
Глава 4

ЦИФРОВЫЕ МИКРОЭЛЕКТРОННЫЕ УСТРОЙСТВА КОМБИНАЦИОННОГО ТИПА

4.1 Основные положения

В общем случае комбинационное цифровое устройство (КЦУ) может иметь $n \geq 1$ входов и $m \geq 1$ выходов. Если информационные значения входных сигналов обозначить как x_i ($i = \overline{1, n}$), а выходных сигналов — y_j ($j = \overline{1, m}$), то на каждом выходе КЦУ формируется некоторая булева функция $y_j = f_j(x_1, x_2, \dots, x_n)$, $j = \overline{1, m}$. Указанная запись говорит о том, что любому набору значений входных переменных x_i ($i = \overline{1, n}$) такого устройства, поданному в произвольный момент времени, однозначно соответствует набор значений переменных y_j ($j = \overline{1, m}$) на его выходах. Комбинационное устройство часто рассматривают как логический $n \times m$ -полюсник, а булевы функции $f_j(x_1, x_2, \dots, x_n)$, $j = \overline{1, m}$ называют системой собственных функций $n \times m$ -полюсника.

Исходными данными для проектирования цифрового устройства комбинационного типа являются его функциональное описание и требования к основным электрическим параметрам. Функциональное описание комбинационного устройства обычно дается в виде таблицы истинности или алгебраического выражения. На основе функционального описания синтезируют структурную схему минимальной сложности, после чего разрабатывают схему электрическую принципиальную на заданной или выбранной элементной базе.

При выборе оптимального варианта цифрового устройства (например, по критерию сложности), в том числе и комбинационного, необходимо учитывать ограничения, которые накладываются характеристиками реальных логических элементов:

- к выходу всякого реального логического элемента можно подключить лишь ограниченное число входов других элементов;

- общее число входов логического элемента ограничено;
- конечное время распространения сигнала в логических элементах может в отдельных случаях привести к нарушению работоспособности цифрового устройства.

4.2 Логические элементы

Логические элементы являются простейшими комбинационными цифровыми устройствами и выполняют элементарные логические операции над двоичными переменными.

Инвертор (логический элемент НЕ) содержит один вход и один выход и реализует логическую функцию «инверсия» (отрицание) $y = \bar{x}$.

Конъюнктор (логический элемент И) содержит $n \geq 1$ входов и один выход и реализует булеву функцию «конъюнкция» $y = \prod_{i=1}^n x_i = x_1 x_2 \dots x_n$. Таким образом, выходной сигнал конъюнктора принимает значение $y = 1$ тогда и только тогда, когда на все его входы одновременно поданы сигналы $x_i = 1$ ($i = \overline{1, n}$), а если хотя бы на один из входов подан сигнал $x_i = 0$, то на выходе также будет сигнал $y = 0$.

Дизъюнктор (логический элемент ИЛИ) также содержит $n \geq 1$ входов и один выход и реализует булеву функцию «дизъюнкция» $y = \sum_{i=1}^n x_i = x_1 + x_2 + \dots + x_n$. Выходной сигнал дизъюнктора принимает значение $y = 1$ тогда, когда хотя бы на один из его входов подан сигнал $x_i = 1$, и значение $y = 0$, когда одновременно на все входе поданы сигналы $x_i = 0$ ($i = \overline{1, n}$).

Логический элемент Шеффера (элемент И-НЕ) содержит $n \geq 1$ входов и один выход и реализует булеву функцию «штрих Шеффера» (логическую функцию И-НЕ) $y = \prod_{i=1}^n x_i = \overline{x_1 x_2 \dots x_n} = \sum_{i=1}^n \bar{x}_i = \bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n$.

Когда на все входы элемента Шеффера одновременно поданы сигналы $x_i = 1$ ($i = \overline{1, n}$), на его выходе формируется сигнал $y = 0$; если же хотя бы на один из входов подан сигнал $x_i = 0$, то на выходе формируется сигнал $y = 1$. Из теоремы де Моргана следует, что элемент Шеффера при переходе от положительной логики к отрицательной становится дизъюнктором.

Логический элемент Пирса (элемент ИЛИ-НЕ) содержит $n \geq 1$ входов и один выход и реализует булеву функцию «стрелка Пирса» (логическую функцию ИЛИ-НЕ) $y = \sum_{i=1}^n \bar{x}_i = \overline{x_1 + x_2 + \dots + x_n} = \prod_{i=1}^n \bar{x}_i = \bar{x}_1 \cdot \bar{x}_2 \cdot \dots \cdot \bar{x}_n$. Когда на все входы элемента Шеффера одновременно поданы сигналы $x_i = 0$ ($i = \overline{1, n}$), на его выходе формируется сигнал $y = 1$; если же хотя бы на один из входов подан сигнал $x_i = 1$, то на выходе формируется сигнал $y = 0$. По аналогии с элементом Шеффера элемент Пирса при переходе от положительной логики к отрицательной становится конъюнктором.

Логический элемент «исключающее ИЛИ» содержит $n \geq 1$ входов и один выход и реализует булеву функцию «исключающее ИЛИ» (сложение по модулю 2) $y = x_1 \oplus x_2 \oplus \dots \oplus x_n$. Выходной сигнал элемента «исключающее ИЛИ» принимает значение $y = 1$ тогда, когда сигналы $x_i = 1$ поданы на нечетное количество входов.

Перед обозначением многовходовых логических элементов обычно указывается число их входов, по которым реализуется соответствующая логическая функция, то есть коэффициент объединения по входу.

Подобно тому, как сложная булева функция может быть получена суперпозицией более простых функций, так и любое комбинационное цифровое устройство может быть реализовано комбинацией из логических элементов. При этом используется технический аналог операции суперпозиции: последовательное соединение комбинационных схем, в том числе и логических элементов, соответствует подстановке в булевы функции в качестве аргументов других булевых функций, а пересоединение по входам комбинационных схем соответствует перестановке аргументов булевых функций.

В одной интегральной микросхеме может быть несколько логических элементов, поэтому для сокращения обозначения состава микросхемы перед помещенным в круглые скобки наименованием элемента иногда указывают число этих элементов в одном корпусе микросхемы. Например, обозначению 4 (2И-НЕ) соответствует интегральная микросхема в составе 4 двухвходовых логических элементов И-НЕ.

При построении цифровых устройств часто возникает необходимость объединения выходов нескольких логических элементов с целью перехода на один общий выход. Эта задача может решаться разными способами. Можно выполнить объединение нескольких выходов с помощью логического элемента ИЛИ, однако это сопровождается дополнительными аппаратными затратами и ухудшением основных электрических параметров (увеличением среднего времени задержки распространения сигнала, повышением потребляемой мощности и т. д.). Другой способ связан с применением монтажной логики, основанной на соединении выходов нескольких логических элементов непосредственно либо с использованием диодных логических схем.

Для непосредственного соединения выходов нескольких логических элементов необходимо использовать элементы с открытым коллектором (стоком) или открытым эмиттером (истоком). На условных графических обозначениях логических элементов вывод с открытым коллектором (стоком) обозначается меткой ∇ , а вывод с открытым эмиттером (истоком) — меткой ∇ .

Один из наиболее широко используемых способов объединения выходов логических элементов основан на применении элементов, содержащих выходы с состоянием высокого импеданса (с тремя состояниями), которые на условных графических обозначениях обозначаются меткой \diamond . Состояние высокого импеданса соответствует отключению выходного каскада микросхемы от нагрузки, что позволяет объединять выходы непосредственно.

4.3 Методика синтеза комбинационных устройств

Синтез комбинационного устройства предполагает построение схемы минимальной сложности на основе логических элементов выбранного или заданного базиса по заданному алгоритму его функционирования. Процесс синтеза комбинационных устройств состоит из двух этапов:

- этап структурного (абстрактного) синтеза заключается в формализованном описании устройства с помощью аппарата булевых функций, их минимизации и построении структурной схемы устройства;
- схемный синтез сводится к выбору элементной базы и построению схемы электрической принципиальной.

Реализация задачи структурного синтеза сводится к четырем последовательным этапам.

1. *Формализованная запись условий функционирования комбинационного устройства* предполагает формирование таблиц истинности или запись алгебраических выражений реализуемых устройством булевых функций. Кроме того, условия функционирования могут быть заданы с помощью некоторой функциональной схемы. В этом случае необходимо проверить, действительно ли функциональная схема имеет минимальную сложность, используя один из критериев оценки сложности цифрового устройства, например на основе подсчета суммарного числа входов логических элементов, входящих в состав устройства (цена схемы по Квайну). Обычно устройство с минимальным суммарным числом входов содержит и минимальное число корпусов интегральных схем, что является дополнительным доводом в пользу такой оценки.

