

СТРУКТУРНЫЙ АНАЛИЗ ФУНКЦИОНАЛЬНЫХ СХЕМ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

В. К. ПОГРЕБНОЙ

(Представлена научно-техническим семинаром кафедры АСУ
и лаборатории управления)

Исследование проблемы унификации при проектировании вычислительных устройств обнаруживает многообразие возможных подходов к решению данной проблемы. Это многообразие, с одной стороны, порождается наличием большого количества условий, каждое из которых в отдельности либо их комбинация в конкретной ситуации приобретают первостепенное значение и тем самым определяют подход к решению проблемы. С другой стороны, отсутствие единой методологии и какой-либо математической теории решения данной проблемы и в конкретных условиях порождает многообразие возможных методов и алгоритмов решения задач унификации.

В связи с этим необходимо отметить, что в настоящее время любые удачные или менее удачные практические попытки решения конкретных задач унификации являются актуальными, так как наряду с рационализацией процесса проектирования и непосредственного улучшения характеристик вычислительных устройств эти попытки позволяют накопить материал, необходимый для разработки методологии и математической теории решения задач унификации.

Вместе с тем, несмотря на наличие такого многообразия в подходах и методах решения задач унификации, уже на данной стадии исследований намечаются пути к разработке более универсальных, единых подходов и методов. Так, следуя уже намечившемуся в ходе исследований принципу решения задач унификации путем проектирования унифицированного набора блоков [1, 2, 3], можно, независимо от физической природы проектируемых систем и уровня иерархии, на котором решается задача унификации, использовать одну и ту же схему решения задачи. Среди основных этапов такой схемы можно выделить следующие:

- построение графов взаимосвязей объектов систем соответствующего уровня иерархии;
- структурный анализ полученных графов с целью определения часто повторяющихся (типовых) подграфов (типовых структур);
- формирование из полученных типовых структур унифицированных блоков;
- разработка методов реализации систем блоками унифицированного набора с минимальной избыточностью.

Построение графов взаимосвязей объектов систем зачастую не вызывает принципиальных трудностей. Однако как построение, так и

способ представления данного графа в ЭВМ оказывает существенное влияние на специфику и эффективность алгоритма структурного анализа. В качестве примера построения графа и способа представления его в ЭВМ для самого низкого уровня иерархии вычислительного устройства (уровень логических элементов), реализованных на микромодулях, можно рассматривать способ, предложенный в [2]. Здесь же выделяются и обсуждаются основные требования к способу описания функциональных схем вычислительных устройств на данном уровне иерархии. Указанные требования носят достаточно общий характер и поэтому приемлемы в любых конкретных условиях. Однако способы описания, отвечающие данным требованиям, в каждом конкретном случае могут существенно отличаться.

Структурный анализ полученного графа с целью выделения повторяющихся подграфов (типовых структур) приводит к необходимости решения ряда сложных задач, например задачи идентификации графов. Следует отметить, что результаты проведенных исследований по решению задачи тождественного вхождения типовых структур в схемы вычислительных устройств позволяют сделать вывод о возможности эффективного использования имеющихся алгоритмов [1, 2] для решения данной задачи применительно к типовым структурам, содержащим до 10 элементов. Кроме того, намечившиеся пути совершенствования

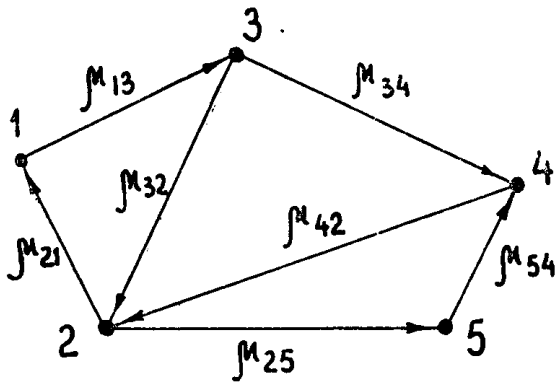


Рис. 1.

данных алгоритмов позволяют надеяться на успешное решение задачи для типовых структур, содержащих до 20 элементов. Возможность выделения типовых структур такого размера с помощью указанных алгоритмов позволит эффективно использовать их при решении многих практических задач, например, при определении рациональной номенклатуры больших интегральных схем и их использовании.

В данной работе рассматриваются задачи структурного анализа функциональных схем вычислительных устройств в условиях, когда какой-либо перечень априори заданных типовых структур (ТС) отсутствует. Заметим, что если некоторое множество ТС задано, то они могут быть выделены в схемах по алгоритму, изложенному в [2]. Об использовании результатов выделения таких ТС будет сказано ниже. Здесь лишь заметим, что в процессе выделения определяются частоты использования данных ТС, а функциональная схема никаких изменений не претерпевает, то есть используется второй вариант алгоритма выделения [2].

Представление функциональной схемы в ЭВМ осуществляется в соответствии с методикой, изложенной в [2]. Для удобства использования в последующих алгоритмах структурного анализа список связей графа схемы $G = (P, U)$ преобразуется программным путем в список вершин p_i с указанием прямых F_{p_i} и обратных $F_{p_i}^{-1}$ отображений, значений полустепеней исхода α_i и захода β_i , и соответствующих значений кодовых номеров [2]. Пример списка вершин с указанием перечисленных выше данных для графа, приведенного на рис. 1, представлен табл. 1.

Для каждой i -й вершины в таблице указывается мощность m_i , которая характеризует количество конструктивных единиц либо их частей, реализующих данную вершину. Для случая, когда вершины p отождествляют элементы интегральных схем [2], мощность m_i характеризует часть модуля соответствующей интегральной схемы.

