ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Том XV, № 11 РАДИОФИЗИКА 1972

УДК 62—507

ОБ ОДНОМ МЕТОДЕ СИНТЕЗА ФУНКЦИЙ k-ЗНАЧНОЙ ЛОГИКИ

О. А. Башкиров

Рассматривается метод синтеза функций k-значной логики при использовании частоты в качестве носителя информации. Метод состоит в отыскании множества линейных форм (гиперплоскостей в $(n+1)$-мерном пространстве переменных и функции), покрывающих все точки синтезируемой функции и легко реализуемых с помощью операций сложения и вычитания частоты. Из найденного множества гиперплоскостей образуется тупиковое покрытие, состоящее из кусков плоскостей, отделенных друг от друга линейными границами.

В работе [1] изложены теоретические результаты по построению k-значных логических функций. С появлением возможностей реализации некоторых k-значных логических систем представляет интерес изучение практических вопросов синтеза k-значных функций.

В работе [2] было показано, что с помощью операций сложения и вычитания частоты, фильтрации и соединения можно построить произвольную функцию k-значной логики при частотном представлении логических переменных. В данной работе рассматривается метод синтеза функций с использованием частотных элементов, учитывающий некоторые особенности использования частоты в качестве носителя информации.

Пусть для частотных переменных, принимающих значения $0, 1, 2, \dots, N$, определены операции

$$x_1 \dot{+} x_2 = \begin{cases} x_1 + x_2 & (x_1 \neq 0, \quad x_2 \neq 0), \\ 0 & (x_1 = 0 \text{ или } x_2 = 0), \end{cases} \tag{1}$$

$$x_1 \dot{-} x_2 = \begin{cases} x_1 - x_2 & (x_1 \neq 0,\ x_2 \neq 0,\ x_1 \geqslant x_2), \\ 0 & (x_1 = 0 \quad \text{или} \quad x_2 = 0), \\ x_2 - x_1 & (x_1 < x^2), \end{cases} \tag{2}$$

$$x^i = \begin{cases} x & (x = i), \\ 0 & (x \neq i), \end{cases} \tag{3}$$

$$x_1 \dot{\vee} x_2 = \begin{cases} x_1 & (x_2 = 0), \\ x_2 & (x_1 = 0) \\ \text{не определено при } x_1 \neq 0 \text{ и} \\ x_2 \neq 0 \ \text{ одновременно}, \end{cases} \tag{4}$$

где 0 означает отсутствие сигнала, и имеется набор констант $1, 2, 3, \dots, N$.

Пусть задана некоторая функция k-значной логики $f(x_1, x_2, \dots, x_i, \dots, x_n)$. В соответствии с [2] сопоставим с каждым значением переменной $x_i = 0, 1, 2, \dots, k-1$ значения частоты $y_i = c_i,\ c_i + \Delta,\ c_i + 2\Delta, \dots, c_i + (k-1)\Delta$, где $c_i \neq 0$ — несущая частота, и будем строить функцию $g(y_1, y_2, \dots, y_i, \dots, y_n)$, принимающую значения $c_0,\ c_0 + \Delta,\ c_0 + 2\Delta, \dots, c_0 + (k-1)\Delta$, соответствующие значениям $0, 1, 2, \dots, k-1$, где $c_0 \neq 0$. Наличие операций сложения и вычитания частоты позволяет искать форму представления функции в виде некоторого множества линейных комбинаций переменных вида

$$f_j = a_1 x_1 + a_2 x_2 + \dots + a_n x_n + a_0, \tag{5}$$

где $f = \dot{\bigvee} f_j$, причем требования, предъявляемые выбором несущих частот, можно учитывать после выполнения первых этапов синтеза. Предположим, что можно выполнять операции обычного сложения и вычитания над переменными x_i. Тогда первый этап алгоритма синтеза сводится к следующему. Выражение (5) представляет уравнение гиперплоскости в $(n+1)$-мерном пространстве переменных и значений функции, а синтезируемая функция задает некоторое множество точек в этом пространстве (функция и аргументы принимают значения 0, 1, 2, ..., $k-1$).

