

что все ДПИ обязательно должны работать с двоичными сигналами. Уже сегодня на аппаратном уровне созданы опытные образцы ДПИ, которые могут с аналогичной скоростью обрабатывать сигналы, имеющие больше чем два уровня. В качестве носителя информации привлекаются оптические, а не электрические сигналы. Имеется несколько причин, из-за которых в ближайшем будущем должен смениться принцип кодирования информации.

Во-первых, увеличение скорости работы "двоичных" компьютеров не может происходить бесконечно. Это увеличение достигается за счет все большей миниатюризации базовых элементов и уменьшения расстояний между ними. Имеется точный порог, сравнимый с размерами атома, за который эти размеры перешагнуть в принципе не смогут.

Во-вторых, чем больше четко различимых уровней может принимать сигнал, тем больше информации можно передать с его помощью при прочих равных условиях. Например, общепринятая единица информации байт состоит из 8 двоичных битов. Таким образом, один байт может принимать 2^8 (256) различных состояний и, соответственно, с помощью одного байта можно передать или сохранить один из 256 различных символов. Теперь предположим, что создано устройство, которое при тех же размерах и скорости работает с троичными разрядами. Одной восьмеркой троичных разрядов можно закодировать 3^8 (6561) различных символов. Таким образом один "троичный байт" при тех же размерах может кодировать не одну букву, а несколько (условно говоря — "слог"), что позволит значительно увеличить скорость обработки информации и уменьшить физический объем, необходимый для ее хранения. А если сигнал может принимать больше чем три различных уровня? Очевидно, что k -значные функции являются весьма перспективным направлением исследования как с теоретической, так и с практической точек зрения.

В заключение отметим, что для успешного усвоения материала данного учебного пособия, необходимо владение базовыми понятиями теории булевых функций, такими как представление булевых функций в табличном или в термальном виде, эквивалентные преобразования термов, различные канонические нормальные фор-

мы булевых функций, а так же представление о сопутствующих этим понятиям методах и определениях. Со всеми этими понятиями можно познакомиться в [9], [10] или [13].

Принятые сокращения:

- ДПИ — дискретный преобразователь информации;
- СФЭ — схема из функциональных элементов;
- КА — конечный автомат;
- ДНФ — дизъюнктивная нормальная форма;
- ПЛМ — программируемая логическая матрица;
- МТ — машина Тьюринга;
- СЗУ — считывающе-записывающее устройство.

1 Схемы из функциональных элементов

1.1 Основные понятия и определения

Итак, в данном разделе рассматриваются устройства с входами и выходами, работающие в дискретное время с дискретной информацией. В простейшем случае значения на входах ДПИ зависят исключительно от того, что подается на вход в этот же такт. Это значит, что ранее поступившая информация не может быть учтена при решении задачи, т. е. устройство не обладает памятью. Моделью такого устройства является *схема из функциональных элементов* (СФЭ). Очевидно, что в случае, когда у устройства нет памяти, значение каждого выхода f_i есть функция входов x_1, x_2, \dots, x_n и f_i есть $f_i(x_1, x_2, \dots, x_n)$.

Если входы и выходы устройства могут принимать k различных между собой значения, то f_i есть функция k -значной логики. В этом случае для нахождения внутреннего устройства ДПИ, представленного на рис. 2 в виде прямоугольника, нужно использовать теорию k -значных функций.

Идея построения СФЭ, реализующей некоторое преобразование информации, напрямую связана с представлением данного процес-

са в виде системы k -значных функций и их дальнейшей реализации в виде термов.

При построении СФЭ важно понятие базиса, в котором происходит построение этой схемы. Под функциональным элементом будем понимать объект с v занумерованными входами и одним выходом. На своем выходе функциональный элемент реализует k -значную функцию, зависящую от v аргументов (рис. 2). Данный элемент будем считать неделимым.

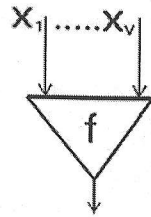


Рис. 2

Пусть задан некоторый конечный набор функциональных элементов $B = \{f_1, f_2, \dots, f_m\}$. Этот набор будем называть базисом, а его элементы – базисными. Из этих элементов по определенным правилам строятся СФЭ над B .

Далее будем рассматривать базисы из двоичных функциональных элементов или, иначе говоря, базисы из функциональных элементов, реализующих булевы функции. Эти функции являются важнейшим случаем k -значных функций как с теоретической, так и практической точек зрения.

1.2 СФЭ с двоичными входами и выходами

Пусть имеется базис из двоичных функциональных элементов $B = \{f_1, f_2, \dots, f_m\}$. Дадим следующее определение *схемы из функциональных элементов (СФЭ) над B* .

а) Вход с поставленной ему в соответствие переменной x_i есть СФЭ. В этом случае выход СФЭ совпадает с его входом (рис. 3).



Рис. 3

б) Если Σ_1 и Σ_2 есть схемы из функциональных элементов, то конструкция Σ , полученная путем их объединения есть СФЭ (рис. 4). В этом случае множество входов полученной схемы Σ есть объединение входов схем Σ_1 и Σ_2 . Множество выходов полученной схемы Σ есть объединение выходов схем Σ_1 и Σ_2 .

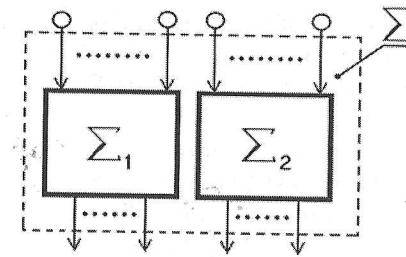


Рис. 4

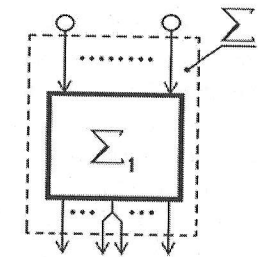


Рис. 5

в) Если Σ_1 есть СФЭ, то конструкция Σ , полученная расщеплением i -го выхода схемы Σ_1 на два выхода есть СФЭ (рис. 5). В этом случае множество входов схемы Σ совпадает с множеством входов схемы Σ_1 . Множество выходов схемы Σ есть все выходы схемы Σ_1 , кроме i -го. Вместо него к выходам схемы Σ добавляются два вновь образованных выхода.

г) Если Σ_1 есть СФЭ и f_i – функциональный элемент из базиса B , то конструкция Σ , полученная путем присоединения всех входов элемента f_i к некоторым выходам схемы Σ_1 есть СФЭ (рис. 6). В этом случае множество входов схемы Σ совпадает с множеством входов схемы Σ_1 . Множество выходов схемы Σ есть все выходы схемы Σ_1 , за исключением тех, которые были присоединены к входам элемента f_i . Вместо них к выходам схемы Σ добавляется выход элемента f_i .

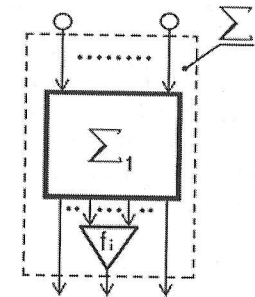


Рис. 6

1.2.1 Анализ и синтез СФЭ

Под анализом заданной СФЭ будем понимать решение задачи на нахождение множества булевых функций, которые реализуются на ее выходах.