Таблица 1

P_i	α_i	β_i	F_{P_i}	μ_{ij}	$F_{P_i}^{-1}$	μ_{ji}	m_i
P_1	1	1	P_3	μ_{13}	P_2	μ_{21}	m_1
P_2	2	2	P_1, P_5	μ_{21}, μ_{25}	P_3, P_4	μ_{32}, μ_{42}	m_2
P_3	2	1	P_2, P_4	μ_{32}, μ_{34}	P_1	μ_{13}	m_3
P_4	1	2	P_2	μ_{42}	P_1, P_5	μ_{34}, μ_{54}	m_4
P_5	1	1	P_4	μ_{54}	P_2	μ_{25}	m_5

Если вершины p_i отождествлены некоторым структурам (узлам вычислительных устройств), то мощность таких вершин показывает количество модулей, необходимых для реализации соответствующих структур.

Основной момент, характеризующий предлагаемый метод структурного анализа и определяющий пути его реализации, состоит в том, что анализ проводится путем формирования множества всех возможных структур с ограниченным диапазоном мощностей и выбора среди них оптимальной совокупности по определенному критерию. Такой подход можно характеризовать как точный метод решения, так как, осуществляя перебор множества сформированных структур в соответствии с принятой их оценкой, метод позволяет получить наилучшее решение, то есть наиболее качественно провести структурный анализ. Вместе с тем точный метод может быть использован лишь для анализа небольших схем. Поэтому такой подход в основном представляет только методологический и теоретический интерес.

При анализе реальных схем вычислительных устройств, особенно на уровне элементов, оказывается возможным использование лишь приближенных методов. При получении приближенного или рационального решения в данном случае, впрочем как и при решении других задач, возможны два подхода. Первый заключается в исследовании рассматриваемой задачи в оптимизационной постановке и получении приближенного решения. Второй подход состоит в разработке некоторого эвристического алгоритма, позволяющего получить рациональное решение. Рассматриваемый ниже метод позволяет получать приближенное решение через оптимизационную постановку. При этом выделяются следующие основные этапы решения задачи структурного анализа:

- формализация оценки качества структуры;
- формирование множества структур заданного диапазона мощностей;
- вычисление оценки качества сформированных структур;
- выбор оптимальной совокупности типовых структур.

Формализация оценки качества структуры

Прежде чем перейти к непосредственному изучению факторов, влияющих на оценку структур, проведем качественный анализ влияния

результатов структурного анализа на результат решения всей задачи, заключающейся в достижении максимально возможного уровня унификации. Не вызывает сомнения тот факт, что сокращение номенклатуры ТС, используемых для проектирования УНБ, способствует сокращению его размеров. В то же время стремление покрыть большую часть элементов схем типовыми структурами минимальной номенклатуры равнозначно увеличению частот использования соответствующих ТС.

Таким образом, частоты использования ТС в анализируемых схемах во многом определяют качество решения задачи унификации. Вместе с тем частота использования ТС зависит от их мощности. Понятно, что чем менее мощными будут выбраны ТС, тем выше будет их частота. Проектирование УНБ на основе более мощных структур положительно сказывается на ряде важных характеристик, таких как соотношение числа выводов и модулей блока, коэффициент использования блоков и ряд других. Поэтому важно, чтобы ТС были достаточно мощными.

Анализируя сказанное выше, можно выделить два основных фактора для оценки качества структур — мощность и частоту использования. Определение мощности структуры было дано выше. Мощность i -й ТС обозначим через M_i . Величина M_i легко определяется как сумма мощностей m_j элементов p_j , входящих в множество A_i элементов i -й ТС

$$M_i = \sum_{p_j \in A_i} m_j. \quad (1)$$

Понятие частоты использования применительно к оценке качества структур требует уточнения. Дело в том, что частота использования, полученная в результате структурного анализа, зависит от других структур. Частоту, с которой встречается ТС в схемах, подсчитанную независимо от других ТС, будем называть повторяемостью. Повторяемость i -й структуры g_i характеризует максимально возможную частоту использования структуры i -го вида. Заметим, что по значению величины g_i нельзя судить о значении частоты использования, которое будет получено после анализа. Может оказаться, что структура с высоким значением g_i получила низкое или нулевое значение частоты использования. Это лишь означает, что элементы, составляющие структуры i -го вида вошли в другие структуры. Повторяемость характеризует, таким образом, потенциальную возможность структуры иметь соответствующее значение частоты использования.

Общая оценка структуры i -го вида может быть получена как взвешенная сумма отнормированных значений мощности m_i и повторяемости g_i

$$C_i = \beta_m m_i + \beta_g g_i, \quad \beta_m + \beta_g = 1, \quad (2)$$

где C_i — оценка структуры i -го вида;

m_i — нормированное значение мощности структуры i -го вида;

g_i — нормированное значение повторяемости структуры i -го вида;

β_m, β_g — значения весовых коэффициентов, характеризующих степень влияния мощности и повторяемости соответственно на величину оценки.

Значения весовых коэффициентов предварительно могут быть определены экспертным методом и уточнены в ходе эксперимента. Отнормированные значения мощности и повторяемости m_i и g_i имеют математическое ожидание, равное нулю, и дисперсию, равную единице.

Оценка C_i ставит сравниваемые структуры в равные условия. Действительно, если структура имеет малую мощность, то она, как правило, будет иметь достаточно высокую повторяемость и наоборот. В ходе структурного анализа введенная оценка позволяет отсеивать неперспек-

тивные структуры и не учитывать их при решении задачи выбора оптимальной совокупности типовых структур. Основным недостатком данной оценки состоит в том, что она может быть подсчитана только после формирования полного множества структур по всем анализируемым схемам. Тем самым исключается возможность отсеивания структур непосредственно в ходе их формирования.