Пусть задан некоторый набор значений аргументов $x_1^0, x_2^0, \dots, x_n^0$. Возьмем множество наборов вида

$$\begin{gathered} x_1^0 + \alpha,\ x_2^0, \dots, x_n^0; \\ x_1^0,\ x_2^0 + \alpha, \dots, x_n^0, \\ x_1^0,\ x_2^0, \dots, x_n^0 + \alpha, \end{gathered} \tag{6}$$

где $\alpha = \pm 1$.

Очевидно, для каждого набора $x_1^0, x_2^0, \dots, x_n^0$, если соответствующая ему точка (назовем ее исходной точкой) не лежит на границе области определения аргументов, существует 2^n различных множеств наборов вида (6). Проведем гиперплоскости через точки значений функции, соответствующие исходной точке и каждым ее n соседним точкам вида (6). Получим 2^n гиперплоскостей для каждой исходной точки (меньше, чем 2^n для точек, лежащих на границах области). Проведем такие гиперплоскости для всех исходных точек, причем будем запоминать только несовпадающие между собой гиперплоскости. Полученное множество гиперплоскостей можно сравнить с сокращенной ДНФ для булевой функций. Действительно, каждая гиперплоскость покрывает не менее $n+1$ точек (для булевой функции член сокращенной ДНФ покрывает не менее одной точки), в то же время каждая точка функции может быть покрыта несколькими гиперплоскостями (каждая точка может быть покрыта несколькими членами сокращенной ДНФ).

Следующий этап синтеза состоит в выборе какого-либо тупикового покрытия значений функции некоторым подмножеством гиперплоскостей. Эта задача несколько отличается от поиска тупиковой ДНФ по следующим причинам. Во-первых, каждая гиперплоскость может покрыть точки, не принадлежащие функции. Во-вторых, операция (4) не определена при $x_1 \neq 0$ и $x_2 \neq 0$ одновременно, следовательно, для каждого набора значений переменных должна «работать» только одна гиперплоскость. Поэтому будем искать покрытие значений функции в виде множества непересекающихся выпуклых гипермногоугольников, вырезанных из гиперплоскостей. Проекции границ гипермногоугольников в n-мерном пространстве аргументов будут представлять n-мерные гиперплоскости. Каждая такая гиперплоскость (граница), отсекающая часть пространства аргументов, может быть реализована с помощью операции фильтрации, если перейти к переменным y_i. Уравнение границы будет иметь вид

$$b_1 x_1 + b_2 x_2 + \dots + b_n x_n + b_0 \geqslant 0. \tag{7}$$

Пусть в нашем распоряжении имеются фильтры, позволяющие разделять частоты, отличающиеся на Δ. Как легко видеть из геометрического рассмотрения, это ограничение выполняется для границ, имеющих коэффициенты $b_i = 0, \pm 1$. При увеличении коэффициентов b_i до ± 2 необходимо разделять частоты, отличающиеся на $\Delta/2$ и т. д. Поэтому дальше рассматривается алгоритм поиска границ, коэффициенты которых принимают значения $0, \pm 1$, однако алгоритм можно обобщить и для других значений b_i.

Несмотря на имеющиеся отличия, проблема выбора подмножества гиперплоскостей остается аналогичной проблеме поиска минимальной ДНФ для булевых функций. Эта проблема подробно исследована многими авторами. В работе [3] имеется обзор полученных результатов, там же отмечается необходимость выполнения значительного (практически неосуществимого) перебора для поиска минимальной ДНФ среди тупиковых ДНФ. С другой стороны, в [3] доказано, что любая тупиковая ДНФ для почти всех функций не сильно отличается от минимальной.

В данной работе для выбора подмножества гиперплоскостей используется модификация простого алгоритма, использованного в [4] для поиска тупиковой ДНФ булевой функции. Его основная идея состоит в том, что на каждом шаге алгоритма выбирается член ДНФ, покрывающий максимальное число «новых» точек (не покрытых уже выбранными членами). Практическое использование алгоритма показало удовлетворительные результаты [4], однако при этом так же, как и для других алгоритмов, необходимо учитывать существование функций, для которых применяемый алгоритм окажется наихудшим.

