

## ПЕРЕВОД ФУНКЦИОНАЛЬНОЙ СХЕМЫ НА РЕАЛЬНЫЙ КОНСТРУКТИВНЫЙ БАЗИС

Е. Ф. ЖИГАЛОВА

(Представлена научно-техническим семинаром кафедры автоматизированных систем управления и лабораторией управления)

Современные темпы развития микроэлектроники и совершенствование технологии ее производства в значительной мере определяют темпы проектирования вычислительных устройств. Эффективным средством достижения высокого качества и существенного сокращения сроков проектирования являются ЭВМ. За последнее десятилетие в СССР появилось много теоретических и практических работ, посвященных проектированию вычислительных устройств с применением ЭВМ. При этом наибольшие практические результаты получены при проектировании логического и технического этапов. Наряду с этим получены положительные результаты полной автоматизации проектирования. Здесь можно сослаться на работы института кибернетики АН УССР, а также фирмы IЭВМ США. Одним из преимуществ полной автоматизации проектирования является непрерывность проектирования, что способствует сокращению времени на подготовку исходной информации, ее ввод и вывод из ЭВМ. С этой точки зрения, а также с учетом полученных практических результатов автоматизации логического и технического этапов проектирования представляет интерес задача перевода функциональной схемы на реальный конструктивный базис с применением ЭВМ, в которой можно выделить ряд вопросов, в том числе:

- а) покрытие функциональной схемы конструктивными элементами;
- б) согласование элементов принципиальной схемы по времени;
- в) согласование элементов принципиальной схемы по нагрузке.

Перевод функциональной схемы на реальный конструктивный базис представляет практический интерес и в том смысле, что смена конструктивной базы элементов не обязательно влечет за собой смену алгоритма функционирования того или иного вычислительного устройства. Это позволяет спроектированную функциональную схему реализовать несколькими конструктивными вариантами.

Для однозначного понимания дальнейшего изложения будем пользоваться основными определениями, данными в [2]. В зависимости от принятой конструктивной базы элементов возможны следующие случаи реализации функциональной схемы:

- 1) каждый логический элемент может быть реализован одной конструктивной единицей;

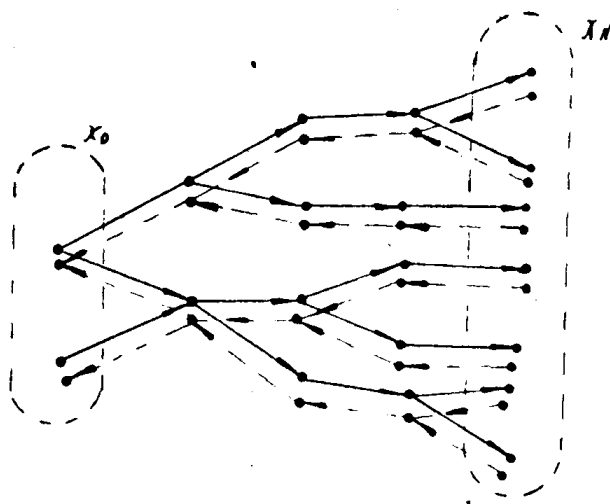


Рис. 1

2) возможна реализация некоторых функциональных узлов одной конструктивной единицей (таких, как триггер);

3) каждый логический элемент может быть реализован функционально связанной группой конструктивных элементов.

В соответствии с этим покрытие функциональной схемы модулями принятого конструктивного базиса сводится к приписыванию каждому логическому элементу либо функциональному узлу некоторого количества соответствующих конструктивных единиц с последующим запоминанием их функциональных связей, что с теоретической точки зрения не представляет каких-либо трудностей и легко реализуется на ЭВМ.

Способ реализации функциональной схемы, а также физическая природа конструктивных элементов могут внести определенные неувязки времени функционирования отдельных узлов и элементов соответствующей принципиальной схемы. Для правильного функционирования устройства необходимо построить реальную временную диаграмму, учитывающую эти особенности. На основе реальной временной диаграммы производится выравнивание запаздывания в информационных цепях. Для контроля правильности прохождения сигналов по устройству может быть использован метод анализа тестов, описанный Е. И. Гурвичем, Т. П. Куликовской [2].