Обработка структур полного множества потребует значительного объема вычислений. Вместе с тем необходимо заметить, что формирование оценки, позволяющей отсеивать неперспективные структуры в ходе их формирования, связано, по-видимому, с трудностями принципиального характера. Поэтому в последующем изложении основное внимание уделено вопросам повышения эффективности алгоритмов формирования и оценки структур, а не поиску более эффективных оценок.

Формирование полного множества структур

Рассматриваемый ниже алгоритм позволяет формировать полное множество структур в заданном диапазоне уровней (числе элементов, составляющих структуру). При этом алгоритм позволяет исключать дублирующие структуры в ходе их формирования, а в памяти хранить в основном только исходный граф схемы. Алгоритм построен таким образом, что позволяет последовательно для каждой вершины графа проводить анализ области графа вокруг данной вершины. Анализ осуществляется распространением «волны» вокруг вершины до получения структур с заданными границами уровней.

Распространение такой «волны» осуществляется по определенным правилам последовательно вокруг каждой вершины графа. Этот процесс в последующем будем отождествлять с построением дерева структур. Вершинами дерева структур являются вершины анализируемого графа, а связи отражают принадлежность связываемых вершин к некоторой структуре.

Каждой ветви дерева с началом в исходной вершине соответствует некоторая структура. Общее число структур определяется числом вершин дерева за исключением исходной. Вершины дерева и соответствующие им структуры можно распределить по уровням. Принадлежность структуры к некоторому, например, j -му уровню указывает на наличие в соответствующей ветви дерева j вершин.

Для формализации процесса построения дерева структур введем понятие цепи отображений. Цепью отображений будем называть такое дерево, у которого на каждом уровне всего лишь одна вершина имеет непустое отображение. Цепь отображений является основным элементом алгоритма формирования множества структур, который заключается в последовательном построении цепей отображений дерева структур. Рассмотрим подробнее процесс перехода от одной цепи отображений к другой. Выделим в цепи отображений последовательность вершин, которые имеют непустые отображения. На рис. 2, а последовательность таких вершин выделена жирной линией.

Множество вершин отображений $F_{p_i}^j$, расположенных левее вершины p_i с непустым отображением, включающее эту вершину, обозначим через F_{p_i, p_j}^j . Цепь таких отображений приведена на рис. 2, б. По аналогии вершины, расположенные справа от вершин с непустым отображением, обозначены через F_{p_i, p_j}^j . Множества таких вершин показаны на рис. 2, в. Здесь необходимо отметить, что последовательность вершин в отображениях исходной цепи и вновь формируемых отображе-

ниях может быть произвольной. Однако в последующем при переходе от одной цепи отображений к другой эта последовательность должна оставаться неизменной.

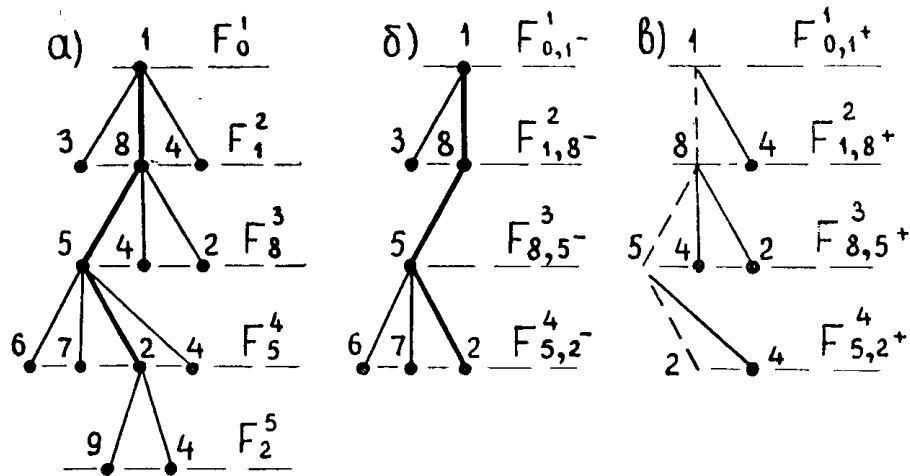


Рис. 2.

Формирование отображений осуществляется на основе графа схемы и имеющейся цепи отображения. При формировании отображения $F_{p_e}^f$ на f -м уровне вершине p_e предварительно формируется два множества A^f и B^f :

$$A^f = \{F_{p_e}^{-1} \cup F_{p_e} \cup F_{p_k, p_e}^{f-1}\}, \quad (3)$$

$$B^f = \{F_{0, p_i}^1 \cup F_{p_i, p_j}^2 \cup \dots \cup F_{p_k, p_e}^{f-1}\}. \quad (4)$$

Отображение $F_{p_e}^f$ определяется на основе множеств A^f и B^f .

$$F_{p_e}^f = \{A^f/B^f\}. \quad (5)$$

Последовательность построения отображений организуется таким образом, чтобы цепь отображений обходила дерево структур, перемещаясь слева направо. При этом вновь формируемое отображение записывается на место отображения максимального уровня предшествующей цепи отображений. Таким образом, в памяти ЭВМ всегда оказывается достаточным хранить одну цепь отображений, а не все дерево структур. После построения очередного отображения f -го уровня соответствующие ему структуры запоминаются в f -м массиве M_f . Максимальный уровень структур, который необходимо получить в процессе анализа, должен указываться перед построением дерева структур, так как само дерево не запоминается, и, следовательно, последующее наращивание без его повторного построения оказывается невозможным. Запись структур в соответствующие массивы осуществляется в форме табл. 1. Необходимые при этом данные берутся из описания исходного графа анализируемой схемы.