2. *Запись и минимизация алгебраических выражений булевых функций.* Минимизация логических выражений булевых функций, определяющих алгоритмы функционирования комбинационного цифрового устройства, обеспечивает уменьшение числа логических элементов, требуемых для его аппаратной реализации, что ведет к улучшению основных показателей по быстродействию, потребляемой мощности, степени миниатюризации. В процессе минимизации широко используются преобразования булевых функций с помощью соотношений алгебры логики, а также графические и специальные алгебраические методы. Для минимизации булевых функций относительно небольшого числа аргументов ($n \leq 6$) наиболее простым и наглядным является графический метод, основанный на использовании карт Карно. Для функций большего числа аргументов ($n > 6$) можно путем декомпозиции выделить более простые булевы функции с числом аргументов не более 6, которые затем минимизировать с помощью карт Карно. Комбинационные схемы, содержащие несколько выходов, на которых реализуются булевы функции f_1, f_2, \dots, f_m , часто синтезируются как несколько комбинационных устройств, имеющих общие входы и по одному отдельному выходу. В этом случае булевы функции f_1, f_2, \dots, f_m минимизируются независимо друг от друга, а общая схема состоит из изолированных подсхем. Иногда ее удается упростить за счет объединения участков подсхем, реализующих одинаковые члены, входящие в булевы функции f_1, f_2, \dots, f_m . Во многих случаях целесообразно проводить совместную минимизацию булевых функций f_1, f_2, \dots, f_m , то есть получать такие логические выражения, которые обеспечивают наиболее простую логическую структуру схемы в целом.

3. *Запись минимизированных выражений булевых функций в заданном базисе.* Поскольку базовые логические элементы современных цифровых микросхем выполняют операции И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ, то часто возникает необходимость записи выражений булевых функций в одном из этих базисов. При этом необходимо учитывать, что к выходу всякого реального логического элемента можно подключить лишь ограниченное число входов других элементов. Нагрузочная спо-

способность задается коэффициентом разветвления, который и определяет наибольшее допустимое количество входов логических элементов, подключаемых к выходу данного элемента. В некоторых случаях приходится обеспечивать разгрузку элемента, то есть схемным путем перераспределять часть нагрузки на другие элементы. Кроме того, необходимо учитывать ограниченное число входов реальных логических элементов, которое задается коэффициентом объединения по входу. Поэтому в булевом выражении, на основе которого реализуется комбинационное устройство, дизъюнкции (конъюнкции) могут содержать лишь ограниченное число членов. Если реализация функции требует использования логических элементов с коэффициентами объединения и разветвления, большими заданных, то следует провести необходимые дополнительные преобразования булева выражения так, чтобы получить выражение, для реализации которого требуются логические элементы с коэффициентами объединения и разветвления, не большими заданных. Однако при этом возрастают общее число логических элементов в схеме и число последовательно включенных каскадов элементов, то есть увеличиваются потребляемая мощность и среднее время задержки распространения сигнала. Таким образом, снижение требований к значениям коэффициентов разветвления и объединения элементов либо приводит к снижению быстродействия и экономичности комбинационных устройств, либо требует соответствующего уменьшения значений потребляемой мощности и среднего времени задержки распространения сигнала элементов.

4. *Составление структурной схемы.* На этом этапе каждой логической операции преобразованного булева выражения ставится в соответствие определенный логический элемент заданного (или выбранного) базиса и производятся необходимые соединения между элементами.

Например, требуется синтезировать структурную схему комбинационного цифрового устройства в базисе И-НЕ, алгоритм функционирования которого задан таблицей истинности (табл. 4.1).

Таблица 4.1 – Таблица истинности комбинационного цифрового устройства

Номер набора	x_1	x_2	x_3	x_4	y	Номер набора	x_1	x_2	x_3	x_4	y
0	0	0	0	0	х	8	1	0	0	0	0
1	0	0	0	1	0	9	1	0	0	1	х
2	0	0	1	0	х	10	1	0	1	0	1
3	0	0	1	1	0	11	1	0	1	1	0
4	0	1	0	0	х	12	1	1	0	0	1
5	0	1	0	1	1	13	1	1	0	1	0
6	0	1	1	0	х	14	1	1	1	0	1
7	0	1	1	1	1	15	1	1	1	1	0

Так как задан базис И-НЕ, то целесообразно использовать СДНФ. Составим карту Карно, представленную на рис. 4.1.

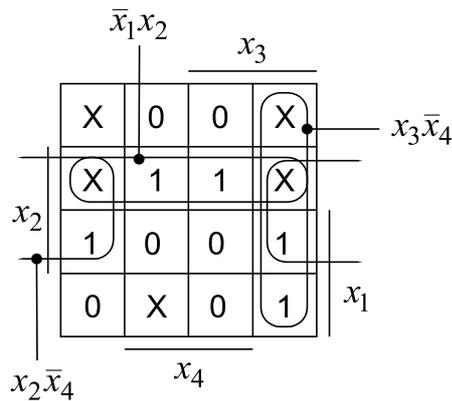


Рис. 4.1 – Карта Карно функции четырех аргументов

Минимизированная ДНФ функции имеет вид:

$$y = \bar{x}_1x_2 + x_2\bar{x}_4 + x_3\bar{x}_4. \quad (4.1)$$

Для перехода в базис И-НЕ ставим два знака инверсии над правой частью полученного минимизированного выражения (4.1) и, применив формулу де Моргана, получим:

$$y = \overline{\overline{\bar{x}_1x_2 + x_2\bar{x}_4 + x_3\bar{x}_4}} = \overline{\overline{\bar{x}_1x_2} \cdot \overline{x_2\bar{x}_4} \cdot \overline{x_3\bar{x}_4}}.$$

Окончательное булево выражение имеет вид:

$$y = \overline{\overline{\bar{x}_1x_2} \cdot \overline{x_2\bar{x}_4} \cdot \overline{x_3\bar{x}_4}} = \overline{\overline{\bar{x}_1x_1x_2} \cdot \overline{x_2\bar{x}_4x_4} \cdot \overline{x_3\bar{x}_4x_4}},$$

а соответствующая этому выражению структурная схема представлена на рис. 4.2, а.

Рассмотрим другой подход к синтезу структурной схемы, для чего преобразуем выражение (4.1) путем вынесения за скобки общего члена из первых двух конъюнкций $y = x_2(\bar{x}_1 + \bar{x}_4) + x_3\bar{x}_4$.

Используя формулу де Моргана, получим:

$$y = x_2(\bar{x}_1 + \bar{x}_4) + x_3\bar{x}_4 = x_2\overline{\bar{x}_1\bar{x}_4} + x_3\bar{x}_4,$$

после чего перейдем в базис И-НЕ:

$$y = \overline{\overline{x_2\bar{x}_1\bar{x}_4} + x_3\bar{x}_4} = \overline{\overline{x_2\bar{x}_1\bar{x}_4} \cdot \overline{x_3\bar{x}_4x_4}}. \quad (4.2)$$

Булеву выражению (4.2) соответствует структурная схема, представленная на рис. 4.2, б. Этот вариант проще предыдущего.

Пример показывает, что после применения карт Карно возможно дополнительное упрощение булевых выражений с помощью соотношений алгебры логики.

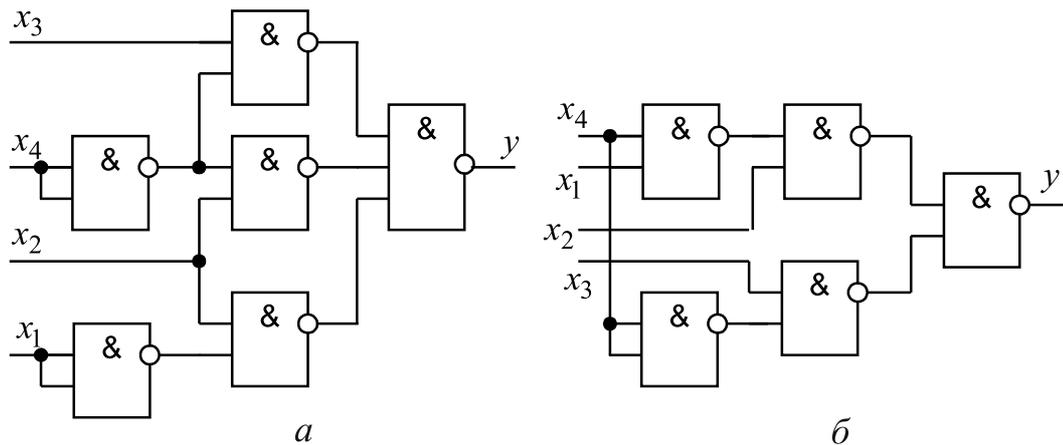


Рис. 4.2 – Схемы комбинационных устройств, реализующих логическую функцию четырех переменных

4.4 Мультиплексоры и демультиплексоры

Назначение мультиплексоров (от англ. *multiplex* — многократный) — коммутировать в желаемом порядке информацию, поступающую с нескольких входов, на один выход. Мультиплексоры в цифровой аппаратуре используются для временного разделения информации, поступающей по разным каналам [6].

