Настраиваемые модули из функциональных элементов

3-1. ПОРОЖДАЮЩИЕ ФУНКЦИИ. СВОЙСТВА И ХАРАКТЕРИСТИКИ

Выше мы уже говорили, что основная цель создания НЛМ - сокращение номенклатуры составляющих схем при повышении уровня интеграции. Платой за достигаемый эффект при этом является избыточность по числу наружных выводов и внутренних элементов.

С теоретической точки зрения построить НЛМ - это значит так объединить между собой заданные функции, чтобы затем путем настройки можно было получить любую из этих функций.

Например, широко используют такие функции от двух переменных, как «И» и «ИЛИ», т. е. $x_1x_2, x_1 \vee x_2$. Как их объединить? Это зависит от выбранного способа настройки (классификация способов приведена в предыдущей главе).

Простейший вариант - механическое объединение функций. Он малоинтересен, так как при этом у соответствующего модуля отсутствует функциональная интеграция: сложность модуля по числу внешних выводов и числу составляющих совпадает с этими характеристиками у объединяемых элементов.

Другой вариант - настройка путем выбора нужного выхода, в случае если входные выводы соответствующих элементов объединены.

Третий вариант - настройка по входам. Выходной вывод один, а входных три: один настроечный и два информационных. При этом модуль выполняет функцию $y = z_1z_2 \vee z_3$.

Легко заметить, что при подаче сигналов $z_1 = x_1, z_2 = x_2, z_3 = 0$ получим первую из требуемых функций $y = x_1x_2$, а при подаче $z_1 = x, z_2 = 1, z_3 = x_3$ - вторую функцию $y = x_1 \vee x_2$.Аналогичным будет результат в случае приравнивания входов: $z_1 = z_2 = x_1, z_3 = x_2$. Может сложиться впечатление, что настройка по входам ничего не дает по сравнению с настройкой по выходам: в обоих случаях число внешних выводов равно четырем, а внутренняя сложность модуля определяется суммой объединяемых элементов «И» и «ИЛИ». Однако так получилось лишь в частном случае, из-за того, что объединяемые функции не имеют общих составных частей. Если добавить к ним еще одну функцию, например $x_1\overline{x_2} \vee \overline{x_1}x_2$, то настройка выбором выхода потребует при наилучшем построении схемы пять внешних выводов и четыре двухвходовых элемента. Настройка же по входам также потребует четырех внутренних элементов, но число внешних выводов будет меньше - четыре вместо пяти.

Функция, выполняемая модулем, в этом случае имеет вид:

$$y = z_1\overline{z_2z_3} \vee z_1z_2z_3.$$

При $z_3 = 1$; $z_1 = \overline{x}_1$; $z_2 = \overline{x}_2$ $y = x_1 \vee x_2$;

При $z_3 = 0$; $z_1 = x_1$; $z_2 = x_2$ $y = x_1x_2$;

При $z_3 = x_2$; $z_1 = z_2 = x_1$ $y = \overline{x}_1x_2 \vee x_1\overline{x}_2$;

При настройке по входам применяются следующие операции: подача на определенные входы констант 0 и 1, приравнивание входов друг к другу, приравнивание к инверсии и подача на информационные входы нужного сочетания прямых и инверсных значений входных переменных.

Если последнее возможно, то должна использоваться PN - классификация; если допустима только перестановка входных переменных без их инвертирования, то – Р - классификация, и, наконец, при жесткой связи с источниками информации приходится объединять заданные функции, даже если отличие между ними состоит лишь в порядке следования букв.

С возрастанием числа объединяемых функций относительные преимущества настройки по входам возрастают, даже если общие части отсутствуют.

Пусть, например, в списке имеется сто функций от десяти переменных. Тогда при настройке по выходам потребуется сто десять внешних выводов, а при самой элементарной настройке по входам (без использования операции приравнивания входов и при полном разделении информационных и настроечных входов): $10 +]\log_2 100[= 17$ внешних выводов, т. е. имеет место выигрыш в 6,5 раза!

Иногда варианты настройки по входам и выходам объединяют. Объясняется это тем, что от одних и тех же переменных могут потребоваться одновременно разные функции. Настройка по выходам это обеспечит, в то время как настройка по входам - нет. Чаще всего требуется одновременно использовать прямое и

Таблица 3-1

Оценки числа входов модулей, универсальных в классе всех булевых функций от *n* переменных и имеющих два выхода - прямой и инверсный

n	2	3	4	5	6	7	8	9	10
Нижняя оценка	3	5	6	10	17	32	62	119	230
Верхняя оценка	3	5	7	16	28	39	72	137	266

инверсное значения функции и поэтому у модулей в этих случаях предусматривают два выхода: y и $\overline{y}$.

Функция, которую выполняет объединенный модуль при использовании настройки по входам, носит название порождающей (ПФ). Она порождает, в первую очередь, требуемый список функций, а кроме того, еще ряд функций, которые также могут оказаться полезными. При этом порожденные функции зависят от меньшего числа переменных, чем ПФ.

Различные ПФ обладают различной порождающей способностью. В литературе [16, 34, 37] были исследованы признаки, по которым можно найти наиболее эффективные функции, порождающие максимальное число подфункций. Однако, взяв одну из таких функций за основу при построении модуля, мы можем ошибиться, так как подфункций она порождает много, но, возможно, не тех, которые нам требуются. Проверка и выбор «правильной» функции связаны с большим перебором, затруднительным даже при использовании ЦВМ.

Подход, состоящий в выборе нужной ПФ из числа наиболее эффективных, применялся при построении модулей, универсальных в классе всех булевых функций от двух, трех и четырех аргументов [9, 69]. При большем числе переменных удалось установить только верхние и нижние оценки числа внешних выводов универсальных модулей, приведенные в табл. 3-1.

Для модулей этого типа была найдена также асимптотическая оценка числа входов, которая имеет следующий вид [25, 52]:

$$\underset{n\to\infty}{M}(n) = \frac{2^n}{\log_2 n}. \qquad (3\text{-}1)$$

Как было отмечено выше, подход, основанный на исчерпывающем исследовании логических возможностей булевых функций, связан с большим перебором, поэтому в литературе рассматривались также процедуры построения модулей без использования перебора или с меньшим его объемом, приводящие обычно к построению модулей с большим числом внешних выводов.

Наиболее простой метод построения модулей основан на том факте, что произвольная булева функция представима в виде

$$f(x_1,...,x_n) = \sum_{i=0}^{n-1} x_1^{i_1}...x_n^{i_1} f(i_1,...,i_n), \qquad (3\text{-}2)$$

где

$$x_j^i j = \begin{cases} x_j \;\; при \;\; i_j = 1; \\ \overline{x_j} \;\; при \;\; i_j = 0; \end{cases}$$

На основе этого соотношения может быть построен модуль, универсальный в классе всех булевых функций от *n* переменных, имеющий число внешних выводов

$$M(n) = 2^n + n + 1. \qquad (3\text{-}3)$$

Этот модуль имеет 2^n настроечных входов, на которые должны подаваться только константы 0 и 1.

Если допустить подачу на настроечные входы не только констант, но и переменных, то число внешних выводов универсальных модулей может быть уменьшено [74]. Это достигается при построении модуля на основе соотношения

$$f(x_1,...,x_n)=\sum_{j=0}^{2^{n-1}-1} x_1^{i_1}...x_{n-1}^{i_{n-1}} f_i(i_1,...,i_{n-1},x_n). \quad (3\text{-}4)$$

Число внешних выводов такого модуля

$$M(n)=2^{n-1}+(n-1)+1=2^{n-1}+n. \quad (3\text{-}5)$$

В этой же работе впервые для класса модулей из ФЭ был предложен метод построения модулей, универсальных в классе всех булевых функций из двух составляющих.

В изложенных подходах основное внимание уделялось построению модулей, универсальных в классе всех булевых функций. Рассмотрим методы построения модулей, реализующих путем настройки заданный набор функций.

Модуль, реализующий путем настройки *N* заданных булевых функций я переменных, может быть построен на основе соотношения

$$F(x_1,...,x_{n+d})=\sum_{i=1}^{N} x_{n+d}^{j_{n+1}}...x_{n+1}^{j_{n+1}} f_i(x_n,...,x_1), \quad (3\text{-}6)$$

где $d=]\log_2 N[;][$ - символ округления до ближайшего большего целого числа.

Информационные и настроечные входы этого модуля независимы, а его настройка осуществляется путем подачи констант. Число внешних выводов такого модуля

$$M=]\log_2 N[+n-1. \quad (3\text{-}7)$$

Для уменьшения числа внешних выводов необходимо отказаться, хотя бы частично, от жесткого разделения информационных и настроечных входов. В предельном случае любой вход может являться как информационным, так и настроечным:

$$F(x_1,...,x_{n+d})=\sum x_{n+d}^{j_{n+d}}...x_1^{j_1} c_i(j_{n+d},...,j_1), \quad (3\text{-}8)$$

где

$$c_i(j_{n+d},...,j_1)=\begin{cases}0;\\1.\end{cases}$$

Из этого соотношения следует, что построение такого модуля связано с нахождением значений c_i и d.

Э. А. Якубайтисом [59] был предложен метод построения модулей с минимальным числом выводов, реализующих путем настройки заданный набор из *N* булевых функций, базирующийся на соотношении (3-8). Сущность метода состоит в отборе на каждом шаге конъюнкций определенного ранга, включаемых в порождающую функцию, и в нахождении настроек, обеспечивающих выделение из получаемой порождающей функции заданных функций *n* переменных. Метод позволяет строить модули с минимальным числом внешних выводов, вследствие того что поиск оптимального значения выполняется «снизу вверх» от конъюнкций ранга $n+1$. Однако использование этого метода ограничивается относительно небольшими значениями *N* и *n,* так как в противном случае объем перебора становится чрезвычайно большим.

Наряду с общими подходами к построению модулей, изложенными выше, в настоящее время из литературы известны работы, посвященные модулям, универсальным в некоторых классах булевых функций.

В работах [18, 74] рассматривались модули, универсальные в классе симметрических функций *n* переменных. При этом было установлено, что число внешних выводов таких модулей равно *2n,* а их сложность линейно зависит от числа переменных.

В работе [60] были рассмотрены модули, универсальные в классе пороговых функций. Построение этих модулей выполнялось с использованием ЦВМ. При этом было установлено, что существует модуль с восьмью входами, универсальный в указанном классе функций от пяти переменных, и модуль с десятью входами, универсальный в том же классе функций от шести переменных.