Логические элементы функциональной схемы не имеют ограничения на нагрузочную способность, поэтому при переходе к реальному конструктивному базису возможны рассогласования элементов по нагрузке, т. е. некоторые конструктивные элементы могут оказаться перегруженными. Как правило, серии конструктивных элементов содержат элементы нескольких групп, отличающихся друг от друга нагрузочной способностью. На таблице приведена таблица нагрузочных способностей различных групп элементов некоторой серии конструктивных элементов, где  $A, B, \dots, F$  — группы элементов,  $a_{11}; a_{12}; a_{21}; a_{mn}$  — предельные нагрузочные способности одного элемента групп  $A, B, \dots, F$ .

Рассмотрим задачу согласования элементов принципиальной схемы по нагрузке для серии конструктивных элементов, содержащей две нагрузочные группы. Примем следующие исходные посыпки:

1. Функциональная схема представлена графом Бержа, вершины которого соответствуют логическим элементам и узлам, а дуги — функциональным связям между ними.

2. Нагрузочная способность конструктивных элементов, а также и сама нагрузка выражаются в количествах элементов.

3. Каждому логическому элементу функциональной схемы поставлен в соответствие конструктивный элемент (либо группа элементов), реализующий данный логический оператор.

4. Данная конструктивная серия содержит элементы двух групп  $A$  и  $B$ . Элементы группы  $B$  более мощные, чем элементы группы  $A$ . Тогда справедливы соотношения:

$$\begin{aligned} a_{11} &> a_{12}. \\ a_{21} &> a_{22}. \end{aligned} \quad (1)$$

Т а б л и ц а

Тип нагружаемого элемента \ Тип элементов нагрузки	$A$	$AU \ BU \dots UF$	$B$	$\dots$	$F$
$A$	$a_{11}$	$a_{11} + a_{12} + a_{1n}$	$a_{12}$	$\dots$	$a_{1n}$
$B$	$a_{21}$	$a_{21} + a_{22} + a_{2n}$	$a_{22}$	$\dots$	$a_{2n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$F$	$a_{m1}$	$a_{m1} + a_{m2} + a_{mn}$	$a_{m2}$	$\dots$	$a_{mn}$

Из приведенных соотношений видно, что количество конструктивных элементов в устройстве зависит от выбора группы элементов при покрытии функциональной схемы, так как в зависимости от количества элементов, а также их группы в нагрузке нагружаемый логический элемент или узел могут быть реализованы соответствующими элементами, принадлежащими одной группе либо двум группам. В основе алгоритма согласования лежит простое соотношение

$$k \cdot x_i^{A, B} \geq p_j, \quad (2)$$

где

$x_i^{A, B}$  — количество элементов какой-либо группы в  $y_i$ -й вершине;  
 $k$  — соответствующий нагрузочный коэффициент;  
 $p_j$  — нагрузка, или потребляемая мощность в  $y_j$ -й вершине, где

$$y_j \in \Gamma y_i.$$

Назовем оптимальным такое решение, при котором в каждой вершине элементы той или иной группы, с одной стороны, уравнивают соответствующую нагрузку, т. е. выполняется соотношение (2), а с другой стороны, сами представляют минимальную нагрузку, иначе говоря, потребляют меньшую мощность.

Введем в рассмотрение некоторую функцию потребляемой мощности  $\varphi(x_A; x_B)$ , определенную на множестве всех вершин графа  $G = (y, \vec{u})$ . Очевидно, что

$$\varphi(x_A, x_B) = \sum_{i=1}^N (\gamma_A x_i^A + \gamma_B x_i^B), \quad (3)$$

где

$i = \overline{1, N}$  — множество всех вершин,

$\gamma_A; \gamma_B$  — коэффициенты потребляемой мощности.

Тогда задача согласования по нагрузке является задачей минимизации линейной функции  $\varphi(x_A, x_B)$  при выполнении условия (2).