а) Если СФЭ есть переменная x_i , то значение функции на выходе равно значению переменной x_i .

б) Если схема Σ получена путем объединения схем Σ_1 и Σ_2 , причем на выходах Σ_1 реализованы функции g_1, g_2, \dots, g_m , на выходах Σ_2 реализованы функции h_1, h_2, \dots, h_s , то на выходах Σ реализованы функции $g_1, \dots, g_m, h_1, \dots, h_s$.

в) Если схема Σ получена путем расщепления i -го выхода схемы Σ_1 , причем на выходах Σ_1 реализованы функции $g_1, g_2, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_m$, то на выходах Σ реализованы функции $g_1, g_2, \dots, g_{i-1}, g_i, g_i, g_{i+1}, \dots, g_m$.

г) Если схема Σ получена путем присоединения к выходам схемы Σ_1 с номерами i_1, i_2, \dots, i_k входов функционального элемента f_j , причем на выходах схемы Σ_1 реализованы функции g_1, g_2, \dots, g_m , то на выходах схемы Σ реализованы функции $g_1, \dots, g_{i_1-1}, g_{i_1+1}, \dots, g_{i_k-1}, g_{i_k+1}, \dots, g_m$, а на новом выходе реализована функция $f_j(g_{i_1}, g_{i_2}, \dots, g_{i_k})$.

Система уравнений, ставящая в соответствие каждому выходу схемы булеву функцию, которая на нем реализуется по вышеописанным правилам называется проводимостью схемы.

Под синтезом СФЭ будем понимать построение СФЭ, реализующей над заданным базисом заданную систему булевых функций. Как правило, при постановке задач на синтез СФЭ, требуется найти достаточно простую схему, реализующую данную систему, и это значительно затрудняет решение задачи.

Под сложностью СФЭ будем понимать число функциональных элементов, из которых она состоит. Обозначим сложность схемы Σ через $L(\Sigma)$. Ясно, что чем меньше элементов содержит схема, тем она надежнее и экономичнее как по себестоимости, так и по времени прохождения сигнала. Заданную систему булевых функций можно реализовать разными схемами, у каждой из которых может быть своя сложность. Схему с минимальной сложностью, реализующую некоторую систему булевых функций, будем называть минимальной для этой системы, а ее сложность — сложностью данной системы булевых функций.

Нахождение минимальной схемы в некотором базисе для произвольной системы булевых функций является весьма трудоемкой задачей, связанной с почти полным перебором. Как правило, на практике такой перебор неосуществим, в связи с чем при постанов-

ке задачи на синтез СФЭ ставится задача минимизировать сложность схемы в классе схем специального вида.

Далее рассмотрим некоторые методы синтеза СФЭ в наиболее простом базисе B_0 , состоящем из конъюнктора, дизъюнктора и инвертора.

1.2.2 Некоторые методы синтеза СФЭ в базисе $\{\&, \vee, -\}$ и их сложность

A1. Алгоритм, основанный на представлении булевой функции в виде СДНФ или СКНФ.

Введем прежде всего специальное представление для n -местной конъюнкции (рис. 7).

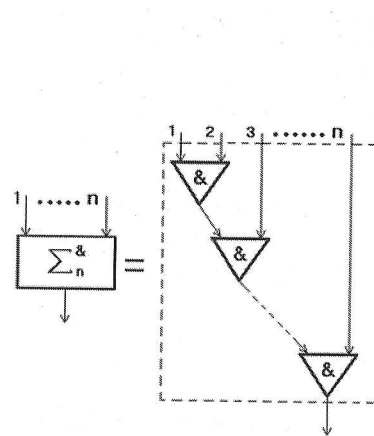


Рис. 7

Видно, что сложность $\Sigma_n^{\&}$ равна $n - 1$. По аналогии построим подсхему Σ_n^{\vee} .

Рассмотрим представление булевой функции в виде СДНФ:

$$f(x_1, \dots, x_n) = \bigvee_{\bar{\alpha}: f(\bar{\alpha})=1} x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n} = \bigvee_{i=1}^s K_i,$$

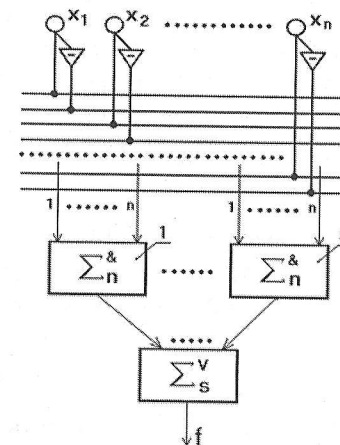


Рис. 8

$$\text{где } x_i^{\alpha_i} = \begin{cases} x_i, & \text{если } \alpha_i = 1; \\ \bar{x}_i, & \text{если } \alpha_i = 0. \end{cases}$$

Реализуя эту конструкцию напрямую, получим следующую схему (рис. 8). Здесь n – число аргументов у функции f , s – число полных элементарных конъюнкций в ее СДНФ. Сложность данной схемы равна $n + s(n - 1) + (s - 1)$, откуда

$$L_{A1}(n) \leq n + ns, \quad (1)$$

где $L_{A1}(n)$ – сложность самой сложной схемы среди всех схем с n входами, полученных алгоритмом $A1$.

Рассмотрим далее представление некоторой булевой функции в виде СДНФ или СКНФ. Выбирать из этих двух вариантов будем в зависимости от числа единиц и нулей в векторе, задающем эту функцию. Если в векторе меньше нулей, используем СКНФ, если единиц – СДНФ. Очевидно, что для СКНФ можно построить схему с помощью несколько модернизированного алгоритма $A1$, поменяв местами многоместные конъюнкции и дизъюнкции. Построенная схема будет аналогична приведенной на рис. 8.

Ясно, что для произвольной булевой функции от n аргументов можно выбрать СДНФ или СКНФ, где число элементарных конъюнкций или элементарных дизъюнкций не превышает 2^{n-1} , т. е. $s \leq 2^{n-1}$, и, используя (1), получим:

$$L_{A1}(n) \leq \frac{n}{2} \cdot 2^n + n \sim \frac{n}{2} \cdot 2^n.$$

A2. Алгоритм, основанный на более компактной реализации множества всех полных элементарных конъюнкций.

Напомним, что полной элементарной конъюнкцией от n переменных называется терм вида $x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n}$, т. е. конъюнкция всех n переменных, часть которых может входить с отрицанием. Если строить в виде СФЭ не каждую конъюнкцию в отдельности, а все их множество сразу, то можно значительно уменьшить сложность схемы. На рис. 9 приведено индуктивное построение многополосника M^n , который реализует на своих выходах все 2^n полных элементарных конъюнкций. Для построения СФЭ, реализующей заданную функцию f , необходимо для нее построить СДНФ.

Пусть она имеет вид $K_1 \vee K_2 \vee \dots \vee K_s$. Далее для каждой из s полных элементарных конъюнкций необходимо найти в многополоснике M^n реализующий ее выход и все их соединить через многоместную дизъюнкцию Σ_s^{\vee} .

