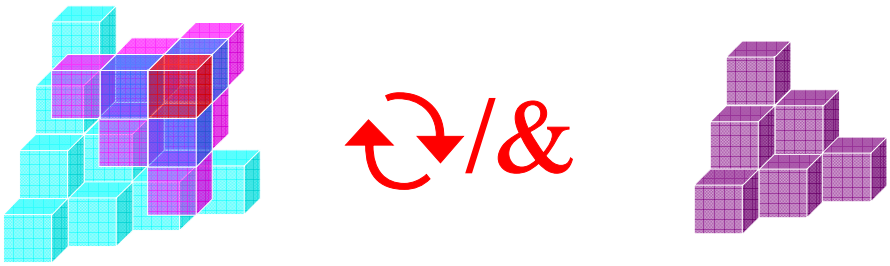
В.И. Кочергин



ТЕОРИЯ МНОГОМЕРНЫХ ЦИФРО-ВЕКТОРНЫХ МНОЖЕСТВ

ФЕДЕРАЛЬНОЕ КОСМИЧЕСКОЕ АГЕНТСТВО

Федеральное государственное унитарное предприятие

"Научно-производственный центр "Полюс"

В.И. Кочергин

ТЕОРИЯ МНОГОМЕРНЫХ ЦИФРО-ВЕКТОРНЫХ МНОЖЕСТВ

Издательство Томского университета

2006

УДК 681.326
ББК 32.973.2
К 55

Кочергин В.И.
К 55 Теория многомерных цифро-векторных множеств. – Томск: Изд-во Том. ун-та, 2006. – 380 с.

ISBN 5-7511-1987-2

Книга содержит систематическое и доступное изложение результатов по созданию теории многомерных цифро-векторных множеств. В основу теории положены классическая теория множеств, цифровая версия пространства и финитная точка зрения, которая рассматривает начала логики и арифметики в виде мысленных экспериментов над наглядно представляемыми овеществленными цифрами расширенного натурального ряда и при этом ограничивает теорию множеств конечными объектами.

Изложение основ теории сопровождается большим количеством примеров синтеза логических и цифровых устройств управляемого быстродействия и повышенной помехозащищенности, работающих в режиме реального времени.

Для разработчиков цифровых систем управления, а также аспирантов и научных работников.

УДК 681.326
ББК 32.973.2

ISBN 5-7511-1987-2
© В.И. Кочергин, 2006

Оглавление

Предисловие 5
Введение 7
Список обозначений и сокращений 19
Глава 1. **Основные положения теории**. 21
1.1. Обычные двухзначные логические функции в многомерном цифровом пространстве 21
1.2. Позиционные системы счисления. 29
1.3. Основные логические операции над подмножествами многомерного цифрового пространства. 35
1.4. Бинарные логические операции в двухмерном цифровом пространстве. 38
1.5. Сложные логические функции в двухмерном цифровом пространстве. 40
1.6. Тернарные логические операции в трехмерном цифровом пространстве. 46
1.7. Сложные логические функции в многомерном цифровом пространстве 47
1.8. Непрерывное множество ряда натуральных чисел в многомерном цифровом пространстве. 53
1.9. Многозначные логические функции в многомерном цифровом пространстве 57
1.10. Алгоритмы синтеза двухзначных логических функций. 61
1.11. Примеры синтеза одноразрядного устройства суммирования и вычитания 65
1.12. Пример анализа неизбыточных кодов позиционной системы счисления 76
1.13. Синтез суммирующих и вычитающих устройств в нетрадиционных двоичных кодах 79
1.14. Синтез одноразрядных умножителей 88
Глава 2. **Синтез умножителей, делителей и счетчиков**. 96
2.1. Синтез устройств умножения 96
2.2. Синтез устройств деления 112
2.3. Синтез реверсивных счетчиков 116
2.4. Синтез управляемых делителей-счетчиков 127
Глава 3. **Контролеспособность позиционных систем счисления** 136
3.1. Расположение ошибок позиционных систем счисления в многомерном пространстве 139
3.2. Многофазный код 152
3.3. Анализ контролеспособности многофазного кода методом многомерных цифровых множеств 153
3.4. Анализ контролеспособности кодов Хемминга методом многомерных цифровых множеств 159
3.5. Анализ контролеспособности обычного цифрового кода методом многомерных цифровых множеств 166
3.6. Анализ контролеспособности кода реверсивного двоичного делителя-счетчика 171
3.7. Геометрический синтез контролеспособных кодов позиционных систем счисления 179
3.7.1. Коды с обнаружением одиночных ошибок 180

3.7.2. Коды с исправлением одиночных ошибок 182
3.7.3. Коды больших оснований систем счисления с исправлением одиночных ошибок 191
3.8. Решение задачи повышения надежности логических и цифровых устройств в режиме реального времени. 195
Глава 4. **Примеры синтеза комбинационных схем** 204
Пример 1. Синтез устройства исправления одиночных ошибок двоичной системы счисления основания n =16. 205
Пример 2. Синтез устройства исправления одиночных ошибок двоичной системы счисления для любых синтезированных кодов основания $n = 2^4$. 219
Пример 3. Синтез одноразрядного сумматора (A ± B) с исправлением одиночных ошибок в системе счисления основания n = 16. 229
Пример 4. Синтез сумматора основания n = 16 предельного быстродействия с исправлением одиночных ошибок. 238
Пример 5. Синтез устройства исправления одиночных ошибок двоичной системы счисления основания $n = 2^4$ без использования записи в ячейки пространства эквивалентных цифр его информационной части. 243
Глава 5. **Систематический код с исправлением одиночных ошибок системы счисления основания $n = 2^{11}$**. 250
5.1. Синтез кода и его кодирующего устройства. 250
5.2. Исправление одиночных ошибок в информационной части кода. . . 257
5.3. Исправление одиночных ошибок в контрольной части кода. 301
5.4. Структурная схема исправления ошибок во всех разрядах систематического кода. 325
Глава 6. **Продолжение геометрического синтеза кодов, исправляющих ошибки**. 328
6.1. Синтез кодов с информационной частью в основном двоичном коде основания $n = 2^4$, исправляющих одиночные и двойные ошибки 335
6.2. Синтез кодов с информационной частью в основном двоичном коде основания 2^{11}, исправляющих все группы ошибок. 351
6.3. Синтез систематических многофазных и интегральных кодов. . . . 356
6.3.1. Двухфазный код 356
6.3.2. Трехфазный код 358
6.3.3. Многофазные коды с числом фаз m = 4, 8, 16, … 361
6.3.4. Многофазные коды с числом фаз m = 6, 9, 12, 15, 18 … 362
6.3.5. Интегральные коды. 364
6.3.6. Синтез декодирующих блоков для системных многофазных и интегральных кодов. 365
6.3.7. Синтез кода, управляющего переключением транзисторов трехфазного инвертора напряжения и исправляющего все одиночные ошибки. 367
Заключение. 368
Литература. 369

Предисловие

Теория многомерных цифро-векторных множеств может рассматриваться как один из многочисленных инструментов теории конечных автоматов. Нередко можно встретить определение, что теория автоматов является частью теории систем – научной дисциплины, появившейся в середине XX в. При этом в теорию систем включают также теорию информации, теорию линейных и нелинейных систем, теорию управления и т.д.

По определению А. Беркса, в более широком понимании, чем представлено выше, «теория автоматов – это наука об основных принципах, общих для искусственных автоматов (цифровые вычислительные машины, аналоговые вычислительные машины, управляющие системы) и естественных автоматов (нервная система человека, самовоспроизводящиеся клетки, организмы в эволюционном аспекте)».