Целевая функция, минимизирующая значение потребляемой мощности конструктивными элементами, является достаточно рациональным критерием оптимальности, в значительной мере определяющим количество элементов в реальной схеме. Рассмотрим процесс вычисления данной задачи. Так как потребляемая мощность определяется количеством и группой элементов нагрузки, то соотношение (2) можно записать в виде

$$k_j x_i^{A,B} \geq \sum_{y_j \in \Gamma y_i} (x_j^A + x_j^B), \quad (4)$$

где  $x_j^A, x_j^B$  — количество элементов групп  $A$  и  $B$  в нагрузке.

Соотношение (2) показывает, что количество элементов в  $y_i$ -й вершине определяется величиной потребляемой мощности в вершинах  $y_j \in \Gamma y_i$ , или количеством элементов соответствующей группы, и совершенно не зависит от решения, принятого в вершине  $y_r \in \Gamma^{-1} y_i$ . Это дает возможность принимать решение в каждой вершине отдельно. Тогда процесс решения задачи будет заключаться в следующем. Задавшись нагрузкой или потребляемой мощностью в вершинах  $\{y_i\} \in Y_N$  таких, что  $\Gamma Y_N = \emptyset$ , получим, что соотношение (4) для вершин  $Y_{N-1} \in \Gamma_{Y_N}^{-1}$  справа будет иметь вполне конкретное численное значение. Решив совместно уравнения (3) и (4), получим решение в вершинах  $Y_{N-1}$ , которое определяет нагрузку в вершинах  $Y_{N-2} \in \Gamma_{Y_{N-1}}^{-1}$  или  $Y_{N-2} \in \Gamma_{Y_N}^{-2}$ . Эти рассуждения можно продолжить и для вершин  $Y_0 \in \Gamma_{Y_N}^{-n}$ . Иначе говоря, процесс принятия решения в каждой вершине графа  $G = (Y, \vec{U})$  представляет последовательный спуск по дереву, имеющего ориентацию, обратную ориентации данного графа. На рис. 2 приведен граф  $G = (Y, \vec{U})$ , а пунктиром показано дерево спуска.

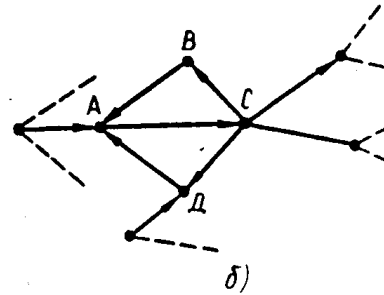
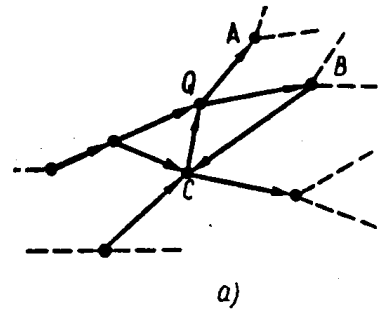


Рис. 2

### Алгоритм согласования по нагрузке

Левая часть соотношения (4) содержит коэффициент нагрузочной способности  $k$ , величина которого зависит как от группы элементов нагрузки, так и от группы нагружаемых элементов (табл. 1). Представим переменную  $x_i^{A,B}$  как сумму переменных  $Z_1^{A,B}$  и  $Z_2^{A,B}$ , компенсирующих соответственно потребляемую мощность элементами групп  $A$  и  $B$  в нагрузке. Тогда соотношение (4) энергетического баланса в вершине  $y_i = \Gamma^{-1} y_j$  можно представить в виде системы соотношений (4\*):

$$\begin{cases} a_{11} Z_1^A + a_{21} Z_1^B \geq x_j^A \\ a_{12} Z_2^A + a_{22} Z_2^B \geq x_j^B, \end{cases} \quad (4^*)$$

где

$$Z_1^A + Z_2^A + Z_1^B + Z_2^B = x_i^{A, B}.$$

Таким образом, мы имеем задачу распределения с двумя типами ресурсов. Определение оптимальной политики при многомерных процессах распределения является довольно сложной задачей. Поэтому используя специфику данной задачи, сведем ее к задаче с одним типом ресурсов. Запишем уравнение энергетического баланса для одного элемента групп  $A$  и  $B$  в соответствии с приведенным соотношением (1).

$$\left\{ \begin{array}{l} a_{11} P_A = a_{12} P_B, \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} a_{21} P_A = a_{22} P_B, \end{array} \right. \quad (6)$$

где  $P_A$  — мощность, потребляемая одним элементом группы  $A$ ,

$P_B$  — мощность, потребляемая одним элементом группы  $B$ .