Перейдем к рассмотрению алгоритма выбора подмножества гиперплоскостей. Выберем первую по порядку гиперплоскость из множества гиперплоскостей, покрывающих значения функций. По таблице функции находим точку, координаты которой удовлетворяют уравнению гиперплоскости. Эта точка принимается за исходную. Описываем вокруг исходной точки гиперкуб в n-мерном пространстве аргументов, грани которого отстоят от исходной точки на расстояние 1. Проверяем, удовлетворяют ли точки, лежащие на гранях, таблице значений функции при подстановке их координат в уравнение плоскости. Все точки, не удовлетворяющие таблице, отделяются границами вида (7). Затем одна из граней куба отодвигается на расстояние 1, затем следующая грань и т. д., пока не перестанут появляться точки, удовлетворяющие таблице функции при подстановке в уравнение плоскости, в результате чего получаем набор уравнений границ, отделяющих кусок плоскости. Этот процесс повторяется для всех гиперплоскостей, среди них находится гиперплоскость, накрывающая максимальное число точек. После выбора куска гиперплоскости накрытые точки исключаются из рассмотрения и отыскивается новый кусок, покрывающий максимальное количество точек и т. д., пока не будут накрыты все точки. Таким образом, отыскивается любое тупиковое покрытие функции, получающееся в результате выполнения рассмотренного алгоритма.

Когда все уравнения гиперплоскостей и границ выбраны, можно приступить к реализации функции, учитывая ограничения, накладываемые выбором несущих частот. Пусть используется трехчастотный вариант выбора несущих, т. е. несущие частоты всех сигналов в схеме принимают значения c_1, c_2 и $c_3 = c_1 + c_2$, при этом входные и выходные сигналы должны иметь несущие c_1 или c_2, а c_3 используется внутри схемы. Отсюда следуют правила построения схемы: сложение можно выполнять для сигналов, имеющих несущие c_1 и c_2, а вычитание для сигналов с несущими c_3 и c_1 или c_3 и c_2. Для того, чтобы удовлетворить этим требованиям, воспользуемся соотношениями

$$\begin{gathered}((c_{1,2} + x_i\Delta) \dot{+} c_{2,1}) \dot{-} c_{1,2} = c_{2,1} + x_i\Delta,\\ (c_{1,2} + x_i\Delta) \dot{+} c_{2,1} = c_3 + x_i\Delta,\end{gathered} \tag{8}$$

позволяющими изменять несущую, не изменяя значения x_i и

$$c_3 \dot{-} (c_{1,2} + x_i \Delta) = c_{2,1} - x_i \Delta, \tag{9}$$

позволяющими заменять сложение значений x_i вычитанием при сложении несущих.

Тогда процесс построения схемы, реализующий функцию, сводится к следующему. С помощью операций сложения и вычитания и фильтров реализуем уравнение первой гиперплоскости. Из уравнения границы строим уравнение второй границы и т. д., из уравнения последней границы строим уравнение первой гиперплоскости. При этом порядок реализации границ подбираем так, чтобы переход от одной границы к другой и к уравнению гиперплоскости происходил достаточно просто. Для этого введем понятие расстояния между гиперплоскостью и ее границей

$$R = \sum_{i=1}^{n} | a_i - b_i | , \tag{10}$$

аналогично расстояние между двумя плоскостями —

$$R = \sum_{i=1}^{n} | b_i^1 - b_i^2 |. \tag{11}$$

Выберем границу, имеющую минимальное расстояние с реализуемой плоскостью, затем найдем границу, имеющую минимальное расстояние с реализуемой границей и т. д. С последней из найденных таким образом границ начинаем построение искомой схемы плоскости. Выходные сигналы всех схем, реализующих плоскости, объединяются с помощью элементов $\dot{\vee}$. Ниже приведены результаты применения программы синтеза, реализующей описанный алгоритм к нескольким функциям двух переменных при $k = 5$. Причем далее, для упрощения записи предполагается $\Delta = 1$, а для операции фильтрации введено обобщенное обозначение

$$y_m^{i,j} = \begin{cases} y_m & (c_m + i \leqslant c_m + x_m \leqslant c_m + j), \\ 0 & (y_m < c_m + i,\ y_m > c_m + j), \end{cases} \tag{12}$$

где $y_m = c_m + x_m$.