Создателями этого нового направления науки являются Дж. фон Нейман и Н. Винер. Первый из них назвал свой вариант «теорией автоматов», а второй – «кибернетикой».

Автор придерживается именно этого более широкого понимания теории автоматов, которая полностью равноценна определению теории систем.

В предисловии к книге Дж. фон Неймана «Теория самовоспроизводящихся автоматов» А. Беркс отмечал: «В планы фон Неймана входило создать систематическую теорию, математическую и логическую по форме, которая упорядочила бы понятия и принципы, касающиеся структуры и организации естественных и искусственных систем, роли языка и информации в таких системах, программирования и управления такими системами. Теория автоматов лежит на стыке разных дисциплин, объединяет различные подходы (с точки зрения логики, теории связи, физиологии), но, в конце концов, ей предстоит стать отдельной самостоятельной дисциплиной».

Существует огромное количество научных работ, например, по теории конечных автоматов. Чем не устраивают эти математические труды практических инженеров – творцов искусственных автоматов? Ответ на этот вопрос насколько прост, настолько и сложен – все эти теоретические работы считают такие системы однородными, что не соответствует действительности. Только весьма простые автоматы можно условно считать однородными, где можно воспользоваться результатами этих теоретических исследований.

В реальных же автоматах, к которым нужно отнести все виды электроприводов, систем энергоснабжения, систем управления космическими объектами, станками, роботами и т.д., применяются не только однородные двоичные системы счисления, но и иные, например многофазные и интегральные коды с различными, даже со смешанными по типу основаниями систем счисления.

Кроме того, как отмечал Дж. фон Нейман, «в сложных организмах цифровые операции чередуются с аналоговыми процессами», что подсказало ему предложить комбинированную аналого-цифровую машину.

Чередование цифровых и аналоговых операций в равной степени относится ко всем действующим искусственным автоматам – наземным и космическим объектам, созданным человеком, что также не находит отражения в существующей теории автоматов.

Поэтому математическая поддержка существующей теории автоматов нуждается в новых идеях, обеспечивающих ее дальнейшее самостоятельное развитие.

Для объяснения и подтверждения этого обратимся к пониманию задачи математики Дж. фон Нейманом: «Я думаю, что рождение математических идей из опыта, хотя генеалогия этого подчас длинна и запутана, достаточно хорошо приближает истину, слишком сложную, чтобы допускать что-либо, кроме приближений. Но как только эти идеи сформировались, математика начинает жить своей собственной жизнью, и ее лучше уподоблять какой-нибудь творческой дисциплине, движимой почти исключительно эстетическими мотивами, чем чему-нибудь еще, например эмпирической науке. Имеется, однако, еще одно обстоятельство, которое, по моему мнению, необходимо подчеркнуть… На большом расстоянии от эмпирического источника или после интенсивного «абстрактного» инбридинга[1] математический предмет оказывается перед опасностью вырождения… Когда такое состояние достигнуто, единственным, как мне кажется, лекарством становится омолаживающее действие источника: более или менее прямое повторное впрыскивание эмпирических идей».

Именно «впрыскиванию» этих идей с целью ответа на первый основной вопрос теории автоматов, работающих в режиме реального времени (каким образом можно сконструировать надежную, помехозащищенную систему из ненадежных элементов?), служит теория многомерных цифро-векторных множеств. Отчасти это было сделано автором в книге «Теория многомерных цифровых множеств в приложениях к электроприводам и системам электропитания» (Томск: Изд-во Том. ун-та, 2002). Однако в ней основное внимание было уделено практическому использованию теории в соответствующих приложениях, что не позволило полностью отразить все ее возможности по синтезу надежных, помехоустойчивых систем из ненадежных элементов.

[1] Инбридинг – скрещивание двух близкородных организмов (*биол.*).

Математика делится на множество специальных областей, и каждая из них может занять всю отпущенную нам короткую жизнь…

А. Эйнштейн

Введение

Приступая к изложению математической теории, нужно дать определение этого понятия и предназначение её как для математиков, так и для инженеров, на которых, прежде всего, рассчитана книга.

При определении понятия «*теория формальная*» в математической науке дается ссылка на следующие определения: *аксиоматический метод*, *формальная система*, *метатеория*.

Аксиоматический метод – способ построения научной теории, основой которой являются некоторые исходные положения, называемые аксиомами, а все остальные предложения теории получаются как логические следствия аксиом.

В математике *аксиоматический метод* зародился в работах древнегреческих геометров («Начала» Евклида около 300 лет до н.э.). Вершина *аксиоматического* метода была достигнута в работах Д. Гильберта и его школы в виде так называемого метода формализма в основаниях математики. В рамках этого направления была выработана следующая стадия уточнения понятия аксиоматической теории, а именно понятие *формальной системы*.

Формальная система, *дедуктивная система* – в математической логике не интерпретированное *исчисление*, задаваемое правилами образования выражений этого исчисления и правилами построения *выводов* в этом исчислении. Выражения *формальной системы* рассматриваются как чисто формальные комбинации символов; правила вывода определяют, в каких случаях одно формальное выражение **A** выводится из других формальных выражений $\mathbf{B_1}$, …, $\mathbf{B_n}$. Если **n**=0, то **A** называется аксиомой. При этом выводы представляют собой либо последовательности, либо древовидные фигуры, составленные из формальных выражений согласно правилам вывода. Если в вершинах дерева вывода находятся только аксиомы, то формальное выражение, завершающее вывод, называется *выводимым* в *формальной системе*.

Критика *аксиоматического метода* применительно к теории числа выдающегося немецкого математика Ф. Клейна была выражена следующим образом: «Невозможно чисто логическим путем показать, что законы, в которых мы обнаружили отсутствие логического противоречия, действительно имеют силу по отношению к числам, столь хорошо нам известным эмпирически, что неопределенные объекты, о которых здесь идет речь, могут быть отождествлены с реальными числами, а выкладки, которые мы над ними производим, – с реальными эмпирическими процессами».

Необходимо отметить, что немецкие математики Д. Гильберт и П. Бернайс признали эту критику. В двухтомной монографии, изданной уже после смерти Ф. Клейна (*Гильберт Д., Бернайс П.* Основания математики) в Германии в 30-х годах прошлого века и переизданной в 1968 г. издательством Шпрингера, они представили дополнительно *финитную* точку зрения, которая содержит «прямые содержательные рассуждения, совершаемые в виде мысленных экспериментов над наглядно представляемыми объектами и не зависящие от предложений аксиоматического характера определенными геометрическими фигурами. Рассуждения такого рода мы для краткости будем называть *финитными*, а методическую установку, лежащую в основе этих рассуждений… *финитной* установкой или *финитной* точкой зрения. В том же самом смысле мы будем говорить о финитных понятиях и утверждениях, подчеркивая всюду словом *финитный*, что рассматриваемое рассуждение, утверждение или определение придерживается рамок принципиальной представимости объектов и принципиальной выполнимости операций, а тем самым происходит в *рамках конкретного рассмотрения*». И далее: «Всеобщее суждение о цифрах *финитно* может быть истолковано в гипотетическом смысле, т.е. как высказывание о произвольно заданной цифре. Такое суждение провозглашает некоторый закон, который должен подтверждаться в каждом конкретном случае». Еще далее: «…и все-таки в *арифметике потребность в выходе за пределы финитной точки зрения на самом деле не является такой уж ощутимой*»

Более определенно можно утверждать, что для рассмотрения законов арифметики и математической логики нет необходимости использовать *аксиоматический метод*, поскольку арифметика и математическая логика имеют дело только с наглядно представляемыми объектами – цифрами натурального ряда, которые не зависят от предложений аксиоматического характера, а, по словам немецкого математика Л. Кронекера, восходят к Богу: «Господь Бог создал натуральные числа, а все остальное дело рук человечества».