Вычисление оценки качества сформированных структур

Выше указывалось, что оценка качества структуры осуществляется по двум факторам: мощность структуры i -го вида m_i и повторяемость g_i .

При наличии значений величин m_i , g_i для структуры i -го вида оценка S_i определяется по выражению (2). Что касается определения величины m_i , то здесь не возникает каких-либо затруднений. Величина m_i вычисляется по выражению (1). Значительные трудности вызывает определение величины g_i . Для этого множество структур, записанных в массивах M_f , необходимо разбить на группы тождественных структур и для каждой группы определить максимальное число непересекающихся структур. Таким образом, задача определения величины g_i разбивается на две — разбиение структур на группы и определение непересекающегося множества структур внутри каждой группы.

Задача разбиения структур на группы рассматривается внутри каждого массива M_f отдельно, так как структуры с различным числом вершин (различный уровень структур) не могут быть тождественны. Количество структур даже в одном массиве M_f может измеряться несколькими тысячами, поэтому алгоритм разбиения структур должен быть достаточно эффективным. Так, если число структур в массиве $n = 1000$, то максимальное число сравнений структур для определения их тождественности составит $n(n-1)/2$, или около 500000. Однако это справедливо для случая, когда все структуры будут различного вида. Если же, например, множество из $n = 1000$ структур разобьется на $m = 10$ равных групп, то число сравнений будет значительно меньше и составит $n(m+1)/2 = 1000(10+1)/2 = 5500$.

Для сокращения числа сравнений структур и повышения эффективности алгоритма разбиения структур на группы представляется целесообразным построить его по следующей схеме. Выделить одно или несколько простых условий, являющихся необходимыми для тождественности структур, и разбить исследуемую группу структур по данным условиям на подгруппы. Тождественность какой-либо пары структур, принадлежащих разным подгруппам, исключается за счет несовпадения условий. Поэтому последующие сравнения структур и выявление тождественных групп производится внутри каждой подгруппы, что позволяет существенно сократить общее число сравнений.

Вторым важным моментом, характерным для рассматриваемой задачи, является то, что в данном случае при сравнении пары структур нет необходимости находить взаимно-однозначное соответствие их вершин. Достаточно лишь убедиться в том, что эти структуры тождественны или нет. В этой связи очень важным представляется вопрос о формировании такого множества некоторых простых условий, выполнение которых было бы не только необходимым, но и достаточным для тождественности сравниваемых структур. Разумеется, что эффективность алгоритма установления тождественности в данном случае зависит от того, насколько быстрее можно осуществить проверку таких условий по сравнению с установлением взаимно-однозначного соответствия вершин.

При сравнении двух структур, описываемых графами $G = (P, U)$ и $G' = (P', U')$, в качестве необходимого условия тождественности может быть принято следующее: для каждой вершины $p_i \in P$ графа G существует такая вершина $p'_i \in P'$ графа G' , что $\alpha_i = \alpha'_i$, $\beta_i = \beta'_i$ множество функций дуг $\{\mu_{ij}\}$ соответствующего отображения F_{p_i} совпадает с множеством $\{\mu'_{ij}\}$, соответствующим отображению $F'_{p'_i}$, а $\{\mu_{ij}\}^{-1}$ отображения $F_{p_i}^{-1}$ совпадает с $\{\mu'_{ij}\}^{-1}$ отображения $F'_{p'_i}$.

Проверка данного условия осуществляется по спискам вершин сравниваемых структур, представленным в форме табл. 1. Процесс сравнения структур состоит в выборе очередной вершины из списка графа G и поиске вершины в графе G' , удовлетворяющей указанному условию. Найденная таким образом вершина в графе G' отмечается и в по-

следующем при проверке условия для следующей вершины графа G не используется. Если для некоторой очередной вершины p_i графа G сформулированное выше условие не удовлетворяется, то сравниваемые структуры не тождественны. Структуры, для вершин которых указанное условие соблюдается, записываются в одну подгруппу. При этом вершины графов G' переписываются в очередности, с которой они выбирались при проверке условия, в соответствии с очередностью записи вершин в списке графа G . Такая запись позволяет в последующем при установлении тождественности графов внутри подгруппы начинать проверку тождественности с одного из возможных вариантов (взаимно-однозначного) соответствия вершин.

Что касается вопроса о формировании совокупности условий, необходимых и достаточных для установления тождественности структур, то этот вопрос требует проведения специального исследования и в настоящей работе не рассматривается.

Сформированные по введенному необходимому условию тождественности группы структур разбиваются на подгруппы тождественных структур. Процесс разбиения осуществляется в каждой из групп независимо от других и состоит в последовательной проверке пар структур на тождественность. Такая проверка осуществляется в соответствии с алгоритмом, изложенным в [2]. Алгоритм в данном случае работает достаточно эффективно, поскольку велика вероятность того, что структуры, принадлежащие одной подгруппе, являются тождественными и, следовательно, количество обратных ходов будет невелико.

После выполнения этого этапа над всеми массивами M_i структуры всех уровней будут разбиты на группы тождественных структур. В каждой группе необходимо найти максимальное число непересекающихся структур g_i , которое характеризует повторяемость структур i -й группы. Так как данная задача ниже рассматривается применительно к одной группе, то индекс i в последующем изложении опускается. Задача определения непересекающегося множества структур рассматривалась в [2]. При этом предполагалось, что структуры имеют различную мощность. Поэтому в оптимизационной постановке задача сводилась к взвешенной задаче покрытия. В рассматриваемом случае все структуры принадлежат одному уровню, то есть имеют одинаковую мощность. Это позволяет рассматривать задачу определения $\max g$ как задачу нахождения числа внутренней устойчивости графа. Действительно, построим граф $G = (S, U)$, вершинами s_i которого являются структуры, а ребра $s_i - s_j$ указывают на наличие общих элементов у i -й и j -й структур. Иными словами, на множестве структур группы построим граф пересечений. Определение для данного графа числа внутренней устойчивости соответствует нахождению максимального числа непересекающихся (несвязанных) вершин.