Пример 1. $f(x_1, x_2) = x_1 + x_2 (\mathrm{mod}\ 5)$.

Входные переменные $y_1 = c_1 + x_1$, $y_2 = c_2 + x_2$. Функция реализуется в виде двух кусков плоскостей

$$f = x_1 + x_2 \qquad (x_1 + x_2 \leqslant 4),$$
$$f = x_1 + x_2 - 5 \qquad (x_1 + x_2 > 4).$$

Перейдем к представлению с учетом несущих частот

$$g(y_1, y_2) = ((y_1 \dot{+} y_2)^{0,4} \dot{-} c_1) \dot{\vee} ((y_1 \dot{+} y_2)^{5,8} \dot{-} (c_1 + 5)).$$

Несущая выходного сигнала c_2.

Блок-схема реализации функции приведена на рис. 1, где 1 — элемент, выполняющий операцию сложения, 2 — элемент вычитания, 3 — фильтр, 4 — элемент $\dot{\vee}$.

Пример 2. $f(x_1, x_2) = \max(x_1, x_2)$.

Функция реализуется в виде двух кусков плоскостей

$$f = x_1 \qquad (x_1 - x_2 \geqslant 0),$$
$$f = x_2 \qquad (x_2 - x_1 > 0).$$

Перейдем к представлению с учетом несущих частот

$$g(y_1,\ y_2) = (((c_2 \dot{+} y_1) \dot{-} y_2)^{0,4} \dot{+} y_2) \dot{\vee} ((c_1 \dot{+} y_2) \dot{-} y_1)^{1,4} \dot{+} y_1)) \dot{-} c_1.$$

Блок-схема реализации функции приведена на рис. 2.

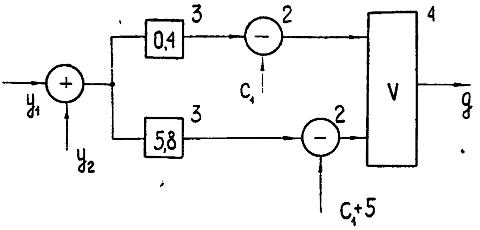

Рис. 1.

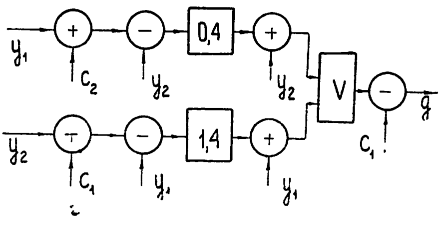

Рис. 2.

Пример 3. Функция $f(x_1,\ x_2)$ задана табл. 1.

Таблица 1

x_2 \ x_1	0	1	2	3	4
0	3	1	4	1	1
1	4	2	1	0	3
2	3	4	2	4	3
3	3	2	3	0	1
4	1	2	2	1	2

Функция реализуется в виде девяти кусков плоскостей, приведенных на рис. 3.

1. $f = -x_1 + x_2 + 2,\quad x_2 \geqslant 1,$

$x_1 + x_2 \leqslant 4,\quad -x_1 + x_2 - 2 \leqslant 2;$

2. $f = x_1 - 2x_2 - 7,$

$1 \leqslant -x_1 + x_2 \leqslant 2,\quad 2 \leqslant x_2 \leqslant 3,$

3. $f = x_1 + x_2 - 6,\quad x_1 \geqslant 3,\quad x_2 \geqslant 3;$

4. $f = -2x_1 + x_2 + 3,\quad x_1 + x_2 \leqslant 1;$

5. $f = x_2 + 1,\quad -x_1 + x_2 \leqslant -3;$

6. $f = x_1 - x_2 + 9,\quad x_1 - x_2 \leqslant -3;$

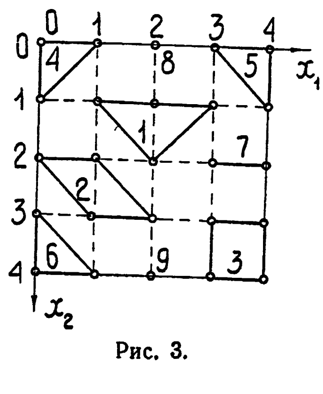

Рис. 3.