Теория многомерных цифро-векторных множеств является математической теорией и, хотя она отстранена от *аксиоматического метода*, тем не менее не входит в противоречие со всеми глобальными установками математической науки. В самом деле, связь комбинационных логических и арифметических функций с цифрами расширенного натурального ряда очевидна и реализуется через многомерные таблицы истинности. Следовательно, прямые содержательные рассуждения, совершаемые в виде мысленных экспериментов над истинными объектами, также являются истинными, а *финитный метод* исследования здесь является основным.

Кратко теорию многомерных цифро-векторных множеств можно охарактеризовать единством следующих составляющих:

1. Классическая теория множеств Г. Кантора.
2. Расширенный натуральный ряд чисел (0, 1, 2, …), которые выступают в качестве элементарных подмножеств этого множества.

3. Векторное представление цифр разрядов числа, где вершина цифрового вектора обозначается максимальной (n–1)-й цифрой разряда n-го основания системы счисления.

4. Идеи великого русского ученого Е.С. Федорова [58] по упаковке физического пространства определенными геометрическими фигурами, например кубиками, и наше предложение нумерации их числами расширенного натурального ряда.

5. Представление мерности цифрового пространства разрядностью числа.

6. Введение в эту наглядную геометрию пространства мысленных поворотов относительно осей симметрии цифро-векторного пространства.

7. Многозначные таблицы истинности для логических функций и многозначные таблицы математических операций в пределах разрядов операндов, которые также истинны. Именно эти таблицы осуществляют непосредственную связь логических и арифметических функций с числами расширенного натурального ряда.

8. Эквивалентность геометрических фигур, которые представляют собой соответствующую логическую функцию в пространстве различной мерности.

9. Определение логических и арифметических функций посредством покрытия их геометрических образов (цифровых множеств), где покрытием данного цифрового множества называется конечная совокупность его подмножеств, дающая в своей сумме все это цифровое множество.

10. Применение финитной точки зрения, которая рассматривает начала логики и арифметики «в виде мысленных экспериментов над наглядно представляемыми объектами» (в нашем случае – овеществленными цифрами расширенного натурального ряда) и не зависит от предложений аксиоматического характера, а также использует конечные множества конечных объектов.

Следует также признать, что Д. Гильберт не считал *аксиоматический метод* единственным для доказательства построения той и или иной теории. Обратимся к общеизвестной книге, изданной в Германии в 1932 г. и неоднократно переиздаваемой в СССР [2], в предисловии к которой он написал: «В математике, как вообще в научных исследованиях, встречаются две тенденции: тенденция к абстрактности – она пытается выработать логическую точку зрения на основе различного материала и привести весь этот материал в систематическую связь – и другая тенденция, тенденция к наглядности, которая стремится к живому пониманию объектов и их *внутренних* отношений». И далее: «…наглядное понимание играет первостепенную роль в геометрии, и притом не только как обладающее большой доказательной силой при исследовании, но и для понимания и оценки результатов исследования».

Многие исследователи ссылаются на классиков, но, по словам канадского математика Г. Глинского (Международный симпозиум по теории релейных устройств и конечных автоматов, 1962 г.), «есть одна опасность в обращении к классическим работам. Кажется, что на них все ссылаются, но редко кто их действительно читает». По этой причине аксиоматический метод многие исследователи считают единственным для построения математической теории,

ссылаясь при этом на великого ученого Д. Гильберта, что не соответствует действительности.

Теория многомерных цифро-векторных множеств не является алгебраической, а скорее относится к геометрии, хотя и необычной. Она опирается на идею академика Е.С. Федорова по упаковке физического пространства определенными геометрическими фигурами. Здесь следует особо отметить, что Д. Гильберт является единственным из великих математиков XX в., кто уделил должное внимание математическим работам Е.С. Федорова. В отмеченной выше книге содержится несколько разделов, где он рассматривает «кристаллографические (Федоровские) группы движений на плоскости и пространстве» [2. С. 78–101].

Теперь обратимся к философскому определению термина «теория» (Философский словарь. М.: Политиздат, 1968).

Теория – система обобщенного знания, объяснения тех или иных сторон действительности. Термин «*теория*» имеет различные значения: для того чтобы отличать ее от практики или для противопоставления *гипотезе* (как непроверенному, предположительному знанию).

Этот термин, как нам кажется, полностью применим и для математической науки, в которой, по мнению известного польского математика Анджея Мостовски, существует «единственная непротиворечивая точка зрения, согласующаяся не только со здравым смыслом, но и с математической традицией, которая сводится по существу к допущению того, что источник и высший смысл понятия числа лежит в опыте и практической применяемости. То же относится и к теории множеств в том объеме, в каком они необходимы для классических областей математики» [*Клайн М.* Математика. Утрата определенности. М.: Мир, 1984]. Поэтому, если угодно, представляемая теория многомерных цифро-векторных множеств относится к прикладной математике и предназначается для решения практических задач анализа и синтеза логических и цифровых систем, устанавливаемых непосредственно в системах автоматического управления, а также и к теории автоматов, история которых начинается в сороковых годах XX в. с исторических работ Дж. фон Неймана, который мечтал построить систематическую теорию, логико-математическую по форме, позволяющую понять как естественные, так и искусственные автоматы. При этом в качестве разновидности сложных автоматов им рассматривались вычислительные машины, «которые выделяются тем, что дают нам надежду хоть что-то понять». Он считал, что естественные автоматы являются тем идеалом, к которому должны приближаться искусственные автоматы.

Естественные автоматы работают только в режиме реального времени по параллельному принципу, а искусственные – по последовательному принципу. Стремление распараллеливания вычислений является одной из важных задач увеличения быстродействия вычислительной техники.

Другим важным преимуществом естественных автоматов являются эффективные меры самоконтроля и самоисправления. «Если бы каждую ошибку надо было немедленно обнаруживать, объяснять и исправлять, то система та-

кой сложности, как живой организм, не смогла бы проработать и миллисекунды. Система такого типа так хорошо скомпонована, что может работать, когда в ней возникают отдельные ошибки. Одна ошибка, как правило, еще не означает тенденцию к разрешению… Эта философия коренным образом отличается от той, по которой первая же ошибка означает конец мира» [7]. К такому построению системы также должны приближаться искусственные автоматы.

Работа цифровых устройств систем автоматического управления в режиме реального времени предъявляет высокие требования к ним по быстродействию и надежности. Именно эти аспекты являются предметом наших исследований.

В 80-х годах XX в. многие ученые считали, что «модели одиночного вычислителя исчерпаны, а скорость работы элементов близка к теоретической. На смену модели одиночного вычислителя пришла модель коллектива вычислителей…» [29]. При этом использование сложной компьютерной техники, состоящей из нескольких компьютеров, становилось «модным», и их начинали применять не только там, где это было правомерно, но и там, где оно было ничем не оправдано и вело только к увеличению стоимости изделий и снижению надежности системы.