Мультиплексоры обладают двумя группами входов и одним, реже двумя (взаимодополняющими) выходами, один из которых прямой, а другой — инверсный. Одна группа входов объединяет информационные входы, а другая служит для управления работой мультиплексора. Управляющие входы подразделяются на адресные входы и разрешающие (стробирующие) входы. Полный мультиплексор, обладающий n адресными входами, содержит 2^n информационных входов и обозначается как «мультиплексор $2^n - 1$ ». Если на адресные входы подать n -разрядный двоичный код числа $i \in \{0, 1, 2, \dots, 2^n - 1\}$, то выход подключится к i -му информационному входу, то есть информация, поступающая на i -ый информационный вход, будет проходить на выход независимо от того, какие сигналы поступают на остальные информационные входы.

Разрешающий (стробирующий) вход управляет одновременно всеми информационными входами независимо от состояния адресных входов. Запрещающий сигнал на этом входе блокирует действие всей комбинационной схемы мультиплексора. Наличие разрешающего входа расширяет функциональные возможности мультиплексора, позволяя синхронизировать его работу с работой других узлов цифровой техники. Разрешающий вход используется также для наращивания разрядности мультиплексора. Логическая функция, выполняемая полным мультиплексором с n адресными входами и одним прямым входом разрешения на прямом выходе, имеет вид:

$$f = E \sum_{i=0}^{2^n-1} m_i d_i, \quad (4.3)$$

где m_i — минтерм, соответствующий i -му набору переменных на адресных входах; E — сигнал на входе разрешения; d_i — сигнал на i -ом информационном входе.

Например, для полного мультиплексора 8–1 (рис. 4.3) логическая функция имеет вид:

$$f = E \sum_{i=0}^7 m_i d_i = E(\bar{a}_2 \bar{a}_1 \bar{a}_0 d_0 + \bar{a}_2 \bar{a}_1 a_0 d_1 + \bar{a}_2 a_1 \bar{a}_0 d_2 + \bar{a}_2 a_1 a_0 d_3 + a_2 \bar{a}_1 \bar{a}_0 d_4 + a_2 \bar{a}_1 a_0 d_5 + a_2 a_1 \bar{a}_0 d_6 + a_2 a_1 a_0 d_7),$$

где a_i — сигналы, подаваемые на адресные входы мультиплексора.



Рис. 4.3 – Полный мультиплексор 8–1

У интегральных микросхем мультиплексоров число информационных входов не превышает 16. Большее число входов обеспечивается наращиванием двумя способами: объединением нескольких мультиплексоров в пирамидальную (древовидную) систему либо последовательным соединением разрешающих входов и внешних логических элементов.

На рис. 4.4 показана организация мультиплексора 32–1 из двух мультиплексоров 16–1 с использованием разрешающих входов мультиплексоров в качестве адресных входов высшего разряда. Такой мультиплексор должен иметь $\log_2 32 = 5$ адресных входов. Адресными входами низших разрядов служат входы a_0, a_1, a_2, a_3 . Разрешающие входы в данном случае используются для подачи высшего разряда a_4 : на мультиплексор *DD1* в прямом виде, на мультиплексор *DD2* — в инверсном.

Мультиплексор *DD1* работает при $a_4 = 0$, мультиплексор *DD2* — при $a_4 = 1$. Благодаря логическому элементу И-НЕ сигналы на выходе f будут одинаковыми с входными.

Мультиплексоры помимо прямого назначения могут выполнять и другие функции, например использоваться для преобразования параллельного двоичного кода в последовательный, работать в качестве универсального логического элемента, реализующего любую логическую функцию, содержащую до $(n + 1)$ аргументов. Применение мультиплексора в качестве универсального логического элемента особенно оправдано, когда число переменных достаточно велико (4–5 и более).

Использование мультиплексора в качестве универсального логического элемента основано на общем свойстве логических функций — независимо от числа аргументов всегда равняться логической единице или нулю: $f(x_1, x_2, \dots, x_n) \in \{0, 1\}$.

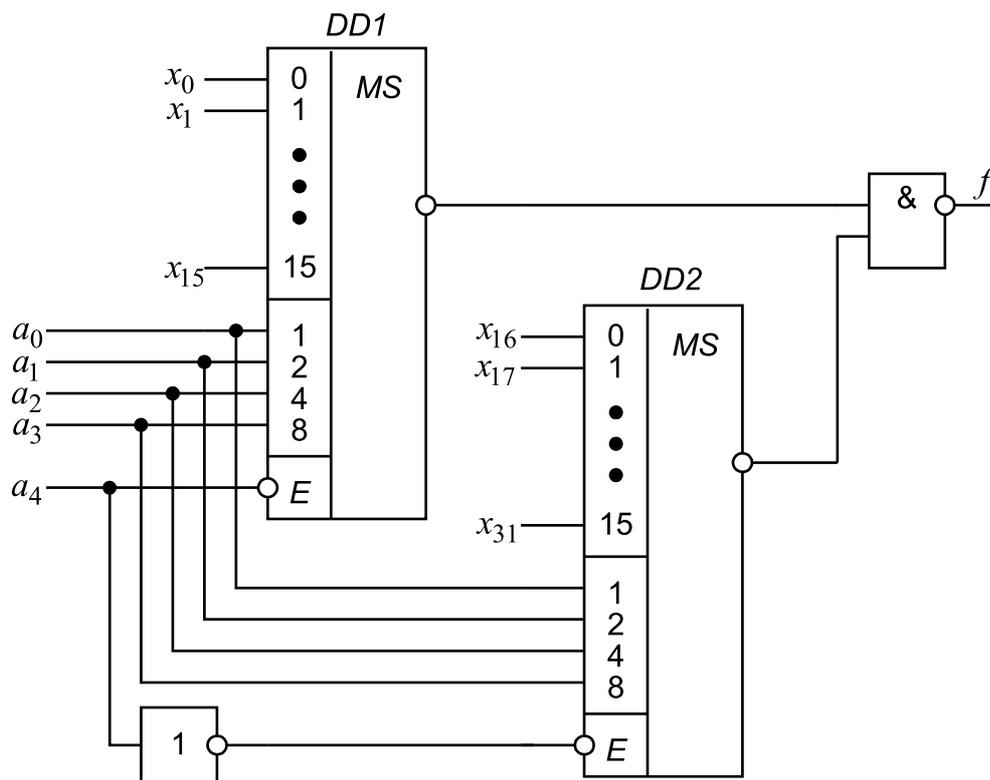


Рис. 4.4 – Нарастание разрядности мультиплексора последовательным соединением разрешающих входов

Если на адресные входы мультиплексора подавать входные переменные, зная, какой выходной уровень должен отвечать каждому сочетанию этих сигналов, то, предварительно установив на информационных входах потенциалы нуля и единицы согласно заданному алгоритму, получим устройство, реализующее требуемую функцию.

В качестве примера на рис. 4.5 представлена реализация с помощью четырехвходового мультиплексора функции «исключающее ИЛИ» двух аргументов.

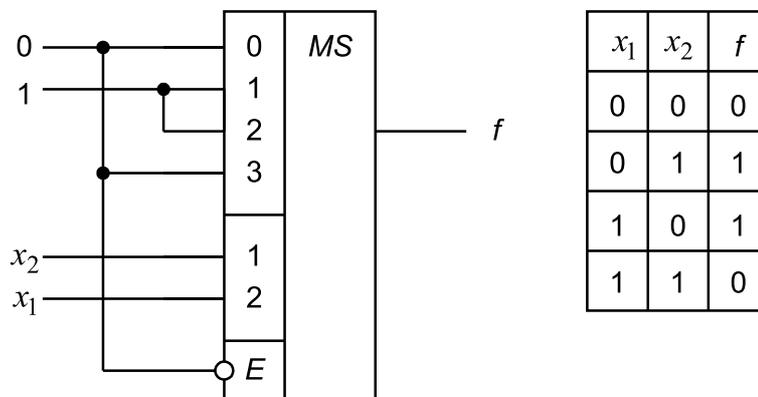


Рис. 4.5 – Реализация функции «исключающее ИЛИ» двух аргументов с помощью мультиплексора

Как следует из таблицы истинности для функции «исключающее ИЛИ», сочетаниям $x_1x_2 = 00$ и $x_1x_2 = 11$ отвечает значение $f = 0$, а двум другим $x_1x_2 = 01$ и $x_1x_2 = 10$ — значение $f = 1$. Для выполнения этих условий достаточно подать на адресные входы мультиплексора сигналы $a_1 = x_2$ и $a_0 = x_1$, а на информационные входы — сигналы $d_0 = d_3 = 0$, $d_1 = d_2 = 1$. Разрешающий вход при этом должен быть под действием напряжения логического нуля.

