Комбинационные схемы

Лабораторная работа 2: Комбинационные схемы

Цель работы: изучить основы булевой алгебры, методы синтеза комбинационных схем в потенциальной системе элементов; получить навыки работы с САПР БИС «Ковчег 3.04» в подсистемах графического редактора и функционально-логического моделирования; получить навыки в синтезе, наладке и экспериментальном исследовании синтезируемых схем.

2.1. Теоретические сведения

2.1.1. Введение в булеву алгебру

Логика — это наука о законах и формах мышления, математическая же логика занимается применением формальных математических методов для решения логических задач. В цифровой схемотехнике чрезвычайно широко используется простейший раздел математической логики — исчисление высказываний или алгебра логики. Алгебру логики часто называют ***булевой алгеброй*** в честь английского математика Джорджа Буля, который в 1847 году опубликовал краткую брошюру «Математический анализ логики, сопровождаемый наброском исчисления дедуктивных рассуждений», а в 1854 году вышел его основной труд «Исследование законов, на которых основаны математические теории логики и вероятностей».

Базовым понятием булевой алгебры является понятие ***высказывания***, под которым понимается любое утверждение, рассматриваемое только с точки зрения его истинности или ложности. В булевой алгебре не существует истинно-ложных или ложно-истинных высказываний. Высказывание можно рассматривать как логическую переменную, которая может принимать различные значения. Например, высказывание «сегодня понедельник» будет истинным в понедельник и ложным во все остальные дни недели. Исчисление высказываний как раз и основано на том, что их можно рассматривать как двоичные переменные, которые могут принимать одно из двух своих значений. Примерами двоичных логических переменных являются разряды чисел, представленных в двоичной системе счисления: замкнутый или разомкнутый контакт, наличие или отсутствие тока в цепи, высокий или низкий потенциал в какой-либо точке схемы и т.п.

Высказывание называется ***простым***, если значение его истинности не зависит от значений истинности других высказываний, и ***сложным***, если значение его истинности зависит от других высказываний. Сложное высказывание можно считать логической функцией, зависящей от простых высказываний и принимающей также два значения (истина, ложь). В свою очередь, сложные высказывания могут служить переменными (аргументами) ещё более сложных функций, то есть при построении логических функций справедлив ***принцип суперпозиции***.

2.1.1.1. Аксиомы булевой алгебры

Булеву алгебру (БА) как математическую структуру представляют совокупностью следующих объектов:

$$\text{БА} = \{0, 1, x_i, \text{И}, \text{ИЛИ}, \text{НЕ}, =\}, \qquad (2.1)$$

где 0 — символ, обозначающий абсолютную ложь (константа «0»),
1 — символ, обозначающий абсолютную истину (константа «1»).

Примечание: здесь 0 и 1 не цифры, а символы, обозначающие ложь и истину.

x_i — *i-я* логическая переменная, от которой зависит какая-либо логическая функция.

И — как минимум двухместная (то есть зависящая от двух переменных) логическая операция, определяемая как ***логическое произведение*** (другое название — ***конъюнкция***). Это такое сложное высказывание, которое истинно только в том случае, когда истинны высказывания, от которых оно зависит. В остальных случаях оно ложно.

ИЛИ — как минимум двухместная логическая операция, определяемая как ***логическая сумма*** (другое название — ***дизъюнкция***). Это такое сложное высказывание, которое ложно только в том случае, когда ложны высказывания, от которых оно зависит. В остальных случаях оно истинно.

НЕ — одноместная логическая операция, определяемая как ***логическое отрицание*** (другое название — ***инверсия***).

= — отношение эквивалентности.

Объекты (2.1) булевой алгебры определяются следующими аксиомами:

$x = 0$, если x не равно 1;

$x = 1$, если x не равно 0. (2.2)

Аксиома (2.2) утверждает, что в булевой алгебре рассматриваются только двоичные переменные (закон исключённого третьего).

$$\left.\begin{array}{ll} & 0\cdot 0=0 \\ а) & 0\cdot 1=1\cdot 0=0 \\ & 1\cdot 1=1 \end{array} \qquad \begin{array}{ll} & 1+1=1 \\ б) & 1+0=0+1=1 \\ & 0+0=0 \end{array}\right\} \tag{2.3}$$

Аксиомы (2.3, *а*) определяют логическую операцию И. В качестве знака операции И, кроме точки, используются следующие знаки: ×, отсутствие знака, &, ∧, П.

Аксиомы (2.3, *б*) определяют логическую операцию ИЛИ. В качестве знака операции ИЛИ, кроме +, используются знаки: ∨, Σ.

$$\left.\begin{array}{l} \bar{0}=1 \\ \bar{1}=0 \end{array}\right\} \tag{2.4}$$

Аксиома (2.4) определяет операцию логического отрицания.

Отношение эквивалентности удовлетворяет следующим свойствам:

- рефлексивности: $x = x$,
- симметричности: если $x_1 = x_2$, то $x_2 = x_1$,
- транзитивности: если $x_1 = x_2$ и $x_2 = x_3$, то $x_1 = x_3$.

Из свойств отношения эквивалентности следует ***принцип подстановки:*** если $x_1 = x_2$, то в любом логическом выражении, содержащем x_1, вместо него можно подставить x_2. В результате будет получено эквивалентное выражение.

2.1.1.2. Основные законы булевой алгебры

С помощью аксиом булевой алгебры можно доказать целый ряд законов методом перебора всех значений переменных. Если закон истинен, то с учётом аксиом (2.2–2.4) при подстановке любых значений переменных в обе части выражения, формулирующего закон, должно получиться тождество.

Основные законы алгебры логики принято разбивать на три группы.

1. Законы одинарных элементов:

а) законы универсального множества:

$$\left.\begin{aligned} x \cdot 1 &= x \\ x + 1 &= 1 \end{aligned}\right\} \tag{2.5}$$

б) законы нулевого множества:

$$\left.\begin{aligned} x \cdot 0 &= 0 \\ x + 0 &= x \end{aligned}\right\} \tag{2.6}$$

2. Законы отрицания:

а) закон двойного отрицания:

$$\overline{\overline{x}} = x \tag{2.7}$$

б) законы дополнительности:

$$\left.\begin{aligned} x \cdot \overline{x} &= 0 \\ x + \overline{x} &= 1 \end{aligned}\right\} \tag{2.8}$$

в) законы двойственности (де Моргана):

$$\left.\begin{aligned} \overline{x_1 x_0} &= \overline{x}_1 + \overline{x}_0 \\ \overline{x_1 + x_0} &= \overline{x}_1 \overline{x}_0 \end{aligned}\right\} \tag{2.9}$$

3. Комбинационные законы:

а) законы тавтологии (идемпотентности):

$$\left.\begin{aligned} x \cdot x \cdot x \cdot \ldots \cdot x &= x \\ x + x + \ldots + x &= x \end{aligned}\right\} \tag{2.10}$$

Отсюда следует, что булева алгебра является алгеброй без степеней и коэффициентов;

б) переместительные законы (коммутативные):

$$\left.\begin{aligned} x_1 x_0 &= x_0 x_1 \\ x_1 + x_0 &= x_0 + x_1 \end{aligned}\right\} \tag{2.11}$$

в) сочетательные законы (ассоциативные):

$$\left.\begin{aligned} x_2 x_1 x_0 &= (x_2 x_1) x_0 = (x_2 x_0) x_1 + (x_1 x_0) x_2 \\ x_2 + x_1 + x_0 &= (x_2 + x_1) + x_0 = (x_2 + x_0) + x_1 = (x_1 + x)_0 + x_2 \end{aligned}\right\} \tag{2.12}$$

г) распределительные законы (дистрибутивные):

первого рода — умножение относительно сложения:

$$x_2 (x_1 + x_0) = x_2 x_1 + x_2 x_0 \tag{2.13}$$

второго рода — сложение относительно умножения:

$$x_2 + x_1 x_0 = (x_2 + x_1)(x_2 + x_0) \tag{2.14}$$

В обычной алгебре не действуют законы тавтологии и распределительный закон второго рода. Обратите внимание, что аксиомы (2.3) записаны с учётом закона двойственности. Если в любой строке одного из столбцов произвести взаимную замену 0 на 1 и операций И и ИЛИ, то получим аксиому в этой же строке другого столбца.

2.1.1.3. Следствия из основных законов булевой алгебры

Прежде чем формулировать важнейшие следствия из основных законов булевой алгебры, рассмотрим некоторые новые понятия. Совокупность конкретных значений логических переменных, от которых зависит булева функция, называется ***набором логических переменных***. Если булева функция зависит, например, от трёх переменных x_1, x_2 и x_0, тогда $x_2 = 0, x_1 = 1, x_0 = 1$ является набором. Наборы могут быть представлены различными способами. Указанный выше набор можно представить как $\bar{x}_2, x_1$ и x_0, где знак инверсии говорит о том, что $x_2 = 0$, а отсутствие знака инверсии $x_1 = x_0 = 1$. Тот же набор можно представить как 0,1,1 или просто 011. Рассматривая последнее представление как двоичное число, можно записать его в виде десятичного эквивалента $3 = 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$. Отсюда видно, что логическим переменным, как и разрядам двоичного числа, можно ***условно*** присвоить арифметические «веса»: для x_0 вес будет $2^0 = 1$, для x_1 – $2^1 = 2$, для x_2 – $2^2 = 4$ и т.д. Именно поэтому логические переменные удобно обозначать индексированными буквами, начиная с индекса 0 для младшей переменной, 1 для следующей переменной и т.д. до индекса $n - 1$, если функция зависит от n переменных. При таком обозначении индекс переменной совпадает с показателем степени основания двоичной системы счисления, т.е. характеризует вес переменной. Так, для переменной x_5 вес будет равен $2^5 = 32$.

Если функция алгебры логики зависит от n переменных, то всего для них существует 2^n наборов, так как добавление одной переменной увеличивает число наборов в два раза. Одна переменная имеет два набора: 0 и 1, две переменные — четыре набора: 0 = 00, 1 = 01, 2 = 10, 3 = 11 и т.д. Десятичный эквивалент набора логических переменных называется ***номером набора***. Наборы можно рассматривать как ***двоичные векторы*** X_i, где i — номер набора, однако надо помнить, что они не являются векторами в классическом смысле, так как над ними нельзя выполнять векторные операции.

Наборы можно представить в виде вершин n-мерного куба. Это представление здесь рассматриваться не будет.

Число переменных, имеющих в наборе значение 1, называется ***весом набора*** (не путать с двоичным весом переменных). Вес набора удобно изображать римскими цифрами: так, для $n = 3$ имеем: набор 000 с весом 0, наборы 001, 010, 100 с весом I, наборы 011, 101, 110 с весом II и набор 111 с весом III. Двум любым наборам, в состав которых входят n переменных, ставится в соответствие целое число, которое называется ***расстоянием по Хеммингу***. Это число совпадает с числом переменных, входящих в наборы различным образом. Расстояние обозначается $d(X_i, X_j)$. Так, для $n = 4$ имеем $d(X_1, X_{13}) = d(0001, 1101) = 2$, так как только переменные x_3 и x_2 входят в наборы

различным образом. Если $d(X_i, X_j) = 1$, то наборы называются ***соседними***. Вес набора равен расстоянию по Хеммингу от нулевого набора.

Расстояние по Хеммингу удовлетворяет следующим условиям:

$d(X_i, X_j) \geq 0$,

$d(X_i, X_j) \geq 0 = 0$ тогда и только тогда, когда $X_i = X_j$,

$d(X_i, X_j) = d(X_j, X_i)$,

$d(X_i, X_j) + d(X_j, X_k) \geq d(X_i, X_k)$.

Полезно помнить, что $d(X_i, 00...0) = n - d(X_i, 11...1)$, где n — число переменных.

Если наборы рассматривать как их десятичные эквиваленты, то для любых двух наборов можно ввести естественную или лексикографическую упорядоченность или ***отношение порядка*** (≤).

Аксиомы отношения порядка:

рефлексивность: $X_i \leq X_i$,

антисимметричность: если $X_i \leq X_j$ и $X_j \leq X_i$, то $X_i = X_j$,

транзитивность: если $X_i \leq X_j$, а $X_j \leq X_k$, то $X_i \leq X_k$.

Наиболее простым примером использования отношения полного линейного порядка является естественное расположение наборов переменных в таблицах истинности функций алгебры логики.

Рассмотрим два набора переменных: $X^{/} = (x^{/}_{n-1},\ldots,x^{/}_i,\ldots,x^{/}_0)$ и $X^{//} = (x^{//}_{n-1},\ldots,x^{//}_i,\ldots,x^{//}_0)$, где $x^{/}_i$ и $x^{//}_i$ — это одна и та же переменная x_i, входящая соответственно в наборы $X^{/}$ и $X^{//}$, такие, что удовлетворяется неравенство $X^{/}_i \leq X^{//}_i$ для всех i, т.е. $x^{/}_0 \leq x^{//}_0, x^{/}_1 \leq x^{//}_1, x^{/}_2 \leq x^{//}_2$ и т.д. Тогда говорят, что выполнено ***отношение предшествования*** $X^{/}_i \leq X^{//}_i$. Например, для $n = 3$: $010 \leq 110$.

Не все пары наборов находятся в отношении предшествования, например 010 и 101, наборы одного веса и др. Поэтому наборы в отношении предшествования являются лишь частично упорядоченными. Наборы, для которых отношение предшествования не выполняется, называются ***несравнимыми***. Отношение предшествования используется для определения класса монотонных булевых функций.

Логическое произведение любого числа переменных из конечного набора n переменных называется ***элементарным***, когда сомножителями в нем являются либо переменные, либо их отрицания. Например, $x_3\bar{x}_2\bar{x}_1x_0$ является элементарным, а $(\bar{x}_3 + x_2)\bar{x}_0, x_3\overline{\bar{x}_2x_1}x_0$ — нет.

Количество сомножителей в элементарном произведении называется его ***рангом r***. Так, для $x_3\bar{x}_2\bar{x}_1x_0$ $r = 4$, а для $x_3\bar{x}_0$ $r = 2$. Логическое произведение, являющееся функцией всех n переменных, называется ***конституентой единицы*** (составляющей единицы). Смысл этого термина будет пояснён позже. Для n переменных существует 2^n конституент единицы, причём на данном конкретном наборе лишь одна конституента единицы будет равна 1, все другие будут равны 0.

Два элементарных произведения одинакового ранга r называются ***соседними***, если они являются функциями одних и тех же переменных и отличаются только знаком инверсии лишь у одной переменной. Например, элементарные произведения $x_2\bar{x}_1\bar{x}_0$ и $x_2\bar{x}_1x_0$ соседние, а $x_2\bar{x}_1\bar{x}_0$ и $\bar{x}_2\bar{x}_1x_0$ — нет.

Логическая сумма любого числа переменных их конечного набора *n* переменных называется ***элементарной***, когда слагаемыми в ней являются либо переменные, либо их отрицания. Например, сумма $\bar{x}_3 + x_2 + x_1 + \bar{x}_0$ элементарная, а суммы $\bar{x}_3 x_2 + x_0, x_3 + \overline{\bar{x}_2 x_1} + x_0$ и $x_3 + \overline{x_2 + \bar{x}_1} + x_0$ — нет.

Количество слагаемых в элементарной сумме называется её ***рангом r***. Так, для $\bar{x}_3 + x_2 + x_1 + \bar{x}_0$ $r = 4$, а для $x_3 + \bar{x}_0$ $r = 2$. Логическая сумма, являющаяся функцией всех *n* переменных, называется ***конституентой нуля*** (составляющей нуля). Смысл этого термина будет пояснён позже. Для *n* переменных существует 2^n конституент нуля, причём на данном конкретном наборе лишь одна конституента нуля будет равна 0, все остальные будут равны 1.

Две элементарные суммы одинакового ранга *r* называются ***соседними***, если они являются функциями одних и тех же переменных и отличаются только знаком инверсии лишь у одной переменной. Например, суммы $\bar{x}_2 + x_1 + \bar{x}_0$ и $x_2 + x_1 + \bar{x}_0$ являются соседними, а $\bar{x}_2 + x_1 + \bar{x}_0$ и $\bar{x}_2 + \bar{x}_1 + x_0$ — нет.

Сформулируем теперь важнейшие следствия из основных законов булевой алгебры, представив их в виде следующих правил.

Правило старшинства операций

Пусть требуется определить значение истинности функции $y = \bar{x}_3 x_2 + x_3 \overline{x_1 + \bar{x}_0}$ на наборе 11 (одиннадцать). Представив десятичное число 11 в виде двоичного числа 1011, мы определяем, что $x_3 = 1;\ x_2 = 0;\ x_1 = x_0 = 1$. Возникает вопрос: в каком порядке следует выполнять различные логические операции в рассматриваемой функции? Из законов инверсии вытекает, что старшей (первой) операцией является операция отрицания. Это значит, что операции логического умножения и сложения нельзя производить, игнорируя знак отрицания над переменными, т.е. операцию отрицания следует выполнять в первую очередь.

Операции логического умножения и сложения, в силу симметричности законов булевой алгебры, равноправны, однако на практике принято считать старшей операцию логического умножения, что соответствует правилам, к которым мы привыкли в обычной алгебре.

Итак, если в логическом выражении встречаются все три операции И, ИЛИ, НЕ, то сначала выполняется операция НЕ над отдельными переменными, затем операция И и в конце ИЛИ. Действия под групповым знаком инверсии выполняются по этому же правилу. Отклонение от этого правила должно быть обозначено скобками. В рассматриваемом примере

$$y = \bar{x}_3 x_2 + x_3 \overline{x_1 + \bar{x}_0} = \bar{1} \cdot 0 + 1 \cdot \overline{1 + \bar{1}} = 0 \cdot 0 + 1 \cdot \overline{1 + 0} = 0 + 1 \cdot \bar{1} =$$

$$= 0 + 1 \cdot 0 = 0 + 0 = 0.$$

Правило склеивания

Правило склеивания для суммы элементарных произведений следует из распределительного закона первого рода (2.13), закона дополнительности (2.8) и закона универсального множества (2.5): ***логическую сумму двух соседних произведений ранга r можно заменить одним элементарным произведением ранга r – 1, являющимся общей частью исходных слагаемых.***

Пример: $y = x_2 \bar{x}_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 = x_2 \bar{x}_1 (\bar{x}_0 + x_0) = x_2 \bar{x}_1 \cdot 1 = x_2 \bar{x}_1$.

Правило склеивания для произведения элементарных сумм следует из распределительного закона второго рода (2.14), закона дополнительности (2.8) и за-

кона нулевого множества (2.6): ***логическое произведение двух соседних сумм некоторого ранга r можно заменить одной элементарной суммой ранга r − 1, являющейся общей частью исходных сомножителей.***

Пример: $y=(\bar{x}_2+x_1+\bar{x}_0)(x_2+x_1+\bar{x}_0)=(x_1+\bar{x}_0)+\bar{x}_2x_2=(x+\bar{x}_0)+0=x_1+\bar{x}_0$.

Правило поглощения

Правило поглощения для суммы двух элементарных произведений следует из распределительного закона первого рода (2.13) и законов универсального множества (2.5): ***логическую сумму двух элементарных произведений разных рангов, из которых одно является составной частью другого, можно заменить произведением, имеющим меньший ранг.***

Пример: $y=x_3\bar{x}_1+x_3\bar{x}_2\bar{x}_1x_0=x_3\bar{x}_1(1+\bar{x}_2x_0)=x_3\bar{x}_1\cdot1=x_3\bar{x}_1$.

Правило поглощения для произведения двух элементарных сумм следует из распределительного закона второго рода (2.14) и законов нулевого множества (2.6): ***логическое произведение двух элементарных сумм разных рангов, из которых одна является составной частью другой, можно заменить элементарной суммой, имеющей меньший ранг***.

Пример: $y=(\bar{x}_3+\bar{x}_1)(\bar{x}_3+\bar{x}_2+\bar{x}_1+x_0)=(\bar{x}_3+\bar{x}_1+0)(\bar{x}_3+\bar{x}_2+\bar{x}_1+x_0)=$
$=(\bar{x}_3+\bar{x}_1)+0(\bar{x}_2+x_0)=(\bar{x}_3+\bar{x}_1)+0=\bar{x}_3+\bar{x}_1=\bar{x}_3+\bar{x}_1$.

Правило развёртывания

Это правило определяет действия, обратные склеиванию.

Правило развёртывания элементарного произведения в логическую сумму элементарных произведений большего ранга (в пределе до $r=n$, т.е. до конституент единицы, как и будет рассмотрено ниже) следует из законов универсального множества (2.5), распределительного закона первого рода (2.13) и производится в три этапа:

- в развёртываемое элементарное произведение ранга r вводится в качестве сомножителей $n-r$ единиц, где n — ранг конституенты единицы;
- каждая единица заменяется логической суммой некоторой не имеющейся в исходном элементарном произведении переменной и её отрицания: $x_i+\bar{x}_i=1$;
- производится раскрытие всех скобок на основе распределительного закона первого рода (2.13), что приводит к развёртыванию исходного элементарного произведения ранга r в логическую сумму 2^{n-r} конституент единицы.

Пример: требуется развернуть элементарное произведение $x_3\bar{x}_1$ в логическую сумму конституент единицы, зависящих от четырёх переменных. Последнее следует из того, что максимальный индекс у переменной равен 3, отсутствуют переменные x_2 и x_0.

Решение: $x_3\bar{x}_1=x_3\cdot1\cdot\bar{x}_1\cdot1=x_3(x_2+\bar{x}_2)\bar{x}_1(x_0+\bar{x}_0)=(x_3x_2\bar{x}_1+x_3\bar{x}_2\bar{x}_1)(x_0+\bar{x}_0)=$
$=x_3x_2\bar{x}_1x_0+x_3\bar{x}_2\bar{x}_1x_0+x_3x_2\bar{x}_1\bar{x}_0+x_3\bar{x}_2\bar{x}_1\bar{x}_0$.

Пусть $n=3$. Из преобразования (развёртывания) $1=1\cdot1\cdot1=$
$=(x_2+\bar{x}_2)(x_1+\bar{x}_1)(x_0+\bar{x}_0)=$

$$\bar{x}_2\bar{x}_1\bar{x}_0+\bar{x}_2\bar{x}_1x_0+\bar{x}_2x_1\bar{x}_0+\bar{x}_2x_1x_0+x_2\bar{x}_1\bar{x}_0+x_2\bar{x}_1x_0+x_2x_1\bar{x}_0+x_2x_1x_0$$

следует смысл термина «**конституента (**составляющая**) единицы**».

Правило развёртывания элементарного произведения используется для минимизации функций алгебры логики (ФАЛ).

Пример: пусть $y = x_2\bar{x}_1 + x_1x_0 + x_2x_0$. Видно, что все элементарные произведения имеют ранг $r = 2$, следовательно, правило поглощения нельзя применить; кроме того ни одна пара произведений не является соседней, так как произведения имеют различные переменные. Если же развернуть произведение x_2x_0 до конституент единицы (в данном случае $n = 3$), то выражение упростится:

$$y = x_2\bar{x}_1 + x_1x_0 + x_2 \cdot 1 \cdot x_0 =$$

$$= x_2\bar{x}_1 + x_1x_0 + x_2(x_1 + \bar{x}_1)x_0 = x_2\bar{x}_1 + x_1x_0 + x_2x_1x_0 + x_2\bar{x}_1x_0 =$$

$$= x_2\bar{x}_1(1 + x_0) + x_1x_0(1 + x_2) = x_2\bar{x}_1 \cdot 1 + x_1x_0 \cdot 1 = x_2\bar{x}_1 + x_1x_0,$$

т.е. произведение x_2x_0 оказалось лишним. Общие формальные правила определения лишних произведений будут рассмотрены в последующих учебно-методических пособиях.

Правило развёртывания элементарной суммы ранга r до произведения элементарных сумм ранга n (конституент нуля) следует из законов нулевого множества (2.6) и распределительного закона второго рода (2.14) и производится в три этапа:

- в развёртываемую сумму ранга r в качестве слагаемых вводится $n - r$ нулей;
- каждый нуль представляется в виде логического произведения некоторой не имеющейся в исходной сумме переменной и её отрицания: $x_i\bar{x}_i = 0$;
- получившееся выражение преобразуется на основе распределительного закона второго рода (2.14) таким образом, чтобы исходная сумма ранга r развернулась в логическое произведение 2^{n-r} конституент нуля.

Пример: развернуть элементарную сумму $x_3 + \bar{x}_2$ в логическое произведение конституент нуля, зависящих от четырёх переменных. Последнее следует из того, что максимальный индекс равен 3, отсутствуют переменные x_1 и x_0.

Решение:

$$x_3 + \bar{x}_2 = x_3 + \bar{x}_2 + 0 + 0 = x_3 + \bar{x}_2 + x_1\bar{x}_1 + x_0\bar{x}_0 = (x_3 + \bar{x}_2 + x_1)(x_3 + \bar{x}_2 + \bar{x}_1) + x_0\bar{x}_0 =$$

$$= (x_3 + \bar{x}_2 + x_1 + x_0)(x_3 + \bar{x}_2 + x_1 + \bar{x}_0)(x_3 + \bar{x}_2 + \bar{x}_1 + x_0)(x_3 + \bar{x}_2 + \bar{x}_1 + \bar{x}_0).$$

Пусть $n = 2$. Из преобразования (развёртывания) $0 = 0 + 0 = x_1\bar{x}_1 + x_0\bar{x}_0 = (x_1\bar{x}_1 + x_0)(x_1\bar{x}_1 + \bar{x}_0) = (x_1 + x_0)(x_1 + \bar{x}_0)(\bar{x}_1 + x_0)(\bar{x}_1 + \bar{x}_0)$ следует смысл термина «**конституента** (составляющая) **нуля**».

Правило развёртывания элементарной суммы также используется для минимизации ФАЛ.

Пример: пусть $y = (x_2 + \bar{x}_1)(x_1 + x_0)(x_2 + x_0)$. Ясно, что операции склеивания и поглощения здесь применить нельзя. Однако, если развернуть сумму $x_2 + x_0$ до конституент нуля (в данном случае $n = 3$), то выражение упростится:

$$y = (x_2 + \bar{x}_1)(x_1 + x_0)(x_2 + 0 + x_0) = (x_2 + \bar{x}_1)(x_1 + x_0)(x_2 + x_1\bar{x}_1 + x_0) =$$

$$= (x_2 + \bar{x}_1 + 0)(x_1 + x_0 + 0)(x_2 + x_1 + x_0)(x_2 + \bar{x}_1 + x_0) = (x_2 + \bar{x}_1 + 0 \cdot x_0)(x_1 + x_0 + 0 \cdot x_2) =$$

$= (x_2 + \bar{x}_1)(x_1 + x_0)$, т.е. сумма $x_2 + x_0$ оказалась лишней.

Правила склеивания, поглощения и развёртывания лежат в основе различных методов минимизации ФАЛ.

2.1.2. Формы представления и классификация функций алгебры логики

Функцией алгебры логики (ФАЛ) называется функция, которая, как и её аргументы, может принимать только два значения: 0 и 1. ФАЛ могут быть представлены различными способами:

1. Словесным описанием (назначением, определением).
2. Таблицей истинности.
3. Аналитическим выражением.
4. Картой Карно.
5. Переключательной схемой.
6. Диаграммой Венна.
7. Геометрическим способом (гиперкубами).
8. Диаграммой двоичного решения.
9. Временной диаграммой.

Если ФАЛ зависит от ***n*** переменных, то она может быть определена на $N = 2^n$ наборах. Когда функция определена на всех этих наборах, она называется ***полностью определённой***. Число различных полностью определённых ФАЛ равно $F = 2^N = 2^{2^n}$. Если на некоторых наборах ФАЛ не определена, то она называется ***не полностью определённой***. В последнем случае возможны две ситуации: либо нас не интересует значение функции на некоторых наборах, либо на цифровое устройство никогда не поступают некоторые наборы. Имеется три возможности задать не полностью определённую ФАЛ на любом из *N* наборах: выбрать её значение равным 0, 1 или безразличным значением. Если исключить случай, когда ФАЛ не определена ни на каком наборе, то общее число возможных способов задания не полностью определённой ФАЛ от ***n*** переменных будет равно $3^{2^n} - 1$.

Рассмотрим все перечисленные выше формы представления для полностью определённой ФАЛ, зависящей от трёх переменных $y = f(x_2; x_1; x_0)$, ***словесное описание*** которой формулируется следующим образом: функция принимает истинное значение, когда либо истинна переменная x_2, либо вместе истинны переменные x_1 и x_0.

Таблица истинности (см. табл. 2.1) для ФАЛ имеет следующую структуру: в столбце «номер набора» указаны десятичные эквиваленты двоичных 3-разрядных чисел, условно составленных из значений трёх аргументов x_2, x_1 и x_0 с учётом соответственно их двоичного веса 4, 2 и 1, то есть наборы расположены в лексикографической упорядоченности. Чтобы быстро и без ошибок заполнить все 2^n строк с различными наборами, рекомендуется заполнять их по столбцам. Так как переменная x_0 имеет вес 1, то в соответствующем столбце сверху вниз записывается последовательность 0101...; x_1 имеет вес 2, поэтому в соответствующем столбце записывается последовательность 0011...; для x_2 — 00001111... и т. д. В столбце *y* записываются значения ФАЛ на каждом наборе. Рассматриваемая ФАЛ принимает значение 0 на первых трёх наборах и значение 1 на остальных пяти.

Представление ФАЛ в виде таблицы истинности удобно и наглядно для $n \le (4–6)$. При больших ***n*** таблицы становятся громоздкими и трудно обозримыми, что является их основным недостатком.

Таблица 2.1. Таблица истинности произвольной ФАЛ

Номер набора	x_2	x_1	x_0	y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Аналитический способ задания ФАЛ основывается на использовании базовых логических операций И, ИЛИ, НЕ, определяемых аксиомами булевой алгебры. Ограничимся рассмотрением только двух канонических (основных) аналитических выражений ФАЛ по значениям истинности и по значениям ложности. Если ФАЛ задаётся по значениям истинности, то говорят, что аналитическое выражение записано в ***совершенной дизъюнктивной нормальной форме*** (СДНФ). ***Нормальной*** эта форма называется потому, что все члены выражения имеют вид элементарных произведений. ***Дизъюнктивной*** эта форма называется потому, что все они соединены знаком дизъюнкции, а ***совершенной*** потому, что все члены формулы имеют высший ранг $r = n$, являясь конституентами единицы.

Для того чтобы получить аналитическое выражение для ФАЛ, заданной таблицей истинности, в СДНФ, нужно записать логическую сумму конституент единицы для тех наборов входных переменных, на которых ФАЛ принимает единичное значение, причём переменная берётся без знака инверсии, если ее значение в наборе равно 1, и с инверсией, если ее значение в наборе равно 0.

СДНФ ФАЛ, заданной табл. 2.1, имеет следующий вид:

$$y = \overline{x_2}x_1x_0 + x_2\overline{x_1}\,\overline{x_0} + x_2\overline{x_1}x_0 + x_2x_1\overline{x_0} + x_2x_1x_0. \qquad (2.15)$$

Чтобы сделать запись ФАЛ более компактной, можно заменить конституенты единицы номерами соответствующих наборов:

$$y = 3 + 4 + 5 + 6 + 7, \qquad (2.16)$$

либо использовать следующую запись:

$$y = \mathrm{V}(3, 4, 5, 6, 7). \qquad (2.17)$$

Если ФАЛ задаётся по значениям ложности, то говорят, что аналитическое выражение записано в ***совершенной конъюнктивной нормальной форме*** (СКНФ). ***Нормальной*** эта форма называется потому, что члены выражения имеют вид элементарных сумм. ***Конъюнктивной*** эта форма называется потому, что все члены формулы соединены знаком конъюнкции, а ***совершенной*** потому, что все члены формулы имеют высший ранг $r = n$, являясь конституентами нуля.

Для того чтобы получить аналитическое выражение для ФАЛ, заданной таблицей истинности, в СКНФ, нужно записать логическое произведение конституент нуля для тех наборов входных переменных, на которых ФАЛ принимает

нулевое значение, причём переменная берётся со знаком инверсии, если её значение в наборе равно 1, и без знака инверсии, если её значение в наборе равно 0.

СКНФ ФАЛ, заданной табл.2.1, имеет следующий вид:

$$y = (x_2 + x_1 + x_0)(x_2 + x_1 + \overline{x_0})(x_2 + \overline{x_1} + x_0), \tag{2.18}$$

$$y = 0 \cdot 1 \cdot 2, \tag{2.19}$$

$$y = \Lambda(0, 1, 2). \tag{2.20}$$

Для любой ФАЛ СДНФ и СКНФ единственны. Так как в СДНФ и СКНФ можно представить любую ФАЛ, а в выражениях (2.15) и (2.18) используются три логические операции И, ИЛИ и НЕ, то говорят, что набор этих операций образует функционально полную систему (ФПС), позволяющую реализовать произвольные ФАЛ. Совокупность И, ИЛИ и НЕ называют ***основной функционально полной системой*** (ОФПС).

Если к выражению (2.15) применить закон двойного отрицания и правило де-Моргана, то получим следующее выражение:

$$\begin{aligned} y &= \overline{\overline{x_2 x_1 x_0} + x_2 \overline{x_1}\,\overline{x_0} + x_2 \overline{x_1} x_0 + x_2 x_1 \overline{x_0} + x_2 x_1 x_0} = \\ &= \overline{\overline{x_2 x_1 x_0} \cdot \overline{x_2 \overline{x_1}\,\overline{x_0}} \cdot \overline{x_2 \overline{x_1} x_0} \cdot \overline{x_2 x_1 \overline{x_0}} \cdot \overline{x_2 x_1 x_0}}, \end{aligned} \tag{2.21}$$

из которого видно, что для представления любой ФАЛ достаточно использовать только две операции И и НЕ, которые тоже являются ОФПС.

Если к выражению (2.18) применить закон двойного отрицания и правило де-Моргана, то получим следующее выражение:

$$\begin{aligned} y &= (x_2 + x_1 + x_0)(x_2 + x_1 + \overline{x_0})(x_2 + \overline{x_1} + x_0) = \\ &= \overline{\overline{x_2 + x_1 + x_0} + \overline{x_2 + x_1 + \overline{x_0}} + \overline{x_2 + \overline{x_1} + x_0}}, \end{aligned} \tag{2.22}$$

из которого видно, что для представления любой ФАЛ достаточно использовать только две операции ИЛИ и НЕ, которые тоже являются ОФПС.

Применяя операции склеивания, поглощения и развёртывания, можно упростить выражения (2.15) и (2.18), однако вопросы минимизации ФАЛ будут рассмотрены не здесь, а в разделе 2.3.

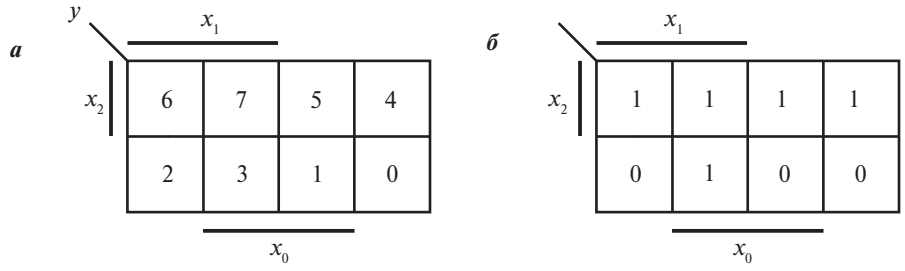

Рис. 2.1. Карты Карно – эталонная (*а*) и рабочая (*б*)

Карта Карно является специальной компактной формой таблицы истинности, которая позволяет не только представить ФАЛ, но и минимизировать её. На рис. 2.1 показана карта Карно для ФАЛ, заданной табл. 2.1. Структуры карт Карно для $n = 1 – 6$ и методы использования их для минимизации ФАЛ будут подробно рассмотрены в разделе 2.3.2.

Переключательная схема может рассматриваться как техническая модель логических выражений. Одну из первых моделей предложил в 1910 году физик П.С. Эренфест, в ней использована аналогия между высказываниями и электрическими переключателями, поскольку и те и другие имеют двоичную природу. На рис. 2.2 приведены модели для констант 0 и 1 и базовых функций И, ИЛИ, НЕ. Положение переключателей, указанное на рис. 2.2, соответствует нулевому значению переменной. Переключательная схема для ФАЛ, заданной табл. 2.1, приведена на рис. 2.3 в минимизированном варианте.

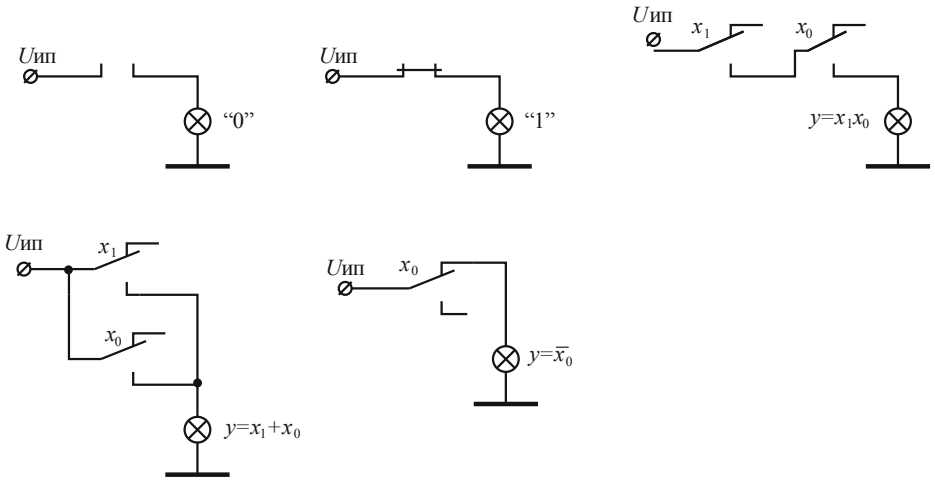

Рис. 2.2. Модели для констант 0 и 1 базовых функций И, ИЛИ, НЕ

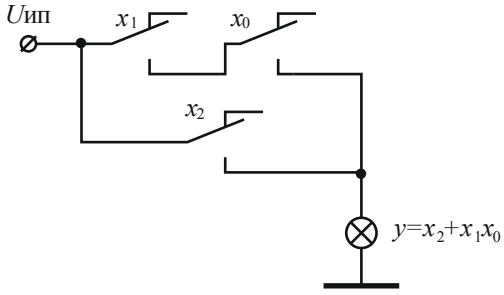

Рис. 2.3. Переключательная модель ФАЛ, заданной в таблице 2.1

Диаграммы Венна, названные по имени священника Джона Венна (1834–1923), применявшего их в исследованиях по логике, являются графическим представлением, демонстрирующим соотношения между множествами. На основе соответствия между теорией булевой алгебры и теорией множеств диаграммы Венна очень полезны для визуального представления аксиом и законов булевой алгебры, однако их не следует использовать для построения доказательств тождеств, поскольку большинство общих положений эти диаграммы показать не могут.

На рис. 2.4 приведены диаграммы Венна для констант 0, 1 и базовых функций И, ИЛИ, НЕ, где область, ограниченная кружком, соответствует одной переменной. На рис. 2.5 приведена диаграмма Венна для ФАЛ, заданной табл. 2.1. Диаграммы Венна удобны только для $n \le 3$.

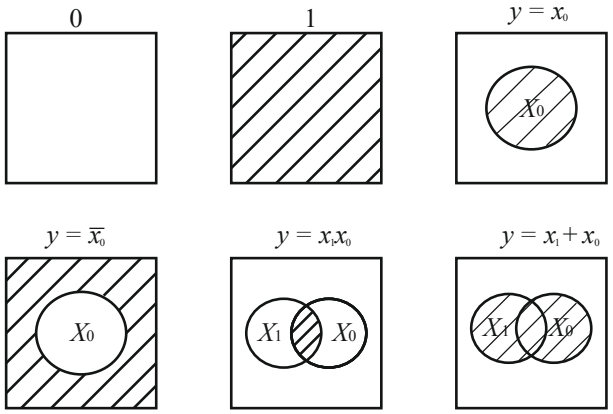

Рис. 2.4. Диаграммы Венна

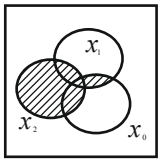

Рис. 2.5. Диаграмма Венна для $y = x_2 + x_1x_0$

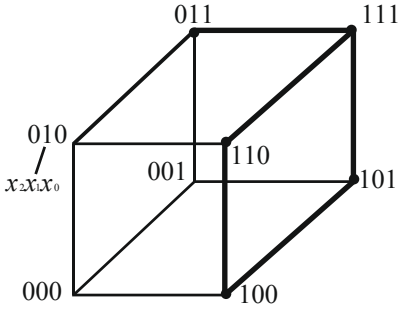

Рис. 2.6. Геометрическое представление для $y = x_2 + x_1x_0$

Геометрическое представление булевой функции получается путём отображения булевой функции n переменных на n-мерный куб. Для отображения булевой функции n переменных на n-кубе устанавливается соответствие между членами СДНФ и вершинами n-куба. Вершины (наборы), на которых функция принимает единичное значение, выделяются жирными точками. На рис. 2.6 представлена в виде куба ФАЛ, заданная табл. 2.1. Геометрическое представление удобно только для $n \le 3$. Для $n = 4$ геометрическое представление уже довольно сложное, поэтому для $n \ge 4$ используют аналитическое представление n-кубов [5].

Диаграмма двоичного решения (ДДР) является разновидностью корневого ориентированного графа, обеспечивающая полное, краткое и простое описание сложных цифровых функций. На рис. 2.7 приведена ДДР ФАЛ, представленной табл. 2.1. На рис. 2.7 прямоугольники с цифрами 0 и 1 соответствуют финишным

(окончательным) значениям ФАЛ. Узел, обозначенный кружком, соответствует переменной, от которой зависит ФАЛ, а цифры у ветвей — значениям этих переменных. ДДР широко используются для анализа сложных функций, формирования тестов, задач верификации и т.д. [7], [8].

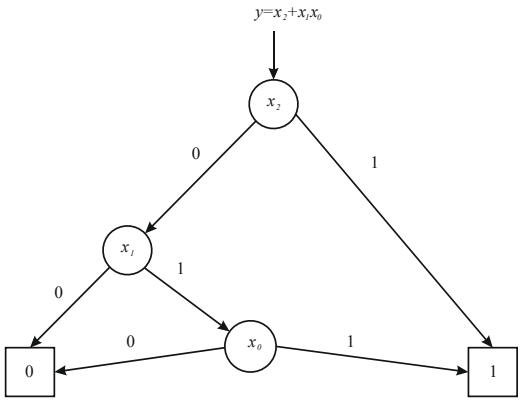

Рис. 2.7. Диаграмма двоичного решения для $y = x_2 + x_1x_0$

2.1.3. Классификация функций алгебры логики

Прежде чем формулировать свойства различных видов ФАЛ, представим в виде обобщенной таблицы истинности полный набор ФАЛ, зависящих от двух переменных. Обоснованием этого можно рассматривать следующие моменты: общее число таких функций невелико ($F = 2^{2^2} = 16$); все эти функции находят широкое практическое применение, обращение к таблице позволит на простых примерах осмыслить особенности различных ФАЛ, подсчитать их количество и т.п.

Таблица 2.2. Полный набор ФАЛ, зависящих от двух переменных

Номер набора	x_1	x_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

В табл. 2.2 индекс ***i*** функции y_i совпадает с десятичным эквивалентом двоичного числа, условно составленного из значений функций на четырёх наборах. Все 16 функций являются полностью определёнными. Запишем аналитические выражения для всех этих функций в СДНФ для случая, когда число единичных значений функции меньше или равно двум (то есть половине числа всех наборов), и в СКНФ, если функция имеет один нуль, с последующей минимизацией и с использованием правила склеивания.

$y_0 = 0$ константа «0»

$y_1 = x_1\underline{x_0}$ операция И

$y_2 = x_1 x_0$ запрет переменной x_1

$y_3 = x_1\overline{x_0} + x_1x_0 = x_1$	повторение переменной x_1
$y_4 = x_1x_0$	запрет переменной x_0
$y_5 = \overline{x_1}x_0 + x_1\overline{x_0} = x_0$	повторение переменной x_0
$y_6 = \overline{x_1}x_0 + x_1\overline{x_0} = x_1 \oplus x_0$	неравнозначность или сумма по mod2
$y_7 = x_1 + x_0$	операция ИЛИ
$y_8 = \overline{x_1}\,\overline{x_0} = \overline{x_1 + x_0}$	операция ИЛИ-НЕ, стрелка Пирса ($x_1 \downarrow x_0$)
$y_9 = \overline{x_1}\overline{x_0} + x_1x_0$	равнозначность, x_1 совпадает с x_0
$y_{10} = \overline{x_1}\,\overline{x_0} + x_1\overline{x_0} = \overline{x_0}$	отрицание переменной x_1
$y_{11} = \overline{x_1} + x_0$	импликация (включение) $x_0 \mid x_1$
$y_{12} = \overline{x_1}\,\overline{x_0} + x_1x_0 = \overline{x_1}$	отрицание переменной x_1
$y_{13} = \overline{x_1} + x_0$	импликация (включение) x_1 в x_0
$y_{14} = \overline{x_1} + \overline{x_0} = \overline{x_1x_0}$	операция И-НЕ, штрих Шеффера $x_1 \mid x_0$
$y_{15} = 1$	константа «1»

Из табл. 2.2 и уравнений следует, что функции y_0 и y_{15} не зависят ни от одной переменной и являются константами, y_3, y_5, y_{10} и y_{12} являются функцией только одной переменной. Функции, значение истинности которых не зависят от одной или нескольких переменных, называются ***вырожденными***. Можно заметить также, что $y_i = \overline{y}_{15-i}$, то есть функция y_i является ***инверсной*** по отношению к функции $\overline{y}_{15-i}$ и наоборот.

Рассмотрим теперь пять замечательных классов булевых функций, найденных впервые Э. Постом при установлении необходимых и достаточных условий функциональной полноты системы логических функций. Суперпозиция любых функций, принадлежащих к одному из пяти классов, даёт функцию, принадлежащую к тому же классу.

Первый класс составляют ***функции, сохраняющие константу*** 0, то есть функции $y = f(x_{n-1}; x_{n-2}; ...; x_0) = f(0, 0, ..., 0) = 0$.

Поскольку на одном наборе (00...0) значения функций фиксированы, произвольными остаются лишь $2^n - 1$ наборов. Следовательно, имеется $2^{2^n-1} = \frac{1}{2}2^{2^n}$ булевых функций, сохраняющих константу 0. В табл. 2.2 таковыми являются функции с индексами от 0 до 7.

Второй класс составляют ***функции, сохраняющие константу*** 1, то есть функции $y = f(x_{n-1}; x_{n-2}; ...; x_0) = f(1, 1, ..., 1) = 1$. Как и в предыдущем случае, имеется $\frac{1}{2}2^{2^n}$ булевых функций, сохраняющих константу 1. В табл. 2.2 таковыми являются все функции с нечётными индексами.

Третий класс составляют ***самодвойственные функции***. Булевы функции $f_1(x_{n-1}; x_{n-2}; ...; x_0)$ и $f_2(x_{n-1}; x_{n-2}; ...; x_0)$ называются ***двойственными*** друг другу, если $f_1(x_{n-1}; x_{n-2}; ...; x_0) = \overline{f_2(\overline{x}_{n-1}; \overline{x}_{n-2}; ...; \overline{x}_0)}$ и, разумеется, $f_2(x_{n-1}; x_{n-2}; ...; x_0) = \overline{f_1(\overline{x}_{n-1}; \overline{x}_{n-2}; ...; \overline{x}_0)}$. Смысл введения таких функций становится более ясен, если рассмотреть понятия ***положительной и отрицательной логики***. Последние понятия определяют соглашение о способе кодирования логических переменных реальными физическими сигналами. Так, если для схем, вы-

полненных на биполярных ***n-p-n-***, ***n-***МОП- или КМОП-транзисторах, логическая единица кодируется более высоким (более положительным) потенциалом, то такой способ кодирования называется ***положительной логикой***. Если же «1» кодируется менее высоким (менее положительным) потенциалом, то такой способ кодирования называют ***отрицательной логикой***. Пусть мы имеем логический элемент, выполняющий функцию $y' = x_1 x_0$ в положительной логике. Какую функцию он будет выполнять в отрицательной логике? Очевидно, что это будет $\overline{\overline{y}} = \overline{\overline{x_1}\,\overline{x_0}}$ или $y'' = x_1 + x_0$. y' и y'' и являются двойственными по отношению друг к другу функциями.

Булева функция называется ***самодвойственной***, если она двойственна по отношению к себе самой, то есть выполняется соотношение $f(x_{n-1}; x_{n-2}; ...; x_0) = \overline{f(\overline{x}_{n-1}; \overline{x}_{n-2}; ...; \overline{x}_0)}$. Если, например, при ***n*** = 3 наборы 010 и 101 называть ***противоположными*** (***инверсными***) наборами, тогда можно дать следующее определение самодвойственной функции: ***булева функция называется самодвойственной, если на любых противоположных наборах она принимает противоположные значения***. При расположении наборов в естественном (линейном, лексикографическом) порядке, как в табл. 2.2, противоположные друг другу наборы расположены симметрично относительно середины столбцов. Следовательно, столбец значений самодвойственной функции должен быть антисимметричен своей середине. Для ***n*** переменных существует $\frac{1}{2}2^n = 2^{n-1}$ противоположных наборов, и, следовательно, всего будет $2^{2^{n-1}}$ самодвойственных функций. При ***n*** = 2 существует четыре самодвойственные функции. В табл. 2.2 таковыми являются $y_3; y_5; y_{10}$ и y_{12}.

Четвёртый класс составляют ***линейные функции***. Булева функция называется линейной, если её можно представить в виде: $y = a_{n-1}x_{n-1} \oplus a_{n-2}x_{n-2} \oplus ... \oplus a_0 x_0 \oplus a_{-1}$, где $a_{n-1}; a_{n-2}; ...; a_{-1}$ могут принимать значения 0 и 1, а так как всего этих коэффициентов ***n*** + 1, то может существовать 2^{n+1} наборов коэффициентов и соответственно линейных булевых функций от ***n*** переменных. В табл. 2.2 таковыми являются

$y_0; y_3; y_5; y_6;$

$y_9 = \overline{x}_1\overline{x}_0 + x_1 x_0 = \overline{x_1 \oplus x_0} = x_1 \oplus x_0 \oplus 1;$

$y_{10} = \overline{x}_0 = x_0 \oplus 1;$

$y_{12} = \overline{x}_1 = x_1 \oplus 1; y_{15}.$

Пятый класс булевых функций составляют ***монотонные функции***. Булева функция называется монотонной, если для любых пар наборов X' и X'', для которых выполняется отношение предшествования $X' \le X''$, справедливо неравенство $f(X') \le f(X'') \le$. При ***n*** = 2 существуют следующие последовательности наборов с отношением предшествования: 00-01-11 и 00-10-11, а наборы 01 и 10 несравнимые. Любая монотонная ФАЛ может быть представлена в форме, которая не содержит переменных с отрицанием.

Для рассмотренных выше четырёх классов легко вычисляется число всех функций, принадлежащих к данному классу. Определение же числа всех монотонных функций от ***n*** переменных оказывается очень сложным. Эта задача в общем

случае, по-видимому, не решена до сих пор. Известна формула для вычисления числа F_M монотонных функций только для $\boldsymbol{n} \le 3$: $F_M = 2 + n \cdot n!$. При $\boldsymbol{n} = 2$ $F_M = 6$. В табл. 2.2 таковыми функциями являются y_0; y_1; y_3; y_5; y_7; y_{15}. При $n = 3$ $F_M = 20$, при $\boldsymbol{n} = 4$ $F_M = 168$ (точное значение), при $\boldsymbol{n} = 5$ $F_M = 7581$ (точное значение), при $\boldsymbol{n} = 6$ $F_M = 7828354$ (точное значение).

Теперь сформулируем основную теорему булевой алгебры — ***теорему о функциональной полноте***. Для того чтобы система булевых функций была функционально полной, необходимо и достаточно, чтобы эта система содержала хотя бы одну функцию, не сохраняющую константу 0, хотя бы одну функцию, не сохраняющую константу 1, хотя бы одну несамодвойственную функцию, хотя бы одну нелинейную функцию и хотя бы одну немонотонную функцию. Для удобства выбора функционально полных систем функций двух переменных составим табл. 2.3 на базе табл. 2.2 и рассмотренных пяти классов. Принадлежность к классу отмечена в табл. 2.3 крестиком.

Табл. 2.3 подтверждает, что системы функций И, ИЛИ, НЕ; И, НЕ; ИЛИ, НЕ являются функционально полными, причём первая система сократима. Минимальная функционально полная система, то есть такая полная система функций, удаление из которой любой функции делает систему неполной, называется ***базисом***. В современной цифровой схемотехнике широко используются базисы И-НЕ и ИЛИ-НЕ.

Таблица 2.3. Классы ФАЛ, зависящих от двух переменных

Функция	Класс функции				
	1	2	3	4	5
	Сохраняющая 0	Сохраняющая 1	Самодвойственная	Линейная	Монотонная
y_0	+			+	+
y_1	+	+			+
y_2	+				
y_3	+	+	+	+	+
y_4	+				
y_5	+	+	+	+	+
y_6	+			+	
y_7	+	+			+
y_8					
y_9		+		+	
y_{10}			+	+	
y_{11}		+			
y_{12}			+	+	
y_{13}		+			
y_{14}					
y_{15}		+		+	+

2.1.4. Специальные классы функций алгебры логики

В заключение определим некоторые специальные классы булевых функций, которые заслуживают отдельного рассмотрения: симметричные, однородные, мажоритарные и пороговые. На основе таких функций создаются специальные методы синтеза комбинационных схем.

Функция $f(x_{n-1};x_{n-2};...;x_0)$ называется ***симметричной относительно пары переменных*** x_i и x_j, если эта функция не изменяется при перемене местами переменных x_i и x_j: $f(x_{n-1};...x_i;...;x_j;...x_0)=f(x_{n-1};...x_j;...;x_i;...x_0)$.

Функция называется ***полностью симметричной***, если она симметрична относительно всех пар переменных x_i; x_j, $1 \le i; j \le n$. Такие функции часто называют просто ***симметричными***, а функции, которые симметричны относительно только некоторых, а не всех пар переменных, — ***частично симметричными функциями***.

Симметричные функции просто реализуются на основе устройств с двухсторонней проводимостью, таких как механические контакты и двунаправленные КМОП-ключи.

Важным свойством ФАЛ является ***однородность***. ФАЛ называется ***положительной относительно переменной*** x_i, если для всех 2^{n-1} возможных комбинаций значений оставшихся ***n*** — 1 переменных $f(x_i=1) \ge f(x_i=0)$. Существует такая нормальная форма выражения для ФАЛ, в которой x_i не появляется с отрицанием. ФАЛ называют ***положительной*** или ***монотонной***, если она положительна относительно всех переменных.

Аналогично ФАЛ ***отрицательна относительно переменной*** x_i, если выполняется соотношение $f(x_i=1) \le f(x_i=0)$. Существует такая нормальная форма выражения для ФАЛ, в которой x_i не появляется без отрицания. ФАЛ называется ***отрицательной***, если она отрицательна относительно всех переменных.

ФАЛ, которая на некоторых переменных является положительной, а на других — отрицательной, называется ***однородной.*** ФАЛ $y=\bar{x}_2x_1+x_1x_0$ является однородной, так как она положительна по переменным x_1 и x_0 и отрицательна по переменной x_2, а ФАЛ $y=\bar{x}_2x_1+\bar{x}_1x_0$ не является однородной, так как переменная x_1 присутствует без инверсии и с инверсией. Однородную ФАЛ иногда называют ***смешанной монотонной***.

Если число переменных ***n*** нечётно и ФАЛ принимает значение 1; если $\frac{n+1}{2}$ или более (то есть более половины) переменных, принимает значение 1, и 0 — в противном случае, и такая ФАЛ называется ***мажоритарной функцией***. Наиболее широко используется мажоритарная функция, называемая схемой голосования 2 из 3, в качестве восстанавливающего органа в схемах с многократным резервированием.

Обобщением мажоритарной функции является ***пороговая функция***.

ФАЛ называется пороговой, если существует множество действительных чисел $w_{n-1};w_{n-2};...;w_0$, называемых ***весами переменных***, и действительное число T, называемое ***порогом функции***, такие, что

$$f(x_{n-1};x_{n-2};...;x_0)=1, \text{ если } \sum_{i=0}^{n-1}x_iw_i \ge T \text{ и } f(x_{n-1};x_{n-2};...;x_0)=0, \text{ если } \sum_{i=0}^{n-1}x_iw_i < T$$

где x_1 принимает только значения 0 и 1, которые ***арифметически*** умножаются на веса w_i, а $\sum$ являются ***арифметическими суммами***. Мажоритарные и порого-

вые функции являются подклассами однородных функций. Пороговые функции были впервые введены для моделирования функций нервных клеток (нейронов), но в дальнейшем изучались в теории схем из функциональных элементов, в теории распознавания образов, для повышения надёжности избыточных переключательных схем и в других областях.

Специальные классы функций и их свойства, представляющие интерес, изменяются с возникновением новых реальных задач и с изменением схемотехники и технологии цифровых элементов [6].

2.1.5. Минимизация функций алгебры логики

Минимизация функций алгебры логики — это процедура нахождения наиболее простого представления ФАЛ в виде суперпозиции функций, составляющих функционально полную систему, при одновременной оптимизации ее технической реализации по некоторым критериям в условиях ряда ограничений. Критериями оптимизации могут быть объем оборудования (количество вентилей, корпусов), габариты, вес, энергопотребление, стоимость, быстродействие, надежность. В качестве ограничений могут выступать допустимые к использованию системы элементов, число элементов в корпусе, коэффициенты объединения по входу и разветвления по выходу логических элементов, необходимость реализации системы ФАЛ, а также ряд перечисленных выше критериев оптимизации.

Решение задачи минимизации ФАЛ в полном объёме является трудной проблемой, хотя бы потому, что ряд критериев оптимизации находятся в противоречивом отношении друг к другу, например одновременное снижение энергопотребления и повышение быстродействия. На практике обычно решается задача оптимизации по нескольким или даже одному из критериев. Наиболее часто это делается по минимуму необходимого числа базовых логических элементов И, ИЛИ, НЕ, так как при этом в большинстве случаев удовлетворяются требования получения минимальных габаритов, веса, энергопотребления, стоимости, а также повышения быстродействия и надёжности. Иногда ограничиваются ещё более простой задачей представления ФАЛ в дизъюнктивной или конъюнктивной форме, содержащей наименьшее возможное число букв, когда, например, для дизъюнктивных форм в выражении присутствует как можно меньше слагаемых, являющихся элементарными произведениями, которые, в свою очередь, содержат как можно меньше сомножителей. Такую задачу принято называть ***канонической задачей минимизации ФАЛ***.

Пример. ФАЛ, заданную таблицей истинности (табл. 2.4), можно представить следующими выражениями:

Таблица 2.4. Произвольная ФАЛ

Номер набора	x_2	x_1	x_0	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

$$y = x_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0, \quad (2.23)$$

$$y = x_2 \bar{x}_1 + x_2 \bar{x}_0 + \bar{x}_2 x_0, \quad (2.24)$$

$$y = (x_2 + x_1 + x_0)(x_2 + \bar{x}_1 + x_0)(\bar{x}_2 + \bar{x}_1 + \bar{x}_0), \quad (2.25)$$

$$y = (x_2 + x_0)(\bar{x}_2 + \bar{x}_1 + \bar{x}_0). \quad (2.26)$$

В выражении (2.23), записанном в СДНФ, пять слагаемых по три буквы в каждом, а всего 15 букв

и три инвертора, в то время как в выражении (2.24) три слагаемых по две буквы в каждом, а всего шесть букв и три инвертора. Выражение (2.24) является минимальной дизъюнктивной формой данной ФАЛ.

В выражении (2.25), записанном в СКНФ, три сомножителя по три буквы в каждом, а всего девять букв и три инвертора, в то время как в выражении (2.26) два сомножителя по две и три буквы, а всего пять букв и три инвертора. Выражение (2.26) является минимальной конъюнктивной формой данной ФАЛ.

Применяя скобочные формы и формы с групповыми инверсиями, выражения (2.24) и (2.26) можно ещё упростить:

$$y = x_2\bar{x}_1 + x_2\bar{x}_0 + \bar{x}_2x_0 = x_2(\bar{x}_1 + \bar{x}_0) + \bar{x}_2x_0 = x_2 \cdot \overline{x_1x_0} + \bar{x}_2x_0, \qquad (2.27)$$

где пять букв и два инвертора;

$$y = (x_2 + x_0)(\bar{x}_2 + \bar{x}_1 + \bar{x}_0) = (x_2 + x_0) \cdot \overline{x_2x_1x_0}, \qquad (2.28)$$

где пять букв и один инвертор.

В настоящее время в теории проектирования логических схем наиболее полно исследованы именно задачи минимизации дизъюнктивных и конъюнктивных нормальных форм, обеспечивающих рациональное решение при синтезе комбинационных схем, на входах которых доступны как переменные, так и их инверсии. Парафазное представление переменных легко обеспечивается, если они снимаются с выходов триггеров, используемых в качестве запоминающих ячеек разрабатываемых цифровых устройств.

Справедливости ради надо отметить, что сформулированная выше задача минимизации ФАЛ была чрезвычайно актуальной в тот период времени, когда разработка цифровых устройств велась на электромеханических переключательных элементах, дискретных радиокомпонентах и интегральных схемах малой степени интеграции.

Имеющиеся в настоящее время возможности микроэлектроники, в частности наличие схем средней, большой и сверхбольшой интеграции, таких как БМК, ПЛИС, постоянные запоминающие устройства (ПЗУ), мультиплексоры и другие разновидности программируемых логических интегральных схем, позволяют реализовать очень сложные системы ФАЛ, используя один типономинал микросхемы без каких-либо процедур минимизации.

Пример. Используя ПЗУ или флеш-память с организацией 8К×8, можно реализовать систему восьми ФАЛ, зависящих от 13 переменных, где 13 — разрядность адреса ПЗУ.

Учитывая, что ПЛИС и ПЗУ требуют сложной аппаратно-программной поддержки для их программирования, а очень часто в инженерной практике решаются более простые задачи, рассмотрим вопросы минимизации ФАЛ, остановившись на некоторых нашедших наибольшее распространение методах минимизации ФАЛ.

К настоящему времени широкое применение получили:

1. Расчётный метод (метод непосредственных преобразований).
2. Расчётно-табличный метод (метод Квайна—Мак-Класки).
3. Метод Петрика (развитие метода Квайна—Мак-Класки).
4. Табличный метод (карты Карно).
5. Метод гиперкубов.

6. Метод факторизации.

7. Метод функциональной декомпозиции и др.

Первый метод применяется при числе переменных $n \le 3$ и основан на использовании операций склеивания, поглощения и развёртывания. Ниже он будет рассмотрен подробно. Второй и третий методы используются при $n \le 16$ в профессиональных разработках и ориентированы на использование САПР с применением ЭВМ [4], [5], [6], [8], [9], [10]. Здесь они рассматриваться не будут. Четвёртый метод является самым распространённым инженерным методом минимизации ФАЛ для $n \le 6$ и далее будет подробно рассмотрен. Шестой метод не имеет каких-либо существенных достижений при решении общих задач, более простых, чем метод перебора всех формул ФАЛ даже для $n = 3$. Практически он используется для уменьшения сложности минимальных ДНФ и КНФ, полученных с использованием первого или четвёртого методов. Он основан на использовании скобочных форм и форм с групповыми инверсиями [4], [6], [8].

Седьмой метод основан на представлении ФАЛ, зависящей от n переменных, в виде суперпозиций функций, зависящих от меньшего числа переменных, для которых можно применить вышеперечисленные методы.

Исходной формой для большинства методов являются либо таблица истинности, либо одна из совершенных форм СДНФ или СКНФ. Если ФАЛ задана в другом виде, то предполагается, что она сначала переводится в СДНФ или СКНФ с использованием основных законов булевой алгебры (см. раздел 2.1.2).

При выполнении процедур канонической минимизации большую роль играют понятия «***импликанты***» и «***простой импликанты***» ФАЛ. Булева функция $z = f_1(x_{n-1};x_{n-2};...;x_0)$ называется ***импликантой*** булевой функции $y = f_2(x_{n-1};x_{n-2};...;x_0)$, если на любом наборе значений переменных $x_{n-1};x_{n-2};...;x_0$, на котором значение функции z равно 1, значение функции y также равно 1.

Простой импликантой функции y называется всякое элементарное произведение $z = \tilde{x}_{n-1} \cdot \tilde{x}_{n-2} \cdot ... \cdot \tilde{x}_0$, являющееся импликантой функции y и такое, что никакая его собственная часть (то есть произведение, полученное из произведения z выбрасыванием одного или нескольких сомножителей $\tilde{x}_i$) уже не является импликантой функции y. Так как в дальнейшем будут использоваться только простые импликанты, опустим слово «простые», то есть если в тексте встречается понятие «импликанта», то надо помнить, что имеется в виду «простая импликанта».

В общем случае минимизация ФАЛ, заданной в СДНФ, требует выполнения процедур следующих трёх этапов.

1 этап — переход от СДНФ к ***сокращённой ДНФ*** (СокрДНФ). СокрДНФ — это форма ФАЛ, членами которой являются изолированные (не склеивающиеся) элементарные произведения. Этот этап основан на выполнении всех возможных склеиваний друг с другом: сначала конституент единицы, а затем произведений меньшего ранга (импликант). Отметим, что существуют ФАЛ, у которых СДНФ совпадает с СокрДНФ.

2 этап — переход от СокрДНФ к ***тупиковой ДНФ*** (ТДНФ). ТДНФ — это форма ФАЛ, членами которой являются импликанты, среди которых нет ни одной лишней. ***Лишней импликантой*** называется член ФАЛ, удаление которого из выражения не изменяет ФАЛ. Отметим, что возможны случаи, когда в СокрДНФ нет лишних импликант. Иногда из одной СокрДНФ можно получить несколько раз-

личных ТДНФ. Термин «тупиковая» говорит о том, что дальнейшая минимизация в рамках нормальных форм уже невозможна.

3 этап — переход от ТДНФ к минимальной форме. Этот этап основывается на использовании групповых инверсий и скобочных форм, не является систематическим и во многом определяется опытом разработчика.

2.1.5.1. Расчётный метод минимизации

Пусть нам требуется минимизировать ФАЛ, заданную выражением (2.23).

1 этап. Выполняем операции склеивания конституент единицы. Для упорядочения этой процедуры запишем выражение (2.23) в виде нескольких строк по следующему правилу: первая строка — это исходное уравнение, вторая строка — это вторая конституента и все последующие, третья строка — это третья конституента и все последующие и т.д. Это допустимо, так как в булевой алгебре действует закон тавтологии.

$$\left.\begin{aligned} y = {} & x_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0 + \\ & + x_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0 + \\ & + x_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0 + \\ & + \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0 \end{aligned}\right\} \qquad (2.29)$$

Производится проверка на склеивание первого члена в каждой строке со всеми остальными в данной строке. В первой строке склеиваются первая и третья конституенты, во второй строке — первая со второй и четвертой, в третьей строке первая конституента с остальными не склеивается, и в последней строке конституенты склеиваются. Поскольку все конституенты участвовали хотя бы в одном склеивании, то в СокрДНФ ни одной конституенты не будет. После этой процедуры получаем следующее выражение:

$$y = x_2 \bar{x}_0 + x_2 \bar{x}_1 + \bar{x}_1 x_0 + \bar{x}_2 x_0. \qquad (2.30)$$

Дальнейшее склеивание не может быть выполнено, так как все члены выражения (3.8) являются изолированными.

2 этап. Необходимо выявить лишние импликанты в выражении (2.30). Это можно сделать двумя способами. При первом способе развёртывают одну импликанту до конституент единицы, а затем смотрят, не поглощаются ли эти конституенты остальными импликантами. Первая импликанта развёртывается до суммы $x_2 \bar{x}_0 = x_2 \cdot 1 \cdot \bar{x}_0 = x_2(x_1 + \bar{x}_1) = x_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0$, причём конституента $x_2 x_1 \bar{x}_0$ не поглощается ни одной импликантой, следовательно, импликанта $x_2 \bar{x}_0$ не является лишней. Вторая импликанта $x_2 \bar{x}_1$ развёртывается до суммы $x_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0$, причём обе конституенты поглощаются остальными импликантами, следовательно, импликанта $x_2 \bar{x}_1$ лишняя. Продолжим эту процедуру, оставив пока импликанту $x_2 \bar{x}_1$ в выражении (2.30). Импликанта $\bar{x}_1 x_0$ развёртывается до суммы $x_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 x_0$, причём обе конституенты поглощаются остальными импликантами, следовательно, импликанта $\bar{x}_1 x_0$ лишняя. Продолжим, оставив в выражении (2.30) и эту импликанту. Развёртывание последней импликанты даёт

сумму $\bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0$, в которой конституента $\bar{x}_2 \bar{x}_1 x_0$ не поглощается ни одной импликантой, следовательно, импликанта $\bar{x}_2 x_0$ не является лишней. Выявлены две лишние импликанты, но это не значит, что обе они могут быть отброшены, так как каждая из них проверялась при вхождении второй в выражение (2.30). Следовательно, отбросить наверняка можно одну из них, а затем снова произвести проверку возможности отбросить и другую. Если отбросить импликанту $x_2 \bar{x}_1$, то проверка показывает, что импликанта $\bar{x}_1 x_0$ не будет лишней, а если отбросить $\bar{x}_1 x_0$, то $x_2 \bar{x}_1$ не будет лишней. Итак, можно отбросить одну из выявленных двух импликант и в результате получаются две ТДНФ одинаковой сложности, содержащих по шесть букв:

$$y = x_2 \bar{x}_0 + \bar{x}_1 x_0 + \bar{x}_2 x_0, \tag{2.31}$$

$$y = x_2 \bar{x}_0 + x_2 \bar{x}_1 + \bar{x}_2 x_0. \tag{2.32}$$

3 этап. Выражение (2.31) можно записать в виде

$$y = x_2 \bar{x}_0 + (\bar{x}_2 + \bar{x}_1) x_0. \tag{2.33}$$

Оно содержит пять букв и требует три инвертора. Применив закон двойного отрицания и правило де-Моргана, выражение (2.33) можно преобразовать:

$$y = x_2 \bar{x}_0 + \overline{\overline{(\bar{x}_2 + \bar{x}_1)}} x_0 = x_2 \bar{x}_0 + \overline{x_2 x_1} \cdot x_0. \tag{2.34}$$

Последнее выражение содержит пять букв и требует два инвертора.

Аналогично можно упростить и выражение (2.32):

$$y = x_2 \bar{x}_0 + x_2 \bar{x}_1 + \bar{x}_2 x_0 = x_2 (\bar{x}_1 + \bar{x}_0) + \bar{x}_2 x_0 = x_2 \cdot \overline{x_1 x_0} + \bar{x}_2 x_0. \tag{2.35}$$

Второй способ выявления лишних импликант заключается в следующем. На значение истинности функции влияет только та импликанта, которая сама равна 1. Любая импликанта принимает значение 1 только на одном наборе своих аргументов. Но если именно на этом наборе сумма остальных импликант также обращается в 1, то рассматриваемая импликанта не влияет на значение истинности функции даже в этом единственном случае, то есть является лишней.

Применим это правило к выражению (2.30). Импликанта $x_2 \bar{x}_0$ принимает значение 1 на наборе $x_2 = 1, x_0 = 0$. Подставив этот набор в оставшуюся сумму $x_2 \bar{x}_1 + \bar{x}_1 x_0 + \bar{x}_2 x_0$, получим $\bar{x}_1$, что говорит о том, что первая импликанта не является лишней. Импликанта $x_2 \bar{x}_1$ принимает значение 1 на наборе $x_2 = 1, x_1 = 0$. Подставив этот набор в сумму $x_2 \bar{x}_0 + \bar{x}_1 x_0 + \bar{x}_2 x_0$, получим $\bar{x}_0 + x_0 = 1$, что говорит о том, что импликанта $x_2 \bar{x}_1$ лишняя. Оставим пока эту импликанту и продолжим анализ других импликант. Импликанта $\bar{x}_1 x_0$ принимает значение 1 на наборе $x_1 = 0, x_0 = 1$. Подставив этот набор в сумму $x_2 \bar{x}_0 + x_2 \bar{x}_1 + \bar{x}_2 x_0$, получим $x_2 + \bar{x}_2 = 1$, что говорит о том, что импликанта $\bar{x}_1 x_0$ лишняя. Оставляем и её и продолжаем процедуру. Импликанта $\bar{x}_2 x_0$ принимает значение 1 на наборе $x_2 = 0,\ x_0 = 1$. Под-

ставив этот набор в сумму $x_2\bar{x}_0 + x_2\bar{x}_1 + \bar{x}_1x_0$, получаем $\bar{x}_1$, что говорит о том, что импликанта $\bar{x}_2x_0$ не является лишней.

Как и в первом случае, нельзя отбрасывать обе обнаруженные лишние импликанты, так как каждая из них проверялась при вхождении второй в оставшуюся сумму. Опустим рассмотрение дальнейших процедур, так как они аналогичны процедурам, выполненным первым способом.

Можно сделать вывод, что даже для этого простого примера пришлось выполнять достаточно много однообразных действий, требующих внимания и времени, поэтому расчётный метод минимизации ФАЛ применяется в основном для ФАЛ, зависящих от двух или трёх переменных.

2.1.5.2. Табличный метод минимизации с помощью карт Карно

При этом методе два первых этапа выполняются при помощи специальных карт, впервые предложенных Вейтчем [11] и модернизированных Карно [12]. Практическое применение получили именно карты Карно, а не диаграммы Вейтча, и хотя с момента опубликования их оригинальных работ прошло более 50 лет, до сих пор многие авторы называют карты Карно диаграммами Вейтча.

Приведём цитату из работы [8]: «Матрица Вейтча отличается от матрицы Карно расположением столбцов и строк. В то время как Карно пользуется циклическим порядком следования символов, а именно 00, 01, 11, 10, Вейтч располагает символы в порядке возрастания двоичных чисел, а именно 00, 01, 10, 11. Столбцы или строки 00 и 01, так же как столбцы или строки 10 и 11, являются в матрице Вейтча соседними, но столбцы или строки 01 и 10 в ней не являются ни соседними, ни крайними. Хотя матрица Вейтча и обладает некоторыми преимуществами по сравнению с алгебраическими методами, матрица Карно более удобна в обращении и не требует столь большой затраты времени». Итак, табличный метод минимизации ФАЛ — это метод, основанный на использовании карт Карно.

Карта Карно является специальной формой таблицы истинности ФАЛ, позволяющей не только задать ФАЛ, но и выполнить первый и второй этапы минимизации. Таблица истинности (см. табл. 2.1) содержит 2^n строк, в которых наборы $\boldsymbol{n}$ переменных расположены в линейной лексикографической последовательности, а также столбец значений ФАЛ на этих наборах. Напомним, что в таблице истинности переменные с большим весом располагаются на левой позиции набора.

Карта Карно содержит 2^n клеток (квадратов), расположенных в виде строки ($\boldsymbol{n} = 1, 2$) либо в виде двумерной матрицы ($\boldsymbol{n} \geq 2$). Каждая клетка, как и строка в таблице истинности, соответствует одному набору. Для того чтобы можно было производить минимизацию ФАЛ, необходимо в смежных в геометрическом смысле клетках карты расположить соседние наборы. Это можно обеспечить, если наборы переменных, определяющих «координаты» клетки карты Карно, расположить в циклическом коде Грея, у которого каждое следующее значение отличается от предыдущего только в одном разряде.

На рис. 2.8 представлена так называемая ***эталонная карта Карно*** для $\boldsymbol{n} = 3$. Она служит для указания расположения переменных как

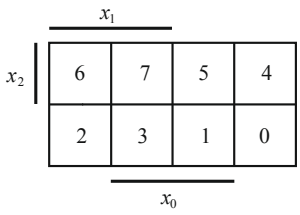

Рис. 2.8. Эталонная карта Карно для $n = 3$

координат клеток, так и наборов этих переменных. Координатой клеток в горизонтальном направлении служат наборы переменных x_1, x_0, а координатой клеток в вертикальном направлении служит одна переменная x_2.

Известно, что каждая из ***n*** переменных встречается в половине наборов без инверсии, а в другой половине — с инверсией. Три толстые линии, расположенные с внешней стороны карты Карно, указывают, что в соответствующих им половинах клеток указанная рядом с этой линией переменная встречается в наборе без инверсии и, соответственно, в другой половине с инверсией. Так как переменным x_0, x_1, x_2 условно присваиваются «веса» соответственно $2^0 = 1$; $2^1 = 2$; $2^2 = 4$, то восемь наборов в клетках карты Карно будут расположены так, как указано на рис. 2.8. Итак, расположение переменных как координат клеток карты Карно и номеров наборов в эталонной карте должно строго соответствовать друг другу. Можно произвольно поменять местами переменные x_0, x_1, x_2, но тогда обязательно надо поменять местами и расположение наборов.

Правильность оформления эталонной карты Карно можно проверить следующим образом. Если толстую линию, соответствующую переменной x_2, протянуть вправо по горизонтали над клетками карты Карно, то она пройдёт над клетками, в которых минимальный номер набора должен совпадать с весом переменной x_2, равным $2^2 = 4$. Аналогично толстая линия, соответствующая переменной x_1, при перемещении вниз по вертикали пройдёт над клетками, в которых минимальный номер набора должен совпадать с весом переменной x_1, равным 2. Это должно выполняться для всех переменных.

Несмотря на то что карты Карно изображаются на плоскости, с точки зрения обеспечения соседства их клеток карты нужно считать трёхмерными объектами, так как клетки, расположенные на концах одних и тех же строк и столбцов, также являются соседними. Так, карту для трёх переменных следует рассматривать как цилиндр со склеенными правым и левым краями. Карту Карно для четырёх переменных (см. рис. 2.11, *а*) нужно считать склеенной не только по правому и левому краям, но и по верхнему и нижнему. Таким образом, карта Карно для четырёх переменных должна рассматриваться как поверхность тора.

Рабочая карта Карно, соответствующая табл. 2.4, будет иметь вид, представленный на рис. 2.9. Буква ***y*** рядом с косой линией, расположенной в левом верхнем углу карты Карно, обозначает реализуемую функцию, а цифры 0 и 1 в клетках карты указывают значения этой функции на соответствующих наборах. Полученную рабочую карту Карно можно интерпретировать как компактное представление ФАЛ в СДНФ (по значениям истинности) либо в СКНФ (по значениям ложности). Дальнейшее изложение ведётся в предположении, что минимизация ведётся в дизъюнктивных формах.

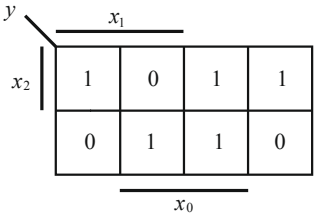

Рис. 2.9. Рабочая карта Карно для ФАЛ, заданной таблицей 2.4

Процесс минимизации с помощью карт Карно базируется на использовании операции склеивания и основан на следующих положениях:

1. На картах Карно необходимо выделить монолитные области клеток, содержащих «1» и образующих строку, столбец, прямоугольник или квадрат с числом

клеток в них, равным 1, 2, 4, 8 и т.д. Эти выделенные области (или контуры покрытия) будут соответствовать импликантам. Очевидно, что одна изолированная единичная клетка будет соответствовать конституенте единицы. Две смежные клетки будут соответствовать импликанте, ранг которой $\boldsymbol{r = n}$ — 1, четыре смежные клетки будут соответствовать импликанте, ранг которой $\boldsymbol{r = n}$ — 2, и т.д.

2. Переменные, от которых импликанта не зависит, входят в соответствующий выделенный контур как в виде x_i, так и в виде $\bar{x}_i$, а остальные переменные — только либо в виде x_i, либо в виде $\bar{x}_i$.

3. На основании закона тавтологии любая единичная клетка может быть включена в любое число различных контуров.

4. Для получения минимальных ТДНФ в карте Карно не должно быть лишних покрытий, то есть каждую единичную клетку достаточно использовать хотя бы один раз. Это положение не действует, если в схеме необходимо устранить риски сбоя.

5. Существуют эквивалентные покрытия для получения различных минимальных ТДНФ.

6. Существуют функции, для которых СДНФ совпадает с минимальной ТДНФ (в этом случае на карте Карно все единичные клетки изолированные).

7. Если в карте Карно нет ни одной 1, то ФАЛ эквивалентна константе 0; если нет ни одного 0, то ФАЛ эквивалентна константе 1; если единицы занимают половину клеток карты Карно и представляют собой монолитный массив в виде строки, столбца, прямоугольника или квадрата, то соответствующая импликанта состоит из одной переменной со знаком или без знака инверсии.

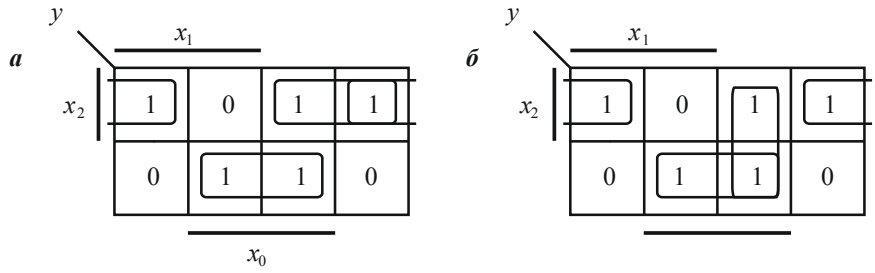

Рис. 2.10. Рабочие карты Карно с двумя эквивалентными покрытиями

С учётом сказанного на картах Карно рис. 2.10 можно выделить три контура, содержащих по две 1. Два варианта покрытия обусловлены тем, что 1 в клетке с набором 5 может образовать контур из двух клеток либо с набором 4 (рис. 2.10, *а*), либо с набором 1 (рис. 2.10, *б*). Поясним получение импликанты для контура, образованного двумя клетками в нижней строке карты.

Переменная x_2 входит в этот контур только с инверсией, переменная x_1 входит в этот контур и с инверсией, и без инверсии, поэтому по ней осуществляется склеивание и она исчезает; переменная x_0 входит в этот контур только без инверсии, поэтому импликанта имеет вид $\bar{x}_2 x_0$. Другой способ определения импликанты будет показан ниже. Для выявленных двух покрытий можно записать

$$y = x_2\bar{x}_1 + x_2\bar{x}_0 + \bar{x}_2x_0, \tag{2.36}$$

$$y = x_2\bar{x}_0 + \bar{x}_2x_0 + \bar{x}_1x_0. \tag{2.37}$$

Простота получения уравнений (2.36) и (2.37) показывает существенное преимущество табличного метода карт Карно перед расчётным методом.

На рис. 2.11 показаны эталонные карты Карно для $\boldsymbol{n} = 4$, 5 и 6, причём карты Карно для $\boldsymbol{n} = 5$ и 6 можно рассматривать как соответственно две и четыре карты Карно для $\boldsymbol{n} = 4$, имеющие общие границы (они выделены толстыми центральными линиями).

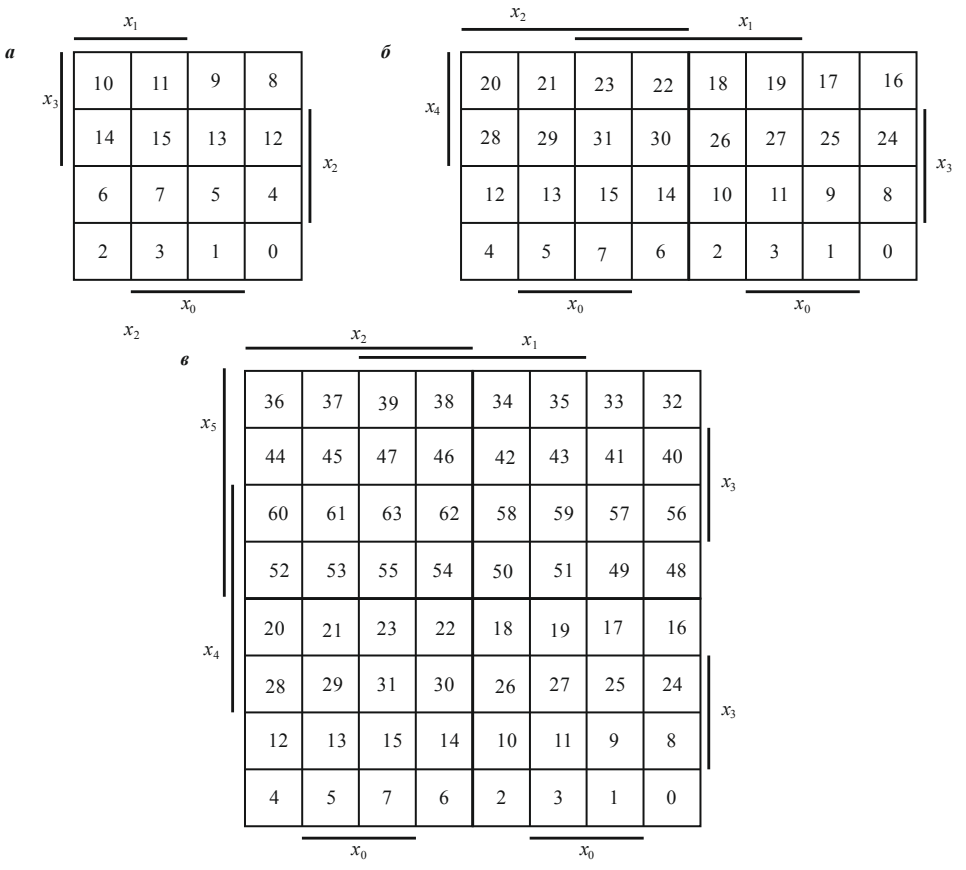

Рис. 2.11. Эталонные карты для $n = 4$, 5 и 6

Карты Карно для $\boldsymbol{n} = 4$, являющиеся составной частью карт Карно для $\boldsymbol{n} = 5$ и 6 и имеющие общие границы, называются ***соседними***. Правило соседства для какой-либо клетки в этих случаях будет выглядеть так: для любой выделенной клетки соседними являются четыре соседние клетки в карте Карно для $\boldsymbol{n} = 4$ и клетки, расположенные в соседних картах Карно для $\boldsymbol{n} = 4$ симметрично выделенной клетке относительно границ соседних карт Карно.

Пример. Для клетки с набором 25 на рис. 2.11, *б* соседними являются клетки с номерами наборов 9, 27, 17, 24 и 29. Для клетки с набором 2 на рис. 2.11, *б* соседними являются клетки 3, 10, 0, 18 и 6. Для клетки с набором 43 на рис. 2.1, *в* соседними являются клетки с наборами 59, 42, 35, 41 и 47, 11. Для клетки с набором 22 на рис. 2.11, *в* соседними являются клетки с наборами 23, 30, 20, 6 и 54, 18.

Рассмотрим ещё несколько примеров для функций, зависящих от четырех, пяти и шести переменных. На рис. 2.12, *а* четыре первые клетки образуют квадрат, которому соответствует импликанта $\bar{x}_2\bar{x}_0$. На рис. 2.12, *б* контур, выделенный штриховой линией, оказывается лишним, так как все его клетки являются составными частями четырёх контуров из двух клеток. Из карты Карно (рис. 2.12, *б*) получаем:

$$y_1 = x_3x_2\bar{x}_0 + x_3\bar{x}_2x_0 + x_2x_1x_0 + \bar{x}_2x_1\bar{x}_0. \qquad (2.16)$$

Для карты Карно (рис. 2.12, *в*) покажем ещё один способ определения импликант, соответствующих выделенным контурам, состоящих в данном случае из двух столбцов. Для левого контура запишем минимальный и максимальный наборы x_3, x_2, x_1, x_0. Таковыми являются наборы 2 и 14.

Запишем их двоичные представления 0010 и 1110 одно на другом. Переменные, соответствующие позициям с наложенными 0 и 1, склеиваются, а совпадающие позиции соответствуют искомой импликанте $x_1\bar{x}_0$. Аналогичная процедура для правого контура даёт импликанту $\bar{x}_1x_0$. В итоге получаем

$$y_2 = x_1\bar{x}_0 + \bar{x}_1x_0 = x_1 \oplus x_0. \qquad (2.17)$$

Если теперь на той же карте Карно выделить контуры, соответствующие импликантам x_1 и x_0 (см. рис. 2.12, *г*), то окажется, что общая часть этих контуров будет содержать нулевые клетки.

Для ФАЛ, представленной на рис. 3.5, *д*, можно записать

$$y_3 = x_3x_1 + \bar{x}_2x_1 + x_1\bar{x}_0 + \bar{x}_3x_2\bar{x}_1x_0. \qquad (2.18)$$

Преобразуем это выражение:

$$y_3 = (x_3 + \bar{x}_2 + \bar{x}_0)x_1 + \bar{x}_3x_2x_0 \cdot \bar{x}_1 = \overline{\bar{x}_3x_2x_0} \cdot x_1 + \bar{x}_3x_2x_0 \cdot \bar{x}_1 = \bar{x}_3x_2x_0 \oplus x_1. \qquad (2.19)$$

Если на той же карте Карно выделить контуры, соответствующие импликантам $\bar{x}_3x_2x_0$ и x_1 (см. рис. 2.12, *е*), то окажется, что общая часть этих контуров содержит нуль. Теперь можно сделать следующий вывод: если в карте Карно можно выделить два пересекающихся контура с общей нулевой частью, то импликанты, соответствующие этим контурам, объединяются знаком операции «сумма по *mod*2».

Картам Карно, показанным на рис. 2.13, *а*–*г*, соответствуют следующие выражения:

$$y_4 = x_4\bar{x}_3 + x_4\bar{x}_1\bar{x}_0 + x_2x_1x_0 + \bar{x}_4x_3\bar{x}_2\bar{x}_1x_0, \qquad (2.20)$$

$$y_5 = x_3\bar{x}_0 + x_2\bar{x}_0 + x_1\bar{x}_0 + \bar{x}_4\bar{x}_3\bar{x}_2\bar{x}_1. \qquad (2.21)$$

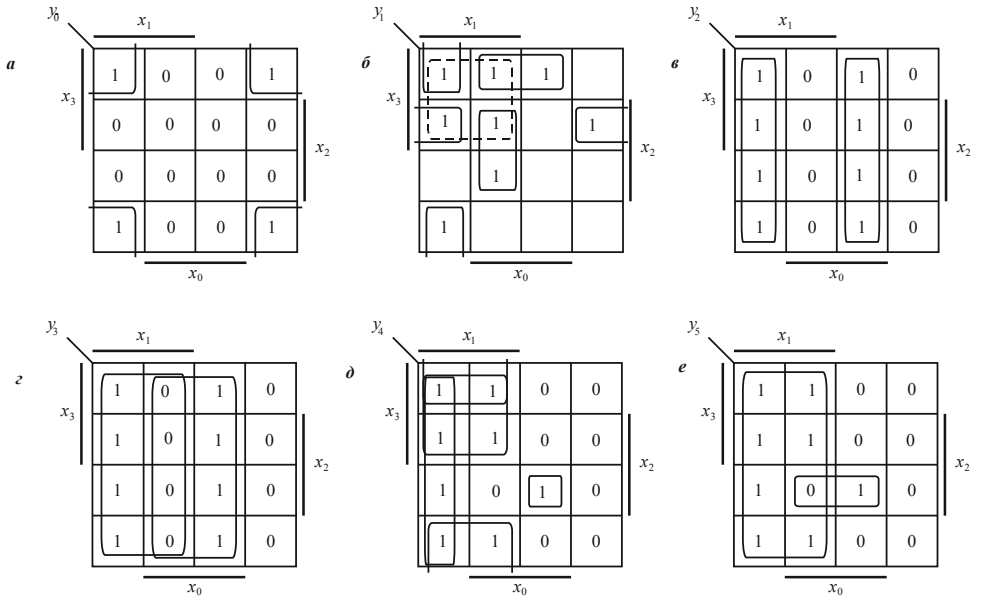

Рис. 2.12. Рабочие карты Карно произвольных ФАЛ, зависящих от четырёх переменных

$$y_6 = x_3\bar{x}_2x_0 + x_3\bar{x}_1x_0 + x_5\bar{x}_4x_0 + x_4\bar{x}_3x_1\bar{x}_0 + \bar{x}_5\bar{x}_3x_2x_1x_0, \tag{2.22}$$

$$y_7 = x_5\bar{x}_4x_3 + x_4x_1\bar{x}_0 + \bar{x}_4x_3x_2 + x_4\bar{x}_3\bar{x}_2\bar{x}_1. \tag{2.23}$$

В работе [13] рассмотрен способ минимизации функций пяти и шести переменных с помощью одной карты Карно для $n = 4$.

Карты Карно удобно использовать и для минимизации неполностью определённых функций. Пусть требуется выработать осведомительный сигнал y_8 о том, что содержимое одноразрядного двоично-десятичного счётчика равно 6 или 7. Выходные переменные его четырёх двоичных разрядов обозначим $x_3;x_2;x_1;x_0$. Очевидно, что на наборах 0–5 и 8, 9 $y_8 = 0$, на наборах 6 и 7 $y_8 = 1$, а наборов 10–15 никогда не будет в нормально работающем двоично-десятичном счётчике и, следовательно, значение сигнала y_8 на этих наборах безразлично. Безразличные значения ФАЛ на картах Карно обозначаются каким-либо символом: крестиком, чертой, буквой и т. п. Карта Карно для этого случая приведена на рис. 2.14.

Доопределив безразличные значения y_8 на наборах 14 и 15 единицами, получим следующее минимальное выражение:

$$y_8 = x_2x_1. \tag{2.24}$$

После реализации этой функции она становится полностью определённой, то есть на безразличных наборах, включённых в контур, будут реализовываться значения 1, а на невключённых в контур — значение 0.

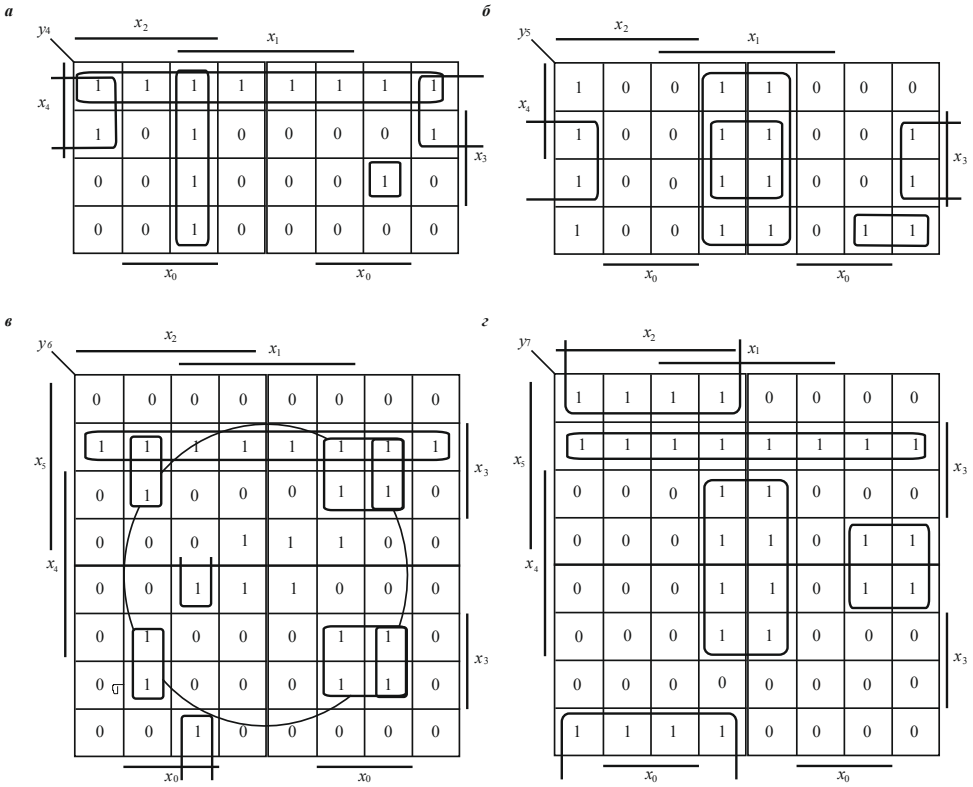

Рис. 2.13. Рабочие карты Карно произвольных ФАЛ, зависящих от пяти и шести переменных

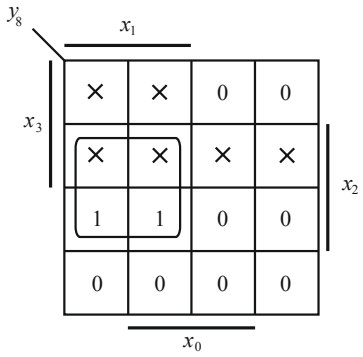

Рис. 2.14. Рабочая карта Карно для неполностью определённой ФАЛ

Сформулируем в заключение достоинства и недостатки метода минимизации ФАЛ с помощью карт Карно. ***Достоинства:***

1. Основными достоинствами применения карт Карно являются компактность, простота и наглядность представления полностью и не полностью определённых функций.

2. Их применение оправданно для $n = 2 – 6$, а при определённых навыках — даже для $n = 7$ и 8, что соответствует большинству реально встречающихся инженерных задач.

3. Карты Карно можно использовать для минимизации ФАЛ, заданных как в СДНФ, так и в СКНФ.

4. Удобно минимизировать системы булевых функций, так как на картах Карно легко выделять общие части реализуемой системы ФАЛ.

5. Легко находятся минимальные комбинации контуров по их виду на карте Карно.

6. Просто получить запись ФАЛ с одной связкой «сумма по модулю 2».

7. На одной карте Карно можно изобразить систему ФАЛ, в каждой из которых одна 1 или один 0 (например, для выходов дешифраторов).

8. Для заполнения карты Карно не обязательно задавать ФАЛ в СДНФ или в СКНФ (можно подставлять значение набора в любой вид ФАЛ и заносить значение функции на этом наборе в соответствующую клетку карты Карно.

9. Карты Карно сразу позволяют реализовать первые два этапа минимизации (склеивание и выявление лишних импликант).

Недостатки:

1. Затруднительно использовать карты Карно при $n \geq 6$.

2. Метод не является алгоритмически систематическим, многое зависит от навыков разработчика. Удобство обращения и экономия времени во многом зависят от его способности распознавать оптимальные конфигурации покрытия карт Карно.

2.1.6. Синтез комбинационных схем

Все схемы в цифровой схемотехнике можно разбить на два класса: комбинационные и последовательностные (цифровые автоматы).

Комбинационными называют схемы, выходные сигналы которых зависят от *комбинации* входных логических переменных. Их часто называют схемами без обратных связей или схемами без элементов памяти. Основными задачами теории комбинационных схем являются задачи анализа и синтеза этих схем. ***Задача анализа*** включает в себя нахождение функции, реализованной конкретной схемой, определение по ней зависимости выходного сигнала на каждом из входных наборов и выявление рисков сбоя на выходе при их смене. ***Задача синтеза*** сводится к представлению необходимой функции в виде суперпозиции функций, реализуемых некоторым заранее заданным функционально полным набором логических элементов.

Комбинационная схема с несколькими выходами всегда может быть представлена в виде совокупности схем, каждая из которых обладает лишь одним выходом, что позволяет свести решение задачи синтеза схем с произвольным числом выходов к решению задачи синтеза комбинационной схемы с одним выходом. Работа комбинационной схемы полностью описывается либо таблицей истинности, либо булевым выражением.

В цифровой технике широко применяется ограниченное число комбинационных схем, выполненных в виде интегральных схем малой и средней степени интеграции, которые используются практически в любом цифровом блоке. Так как все комбинационные схемы образуют единый класс, то дальнейшая их клас-

сификация может быть выполнена по функциональному признаку. Однако необходимо отметить, что такое разбиение схем условно, так как все конкретные комбинационные схемы многофункциональны и многие из них могут быть отнесены к различным подклассам. Определять и относить схемы к какому-либо подклассу принято по ***основному функциональному назначению***. Сами подклассы со сходными признаками объединяются в группы. К настоящему времени сложилась следующая система классификации комбинационных схем.

• **Схемы общего назначения** — к ним относят, как правило, схемы малой степени интеграции, такие как «И»; «И-НЕ»; «ИЛИ»; «ИЛИ-НЕ»; «И-ИЛИ-НЕ» и др. Они образуют ***технически полную систему элементов***, т.е. систему, удовлетворяющую требованиям функциональной и физической полноты. ***Функционально полная система элементов*** — система, позволяющая реализовать любые, сколь угодно сложные переключательные функции путём суперпозиции простейших функций. ***Физически полная система элементов*** — система, содержащая специальные элементы, обеспечивающие возможность построения управляющих цепей, запоминающих устройств и цепей связи, восстанавливающих информационные сигналы до стандартной амплитуды и формы и обеспечивающих надёжное взаимодействие элементов при всевозможных комбинациях связи между ними, элементы, обеспечивающие работу электромеханических узлов типа реле, переключателей, механизмов печати и т.п., а также схем связи различных узлов цифрового устройства с устройствами ввода-вывода, элементы индикации информационных состояний цифрового устройства и генераторы тактовых сигналов.

• **Схемы контроля и индикации** — это фактически схемы общего назначения, выполненные с открытым коллектором (эмиттером, стоком), которые позволяют подключать внешние дискретные элементы контроля и индикации (лампы, светодиоды, реле, исполнительные механизмы и т.п.) для определения состояния цифровых цепей в блоке.

• **Преобразователи кодов —** дешифраторы, детекторы состояний, шифраторы, приоритетные шифраторы, специализированные преобразователи кодов, ПЗУ и др.

• **Коммутационные узлы —** «идеальные» ключи (реле, герконы и т.п.), диодные и транзисторные ключи, двунаправленные полупроводниковые ключи, селекторы, мультиплексоры, универсальные селекторы-мультиплексоры и др.

• **Арифметические узлы —** схемы контроля на чётность или нечётность, четверть-сумматоры, полусумматоры, сумматоры, схемы ускоренного переноса, арифметико-логические устройства, компараторы, адресные компараторы, матричные цифровые умножители, схемы формирования кода Хемминга и др.

• **Специальные узлы** (некоторые из них не являются чисто комбинационными) — логические расширители, генераторы, одновибраторы, таймеры, буферные усилители, преобразователи уровней, триггеры Шмидта, разностные элементы (детекторы событий), пороговые (в том числе мажоритарные) элементы и др.

По способу кодирования двоичных переменных элементы цифровых устройств подразделяют на импульсные, динамические, потенциальные, импульсно-потенциальные и фазовые [14]. В настоящее время в цифровой технике наиболее широко используются потенциальные системы элементов. Независимо от способа кодирования информации в любых схемах должен соблюдаться ***принцип совместимости входных и выходных сигналов***, означающий, что ориентировка уровней

«0» и «1» для них должна совпадать в определённой зоне значений, отображающих «0» и «1». Принцип совместимости входных и выходных сигналов должен выполняться при воздействии на элемент нагрузок и дестабилизирующих факторов, количество и параметры которых определяются техническими условиями на элементы. К дестабилизирующим факторам относят изменение питающих напряжений, разброс параметров и характеристик компонентов, изменение температуры окружающей среды, наличие электромагнитных излучений, механические воздействия и т.п.

Особенность работы с интегральными схемами в том, что контролировать их характеристики и параметры невозможно путём измерения параметров их отдельных компонентов. ***Параметры интегральных схем могут быть определены только по входным, передаточным и выходным характеристикам*** [14]. Существующие методы определения параметров интегральных схем основываются на выполнении функционального контроля и параметрических измерений.

Для логических элементов под ***функциональным контролем*** подразумевается процедура определения вида переключательной функции, которую реализует логический элемент, а под ***параметрическими измерениями*** — процедура определения индивидуальных числовых значений параметров элемента, таких как уровни «1» и «0», входные и выходные токи и т.п. Именно по результатам параметрических измерений делается вывод о пригодности к использованию данного логического элемента. ***Работоспособность логического элемента*** — правильная передача информационных сигналов при одновременном выполнении заданных техническими условиями требований к числовым значениям параметров.

При проектировании цифровой аппаратуры необходимо знать основные статические, динамические, схемотехнические и конструктивные параметры и характеристики, подробно описанные в литературе, например в [14].

Отметим особенности построения выходных цепей логических элементов. Выходные цепи могут быть реализованы в трёх модификациях: выход стандартного элемента, выход с открытым коллектором (эмиттером, стоком) и выход с тремя состояниями.

Элемент со ***стандартным выходом*** реализует два значения выходного сигнала — логические уровни «0» и «1». Выходы таких элементов не допускается соединять вместе и тем более подключать их к шинам питания и земли, так как выходные транзисторы таких элементов могут быть повреждены (это не относится к некоторым элементам, выполненным по КМДП-технологии).

Упрощённая схема двухвходового логического элемента, реализованного в базисе транзисторно-транзисторной логики (ТТЛ) для функции «**И-НЕ**» ***с открытым коллектором*** (ОК), представлена на рис. 2.15, *а*. Такой элемент на выходе формирует два состояния: логический «0» и обрыв.

Для формирования стандартного сигнала логической «1» к открытому коллектору должен быть подключен внешний резистор *R* (рис. 2.15, *б),* номинал которого определяется требуемыми статическими и динамическими параметрами схемы. Вместо резистора могут быть включены элементы индикации (лампа, светодиод с токоограничивающим резистором, обмотка реле и т.п.). Выходной транзистор многих элементов с открытым коллектором имеет повышенный допустимый ток, а иногда и повышенное допустимое коллекторное напряжение. Выход с ОК имеют не только простейшие логические элементы, но и сложные функциональные узлы,

такие как дешифраторы, мультиплексоры, элементы памяти и др. Выход с ОК помечают специальным значком — ромбиком с чертой внизу (рис. 2.15, *в*).

Упрощённая схема двухвходового логического элемента, реализованного в базисе эммитерно-связанной логики (ЭСЛ) для функции «**ИЛИ**» ***с открытым эмиттером*** (ОЭ) приведена на рис. 2.16, *а*. Такой элемент на выходе формирует два состояния: логическую «1» и обрыв.

Для формирования логического «0» к открытому эмиттеру должен быть подключен внешний резистор *R* (рис. 2.16, б). Выход элемента с ОЭ помечается специальным знаком — ромбиком с чертой сверху (рис. 2.16, в).

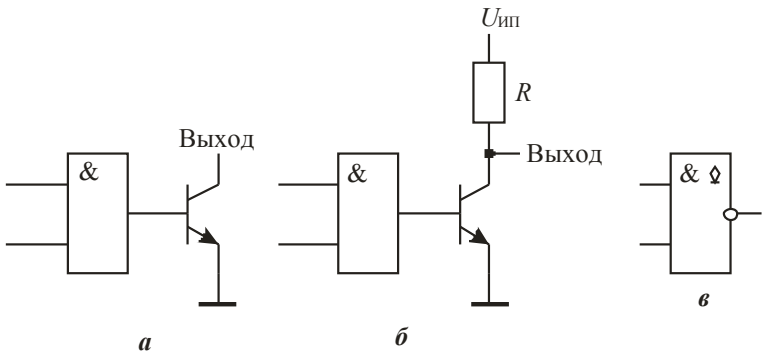

Рис. 2.15. Упрощённая схема вывода логического элемента: *а* — ТТЛ с открытым коллектором; *б* — ТТЛ с открытым коллектором и резистором, формирующим уровень «1»; *в* — функциональное обозначение ТТЛ с открытым коллектором

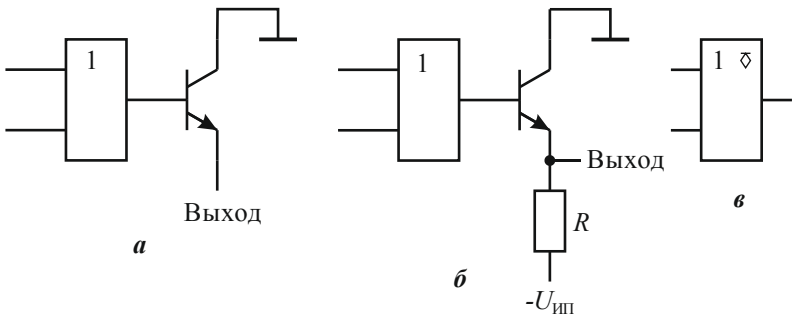

Рис. 2.16. Упрощённая схема выхода логического элемента: *а* — ЭСЛ с открытым эмиттером (ОЭ); *б* — ЭСЛ с ОЭ и резистором, формирующим уровень «0»; *в* — функциональное обозначение ЭСЛ с открытым эмиттером

Элементы с ОК и ОЭ в отличие от стандартных элементов позволяют объединять выходы в единую цепь. На рис. 2.17 показано объединение выходов двух двухвходовых ТТЛ-элементов «И-НЕ».

В дальнейшем будем предполагать, что логические переменные кодируются в положительной логике. *Для отдельно взятых элементов* выполняются соотношения

$$y_1 = \overline{x_3 x_2}; \qquad (2.25)$$

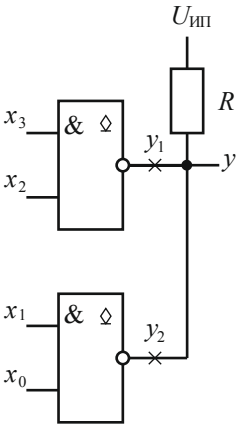

Рис. 2.17. Объединение выходов двух ТТЛ-элементов с ОК

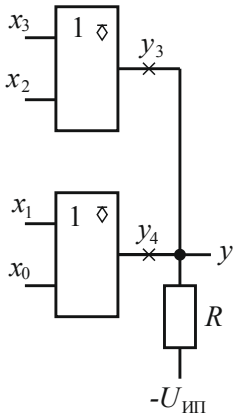

Рис. 2.18. Объединение выходов двух ЭСЛ-элементов с ОЭ

$$y_2 = \overline{x_1 x_0}. \qquad (2.26)$$

В свою очередь, для выходного сигнала *y,* зависящего от переменных y_1 и y_2, выполняется соотношение

$$y = y_1 y_2. \qquad (2.27)$$

Действительно, уровень «1» на выходе *y* будет только в том случае, когда закрыты оба транзистора в двух элементах, т.е. ***объединение в единую цепь выходов элементов с ОК реализует монтажную операцию И для переменных, рассматриваемых на отдельно взятых выходах.***

Подставляя выражения (1.1) и (1.2) в формулу (1.3), имеем

$$y = \overline{x_3 x_2} \cdot \overline{x_1 x_0} = \overline{\overline{x_3 x_2} \cdot \overline{x_1 x_0}} = \overline{x_3 x_2 + x_1 x_0},$$

т.е. получается эквивалентный элемент типа «И-ИЛИ-НЕ».

На рис. 2.18 показано объединение выходов двух двухвходовых ЭСЛ-элементов «ИЛИ».

Для отдельно взятых элементов выполняются соотношения

$$y_3 = x_3 + x_2; \qquad (2.28)$$

$$y_4 = x_1 + x_0. \qquad (2.29)$$

В свою очередь, для выходного сигнала *y*, зависящего от переменных y_3 и y_4, выполняется соотношение

$$y = y_3 + y_4. \qquad (2.30)$$

Действительно, уровень «0» на выходе *y* будет только в том случае, когда оба транзистора в двух элементах закрыты, т.е. объединение в единую цепь выходов элементов с ОЭ реализует монтажную операцию ИЛИ для переменных, рассматриваемых на отдельно взятых выходах.

Подставляя выражения (2.28) и (2.29) в уравнение (2.30), получаем

$$y = x_3 + x_2 + x_1 + x_0.$$

Рекомендуется самостоятельно определить выходной сигнал в объединённых схемах с ОЭ, отдельно реализующих операцию «ИЛИ-НЕ».

Схемы с ОК и ОЭ используются либо как элементы контроля и индикации, либо как буферные усилители для подключения различных блоков к общей информационной шине.

На рис. 2.19, *а* показана *качественная модель* трёх состояний выходной цепи *элемента с тремя состояниями выхода*, а обозначение соответствующего инвертора — на рис. 2.19, *б*. Выход с тремя состояниями для логического элемента помечается специальным знаком — ромбиком с чертой внутри него. У элемента с тремя состояниями выхода всегда присутствует управляющий сигнал *z*. Когда на входе *z* активный сигнал, то инвертор на рис. 2.19 реализует функцию $y = \overline{x_0}$, а когда неактивный сигнал, то выход инвертора переводится в третье состояние, т.е. реализуется полный обрыв, когда элемент не выдаёт ток в нагрузку и не принимает его от нагрузки.

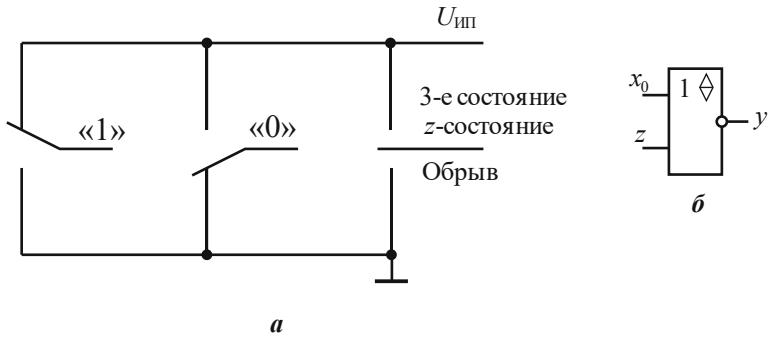

Рис. 2.19. Модель элемента с тремя состояниями выхода: *а* — переключательная схема; *б* — функциональное обозначение инвертора

Элементы с тремя состояниями выхода специально разработаны для применения в качестве выходного управляемого буфера для подключения различных цифровых блоков к общим информационным шинам и магистралям. Важно помнить, что подача двух и более активных сигналов *z* различных буферов магистрали ***недопустима***, так как результат будет такой же, как и при объединении выходов стандартных логических элементов.

В заключение рассмотрим синтез комбинационной схемы, имеющей определенный практический интерес.

Постановка задачи. Двери двух квартир выходят в коридор, не имеющий окон, а в конце коридора находится дверь, ведущая на улицу. Коридор освещается одной лампочкой. У дверей двух квартир и двери, ведущей на улицу, установлены выключатели. Требуется разработать схему, которая позволяла бы включить и выключить лампочку любым выключателем. Например, человек, выходя из квартиры, может включить лампочку у своей двери, а выходя на улицу, выключить её у выходной двери и, наоборот, входя во входную дверь, включить лампочку, а у своей квартиры выключить. Если лампочка уже была включена, то её можно выключить либо у своей квартиры, либо у выходной двери.

Решение задачи. Задачу решим двумя способами: во-первых, разработаем логическую схему, формирующую выходной сигнал *y* в предположении, что им управляется реле, контакты которого включают или выключают лампочку, и, во-вторых, разработаем схему только с использованием *переключателей*.

Для решения задачи нужно составить таблицу истинности (табл. 2.5). Обозначим входные логические переменные, значения которых устанавливаются

Таблица 2.5. Таблица истинности реализуемой функции

Номер набора	x_2	x_1	x_0	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

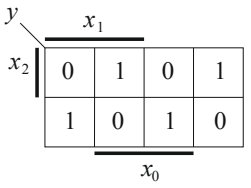

Рис. 2.20. Карта Карно реализуемой функции

выключателями у дверей, через x_2, x_1 и x_0. Будем считать, что верхнее положение выключателей устанавливает значение «1» для соответствующей переменной, а нижнее — «0». Примем, что когда все выключатели находятся в нижнем положении, то лампочка не горит ($y = 0$). Очевидно, что установка в верхнее положение любого одного выключателя включит лампочку ($y = 1$). Установка в верхнее положение любых двух выключателей выключит лампочку ($y = 0$), а установка в верхнее положение всех трёх выключателей включит лампочку ($y = 1$).

На основании этих простых рассуждений составляем таблицу истинности. Представим полученную функцию в виде карты Карно (рис. 2.20) [15].

Так как все единичные клетки в карте Карно являются изолированными, то минимальная дизъюнктивная форма совпадает с совершенной нормальной дизъюнктивной формой. Из таблицы истинности или из карты Карно получаем

$$y = \bar{x}_2\bar{x}_1x_0 + \bar{x}_2x_1\bar{x}_0 + x_2\bar{x}_1\bar{x}_0 + x_2x_1x_0. \qquad (2.31)$$

Из формулы (1.7) следует, что для реализации логического блока требуются три инвертора, четыре трёхвходовых элемента И и один четырёхвходовый элемент ИЛИ. Уменьшим аппаратные затраты, преобразовав выражение (2.31):

$$\begin{aligned} y &= (\bar{x}_2\bar{x}_1x_0 + x_2x_1x_0) + (\bar{x}_2x_1\bar{x}_0 + x_2\bar{x}_1\bar{x}_0) = \\ &= (\bar{x}_2\bar{x}_1 + x_2x_1)x_0 + (\bar{x}_2x_1 + x_2\bar{x}_1)\bar{x}_0 = \\ &= \overline{(x_2 \oplus x_1)}x_0 + (x_2 \oplus x_1)\bar{x}_0 = x_2 \oplus x_1 \oplus x_0. \end{aligned} \qquad (2.32)$$

Из формулы (2.32) следует, что логический блок для своей реализации потребует один трёхвходовый или два двухвходовых элемента «сумма по mod 2». Ясно, что последняя реализация логического блока предпочтительнее.

Второе решение получаем, воспользовавшись переключательными моделями функций алгебры логики [16]. Схема приведена на рис. 2.21, где все переключатели показаны в нижнем положении ($y = 0$). Из рис. 2.21 видно, что необходимо использовать два одиночных переключателя и один сдвоенный, причём в цепи, соединяющей переключатели, требуется двухпроводная линия связи. Данную схему нетрудно расширить до любого числа переменных.

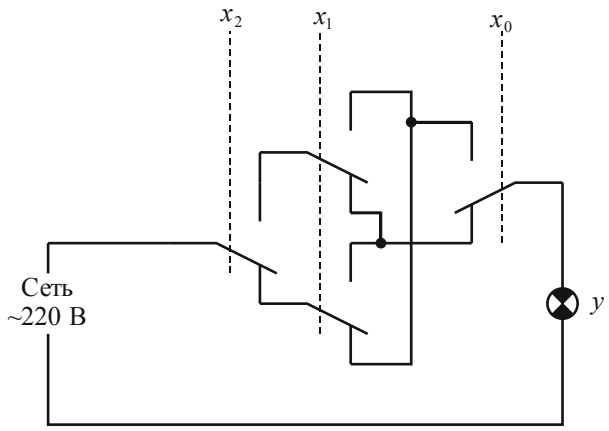

Рис. 2.21. Схема управления лампой тремя переключателями

2.2. Лабораторное задание

1. Выполнить синтез структур комбинационных схем, заданных в индивидуальном задании, построить временные диаграммы работы.

2. Разработать синтезированные схемы в среде САПР БИС «Ковчег 3.04».

3. Выполнить моделирование работы схемы и убедиться, что результат моделирования совпадает с расчётными временными диаграммами, построенными на этапе синтеза схем.

2.2.1. Пример индивидуального задания

Используя карты Карно, минимизировать функции алгебры логики, заданные числовым представлением СКНФ или СДНФ:

$y_1 = \wedge(3,6,7,12-15;\ \times: 2,9,11);$

$y_2 = \vee(5,7,8,13,15;\ \times: 0-3,9-11);$

$y_3 = \wedge(0,4,25,27,29,31;\ \times: 1,2,6,19,23,26,28,30);$

$y_4 = \vee(0,1,3,7,8,9,11,15,21,24,25,27,31;\ \times: 5,13,16,17,19,23,26,28,29,30).$

2.2.2. Порядок выполнения работы на примере выполнения индивидуального задания

Для минимизации ФАЛ, заданных в индивидуальном задании, воспользуемся эталонными картами Карно для ФАЛ четырёх переменных (для y_1, y_2) и пяти переменных (для y_3, y_4), которые представлены на рис. 2.11, *а* и рис. 2.11, *б* соответственно.

В индивидуальном задании в скобках указаны номера наборов, в которых функция должна принимать значение логического «0», если ФАЛ задана числовым представлением СКНФ (символ « $\wedge$ » перед скобкой), и значение логической «1», если ФАЛ задана числовым представлением СДНФ (символ « $\vee$ » перед скобкой). Знак «×» ставится в тех наборах, которые указаны в скобках после «×», оста-

вшиеся пустые клетки заполняются «1», если ФАЛ задана числовым представлением СКНФ, и «0», если ФАЛ задана числовым представлением СДНФ.

Заполнив все клетки нужными значениями, получим рабочие карты Карно, показанные на рис. 2.22.

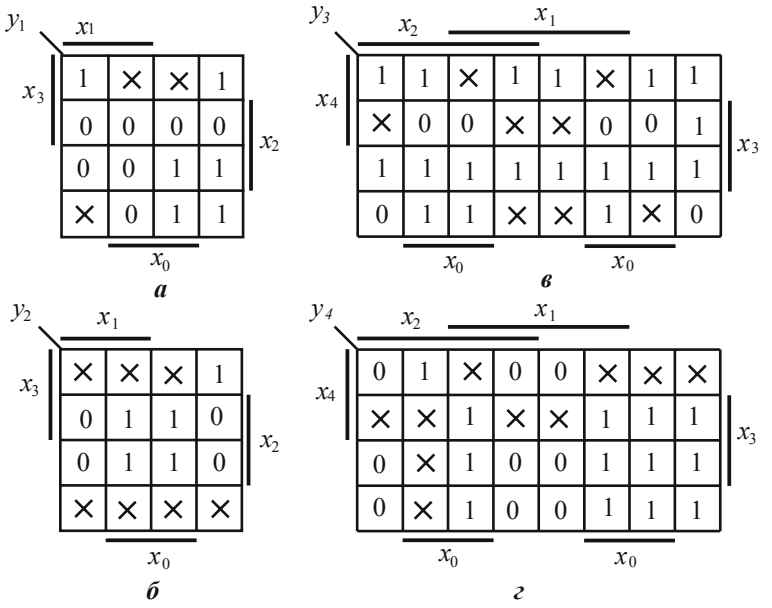

Рис. 2.22. Рабочие карты Карно для функций алгебры логики:
а — y_1; *б* — y_2; *в* — y_3, *г* — y_4

При минимизации ФАЛ в формате ДНФ с помощью карт Карно выбираем набор покрытий (прямоугольная область, состоящая из 1, 2, 4, 8 или 16 клеток), которые покрывают все единичные клетки в карте и не покрывают ни одну клетку, содержащую логический «0». Покрытия должны иметь максимальную площадь, они также могут пересекаться. Клетки, отмеченные знаком «×», могут попадать в покрытия, а могут и остаться непокрытыми. Знак «×» находится в тех номерах наборов, в которых разработчику безразлично, какое значение примет функция — «0» или «1». После того как набор покрытий выбран (функция минимизирована), все номера наборов, в которых стоял знак «×», доопределились до конкретных значений логического «0» или логической «1»: если клетка, содержащая «×», попала хотя бы в одно покрытие, её значение стало логической «1», если клетка не попала ни в одно покрытие, то её значение доопределилось до логического «0».

Для минимизации функций алгебры логики выбираем покрытия, показанные на рис. 2.23.

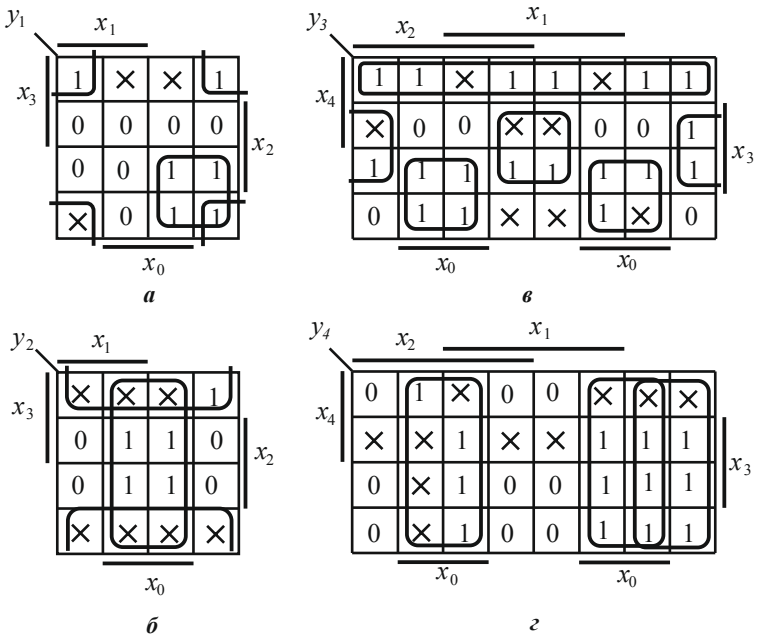

Рис. 2.23. Рабочие карты Карно с покрытиями для:
а — y_1; *б* — y_2; *в* — y_3, *г* — y_4

Минимизируя функции, получаем следующие аналитические выражения для ФАЛ y_1, y_2, y_3, y_4:

$$y_1 = \overline{x_2} \cdot \overline{x_0} + \overline{x_3} \cdot \overline{x_1};$$
$$y_2 = \overline{x_2} + x_0;$$
$$y_3 = x_4 \cdot \overline{x_3} + x_3 \cdot \overline{x_0} + \overline{x_4} \cdot x_0;$$
$$y_4 = x_0 + \overline{x_2} \cdot \overline{x_1}.$$

По полученным ФАЛ составим временные диаграммы работы, которые показаны на рис. 2.24. Временные диаграммы можно составлять как по аналитическим выражениям для y_1, y_2, y_3, y_4, так и по картам Карно. Действительно, «0» или «1» в клетках карты Карно однозначно показывают значение функции в каждом из наборов. А каждая клетка, содержащая «×», доопределилась до логической «1», если попала хотя бы в одно из покрытий, либо до логического «0», если осталась непокрытой.

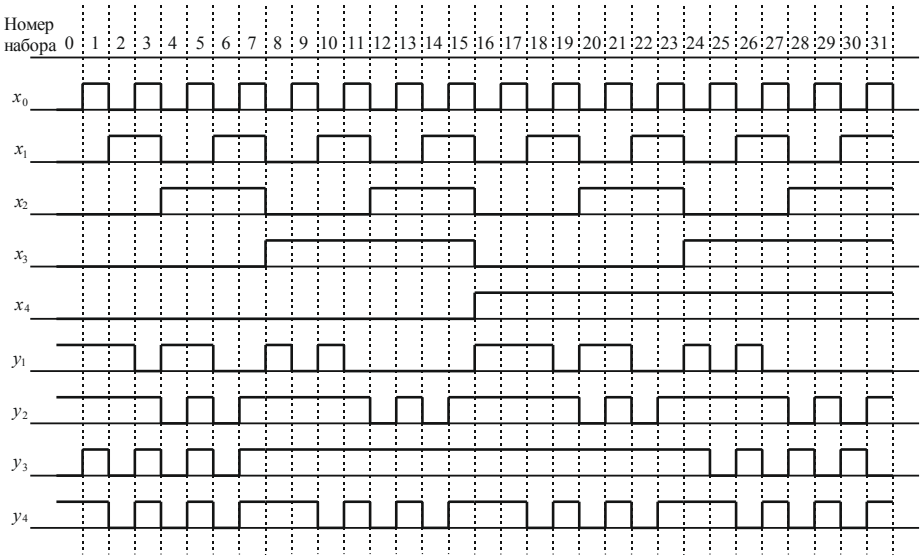

Рис. 2.24. Временные диаграммы работы минимизируемых функций

Реализуем электрическую схему полученных функций в среде САПР БИС «Ковчег 3.04». Для этого необходимо запустить программу (кнопка «**Пуск**» → «**Все программы**» → НПК «Технологический центр» → *Kovcheg*5503 → *Kovcheg*). В результате открывается окно «**САПР БИС «Ковчег 3.04**». Далее необходимо создать проект. Для этого в меню «**Проект**» активизируем команду «**Новый проект...**» (рис. 2.25):

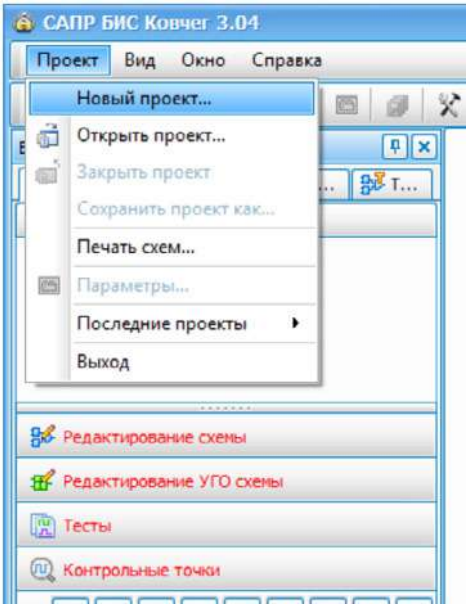

Рис. 2.25. Создание нового проекта в САПР БИС «Ковчег 3.04»

Далее необходимо настроить создаваемый прооект. В первом окне предлагается задать имя проекта *lab*2, имя головной схемы *MAIN* и его размещение. Укажите данные параметры и нажмите кнопку «**Далее**» (рис. 2.26):

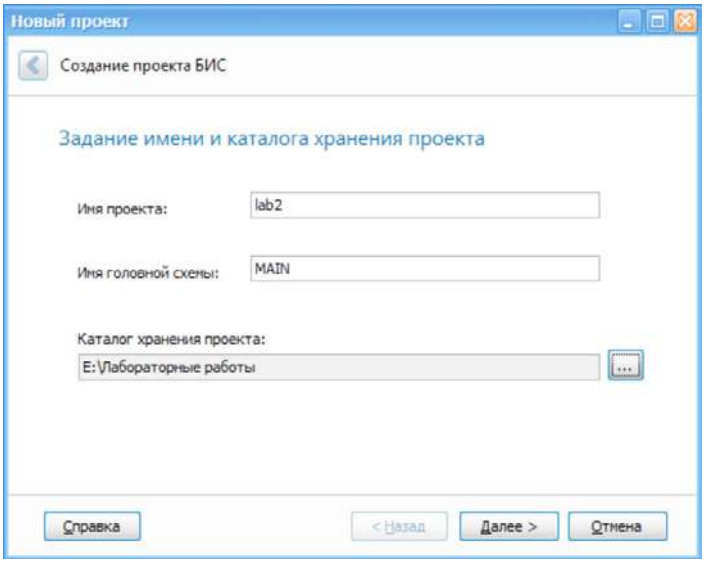

Рис. 2.26. Задание имени и расположения проекта

Далее необходимо выбрать тип БМК и используемые для разработки библиотеки ячеек. Выберите «**H55033XM1**» во вкладке «**БМК**», во вкладке «**Библиотека**» отметьте «**5503**» (во вкладке «**Имитатор**» доступен «**5503XC1**») (рис. 2.27).

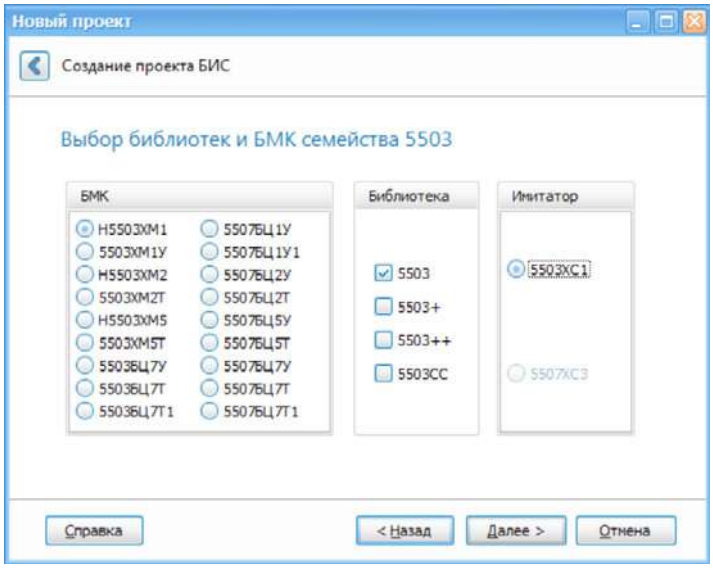

Рис. 2.27. Выбор типа БМК и библиотеки ячеек

В реквизитах проекта обычно указывается контактная информация, в нашем случае укажите произвольные сведения (например, как показано на рис. 2.28).

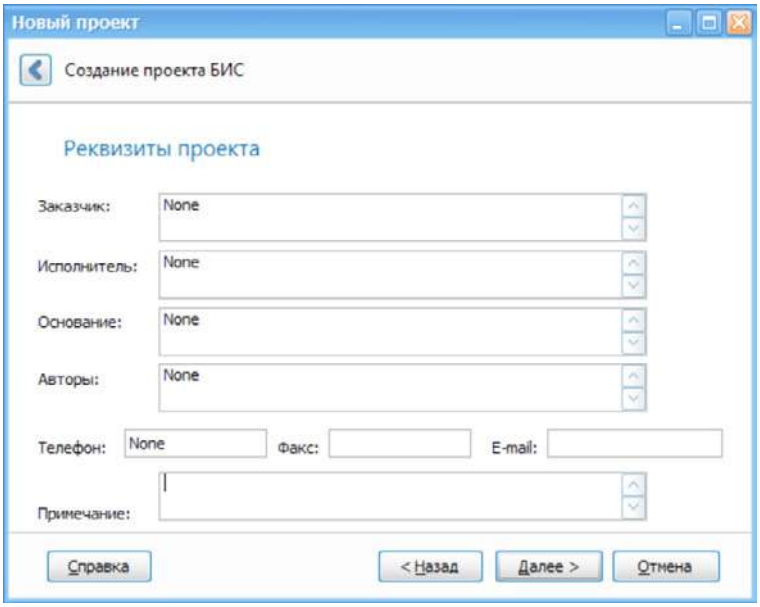

Рис. 2.28. Реквизиты проекта

Далее автоматически формируется окно с заданными параметрами создаваемого проекта для проверки. Убедитесь, что всё было задано верно и нажмите кнопку «**Создать**» (рис. 2.29).

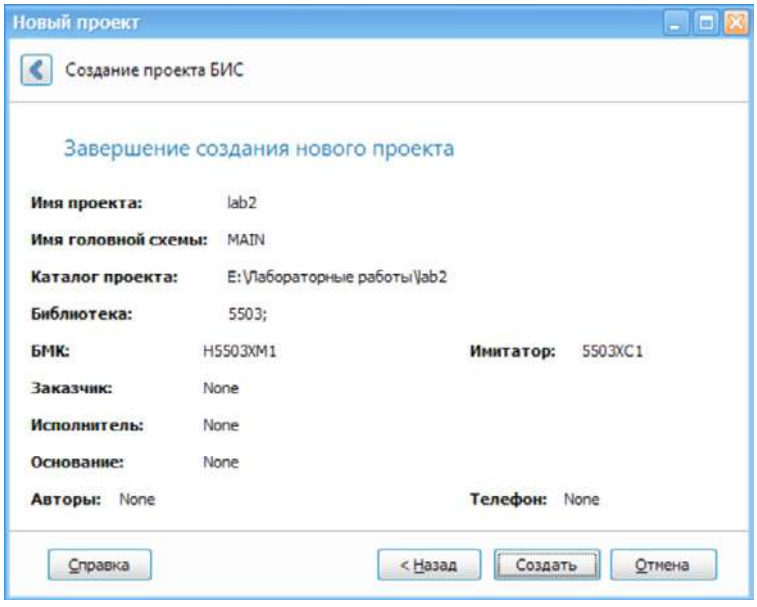

Рис. 2.29. Итоговая информация о проекте

Перейдём к созданию схемы. На панели инструментов найдите кнопку «**Создать схему**», в выпадающем меню выберите пункт «**Графический формат**» (рис. 2.30).

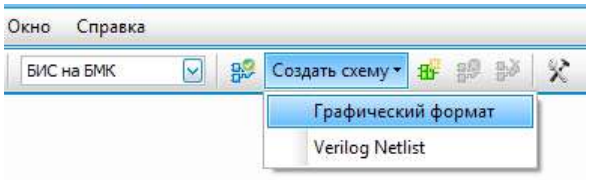

Рис. 2.30. Выбор формата представления создаваемой схемы

Далее в появившемся окне в поле «**Имя схемы**» задайте имя *MAIN* создаваемой графической схемы (рис. 2.31). Следует отметить, что при задании имени регистр (заглавные или строчные буквы) значения не имеет.

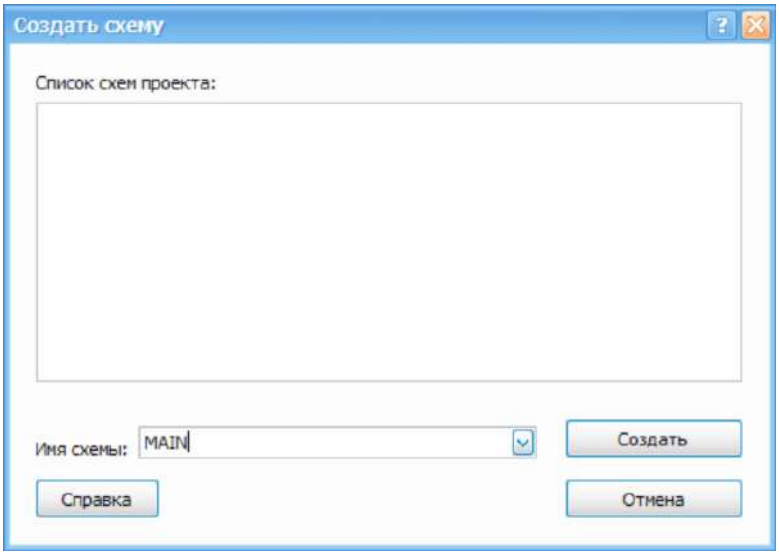

Рис. 2.31. Задание имени создаваемой схемы

После этого открывается окно графического схемного редактора с именем создаваемой схемы. Перейдём к построению схемы. На панели инструментов найдите кнопку «**Порт групповой…**» (рис. 2.32), обеспечивающую создание графического образа входного порта.

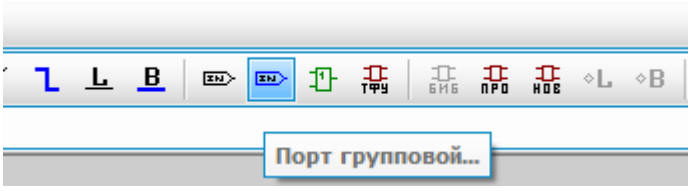

Рис. 2.32. Создание группового порта

В появившемся окне (рис. 2.33) в поле «**Имя порта**» задайте имя «*X*», старший разряд «4» по номеру самой старшей переменной, используемой в синтезированных схемах. Младший разряд — соответственно 0:

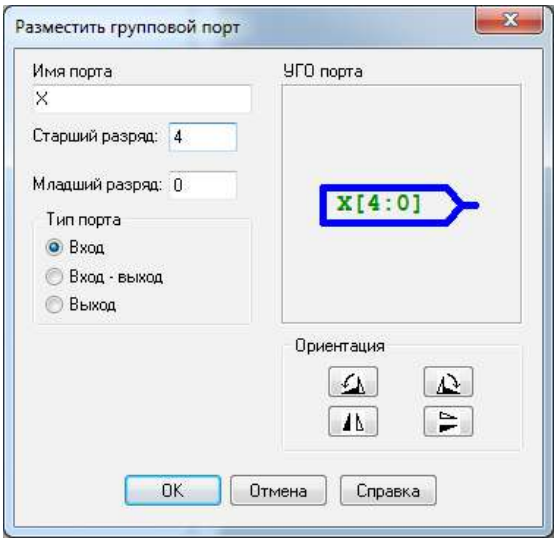

Рис. 2.33. Окно настроек группового порта

Разместите порт на рабочем поле (рис 2.34)

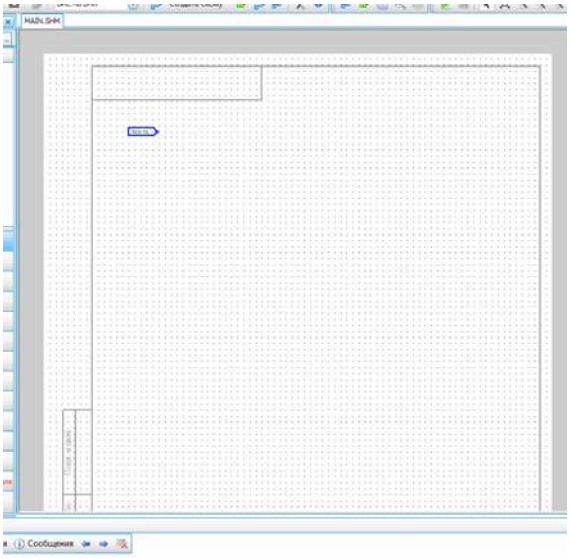

Рис. 2.34. Размещение порта в поле схемного редактора

Далее необходимо нарисовать шину после нажатия кнопки «**Шина**» (рис. 2.35) и затем создать «Метку шины» на только что созданной шине.

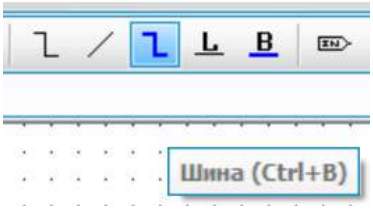

Рис. 2.35. Кнопка «**Шина**» на панели инструментов

Для этого нажмите кнопку «**Метка шины**» в виде подчёркнутой буквы «**B**» на панели инструментов (рис. 2.36).

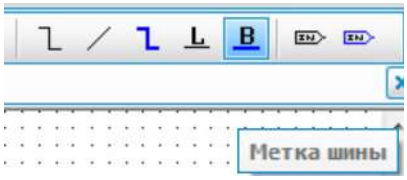

Рис. 2.36. Кнопка «**Метка шины**» на панели инструментов

В открывшемся окне задайте имя и соответствующие разряды, затем разместите метку на шине (рис. 2.37).

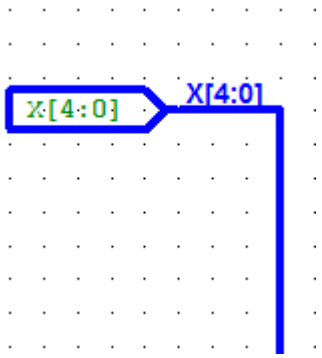

Рис. 2.37. Размещённая метка шины

Далее задайте входные сигналы шины. Для этого выберите инструмент «**Связь ортогональная**» на панели инструментов (первая кнопка в той же группе, что и «**Метка соединения**» и «**Метка шины**», рис. 2.36). Начертите ответвлетвения от шины и нажатием правой кнопки мыши выберите в выпадающем меню пункт «**Задать метку**» и задайте именование соединений (рис. 2.38).

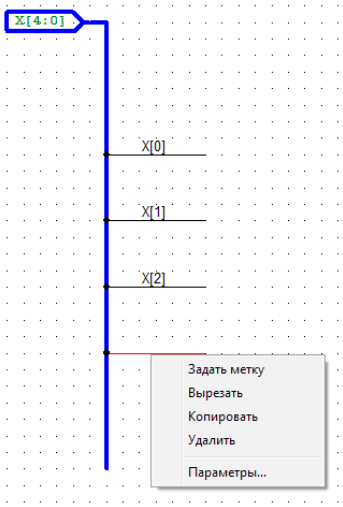

Рис. 2.38. Задание метки соединения

Далее необходимо создать выходные порты схемы. Для этого на панели инструментов нажмите кнопку «**Порт одиночный**...». Задайте имя и выберите тип порта «**Выход**» (рис. 2.39).

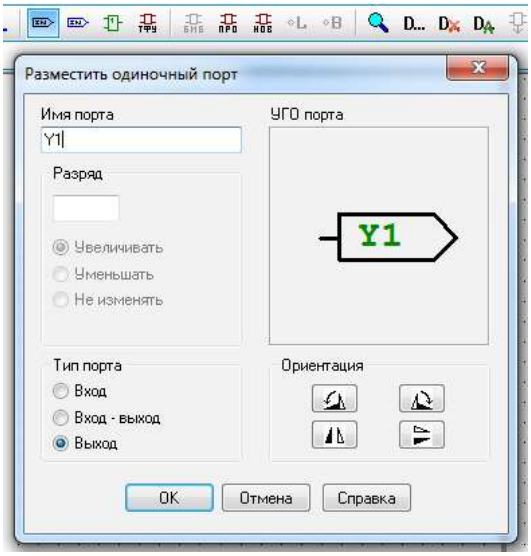

Рис. 2.39. Окно размещения одиночного порта

Разместите на схеме четыре выхода (быстро переименовывать порты можно двойным нажатием ЛКМ на порту). Далее перейдём к реализации основной логики схемы. Для этого, во-первых, необходимо раместить «драйверы» входов и выходов схемы, во-вторых, сами логические эелементы, реализующие работу создаваемой схемы. Все доступные функциональные ячейки можно найти, нажав внопку «**Ячейка**» на панели инструментов (рис. 2.40) или выбрав вкладку «Базовые ячейки» в области панелей быстрого доступа (рис. 2.41).

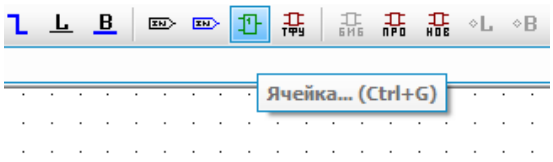

Рис. 2.40. Кнопка «**Ячейка**» на панели инструментов

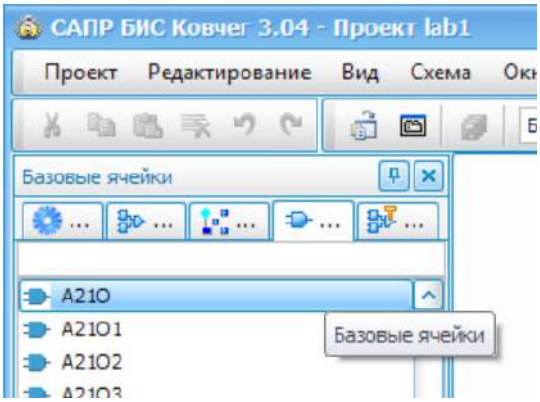

Рис. 2.41. Вкладка «Базовые ячейки»

При нажатии кнопкии «**Ячейка…**» будет появляться окно, где все логические ячейки поделены на различные группы. Разместим драйверы входа/выхода. Для этого надо перейти в группу «**Периферийные ячейки/входные**» и выбрать ячейку под названием *IDP* (рис. 2.42), разместить их на входной шине нажатием ЛКМ (рис. 2.43).

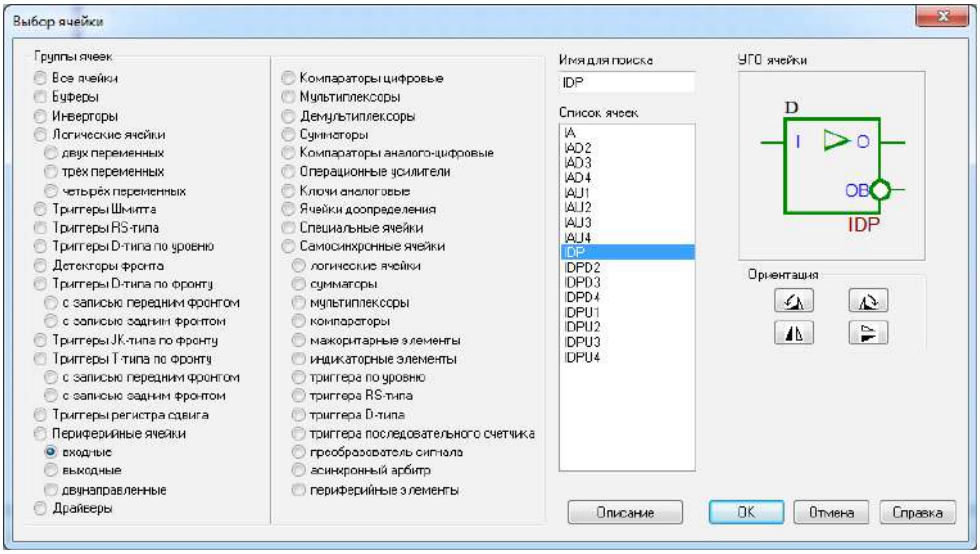

Рис. 2.42. Выбор ячейки «*IDP*»

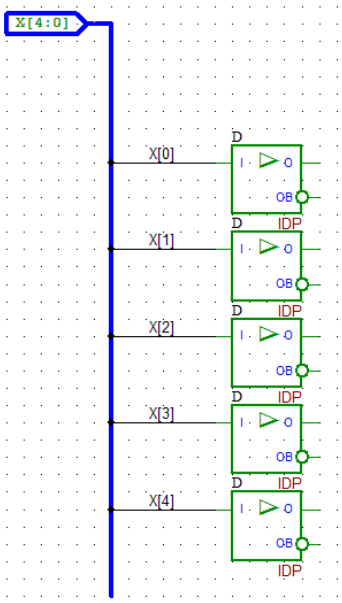

Рис. 2.43. Размещение ячеек в рабочем поле

Аналогичным образом в категории «**Периферийные ячейки/Выходные**» нужно найти ячейку «*OA*» и разместить её на выходах схемы (рис. 2.44).

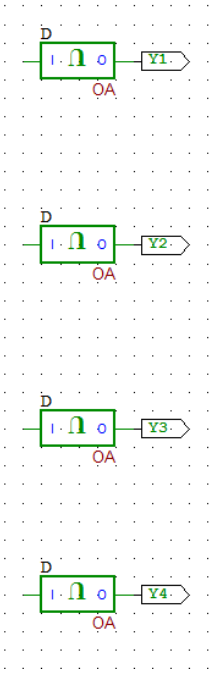

Рис. 2.44. Выходные порты схемы

Следует отметить, что в реальных микросхемах выход *ОА* используется только для обеспечения выдачи аналоговых сигналов из микросхемы. Для организации выхода цифрового сигнала применяются библиотечные ячейки драйверов с периферийными ячейками, которые выбираются в зависимости от требуемой функции и параметров выхода. В библиотеке 5503 реализованы восемь типов драйверов и десять типов выходных цифровых периферийных ячеек. Подробно указанные ячейки описаны в книге 3 «Библиотека функциональных ячеек для проектирования полузаказных микросхем серии 5503 и 5507», а правила их применения в книге [1] «Методология проектирования и освоение производства» (библиографические данные приведены в списке литературы). Для упрощения в лабораторных работах будем применять выходные ячейки *ОА*.

Далее реализуем основную логику. Все логические элементы в разных их комбинациях располагаются в категории «**Логические ячейки**» с указанием числа переменных. Каждая категория включает в себя полный перебор всех логических элементов (логическое сложение «*OR*2», логическое умножение «*AND*2», сложение по модулю два «*XOR*2», инверсия «*NOT*») с учётом инверсии входных сигналов. Далее, внимательно следуя синтезированным формулам, строим схему. Результат представлен на рис. 2.45.

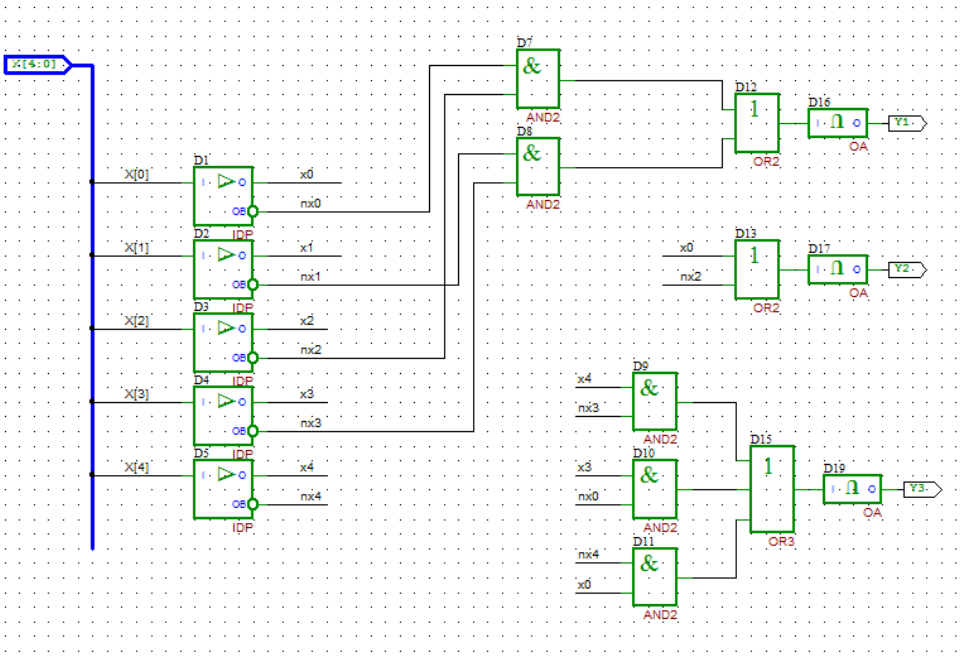

Рис. 2.45. Созданная схема

Последним шагом перед трансляцией схемы необходимо осуществить нумерацию УГО. Перейдите в меню «**Редактор**» и выберите функцию «**Автоматически нумеровать УГО**» (рис. 2.46).

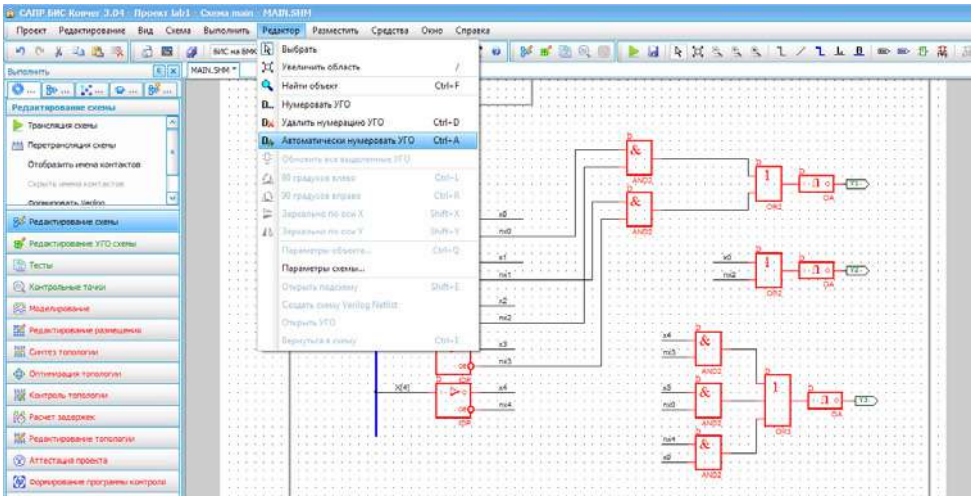

Рис. 2.46. Команда «Автоматически нумеровать УГО»

Схемный редактор САПР БИС «Ковчег 3.04» позволяет добавить к схеме графические (линия и прямоугольник) или текстовые комментарии. Для их введения нужно использовать соответствующие кнопки с изображением вертикальной линии, прямоугольника и буквы Т, находящиеся в конце панели инструментов (рис. 2.47).

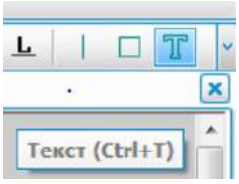

Рис. 2.47. Кнопки задания графических и текстовых комментариев на панели инструментов

Укажем имя созданной схемы для её более удобной идентификации. Для этого необходимо на панели инструментов нажать кнопку «**Текст**» (рис. 2.47), после чего появится окно «**Разместить текст**», в котором можно задать сам текст, а также его шрифт, ориентацию и выравнивание.

Переходим к проверке созданной схемы на ошибки. Для этого необходимо выбрать либо команду «**Трансляция схемы**» в верхней части области панелей быстрого доступа, либо изображение зелёной стрелки в панели инструментов. В нижней части экрана появится информационное окно с результатами трансляции схемы (рис. 2.48).

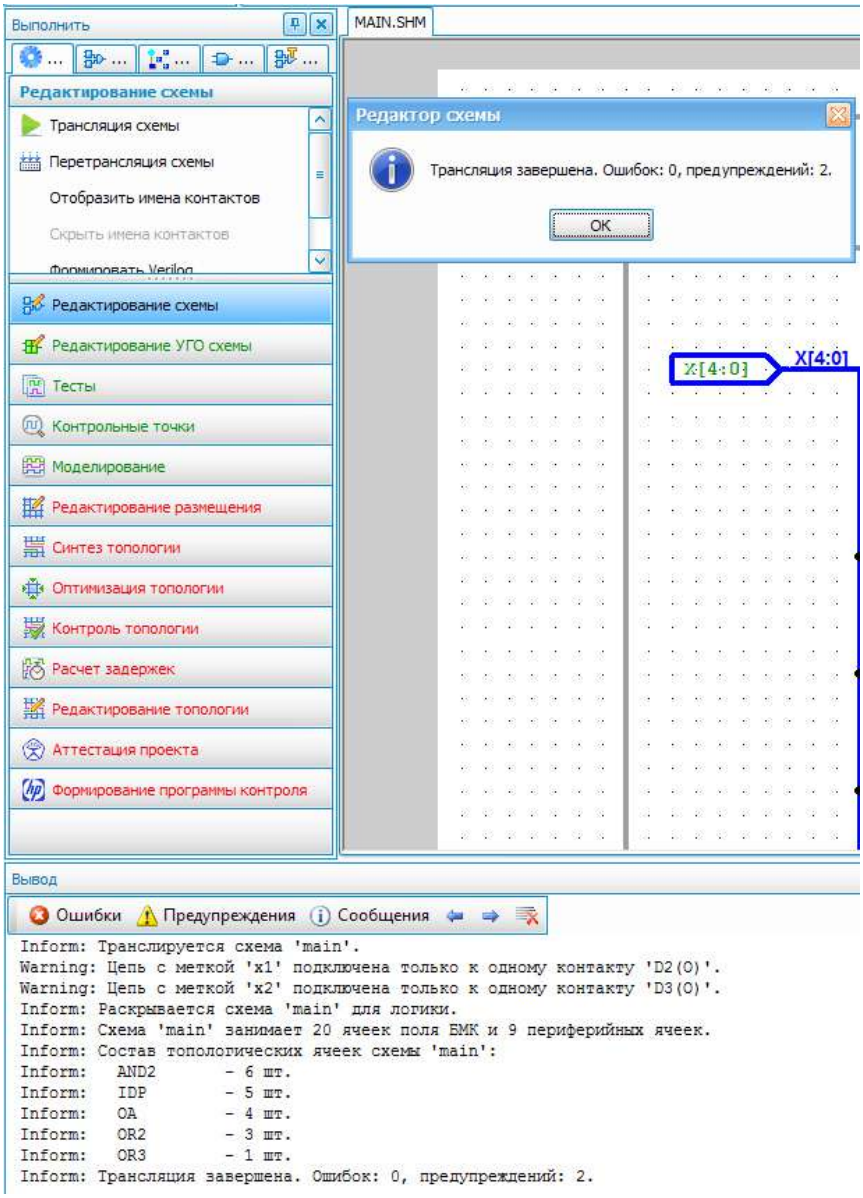

Рис. 2.48. Информация о трансляции схемы

Далее необходимо описать «**Контрольные точки**». Ими могут быть любые сигналы (входные, выходные, внутренние), которые будут отображаться при моделировании схемы. Выберите команду «**Контрольные точки**» в нижней части области панелей быстрого доступа, при этом автоматически формируется файл контрольных точек с примером их описания, который открывается в окне текстового редактора (рис. 2.49). Для нашей задачи интересно отобразить шины входных сигналов «*X*[4..0]» и выходных сигналов «*Y*1, *Y*2, *Y*3, *Y*4», для того чтобы сравнить их

с теоретической временной диаграммой. Добавьте в файл следующий текст (текстовый редактор САПР БИС «Ковчег 3.04» допускает описания на кириллице):

```
1. ALL, шинаВход, шинаВыход;
2. ALL: *;
3. шинаВход: X{X[4], X[3], X[2], X[1], X[0]}D;
4. шинаВыход: Y1, Y2, Y3, Y4;
```

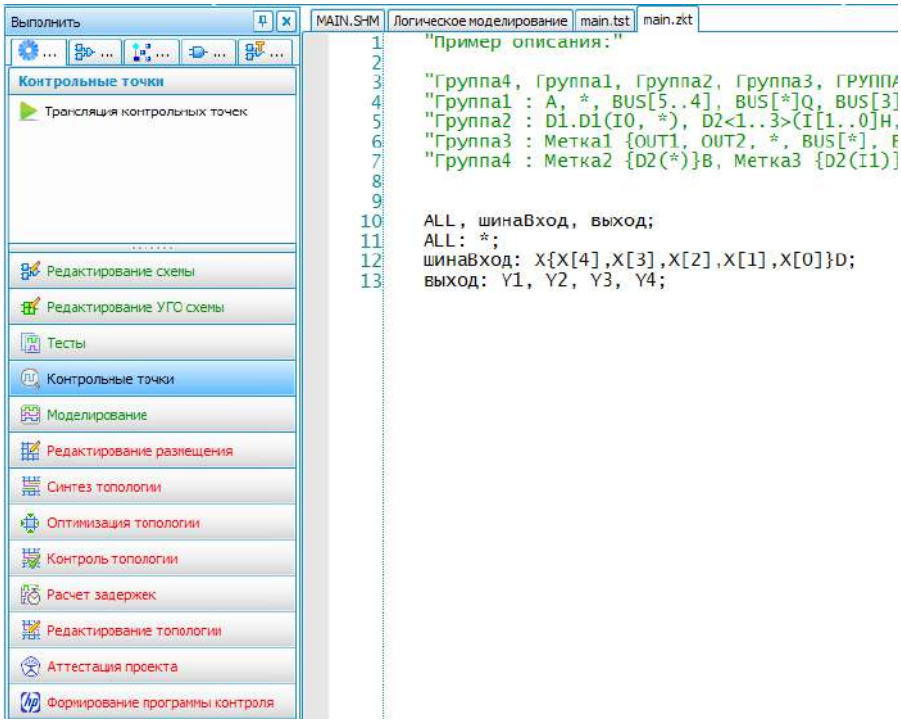

Рис. 2.49. Описание контрольных точек

Сигналы, которые будут отображаться на временной диаграмме, уже описаны. Осталось описать сами тестовые воздействия, которые будут подаваться на разрабатываемую схему. Для этого нужно выбрать команду «Тесты» в нижней части области панелей быстрого доступа, при этом открывается окно текстового редактора для описания тестовых воздействий (рис. 2.50). Добавьте туда следующий текст (ниже продемонстрированы два способа задания тестовых воздействий — групповой для шины данных в десятичной форме (приведен в кавычках, поэтому воспринимается как комментарий и отображается зеленым цветом) и индивидуальный для каждого разряда шины):

```
1.  ПроверкаСинтезаСхемы;
2.
3.  ПроверкаСинтезаСхемы:
4.  «X[4..0]D = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15»
```

```
5.  X[0] = (0,1):16;
6.  X[1] = (0:2,1:2):8;
7.  X[2] = (0:4,1:4):4;
8.  X[3] = (0:8,1:8):2;
9.  X[4] = (0:16,1:16):1;
```

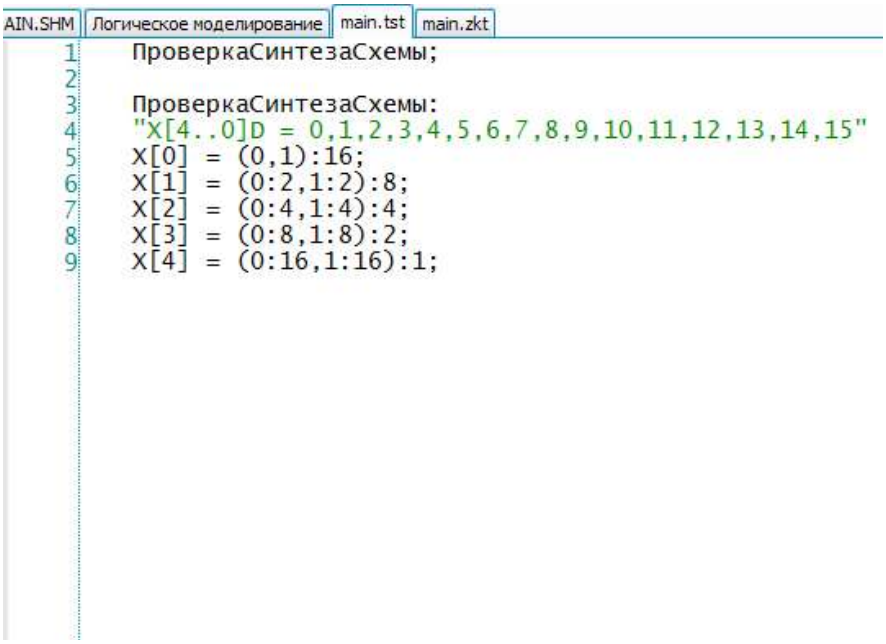

Рис. 2.50. Описание тестовых воздействий

Сохраните тест и выполните трансляцию тестов, выбрав либо команду «**Трансляция тестов**» в верхней части области панелей быстрого доступа (рис. 2.50), либо изображение зелёной стрелки в панели инструментов. Если трансляция выполнена без ошибок, перейдите в панель «**Моделирование**» в нижней части области панелей быстрого доступа. Нажмите изображние зелёного треугольника (или нажмите *F*12). После завершения процесса моделирования вы должны увидеть временную диграмму функционирования разработанной схемы (рис. 2.51).

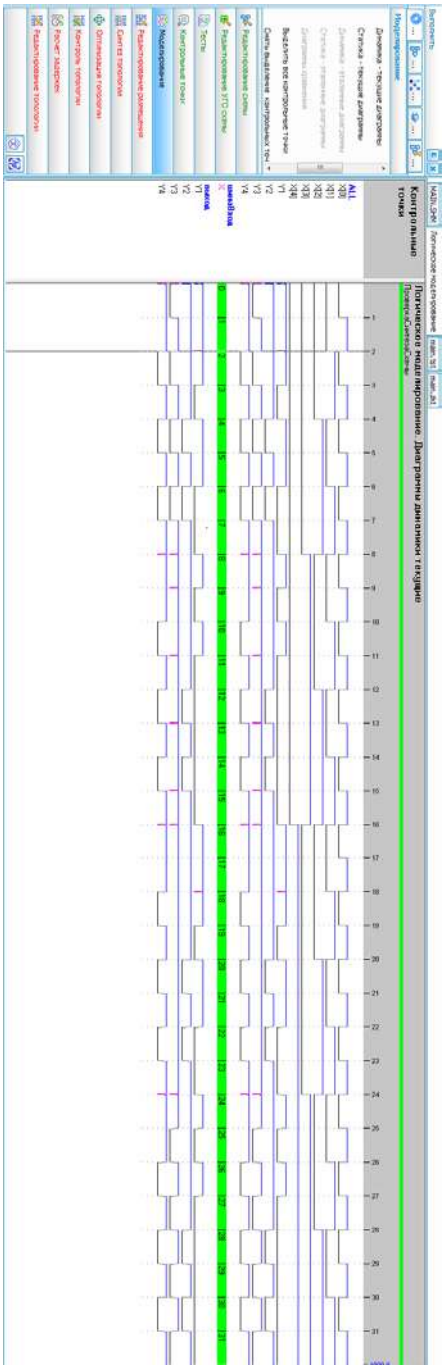

Рис. 2.51. Временная диаграмма, отражающая работу синтезированной схемы

Практические результаты моделирования цифровой схемы совпадают с предсказанными теоретическими.

2.3. Перечень индивидуальных заданий

1. Используя карты Карно, минимизировать функции алгебры логики y_1, y_2, y_3, y_4, заданные в индивидуальном варианте числовым представлением СДНФ или СКНФ. Построить временные диаграммы минимизированных ФАЛ.

2. Используя полученные выражения, реализовать схемы функций в одном схематическом файле в среде САПРС БИС «Ковчег 3.04».

3. Выполнить моделирование полученных функций.

4. Сравнить временные диаграммы, полученные на этапе синтеза с результатом моделирования в среде САПРС БИС «Ковчег 3.04».

Вариант 1

$y_1 = \vee(5,8,9,12,15; \times: 1,7,13)$;
$y_2 = \vee(0,1,2,9,11,13; \times: 3,4,8,10,12)$;
$y_3 = \vee(0,4,6,16,18,20,22,24,26,28; \times: 2,5,7,17,19,30)$;
$y_4 = \vee(1,4,5,6,9,12,14,17,20,21,22; \times: 0,2,3,7,13,15,25,28,29,30)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:

$$y_1 = x_1 + \overline{x_1} x_0;$$

$$y_2 = \overline{x_2} x_1 + x_2 x_0 + x_1 x_0.$$

Вариант 2

$y_1 = \wedge(0,2,3,6,7,8,10,14; \times:1,9,13)$;
$y_2 = \wedge(5,7,13,15; \times:3,4,6,11,12,14)$;
$y_3 = \wedge(11,15,17,19,21,23,25,27,29; \times:1,3,7,9,10,24,28,30,31)$;
$y_4 = \wedge(0,2,4,6–8,10,12,14,15,24,26,28,30,31;\times:11,16,18,20,22,23,25,29)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:

$$y1 = \overline{x_2} x_0 + \overline{x_1} x_0 + x_2 \overline{x_1};$$

$$y_2 = x_3 \overline{x1} + x3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0.$$

Вариант 3

$y_1 = \vee(0,3,6,8,10; \times:1,2,7)$;
$y_2 = \vee(1,3,4,6,11; \times:0,2,8,9,12)$;
$y_3 = \vee(1,3,7,9,13,15,18,19,22,23,26,27,31; \times:5,11,16,17,20,21,30)$;
$y_4 = \vee(8,9,12,13,14,25,28,30; \times:0,2,4,6,10,11,16,18,20,22,24,26,29)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:

$$y_1 = x_2 \overline{x_1} x_0 + x_2 \overline{x_1} \cdot \overline{x_0};$$

$$y_2 = \overline{x_2} x_0 + \overline{x_2} \cdot \overline{x_1} + \overline{x_1} \cdot \overline{x_0}.$$

Вариант 4

$y_1 = \wedge(0,2,4,6,10,11,14,15; \times:1,3,8);$
$y_2 = \wedge(3,5,7,10,11,13,14,15; \times:0,2,8,12);$
$y_3 = \wedge(1,5,8,9,12,13,24–31; \times:0,11,14,17,18,19,21,23);$
$y_4 = \wedge(9,11,12,13,17,19,20,21,25,27,28,29; \times:1,3,4,5,10,18,24,30,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_1} x_0 + \overline{x_3} x_2 \overline{x_1} x_0;$
$y_2 = x_3 \overline{x_1} + \overline{x_3} \cdot \overline{x_1}.$

Вариант 5

$y_1 = \vee(5,11,13,15; \times:4,10,12,14);$
$y_2 = \vee(0,1,9,10,12,14; \times:2,3,8);$
$y_3 = \vee(2,6,7,10,14,15,18,23,30; \times:0,1,3,4,5,22,26,31);$
$y_4 = \vee(2,6,7,16,18,19,20,21,23,25,29; \times:3,10,14,17,22,24,26,28,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_3} + x_3 \overline{x_2} x_1 \overline{x_0};$
$y_2 = x_2 \overline{x_1} \cdot \overline{x_0} + \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \overline{x_1} \cdot \overline{x_0}.$

Вариант 6

$y_1 = \wedge(5,6,7,13; \times:3,4,11,12);$
$y_2 = \wedge(5,7,8,12,13,15; \times:0,1,4,14);$
$y_3 = \wedge(9,11,13,15,16,17,20,21,24–31; \times:1,4,5,10,19);$
$y_4 = \wedge(8,10–12,14,16,18–20,22,24,26,27,28,30; \times:0,2,3,4,6,9,23,29).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2} \cdot \overline{x_0} + x_3 \overline{x_2} x_1 \overline{x_0};$
$y_2 = x_1 \overline{x_0} + \overline{x_1}.$

Вариант 7

$y_1 = \vee(0,2,3,7,8; \times:4,6,10,12);$
$y_2 = \vee(1,3,8,9,10,12; \times:0,4,11,14);$
$y_3 = \vee(1,4,5,8,10,12,13,14,18,22,24,26,28; \times:0,2,3,6,7,9,16,20,30);$
$y_4 = \vee(1,3,5,9,11,13,15,16,17,20,21,28; \times:7,18,19,22,23,24,25,29).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2} x_1 + x_3 \overline{x_2} x_1 \overline{x_0};$
$y_2 = x_3 x_2 \overline{x_0} + x_3 x_2 \overline{x_0} + x_3 x_2 \overline{x_1} \cdot \overline{x_0}.$

Вариант 8

$y_1 = \wedge(5,7,11,15; \times:1,3,9,12,14);$
$y_2 = \wedge(2,3,5,7,13,15; \times:1,10,11,12);$
$y_3 = \wedge(1,3,5,7,9,11,13,15,24,25,28,29; \times:0,2,6,16,17,20,21,27);$
$y_4 = \wedge(9-11,13,15,19,21,23,25,26,27,29,31; \times:1,2,3,5,7,17,18,20,24,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_2x_1 + x_1\overline{x_0} + x_2x_0;$
$y_2 = x_3 + x_3x_2\overline{x_1}x_0.$

Вариант 9

$y_1 = \vee(2,3,8,10,14; \times:0,4,11,12);$
$y_2 = \vee(4,5,11,13,14,15; \times:0,8,10,12);$
$y_3 = \vee(9,13,25,26,27,29,30,31; \times:0,7,8,11,12,17,21,24,28);$
$y_4 = \vee(1,3,5,7,9,11,13,15,25,27,29,31; \times:6,14,17,19,21,23,24,26,28,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_1x_0 + \overline{x_2}\overline{x_0} + x_2\overline{x_1};$
$y_2 = x_2 + \overline{x_2}x_1x_0.$

Вариант 10

$y_1 = \wedge(4,5,6,7,11,14,15; \times:1,3,8,12);$
$y_2 = \wedge(5,7,13,14,15,10; \times:3,8,12);$
$y_3 = \wedge(8,9,12,13,16,18,20,22,24,26,28,30; \times:0,1,4,5,15,21,27);$
$y_4 = \wedge(0,2,4,6,8,10,12,14,24-26,28-30; \times:1,16-18,20-22,27,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2}x_1x_0 + \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0;$
$y_2 = \overline{x_3}x_2 + x_3.$

Вариант 11

$y_1 = \vee(0,2,4,6,8-10; \times:1,3,12,14,15);$
$y_2 = \vee(0,1,4,9,13; \times:2,5,12);$
$y_3 = \vee(0,1,4-6,8,10,12,16,17,20,21; \times:2,3,7,14);$
$y_4 = \vee(2,3,6,7,10,11,15,17,19,21,25,27,29,31; \times:0,1,4,5,14,23).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_2\overline{x_1}x_0 + \overline{x_2}\,\overline{x_1}x_0 + x_2x_1x_0;$
$y_2 = x_2 + \overline{x_2}\,\overline{x_1}\,\overline{x_0}.$

Вариант 12

$y_1 = \wedge(1,5,7,9,13-15; \times:3,8,11,12);$
$y_2 = \wedge(5,7,13,15; \times:3,4,11);$
$y_3 = \wedge(0,1,8,9,13,24-27,29-31; \times:5,11,12,16-19,21,28);$
$y_4 = \wedge(8-15,19,23,26,27,30,31; \times:1,3,5,7,16,22,25).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = (x_1 + x_0)\cdot(x_2 + x_1 + x_0);$
$y_2 = x_2\overline{x_1}x_0 + x_1.$

Вариант 13

$y_1 = \vee(0,2,10,11,15; \times:4,8,12,14);$
$y_2 = \vee(0,2,6,7,8,14; \times:1,3,4,10,15);$
$y_3 = \vee(10,11,14,15,24,26,27,28-31,17,20,21; \times:2,16,18,22,25);$
$y_4 = \vee(0,3,4,7,16,19,20,24,27,28,31; \times:8,11,12,15,17,18,21-23).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_3} + x_4x_3x_0;$
$y_2 = (x_1 + x_0)(\overline{x_3} + \overline{x_2} + x_1 + x_0).$

Вариант 14

$y_1 = \wedge(1,3,5,7,14,15; \times:2,9-11);$
$y_2 = \wedge(7,12,13,15; \times:3,4-6,11,14);$
$y_3 = \wedge(9,11,13,15,16,20,24,25,27,28,29,31; \times:0,2,7,18,22,26,30);$
$y_4 = \wedge(1,3,5,7,8,12,17,19,21,23,24,28; \times:9,10,14,26,30,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2}x_0 + \overline{x_3}\overline{x_2}\ \overline{x_1}x_0;$
$y_2 = x_2x_1 + \overline{x_1}\ \overline{x_0} + x_2\overline{x_0}.$

Вариант 15

$y_1 = V(1,2,4,6,8,9; \times:0,3);$
$y_2 = V(0,1,8-10,15; \times:2,6,11,14);$
$y_3 = V(2,3,4,6,16,18,19,20,23; \times:10,14,22,26,27,30,31,7,0);$
$y_4 = V(1,5,7,9,11,13,16,17,19,20,22,23; \times:2,3,6,10,14,15,18,21,26,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = (\overline{x_1} + x_0)\cdot(x_2 + \overline{x_1} + x_0);$
$y_2 = x_2 + x_3\overline{x_2}\ \overline{x_1}x_0.$

Вариант 16

$y_1 = \wedge(7,8,10,12,14; \times:2,6)$;
$y_2 = \wedge(6,7,11,14,15; \times:2,4,5,9,10)$;
$y_3 = \wedge(9-13,18,24; \times:14,16,25-27)$;
$y_4 = \wedge(3-5,16,21,28; \times:6,17,20,29,31)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2}\,\overline{x_1}\,\overline{x_0} + \overline{x_2}\,\overline{x_1}\,\overline{x_0} + \overline{x_2}\,x_1\,\overline{x_0}$;
$y_2 = \overline{x_2} + x_3 x_2 x_1 \overline{x_0}$.

Вариант 17

$y_1 = \vee(0,4,6,8,9; \times:1,2,10)$;
$y_2 = \vee(7,10,14,15; \times:3,11,13)$;
$y_3 = \vee(0,2,4,8,12,14,16,18,19,20,23,26,27,30,31; \times:6,10,17,21,22,24,28)$;
$y_4 = \wedge(8-10,12,14,16-18,20,22,24-26,28,30; \times:0-2,4,6,21,23,27)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_0} + x_3 \overline{x_1} x_0$;
$y_2 = (x_2 + x_1)\cdot(\overline{x_3} + x_2 + x_1 + \overline{x_0})$.

Вариант 18

$y_1 = \wedge(6,7,9,11,12,13,14,15; \times:2,4,8,10)$;
$y_2 = \wedge(5,7,13,15; \times:1,3,4,6,9,11)$;
$y_3 = \wedge(8-15,19,23,25,27,29,31; \times:2,3,4,6,7,16)$;
$y_4 = \vee(0,2,4,6,8,10,12,16,19,22,23,27,28,30; \times:1,3,5,7,14,18,20,24,26,31)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_2\overline{x_1} + x_3 x_2 \overline{x_1}\,\overline{x_0}$;
$y_2 = x_3 x_1\,\overline{x_0} + x_3 x_1 x_0 + x_3 x_1 x_0$.

Вариант 19

$y_1 = \vee(2,4,6-8,10; \times:0,1,3,12,14)$;
$y_2 = \vee(1,7,9,12,13; \times:0,2,3,5,8)$;
$y_3 = \vee(2,3,7,10,12,18,19,22,23,24,26,28,30; \times:0,1,4-6,8,14)$;
$y_4 = \vee(4,6,7,13,15,16-18,20,22,23; \times:0,5,8,12,14,19,21,24,28)$.
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_3\overline{x_0} + \overline{x_3}\,\overline{x_0}$;
$y_2 = \overline{x_2} x_0 + \overline{x_3}\,\overline{x_2} x_1 x_0$.

Вариант 20

$y_1 = \wedge(1,2,4,6,8–10; \times:0,3,11,12,14);$
$y_2 = \wedge(2,3,7,9,13,15; \times:6,8,10,11);$
$y_3 = \wedge(9,11,13,15,17,21,25,29; \times:1,3,5,7,14,24,26,31);$
$y_4 = \wedge(0,1,3,4,5,7,8,9,12,13,15,24,28; \times:11,16,19,20,23,26,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2} + \overline{x_1} + \overline{\overline{x_2}x_1x_0};$
$y_2 = \overline{x_3}x_2 + x_3\overline{x_1} + x_2\overline{x_1}.$

Вариант 21

$y_1 = \vee(0,6,8,10,12,14,15; \times:2,4,9,11);$
$y_2 = \vee(0,1,5,8,9,13–15; \times:2–4,12);$
$y_3 = \vee(1,4,5,16–18,20,24,26,28,30; \times:0,2,6,10,14,21,22);$
$y_4 = \vee(0–2,5,9,10,13,17,18,21,24,25; \times:3,4,6,7,8,16,26,29).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_3 + \overline{x_3}x_0;$
$y_2 = x_2\overline{x_0} + \overline{x_1}x_0 + x_2\overline{x_1}.$

Вариант 22

$y_1 = \wedge(8,9,12,15; \times:1,5,7,13,14);$
$y_2 = \wedge(4–7; \times:12–15);$
$y_3 = \wedge(0–7,8,12,24,25,28,29; \times: 9,13,16,17,19,20,21,23);$
$y_4 = \wedge(2,3,6,7,11,15,16,20,25,29; \times:8,9,17,18,21,22).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_1} + \overline{x_2}\,\overline{x_1};$
$y_2 = x_3\overline{x_1} + x_3x_2\overline{x_1}\,\overline{x_0}.$

Вариант 23

$y_1 = \vee(0,8,10,14,15; \times:2,9,11);$
$y_2 = \vee(5,10,11,12–14; \times:0,4,8,15);$
$y_3 = \vee(3,5,7,17–19,20,22,23,24,28,30; \times:0,1,6,8,14,16,21,26);$
$y_4 = \vee(3,7,8,11,12,14,15,19,24,26,27,30; \times:4–6,10,16–18,23,28,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_3}\overline{x_2}\overline{x_1}\,\overline{x_0} + \overline{x_2}\overline{x_0};$
$y_2 = x_3\overline{x_2}\,\overline{x_1}x_0 + x_3x_2\overline{x_1}x_0 + x_3x_2x_1x_0.$

Вариант 24

$y_1 = \wedge(1,5,10,11,14,15; \times:3,4,7,13);$
$y_2 = \wedge(2,3,8{-}10,12,13; \times:0,4,11);$
$y_3 = \wedge(7,10,11,15,17,19,21,23,24{-}27,28,29,31; \times:3,8,12,14,30);$
$y_4 = \wedge(1{-}3,10,12,14,17{-}19,26,28,30; \times:0,6,7,8,9,16,21,24,27,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_3x_2x_1x_0 + x_2x_1x_0 + x_1x_0;$
$y_2 = x_2 + \overline{x_2}\,\overline{x_1x_0}.$

Вариант 25

$y_1 = \vee(7{-}9,12,13; \times:2,4,6,14,15);$
$y_2 = \vee(1,3,5{-}7,11,14,15; \times:8,12);$
$y_3 = \vee(2,3,7,18,19,20,22,23,24,26,28; \times:0,4{-}6,8,16,30);$
$y_4 = \vee(3,4,7,14,15,19,20,23,27,28,30; \times:5,6,11,12,16{-}18,31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2}x_1 + x_2\overline{x_1};$
$y_2 = \overline{x_3} + x_3x_2x_1.$

Вариант 26

$y_1 = \wedge(0{-}2,5,8,9,10; \times:11,13,15);$
$y_2 = \wedge(5,7,10{-}15; \times:3,6,8);$
$y_3 = \wedge(9{-}11,12,13,15,26,27,31; \times:3,5,8,14,16,21,24,30);$
$y_4 = \wedge(0{-}3,8{-}11,12,13,15,28,29,31; \times:5,14,18,21,27,30).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = (\overline{x_2} + x_0)\cdot(x_3 + \overline{x_2} + \overline{x_1} + x_0);$
$y_2 = x_1 + x_2\overline{x_1}x_{0.}$

Вариант 27

$y_1 = \vee(0,1,9,10,12,14; \times:4,8);$
$y_2 = \vee(2,3,6,7,10,11; \times:0,4,8,12,14);$
$y_3 = \vee(1,5,9,11,13,15,16,17,20,24,28,29; \times:0,2,3,4,6,7,21,25);$
$y_4 = \vee(3,5,7,11,15,19,21,29,31; \times:0{-}2,4,6,13,23,27).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = (\underline{x_3} + \overline{x_1} + \underline{x_0})\cdot(\overline{x_3} + \overline{x_1} + x0);$
$y_2 = x_2x_1 + x_3x_2x_1x_0.$

Вариант 28

$y_1 = \wedge(3,6,9,12; \times:2,8);$
$y_2 = \wedge(8,9,10,13,15; \times:0,1,3);$
$y_3 = \wedge(3,5,9,13,18,19,23,26,30; \times:0,1,4,22);$
$y_4 = \wedge(9,13,25,29; \times:16,17).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = (\overline{x_2} + \overline{x_1})\cdot(x_3 + \overline{x_2} + \overline{x_1})\cdot(x_2 + \overline{x_1});$
$y_2 = x_3\overline{x_0} + x_3\overline{x_1}\,\overline{x_0}.$

Вариант 29

$y_1 = \vee(0{-}7,8{-}11; \times:13{-}15);$
$y_2 = \vee(0{-}3,5{-}7,10,14{-}15; \times:8);$
$y_3 = \vee(0,7,13,15,18; \times:5,6,11,19,24{-}27);$
$y_4 = \vee(2,9,11,15,19; \times:7,15,23,28{-}31).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_3 + \overline{x_3}\,\overline{x_0};$
$y_2 = x_2\,\overline{x_1} + \overline{x_2}x_0 + \overline{x_1}x_0.$

Вариант 30

$y_1 = Л(0,1,3; \times:2,4,5,6,9{-}11,13);$
$y_2 = Л(0,2,4{-}6; \times:1,3,7,8,12,14);$
$y_3 = Л(2,7,17,19,20,25,28; \times:3,6,16,21,24,27,29);$
$y_4 = Л(3,5,11,13,19,23,27,31; \times:17,21,25,29).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = \overline{x_2}\,\overline{x_1}\,\overline{x_0} + \overline{x_2}x_1\overline{x_0} + \overline{x_2}x_1x_0;$
$y_2 = x_2\overline{x_0} + (\overline{x_2} + x_0)x_1.$

Вариант 31

$y_1 = V(0{-}5,7{-}14; \times:15);$
$y_2 = V(1,3,9,10; \times:2,7,13,14);$
$y_3 = V(0{-}4,12{-}14,20{-}22,24{-}26; \times:5,6,10,18,30);$
$y_4 = V(0{-}4; \times:16{-}18).$
Задание для защиты лабораторной работы № 2.
Упростить выражение:
$y_1 = x_2x_1 + x_2 \oplus x_1;$
$y_2 = x_3 \oplus 1.$