Это предсказание не оправдалось, так как возможности одиночного вычислителя далеко не исчерпаны, и поэтому быстродействие компьютеров в течение последних двадцати лет росло впечатляющими темпами. Этот рост достигался только за счет увеличения рабочих частот, что тоже имеет свои естественные пределы, и нельзя надеяться, что рабочая частота станет выше частоты атомных переходов (порядка 10^{14}–10^{16} Гц). Следовательно, необходимо искать иные пути повышения быстродействия операций машинной арифметики для систем реального времени, предлагая нетрадиционные алгоритмы параллельных вычислителей.

Тем не менее в той части утверждения [29], что наступила эпоха коллектива вычислителей, авторы оказались абсолютно правы. В настоящее время весьма остро стоит вопрос о параллелизме на различных уровнях, где многие виды параллелизма гибко реализуются в вычислительных сетях программными методами, а пользователям предоставляется возможность распоряжаться ими по собственному усмотрению.

Однако это обстоятельство не снимает с повестки дня поиск путей повышения быстродействия, помехозащищенности и надежности одиночных вычислителей, что автоматически повышает также и возможности коллектива вычислителей.

Именно одиночные вычислители составляют предмет данного исследования, исходя из их возможностей по параллельному выполнению операций, а также устойчивости к помехам. В этом и заключается отличие управляющего цифрового устройства от обычного, для которого естественным является последовательный режим выполнения операций.

Одним из путей решения этих задач является использование программируемых логических интегральных схем [ПЛИС или ПЛУ – Programmable Logic

Devices (PLDs)]. Это новая технология проектирования электронных схем, где в основу архитектуры положена структура программируемых матриц логики [ПЛМ – Programmable Array Logic (PALs)]. Основу ПЛМ составляют две матрицы «И» и «ИЛИ», где программируется матрица «И», а матрица «ИЛИ» имеет фиксированную настройку.

Проектирование таких устройств получило название двухуровневого синтеза, в отличие от многоуровневого, который используется при проектировании цифровых систем на основе FPGA (Field Programmable Gate Array) или CPLD (Copmlex PLD).

В отечественной литературе нет четкости в определении таких схем, которые обычно называются ПЛИС или ПЛУ. Поэтому остановимся на этих определениях более подробно.

История PLD начинается с 70-х годов XX в. с появлением программируемых постоянных запоминающих устройств [ППЗУ – Programmable Read Only Memory (PROM)]. В дальнейшем для реализации систем булевых функций стали выпускаться программируемые логические матрицы [ПЛМ – Programmable Logic Array (PLAs)], которые можно считать первыми PLD. Совершенствование архитектуры PLD привело к созданию программируемых матриц логики ПМЛ. В дальнейшем при расположении на одном кристалле нескольких PAL, которые объединялись программируемыми соединениями, были созданы сложные ПЛУ [Copmlex PLD (CPLD)], а созданные ранее PLD стали называть стандартными [Standart PLD (SPLD)].

Одновременно с PLD создавались структуры вентильных матриц [Gate Array (GA)] и матриц логических ячеек [Logic Cell Array (LCA)], которые в дальнейшем превратились в программируемую пользователем вентильную матрицу [Field Programmable Gate Array (FPGA)].

Теоретическую основу методов синтеза этих устройств составляют алгебра Буля (булева алгебра), математическая логика [18, 61, 65] и др. Методы двухуровневого синтеза комбинационных схем используют теорию булевых или переключательных функций, которые также применяются на ранних этапах многоуровневого синтеза [11]. Решению задач минимизации булевых функций посвящено такое большое количество работ, что простое их перечисление вызывает значительные сложности. Остановимся только на некоторых из них.

Классическим решением задачи минимизации булевых функций принято считать работу E.J. Mc Cluskey [101], а все последующие можно разбить на группы: *эвристические алгоритмы* [32, 41, 69, 81, 82]; *точные алгоритмы* [32, 65, 92, 120]; *алгоритмы минимизации частично определенных булевых функций* [105]; *алгоритмы минимизации слабо определенных булевых функций* [104], *алгоритмы минимизации системы булевых функций* [30] и многие другие.

В [75] для операции с булевыми функциями используется диаграмма двоичных решений [Dinary Decision Diagrams (DDs)], а в [111] предложено использовать графическое представление этих функций [IF-decision Diagrams (IFDs)], которое, по мнению авторов, позволяет повысить уровень параллель-

ности логических вычислений. *Метод неявного представления простых импликант* [81] позволяет осуществлять минимизацию булевых функций с большим числом аргументов.

Классические методы декомпозиции булевых функций лежат в основе методов многоуровневого синтеза. В [66] представлен *метод дизъюнктивной декомпозиции на основе декомпозиционных диаграмм,* который в [83] был распространен на множественную композицию. Более компактная форма представления булевых функций в виде покрытия нулевых и единичных значений была выполнена в [115]. Все эти методы были развиты в [84], где приведены точные и приближенные алгоритмы минимизации многовыходной логической сети при ограничениях на число уровней сети и на коэффициенты расширения выходных вентилей. Однако представленные здесь методы приспособлены только для решения задач малой размерности, а для практического применения возможно использование только приближенных методов многоуровневого синтеза, которые основаны на локальных преобразованиях логических сетей [57].

Первые методы для решения задач многоуровневого синтеза были основаны на дизъюнктивном разложении К.А. Шеннона [*Shannon C.* A symbolic analisis of relay and switching circuts // Trans. of American Inst. of Electrical Engineirs. 1938. № 57], который был создан им для контактных схем, но они не привлекли внимания разработчиков аппаратуры. В [31] представлено несколько подходов к решению задачи декомпозиции при синтезе комбинационных схем на PLA, которые получили названия: *стандартная декомпозиция* [31]; *ортогонализация интервалов* [4]; *тождественные отображения и эвристические алгоритмы реализации метода тождественных отображений* [62], которые в работах большого числа авторов [63, 64, 71, 74, 80, 94, 113] были усовершенствованы и позволили решать задачи декомпозиции с точки зрения минимизации числа входов PLA нижнего уровня и числа используемых промежуточных шин PLA.

Наиболее популярными подходами к синтезу PLA комбинационных схем с большим числом входов являются *методы матричной факторизации* [30, 38, 37] и *метод дизъюнктивно-конъюнктивного разложения* [52].

Методы совместной декомпозиции рассматриваются в [3], а задача *совместного дизъюнктивно-инверсного* разложения системы булевых функций – в [5]. В [10] рассматривается *множественная декомпозиция булевых функций*, сводящихся к задаче решения логического уравнения.

Другие способы многоуровневого синтеза успешно используют при проектировании логической сети алгебраические методы [70, 71, 73], которые, в отличие от булевых методов, могут применяться на технологически независимом уровне. Они быстрее булевых и поэтому позволяют решать задачи большого размера, но булевы методы, по мнению ряда авторов, более точные, и поэтому в [72] используется сочетание алгебраических и булевых методов.

При многоуровневом синтезе, как и в двухуровневом, используются многозначные функции [3, 96], в том числе с многозначными входами и выходами [107].

Другие методы и программы многоуровневого синтеза логических функций приведены в работах [30, 78, 80, 90, 95, 97–99, 107–109, 114, 121].

Подробные обзоры методов синтеза комбинационных схем можно найти в работах [116, 118, 119] и в отечественных: [4, 53, 60].