Среди известных алгоритмов определения числа внутренней устойчивости можно выделить алгоритм Магу — Уэйсмана [4]. Существенным для данного алгоритма является то, что он находит все максимально пустые подграфы. Для графа, содержащего 200—300 вершин, число максимально пустых подграфов становится необозримо большим и использование алгоритма приводит к значительным трудностям. В рассматриваемой задаче важно находить само число внутренней устойчивости, а не все максимально пустые подграфы. Здесь необходимо заметить, что количество групп тождественных структур будет велико, а соответствующие им графы пересечений могут содержать несколько сот вершин. Поэтому рассчитывать на использование пусть даже сравнительно эффективного алгоритма решения данной задачи не приходится. Единственно возможным в данных условиях является разработка

эффективного алгоритма приближенного решения задачи определения числа внутренней устойчивости.

Рассмотрим граф пересечений $G = (S, U)$, построенный на множестве структур, принадлежащих одной группе. Если граф не связан, то каждую его компоненту рассматриваем отдельно. Алгоритм приближенного вычисления числа g основан на предварительном ранжировании графа G . Ранжирование может быть начато с любой вершины. На рис. 3, а показан граф G , а на рис. 3, б — результат его ранжирования.

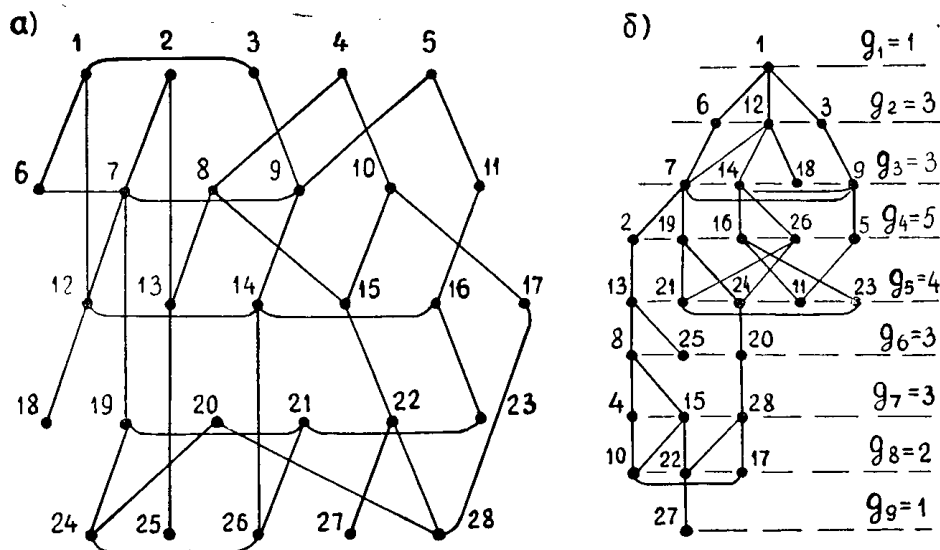


Рис. 3

Процесс ранжирования осуществляется следующим образом. Исходной вершине s_i , с которой начинается ранжирование, присваивается 1-й ранг. Ко второму рангу относится множество вершин отображения F_{s_i} . Третий ранг формируется из множества вершин, содержащихся в отображениях вершин множества F_{s_i} (вершин второго ранга). При этом не включаются вершины, уже вошедшие в множество вершин второго ранга. В ранжированном графе все связи исходного графа сохраняются. После ранжирования множество вершин каждого ранга анализируется с целью определения максимального числа непересекающихся структур. Если для некоторого t -го ранга множество вершин не связано (например, для $t = 4$, рис. 3, б), максимальное число непересекающихся структур для такого ранга g_t определяется числом вершин, содержащихся в данном множестве. Так, для $t = 4$, $g_4 = 5$. В случае, если все множество вершин (либо его часть) некоторого t -го ранга связано, то число g_t ищется так же, как и для исходного графа, то есть осуществляется ранжирование и определение величин g'_t , по которым определяются величины g_t . При определении величин g'_t также может встретиться ситуация, когда на соответствующих рангах множество вершин связано. В этом случае определяются величины g''_t , по которым определяются величины g'_t . Таким образом, процесс ранжирования при определении величины g осуществляется до тех пор, пока ни одно из множеств всех рангов не будет содержать связанную совокупность вершин.

После вычисления всех величин g_t получается последовательность чисел $g_1, g_2, \dots, g_t, \dots, g_{u-1}, g_u$, где u — число рангов, на кото-

рое распределяется граф при ранжировании. Для вычисления величины g необходимо решить задачу выбора такой совокупности A чисел g_t , которые не были бы соседними в последовательности $t = 1, 2, \dots, u$, а $\sum_{t \in A} g_t \rightarrow \max$.

Эффективным методом решения данной задачи является схема, в соответствии с которой строится граф и определяется маршрут максимальной длины.