Умножив обе части уравнения (5) на величину  $\frac{1}{a_{12} P_A}$ , а уравнение (6) —  $\frac{1}{a_{22} P_A}$ , получим соотношения:

$$\left\{ \begin{array}{l} \frac{a_{11}}{a_{12}} = \frac{P_{13}}{P_A}, \end{array} \right. \quad (7)$$

$$\left\{ \begin{array}{l} \frac{a_{21}}{a_{22}} = \frac{P_{13}}{P_A}, \end{array} \right. \quad (8)$$

откуда следует, что

$$\frac{P_B}{P_A} = \frac{a_{11}}{a_{12}} = \frac{a_{21}}{a_{22}} = j, \text{ т. е.}$$

$$P_B = j P_A. \quad (9)$$

Так как мощность здесь измеряется в условных единицах, т. е. в количествах элементов, перепишем соотношение (7) в соответствующих обозначениях:

$$x_B = j x_A. \quad (10)$$

Соотношение (8) позволяет потребляемую мощность каждого элемента группы  $B$  в нагрузке выразить через потребляемую мощность одного элемента группы  $A$ . Теперь данную задачу можно записать: минимизировать линейную форму

$$\varphi(x^A, x^B) = x_i^A + j x_i^B \quad (11)$$

при условии

$$\left. \begin{array}{l} a_{11} x_i^A + a_{21} x_i^B \geq P_j \\ x_i^{A, B} = 0, 1, 2, \dots \end{array} \right\} \quad (12)$$

Если бы условие (12) сводилось просто к ограничению  $x_i^{A, B} \geq 0$ , задача была бы совсем несложной и ее решение можно было бы определить следующим образом. Рассмотрим отношения  $\frac{P_j}{a_{11}}$  и  $\frac{P_j}{a_{21}}$ , и пусть  $b$  индекс, для которого это отношение принимает минимальное значение. Затем выберем только элементы типа  $b$ , при этом получим потребляемую мощность элементов

$$\varphi_b = \frac{P_j}{a_b} \cdot \gamma_b.$$

Очевидно, что это решение не всегда будет удовлетворять условию (12). Округление решения задачи до ближайшего целого числа может привести к решению, далекому от оптимального. Рассмотрим задачу, включающую две группы элементов:

$$\begin{aligned} P_j &= 18 \text{ — нагрузка} \\ a_{11} &= 4 \quad \gamma_1 = 1 \\ a_{21} &= 5 \quad \gamma_2 = 1,25. \end{aligned}$$

Если бы не условие целочисленности, мы для компенсации этой мощности выбрали бы  $18:5=3,6$  элемента с потребляемой мощностью  $18:5 \cdot 1,25=4,5$ . Чтобы выполнялось условие (12), можно 3,6 округлить до 3 и взять 1 элемент первой группы, чтобы скомпенсировать избыточную мощность. Это решение дает потребляемую мощность 4,75. Однако настоящий оптимум легко определяется простым перебором, он равен 4,5 и достигается посредством выбора двух элементов первой группы и двух элементов второй группы. Для решения этой задачи может быть использован алгоритм, основанный на идее динамического программирования.

Запишем рекуррентное соотношение для вычисления функции  $f_N(P_j) = \min L_N(x)$  в каждой вершине графа

$$f_N(P_j) = \min_{\{x_N\}} \{x_N \gamma_N + f_{N-1}(P_j - x_N \cdot a_N)\}, \quad (13)$$

где  $x_N$  — пробегает значение  $0, 1, 2, \dots, \left[ \frac{P}{a_N} \right]$ .

Предложенная Р. Беллманом [3] вычислительная процедура, дающая оптимальное целочисленное решение, может быть применена и для вычисления формулы (13).