Необходимо отметить недостатки, которые не позволяют непосредственно применять ряд этих методов для синтеза комбинационных схем на современных ПЛИС. К ним следует отнести следующие:

а) часть научных статей и методов, которые в них предлагаются, носят абстрактный характер и представляют только теоретический интерес с точки зрения постановки математической задачи, но не дают четких практических рекомендаций;

в) некоторые методы ориентированы на очень узкий класс решаемых задач, и нередко примеры, которые приводятся в этих работах, весьма просты;

б) почти все методы не позволяют эффективно использовать архитектурные особенности современных схем ПЛИС;

г) непригодность булевой алгебры логики для синтеза недвоичных контролеспособных систем счисления, которые используют, например, геометрические коды.

Устранение этих недостатков и разработка универсального метода синтеза любых логических и цифровых устройств являются основным предназначением данной книги.

Опыт применения современных компьютерных систем убеждает в том, что интегральная технология уменьшает интенсивность катастрофических отказов элементов, но не уменьшает интенсивности случайных сбоев. Поэтому значение автоматического контроля современных цифровых систем не стало менее важным, чем оно было для изделий прежних поколений. Эта роль контроля значительно возрастает вследствие того, что успехи цифровой техники значительно расширили ее область применения в тех задачах, решение которых требует большой ответственности. Помехоустойчивость цифровых и логических устройств является второй областью исследования данной книги.

Для обнаружения и исправления ошибок при выполнении арифметических операций рядом исследователей было предложено использовать специальные арифметические коды. Эти коды имеют большое число разновидностей, и с их помощью предполагалось контролировать не только арифметические операции, но и правильность передачи и хранения информации, что являлось предпосылками для создания единой аппаратной системы контроля и устранения последствий неисправностей. Кратко остановимся на этих исследованиях. Прежде всего, это AN-*коды* и *остаточные коды* (*коды в остаточных классах*).

Применение AN-*кодов* требует предварительного умножения исходных чисел на некоторое заранее выбранное число A с последующим выполнением арифметических операций над числами вида AN, где в дальнейшем осуществляется декодирование с исправлением или обнаружением ошибок, которые также требуют проверки определенных вычислений. Исследования AN-*кодов* начинались с работ [85, 76], а в дальнейшем развивались в работах зарубежных [40, 112, 86–89] и отечественных авторов [6–8, 33, 94, 16, 122].

Теория AN-*кодов* интенсивно развивалась в 70-е годы XX в., но затем многие из исследователей пришли к выводу, что для практического применения в вычислительных системах наиболее приспособлены *остаточные коды*.

Остаточные коды являются систематическими, т.е. они содержат в себе кроме информационных и контрольные разряды. В этих кодах арифметические операции выполняются параллельно над исходными операндами и их наименьшими вычетами по некоторым выбранным модулям. После завершения операций вычисленные вычеты результата операции по этим модулям сравниваются с результатами аналогичной операции над вычетами операндов. Теория остаточных кодов впервые была представлена в [110] и очень успешно развивалась в работах отечественных авторов [1, 15 19–26, 28, 39]. В дальнейшем исследователи обратили внимание на необходимость использования циклической структуры AN-*кодов*, где существовала аналогия между циклическими AN-*кодами* и кодами для передачи информации в системах связи, например кодами Хемминга. Теоретическим исследованиям циклических AN-кодов с контролем по модулю посвящено большое количество работ, среди которых необходимо отметить следующие: [17, 27, 36, 56, 51, 102, 103, 67, 77, 106, 93, 117].

Однако в практическом плане радужные перспективы применения этих кодов в цифровых устройствах не дали каких-либо положительных результатов и ограничились только контролем четности (нечетности). Недостатки применения этих кодов рассматриваются в работах [12–14]. Самый существенный из них заключается в уменьшении номинального быстродействия цифровых вычислительных систем, по мнению Е.И. Брюховича, не менее чем вдвое. Эта оценка явно занижена, так как быстродействие будет еще ниже, поскольку им учитывалась только операция обнаружения ошибок и не учтено время на исправление этих ошибок.

Известно, что кодирование для любых типов кодов, в том числе и арифметических, не составляет проблемы, и процесс кодирования сводится к вычислению наименьших вычетов по заданному модулю. Подобную задачу необходимо решать на первом этапе декодирования (обнаружение ошибок) при вычислении синдрома, для чего используются схемы свертки [42, 44]. Организация работы на втором этапе декодирования (исправление ошибок) вычислительного комплекса, использующего арифметические коды, обсуждается в работах [39, 43, 59, 55], а перестановочное декодирование – в [91] и частично в [122].

Декодирование при исправлении одиночных пакетов ошибок для циклических AN-кодов было исследовано в [9], а для двойных пакетов ошибок – в [54].

Широкое применение в технике связи для операции исправления ошибок методов мажоритарного декодирования циклических кодов вызвало попытку решить подобную задачу в вычислительных комплексах для AN-*кодов* [79, 93], но кроме теоретических предпосылок это не принесло практических результатов. Это также можно сказать о декодировании m-мерных итеративных циклических кодов [6].

Рассмотрение арифметических кодов проводится всеми исследователями для реализации только одной арифметической операции – суммирования, которая считается ими базовой для всех других операций (умножения, деления, возведения в степень и т.д.). Такой подход нельзя считать корректным, но даже при таком упрощенном решении возникают сложности с сигналами переноса. В [122] рассматривается в теоретическом плане использование весовых соотношений, связывающих операнды, сумму и переносы, для исправления одиночных ошибок в работе сумматора.

Решение задач обнаружения и исправления ошибок цифровых систем видится Ю.Г. Дадаевым в «создании специального процессора исправления, работающего независимо и параллельно с выполнением ЭВМ основных функций. Такой процессор можно использовать для исправления ошибок, возникающих не только при выполнении операций, но и при продвижении операндов на всем пути от машины до исполнительных устройств» [28]. Это предложение решения задачи можно принять к реализации, но только не для систем реального времени, где необходимо иметь подобный «сопроцессор», превосходящий по быстродействию основной вычислительный комплекс, быстродействие которого здесь находится на пределе возможного.

Ряд исследователей видит решение задачи повышения помехоустойчивости цифровых систем в использовании систем счисления с иррациональным основанием типа «золотой пропорции», которая была предложена в [68]. Избыточные самосинхронизирующие коды Фибоначчи [100] рассматриваются в [45–50] как альтернатива существующим позиционным системам счисления и, по мнению А.П. Стахова, сохраняют все их известные преимущества: «простота сравнения чисел по величине и “наглядность” в изображении чисел, простота арифметических правил, возможность представления чисел с фиксированной и плавающей запятой, однородность и итеративность реализующих арифметику структур и др.». Однако здесь же говорится о том, что применение этих кодов в вычислительной технике «является спорным и его решение требует дальнейших исследований».

Основным предназначением *теории многомерных цифро-векторных множеств* является развитие методов анализа и синтеза цифровых устройств систем управления, работающих в режимах реального времени, где применяются не только двоичные системы счисления, но и контролеспособные избыточные системы при выполнении любых арифметических и логических операций.

В [34, 35] автором были кратко представлены основные положения теории многомерных цифро-векторных множеств, где основное внимание было уделено синтезу автоматов, использующих многофазные коды.

Рассматривать этот новый тип автоматов стало возможным после того, как многофазные напряжения, составляющие основу большого количества технических устройств (преобразовательная и измерительная техника, электропривод, электротехнические приборы и т.д.), были представлены автором не импульсными, а полностью цифровыми. При таком подходе стало возможным решить следующие задачи:

Создать устройства машинной арифметики, выполняющие все известные операции в любых кодах позиционных систем счисления, в том числе естественных кодах систем энергоснабжения и электроприводов, – многофазных кодах и кодах многомерных цифровых множеств.