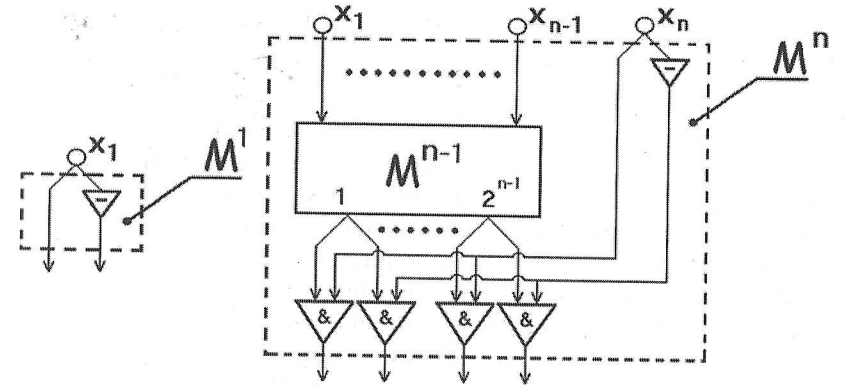


Рис. 9

Оценим теперь сложность схем, которые получаются данным алгоритмом. Видно, что

$$L(M^1) = 1,$$

$$L(M^n) = L(M^{n-1}) + 2^n + 1.$$

Раскрывая последнее рекуррентное соотношение до M^1 , получим:

$$L(M^n) = L(M^1) + 2^2 + 1 + 2^3 + 1 + \dots + 2^n + 1 = n + 2^{n+1} - 4.$$

Так как s не превосходит 2^n , то $L(\Sigma_s^{\vee})$ не превосходит $2^n - 1$. Таким образом, в итоге получим

$$L_{A2}(n) \leq 3 \cdot 2^n + n - 5 \sim 3 \cdot 2^n.$$

Отметим, что все приведенные рассуждения, сделанные для СДНФ, можно по аналогии сделать и для СКНФ.

А3. Алгоритм, основанный на разложении функции по остаточным.

Нулевой остаточной функции f по переменной x_i будем называть функцию $f_{x_i}^0 = f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$. По аналогии определим единичную остаточную $f_{x_i}^1$.

Легко проверить, что верно следующее разложение:

$$f = \bar{x}_i f_{x_i}^0 \vee x_i f_{x_i}^1.$$

Таким образом, для построения СФЭ, реализующей функцию f размерности n , необходимо построить СФЭ для ее остаточных по некоторой переменной x_i . Для реализации данного способа нужно воспользоваться следующей индуктивной процедурой (рис. 10).

Каждую остаточную, в свою очередь можно разложить тем же способом, например на втором шаге получим:

$$f = \bar{x}_i \bar{x}_j f_{x_i x_j}^{00} \vee \bar{x}_i x_j f_{x_i x_j}^{01} \vee x_i \bar{x}_j f_{x_i x_j}^{10} \vee x_i x_j f_{x_i x_j}^{11}.$$

Произведем подобную операцию до определенного предела, например до тех пор, пока все остаточные не станут размерности два. Далее воспользуемся для каждой такой остаточной ее соответствующим минимальным представлением в базе B_0 :

$(0000) = x \cdot \bar{x}$	$(1000) = \bar{x} \vee \bar{y}$
$(0001) = x \cdot y$	$(1001) = \bar{x} \vee \bar{y} \vee xy$
$(0010) = x \cdot \bar{y}$	$(1010) = \bar{y}$
$(0011) = x$	$(1011) = x \vee \bar{y}$
$(0100) = \bar{x} \cdot y$	$(1100) = \bar{x}$
$(0101) = y$	$(1101) = \bar{x} \vee y$
$(0110) = \bar{x} \cdot \bar{y} \cdot (x \vee y)$	$(1110) = \bar{x} \cdot \bar{y}$
$(0111) = x \vee y$	$(1111) = x \vee \bar{x}$

Видно, что сложность представления любой бинарной булевой функции в базе B_0 не превосходит четырех. Оценим, исходя из этого, сложность схемы, получаемой данным алгоритмом. Видно (рис. 10), что

$$\begin{aligned} L_{A3}(n) &\leq 2 \cdot L_{A3}(n-1) + 4 \leq 2 \cdot (2 \cdot L_{A3}(n-2) + 4) + 4 \leq \dots \\ &\leq 2 \cdot \underbrace{(2 \cdot \dots \cdot 2 \cdot (2 \cdot L_{A3}(2) + 4) + 4)}_{n-2} \dots + 4 = \\ &= 2^{n-2} \cdot L_{A3}(2) + 2^{n-3} \cdot 4 + 2^{n-4} \cdot 4 + \dots + 2^1 \cdot 4 + 2^0 \cdot 4 = \\ &= 2^{n-2} \cdot 4 + 4 \cdot (2^{n-2} - 1) = \frac{8}{4} \cdot 2^n - 4. \end{aligned}$$

Таким образом $L_{A3}(n) \sim 2 \cdot 2^n$. Можно улучшить эту оценку, если остановиться на остаточных размерности три, и для каждой из них подставить в СФЭ ее минимальное представление.

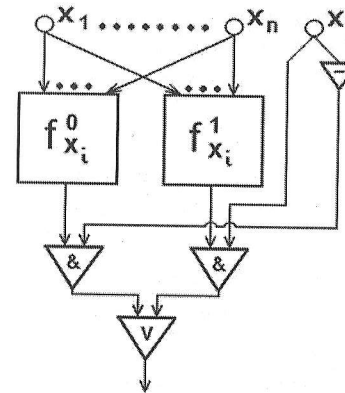


Рис. 10

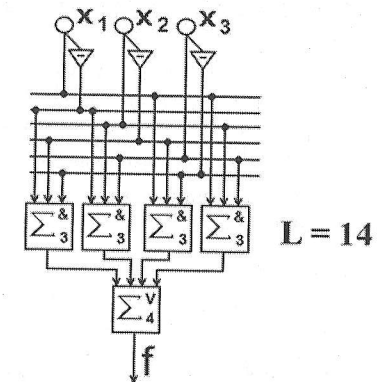


Рис. 11

А4. Асимптотически наилучший алгоритм синтеза СФЭ (метод Лупанова).

При рассмотрении универсальных алгоритмов синтеза СФЭ в базе $B_0 = \{\&, \vee, -\}$, была проведена оценка сложности получающихся в результате схем. Можно видеть, что

$$L_{A1}(n) \leq \frac{n}{2} \cdot 2^n + n \sim \frac{n}{2} \cdot 2^n$$

$$L_{A2}(n) \leq 3 \cdot 2^n + n - 5 \sim 3 \cdot 2^n$$

$$L_{A3}(n) \leq 2 \cdot 2^n - 4 \sim 2 \cdot 2^n$$

Напомним, что величина $L(n)$ асимптотически не превосходит величину $B(n)$, если существует некоторое $C(n)$, такое, что

$$L(n) \leq C(n) \text{ и } \lim_{n \rightarrow \infty} \frac{C(n)}{B(n)} = 1.$$

Отметим, что в этом случае величина $L(n)$ может быть "чуть-чуть" больше $B(n)$, но это различие невелико относительно величин $L(n)$ или $B(n)$ и им можно пренебречь при достаточно больших n .

Асимптотическая оценка дает представление о сложности самых сложных СФЭ, получаемых данным методом, при достаточно больших n . Эта характеристика является значимой для метода, поскольку позволяет сравнивать алгоритмы между собой. Становится ясно, что А1 в этом смысле самый "плохой" из рассмотренных, а А3 — "лучше". Отметим, что данное сравнение происходит по самым худшим, т.е. самым сложным схемам.