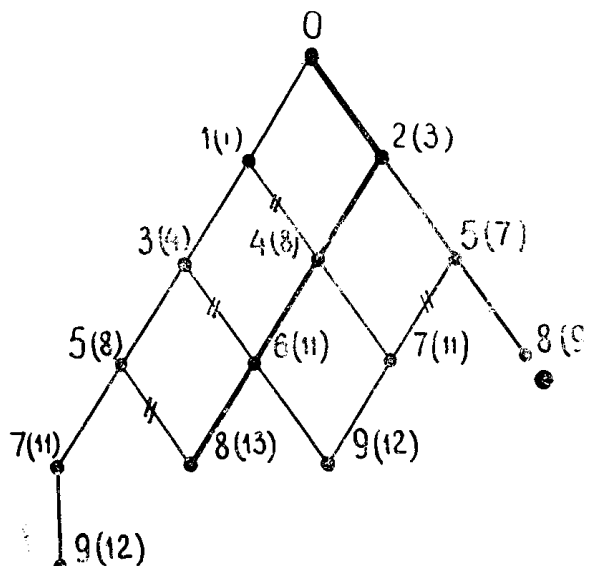


Рис. 4

Построение графа возможных маршрутов обхода последовательности чисел g_t осуществляется по правилу, которое вытекает из конкретных условий задачи. Суть этого правила состоит в том, что если организовать обход рангов с повышением их номеров, из каждого t -го ранга имеет смысл двигаться лишь в ранги с номерами $t+2$ или $t+3$. Движение в ранг $t+4$ из t -го ранга недопустимо, так как в ранг с номером $t+4$ имеется возможность пройти через ранг $t+2$ и суммарное число величин g_t увеличивается на g_{t+2} . Граф, построенный с учетом сказанного выше для девяти рангов (рис. 3, б), показан на рис. 4.

Построение графа и определение максимального маршрута осуществляются одновременно.

Рассмотрим процесс построения графа и определения максимального маршрута на примере рис. 4. Начинается построение с рангов (вершин) 1 и 2. Из вершины 1 можно двигаться в вершины 3 или 4. Рядом с номерами вершин на рисунке в скобках указаны суммы соответствующих значений чисел g_t . Из вершины 2 можно двигаться в вершины 4 и 5. Таким образом, в вершину 4 можно попасть через вершину 1 и 2. При движении через вершину 1 сумма чисел $g_1 + g_4 = 1 + 5 = 6$, при движении через вершину 2 сумма чисел $g_2 + g_4 = 3 + 5 = 8$. Поэтому маршруты с началом 1—4 могут в последующем не рассматриваться и у вершины 4 в скобках проставляется число 8. В ходе построения может оказаться, что различные маршруты приводят в вершины с одинаковым номером. Например, маршрут 1—3—5 и 2—5. В этом случае сравниваются значения величин в скобках у этих вершин и маршруты через вершину с меньшим значением величины в скобках исключаются из рассмотрения. В случае, если величины в скобках оказываются одинаковыми, то исключается любое из множеств маршрутов, проходящих через данные вершины.

Маршрут, заканчивающийся вершиной с максимальной величиной в скобках, является решением задачи. Величина в скобках соответствует искомой величине g .

Ранжированный граф, который не имеет связей между вершинами (назовем его сквозным), обладает хорошим свойством. Независимо от того, с какой вершины начато ранжирование, граф получается сквозным. При этом для каждого, полученного таким образом,

сквозного графа величина g , найденная по приведенному выше алгоритму, остается постоянной.

В общем случае для несквозного ранжированного графа при определении числа g_i на множестве связанных вершин может оказаться не безразличным, какие именно вершины вошли в совокупность, соответствующую числу g_i . Так, для примера, представленного на рис. 5, число $g_2 = 2$ определяется одной из совокупностей вершин (2,4) (2,5), (3,5), число $g_4 = 2$ совокупностями (9,11) или (10, 11). Однако, если при определении g_2 будет выбрана совокупность (2,4) или (2,5), а при определении g_4 будет выбрана совокупность (10,11), то дополнительно может быть включена вершина 6, что позволит величину g , подсчитанную по приведенному алгоритму, увеличить на единицу.

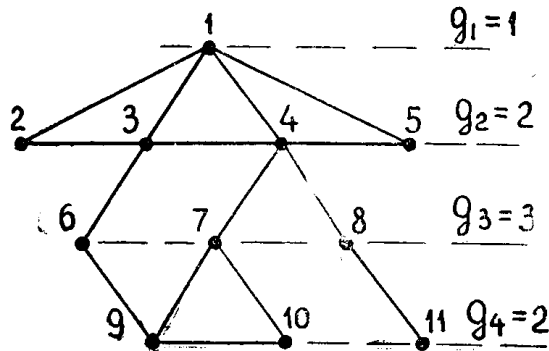


Рис. 5

Учет этих моментов приводит к существенному усложнению приведенного выше алгоритма. Усложнение происходит за счет увеличения числа вариантов множеств вершин, соответствующих числам g_i и, следовательно, увеличению числа вершин в графе, который строится при определении максимального маршрута. В случае, если эта дополнительная возможность увеличения числа g_i не используется, то независимо от этого после определения величины g осуществляется проверка множеств рангов, которые не вошли в решение на наличие возможности включения их в решение.

После определения величин g для всех групп тождественных структур осуществляется нормирование значений g и m так чтобы эти значения имели математическое ожидание, равное нулю, и дисперсию, равную единице. Оценка каждой структуры определяется по выражению (2). Заметим, что все структуры одной группы имеют одинаковые оценки, так как мощность этих структур и значение повторяемости g для них одинаково. Значения весовых коэффициентов подбираются с учетом конкретных условий, а также на основе анализа результатов экспериментов.