Создать методику автоматизированного синтеза цифровых и логических устройств с заданными параметрами контролеспособности и минимизации аппаратурных затрат.

Разработать основные положения теории логических матриц, представляющих геометрический образ логической функции в многомерном цифровекторном пространстве.

Предложить основные правила векторного преобразования геометрических фигур многомерного цифрового пространства с учетом всех видов симметрии, отражающие также симметрию программ покрытия этих фигур.

При обсуждении специалистами этих работ автору были высказаны пожелания расширить теоретическую часть материалов и рассмотреть системы счисления больших оснований, где осуществляется обнаружение и исправление ошибок, а также изложить материалы по повышению живучести автоматических устройств.

Это пожелание автор стремился реализовать в данной книге. Основатель теории автоматов Дж. фон Нейман [7], несмотря на большое значение математической логики в теории автоматов, которое он отмечал, был убежден, что она не адекватна «истинной» логике автоматов, и считал, что это будет новая логика, которая будет объединять различные дисциплины. Эту цель поставил перед собой автор данной работы, стараясь при этом избежать, по словам Д.И. Менделеева, трех губительных крайностей: «утопий мечтательности, желающей всё постичь одним порывом мысли; ревнивой косности, самодовольствующейся обладаемым; кичливостью скептицизма, ни на чем не решающегося остановиться».

«Чистые» математики с XIX в. решают многочисленные проблемы упаковки шаров в Евклидовом пространстве, которые они рассматривают в контексте алгебраической геометрии и алгебраической теории чисел, где основное приложение видится ими в теории чисел, позволяющих, например, исправлять ошибки в различных кодах восьмимерного и двадцатимерного пространства.

Однако неоправданная для задачи исправления ошибок сложность этой теории, которую они трактуют как «глубокую» математику в сравнении с «не-

глубокой» математикой, связанной с укладкой кубиков, ограничивает ее понимание и, следовательно, практическое применение. При этом они считают, что «все задается координатами, и нет необходимости рисовать картинки» [*Конвей Дж.*, *Слоен Н.* Упаковки шаров, решетки и группы: В 2 т. / Пер. с англ. М.: Мир, 1990].

Подобный подход видится автору как непродуктивный, который ведет к излишнему усложнению этой задачи и вводит большое количество ненужных для ее решения терминов. Автор понимает, что он вызовет недовольство определенных кругов «чистых» математиков, но он хочет напомнить им слова известного английского философа Локка Джона (1632–1704): «А знание, конечно, гораздо более продвинулось бы в мире, если бы старанья даровитых и трудолюбивых людей не загромождались ученым, но легкомысленным употреблением неуклюжих жеманных или непонятных терминов».

Если есть возможность обратиться к зрительному восприятию постановки и решения конкретной задачи, то неразумно отказываться от такого подхода. Великий Леонардо да Винчи восклицал: «Неужели не видишь ты, что глаз объемлет красоту всего мира?.. Он направляет и справляет все искусства человеческие, двигает человека в разные части света. Он – *начало математики*. Способности его несомненнейшие. Он измерил высоту и величину звезд. Он нашел элементы и их место. Он породил архитектуру и перспективу, он породил божественную живопись. О, превосходное из всех вещей, созданных богом! Какие хвалы в силах изобразить твое благородство! Какие народы, языки какие сумеют хотя бы отчасти описать истинное твое действие!»

К этому утверждению необходимо добавить высказывание нашего современника А.Д. Гончарова: «Точный язык науки и техники, язык положений и выводов, цифр и формул художник должен обогатить образами, чтобы читатель мог усвоить нужную информацию легко, в краткий срок, в формах зрительно ясных и эстетически полноценных».

Перейдем к изложению «неглубокой» математики по укладке кубиков в Евклидовом пространстве и будем решать задачи, которые пока не решены «глубокой» математикой.

При этом в качестве примеров приложений, без которых невозможно проследить за развитием математических идей, будем ориентироваться большей частью на задачи цифровых устройств электроприводов и систем электроснабжения. Это не является ограничением области применения теории, поскольку без устройств электроприводов и систем энергоснабжения уже давно не мыслится существование любых отраслей техники.

Список обозначений и сокращений

Представленные ниже обозначения используются систематически на протяжении всей книги. Отсутствующие в этом списке символы используются в качестве локальных обозначений, смысл которых меняется при переходе из одной главы в другую и даже из одного пункта в другой.

ППЗУ – программируемое постоянное запоминающее устройство (PROM – Programmable Read Only Memory)]
БИС – большая интегральная схема
СБИС – сверхбольшая интегральная схема
ПЛИС – программируемая логическая интегральная схема или ПЛУ – программируемое логическое устройство (PLD – Programmable Logic Device)
ПЛМ – программируемая логическая матрица (PLAs – Programmable Logic Arrays)
ПМЛ – программируемая матрица логики (PAL – Programmable Array Logic)
GA (Gate Array) – вентильная матрица
LCA (Logic Cell Array) – матрица логических ячеек
FPGA (Field Programmable Gate Array) – программируемая пользователем вентильная матрица
CPLD (Complex PLD) – сложное ПЛУ
SPLD (Standard PLD) – стандартное ПЛУ
ОЦК – обычный цифровой код
$ЛФ_k$ – k-значная логическая функция
ЛФ ($ЛФ_2$) – двухзначная логическая функция
N_0 – расширенный натуральный ряд чисел
ДНФ – дизъюнктивная нормальная форма записи ЛФ
A, B, C, … – запись операндов в прямом коде
$\mathbf{A^{\bullet}, B^{\bullet}, C^{\bullet}, \ldots}$ – запись операндов в обратном коде
0, 1, 2, … – цифры ОЦК
1* – логическая единица
0* – логический нуль
$\underline{x}$ – отрицание
λ – рабочая область многомерного цифрового пространства
A, B, C, … – операнды, множества, векторы, одномерное цифровое пространство
AB, AC, BC, … – двухмерные цифровые пространства
ABD, ACD, BCD, … – трехмерные цифровые пространства и т.д.
$\mathbf{U_i}$ – вектор напряжения, соответствующий кодовой комбинации многофазного кода для цифры **i**

μ (i – 1) = **0** $\vee$ … $\vee$ (**i – 1**) – непрерывное множество цифр ОЦК

M(i) = **0** $\vee$ … $\vee$ **i** = **1** $\vee$ … $\vee$ **i** = … = **i** – любое цифровое множество, содержащее цифру **i**

d_a(i) – кодовое расстояние между кодовыми комбинациями из i разрядов операнда **A**

w_a(i) – весовые значения кодовых комбинаций из i разрядов операнда **A**

$d_{a,x}$ – кодовые расстояния между кодовыми комбинациями систематического кода (**A** – информационная часть кода, **X** – контрольная часть кода)

d_x(2,3) – кодовые расстояния контрольной части квазисовершенного двухфазного кода

d_x(3,3) – кодовые расстояния контрольной части квазисовершенного трехфазного кода

$d_{a,x}$(m;2,3) – кодовые расстояния системного многофазного кода (m – число фаз информационной части кода; 2,3 – контрольная часть квазисовершенного двухфазного кода)

$d_{a,x}$(m;3,3) – кодовые расстояния системного многофазного кода (m – число фаз информационной части кода; 3,3 – контрольная часть квазисовершенного трехфазного кода)

Математика – это не царица наук, она занимает в мире иное, куда более почетное положение: она является служанкой (естественных и гуманитарных) наук, помогая им, доставляя им адекватный аппарат для описания всевозможных фактов и явлений.