В работах [12, 44] были предложены подходы, на основе которых была разработана структура многофункционального модуля, реализующего путем настройки 26. бесповторных в базисе $\{\&,\vee,\bar{\ }\}$ формул из восьми букв. Другой тип многофункциональных модулей, реализующих путем настройки некоторые бесповторные формулы в базисе $\{\&,\vee,\bar{\ }\}$ из четырех букв, выпускается в настоящее время в виде микросхемы, входящей в серию 501 [55].

К сожалению, большинство методов, изложенных в указанных выше работах, носят частный характер, не имеют достаточного математического обеспечения и неприменимы поэтому в общем случае. Перейдем к изложению метода, позволяющего устранить эти недостатки.

3-2. ПОСТРОЕНИЕ МОДУЛЕЙ С МИНИМАЛЬНЫМ ЧИСЛОМ ВНЕШНИХ ВЫВОДОВ

Ограничимся рассмотрением модулей, настраиваемых по входам с помощью следующих операций: а) фиксация входов константами $(x_i = 0;1)$; б) отождествление входов $(x_j = x_i)$; в) антиотождествление входов $(x_j = \overline{x_i})$.

Входы, с помощью которых осуществляется настройка, образуют настроечное поле, остальные - информационное. Иногда настроечное поле жестко отделяют от информационного: например, из пяти входов первые три используют всегда для подачи логических переменных, а два остальных - только для настройки.

Иногда разделение на настроечные и информационные входы носит условный характер - в каждом конкретном случае тот или иной входной вывод может служить для настройки.

Второй из указанных способов «богаче», так как появляется возможность различных вариантов выбора информационных и настроечных входов.

Пример. Пусть задана формула

$$y = \overline{x_1}(x_2 \vee \overline{x_3}) \vee x_1\overline{x_2}x_3$$

Если в этой формуле настроечной переменной считать всегда только x_1, то получим две подфункции от двух аргументов: $y = x_2 \vee \overline{x_3}$ при $x_1 = 0$ и $y = \overline{x_2}x_3$ при $x_1 = 0$. Если же считать, что настроечными могут быть как x_1, так и x_2, то получим три подфункции: $y = x_2 \vee \overline{x_3}$ при $x_1 = 0$, $y = \overline{x_2}x_3$ при $x_1 = 1$ и $y = \overline{x_1}\overline{x_3} \vee x_1x_3$ при $x_2 = 0$.

Как правило, жесткое разделение информационного и настроечного полей выступает как ограничение, связанное с требованиями технологии или выбором структуры. В каких случаях оно имеет место, мы рассмотрим в дальнейшем. А сейчас попробуем глубже разобраться в процессе порождения подфункций. Что именно происходит с ПФ, когда мы фиксируем или отождествляем (антиотождествляем) входы?

Предположим, что ПФ найдена и для нее построена таблица истинности. Условимся в таблице всегда располагать настроечные переменные на позициях старших разрядов. Пусть число информационных переменных равно *a,* а настроечных *d.* Тогда при

Таблица 3-2

Число настроек $\overline{\mathrm{N}}$(d;b)

d	*b*									
	0	1	2	3	4	5	6	7	8	9
1	2	1								
2	4	6	1							
3	8	28	12	1						
4	16	120	100	20	1					
5	32	496	720	260	30	1				
6	64	2016	4816	2800	560	42	1			
7	128	8128	30912	27216	8400	1064	56	1		
8	256	32640	193600	248640	111216	21168	1848	72	1	

9	512	130816	1194240	2182720	1360800	365232	44640	3000	90	1

фиксации всех настроечных переменных константами от столбца значений ПФ останется подфрагмент длиной 2^a. При этом число различных подфрагментов равно 2^d.

Каждый подфрагмент - это не что иное, как одна из остаточных функций в разложении Шеннона [57], которое производится по настроечным аргументам.

Пример. Пусть задана функция

$$y = x_1 x_2 \vee x_3$$

Предположим, что x_1 - настроечная переменная; тогда разложение Шеннона по переменной x_1 имеет следующий вид:

$$y = \bar{x}_1 x_3 \vee x_1 (x_2 \vee x_3).$$

При этом имеют место два подфрагмента: первый x_3, а второй $x_2 \vee x_3$. Как раз такие подфункции и получаются при соответствующей настройке: $y = x_3$ при $x_1 = 0$ и $y = x_2 \vee x_3$ при $x_1 = 1$.

Теперь рассмотрим случай, когда фиксируется только часть переменных настроечного поля, а остальные b переменных остаются свободными. При такой операции осуществляется выбор и объединение некоторых подфрагментов. Они также образуют «остаточную функцию». Тем самым мы можем не только выделять подфрагменты, но и объединять их между собой. Каждый вариант выделения и объединения - это новая настройка, порождающая функцию, но не обязательно новую, так как среди подфрагментов могут быть и одинаковые.

Таблица 3-3

Число настроек N(d;b)

d	*b*								
	0	1	2	3	4	5	6	7	8
1	2	1							
2	4	5	1						
3	8	19	9	1					
4	16	65	55	14	1				
5	32	211	285	125	20	1			
6	64	665	1551	910	245	27	1		
7	128	2059	6069	5901	2380	434	35	1	
8	256	6305	26335	35574	20181	5118	714	44	1

Оценки числа вариантов настройки. Число возможных вариантов настройки определяется числом переменных в настроечном поле d, числом свободных переменных b и числом используемых операций настройки.

В работе [52] найдена формула для подсчета числа настроек $\overline{N}(d;b)$ порождающей функции, содержащей d переменных, для реализации остаточной функции b переменных при использовании операций фиксации, отождествления и антиотождествления:

$$\overline{N}(d;b) = 2^{d-b} S(d+1;b+1), \qquad (3\text{-}9)$$

где $S(d+1;b+1)$ - число Стирлинга второго рода [49].

В табл. 3-2 приведены значения $\overline{N}(d;b)$, подсчитанные по формуле (3-9) при $d \le 9, b \le 9$.

В той же работе найдена также формула для подсчета числа настроек $N(d;b)$ ПФ, содержащей d переменных, для реализации остаточной функции b переменных при использовании операций фиксации и отождествления переменных:

$$N(d;b) = S(d+2;b+2) - S(d+1;b+2). \qquad (3\text{-}10)$$

В табл. 3-3 приведены значения $N(d;b)$, подсчитанные по формуле (3-10) при $d \le 8, b \le 8$.

Приведенные соотношения и количественные характеристики позволяют определить нижние оценки числа переменных g в ПФ, обеспечивающие число настроек, большее или равное числу заданных функций *N,* зависящих от b переменных. Эти оценки могут быть получены в случае, если в ПФ информационные и настроечные переменные совпадают.

При атом $g = d$ и справедливы неравенства

$$\overline{N}(d;b)=2^{d-b}S(d+1;b+1)\geq N(b), \qquad (3\text{-}11)$$

$$N(d;b)=S(d+2;b+2)-S(d+1;b+2)\geq N(b). \qquad (3\text{-}12)$$

Объединение подфрагментов для получения остаточных функций. Если из d переменных настроечного поля b являются свободными, то с их помощью могут объединяться 2^b подфрагментов длины 2^a. Номенклатура объединяемых подфрагментов зависит от вида операции настройки. Обозначим операцию объединения столбцов значений подфрагментов знаком *.

Пример. Пусть задана булева функция от $g = 2$ переменных. Предположим, что $a = 0$, $d = 2$, $b=1$. При этом длина каждого подфрагмента

Таблица 3-4

Объединение подфрагментов при $a = 0$, $d = 2$, $b=1$

x_2	x_1	F	$x_2=0$	$x_2=1$	$x_1=0$	$x_1=1$	$x_2=x_1$	$x_2=\overline{x}_1$
0	0	A	A	-	A	-	A	-
0	1	B	B	-	-	B	-	B
1	0	C	-	C	C	-	-	C
1	1	D	-	D	-	D	D	-

равна $2^a=1$, а их число $2^d=4$. Обозначим их буквами *A, B, C, D.* Выполняя операции настройки, получим шесть различных фрагментов, состоящих из $2^b=2$ подфрагментов (табл. 3-4):

$$A*B;\quad C*D;\quad A*C;\quad B*D;\quad A*D;\quad B*C.$$

Предположим, что $a=1$, $d=1$, $b=0$. При этом длина подфрагментов $2^a=2$, а их число $2^d=2$. Обозначим подфрагменты буквами *G, K.* Выполняя операции настройки, получим два фрагмента, совпадающих с соответствующими подфрагментами $(2^b=1)$ (табл. 3-5).

Определим условия, при выполнении которых имеется возможность объединить любые 2^b подфрагментов ПФ из общего их числа 2^b в остаточную функцию таким образом, чтобы подфрагмент, имеющий меньший номер в столбце значений ПФ, был размещен в остаточной функции над подфрагментом с большим номером. Для того чтобы иметь возможность выполнить указанное объединение подфрагментов при использовании операций фиксации переменных константами и отождествлении переменных, необходимо выполнение соотношения

$$N(d;b)\geq C_{2^d}^{2b}, \qquad (3\text{-}13)$$

Таблица 3-5

Объединение подфрагментов при $a=1$, $d=1$, $b=0$

x_2	x_1	F	$x_2=0$	$x_2=1$
0 0	0 1	G	G	-
1 1	0 1	K	-	K

а при использовании также операции «Антиотождествление» требуется, чтобы выполнялось соотношение

$$\overline{N}(d;b)\geq C_{2^d}^{2b}. \qquad (3\text{-}14)$$

Исследуем, при каких значениях b эти соотношения выполняются.

Случай $1: b=0$**.** Использование операции фиксации переменных константами позволяет выбрать любой одиночный подфрагмент столбца значений ПФ. Действительно, из табл. 3-3 следует, что число настроек при $b=0$ равно 2^d.

Случай $2:b=1$**.** Использование двух операций — фиксации переменных константами и отождествления переменных — не позволяет объединить любую пару подфрагментов ПФ в остаточную функцию так, чтобы подфрагменг с меньшим номером всегда размещался над подфрагментом с большим номером.

Пусть задано настроечное поле ПФ из d переменных. Объединение любых двух из 2^d подфрагментов ПФ так, как это было указано выше, возможно C_{2d}^2 способами. С другой стороны, число различных вариантов настройки, объединяющих подфрагменты, которое обеспечивает это настроечное поле для одной свободной переменной к при использовании указанных операций, на основании (3-10) равно

$$N(d;1)=S(d+2;3)-S(d+1;3). \qquad (3\text{-}15)$$

При этом справедливость неравенства

$$C_{2^d}^2 < S(d+2;3)-S(d+1;3) \qquad (3\text{-}16)$$

доказывает высказанное утверждение.