Если число аргументов равно $(n + 1)$, то мультиплексор следует использовать несколько иначе. Например, требуется с помощью четырехвходового мультиплексора реализовать функцию трех аргументов, заданную таблицей истинности.

x_1	x_2	x_3	f	Примеч.
0	0	0	1	$f = 1$
0	0	1	1	
0	1	0	0	$f = x_3$
0	1	1	1	
1	0	0	0	$f = 0$
1	0	1	0	
1	1	0	1	$f = \bar{x}_3$
1	1	1	0	

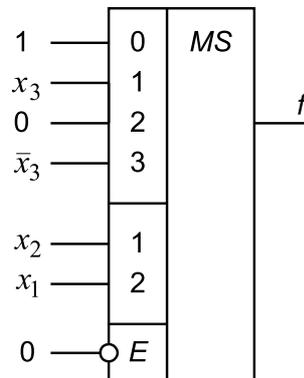


Рис. 4.6 – Реализация функции трех аргументов на четырехвходовом мультиплексоре

Разделение таблицы истинности на группы по две строки в каждой показывает: в каждой группе аргументы x_1 и x_2 неизменны, аргумент x_3 младшего разряда имеет два состояния, выходной сигнал f имеет одно из четырех значений 1, 0, x_3 и \bar{x}_3 . Если значения аргументов x_1 и x_2 подать на адресные входы мультиплексора $a_1 = x_1$ и $a_0 = x_2$, а на информационные входы подать, согласно таблице истинности, сигналы 1, 0, x_3 и \bar{x}_3 , то такая схема (рис. 4.6) будет выполнять заданную логическую функцию.

Аналогично можно проектировать комбинационные цифровые устройства и с большим числом входов.



.....
 Демультимплексоры в функциональном отношении противоположны мультиплексорам: сигналы с одного информационного входа распределяются в необходимой последовательности по нескольким выходам.

Выбор нужного выхода, как и в мультиплексоре, обеспечивается двоичным кодом на адресных входах. При n адресных входах полный демультимплексор имеет 2^n выходов, которые могут быть прямыми или инверсными. На каждом прямом вы-

ходе демультиплексора, содержащего прямой вход разрешения, реализуется булева функция:

$$f_i = Em_i d, \quad (4.4)$$

где m_i — минтерм, соответствующий i -му набору переменных на адресных входах; E — сигнал на входе разрешения; d — сигнал на информационном входе.

Например, полный демультиплексор 1–4 (рис. 4.7) на своих выходах реализует систему булевых функций:

$$f_0 = \overline{E}m_0d = \overline{E}\overline{a_1}\overline{a_0}d, \quad f_1 = \overline{E}m_1d = \overline{E}\overline{a_1}a_0d, \quad f_2 = \overline{E}m_2d = \overline{E}a_1\overline{a_0}d, \\ f_3 = \overline{E}m_3d = \overline{E}a_1a_0d.$$

ИМС демультиплексоров имеют 4, 8 или 16 выходов. Если требуется большее число выходов, демультиплексоры наращиваются в систему, и в этом отношении принципиального различия с мультиплексорами нет.

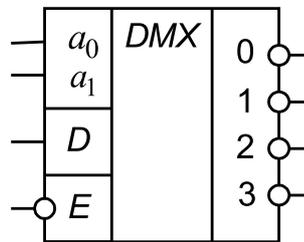


Рис. 4.7 – Условное графическое обозначение полного демультиплексора 1–4

4.5 Шифраторы и дешифраторы

К основным видам преобразования информации в цифровых системах относят шифрацию и дешифрацию, для реализации которых используют комбинационные цифровые устройства, называемые шифраторами и дешифраторами соответственно.



.....
Шифратором называют комбинационную схему, реализующую преобразование унитарного кода «1 из n » $X = x_{n-1} \dots x_0$ в m -разрядный двоичный код $Y = y_{m-1} \dots y_0$.

В унитарном коде «1 из n » только один разряд принимает значение 1, а все оставшиеся разряды — 0. Если в унитарном коде $x_l = 1$, то число $Y = y_{m-1} \dots y_0$ представляет собой двоичный код номера разряда l .

Число входов шифратора не превышает количества возможных комбинаций выходных сигналов: $n \leq 2^m$, причем если $n = 2^m$, то шифратор называют полным, а если $n < 2^m$, то неполным.

Принцип функционирования полного шифратора 8–3, преобразующего унитарный код «1 из 8» в трехразрядный двоичный код, определяется таблицей истинности (табл. 4.2).

Таблица 4.2 – Таблица истинности полного шифратора 8–3

№	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

На основе таблицы истинности можно записать соответствующие булевы выражения для y_2, y_1, y_0 , а затем выполнить необходимые действия по их упрощению. В данном случае можно воспользоваться особенностью входных переменных, которые в интересующих нас комбинациях только в одном разряде имеют единичное значение. Это позволяет избежать записи и преобразования выражений булевых функций в общем виде, достаточно громоздких в случае восьми входных переменных, и представить выражения для выходных переменных в виде:

$$y_2 = x_7 + x_6 + x_5 + x_4 = \overline{x_7}\overline{x_6}\overline{x_5}\overline{x_4}, \quad y_1 = x_7 + x_6 + x_3 + x_2 = \overline{x_7}\overline{x_6}\overline{x_3}\overline{x_2},$$

$$y_0 = x_7 + x_5 + x_3 + x_1 = \overline{x_7}\overline{x_5}\overline{x_3}\overline{x_1}.$$

Реализация полного шифратора 8–3 требует трех четырехходовых логических элементов ИЛИ либо трех четырехходовых и семи двухходовых логических элементов И-НЕ.

Часто на практике возникает необходимость преобразования в двоичный код n -разрядного кода, только один разряд которого принимает значение 0, а все остальные – 1. Для этой цели можно использовать шифраторы с инверсными входами. Например, неполному шифратору 10–4 с инверсными входами, преобразующему унитарный код «1» из «10» в четырехразрядный двоичный код, соответствует таблица истинности (табл. 4.3).

Таблица 4.3 – Таблица истинности, соответствующая неполному шифратору 10–4 с инверсными входами

№	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
2	1	1	1	1	1	1	1	0	1	1	0	0	1	0
3	1	1	1	1	1	1	0	1	1	1	0	0	1	1
4	1	1	1	1	1	0	1	1	1	1	0	1	0	0
5	1	1	1	1	0	1	1	1	1	1	0	1	0	1
6	1	1	1	0	1	1	1	1	1	1	0	1	1	0
7	1	1	0	1	1	1	1	1	1	1	0	1	1	1
8	1	0	1	1	1	1	1	1	1	1	1	0	0	0
9	0	1	1	1	1	1	1	1	1	1	1	0	0	1

По аналогии с шифратором 8–3 в данном случае можно воспользоваться особенностью входных переменных, которые в интересующих нас комбинациях только в одном разряде имеют нулевое значение, что позволяет представить выражения для выходных переменных в виде:

$$\begin{aligned}y_3 &= \bar{x}_9 + \bar{x}_8 = \overline{x_9 x_8}, y_2 = \bar{x}_7 + \bar{x}_6 + \bar{x}_5 + \bar{x}_4 = \overline{x_7 x_6 x_5 x_4}, \\y_1 &= \bar{x}_7 + \bar{x}_6 + \bar{x}_3 + \bar{x}_2 = \overline{x_7 x_6 x_3 x_2}, \\y_0 &= \bar{x}_9 + \bar{x}_7 + \bar{x}_5 + \bar{x}_3 + \bar{x}_1 = \overline{x_9 x_7 x_5 x_3 x_1}.\end{aligned}$$

Реализация неполного шифратора 10–4 с инверсными входами требует одного пятиходового, двух четырехходовых и одного двухходового логических элементов И-НЕ.



.....
Дешифратор выполняет функцию, обратную шифратору, то есть преобразует двоичный код в унитарный код.
.....

Число входов и выходов полного дешифратора связано соотношением $m = 2^n$, а неполного дешифратора — $m < 2^n$.

Дешифратор с прямыми выходами реализует минтермы входных переменных $f_i = m_i$, а дешифратор с инверсными выходами — инверсии минтермов, то есть макс-термы входных переменных $f_i = \bar{m}_i = M_i$.

Например, таблица истинности полного дешифратора 3–8 с инверсными выходами представлена в табл. 4.4.

Таблица 4.4 – Таблица истинности полного дешифратора 3–8 с инверсными выходами

№	x_2	x_1	x_0	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
2	0	1	0	1	1	1	1	1	0	1	1
3	0	1	1	1	1	1	1	0	1	1	1
4	1	0	0	1	1	1	0	1	1	1	1
5	1	0	1	1	1	0	1	1	1	1	1
6	1	1	0	1	0	1	1	1	1	1	1
7	1	1	1	0	1	1	1	1	1	1	1

Записав на основе таблицы истинности выражения для булевых функций y_7, y_6, \dots, y_0 , а затем выполнив необходимые действия по их упрощению, получим:

$$\begin{aligned}y_0 &= x_2 + x_1 + x_0 = M_0 = \bar{m}_0 = \overline{\bar{x}_2 \bar{x}_1 \bar{x}_0}, & y_1 &= x_2 + x_1 + \bar{x}_0 = M_1 = \bar{m}_1 = \overline{\bar{x}_2 \bar{x}_1 x_0}, \\y_2 &= x_2 + \bar{x}_1 + x_0 = M_2 = \bar{m}_2 = \overline{\bar{x}_2 x_1 \bar{x}_0}, & y_3 &= x_2 + \bar{x}_1 + \bar{x}_0 = M_3 = \bar{m}_3 = \overline{\bar{x}_2 x_1 x_0}, \\y_4 &= \bar{x}_2 + x_1 + x_0 = M_4 = \bar{m}_4 = \overline{x_2 \bar{x}_1 \bar{x}_0}, & y_5 &= \bar{x}_2 + x_1 + \bar{x}_0 = M_5 = \bar{m}_5 = \overline{x_2 \bar{x}_1 x_0}, \\y_6 &= \bar{x}_2 + \bar{x}_1 + x_0 = M_6 = \bar{m}_6 = \overline{x_2 x_1 \bar{x}_0}, & y_7 &= \bar{x}_2 + \bar{x}_1 + \bar{x}_0 = M_7 = \bar{m}_7 = \overline{x_2 x_1 x_0}.\end{aligned}$$

Дешифраторы можно использовать для построения произвольного комбинационного цифрового устройства. Поскольку активное значение сигнала на каждом выходе дешифратора определяет одну из комбинаций входных сигналов, то, объединяя с помощью соответствующих логических элементов некоторые выходные сигналы дешифратора, можно реализовать комбинационное цифровое устройство, число наборов таблицы истинности которого не превышает числа выходов используемого дешифратора.