И.М. Яглом

Пространство есть не что иное, как частный случай трижды протяженной величины.

К.Ф. Гаусс

Глава 1

ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ

Предложенный в [17] нестандартный подход к управляющим дискретным логическим устройствам и цифровой технике включает в их состав не только машинную арифметику и цифровые системы управления, но и электропривод, преобразовательную измерительную и силовую технику и т.д. Предлагаемая вниманию теория многомерных цифро-векторных множеств является инструментом для синтеза в этих областях техники оптимальных устройств, исходя из расширенного натурального ряда чисел N_0.

1.1. Обычные двухзначные логические функции в многомерном цифровом пространстве

Первые понятия, с которых начинается любая теория, должны быть ясны и приведены к самому наименьшему числу. Только тогда они могут служить прочным и достаточным фундаментом этой теории.

Одним из таких понятий является представление обычной двухзначной логической функции в многомерном цифровом пространстве. Прежде чем выполнить это представление, дадим некоторые пояснения.

Во-первых, в работе будут использоваться только позиционные системы представления числа, история которых начинается за две тысячи лет до н.э., когда вавилоняне [18] употребляли шестидесятеричную систему счисления с позиционным принципом записи числа. Современное представление позиционной системы счисления определяется следующим образом: «Систему счисления называют позиционной, если одна и та же цифра может принимать различные численные значения в зависимости от номера местоположения (разряда) этой цифры в совокупности цифр, представляющих заданное число. Позиционные системы разделяют на однородные и смешанные. Во всех разрядах числа, представленного в однородной системе, используются цифры из одного и того же множества. Например, в обычной десятичной системе во всех разрядах любого числа используются цифры из множества {0, 1, ... , 9}, в двоичной

системе – цифры из множества {0, 1} и т.п. В смешанных системах множества цифр различны для разных разрядов. В тех случаях, когда в позиционной системе для каждой цифры имеется отдельный символ, её называют системой с непосредственным представлением чисел. В позиционных системах с кодированием чисел количество символов меньше, чем количество цифр, а каждая цифра кодируется определенной комбинацией из нескольких символов» [21].

Во-вторых, под кодом будем понимать «правило преобразования сообщения из одной символьной формы представления (исходного алфавита) в другую (объектный алфавит)» [22]. В нашем случае цифры натурального ряда конкретного основания системы счисления составляют одну символьную систему (систему с непосредственным представлением чисел). Другая символьная система определяет принцип кодирования, где обычно количество символов меньше, чем количество цифр кодируемого основания системы счисления.

В-третьих, поскольку в большей части коды, в том числе и система с непосредственным представлением чисел, являются циклическими, то необходимо дать определение циклического кода: «Подпространство **V** наборов длины **n** называется циклическим подпространством или циклическим кодом, если для любого вектора **V** = (a_{n-1}, a_{n-2}, ..., a_0) из подпространства **V** вектор **V'** = = (a_0, a_{n-1}, a_{n-2}, ..., a_1), получаемый в результате циклического сдвига компонента вектора **V** на единицу вправо, также принадлежит подпространству **V'**» [19].

Теперь можно приступить к представлению обычной двухзначной логической функции в многомерном цифровом пространстве.

Обычной двухзначной логической функцией, в дальнейшем логической функцией ($ЛФ_2$ либо, опуская индекс 2, ЛФ) $y = f(x_1, \ldots, x_k)$ называется функция, принимающая, как и её аргументы x_1 **, ... ,** x_k, два значения, которые обозначим, чтобы отличать от сигналов **0** и **1** обычного цифрового кода (ОЦК), символами 0* и 1*.

Поскольку число различных наборов значений аргументов ЛФ конечно и равно 2^k, любая ЛФ может быть задана (на примере k = 3) таблицей истинности (табл. 1.1.1).

Таблица 1.1.1

ОЦК	x_3	x_2	x_1	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*

В таблице наборы значений аргументов расположены в порядке возрастания: сначала идет набор (иное название набора – кодовое слово или кодовая комбинация), представляющий собой двоичное расположение числа **0** (0* , ... , 0*), затем следует набор, являющийся двоичным расположением числа **1**; за ним

записывается набор, соответствующий числу **2**, и т.д. Последний набор состоит из К единиц (1* , ... , 1*) и является двоичным разложением числа натурального ряда ($2^K - 1$). Такое кодирование цифр натурального ряда называется двоичным.

Если всем наборам значений аргументов ЛФ $y = f(x_1, \ldots, x_K)$ поставить в соответствие сигналы ОЦК от **0** до ($2^K - 1$), т.е. цифры расширенного натурального ряда, то каждый аргумент x_i ($i = 1, \ldots, K$) и сама ЛФ будут представлять собой определенные цифровые множества, что для нашего примера будет выражено следующим образом:

$$x_1 = 1 \vee 3 \vee 5 \vee 7;\quad x_2 = 2 \vee 3 \vee 6 \vee 7;\quad x_3 = 4 \vee 5 \vee 6 \vee 7;\quad y = 1 \vee 2 \vee 4 \vee 7.$$

На отрезке конечной длины цифровой «прямой» (рис. 1.1) графически представлены соотношения между этими цифровыми множествами.

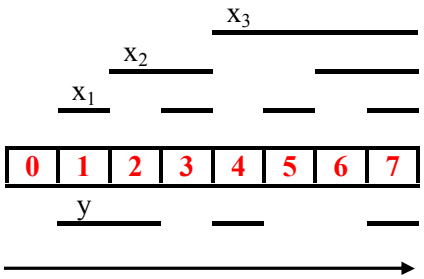


Рис. 1.1

Определение же бесконечной цифровой «прямой» следующее: геометрическое изображение бесконечного множества натуральных действительных чисел N_0 , когда натуральное число получается последовательным прибавлением **1**, начиная с **0**, причем **0** является равноправным элементом наряду со всеми другими цифрами **1**, **2**, **3** и т.д.

Следовательно, если на бесконечной «прямой» g заданием начала отсчета и положительного направления цифрового вектора откладывать последовательно единичные отрезки, соответствующие цифрам **0**, **1**, **2**, ... , то каждый единичный отрезок L «прямой» g однозначно определяется своей координатой N_0. Таким образом, каждому единичному отрезку L «прямой» g соответствует одно действительное натуральное число N_0, и обратно: каждому натуральному числу N_0 соответствует один элемент, расположенный на конце цифрового вектора.

При таком соответствии каждая последовательность 2^K различных наборов значений аргументов с учетом порядка их расположения определяет принцип кодирования сигналов ОЦК от **0** до ($2^K - 1$) .

Наиболее широкое распространение получило так называемое двоичное кодирование, которое является основным во всей вычислительной технике. Исходя из этого, последовательность набора аргументов, соответствующую двоичному принципу кодирования, будем считать стандартной [24] формой представления, что и выполнено в табл. 1.1.1 и на рис. 1.1.