7. $f = -x_1 + 4x_2 - 1,\quad x_1 \geqslant 3,\quad x_1 \leqslant 4,\ x_2 = 2;$

8. $f = 4,\quad x_1 = 2,\quad x_2 = 0;$ 9. $f = 2,\ x_1 = 2,\ x_2 = 4.$

Поскольку 7-й кусок состоит из двух точек, его выгоднее реализовать в виде двух отдельных схем, использовав операцию $x_1 + x_2 - c$. Аналогично реализуются 8-й и 9-й куски. Блок-схема реализации функции приведена на рис. 4.

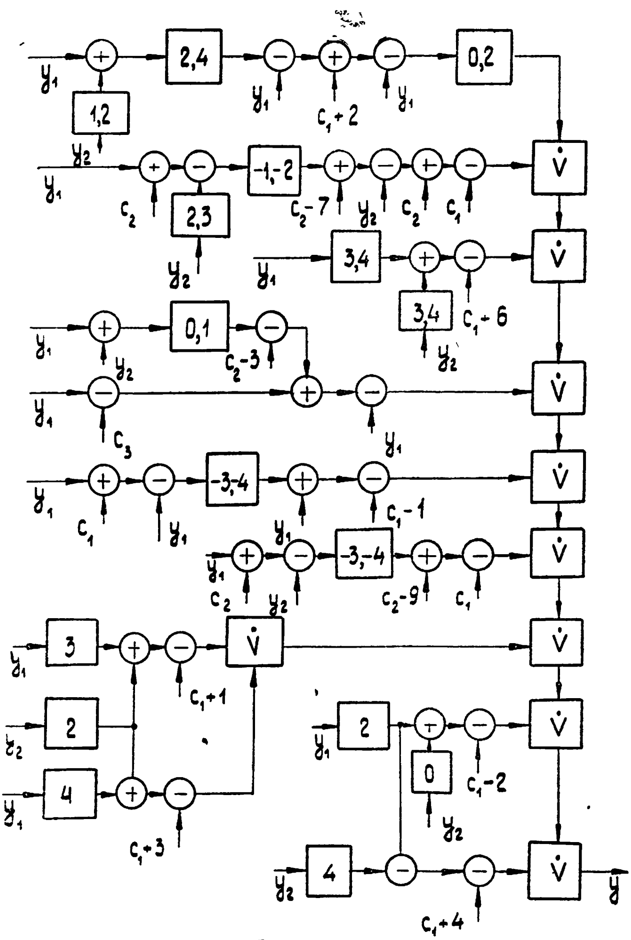

Рис. 4.

ЛИТЕРАТУРА

1. С. В. Яблонский, Тр. Математического института им. Стеклова, т 51, М., 1958.
2. О. А. Башкиров, Тезисы докладов и сообщений на Всесоюзном межвузовском симпозиуме по прикладной математике и кибернетике, Горький, 1967.
3. В. В. Глаголев, Сб. Проблемы кибернетики, вып. 19, изд. Наука, М., 1967, стр. 75.
4. О. А. Башкиров, С. И. Башкирова, Изв. высш. уч. зав. — Радиофизика, 14, № 7, 1122 (1971).

Научно-исследовательский институт
прикладной математики и кибернетики
при Горьковском университете

Поступила в редакцию
2 сентября 1970 г.

ON ONE METHOD FOR SYNTHESIS OF k-VALUED LOGIC FUNCTIONS

O. A. Bashkirov

A method is considered for synthesis of k-valued logic functions with frequency used as the information carrier The method consists in finding the multitude of linear forms (hyperplanes in an $(n+1)$-dimensional space of variables and a function) covering all the points of the synthetized function and easily realized by the operations of frequency addition and subtraction. The found multitude of hyperplanes forms an obtuse covering consisting of plane pieces separated one from another by linear boundaries.