Рассмотрим использование полного дешифратора 3–8 для реализации комбинационного цифрового устройства, заданного таблицей истинности (табл. 4.5).

Таблица 4.5 – Таблица истинности комбинационного цифрового устройства

№	x_1	x_2	x_3	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	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Для реализации цифрового устройства на основе дешифратора с прямыми выходами выражение булевой функции целесообразно представить в СДНФ, а при использовании дешифратора с инверсными выходами — в СКНФ.

Для рассматриваемого комбинационного устройства выражение булевой функции в СДНФ имеет вид $y = m_3 + m_5 + m_6 + m_7$, а в СКНФ — $y = M_0M_1M_2M_4$.

Варианты построения комбинационного устройства на дешифраторах с прямыми и инверсными выходами представлены на рис. 4.8, а и рис. 4.8, б соответственно.

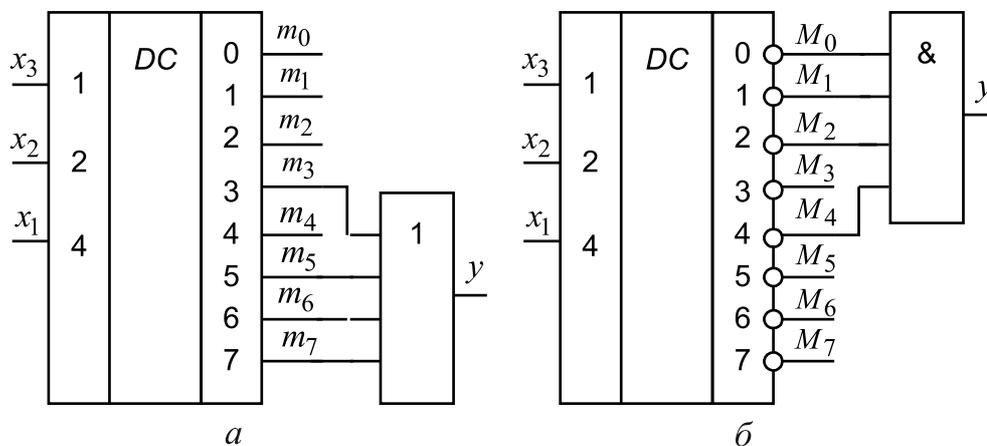


Рис. 4.8 – Варианты реализации функции трех аргументов с помощью дешифраторов



.....
 Дешифраторы подобно демультиплексорам допускают наращива-
 ние разрядности входного двоичного кода.

4.6 Сумматоры и вычитатели

Сумматоры представляют собой функциональные узлы, выполняющие операцию сложения чисел. В устройствах цифровой техники суммирование осуществляется в двоичном или двоично-десятичном кодах. Сумматоры используются также для реализации операций вычитания, умножения и деления, в качестве преобразователей кодов и в ряде других случаев.

По характеру действия сумматоры подразделяются на комбинационные и накопительные (сохраняющие результаты вычислений). В свою очередь, каждый из сумматоров, оперирующий с многоразрядными слагаемыми, в зависимости от способа обработки чисел может быть отнесен к последовательному или параллельному типу. Сложение чисел в последовательных сумматорах осуществляется поразрядно, последовательно во времени. В сумматорах параллельного действия сложение всех разрядов многоразрядных чисел происходит одновременно.

Простейшим суммирующим элементом является полусумматор, условное графическое обозначение которого представлено на рис. 4.9.

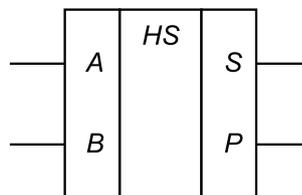


Рис. 4.9 – Условное графическое обозначение полусумматора

Обозначением функции полусумматора служат буквы *HS* (*Half Sum* — полусумма). Полусумматор имеет два входа *A* и *B* для двух слагаемых и два выхода: *S* — сумма и *P* — перенос.

Таблица истинности полусумматора представлена в табл. 4.6.

Таблица 4.6 – Таблица истинности полусумматора

Входы		Выходы	
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Булевы функции, описывающие работу полусумматора, имеют вид:

$$S = \bar{A}B + A\bar{B} = A \oplus B, \quad P = AB.$$

Логическая структура полусумматора в общем и развернутом видах показана на рис. 4.10.

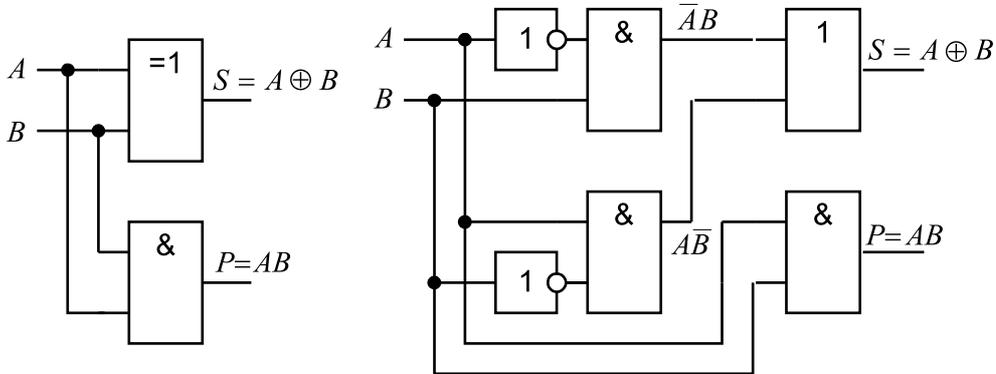


Рис. 4.10 – Логическая структура полусумматора

Полусумматор имеет два входа и поэтому пригоден для использования только в младшем разряде многоразрядных двоичных чисел. Начиная со второго разряда многоразрядных чисел, необходимо использовать полный одноразрядный сумматор, содержащий три входа, на один из которых подается сигнал переноса из предыдущего разряда.

Полный одноразрядный сумматор (рис. 4.11, б) можно представить как объединение двух полусумматоров (рис. 4.11, а).

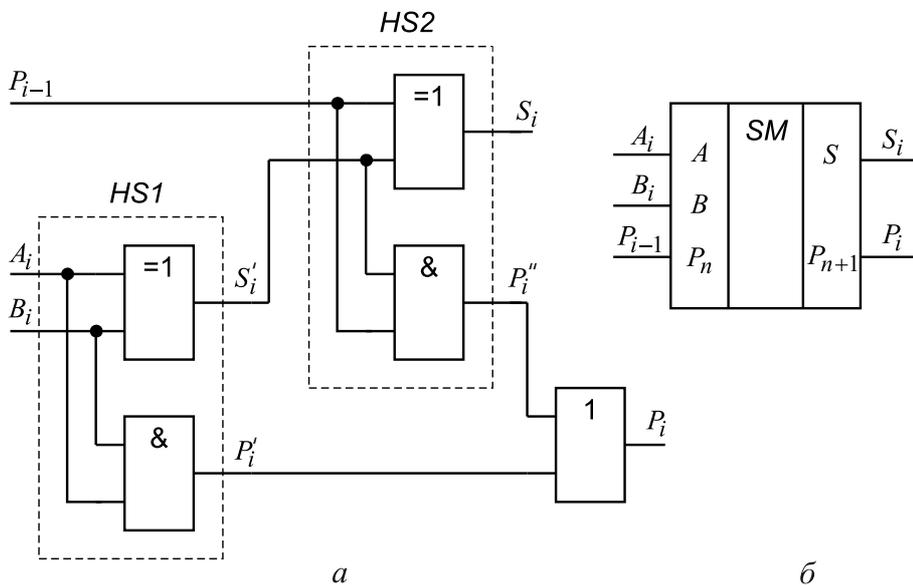


Рис. 4.11 – Полный одноразрядный сумматор: а – реализация на полусумматорах; б – условное графическое обозначение



Для суммирования n -разрядных двоичных чисел объединяется n полных одноразрядных сумматоров (рис 4.12).