Возникает вопрос: до каких пор можно уменьшить коэффициент при 2^n в оценке алгоритма, иными словами, какова асимптотическая оценка самого лучшего универсального алгоритма синтеза СФЭ в базисе B_0 .

Теорема. С ростом n , для почти всех булевых функций от n аргументов сложность самой простой СФЭ над B_0 превосходит величину $\frac{2^n}{n}$.

Здесь выражение "для почти всех" означает, доля функций от n аргументов, у которых сложность лучшей СФЭ больше $\frac{2^n}{n}$, стремится к 1 по отношению к числу всех булевых функций от n аргументов при $n \rightarrow \infty$, т.е. это почти все булевы функции.

Данная теорема говорит о том, что каков бы ни был алгоритм синтеза СФЭ над B_0 , его асимптотическая оценка не может быть ниже $\frac{2^n}{n}$. Доказательство теоремы проводится мощностным методом и не позволяет найти конкретный алгоритм синтеза СФЭ с указанной асимптотической оценкой.

Первый метод синтеза СФЭ в базисе B_0 , асимптотическая оценка которого совпадает с $\frac{2^n}{n}$ был предложен О.Б. Лупановым. Метод основан на матричном представлении булевой функции $f(x_1, \dots, x_n)$ и разложении ее по k аргументам ($k < n$). Одновременно с описанием метода будем пояснять его на примере функции от восьми аргументов, представленной в виде матрицы на рис. 12.

	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
x_5	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
x_6	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1
x_7	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$x_1 x_2 x_3 x_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0 0 0 0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	1
0 0 0 1	1	0	1	1	0	1	0	1	0	0	0	0	1	0	1	0
0 0 1 0	1	1	1	0	1	0	1	1	1	0	1	1	0	0	1	0
0 0 1 1	0	1	1	0	1	1	1	0	1	1	1	1	0	0	1	1
0 1 0 0	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	0
0 1 0 1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1
0 1 1 0	1	1	1	1	1	1	0	0	1	1	0	1	1	0	1	0
0 1 1 1	0	0	0	1	1	1	0	1	1	1	0	0	1	1	1	1
1 0 0 0	1	1	0	1	1	1	1	0	1	0	0	1	1	1	1	1
1 0 0 1	1	1	1	1	0	1	0	1	1	1	1	0	0	1	0	1
1 0 1 0	0	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0
1 0 1 1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	0
1 1 0 0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1
1 1 0 1	0	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1
1 1 1 0	0	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1
1 1 1 1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	1

Рис. 12

Принцип помещения функции $f(x_1, \dots, x_n)$ в матрицу прост: на пересечении строки $(\sigma_1, \sigma_2, \dots, \sigma_k)$ и столбца $(\sigma_{k+1}, \dots, \sigma_n)$ находится значение функции $f(\sigma_1, \sigma_2, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n)$.

Легко видеть, что в столбце $(\sigma_{k+1}, \dots, \sigma_n)$ помещается остаточная функция $f_{x_{k+1} \dots x_n}^{\sigma_{k+1} \dots \sigma_n}(x_1, \dots, x_k)$. Например, на рис. 12 в выделенном столбце 0101 содержится остаточная $f_{x_5 x_6 x_7 x_8}^{0101}(x_1, x_2, x_3, x_4) = (010110111001111)$.

В общем случае таких остаточных будет 2^{n-k} , где x_1, \dots, x_k — переменные, соответствующие строкам матрицы, а $x_{k+1} \dots x_n$ — переменные, соответствующие ее столбцам. В примере таких остаточных 16, для удобства обозначим их f_0, f_1, \dots, f_{15} .

Раскладывая функцию f по переменным $x_{k+1} \dots x_n$, получим

разложение

$$f = \bigvee_{\sigma_{k+1} \dots \sigma_n} x_{k+1}^{\sigma_{k+1}} \dots x_n^{\sigma_n} \cdot f_{x_{k+1} \dots x_n}^{\sigma_{k+1} \dots \sigma_n}(x_1, \dots, x_k). \quad (2)$$

Для нашего примера, раскладывая f по последним четырем переменным получим разложение

$$f = \bar{x}_5 \cdot \bar{x}_6 \cdot \bar{x}_7 \cdot \bar{x}_8 \cdot f_0 \vee \bar{x}_5 \cdot \bar{x}_6 \cdot \bar{x}_7 \cdot x_8 \cdot f_1 \vee \bar{x}_5 \cdot \bar{x}_6 \cdot x_7 \cdot \bar{x}_8 \cdot f_2 \vee \dots \vee x_5 \cdot x_6 \cdot x_7 \cdot x_8 \cdot f_{15}.$$

Зададим некоторое число $s < 2^k$. Пусть в нашем примере $s = 3$. Разобьем каждую из остаточных разложения (2) на куски длиной s согласно рис. 12. Последний кусок каждой остаточной по длине может быть меньше s . Всего таких кусков $p = \lceil \frac{2^k}{s} \rceil$, т.е. ближайшее натуральное число, большее или равное $\frac{2^k}{s}$. В данном примере $p = \lceil \frac{2^4}{3} \rceil = 6$, т.е. каждая остаточная разбивается на 6 кусков.

Обозначим через f_m^i булеву функцию, совпадающую в i -м куске с функцией f_m и равную 0 вне его. В таком случае, например, $f_5^2 = (000 110 000 000 000 0)$.

Так как куски не пересекаются, то дизъюнкция всех f_m^i для i от 1 до p равна f_m :

$$f_m = \bigvee_{i=1}^p f_m^i. \quad (3)$$

Например $f_5 = f_5^1 \vee f_5^2 \vee f_5^3 \vee f_5^4 \vee f_5^5 \vee f_5^6$:

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \vee \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Функцию f_m^i реализуем в виде СДНФ. Ясно, что число полных элементарных конъюнкций в СДНФ(f_m^i) не будет превышать s , так как в f_m^i не более чем s единиц.

Осталось заметить, что для фиксированного i число различных f_m^i не превосходит 2^s , так как длина i -го куска не превышает s . Так, для нашего примера, во второй полосе среди f_m^2 встречается шесть различных, а всего их может быть не больше восьми.

Построение СФЭ для функции f проведем по схеме, представленной на рис. 13.

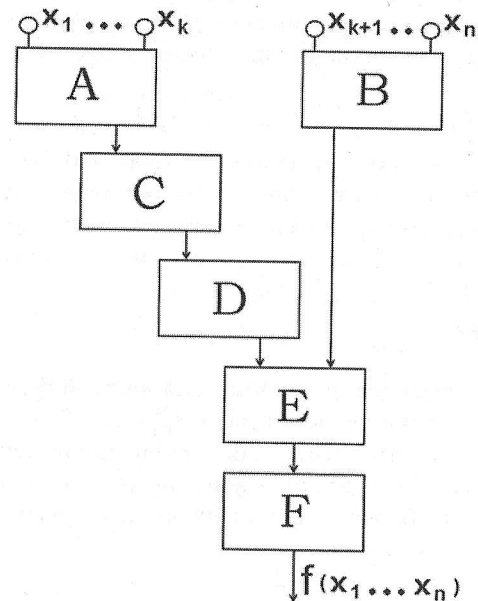


Рис. 13

Блок А реализует все элементарные конъюнкции $x_1^{\sigma_1} \dots x_k^{\sigma_k}$. Всего таких конъюнкций 2^k , каждая из них содержит $k - 1$ конъюнкторов. Добавив k инверторов, получим, что

$$L(A) \leq (k - 1) \cdot 2^k + k \leq k \cdot 2^k.$$

Блок А имеет 2^k выходов.