Полученная таким образом оценка для конкретной типовой структуры зависит от количественного соотношения частот повторения видов структур в анализируемой функциональной схеме. При анализе некоторой совокупности функциональных схем, при оценке структуры необходимо учитывать частоты повторения структур данного вида в других схемах. Действительно, структуры некоторого i -го вида встречаются, например, в каждой из пяти анализируемых схем по два раза. Таким образом, общая частота равна десяти. Структура другого j -го вида встречается всего лишь в одной схеме пять раз. Понятно, что структура i -го вида при анализе каждой схемы будет иметь достаточно низкую оценку и может оказаться исключенной из последующего рассмотрения, в то время как структура j -го вида может иметь оценку, достаточную для того, чтобы быть неисклученной. Очевидно, что такое решение неправильно, так как выгоднее оставить структуру i -го вида. В связи с этим при определении оценки структуры необходимо учитывать ее повторяемость во всех анализируемых схемах.

Обозначим через g_{ih} , $h = 1, 2, \dots, H$ повторяемость структур i -го вида в h -й схеме. Тогда общая повторяемость структур i -го вида g_i , которая используется в выражении (2), определится следующим образом:

$$g_i = \sum_{h=1}^H g_{ih}. \quad (6)$$

После определения оценок C_i для всех групп производится анализ полученных результатов и принимается решение об исключении групп с низкими оценками C_i из последующего рассмотрения. При этом важным моментом является установление минимально допустимой оценки, обозначим ее через C_0 . Группы структур, для которых не выполняется условие

$$C_i \geq C_0, \quad (7)$$

исключаются из последующего рассмотрения. Величина C_0 подбирается в зависимости от конкретных условий: количества видов групп, возможности последующей обработки оставшегося множества групп, количества исключаемых при этом вершин исходных графов и ряда других условий.

Выбор оптимальной совокупности структур

После выполнения изложенных выше этапов структурного анализа имеется n групп структур j -го вида. Каждая группа характеризуется оценкой C_j , общим количеством структур n_j и количеством непересекающихся структур g_j . Известно общее количество структур j -й группы, принадлежащих h -й схеме n_{jh} , и совокупность непересекающихся структур g_{jh} . Необходимо среди всего множества структур выбрать такую совокупность, которая бы не пересекаясь покрывала элементы всех анализируемых схем и в то же время имела минимальное число видов структур. От качества решения данной задачи во многом зависит последующий успех при решении всей задачи унификации.

Характерным для рассматриваемой задачи является то, что она должна решаться совместно для всех схем. Действительно, если решать задачу отдельно для каждой схемы, то полученные при этом совокупности структур могут не совпадать по виду, что приводит к увеличению общего числа видов структур. При одновременном решении задачи покрытия для всех схем имеется возможность учитывать частоту использования структур одного вида во всех схемах. Ранее уже отмечалось, что при выборе номенклатуры структур важно, чтобы они по возможности были крупнее, то есть должна учитываться мощность структур. Таким образом, решение задачи выбора совокупности структур должно отвечать двум основным требованиям — структуры должны быть достаточно мощными и иметь высокую частоту использования. Рассмотрим одну из возможных постановок решения данной задачи.

Введем переменную x_{jk}^h ,

$$x_{jk}^h = \begin{cases} 1, & \text{если } \kappa\text{-я структура в } j\text{-й группе, принадлежащая } h\text{-й схеме,} \\ & \text{включена в решение; } j = 1, 2, \dots, n; \kappa = 1, 2, \dots, n_j; h = \\ & = 1, 2, \dots, H; \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда частота использования структур j -го вида определится как сумма значений x_{jk}^h , $\kappa = 1, 2, \dots, n_j$. Целевая функция с учетом указанных выше требований может быть записана следующим образом:

$$\sum_{j=1}^n m_j^2 \left(\sum_{\kappa=1}^{n_j} x_{jk}^h \right)^2 = Z (\max), \quad (8)$$

где m_j — мощность структур j -го вида. Величина m_j в целевой функции (8) возведена в квадрат, с тем, чтобы отдать предпочтение более мощным структурам. Величина

$$m_j^2 \sum_{\kappa=1}^{n_j} x_{j\kappa}^h$$

по существу является коэффициентом при переменных в целевой функции, так как характеризует частоту использования структур j -го вида и их мощность. Этот коэффициент характеризует участие структур j -й группы в покрытии схемы, поэтому сумма переменных по j -й группе в выражении (8) получается в квадрате. Ограничения для данной задачи формируются из условия того, что ни один из элементов анализируемых схем не может одновременно входить в две или более включенных в решение структур. Это условие можно записать в виде

$$\sum_{j=1}^n \sum_{\kappa=1}^{n_j} a_{lhjk} x_{j\kappa}^h \leq 1; \quad h = 1, 2, \dots, H; \quad l_h = 1, 2, \dots, S_h, \quad (9)$$

где S_h — количество элементов, содержащееся в h -й схеме;

$$a_{lhjk} = \begin{cases} 1, & \text{если } \kappa\text{-я структура } j\text{-й группы содержит } l\text{-й элемент} \\ & h\text{-й схемы,} \\ 0 & \text{в противном случае.} \end{cases}$$

Задача (8), (9) относится к классу задач покрытия с нелинейной целевой функцией. Современная теория математического программирования не располагает достаточно эффективными методами решения таких задач. К тому же размерность рассматриваемой задачи может достигать нескольких тысяч, а количество ограничений, определяемое суммарным числом элементов, во всех анализируемых схемах также может превысить тысячу.

Исходя из этого, представляется целесообразным более тщательно проанализировать группы тождественных структур и выбрать из них наиболее перспективные с точки зрения использования их при покрытии схем. Выбор перспективных групп в данном случае означает выбор номенклатуры типовых структур. Поэтому при таком подходе данный этап является одним из наиболее ответственных при структурном анализе.