При этом отдельные разряды суммируемых чисел A и B подаются на входы a_i и b_i . На вход p_{i-1} подается перенос из предыдущего, более младшего разряда. Формируемый в данном разряде перенос p_i передается в следующий, более старший разряд.

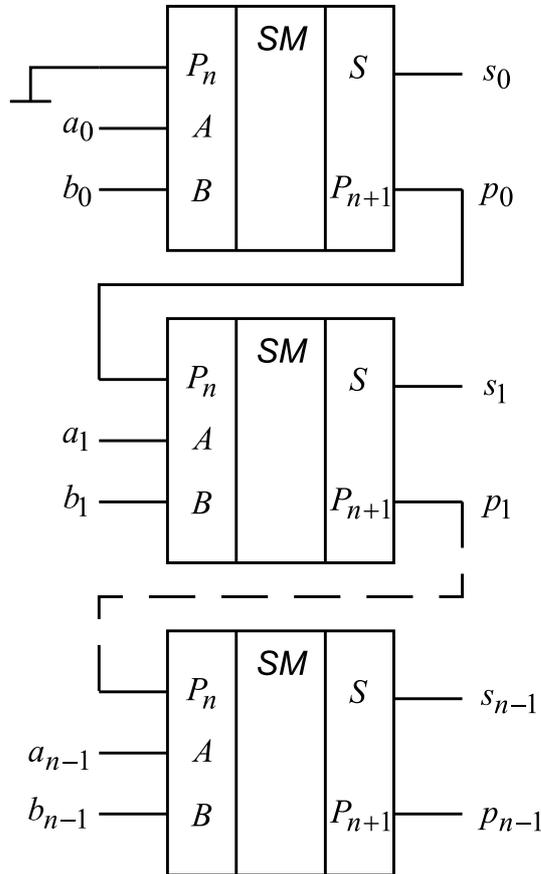


Рис. 4.12 – Схема n -разрядного сумматора с последовательным переносом

В устройствах цифровой техники операция вычитания обычно сводится к операции сложения:

$$\begin{aligned} D &= M - S = M + (2^n - S) - 2^n = \\ &= M + S_{\text{доп}} - 2^n = M + \bar{S} + 1 - 2^n, \end{aligned} \quad (4.5)$$

где \bar{S} – обратный, а $S_{\text{доп}} = 2^n - S = \bar{S} + 1$ – дополнительный код вычитаемого.

Для корректного представления результата вычитания двух n -разрядных чисел требуется дополнительный $(n + 1)$ -ый разряд, отражающий знак разности.

Таким образом, для реализации операции вычитания можно применить сумматор, обеспечивающий вычисление выражения

$$M + \bar{S} + 1 = 2^n + D,$$

причем в качестве знакового разряда разности использовать выход переноса сумматора.

Если $M \geq S$ (то есть $D \geq 0$), то на выходе переноса сумматора формируется логическая единица, тогда как значение знакового разряда разности D должно быть равно нулю.

Если $M < S$ (то есть $D < 0$), то

$$M + \bar{S} + 1 = 2^n - |D| = D_{\text{доп}},$$

а на выходе переноса сумматора формируется логический ноль, тогда как значение знакового разряда разности D должно быть равно единице.



Выводы

Следовательно, для получения значения знакового разряда результата вычитания необходимо проинвертировать сигнал с выхода переноса сумматора. При этом разность чисел оказывается представленной в дополнительном коде.

Схема четырехразрядного вычитателя, реализованного на сумматоре и формирующего результат в дополнительном коде, представлена на рис. 4.13.

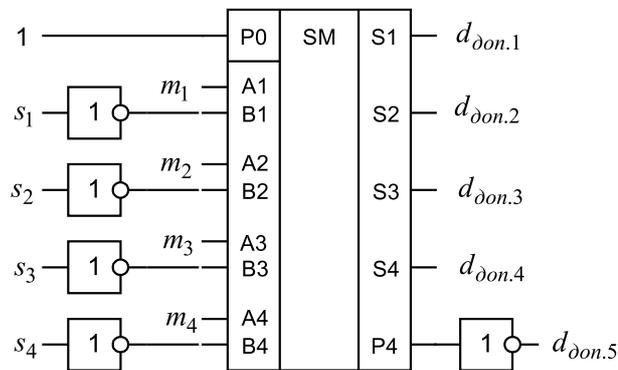


Рис. 4.13 – Схема вычитания четырехразрядных чисел с помощью сумматора

Операции сложения и вычитания можно совместить в одном функциональном узле, если инверторы заменить логическими элементами «исключающее ИЛИ» (рис. 4.14), которые в зависимости от значения управляющего сигнала работают как повторители (при $V = 0$) либо как инверторы (при $V = 1$).

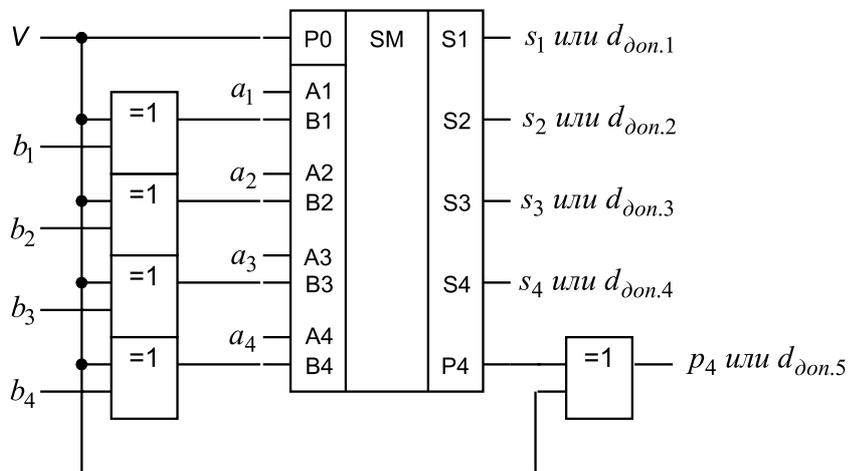


Рис. 4.14 – Схема сложения и вычитания четырехразрядных чисел

4.7 Цифровые компараторы

Цифровые компараторы (от англ. *compare* – сравнивать) выполняют сравнение двух n -разрядных чисел (A и B), заданных в двоичном (двоично-десятичном) коде. В зависимости от схемного выполнения компараторы могут реализовывать следующие функции сравнения: $A = B$ (A равно B), $A = \overline{B}$ (A равно не B), $A \neq B$ (A не равно B), $A < B$ (A меньше B), $A > B$ (A больше B), $A \leq B$ (A меньше либо равно B), $A \geq B$ (A больше либо равно B). Результат сравнения отображается соответствующим логическим уровнем на выходе. Специализированные микросхемы цифровых компараторов, как правило, имеют три выхода, на которых формируются признаки трех основных операций сравнения: $A=B$, $A < B$, $A > B$.

В качестве простейшего одноразрядного компаратора можно использовать двухходовые логические элементы «исключающее ИЛИ» (функции сравнения $A \neq B$, $A = \overline{B}$) и «исключающее ИЛИ-НЕ» (функция сравнения $A = B$).

Условно-графическое обозначение ИМС четырехразрядного компаратора представлено на рис. 4.15.

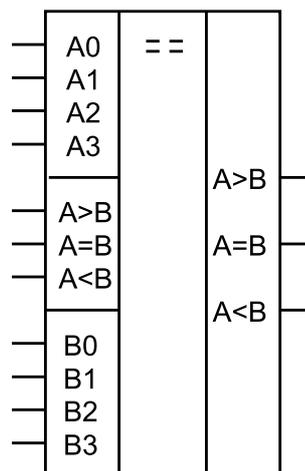


Рис. 4.15 – Условно-графическое обозначение четырехразрядного компаратора

Компаратор имеет расширяющие входы « $A = B$ », « $A < B$ », « $A > B$ », которые позволяют наращивать разрядность сравниваемых чисел без дополнительных логических элементов. Для наращивания разрядности можно использовать каскадное соединение компараторов.

При каскадном соединении (рис. 4.16) входы « $A < B$ » и « $A = B$ » предыдущего компаратора (младшие разряды) подключают к соответствующим входам последующего. На входы « $A < B$ », « $A = B$ », « $A > B$ » компаратора младших разрядов подают сигналы «0», «1», «1» соответственно. В последующих компараторах на входах « $A > B$ » поддерживают сигнал «1».