Рассматривая задачу согласования элементов принципиальной схемы по нагрузке, мы исходили из ряда допущений, в том числе полагая, что логическая схема не содержит контуры. Сделаем обратное допущение: логическая схема содержит контуры и разберем, как будет теперь решаться эта задача. Логическая схема может содержать простые контуры, т. е. каждая дуга, входящая в контур, принадлежит только одному контуру, а также и сложные, т. е. в контуре можно выделить хотя бы одну дугу, принадлежащую двум контурам. В начале рассмотрим первый, более простой случай. На рис. 2, а показана часть графа логической схемы, содержащая контур Q. Следует сразу оговориться, что алгоритм обнаружения циклов здесь рассматриваться не будет. Пусть вершины A, B, C, входят в контур Q. Запишем уравнение баланса для этих вершин

$$\begin{cases} a_{11} x_C^A + a_{21} x_C^B \geq P_i^* + (x_A^A + \gamma x_A^B) \\ a_{11} x_B^A + a_{21} x_B^B \geq P_k^* + (x_C^A + \gamma x_C^B) \\ a_{11} x_A^A + a_{21} x_A^B \geq P_l^* + (x_B^A + \gamma x_B^B), \end{cases} \quad (14)$$

где  $P_i^*$ ,  $P_k^*$ ,  $P_l^*$  — приведенные нагрузки вершин A, B, C в вершинах  $y_i \in \Gamma_{y_c}$ ,  $y_k \in \Gamma_{y_b}$ ,  $y_l \in \Gamma_{y_a}$  при условии  $y_i \notin Q$ ,  $y_k \in Q$ ,  $y_l \in Q$ . Процесс принятия решения в контуре Q нельзя представить деревом спуска, что следует из соотношения (\*), справедливого для любой вершины  $y_i \in Q$ :  $\Gamma^n y_i = y_i$  (\*), где  $n$  — порядок отображения, равный длине контура.

Решение в контуре может быть найдено итеративным способом. «Разорвем» контур  $Q$  в любом месте (это целесообразнее делать в том месте, где вершина  $y_q \in Q$  имеет лишь отображение в множестве  $\{y\} \subset Q$ ) и будем искать решение в вершинах контура без учета этого соединения. После того, как решение будет найдено, «замкнем» место разрыва и проверим, выполняется ли уравнение баланса в вершине  $y_q$ . Если да, то на этом решение заканчивается, если нет, то вершине  $y_q$  добавляется дополнительная нагрузка, равная числу элементов в вершине  $y_j = \Gamma y_q$ , полученное в результате первой итерации, и производится следующая итерация. Очевидно, что этот процесс сходящийся и решение будет найдено за конечное число шагов. Для ускорения сходимости этого процесса можно на первой же итерации вершине  $y_q$  добавить нагрузку в один элемент группы минимальной мощности, что не противоречит здравому смыслу, ибо это минимальная из возможных нагрузок.

В случае, когда логическая сеть содержит не простой контур, как это показано на рис. 2, б, следует в первую очередь исключить дугу, принадлежащую обоим контурам. После чего вновь необходимо проверить, образуют ли оставшиеся дуги контур. Если да, то следует исключить еще одну дугу из контура. После этого проводится первая итерация по определению решения аналогично тому, как это делалось в случае простого контура, с последующей проверкой условия энергетического баланса при замыкании контура.

Таким образом, алгоритм согласования элементов принципиальной схемы по нагрузке включает в себя: а) алгоритм поиска контуров в графе, б) алгоритм согласования элементов по нагрузке. Алгоритм согласования по нагрузке реализован на ЭЦВМ «МИР-1».

#### ЛИТЕРАТУРА

1. А. А. Зыков. Теория конечных графов. Новосибирск, «Наука», 1969.
  2. Применение ЭВМ для проектирования цифровых устройств. Сб. статей под редакцией Матюхина М., «Советское радио», 1968.
  3. Р. Белман, С. Дрейфус. Прикладные задачи динамического программирования. «Наука», 1965.
  4. Я. И. Фет. Некоторые алгоритмы проектирования ЦВМ. Вычислительные системы. Вып. 18, Новосибирск 1965.
-