Важным достоинством предлагаемого подхода является то, что задача выбора групп может рассматриваться для каждой схемы независимо. Это оказывается возможным, если данную задачу рассматривать как задачу покрытия элементов схем группами структур по критерию максимума их суммарных оценок.

Введем переменную x_j , которая равна 1, если j -я группа структур включена в решение, и 0 в противном случае. Тогда задача выбора рациональной совокупности групп для h -й схемы будет заключаться в максимизации функции

$$\max L = \sum_{j=1}^n C_j x_j, \quad (10)$$

при условиях

$$\sum_{j=1}^n a_{lj} x_j \leq 1, \quad l = 1, 2, \dots, S, \quad (11)$$

$$x_j = 0; 1, \quad j = 1, 2, \dots, n, \quad (12)$$

где

$$a_{lj} = \begin{cases} 1, & \text{если } j\text{-я группа структур содержит } l\text{-й элемент схемы,} \\ 0 & \text{в противном случае,} \end{cases}$$

C_j — оценка структур j -й группы.

Выше отмечалось, что задача выбора структур решается для каждой схемы независимо. Поэтому в (11) значения a_{lj} формируются исходя из структур j -й группы, принадлежащих рассматриваемой схеме. При этом учитываются лишь те структуры, которые вошли в непересекающуюся совокупность при определении повторяемости. Остальные структуры не могут использоваться, так как при решении задачи (10) — (12) они наравне с другими структурами будут использоваться для покрытия элементов схемы, а в последующем будут исключены при выборе непересекающейся совокупности структур.

Ограничение (11) исключает возможность пересечения между двумя группами даже по одному из элементов схемы и является, таким образом, слишком жестким. Это может привести к тому, что в конкретной ситуации какие-либо две группы, имеющие, например, самые высокие оценки C_j из-за наличия пересечения по незначительному числу элементов не смогут быть включены в решение. Поэтому вместо задачи (10) — (12) рассматривается задача минимизации функции

$$\min L = \sum_{j=1}^n \frac{1}{C_j} x_j, \quad (13)$$

при условиях
$$\sum_{j=1}^n a_{lj} x_j \geq 1, \quad (14)$$

$$x_j = 0; 1, j = 1, 2, \dots, n. \quad (15)$$

При решении данной задачи некоторая совокупность структур может пересекаться. Поэтому решение задачи необходимо корректировать путем выбора непересекающихся структур, покрывающих максимальное число элементов схемы. В результате такой корректировки часть элементов схемы может оказаться непокрытой структурами. Если это обстоятельство нежелательно или недопустимо, то решение задачи (13) — (15) может быть улучшено решением другой задачи покрытия, в которой используются все структуры выбранных групп. При этом предпочтение следует отдавать более крупным структурам. Фактор повторяемости в данном случае не играет роли, так как номенклатура структур уже определена. В этой задаче переменная x_j принимает значение 1, если j -я структура включена в решение, и 0 в противном случае. Задача состоит в минимизации функции

$$\min L = \sum_{j=1}^n \frac{1}{m_j} x_j, \quad (16)$$

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, 2, \dots, S, \quad (17)$$

$$x_j = 0; 1, j = 1, 2, \dots, n, \quad (18)$$

где m_j — мощность j -й структуры;

n — общее количество структур, сформированных для рассматриваемой схемы;

S — общее количество элементов схемы;

$$a_{ij} = \begin{cases} 1, & \text{если } i\text{-й элемент содержится в } j\text{-й структуре,} \\ 0 & \text{в противном случае.} \end{cases}$$

В случае, если нет необходимости отдавать предпочтение мощности структур, целевая функция (16) может быть заменена на более простую функцию

$$\min L = \sum_{j=1}^n x_j. \quad (19)$$

Задачу (19), (17), (18) можно решать путем нахождения минимального покрытия. При этом может оказаться, что некоторые группы структур не будут включены в решение. Если это повторится для всех анализируемых схем, то такая группа автоматически исключится из выбранной ранее номенклатуры. Возможные пересечения структур могут быть устранены дублированием элементов пересечения в схеме. На этом структурный анализ схем заканчивается. Совокупность структур, полученная при покрытии схем, распределяется по группам тождественности. Результаты анализа можно представить в форме матрицы $A = \|a_{s\omega}\|$, $s = 1, 2, \dots, S$, $\omega = 1, 2, \dots, W$. Элемент матрицы $a_{s\omega}$ характеризует количество структур ω -го вида, используемых в s -й схеме.

Как видно из предшествующего изложения, структурный анализ связан с решением большого числа оптимизационных задач большой размерности и ряда задач комбинаторного характера также большой размерности. Анализируя в целом предложенную методику проведения структурного анализа можно сделать вывод, что в работе предпринята попытка решения задач структурного анализа с помощью строгих методов. При этом большое внимание уделялось вопросам реализуемости формулируемых оптимизационных постановок. Сложные задачи, как правило, расчленялись на более простые либо по возможности сводились к таким задачам, для которых имеются или предлагаются приемлемые методы решения.

ЛИТЕРАТУРА

1. В. К. Погребной, В. З. Ямпольский. Проектирование унифицированного набора блоков для заданного класса функциональных схем вычислительных устройств. Известия ТПИ, т. 269, Томск, 1972.
2. Автоматизация проектирования вычислительных устройств на элементах интегральной микроэлектроники. Отчет по НИР. Томск, ТПИ, 1971.
3. В. К. Погребной. Разработка и исследование алгоритмов автоматизации технического этапа проектирования вычислительных устройств. Диссертация на соискание ученой степени кандидата технических наук. Томск, ТПИ. 1970.
4. А. А. Зыков. Теория конечных графов. Новосибирск, «Наука», 1969.