Блок В реализует все элементарные конъюнкции $x_{k+1}^{\sigma_{k+1}} \dots x_n^{\sigma_n}$ тем же способом. По аналогии

$$L(B) \leq (n - k) \cdot 2^{n-k}.$$

Блок В имеет 2^{n-k} выходов.

Блок С реализует все функции f_m^i в виде их СДНФ. Каждая f_m^i содержит не более s единиц, а значит требует не более s элементарных конъюнкций из блока А, которые соединяются через $s - 1$ дизъюнкций. В i -ой полосе содержится не более чем 2^s различных кусков, т.е. для реализации всех f_m^i из i -ой полосы нужно не более чем $2^s \cdot (s - 1)$ дизъюнкций. Всего таких полос p штук, отсюда

$$L() \leq p \cdot (s - 1) \cdot 2^s < p \cdot s \cdot 2^s.$$

Блок С имеет $p \cdot 2^s$ выходов, по одному для каждой функции f_m^i .

Блок D реализует все функции f_m по формуле (3). Согласно этой формуле, p функций f_m^i из блока С соединяются при помощи $p - 1$ дизъюнкций. Всего функций f_m (так как они остаточные по $x_{k+1} \dots x_n$) ровно 2^{n-k} штук. Отсюда

$$L(D) \leq (p - 1) \cdot 2^{n-k} < p \cdot 2^{n-k}.$$

Блок D имеет 2^{n-k} выходов, по одному для каждой функции f_m .

Блок E реализует конъюнкции вида $x_{k+1}^{\sigma_{k+1}} \dots x_n^{\sigma_n} \cdot f_m$ из формулы (2). Для этого соответствующая элементарная конъюнкция $x_{k+1}^{\sigma_{k+1}} \dots x_n^{\sigma_n}$ из блока В соединяется конъюнктом с соответствующей функцией f_m из блока D. По числу элементарных конъюнкций

$$L() \leq 2^{n-k}.$$

Блок E имеет 2^{n-k} выходов.

Блок F реализует всю функцию $f(x_1 \dots x_n)$ по формуле (2). Для этого в нем через $2^{n-k} - 1$ дизъюнкций соединяются все выходы блока E. Отсюда

$$L(F) \leq 2^{n-k}.$$

Суммируя сложности всех блоков от А до F, получим:

$$L_{A4}(n) \leq k \cdot 2^k + (n - k) \cdot 2^{n-k} + p \cdot s \cdot 2^s + p \cdot 2^{n-k} + 2 \cdot 2^{n-k}, \quad (4)$$

где числа k и s выбираются произвольно с условием, что $k < n$, $s < 2^k$. $p = \lceil \frac{2^k}{s} \rceil$.

Некоторый анализ, на котором не будем здесь останавливаться, показывает, что следует взять для k и s такие значения:

$$k = \lfloor 3 \cdot \log_2 n \rfloor; \quad s = \lfloor n - 5 \cdot \log_2 n \rfloor.$$

Подставляя в (4) соответствующие значения, получим:

$$L_{A4}(n) \leq 3 \cdot \log_2 n \cdot n^3 + \frac{n - 3 \log_2 n}{n^3} \cdot 2^n + \frac{n^3 \cdot 2^n}{n^5} + \frac{2^n}{n - 5 \log_2 n} + 2 \cdot \frac{2^n}{n^3}.$$

Отметим, что в некоторых слагаемых для сохранения неравенства необходимо брать вместо k величину $k+1$, что, впрочем, никак не отразится на асимптотике, и потому, в целях упрощения изложения, не учитывается.

Покажем, что выражение в правой части последнего неравенства асимптотически равно $\frac{2^n}{n}$, для чего поделим каждое слагаемое на эту величину, и покажем, что сумма этих отношений стремится к 1 при $n \rightarrow \infty$.

$$3 \cdot \log_2 n \cdot n^3 / \frac{2^n}{n} = \frac{3 \log_2 n \cdot n^4}{2^n} \rightarrow 0 \quad \text{при } n \rightarrow \infty,$$

$$\frac{n - 3 \log_2 n}{n^3} \cdot 2^n / \frac{2^n}{n} = \frac{n - 3 \log_2 n}{n^2} = \frac{1}{n} - \frac{3 \log_2 n}{n^2} \rightarrow 0 \quad \text{при } n \rightarrow \infty,$$

$$\frac{2^n}{n^2} / \frac{2^n}{n} = \frac{1}{n} \rightarrow 0 \quad \text{при } n \rightarrow \infty,$$

$$2 \cdot \frac{2^n}{n^3} / \frac{2^n}{n} = \frac{2}{n^2} \rightarrow 0 \quad \text{при } n \rightarrow \infty,$$

$$\frac{2^n}{n - 5 \log_2 n} / \frac{2^n}{n} = \frac{n}{n - 5 \log_2 n} \rightarrow 1 \quad \text{при } n \rightarrow \infty,$$

так как обратная к этому выражению величина

$$\frac{n - 5 \log_2 n}{n} = 1 - \frac{5 \log_2 n}{n} \rightarrow 1 \quad \text{при } n \rightarrow \infty.$$

Складывая все асимптотики, получим, что

$L_{A4}(n)$ асимптотически не превосходит $\frac{2^n}{n}$, что и требовалось.

1.2.3 Иллюстрирующие примеры

Подробно разберем работу алгоритмов А1, А2 и А3 на примере функций размерности три и четыре.

1) Построить каждым из трех алгоритмов СФЭ над базисом B_0 для функции $f = (1001\ 0101)$. Найти сложность каждой схемы.

А1. Найдем СДНФ для функции f :

$$f = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3$$

Реализуем этот терм в виде СФЭ (рис. 11 на стр. 15).

А2. Метод основан на той же СДНФ, что использовалась в А1. Основные преимущества данного метода проявляются при достаточно больших размерностях функции. На этом примере продемонстрируем важность выбора правильного порядка подключения переменных. Построим многополосник для данной функции при порядке переменных x_1, x_2, x_3 и при порядке переменных x_1, x_3, x_2 (рис. 14).