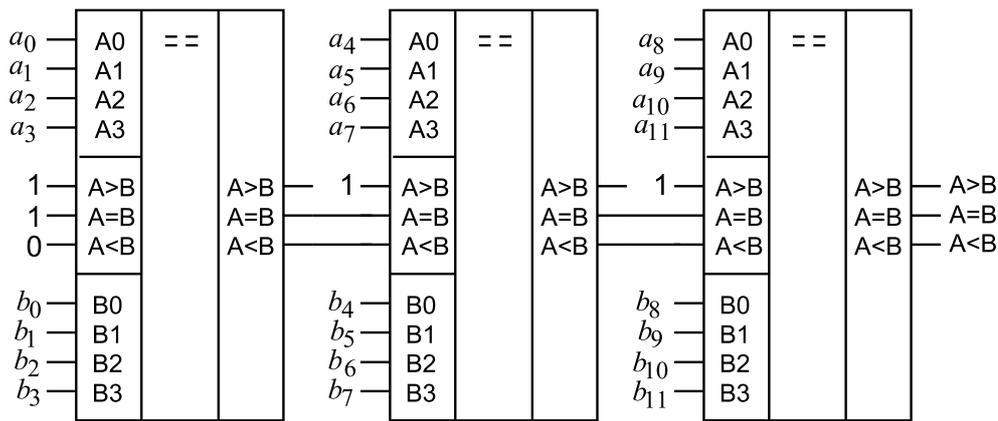


Рис. 4.16 – Каскадное соединение компараторов

4.8 Матричная реализация булевых функций

В качестве функциональных узлов больших интегральных схем, ориентированных на реализацию булевых функций, широко используются так называемые матричные схемы.



Матричная схема представляет собой сетку ортогональных проводников, в местах пересечения которых могут быть установлены полупроводниковые компоненты с односторонней проводимостью — диоды (рис. 4.17) или транзисторы (рис. 4.18).

Матрица конъюнкций M1 (рис. 4.17) имеет шесть горизонтальных и четыре вертикальные шины. Каждая j -ая входная шина этой матрицы ($j = \overline{1,3}$) связана с двумя горизонтальными шинами матрицы M1 (с одной непосредственно, а со второй — через инвертор). Способ включения диодов в местах пересечения матрицы M1 позволяет реализовать на любом из ее выходов p_i ($i = \overline{1,4}$) любую конъюнкцию ее входных переменных x_j ($j = \overline{1,3}$), взятых со знаком либо без знака инверсии. Так, в матрице на рис. 4.17 $p_1 = \bar{x}_1\bar{x}_2\bar{x}_3$, $p_2 = x_1x_3$, $p_3 = x_1x_2$, $p_4 = x_2x_3$.

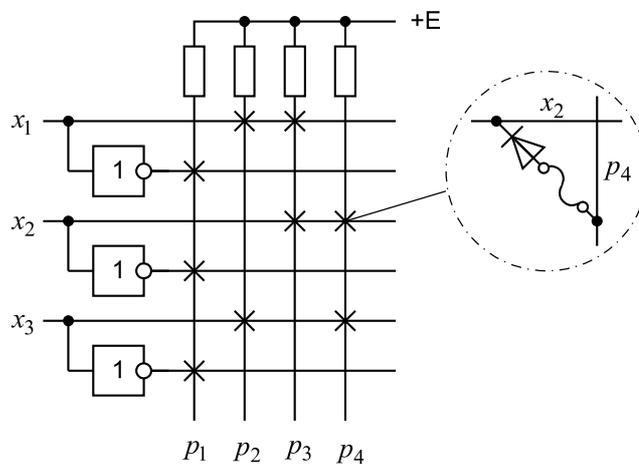


Рис. 4.17 – Матрица конъюнкций

Матрица дизъюнкций М2 (рис. 4.18) имеет четыре вертикальные и две горизонтальные шины. Способ включения транзисторов в местах пересечения шин матрицы М2 позволяет реализовать на любом из ее выходов y_i ($i = \overline{1,2}$) любую дизъюнкцию ее входных переменных p_j ($j = \overline{1,4}$). На рис. 4.18 показан пример реализации элементарных дизъюнкций: $y_1 = p_1 + p_2 + p_3$, $y_2 = p_2 + p_4$.

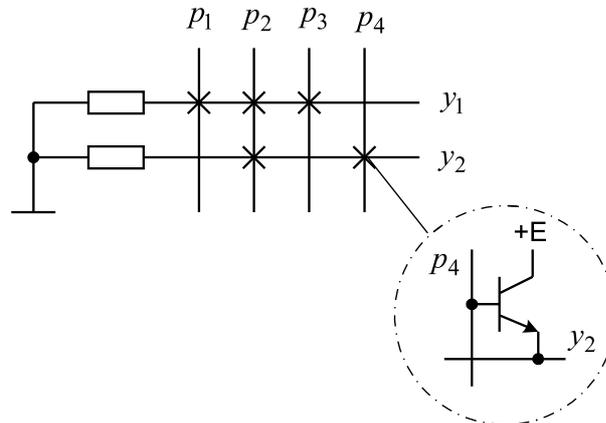


Рис. 4.18 – Матрица дизъюнкций

Для простоты принято матрицы М1 и М2 изображать так, как это показано на рис. 4.19, а и рис. 4.19, б соответственно.

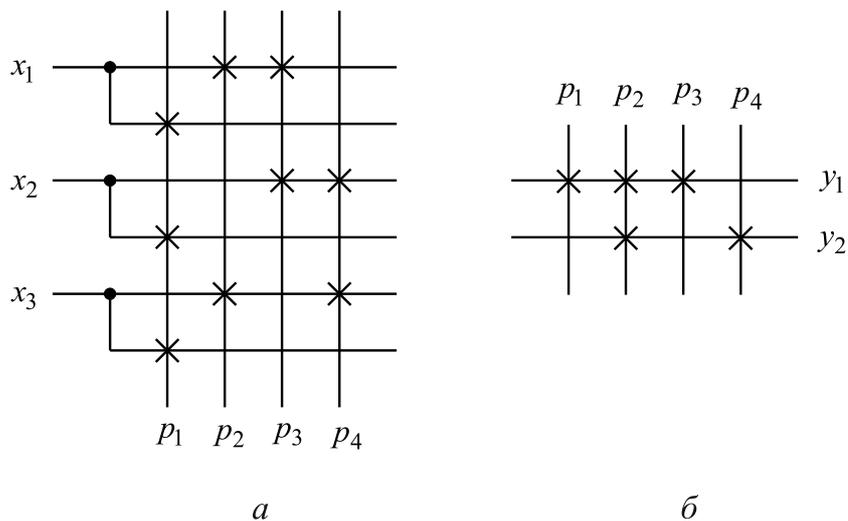


Рис. 4.19 – Упрощенное представление матриц конъюнкций (а) и дизъюнкций (б)

Если соединить матрицы М1 и М2 каскадно, как это показано на рис. 4.20, то полученная двухуровневая матричная схема будет реализовывать следующую систему булевых функций, представленных в ДНФ:

$$y_1 = \bar{x}_1\bar{x}_2\bar{x}_3 + x_1x_3 + x_1x_2, \quad y_2 = x_1x_3 + x_2x_3.$$

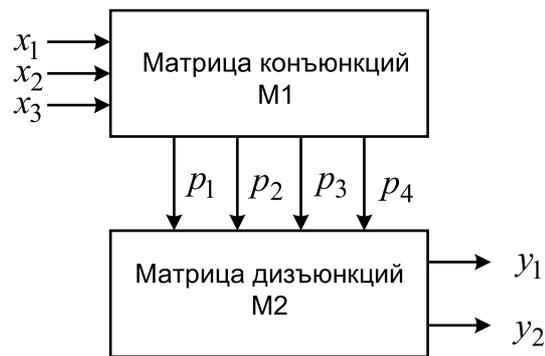


Рис. 4.20 – Двухуровневая матричная схема

Аналогичным образом любая ДНФ системы Θ булевых функций y_1, \dots, y_m входных переменных x_1, \dots, x_n может быть реализована двухуровневой матричной схемой, на первом уровне которой образуются различные элементарные конъюнкции, а на втором — дизъюнкции соответствующих конъюнкций. Таким образом, построение схем с матричной структурой сводится к определению точек пересечения шин, где должны быть включены компоненты с односторонней проводимостью, и настройке матриц (установке компонентов с односторонней проводимостью в найденных точках).



По способу программирования различают:

- матрицы, настраиваемые (программируемые) на заводе-изготовителе;
- матрицы, программируемые пользователем;
- репрограммируемые (многократно настраиваемые) матрицы.

Принято перед сокращенным названием устройств, построенных на основе матриц первого типа, добавлять букву М (от слова «масочно-программируемые»), второго типа — букву П (от слова «программируемые»), и третьего — букву Р (от слова «репрограммируемые»).

В М-матрицах соединение компонентов с односторонней проводимостью с шинами осуществляется один раз с помощью специальных масок, используемых для металлизации определенных участков кристалла БИС. После изготовления БИС полученные соединения изменены быть не могут.

П-матрицы поставляются потребителю ненастроенными и содержащими компоненты с односторонней проводимостью в каждой точке пересечения шин. Настройка П-матриц сводится к удалению (отключению) определенных компонентов. Физически процесс настройки осуществляется различными способами, например путем пропуска серий импульсов тока достаточно большой амплитуды через соответствующий компонент и разрушения плавкой перемычки, включенной последовательно с этим компонентом и соединяющей его с одной из шин в точке их пересечения.