Пример. Пусть $d=3$, тогда $C_{2^3}^2=28$. При этом, так как *N(3, 2) = 19*,

то не удается объединить любую пару подфрагментов ПФ в остаточную. Докажем следующее утверждение. Использование трех операций настройки, указанных в начале параграфа, позволяет объединить любую пару подфрагментов ПФ в остаточную так, что подфрагмент с меньшим номером всегда будет размещен над подфрагментом с большим номером.

Пусть задано настроечное поле ПФ из d переменных. Объединение любых двух из 2^d подфрагментов ПФ так, как это указано в формулировке утверждения, возможно $C_{2^d}^2$ способами. С другой стороны, число различных вариантов настройки для одной свободной переменной и использования указанных в формулировке утверждения операций на основании (3-10) равно

$$\overline{N}(d;1)=2^{d-1}S(d+1;2). \qquad (3\text{-}17)$$

При этом справедливость равенства

$$C_{2^d}^2=2^{d-1}S(d+1;2) \qquad (3\text{-}18)$$

доказывает утверждение.

Введенные выше операции настройки позволяют объединить пару подфрагментов в указанном порядке. Для обеспечения возможности их объединения в произвольном порядке введем четвертую операцию настройки - замену переменных их инверсиями. При этом справедливо следующее утверждение: использование четырех операций настройки позволяет объединить любую пару подфрагментов ПФ в остаточную функцию при произвольном порядке их расположения в ней.

Случай $3:b=2$**.** Пусть в настроечном поле имеются две свободные переменные и используются три операции настройки; тогда для $d=3$ число настроек $\overline{N}(3;2)=12$. Следовательно, в этом случае осуществляется выбор и объединение четырех подфрагментов из восьми в двенадцати различных сочетаниях. С другой стороны, общее число сочетаний из 2^d подфрагментов по четыре при условии, что в столбце значений остаточной функции каждый подфрагмент ПФ с меньшим номером будет расположен над подфрагментом ПФ с большим номером, равно $C_{2^d}^4$. При $d=3$ $C_8^4=70$. Поэтому уже при $d=3$ не удается выбрать и объединить в остаточную функцию произвольные четыре подфрагмента ПФ.

В общем случае объединение произвольных четырех подфрагментов ПФ при использовании трех операций настройки не удается, так как

$$C_{2^d}^4 > \overline{N}(d;2) \qquad (3\text{-}19)$$

Например, при $d=6$ $C_{64}^4=635376$, в то время как $N(6;2)=4816$.

Так как объединение произвольных четырех подфрагментов невозможно, а выбор определенных сочетаний из четырех подфрагментов связан с большим перебором, то в дальнейшем будем рассматривать лишь случай, когда объединению подлежат два подфрагмента.

Построение порождающей функции настраиваемого модуля.

При построении модулей из ФЭ основным критерием является число внешних выводов, которое при заданных функциональных возможностях требуется минимизировать [32].

Рассмотрим следующую задачу. Пусть дан список из *N булевых* функций, каждая из которых существенно зависит от *n* переменных: $f_1(x_n,...,x_1), f_2(x_n,...,x_1),..., f_N(x_n,...,x_1)$. Требуется построить булеву функцию $F(z_g,...,z_1)$, зависящую от возможно меньшего числа переменных $g > n$, которая способна в результате выполнения над ней операций настройки порождать любую из *N* упомянутых функций.

Под возможно меньшим числом переменных *g* будем подразумевать не абсолютный минимум, а минимум, достижимый при реальном объеме перебора.

Из изложенного ранее следует, что для построения ПФ надо решить две основные задачи: а) сформировать на основании списка $f_1, f_2,..., f_N$ определенный состав подфрагментов; б) разместить подфрагменты в рациональном порядке, позволяющем выбирать и объединять их путем настройки. Решение этих задач позволяет математически обеспечить метод построения ПФ.

Простейший вариант решения - использовать в качестве подфрагментов сами функции $f_1, f_2,..., f_N$. При этом порядок их расположения может быть произвольным, а настроечное поле отделено от информационного. Число настроечных переменных в этом случае определяется из соотношения

$$d =]\log_2 N[. \qquad (3\text{-}20)$$

ПФ при таком подходе имеет следующий вид:

$$F = \bigvee_{i=1}^{N} z_{n+d}^{\sigma_d} z_{n+d-1}^{\sigma_{d-1}} ... z_{n+1}^{\sigma_1} f_i(z_n,...,z_1), \qquad (3\text{-}21)$$

где

$$z_{n+j}^{\sigma_j} = \begin{cases} z_j & при \;\; \sigma_j = 1; \\ \overline{z_j} & при \;\; \sigma_j = 0. \end{cases}$$

Пример. Построить ПФ для следующего набора функций:

$$f_1 = x_1 x_2 x_3;$$
$$f_2 = x_1 \overline{x}_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee \overline{x}_1 x_2 x_3 \vee x_1 x_2 x_3;$$
$$f_3 = x_1 \vee x_2 \vee x_3;$$
$$f_4 = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3.$$

Поскольку $N = 4$, то $d = \log_2 4 = 2$.

Введем настроечные переменные x_4 и x_5; тогда функция

$$F_1 = \overline{x}_5 \overline{x}_4 f_1 \vee \overline{x}_5 x_4 f_2 \vee x_5 \overline{x}_4 f_3 \vee x_5 x_4 f_4$$

является порождающей.

Точно так же порождающими являются функции с любым другим расположением подфрагментов, например функция

$$F_2 = \overline{x}_5 \overline{x}_4 f_2 \vee \overline{x}_5 x_4 f_4 \vee x_5 \overline{x}_4 f_1 \vee x_5 x_4 f_3$$

Несмотря на то, что модули, соответствующие этим функциям, имеют одинаковое число внешних выводов, их элементная сложность различна: некоторые формулы минимизируются в большей, некоторые в меньшей степени.

В изложенном способе формирования ПФ вид порождаемых функций f_i никак не влияет на число входов. Оно всегда составляет $g' = n +]\log_2 N[$. В то же время в функциях f_i могут быть общие составные части.

Возникает предположение, что наличие одинаковых составных частей в различных функциях может быть использовано с целью сокращения суммарного числа подфрагментов. В самом деле, какой смысл

дважды вносить в ПФ идентичные «половины» двух функций? Может быть, достаточно записать в ПФ каждую из составных частей функции f_i всего один раз, а потом, объединяя их, составлять требуемые функции?
Предположение это оправданно. Только, к сожалению, в большинстве случаев не удается в качестве «строительного материала» ПФ использовать составные части, меньшие чем «половина» функции f_i, так как выше мы показали, что любые два подфрагмента можно путем настройки объединить в любом порядке. Этот вывод по отношению к четырем, восьми и большему числу подфрагментов несправедлив.
Следовательно, для формирования ПФ достаточно: а) определить номенклатуру подфрагментов длиной: 2^{n-1}, из которых могут быть сформированы заданные функции; б) объединить подфрагменты в одну функцию, которая и будет соответствовать искомой ПФ.
Формирование подфрагментов осуществляется путем разложения каждой из функций f_i по одной из переменных. Так как допустимо произвольное переименование входных переменных, то будем считать, что разложение производится по переменной x_n:

$$f_i(x_n,...,x_1) = \bar{x}_n f_1(0,x_{n-1},...,x_1) \vee x_n f_i(1,x_{n-1},...,x_1) = \bar{x}_n \varphi_j^{'}(x_{n-1},...,x_1) \vee x_n \varphi_j^{''}(x_{n-1},...,x_1). \quad (3\text{-}22)$$

При табличной форме задания, когда переменные имеют порядок расположения $x_n, x_{n-1},...,x_1$ и разложение функции выполняется по переменной x_n, одному подфрагменту соответствует верхняя половина столбца значений функции, а другому - нижняя половина.

Разложим каждую из *N* функций по одной переменной и предположим, что суммарное число различных подфрагментов у всех этих *N* функций равно r. *Два подфрагмента будем называть различными, если порядки расположения нулей и единиц в их столбцах значений не совпадают.*

Необходимым условием для того, чтобы некоторая функция была порождающей, является входимость каждого из r подфрагментоп, указанных выше, в столбец значений этой функции по крайней мере один раз. Выполнение этого условия обеспечивает минимальную номенклатуру подфрагментов, которую должен иметь столбец значений функции, претендующей на роль порождающей для заданного набора из *N* функций.

Если подфрагменты в функции могут быть выбраны в любом сочетании и порядке, то их минимальная номенклатура является и достаточной, так как ни одна из *N* заданных функций не должна иметь двух одинаковых подфрагментов длиной 2^{n-1} ввиду того, что в противном случае она зависела бы от одной из переменных несущественно.

Вторым вопросом, решаемым при построении ПФ, является определение порядка расположения найденных подфрагментов требуемой номенклатуры в столбце значений ПФ.

Пусть имеется столбец значений некоторой функции, разделенный на $2^{]\log_2 r[}$ позиций длиной 2^{n-1}, в котором значения функций еще не проставлены. Разместим на некоторых позициях столбца в произвольном порядке r различных подфрагментов длиной 2^{n-1}, а остальные $2^{]\log_2 r[}$ - r позиций заполним произвольно.

Из сказанного выше следует, что сформированная таким образом функция является порождающей для заданного набора *N* функций при использовании операций настройки четырех типов, так как любая пара подфрагментов, номенклатура которых совпадает с номенклатурой подфрагментов заданных функций, может быть выбрана в любом сочетании и порядке, образуя остаточные функции, совпадающие с заданными.
Таким образом, при указанном способе формирования ПФ число настроечных переменных $d =]\log_2 r[$, а число информационных переменных $a = n - 1$. При этом общее число переменных ПФ

$$g'' =]\log_2 r[+ n - 1, \quad (3\text{-}23)$$

а число различных ПФ

$$v = (2^{]\log_2 r[}). \quad (3\text{-}24)$$

При аналитической форме представления построенная ПФ имеет вид

(3-25)

где

Эффективность метода. Для того чтобы ПФ, построенная при помощи изложенного метода, содержала меньшее число переменных по сравнению с их числом, получаемым при объединении в ПФ самих функций, необходимо выполнение неравенства

8″ < *'.   (3-26)

Подставляя в (3-26) значения g' и g'', получим

(3-27)

Решение этого неравенства имеет следующий вид:

MoBlJVI.   (3-28)

Полученная оценка для числа подфрагментов может быть в ряде случаев улучшена. Для этого определим, какое число различных подфрагментов длиной $2^n{\sim}^1$ содержится во всех булевых функциях *n* переменных. Значение этой величины

Гш.х = 2^2Я-1.   (3-29)

На основе объединения полученных соотношений можно утверждать, что число различных фрагментов *г* должно удовлетворять неравенству

r<min{2^1IogoN[; 2^2"-1}   (3-30)

Полученное соотношение является условием, при выполнении которого удается построить ПФ, зависящую от числа переменных, меньшего чем g'. В противном случае число переменных в ПФ равно g'.

Пример. При $n = 3$, $N = 40$ $г < 16$; при $n = 3$, $N = 4$ $г < 4$.

Минимизация числа переменных порождающей функции. Выше было показано, что при применении предлагаемого метода число переменных в ПФ g'' при n = const зависит только от числа различных подфрагментов *г* в *N* заданных функциях. Поэтому проблема минимизации числа переменных в ПФ сводится к сокращению числа различных подфрагментов. Можно утверждать, что число различных подфрагментов в функциях, зависящих от *n* переменных, в общем случае определяется двумя факторами: а) выбором множества *N* функций, для которых ищется ПФ; б) выбором переменных, по которым осуществляется разложение заданных функций.

Для заданных *N* функций число различных подфрагментов определяется только вторым фактором. Выбор переменных, обеспечивающих минимизацию числа подфрагментов, в общем случае связан с перебором.

Введем критерий, использование которого позволяет выбрать требуемые переменные без перебора. Этот критерий является эвристическим, однако его применение при построении модулей показало, что он достаточно эффективен. Будем использовать при выборе переменной, по которой следует проводить разложение, вес производной булевой функции по переменной [43].

Производной булевой функции, зависящей от n переменных, по переменной x_k *называется функция* n - 1 *переменных, равная сумме по модулю 2 остаточных исходной функции при* $x_k = 0$, *и* $x_k = 1$, *т.е.*

$$\frac{\delta f(x_1,...,x_n)}{\delta x_k} = f(x_k = 0) \oplus f(x_k = 1) \quad (3\text{-}31)$$

Весом производной $P\left(\frac{\delta f}{\delta x_k}\right)$ *называют число наборов, на которых она принимает значение, равное единице.*

Пусть некоторая функция f зависит от нескольких переменных, и в том числе от x_k, и x_i. Будем выполнять разложение по x_k раньше, чем по x_i, если справедливо неравенство

$$P\left(\frac{\delta f}{\delta x_k}\right) > P\left(\frac{\delta f}{\delta x_i}\right) \qquad (3\text{-}32)$$

Введенный критерий предложен для разложения функций, принадлежащих классу произвольных функций. Покажем, что выбор этого критерия является естественным. Пусть задана функция, принадлежащая классу пороговых. Произведем разложение этой функции по переменной, обладающей наибольшим весом [7]. Для этого класса функций выбор переменной, по которой должно выполняться разложение, по любому из двух указанных критериев приводит обычно к одному и тому же результату. Если же некоторая функция является симметричной по переменным x_k и x_i, то разложение может выполняться по любой из этих переменных.

Если задано несколько булевых функций, каждая из которых зависит от *n* переменных, то их первоначально необходимо переобозначить так, чтобы при разложении по какой-либо одной переменной общее число различных фрагментов было минимальным. Для достижения этого воспользуемся следующей процедурой: а) в каждой из заданных формул определяем веса производных по всем переменным; б) производим переобозначение переменных в каждой формуле так, что переменная с большим весом производной будет иметь и больший порядковый номер; в) раскладываем все заданные формулы по переменной с наибольшим порядковым номером.

Использование этой процедуры, как правило, позволяет уменьшить число различных подфрагментов в системе заданных функций по сравнению с разложением в соответствии с исходной нумерацией переменных, хотя это положение и не удается доказать.

Пример. Пусть заданы функции $f_1 = v_1v_2v_3$, $f_2 = v_1(v_2 \vee v_3)$, $f_3 = v_1v_3 \vee v_2$, $f_4 = v_1 \vee v_2 \vee v_3$. Определим нумерацию переменных, при котором число различных подфрагментов будет минимальным. Составим таблицу истинности для заданных функций (табл. 3-6).

Таблица 3-6

Таблица истинности функций $f_1 \div f_4$

v_3	v_2	v_1	f_1	f_2	f_3	f_4	v_3	v_2	v_1	f_1	f_2	f_3	f_4
0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0	0	0	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	1

Таким образом, заданный набор функций содержит шесть различных подфрагментов: $\varphi_1 = 0000$; $\varphi_2 = 0001$; $\varphi_3 = 0011$; $\varphi_4 = 0101$; $\varphi_5 = 0111$, $\varphi_6 = 1111$. Попытаемся уменьшить значение r путем переобозначения переменных. Функции f_1 и f_4 симметричны по всем переменным; в f_2 наибольшим весом производной обладает переменная v_1, а в f_3 — переменная v_2. Поэтому переобозначим переменные следующим образом: $f_1 = x_1x_2x_3$, $f_2 = (x_1 \vee x_2)x_3$, $f_3 = x_1x_2 \vee x_3$, $f_4 = x_1 \vee x_2 \vee x_3$.

Из таблицы истинности этих функций (табл. 3-7) следует, что число различных подфрагментов после переобозначения при разложении по переменной x_3 равно четырем: $\varphi_1 = 0000$; $\varphi_2 = 0001$; $\varphi_3 = 0111$; $\varphi_4 = 1111$.

Таблица 3-7

Минимизация числа подфрагментов

x_3	x_2	x_1	f_1	f_2	f_3	f_4	x_3	x_2	x_1	f_1	f_2	f_3	f_4
0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	0	1	1	1	0	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1	1	1	1

Таким образом, использование предложенной процедуры позволило уменьшить число различных подфрагментов и тем самым сократить число входов модуля.

Минимизация элементной сложности модуля. Эта задача сводится к задаче упрощения ПФ, зависящей от фиксированного числа переменных, и, в частности, к минимизации числа букв в ней. Так как ПФ при большом числе подфрагментов в объединяемых функциях зависит от сравнительно большого числа переменных, то нахождение формулы, минимальной по числу букв, для представления ПФ в аналитическом виде с помощью классического аппарата минимизации является чрезвычайно громоздким и трудоемким. При некоторых условиях, которые будут определены ниже, удается упростить процесс минимизации. Запишем ПФ модуля в следующем виде:

$$F_2(z_g,...,z_1)=\bigvee_{i=1}^{v} z_g^{\sigma_g}...z_{n+1}^{\sigma_{n+1}}\psi_i(z_n,...,z_1), \qquad (3\text{-}33)$$

где

$$v=2^{]\log_2 r[-1}, g=]\log_2 r[+n-1.$$

Такое представление ПФ может быть получено в результате объединения соседних пар подфрагментов. При этом

$$\psi_i(z_n,...,z_1)=z_n\varphi'(z_{n-1},...,z_1)\vee \bar{z}_n\varphi''(z_{n-1},...,z_1). \qquad (3\text{-}34)$$

Поскольку подфрагменты представляют собой функции и поэтому могут быть проминимизированы отдельно, направим усилия на минимизацию числа букв в той части конъюнкции в выражении (3-33), которая состоит из настроечных переменных. Для этого ПФ будем искать в форме, отличающейся от формы выражения (3-33) тем, что буквы с инверсиями, соответствующие настроечным переменным, в ней исключены. При этом указанные части конъюнкций соответствуют всевозможным сочетаниям из $]\log_2 r[-1$ переменных настроечного поля без инверсий, взятым по $0,1,2,...,]\log_2 r[-1$ переменным, а общее число конъюнкций в этой форме не изменяется по сравнению с их числом в выражении (3-33) и равно $2^{]\log_2 r[-1}$. Назовем эту форму ПФ минимизированной. Для того чтобы такая форма представления ПФ была эквивалентна F_2, фрагменты ψ_i должны удовлетворять системе соотношений, получаемой из условий равенства ПФ, представленных в этих формах. Сравнение этих выражений будем выполнять в табличной форме.

Для формализованного обозначения соотношений между фрагментами ψ_i и ψ_j введем в рассмотрение функцию импликации. В первой главе мы уже пользовались функцией, задаваемой таблицей истинности вида табл. 3-8, которая называется импликацией x_1 в x_2.

Говорят, что функция f_i имплицирует $f_j (f_i \to f_j)$, если столбец значений таблицы истинности f_j «покрывает» соответствующий столбец f_i;. При этом, если значение f_i на некотором наборе равно нулю, то и значение f_i на том же наборе также равно нулю; если значение f_j на некотором наборе равно единице, то значение f_i на

Таблица истинности функции импликации Таблица 3-8

x_1	x_2	f
0	0	1
0	1	1
1	0	0
1	1	1

этом наборе может быть произвольным. Поэтому, в частности, можно утверждать, что $v_1v_2 \to v_1\vee v_2$.

Отметим, что если при построении ПФ без учета ее элементной сложности подфрагменты могли размещаться в столбце значений произвольно, то при нахождении ПФ в минимизированной форме, системе импликаций которой должны удовлетворять фрагменты, накладываются ограничения на порядок их размещения, несмотря на возможность использования четырех разновидностей операций настройки, рассмотренных выше.

Пример. Пусть число подфрагментов, зависящих от трех переменных, в ПФ равно восьми. Тогда эта функция имеет вид

$$F_1(z_6,...,z_1)=\bar{z}_6\bar{z}_5\bar{z}_4\varphi_1\vee\bar{z}_6\bar{z}_5z_4\varphi_2\vee\bar{z}_6z_5\bar{z}_4\varphi_3\vee\bar{z}_6z_5z_4\varphi_4\vee z_6\bar{z}_5\bar{z}_4\varphi_5\vee z_6\bar{z}_5z_4\varphi_6\vee z_6z_5\bar{z}_4\varphi_7\vee z_6z_5z_4\varphi_8.$$

Приведем это выражение к виду (3-33):

$$F_2(z_6,...,z_1)=\bar{z}_6\bar{z}_5(\bar{z}_4\varphi_1\vee z_4\varphi_2)\vee\bar{z}_6z_5(\bar{z}_4\varphi_3\vee z_4\varphi_4)\vee z_6\bar{z}_5(\bar{z}_4\varphi_5\vee z_4\varphi_6)\vee z_6z_5(\bar{z}_4\varphi_7\vee z_4\varphi_8)=$$
$$=\bar{z}_6\bar{z}_5\psi_1\vee\bar{z}_6z_5\varphi_2\vee z_6\bar{z}_5\varphi_3\vee z_6z_5\varphi_4.(3-36)$$

Определим условия для ψ_i, при выполнении которых F_2 может быть представлена в минимизированной форме:

$$F_3(z_6,...,z_1)=\psi_1\vee z_5\psi_2\vee z_6\psi_3\vee z_5z_6\psi_4. \qquad (3\text{-}37)$$

Составим таблицы истинности для функции F_2 и F_3 и определим условия, выполнение которых необходимо для эквивалентности этих функций (табл. 3-9)

Таблица 3-9

Условия эквивалентности функции F_2 и F_3

z_6	z_5	F_2	F_3
0	0	ψ_1	ψ_1
0	1	ψ_2	$\psi_1 \vee \psi_2$
1	0	ψ_3	$\psi_1 \vee \psi_3$
1	1	ψ_4	$\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4$

При этом $F_3 = F_2$, если выполняется система соотношений вида:

$\psi_1 = \psi_1;(3-38)$

$\psi_1 \vee \psi_2 = \psi_2;(3-39)$

$\psi_1 \vee \psi_3 = \psi_3;(3-40)$

$\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4 = \psi_4;(3-41)$

Из соотношения (3-41) следует, что фрагмент ψ_1 должен быть построен в результате объединения подфрагментов ψ_i и ψ_j, так чтобы выполнялась система импликации

$\psi_1 \to \psi_4; \psi_2 \to \psi_4; \psi_3 \to \psi_4$, (3-42)

Т. е. столбец значений ψ_4 должен «покрывать» столбцы значений ψ_1, ψ_2, ψ_3.

Из соотношений (3-39) и (3-40) следует, что фрагмент ψ_1 должен удовлетворять системе импликаций

$$\psi_1 \to \psi_2; \psi_1 \to \psi_3, \qquad (3\text{-}43)$$

т. е. столбцы значений ф$_2$ и ф$_3$ должны «покрывать» столбец значений ψ_1. Таким образом, фрагмент ψ_1 должен быть выбран так, чтобы он «покрывался» всеми остальными фрагментами, а фрагмент ψ_4 сам должен «покрывать» все остальные фрагменты. Фрагменты ψ_2 и ψ_3 строятся так, чтобы соответствующие им функции были наиболее простыми. При этом, как следует из табл. 3-9, порядок расположения фрагментов должен быть следующим: ψ_1, ψ_2, ψ_3, ψ_4.

Следовательно, если фрагменты могут быть сформированы так, что соотношения (3-42) и (3-43) выполняются, то ПФ может быть представлена в минимизированном виде (3-37). После этого ПФ может быть проминимизирована еще больше с помощью классических методов. Если хотя бы одна из импликаций не выполняется, то ПФ записывается в виде (3-36), после чего она также минимизируется классическими методами.

Процедура построения порождающей функции. На основании результатов, изложенных выше, может быть предложена процедура построения ПФ модуля, реализующего путем настройки заданный набор из *N* булевых функций, каждая из которых существенно зависит от *n* переменных.

1. В каждой из *N* заданных функций определяем веса производных по всем переменным.
2. Производим переобозначение переменных в каждой функции так, что переменная с большим весом производной будет иметь и больший порядковый номер.
3. Раскладываем все функции по переменной с наибольшим порядковым номером.
4. Формируем список различных подфрагментов φ_i, содержащихся в *N* функциях набора.
5. Подсчитываем число различных подфрагментов φ_i. При этом число переменных в ПФ определяется из соотношения

$$g =]\log_2 r[+ n - 1. \qquad (3\text{-}44)$$

6. Если полученные подфрагменты ф, могут быть объединены таким образом, что выполняется система соотношений вида

$$\left.\begin{array}{l}\psi_1=\psi_1;\\ \psi_2=\psi_1\vee\psi_2;\\ \psi_3=\psi_1\vee\psi_3;\\ \dots\dots\dots\dots\dots\\ \psi_t=\psi_1\vee\psi_2\vee\ldots\vee\psi_t,\end{array}\right\}\qquad(3\text{-}45)$$

где

$\psi_i(z_n,...,z_1)=z_n\varphi'(z_{n-1},...,z_1)\vee z_n\varphi''(z_{n-1},...,z_1),$

то искомая ПФ может быть представлена в виде

$$F(z_g,...,z_1)=\varphi_1\vee z_{n+1}\psi_2\vee z_{n+2}\psi_3\vee\ldots\vee z_g\psi_j\vee z_{n+1}z_{n+2}\psi_k\vee\ldots\vee z_{n+1}z_{n+2}\ldots z_g\psi_t.\qquad(3\text{-}46)$$

Если найденные подфрагменты φ_i, не могут быть объединены так, чтобы выполнялось каждое из соотношений приведенной системы, то ПФ представляется в виде

$$F(z_g,...,z_1)=\bigvee_{i=1}^{t} z_g^{\sigma_g}\ldots z_n^{\sigma_n}\varphi_i(z_{n-1},...,z_1)=\bigvee_{i=1}^{t} z_g^{\sigma_g}\ldots z_{n+1}^{\sigma_{n+1}}\varphi_j(z_n,...,z_1),\qquad(3\text{-}47)$$

где

$$v=2^{]\log_2 r[};t=2^{]\log_2 r[-1}.$$

7. Осуществляем минимизацию найденной функции с помощью классических методов с целью уменьшения элементной сложности настраиваемого модуля.

8. Составляем таблицу истинности из переменных настроечного поля $z_g,...,z_n$, в столбце значений которой проставляем виденные подфрагменты φ_i, в порядке их расположения в ПФ.

9. Каждую из N заданных функций представляем путем объединения столбцов значений подфрагментов в виде

$$f_i=\varphi_j\cdot\varphi_k.\qquad(3\text{-}48)$$

10. Таблицу настроек искомого модуля для реализаций функций, однотипных с заданными, определяем на основе сопоставления каждого из выражений вида $f_i=\varphi_j\cdot\varphi_k$ и таблицы истинности настроечного поля, с помощью которой осуществляется выбор и объединение подфрагментов φ_j и φ_k.

3-3. ПРИМЕР ПОСТРОЕНИЯ ПОРОЖДАЮЩЕЙ ФУНКЦИИ НАСТРАИВАЕМОГО МОДУЛЯ

Воспользуемся предложенной процедурой для нахождения структуры модуля, универсального в классе формул в базисе $\{\&,\vee,\bar{\ }\}$ из пяти букв.

Предполагая, что прямые и инверсные входы *ИИ* равнодоступны и возможна перестановка входных переменных, для построения ПФ модуля воспользуемся *PN*-классификацией. При этом в ПФ достаточно объединить следующие 24 представителя *PN-типов* бесповторных формул в базисе $\{\&,\vee,\bar{\ }\}$ из пяти букв:

1. $f_1 = x_1x_2x_3x_4x_5$;

2. $f_2 = (x_1 \vee x_2)x_3x_4x_5$;

3. $f_3 = (x_1x_2 \vee x_3)x_4x_5$;

4. $f_4 = (x_1 \vee x_2 \vee x_3)x_4x_5$;

5. $f_5 = (x_1x_2x_3 \vee x_4)x_5$;

6. $f_6 = [(x_1 \vee x_2)x_3 \vee x_4]x_5$;

7. $f_7 = (x_1x_2 \vee x_3 \vee x_4)x_5$;

8. $f_8 = (x_1 \vee x_2 \vee x_3 \vee x_4)x_5$;

9. $f_9 = x_1x_2x_3x_4 \vee x_5$;

10. $f_{10} = (x_1 \vee x_2)x_3x_4 \vee x_5$;

11. $f_{11} = (x_1x_2 \vee x_3)x_4 \vee x_5$;

12. $f_{12} = (x_1 \vee x_2 \vee x_3)x_4 \vee x_5$;

13. $f_{13} = x_1x_2x_3 \vee x_4 \vee x_5$;

14. $f_{14} = (x_1 \vee x_2)x_3 \vee x_4 \vee x_5$;

15. $f_{15} = x_1x_2 \vee x_3 \vee x_4 \vee x_5$;

16. $f_{16} = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$;

17. $f_{17} = x_1x_2 \vee x_3x_4 \vee x_5$;

18. $f_{18} = (x_1 \vee x_2)(x_3 \vee x_4) \vee x_5$;

19. $f_{19} = (x_1x_2 \vee x_3x_4)x_5$;

20. $f_{20} = (x_1 \vee x_2)(x_3 \vee x_4)x_5$;

21. $f_{21} = x_1x_2x_3 \vee x_4x_5$;

22. $f_{22} = (x_1 \vee x_2)x_3 \vee x_4x_5$;

23. $f_{23} = (x_1x_2 \vee x_3)(x_4x_5)$;

24. $f_{24} = (x_1 \vee x_2 \vee x_3)(x_4 \vee x_5)$.

Нумерация переменных в этих формулах выбрана так, что переменные с большим весом булевой производной имеют больший порядковый номер. Выбранное правило нумерации обеспечивает сокращение числа различных подфрагментов для заданного набора формул и в конечном итоге приводит к уменьшению числа внешних выводов в модуле.

Для нахождения различных подфрагментов составим таблицу истинности них формул (табл. 3-10). Заданный набор формул содержит 16 различных подфрагментов длиной 16 (табл, 3-11).

Таблица 3-10

x_5	x_4	x_3	x_2	x_1	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1
0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1
1	0	1	0	1	0	0	0	0	0	1	1	1	1	1	1
1	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1
1	0	1	1	1	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0	0	1	0	0
0	0	1	1	1	0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	0	0	1	1	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	0	0	0	0	0	1
1	1	1	1	1	0	1	0	0	0	0	0	1
1	1	1	1	1	1	1	0	0	0	0	1	1
1	1	1	1	1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	1	0	0	0	1	1	1
1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	0	0	1	1
1	1	1	1	1	1	1	0	0	0	0	1	1
1	1	1	1	1	1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

В аналитической форме подфрагменты имеют следующий вид:

$1.\varphi_1 = 0;$	$9.\varphi_9 = (x_1x_2 \vee x_3)x_4;$
$2.\varphi_2 = x_1x_2x_3x_4x_5;$	$10.\varphi_{10} = x_1x_2 \vee x_3;$
$3.\varphi_3 = x_1x_2x_3;$	$11.\varphi_{11} = x_1x_2 \vee x_3x_4;$
$4.\varphi_4 = x_1x_2x_3 \vee x_4;$	$12.\varphi_{12} = x_1x_2 \vee x_3 \vee x_4;$
$5.\varphi_5 = (x_1 \vee x_2)x_3x_4;$	$13.\varphi_{13} = (x_1 \vee x_2 \vee x_3)x_4;$
$6.\varphi_6 = (x_1 \vee x_2)x_3;$	$14.\varphi_{14} = x_1 \vee x_2 \vee x_3;$
$7.\varphi_7 = (x_1 \vee x_2)(x_3 \vee x_4);$	$15.\varphi_{15} = x_1 \vee x_2 \vee x_3 \vee x_4;$
$8.\varphi_8 = (x_1 \vee x_2)x_3 \vee x_4;$	$16.\varphi_{16} = 1;$

Отметим, что эти подфрагменты представляют собой все *PN*-типы формул, бесповторных в базисе $\{\&, \vee, \bar{\ }\}$ из 4,3 и 0 букв.

Так как число различных подфрагментов в этом случае $r = 16$, то общее число переменных, от которого зависит ПФ:

$$g =]\log_2 r[+ n - 1 = 8.$$

При этом ПФ может быть представлена в виде

$$F(z_8,...,z_1)=\bar z_8\bar z_7\bar z_6\bar z_5\varphi_\alpha\vee\bar z_8\bar z_7\bar z_6 z_5\varphi_\beta\vee\bar z_8\bar z_7 z_6\bar z_5\varphi_j\vee\bar z_8\bar z_7 z_6 z_5\varphi_u\vee$$
$$\vee\bar z_8 z_7\bar z_6\bar z_5\varphi_\omega\vee\bar z_8 z_7\bar z_6 z_5\varphi_\nu\vee\bar z_8 z_7 z_6\bar z_5\varphi_p\vee\bar z_8 z_7 z_6 z_5\varphi_e\vee z_8\bar z_7\bar z_6\bar z_5\varphi_t\vee$$
$$\vee z_8\bar z_7\bar z_6 z_5\varphi_h\vee z_8\bar z_7 z_6\bar z_5\varphi_k\vee z_8\bar z_7 z_6 z_5\varphi_n\vee z_8 z_7\bar z_6\bar z_5\varphi_r\vee z_8 z_7\bar z_6 z_5\varphi_\lambda\vee$$
$$\vee z_8 z_7 z_6\bar z_5\varphi_h\vee z_8 z_7 z_6 z_5\varphi_\kappa=\bar z_8\bar z_7\bar z_6(\bar z_5\varphi_\alpha z\vee z_5\varphi_\beta)\vee\bar z_8\bar z_7 z_6(\bar z_5\varphi_j\vee z_5\varphi_u)\vee$$
$$\vee\bar z_8 z_7\bar z_6(\bar z_5\varphi_\omega\vee z_5\varphi_\nu)\vee\bar z_8 z_7 z_6(z_5\varphi_p\vee z_5\varphi_e)\vee z_8\bar z_7\bar z_6(\bar z_5\varphi_t\vee z_5\varphi_s)\vee z_8\bar z_7 z_6(\bar z_5\varphi_k\vee z_5\varphi_n)\vee z_8 z_7\bar z_6(z_5\varphi_r\vee z_5\varphi_\lambda)\vee$$
$$\vee z_8 z_7 z_6(\bar z_5\varphi_h\vee z_5\varphi_\kappa).(3-49)$$

Таким образом, минимизированной формой ПФ является формула вида

$$F(z_8,...,z_1)=\bar z_8\bar z_7\bar z_6\varphi_1\vee\bar z_8\bar z_7 z_6\varphi_2\vee\bar z_8 z_7 z_6\varphi_4\vee z_8\bar z_7\bar z_6\varphi_5\vee$$
$$\vee z_8\bar z_7 z_6\varphi_6\vee z_8 z_7\bar z_6\varphi_7\vee z_8 z_7 z_6\varphi_8. \qquad (3\text{-}50)$$

Определим соотношение для ψ_1, при выполнении которых F может быть представлена в виде

$$F'(z_8,...,z_1)=\psi_1\vee z_6\psi_2\vee z_7\psi_3\vee z_7 z_6\psi_4\vee z_8\psi_5\vee z_8 z_6\psi_6\vee z_8 z_7\psi_7\vee z_8 z_7 z_6\psi_8. \qquad (3\text{-}51)$$

Составим таблицы истинности для функции F и F' и определим условия, при которых эти функции эквивалентны (табл. 3-12).

Таблица 3-11

Таблица истинности подфрагментов

x	x_3	x_2	x_1	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}	φ_{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1
0	1	0	1	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1
0	1	1	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1
0	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1
1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1
1	0	0	1	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	1
1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Таблица 3-12

Условия минимизации ПФ

z_8	z_7	z_6	F	F'
0	0	0	ψ_1	ψ_1
0	0	1	ψ_2	$\psi_1\vee\psi_2$
0	1	0	ψ_3	$\psi_1\vee\psi_3$
0	1	1	ψ_4	$\psi_1\vee\psi_2\vee\psi_3\vee\psi_4$
1	0	0	ψ_5	$\psi_1\vee\psi_5$
1	0	1	ψ_6	$\psi_1\vee\psi_2\vee\psi_5\vee\psi_6$
1	1	0	ψ_7	$\psi_1\vee\psi_3\vee\psi_5\vee\psi_7$
1	1	1	ψ_8	$\psi_1\vee\psi_2\vee\psi_3\vee\psi_4\vee\psi_5\vee\psi_6\vee\psi_7\vee\psi_8$

При этом $F'=F$, если имеет место система соотношений

$\psi_1 = \psi_1;$ $(3-52)$

$\psi_1 \vee \psi_2 = \psi_2;$ $(3-53)$

$\psi_1 \vee \psi_3 = \psi_3;$ $(3-54)$

$\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4 = \psi_4;$ $(3-55)$

$\psi_1 \vee \psi_5 = \psi_5;$ $(3-56)$

$\psi_1 \vee \psi_2 \vee \psi_5 \vee \psi_6 = \psi_6;$ $(3-57)$

$\psi_1 \vee \psi_3 \vee \psi_5 \vee \psi_7 = \psi_7;$ $(3-58)$

$\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4 \vee \psi_5 \vee \psi_6 \vee \psi_7 \vee \psi_8 = \psi_8;$ $(3-59)$

Из этих соотношений следует, что

$\psi \to \psi_2; \psi_1 \to \psi_3; \psi_1 \to \psi_4; \psi_1 \to \psi_5; \psi_1 \to \psi_6; \psi_1 \to \psi_7; \psi_1 \to \psi_8,\ (3-60)$

т.е столбец значений ψ_1 должен «показываться» любым из оставшихся семи фрагментов ψ_i.

Пусть

$$\psi_1 = \varphi_1 \cdot \varphi_2 = z_1 z_2 z_3 z_4 z_5. \quad (3-61)$$

Соотношения (3-53)-(3-55) порождают импликацию

$$\psi_2 \vee \psi_3 \to \psi_4. \quad \text{(3-62)}$$

Соотношения (3-53),(3-56),(3-57)-импликацию

$$\psi_2 \vee \psi_5 \to \psi_6, \quad \text{(3-63)}$$

а соотношения (3-54), (3-56), (3-58)-импликацию

$$\psi_3 \vee \psi_5 \to \psi_7. \quad \text{(3-64)}$$

Из соотношения (3-59) следует, что

$$\psi_1 \to \psi_8; \psi_2 \to \psi_8; \psi_3 \to \psi_8; \psi_4 \to \psi_8; \psi_5 \to \psi_8; \psi_6 \to \psi_8; \psi_7 \to \psi_8; \quad \text{(3-65)}$$

т.е. столбец значений ψ_8 должен «покрывать» любой из оставшихся семи фрагментов ψ_i.

Пусть

$$\psi_8 = \varphi_{15} \cdot \varphi_{16} = z_1 \vee z_2 \vee z_3 \vee z_4 \vee z_5. \quad \text{(3-66)}$$

Если выбрать (табл. 3-13)

$$\left.\begin{aligned} \psi_2 &= \varphi_3 \cdot \varphi_4 = z_1 z_2 z_3 \vee z_4 z_5; \\ \psi_3 &= \varphi_5 \cdot \varphi_6 = (z_1 \vee z_2) z_3 (z_4 \vee z_5); \\ \psi_4 &= \varphi_7 \cdot \varphi_8 = (z_1 \vee z_2)(z_3 \vee z_4) \vee z_4 z_5; \end{aligned}\right\} (3-67)$$

то будет выполняться импликация (3-62).

При выборе

$$\left.\begin{aligned} \psi_5 &= \varphi_9 \cdot \varphi_{10} = (z_1 z_2 \vee z_3)(z_4 \vee z_5); \\ \psi_6 &= \varphi_{11} \cdot \varphi_{12} = z_1 z_2 \vee z_3 z_4 \vee (z_3 \vee z_4) z_5; \end{aligned}\right\} (3-68)$$

удовлетворяется импликации (3-63).

И, наконец, выбрав

$$\psi_7 = \varphi_{13} \cdot \varphi_{14} = (z_1 \vee z_2 \vee z_3)(z_4 \vee z_5), (3-69)$$

закончим формирование фрагментов (табл. 3-13).

При этом ПФ имеет вид

$$\begin{aligned} F(z_8, ..., z_1) &= z_1 z_2 z_3 z_4 z_5 \vee (z_1 z_2 z_3 \vee z_4 z_5) z_6 \vee (z_1 \vee z_2) z_3 (z_4 \vee z_5) z_7 \vee \\ &\vee (z_1 z_2 \vee z_3)(z_4 \vee z_5) z_8 \vee (z_1 \vee z_2)(z_3 \vee z_4) z_6 z_7 \vee z_1 z_2 z_6 z_8 \vee \\ &\vee (z_1 \vee z_2)(z_3 \vee z_4) z_7 z_8 \vee (z_1 \vee z_2 \vee z_3 \vee z_4 \vee z_5) z_6 z_7 z_8. (3-71) \end{aligned}$$

Таблица 3-13

Таблица объединения фрагментов

z_5	z_4	z_3	z_2	z_1	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7	ψ_8
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	0	1	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	0	0	1	0	0	0	1
0	0	1	1	0	0	0	0	1	0	0	0	1
0	0	1	1	1	0	1	0	1	0	1	0	1
0	1	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	0	1	0	0	1	1
0	1	0	1	0	0	0	0	1	0	0	1	1
0	1	0	1	1	0	0	0	1	1	1	1	1
0	1	1	0	0	0	0	0	0	1	1	1	1
0	1	1	0	1	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0	0	0	0	1	1
1	0	0	1	1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	0	0	1	1	1	1
1	0	1	0	1	0	0	1	1	1	1	1	1
1	0	1	1	0	0	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0	0	1	0	1
1	1	0	0	1	0	1	0	0	0	1	1	1
1	1	0	1	0	0	1	0	0	0	1	1	1
1	1	0	1	1	0	1	0	0	1	1	1	1
1	1	1	0	0	0	1	0	0	1	1	0	1
1	1	1	0	1	0	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Таблица 3-14

Таблица истинности переменных настроечного поля

z_8	z_7	z_6	z_5	F
0	0	0	0	φ_1
0	0	0	1	φ_2
0	0	1	0	φ_3
0	0	1	1	φ_4
0	1	0	0	φ_5
0	1	0	1	φ_6
0	1	1	0	φ_7
0	1	1	1	φ_8
1	0	0	0	φ_9
1	0	0	1	φ_{10}
1	0	1	0	φ_{11}
1	0	1	1	φ_{12}
1	1	0	0	φ_{13}
1	1	0	1	φ_{14}
1	1	1	0	φ_{15}
1	1	1	1	φ_{16}

Так как полученная формула достаточно сложна (она содержит 47 букв), продолжим процесс ее минимизации:

$$F(z_8,...,z_1) = z_1z_2(z_3z_4z_5 \vee z_6z_8) \vee \{z_1z_2z_3 \vee z_4z_5 \vee [(z_1 \vee z_2)(z_3 \vee z_4 \vee z_8)(z_3 \vee z_4 \vee z_5)z_8]z_7\}z_6 \vee$$
$$\vee[(z_1 \vee z_2)(z_3 \vee z_8)z_7 \vee (z_1z_2 \vee z_3)z_8](z_4 \vee z_5).(3-72)$$

Эта формула содержит 34 буквы.

Составим таблицу истинности для переменных настроечного поля ПФ с целью определения настроек, порождающих заданные функции (табл. 3-14).

Таблица 3-15

Таблица настроек для К=5

Настройка	Объединяемые фрагменты	Реализуемая формула
$z_8=0, z_7=0, z_6=0$	$\varphi_1 \cdot \varphi_2$	$F_1 = z_1z_2z_3z_4z_5$
$z_8=0, z_6=0, z_5=0$	$\varphi_1 \cdot \varphi_5$	$F_2 = (z_1 \vee z_2)z_3z_4z_7$
$z_7=0, z_6=0, z_5=0$	$\varphi_1 \cdot \varphi_9$	$F_3 = (z_1z_2 \vee z_3)z_4z_8$
$z_8=z_7, z_6=0, z_5=0$	$\varphi_1 \cdot \varphi_{13}$	$F_4 = (z_1 \vee z_2 \vee z_3)z_4z_7$
$z_8=0, z_7=0, z_6=z_5$	$\varphi_1 \cdot \varphi_4$	$F_5 = (z_1z_2z_3 \vee z_4)z_5$
$z_8=0, z_7=z_6=z_5$	$\varphi_1 \cdot \varphi_8$	$F_6 = [(z_1 \vee z_2)z_3 \vee z_4]z_5$
$z_8=z_6=z_5, z_7=0$	$\varphi_1 \cdot \varphi_{12}$	$F_7 = (z_1z_2 \vee z_3 \vee z_4)z_5$
$z_8=z_7=z_6, z_5=0$	$\varphi_1 \cdot \varphi_{15}$	$F_8 = (z_1 \vee z_2 \vee z_3 \vee z_4)z_6$
$z_8=z_7=z_6, z_5=1$	$\varphi_2 \cdot \varphi_{16}$	$F_9 = z_1z_2z_3z_4 \vee z_6$
$z_8=z_6=z_5, z_7=1$	$\varphi_5 \cdot \varphi_{16}$	$F_{10} = (z_1 \vee z_2)z_3z_4 \vee z_5$
$z_8=1, z_7=z_6=z_5$	$\varphi_9 \cdot \varphi_{16}$	$F_{11} = (z_1z_2 \vee z_3)z_4 \vee z_5$
$z_8=1, z_7=1, z_6=z_5$	$\varphi_{13} \cdot \varphi_{16}$	$F_{12} = (z_1 \vee z_2 \vee z_3)z_4 \vee z_5$
$z_8=z_7, z_6=1, z_5=1$	$\varphi_4 \cdot \varphi_{16}$	$F_{13} = z_1z_2z_3 \vee z_4 \vee z_7$
$z_7=1, z_6=1, z_5=1$	$\varphi_8 \cdot \varphi_{16}$	$F_{14} = (z_1 \vee z_2)z_3 \vee z_4 \vee z_8$
$z_8=1, z_6=1, z_5=1$	$\varphi_{12} \cdot \varphi_{16}$	$F_{15} = z_1z_2 \vee z_3 \vee z_4 \vee z_7$
$z_8=1, z_7=1, z_6=1$	$\varphi_{15} \cdot \varphi_{16}$	$F_{16} = z_1 \vee z_2 \vee z_3 \vee z_4 \vee z_5$
$z_8=1, z_7=z_6, z_6=1$	$\varphi_{11} \cdot \varphi_{16}$	$F_{17} = z_1z_2 \vee z_3z_4 \vee z_5$
$z_8=z_5, z_7=1, z_6=1$	$\varphi_7 \cdot \varphi_{16}$	$F_{18} = (z_1 \vee z_2)(z_3 \vee z_4) \vee z_5$
$z_8=z_6, z_7=0, z_5=0$	$\varphi_1 \cdot \varphi_{11}$	$F_{19} = (z_1z_2 \vee z_3z_4)z_6$
$z_8=0, z_7=z_6, z_5=0$	$\varphi_1 \cdot \varphi_7$	$F_{20} = (z_1 \vee z_2)(z_3 \vee z_4)z_6$
$z_8=0, z_7=z_6, z_5=0$	$\varphi_3 \cdot \varphi_4$	$F_{21} = z_1z_2z_3 \vee z_4z_5$
$z_8=0, z_7=0, z_6=1$	$\varphi_6 \cdot \varphi_8$	$F_{22} = (z_1 \vee z_2)z_3 \vee z_4z_6$
$z_8=0, z_7=1, z_5=1$	$\varphi_9 \cdot \varphi_{10}$	$F_{23} = (z_1z_2 \vee z_3)(z_4 \vee z_5)$
$z_8=1, z_7=0, z_6=0$	$\varphi_{13} \cdot \varphi_{14}$	$F_{24} = (z_1 \vee z_2 \vee z_3)(z_4 \vee z_5)$
$z_8=1, z_7=0, z_6=0$		

Заданные формулы из пяти букв могут быть образованы в результате объединения следующих подфрагментов:

$f_1 = \varphi_1 \cdot \varphi_2$; $f_7 = \varphi_1 \cdot \varphi_{12}$; $f_{13} = \varphi_4 \cdot \varphi_{16}$; $f_{19} = \varphi_1 \cdot \varphi_{11}$;

$f_2 = \varphi_1 \cdot \varphi_5$; $f_8 = \varphi_1 \cdot \varphi_{15}$; $f_{14} = \varphi_8 \cdot \varphi_{16}$; $f_{20} = \varphi_1 \cdot \varphi_7$;

$f_3 = \varphi_1 \cdot \varphi_9$; $f_9 = \varphi_2 \cdot \varphi_{16}$; $f_{15} = \varphi_{12} \cdot \varphi_{16}$; $f_{21} = \varphi_3 \cdot \varphi_4$;

$f_4 = \varphi_1 \cdot \varphi_{13}$; $f_{10} = \varphi_5 \cdot \varphi_{16}$; $f_{16} = \varphi_{15} \cdot \varphi_{16}$; $f_{22} = \varphi_6 \cdot \varphi_8$;

$f_5 = \varphi_1 \cdot \varphi_4$; $f_{11} = \varphi_9 \cdot \varphi_{16}$; $f_{17} = \varphi_{11} \cdot \varphi_{16}$; $f_{23} = \varphi_9 \cdot \varphi_{10}$;

$f_6 = \varphi_1 \cdot \varphi_8$; $f_{12} = \varphi_{13} \cdot \varphi_{16}$; $f_{18} = \varphi_7 \cdot \varphi_{16}$; $f_{24} = \varphi_{13} \cdot \varphi_{14}$;

Поэтому таблица настроек (табл. 3-15) искомого модуля для реализаций функций, однотипных с заданными, определяется на основе сопоставления каждого выражения вида $f_i = \varphi_j \cdot \varphi_k$ и таблицы истинности настроечного поля, с помощью которой осуществляется выбор и объединение подфрагментов φ_j и φ_k.

Таблица 3-16

Таблица настроек модуля для К=4

Настройка	Реализуемая формула
$z_8=0, z_7=0, z_6=0, z_5=1$	$F_1 = z_1 z_2 z_3 z_4$
$z_8=0, z_7=1, z_6=0, z_5=0$	$F_2 = (z_1 \vee z_2) z_3 z_4$
$z_8=1, z_7=0, z_6=0, z_5=0$	$F_3 = (z_1 z_2 \vee z_3) z_4$
$z_8=1, z_7=0, z_6=0, z_5=0$	$F_4 = (z_1 \vee z_2 \vee z_3) z_4$
$z_8=0, z_7=0, z_6=1, z_5=1$	$F_5 = z_1 z_2 z_3 \vee z_4$
$z_8=0, z_7=1, z_6=1, z_5=1$	$F_6 = (z_1 \vee z_2) z_3 \vee z_4$
$z_8=1, z_7=0, z_6=1, z_5=1$	$F_7 = z_1 z_2 \vee z_3 \vee z_4$
$z_8=1, z_7=1, z_6=1, z_5=0$	$F_8 = z_1 \vee z_2 \vee z_3 \vee z_4$
$z_8=1, z_7=0, z_6=1, z_5=0$	$F_9 = z_1 z_2 \vee z_3 z_4$
$z_8=0, z_7=1, z_6=1, z_5=0$	$F_{10} = (z_1 \vee z_2)(z_3 \vee z_4)$

Модули, разрабатываемые с помощью изложенного в § 3-2 метода, обладают характерным свойством, состоящим в том, что если модуль универсален в классе формул в некотором базисе из *К* букв, то он также универсален в том же классе формул из *К-1* букв при условии, что его информационные и настроечные входы независимы.

Таким образом, модуль универсален при выполнении указанного условия для *К=4* (табл. 3-16).

Порождающая функция модуля, универсального в классе формул в базисе $\{\&, \vee, \bar{\ }\}$ из трех букв, найденная с помощью той же процедуры, имеет вид

$$F(z_4, z_3, z_2, z_1) = (z_1 \vee z_2 \vee z_3) z_4 \vee z_1 z_2 z_3.$$

Приведем таблицу настроек этого модуля (табл. 3-17).

Настройка	Реализуемая формула
$z_4 = 0$	$F_1 = z_1 z_2 z_3$
$z_4 = 1$	$F_2 = z_1 \vee z_2 \vee z_3$
$z_3 = 0$	$F_3 = (z_1 \vee z_2) z_4$
$z_3 = 1$	$F_4 = z_1 z_2 \vee z_4$

3-4. Техническая реализация настраиваемых логических модулей

В состав некоторых серий интегральных микросхем, выпускаемых отечественной промышленностью в настоящее время, входят НЛМ, принципы построения которых близки принципам, излагаемым в настоящей работе. Рассмотрим функциональные и электрические характеристики этих модулей.

В состав серии 108 [51] входит НЛМ среднего уровня интеграции, реализованный на МОП-транзисторах в виде микросхемы, оформленной в прямоугольном стеклянном корпусе 401.14-1. Эта микросхема имеет следующие электрические характеристики:

Напряжение источника питания, В………………………………………. $-27 \pm 10\%$

Потребляемая мощность, мВт, не более…………………………………100

Потребляемый ток, мА, не более...................................……………... 2,6

Амплитуда импульсов входного напряжения, В, не менее……………. –9,0

Напряжение логической единицы (выходное), В, не менее…………… –9,5

Напряжение логического нуля (выходное), В, не более……………..… –0,7

Частота следования импульсов входного напряжения, кГц, не более….100

Таблица 3-18

Бесповторные формулы из восьми букв, реализуемые модулем К1ЖЛ081 путем настройки

Способ настройки	F_1	F_2
$x_9 = 0, x_{10} = 0$	$F_1^1 = x_1x_2x_3x_4x_5x_6x_7\bar{x}_8$	$F_1^1 = x_1x_2x_3x_4x_5x_6x_7\bar{x}_8$
$x_9 = 0, x_{10} = 1$	$F_1^2 = x_1x_2x_3x_4 \vee x_5x_6x_7x_8$	$F_1^2 = x_1x_2x_3x_4 \vee x_5x_6x_7x_8$
$x_9 = 1, x_{10} = 0$	$F_1^3 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7 \vee x_8$	$F_1^3 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7 \vee x_8$
$x_9 = 1, x_{10} = 1$	$F_1^4 = x_1x_2x_3 \vee x_4x_5x_6 \vee x_7\bar{x}_8$	$F_1^4 = x_1x_2x_3 \vee x_4x_5x_6 \vee x_7\bar{x}_8$
$x_9 = 1, x_7 = 1$	$F_1^5 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_8 \vee x_{10}$	$F_1^5 = x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_8 \vee x_{10}$
$x_{10} = 1, x_7 = 1$	$F_1^6 = x_1x_2x_3x_4 \vee x_5x_6x_8 \vee x_9$	$F_1^6 = x_1x_2x_3x_4 \vee x_5x_6x_8 \vee x_9$
$x_9 = x_{10}, x_8 = 0$	$F_1^7 = x_1x_2x_3x_4x_5x_6x_7 \vee x_9$	$F_1^7 = x_1x_2x_3x_4x_5x_6x_7 \vee x_9$
$x_0 = 0, x_7 = x_8$	$F_1^8 = (x_1x_2x_3x_4 \vee x_5x_6x_7)x_{10}$	$F_1^8 = (x_1x_2x_3x_4 \vee x_5x_6x_7)x_{10}$
$x_9 = 0, x_4 = x_8$	$F_1^9 = (x_1x_2x_3 \vee x_5x_6x_7)x_4x_{10}$	$F_1^9 = (x_1x_2x_3 \vee x_5x_6x_7)x_4x_{10}$
$x_{10} = 0, x_7 = x_8$	$F_1^{10} = (x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7)x_9$	$F_1^{10} = (x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7)x_9$
$x_9 = x_{10}, x_7 = 0$	$F_1^{11} = (x_1x_2x_3 \vee x_4x_5x_6 \vee \bar{x}_8)x_9$	$F_1^{11} = (x_1x_2x_3 \vee x_4x_5x_6 \vee \bar{x}_8)x_9$
$x_9 = x_{10}, x_4 = x_8$	$F_1^{12} = (x_1x_2x_3 \vee x_5x_6 \vee x_7 \vee x_8)x_9$	$F_1^{12} = (x_1x_2x_3 \vee x_5x_6 \vee x_7 \vee x_8)x_9$
$x_{10} = 0, x_6 = x_8$	$F_1^{13} = [x_1x_2 \vee x_3x_4 \vee x_6(x_5 \vee x_7)]x_9$	$F_1^{13} = [x_1x_2 \vee x_3x_4 \vee x_6(x_5 \vee x_7)]x_9$

Модуль имеет десять логических входов, два выхода и реализует функции вида

$$F_1(x_1,...,x_{10}) = x_1x_2x_3x_4x_5x_6x_7\bar{x}_8\bar{x}_9\bar{x}_{10} \vee (x_1x_2 \vee x_3x_4 \vee x_5x_6 \vee x_7x_8)x_9\bar{x}_{10} \vee$$
$$\vee(x_1x_2x_3x_4 \vee x_5x_6x_7x_8)\bar{x}_9x_{10} \vee (x_1x_2x_3 \vee x_4x_5x_6 \vee x_7 \vee \bar{x}_8)x_9x_{10}; (3-73)$$
$$F_2(x_1,...,x_{10}) = \overline{F_1(x_1,...,x_{10})}.$$

Для определения функциональных возможностей модуля в классе формул, бесповторных в базисе $\{\&, \vee, \bar{\ }\}$, приравняем переменные друг другу и константам. Получившиеся при этом бесповторные формулы запишем в таблицу, формируя тем самым таблицу настроек модуля.
В табл. 3-18 приведены все бесповторные формулы максимальной длины (восемь букв), реализуемые модулем путем настроек, а в работе [6] указаны

Таблица 3-19

Бесповторные формулы из четырех букв, реализуемые Модулем К1ЖЛ014 путем настройки

Управляющие входы			F	Арифметический полином
P	0	R		
0	0	0	$F_1 = A_1A_2 \vee A_3A_4$	$2+2$
0	0	1	$F_2 = A_1A_2A_3A_4$	4
0	1	0	$F_3 = A_1A_2 \vee A_3 \vee A_4$	$2+1+1$
0	1	1	$F_4 = A_1A_2(A_3 \vee A_4)$	$2(1+1)$
1	0	0	$F_5 = A_1 \vee A_2 \vee A_3A_4$	$1+1+2$
1	0	1	$F_6 = (A_1 \vee A_2)A_3A_4$	$(1+1)2$
1	1	0	$F_7 = A_1 \vee A_2 \vee A_3 \vee A_4$	$1+1+1+1$
1	1	1	$F_8 = (A_1 \vee A_2)(A_3 \vee A_4)$	$(1+1)(1+1)$

Все бесповторные формулы меньшей длины, получаемые из формул, приведенных в табл. 3-18.
Анализ функциональных возможностей этого модуля, выполненный в § 5-1, позволил установить, что при использовании *PN*-классификации он универсален в классе произвольных формул в базисе $\{\&, \vee, \bar{\ }\}$ из пяти букв и в классе бесповторных ДНФ из шести букв.
Другой настраиваемый логический модуль входит в состав серии К501.

Микросхема К1ЖЛ014 состоит из трех одинаковых НЛМ с прямым и инверсным выходами, настраиваемыми общими входами P, θ, R [55].

Порождающая функция, описывающая поведение каждого из трех модулей на прямом выходе, имеет вид

$$F = \overline{\overline{[(A_1 \vee A_2)P \vee A_1A_2} \vee \overline{[(A_3 \vee A_4)\theta \vee A_3A_4}]R \vee \overline{[(A_1 \vee A_2)P \vee A_1A_2][(A_3 \vee A_4)\theta \vee A_3A_4}}].(3-74)$$

Этот модуль позволяет реализовать путем настройки шесть представителей из десяти существующих типов бесповторных формул в базисе $\{\&, \vee, \bar{\ }\}$ из четырех букв (табл. 3-19).

Как следует из приведенных результатов, этот модуль не универсален в классе формул в базисе $\{\&, \vee, \bar{\ }\}$ из четырех букв, так как не позволяет реализовать путем настройки представителей остальных четырех типов формул: $3+1; (1+1)1+1; (1+1+1)1; (2+1)1$. Однако этот модуль универсален в рассматриваемом классе формул из трех и менее букв.

Микросхема К1ЖЛ014 реализована на МОП-структурах и имеет следующие основные электрические параметры:

Выходные напряжения, В

логической единицы………. $-9,5$

логического нуля…………... $-1,0$

Входное напряжение, В

логические единицы………. $-8,5$

логического нуля………..… $-2,0$

Напряжение питания, В…... $-12,6 \pm 10\%$

Напряжение смещения, В… $-27 \pm 10\%$

Настраиваемые модули в настоящее время реализуется не только средствами микроэлектронной интегральной технологии. Так, в состав универсальной системы элементов промышленной пневмоавтоматики (УСЭППА) [11] входит одновыходное трехмембранное реле Р-3Ф, структура которого описывается функцией четырех переменных:

$$y = x_1(x_1 \vee \bar{x}_3) \vee \bar{x}_2 x_3 x_4 .(3-75)$$

Исследование функциональных возможностей этого модуля показано, что он реализует путем настройки большое число формул, и в том числе следующие формулы из трех букв:

при $x_1 = 0$ $\quad y_1 = \bar{x}_2 x_3 x_4 \equiv 3;$

при $x_1 = 1$ $\quad y_2 = x_2 \vee \bar{x}_3 \vee x_4 \equiv 1+1+1;$

при $x_3 = 1$ $\quad y_3 = x_1(x_2 \vee x_4) \equiv 1(1+1);$

при $x_4 = 0, x_4 = x_2$ $\quad y'_3 = x_1(x_2 \vee \bar{x}_3) \equiv 1(1+1);$

при $x_2 = x_1$ $\quad y_4 = x_1 \vee x_3 x_4 \equiv 1+2;$

при $x_4 = 1, x_4 = x_3$ $\quad y'_4 = x_1 \vee \bar{x}_2 x_3 \equiv 1+2.$

Таким образом, можно утверждать, что рассматриваемый модуль универсален в классе формул в базисе $\{\&, \vee, \bar{\ }\}$ из трех букв при использовании *PN*-классификации.

То, что мы изложили выше, отвечает на вопросы: «Какие модули делать?» и «Как их проектировать?» Но этого, конечно, недостаточно: необходимо еще знать, что делать с готовыми модулями, т.е. как построить из них схемы, обладающие минимальной элементной сложностью. Это и составляет предмет следующей главы.