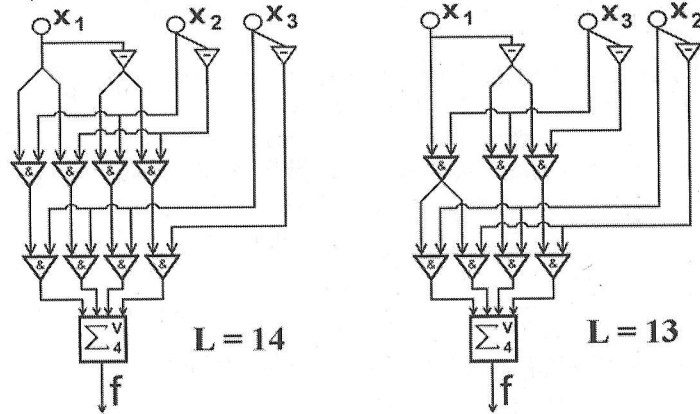


Рис. 14

А3. Этот метод, как было показано выше, является самым хорошим по сложности получающихся схем. Выбор переменных, по

которым берутся остаточные, здесь так же весьма важен. Рассмотрим все три возможных варианта выбора переменной, по которой производится разложение. Прежде всего рассмотрим остаточные по всем трем переменным:

$$f_{x_1}^0 = (1001) \quad f_{x_2}^0 = (1001) \quad f_{x_3}^0 = (1000)$$

$$f_{x_1}^1 = (0101) \quad f_{x_2}^1 = (0101) \quad f_{x_3}^1 = (0111).$$

Отметим, что остаточные функции зависят от переменных, вместо которых не подставляются константы 0 или 1. Тогда, используя таблицу, приведенную на странице 14, получим:

$$f_{x_1}^0 = x_2 \vee x_3 \vee x_2x_3 \quad f_{x_2}^0 = \overline{x_1 \vee x_3} \vee x_1x_3 \quad f_{x_3}^0 = \overline{x_1 \vee x_2}$$

$$f_{x_1}^1 = x_3 \quad f_{x_2}^1 = x_3 \quad f_{x_3}^1 = x_1 \vee x_2.$$

Получим следующие три разложения:

$$f = \bar{x}_1 \cdot (\overline{x_2 \vee x_3} \vee x_2x_3) \vee x_1x_3$$

$$f = \bar{x}_2 \cdot (\overline{x_1 \vee x_3} \vee x_1x_3) \vee x_2x_3$$

$$f = \bar{x}_3 \cdot \overline{x_1 \vee x_2} \vee x_3 \cdot (x_1 \vee x_2).$$

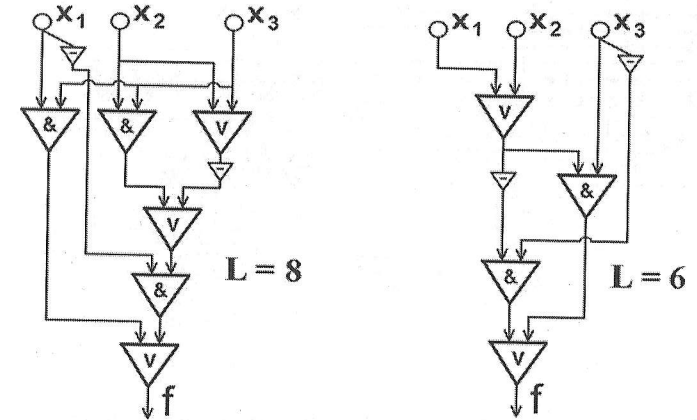


Рис. 15

Очевидно, что схемы для первых двух термов будут идентичны

по сложности, поэтому построим схемы для первого и третьего представления функции f (рис. 15).

Видно, что чем меньше среди остаточных функций вида (0110) и (1001), тем меньше сложность СФЭ, реализующей заданную функцию.

2) Построить каждым из трех алгоритмов СФЭ над базисом B_0 для функции $g = (0110\ 1001\ 1101\ 0111)$. Найти сложность каждой схемы.

A1. Построим СКНФ для функции g :

$$g = (x_1 \vee x_2 \vee x_3 \vee x_4) \cdot (x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4) \cdot (x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4) \& \& (x_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4) \cdot (\bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4) \cdot (\bar{x}_1 \vee \bar{x}_2 \vee x_3 \vee x_4).$$

Реализуем ее в виде СФЭ (рис. 16):

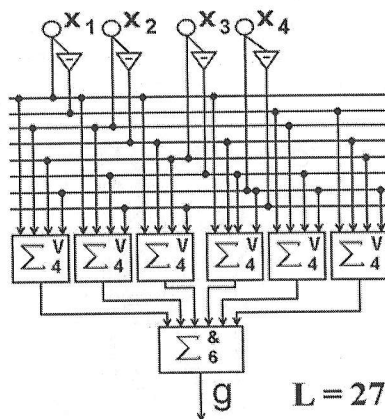


Рис. 16

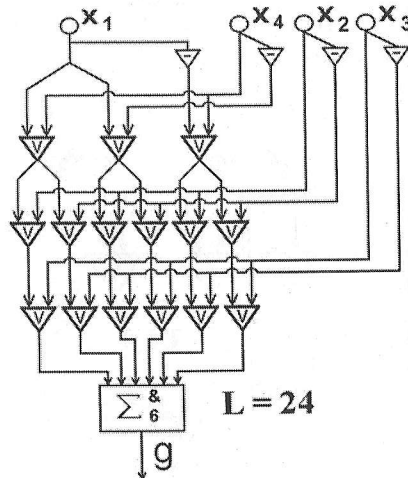


Рис. 17

A2. Построим компактный многополюсник для множества всех полных элементарных дизъюнкций, встречающихся в СКНФ функции g (рис. 17). Учтем, что при порядке подключения x_1, x_4, x_2, x_3 можно сэкономить одну дизъюнкцию.

A3. Для начала построим СФЭ, раскладывая g по переменной

x_2 и используя общий метод. Получим:

$$g_{x_2}^0 = (0110\ 1101) \quad g_{x_2}^1 = (1001\ 0111).$$

$g_{x_2}^0$ разложим по переменной x_3 , получим

$$g_{x_2}^0 = \bar{x}_3(x_1 \vee x_4) \vee x_3(\overline{x_1 \vee x_4} \vee x_1 x_4).$$

$g_{x_2}^1$ разложим по переменной x_3 , получим

$$g_{x_2}^1 = \bar{x}_3(\overline{x_1 \vee x_4} \vee x_1 x_4) \vee x_3(x_1 \vee x_4).$$

Тогда вся функция реализуется следующим термом:

$$\bar{x}_2(\bar{x}_3(x_1 \vee x_4) \vee x_3(\overline{x_1 \vee x_4} \vee x_1 x_4)) \vee x_2(\bar{x}_3(\overline{x_1 \vee x_4} \vee x_1 x_4) \vee x_3(x_1 \vee x_4)).$$

СФЭ имеет вид (рис. 18):

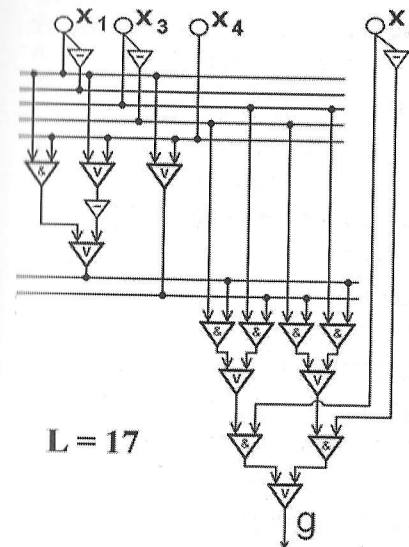


Рис. 18

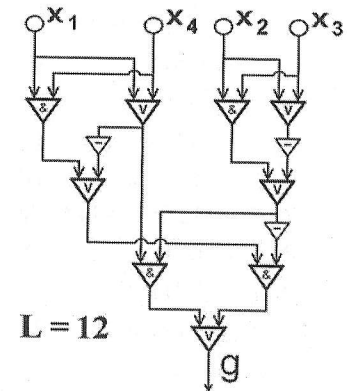


Рис. 19

Попробуем построить более экономную схему. Рассматривая множество всех остаточных размерности два, заметим, что по переменным $x_2 x_3$ имеются следующие остаточные:

$$g_{x_2 x_3}^0 = (0111) \quad g_{x_2 x_3}^0 = (1001) \quad g_{x_2 x_3}^1 = (1001) \quad g_{x_2 x_3}^1 = (0111),$$