R-матрицы позволяют осуществлять многократно. Повторное программирование выполняется электрическим способом после стирания содержимого матриц под действием ультрафиолетового (иногда рентгеновского) облучения или электрическим способом отдельно для каждого компонента с односторонней проводимостью.

Сложность матричной реализации булевых функций принято оценивать суммарной информационной емкостью (площадью) матриц. В общем случае, если в схеме, аналогичной рис. 4.20, имеется n входов, m выходов и L вертикалей, то суммарная информационная емкость определяется выражением:

$$S(M) = S(M_1) + S(M_2) = 2nL + Lm \quad [\text{бит}].$$

Для сокращения информационной емкости при реализации системы булевых функций необходимо представлять ее в ДНФ с минимальным числом различных элементарных конъюнкций.

Матрицы M_1 и M_2 для реализации системы булевых функций принято условно изображать в виде таблицы, столбцы которой отмечаются переменными x_1, \dots, x_n и функциями y_1, \dots, y_m . Каждой промежуточной шине p_1, \dots, p_L ставится в соответствие строка таблицы.

На пересечении j -ой строки и столбца x_i ($i = 1, \dots, n$) записываются:

- 1, если переменная x_i входит в j -ую элементарную конъюнкцию без инверсии;
- 0, если переменная x_i входит в j -ую элементарную конъюнкцию с инверсии;
- (–), если переменная x_i не входит в j -ую элементарную конъюнкцию.

На пересечении j -ой строки и столбца y_i ($i = 1, \dots, m$) записываются:

- 1, если j -ая элементарная конъюнкция входит в ДНФ функции y_i ;
- точка (•), если j -ая элементарная конъюнкция не входит в ДНФ функции y_i .

В табл. 4.7 приведено условное представление матриц M_1 (рис. 4.19, а) и M_2 (рис. 4.19, б).

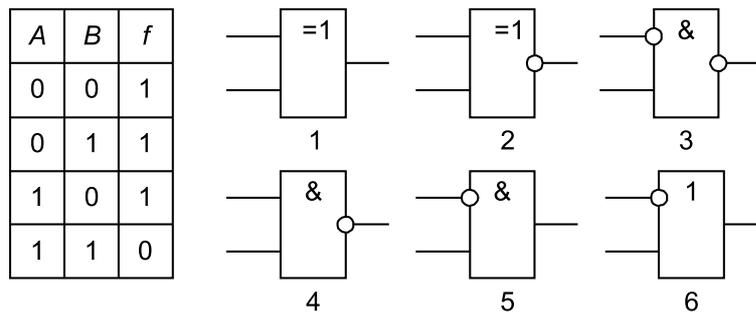
Таблица 4.7 – Условное представление матриц конъюнкций и дизъюнкций

x_1	x_2	x_3	y_1	y_2
0	0	0	1	•
1	–	1	1	1
1	1	–	1	•
–	1	1	•	1

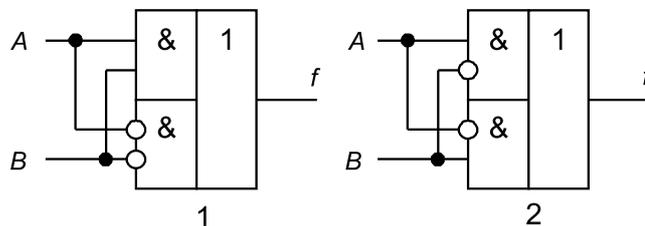


Контрольные вопросы по главе 4

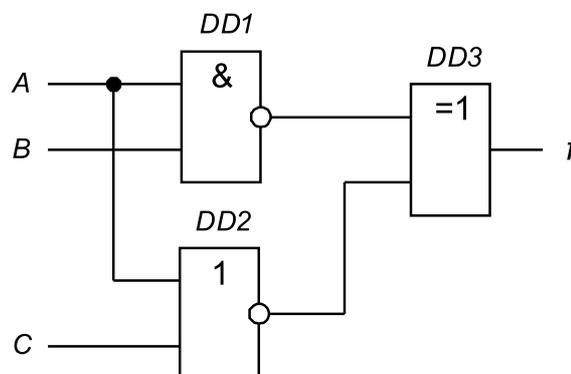
- 1) Указать логические элементы, реализующие булеву функцию, заданную таблицей истинности:



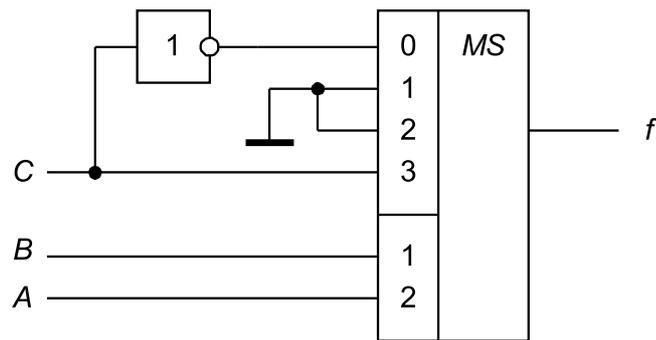
- 2) Записать выражения булевых функций, реализуемых логическими элементами многоступенчатой логики:



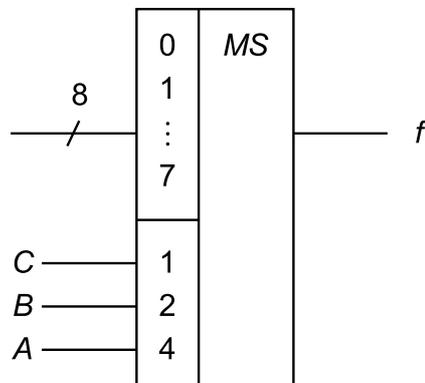
- 3) Записать минимизированное выражение булевой функции, реализуемой комбинационной схемой:



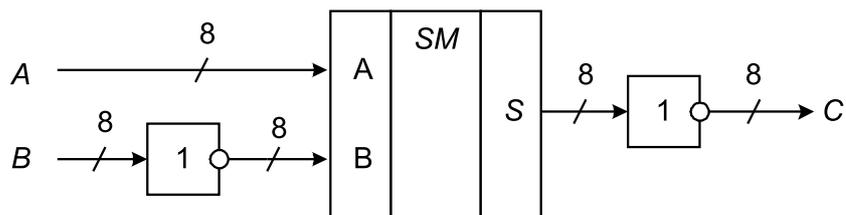
- 4) Записать выражение булевой функции, реализуемой комбинационной схемой:



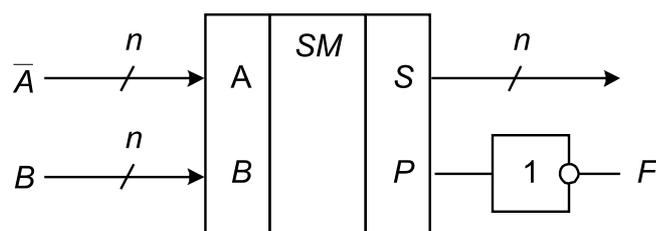
- 5) Указать восьмиразрядное слово, которое необходимо подать на информационные входы мультимплексора для реализации булевой функции $f = ABC\bar{C} + \bar{A}C$:



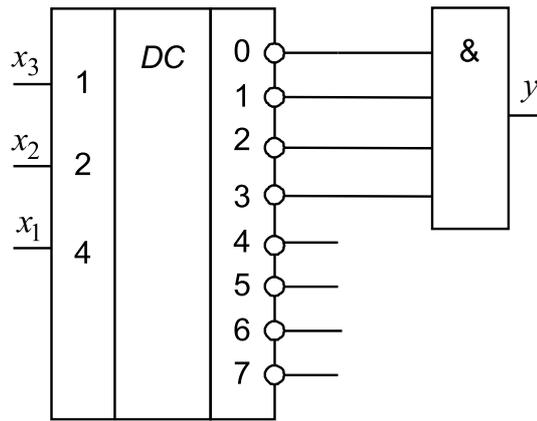
- 6) Представить в десятичной системе счисления число C , формируемое на выходе комбинационной схемы, если $A = 129_{10}$, $B = 150_{10}$:



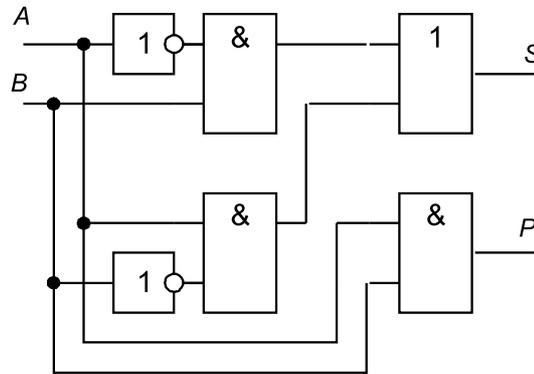
- 7) Определить функцию сравнения цифрового компаратора, выполненного на двоичном сумматоре:



- 8) Записать выражение булевой функции, реализуемой схемой:



9) Записать выражения булевых функций, реализуемых на выходах S и P :



10) Записать выражения булевых функций для матрицы дизъюнкций:

