства к конструктивному,

Т.е. это процесс распределения функциональных элементов схемы по группам, соответствующим модулям различного иерархического уровня

- 1) **Типизация** - разбиение функциональной схемы на части, номенклатура которых известна
- 2) **Покрытие** - преобразование функциональной схемы в схему соединений элементов, номенклатура которых известна
- 3) **Разрезание** - разбиение схемы электрической принципиальной на части, минимально связанные между собой

Математическая формулировка покрытия

Исходные данные:

1. функциональная схема устройства
2. схемы типовых конструктивных элементов используемого набора модулей.

Необходимо:

найти такое распределение логических функций покрываемой схемы по отдельным конструктивным элементам, при котором достигается экстремум целевой функции.

Показатели качества:

- 1) *суммарная стоимость модулей, покрывающих схему;*
- 2) общее число модулей, необходимое для реализации схемы;
- 3) число типов используемых модулей;
- 4) число межмодульных соединений и т.д.

Математическая формулировка покрытия

Ограничения:

требования на совместную или раздельную компоновку в едином конструктивном модуле элементов функциональной схемы, связанные с обеспечением нормального теплового режима, помехозащищенности и простоты диагностики.

Для решения задачи используют методы целочисленного линейного программирования.

необходимо минимизировать целевую функцию

$$F(X) = \min \sum_{i=1}^n C_i X_i$$

где C_i - стоимость i -го конструктивного модуля, X_i - число модулей i -го типа, которые необходимы для покрытия схемы.

Математическая формулировка покрытия

При ограничениях:

- число логических функций любого i -го типа, входящих во все покрывающие модули, должно быть не меньше общего числа элементов соответствующего типа в реализуемой схеме;
- любой модуль используется только полностью, независимо от числа задействованных в нем компонентов

Покрытие выполняют в 2 этапа:

- 1) *Предварительного*
- 2) *Окончательного*

Обобщенный алгоритм покрытия

- 1 Выделяют из заданной функциональной схемы подсхемы (группы максимально связанных между собой логических элементов)
- 2 Перебирают все или достаточно большое их число и проверяют на совпадение логических функций элементов подсхем и компонентов модулей используемого набора.
- 3 Подсхема закрепляется за тем модулем, в состав которого входит наибольшее количество ее логических функций.
- 4 Процесс продолжается до полного распределения элементов функциональной схемы по модулям.

Для улучшения полученного результата при окончательном покрытии осуществляют парные перестановки однотипных логических элементов различных ячеек. Критерием удачной перестановки является увеличение внутренних связей ячеек.

Вопрос 2 Компоновка конструктивных элементов по коммутационным платам

Постановка задачи

Необходимо:

распределить элементы схемы (модулей предыдущего уровня) по коммутационным платам (коммутационным пространствам) данного уровня иерархии.

Основной критерий оптимальности :

минимизация числа межмодульных связей:

- повышение надежности схем (за счет уменьшения числа разъемных соединений),
- уменьшения влияния наводок и времени задержки сигнала в цепях (вследствие минимизации суммарной длины соединений),
- упрощения конструкции и
- повышения технологичности разрабатываемого устройства.

Математическая формулировка покрытия

Электрическую схему интерпретируют ненаправленным мультиграфом $G(X, U)$, в котором каждому конструктивному элементу (модулю) ставят в соответствие вершину мультиграфа X , а электрическим связям схемы - его ребра U .

Требуется «разрезать» мультиграф на отдельные подграфы $G_1(X_1, U_1), G_2(X_2, U_2), \dots, G_k(X_k, U_k)$ так, чтобы число ребер, соединяющих эти куски, было минимальным:

$$\min \sum_{i=1}^k \sum_{j=1}^k U_{ij}, i \neq j$$

U_{ij} - множество ребер, соединяющих подграфы $G_i(X_i, U_i)$ и $G_j(X_j, U_j)$

Конструктивные ограничения в задачах компоновки

- число подграфов k ;
- число вершин в каждом из подграфов (определяется числом конструктивных элементов, которые необходимо разместить на коммутационной плате, пластине БИС, подложке БГИС и т. д.);
- максимальное число внешних связей каждого отдельно взятого подграфа (определяется количеством контактов используемого разъема, числом выводов стандартного корпуса БИС или БГИС);
- требование на отдельную компоновку отдельных вершин в различных подграфах (обусловлено конструктивными соображениями и условиями электромагнитной совместимости).

Вопрос 3 Классификация алгоритмов компоновки

Классификация алгоритмов КОМПОНОВКИ



Классификация алгоритмов КОМПОНОВКИ

3.1 Алгоритмы, использующие методы целочисленного программирования

для устройства реальной сложности фактически
малореализуемы на ЭВМ

3.2 Последовательные алгоритмы

- сначала по определенному правилу выбирают первую вершину графа,
- затем осуществляют последовательный выбор вершин (из числа нераспределенных) и присоединение их к формируемому подграфу.

После образования первого подграфа переходят ко второму и т.д. до получения желаемого разрезания исходного графа

3.2 Последовательные алгоритмы

- 1) В графе $G(X, U)$ находят вершину $x_i \in X$ с минимальной локальной степенью ρ :

$$\rho(x_i) = \min_{x_j \in X} \rho(x_j)$$

- 2) Из подмножества вершин, смежных с вершинами формируемого подграфа $G_1(X_1, U_1)$, выбирают ту, которая обеспечивает минимальное приращение связей подграфа с еще нераспределенными вершинами. Данную вершину x_j включают в $G_1(X_1, U_1)$, если не происходит нарушения ограничения по числу внешних связей подграфа
- 3) Процесс продолжается до тех пор, пока множество X_1 не будет содержать N элементов либо присоединение очередной нераспределенной вершины x_j к подграфу $G_1(X_1, U_1)$ не приведет к нарушению ограничения по числу внешних соединений подграфа.

3.2 Последовательные алгоритмы

Достоинства:

- прост,
- легко реализуется на ЭВМ,
- позволяет быстро получить решение задачи компоновки

может эффективно применяться и при наличии ограничения на совместную компоновку отдельных вершин графа. В этом случае каждая такая вершина жестко закрепляется за определенным подграфом и формирование очередного подграфа начинается с непосредственного выбора этой вершины в качестве исходной.

3.3 Итерационные алгоритмы

- начальное разрезание графа на части выполняют произвольным образом;
- оптимизация компоновки достигается парными или групповыми перестановками вершин графа из различных подграфов;
- перераспределения вершин заканчивают при получении локального экстремума целевой функции, удовлетворяющего требованиям разработчика.

Процесс оптимального разрезания графа на части заключается в отыскании на каждой итерации таких парных перестановок строк и столбцов матрицы смежности A , при которых максимизируется сумма элементов в диагональных подматрицах, что равносильно минимизации числа соединительных ребер.

3.3 Итерационные алгоритмы

Недостатки:

- 1) Оптимальность решения в значительной степени зависит от того, насколько удачно было произведено начальное разделение графа.
- 2) Требуют больших затрат машинного времени, чем последовательные алгоритмы.

Достоинства:

- Обеспечивают высокое качество решения задачи.
- Для сокращения числа итераций обмена вершин между кусками в смешанных алгоритмах для получения начального «разрезания» графа возможно применение последовательные методы формирования его кусков.

3.4 Смешанные алгоритмы

для получения начального варианта «разрезания» используется алгоритм последовательного формирования подграфов; дальнейшая оптимизация решения осуществляется перераспределением вершин между отдельными подграфами.

3.5 Алгоритмы, основанные на методе ветвей и границ

Последовательное разбиение множества допустимых планов задачи целочисленного программирования L на подмножества.

Процесс разбиения продолжается до тех пор, пока каждое подмножество не будет представлять собой точку в многомерном пространстве.

3.5 Алгоритмы, основанные на методе ветвей и границ

Процесс последовательного сечения исходного множества решений L гиперплоскостями на подмножества представляет собой построение дерева ветвлений с соответствующими подмножествами. Чтобы уменьшить объем перебора, производят оценку образуемых подмножеств по наибольшему (наименьшему) значению для задач максимизации (минимизации) целевой функции. Это позволяет на каждом шаге исключать из рассмотрения те подмножества, для которых ожидаемое значение целевой функции оказывается наихудшим.

Метод эффективен для решения частично целочисленных задач, содержащих небольшое число целочисленных переменных, так как в противном случае число итераций возрастает и для реализации алгоритма потребуются значительные затраты машинного времени.

*Вопросы по прочитанному
материалу?*

Спасибо за внимание!