т. е. попарно одинаковые. Используя этот факт, получим:

$$g = \overline{x_2x_3}(x_1 \vee x_4) \vee \overline{x_2x_3}(x_1 \vee x_4 \vee x_1x_4) \vee x_2\overline{x_3}(x_1 \vee x_4 \vee x_1x_4) \vee x_2x_3(x_1 \vee x_4) = (x_1 \vee x_4) \cdot (\overline{x_2x_3} \vee x_2x_3) \vee (x_1 \vee x_4 \vee x_1x_4) \cdot (\overline{x_2x_3} \vee x_2x_3).$$

Заметим, помимо этого, что $\overline{x_2x_3} \vee x_2x_3 = \overline{x_2 \vee x_3} \vee x_2x_3$,

а $\overline{x_2x_3} \vee x_2\overline{x_3} = \overline{x_2 \vee x_3} \vee x_2x_3$.

В итоге схема примет вид, приведенный на рис. 19.

Задачи для самостоятельного решения

Построить каждым из трех алгоритмов СФЭ над базисом B_0 для функции f . При построении с помощью А2 и А3 выбрать самую простую реализацию. Найти сложность полученных схем.

- | | |
|------------------------|------------------------------------|
| 1. $f = (1000\ 0001)$ | 11. $f = (0001\ 0100\ 1000\ 0001)$ |
| 2. $f = (1001\ 0100)$ | 12. $f = (0111\ 1010\ 0101\ 0011)$ |
| 3. $f = (0001\ 0111)$ | 13. $f = (1110\ 0001\ 0110\ 0111)$ |
| 4. $f = (0110\ 1001)$ | 14. $f = (0110\ 1001\ 1001\ 0110)$ |
| 5. $f = (0011\ 0101)$ | 15. $f = (0111\ 1111\ 0010\ 0110)$ |
| 6. $f = (1001\ 1011)$ | 16. $f = (1001\ 0110\ 0011\ 1100)$ |
| 7. $f = (1101\ 1011)$ | 17. $f = (1001\ 0101\ 0011\ 1001)$ |
| 8. $f = (0110\ 1101)$ | 18. $f = (0110\ 1011\ 1001\ 1101)$ |
| 9. $f = (1011\ 0110)$ | 19. $f = (1011\ 0110\ 1101\ 1011)$ |
| 10. $f = (0111\ 0100)$ | 20. $f = (0110\ 1101\ 1011\ 0110)$ |

1.2.4 Построение вычисляющих устройств

Данный раздел посвящен решению задач на построение простейших вычисляющих устройств в виде СФЭ над базисом B_0 . Проектируемые устройства будут работать с числами в двоичной записи. Разрядность обрабатываемого числа будет ограничена количеством входов схемы, выделенных для него. Результат вычислений подается на выходы схемы, для каждого разряда ответа необходим свой выход.

Решим следующую задачу:

Построить двухразрядный двоичный сумматор. Устройство должно находить сумму двух двузначных двоичных чисел.

Очевидно, что сумма двух двузначных чисел может быть трехзначной, например $11 + 10 = 101$. Таким образом, проектируемое устройство должно иметь четыре входа (по два на каждое суммируемое число) и три выхода. Его общий вид представлен на рис. 20. На входы x подается первое суммируемое число, x_i соответствует i -му разряду. На входы y подается второе суммируемое число, y_i соответствует i -му разряду. На выходах z формируется сумма подающихся на вход чисел, z_i соответствует i -му разряду ответа. Решение задачи состоит в построении схемы над B_0 , реализующей данный процесс.

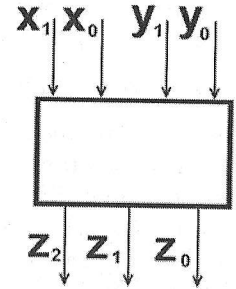


Рис. 20

При этом поставим дополнительное условие: схема должна иметь как можно меньшую сложность.

x_1	x_0	y_1	y_0	z_2	z_1	z_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Построим табличное задание для функций z_2 , z_1 и z_0 . Они будут зависеть от четырех переменных x_1 , x_0 , y_1 и y_0 . Необходимо перебрать все пары двузначных чисел и указать для каждой ее сумму.

Для примера, в таблице выделена строка, соответствующая двойной сумме двух чисел $01_2 + 11_2 = 100_2$.

Дальнейшая задача — реализовать систему булевых функций z_2 , z_1 и z_0 в виде возможно более простой СФЭ над базисом B_0 . Воспользуемся алгоритмом А3, т. е. методом разложения по остаточным.

Отметим, что среди остаточных по переменным x_1 и x_0 встречаются следующие функции: (0000), (0001), (0011), (0101), (0110), (0111) и их отрицания. Раскладывая по x_1 и x_0 получим:

$$z_2 = \bar{x}_1 \bar{x}_0 (\bar{y}_1 \cdot y_1) \vee \bar{x}_1 x_0 (y_1 \cdot y_0) \vee x_1 \bar{x}_0 y_1 \vee x_1 x_0 (y_1 \vee y_0)$$

$$z_1 = \bar{x}_1 \bar{x}_0 y_1 \vee \bar{x}_1 x_0 (\bar{y}_1 y_0 \vee y_1 \bar{y}_0) \vee x_1 \bar{x}_0 \bar{y}_1 \vee x_1 x_0 (\bar{y}_1 y_0 \vee y_1 \bar{y}_0)$$

$$z_0 = \bar{x}_0 y_0 \vee x_0 \bar{y}_0.$$

Реализуя это представление в виде СФЭ, получим требуемый двоичный сумматор. На рис. 21 помимо самой схемы приведено ее тестирование. На входы x_1, x_0 подается 01, на входы y_1, y_0 подается 11. На выходах z_2, z_1, z_0 получается 100, что и требовалось. Можно убедиться, что и для всех остальных вариантов схема работает верно.

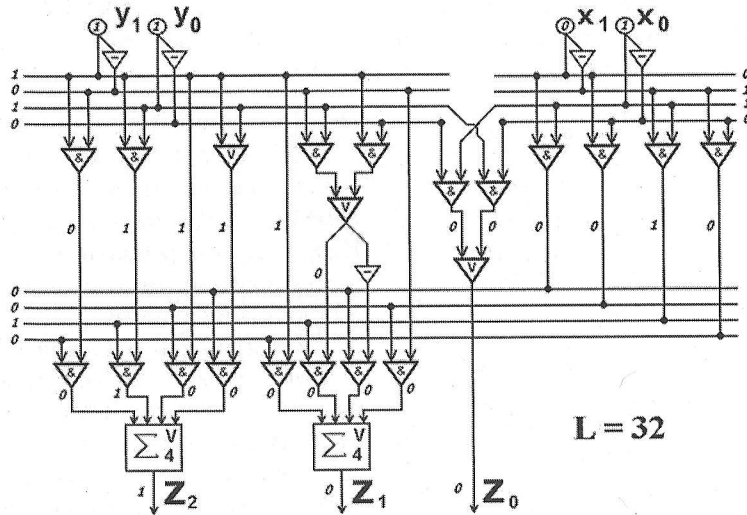


Рис. 21

Давно известно, что универсальные алгоритмы, как правило, дают не самые лучшие результаты. Построить значительно более простую СФЭ, являющуюся двоичным сумматором и имеющую

линейную а не экспоненциальную сложность, можно, следующим образом.

Предлагается воспользоваться алгоритмом сложения столбиком и ввести в рассмотрение перенос из предыдущего разряда q_i :

$$\begin{array}{r} \overbrace{x_n}^{q_{n+1}} \quad \overbrace{x_{n-1}}^{q_n} \quad \overbrace{\dots}^{q_{n-1}} \quad \overbrace{x_1}^{q_2} \quad \overbrace{x_0}^{q_1} \quad \overbrace{}^{q_0} \\ + \\ y_n \quad y_{n-1} \quad \dots \quad y_1 \quad y_0 \\ \hline z_{n+1} \quad z_n \quad z_{n-1} \quad \dots \quad z_1 \quad z_0 \end{array}$$

Легко показать, что

$$z_i = x_i \oplus y_i \oplus q_i,$$

$$q_{i+1} = x_i y_i \vee x_i q_i \vee y_i q_i = x_i y_i \vee (x_i \vee y_i) q_i.$$

Действительно, значение очередного разряда суммы равно сумме по mod 2 соответствующих разрядов слагаемых и переноса из предыдущего разряда. Что касается переноса в следующий разряд, то он равен 1, если хотя бы два из этих трех суммируемых элементов в данном разряде равны 1. Кроме того, ясно, что $q_0 = 0$, $z_0 = x_0 \oplus y_0 = \bar{x}_0 y_0 \vee x_0 \bar{y}_0$, $q_1 = x_0 y_0$.

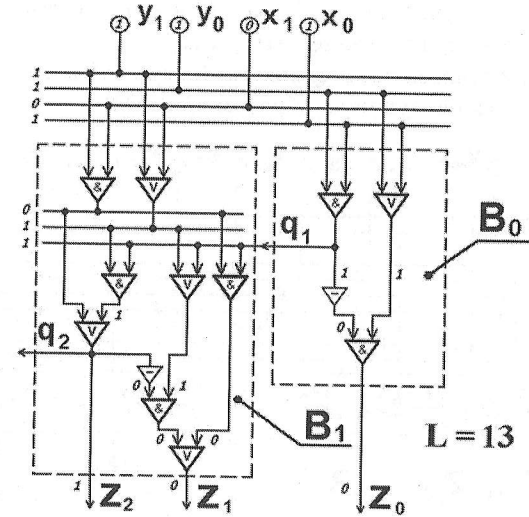


Рис. 22

Осталось заметить следующее тождество:

$$x_i \oplus y_i \oplus q_i = \overline{(x_i y_i \vee (x_i \vee y_i) q_i)} \cdot (x_i \vee y_i \vee q_i) \vee x_i y_i q_i. \quad (5)$$

Отсюда, для двухразрядного сумматора получим:

$$z_1 = x_1 \oplus y_1 \oplus q_1 = \overline{(x_1 y_1 \vee (x_1 \vee y_1) q_1)} \cdot (x_1 \vee y_1 \vee q_1) \vee x_1 y_1 q_1,$$

$$z_2 = q_2 = x_1 y_1 \vee x_1 q_1 \vee y_1 q_1 = x_1 y_1 \vee (x_1 \vee y_1) q_1.$$

Получим значительно более экономную СФЭ для двухразрядного двоичного сумматора (рис. 22). Здесь же приведен тест на том же наборе 1101.

Отметим, что представление (5) позволяет построить сумматор для чисел $x_n x_{n-1} \dots x_1 x_0$ и $y_n y_{n-1} \dots y_1 y_0$ произвольной наперед заданной длины.

Для этого нужно блок B_1 продублировать n раз, в каждом из получившихся блоков B_i , кроме n -го, выход, соответствующий q_2 переименовать в выход q_{i+1} , выход z_2 удалить. При помощи выхода q_{i+1} , блок B_i подключается к соответствующему входу блока B_{i+1} , аналогично тому, как блок B_0 посредством выхода q_1 подключен к блоку B_1 (см. рис. 22). В блоке B_n выход, соответствующий q_2 удалить, выход z_2 переименовать в z_{n+1} .

Блок B_0 остается без изменения. Полученные $n+1$ блоков следует соединить следующим образом (рис. 23):

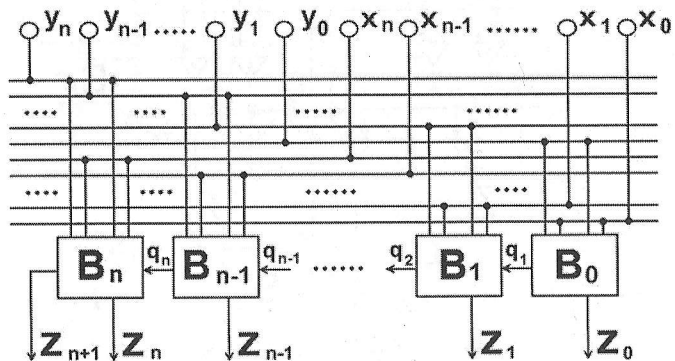


Рис. 23

Учитывая, что $L(B_0) = 4$, а $L(B_i) = 9$ для всех $i > 0$, получим значение сложности всей схемы Σ_n :

$$L(\Sigma_n) = 9n + 4.$$

Таким образом, сложность СФЭ, реализующих сложение чисел, растет линейно по отношению к числу входов, а значит кардинально отличается от схем, построенных при помощи способов, рассмотренных выше, где сложность растет экспоненциально.

Задачи для самостоятельного решения

Построить в виде СФЭ над B_0 устройство, производящее требуемое вычисление. Вычисления производятся в двоичной записи. При построении следует считать, что на входы x_1 и x_0 подается двузначное число a , на входы y_1 и y_0 подается двузначное число b . Для каждой задачи найти требуемое для вывода ответа число выходов схемы.

- | | |
|----------------------------------------------|----------------------------|
| 1. ab | 2. $2a + b$ |
| 3. $2a + 3b$ | 4. $3ab$ |
| 5. $a^2 + b^2$ | 6. a^b |
| 7. $ab^2 + ab + a$ | 8. C_a^b |
| 9. $a! + b!$ | 10. $a^b + b^a$ |
| 11. $2^a + b^2$ | 12. $\max\{a, b\}$ |
| 13. $\max\{2^a, b^2\}$ | 14. $ (a + b)! - 3^{a+b} $ |
| 15. $\min\{(a + b)!, 2^{a+b}, (a + b)^2\}$. | |

2 Конечные автоматы

2.1 Основные понятия, определения и задачи

Рассматривавшиеся в предыдущем разделе СФЭ являются дискретными преобразователями информации (ДПИ) без памяти. Это значит, что значения, которые получаются на выходе СФЭ зависят исключительно от того, что было подано на его входы в течение