При основном двоичном кодировании так же, как и при других способах кодирования, можно рассматривать цифровую «прямую» с элементами от **0** до ($2^K - 1$) как графическое представление системы счисления одного основания, равного 2^K, либо как часть системы счисления с одним основанием, большим чем 2^K. Эту бесконечную «прямую» можно представить также последовательным соединением разрядов числа с меньшими основаниями, равными по зна-

чению либо различными $2^K = 2^L\, 2^S\, 2^T \ldots$ ($K = L + S + T + \ldots$). Такое представление числа позволяет изобразить графически каждый аргумент x_i и саму функцию $y = f\ (x_{1\,,\,\ldots\,,}\ x_K)$ не только на цифровой «прямой» (одномерное цифровое пространство), но и в многомерном варианте этого пространства. Причем каждая координата многомерного цифрового пространства определяется соответствующим разрядом либо группой разрядов числа.

При этом ЛФ $y = f\ (x_{1\,,\,\ldots\,,}\ x_K)$, принимающая для всех наборов аргументов x_i значение 1*, представляет собой непрерывное цифровое множество сигналов от **0** до ($\mathbf{2^K - 1}$), т.е. то многомерное цифровое пространство E (универсальное цифровое множество), в котором размещается «объем» любой ЛФ $y = f\ (x_{1\,,\,\ldots\,,}\ x_K)$ с меньшим числом наборов аргументов x_i, равных 1*.

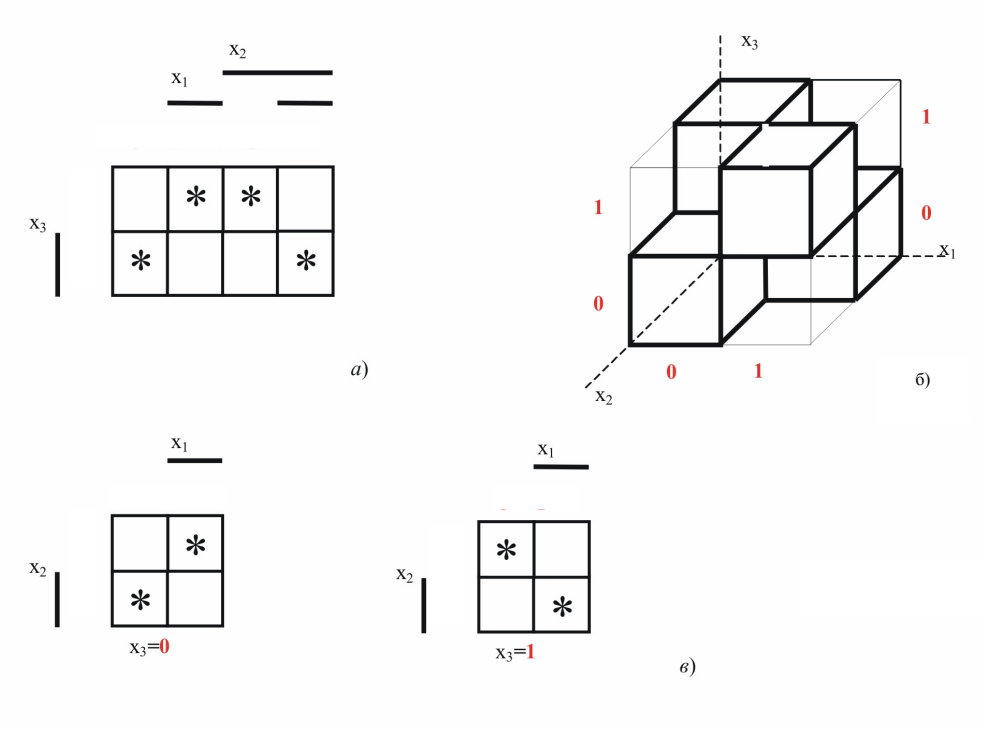


Рис. 1.2

Для трехместной (тернарной) логической операции $y = f\,(x_1\,,\,x_2\,,\,x_3)$ ЛФ, заданная табл. 1.1.1 и представленная (см. рис. 1.1) на цифровой «прямой» (одномерное пространство), может быть также изображена в двухмерном пространстве (см. рис. 1.2, *а*), трехмерном пространстве (см. рис. 1.2, *б*) либо в трехмерном пространстве, нарисованном послойно (см. рис. 1.2, *в*). В этом случае можно говорить, что геометрический образ логической функции $y = f\ (x_{1\,,\,\ldots\,,}\ x_K)$ одномерного цифрового пространства однозначно может отображаться на двухмерных и трехмерных цифровых пространствах.

Во всех этих представлениях число ячеек пространства одно и то же, а число разрядов и их оснований различно. Цифровые пространства с равным количеством ячеек будем называть эквивалентными. Здесь и в дальнейшем

значение ЛФ, равное 1*, обозначим звёздочкой (*), а соответствующая ячейка пространства, где ЛФ равна 0*, будет пустой.

Для того чтобы увидеть все ячейки многомерного цифрового пространства, уменьшим их масштаб, но сохраним их положение в декартовой системе координат. Тогда геометрический образ логической функции рис. 1.2 в трехмерном цифровом пространстве координат x_1, x_2, x_3 будет преобразован в рис. 1.3.

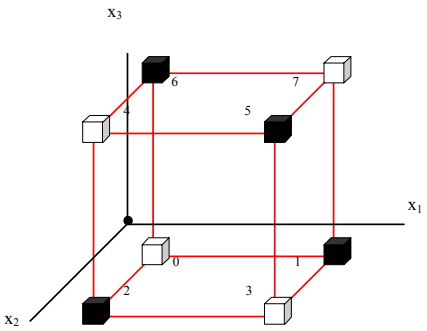


Рис. 1.3

На этих рисунках «длина», «площадь», «объем» определяются мерностью пространства (одно-, двух-, трех-) и эквивалентны друг другу, а также полностью определяют логическую функцию

$$y = x_1 \underline{x_2}\, \underline{x_3} \vee \underline{x_1}\, x_2\, \underline{x_3} \vee \underline{x_1}\, \underline{x_2}\, x_3 \vee x_1\, x_2\, x_3 .$$

В этом выражении и в дальнейшем инверсию функции или аргумента (входными и выходными сигналами) будем обозначать не верхней чертой над сигналом, как это обычно выполняется, а нижней. Это нововведение принято нами для того, чтобы отличать логические функции, полученные из геометрического представления, от логических функций, которые выводятся аналитическим путем, исходя из алгебры Буля. Известно, что булева алгебра может быть использована только при двоичном принципе кодирования соответствующих оснований позиционных систем счисления и не пригодна при геометрических принципах кодирования этих оснований систем счисления.

В общем случае можно утверждать, что покрытие (определение термина «покрытие» приведем ниже) «объема» цифрового множества ЛФ, расположенного в многомерном цифровом пространстве, определяет все возможные варианты построения ЛФ. Здесь следует напомнить, что число таких логических функций в научно-технической литературе известно, но подсчет их числа осуществляется только путем просмотра (или, как говорят, «перебора»). При этом утверждается, что «уже при сравнительно небольших значениях K ($K \geq 6$) перебор становится практически невозможным даже с использованием вычислительной техники» [25].

Использование геометрического образа логических функций позволяет получить точный ответ: это число определяется суммой всех возможных сочетаний из 2^K элементов цифрового множества соответственно по нулю (во множестве нет логических единиц – пустое множество $\varnothing$), по 1 (во множестве содержится одна логическая единица), по 2 (в множестве содержится по две логических единицы) и т.д. вплоть до сочетания из 2^K по 2^K (в множестве все элементы равны логической единице – универсальное множество E).

Отмеченное записывается следующим образом:

$$2^{2^k} = C_{2^k}^{0} + C_{2^k}^{1} + C_{2^k}^{2} + \dots + C_{2^k}^{2^k}. \qquad (1.1.1)$$

В формуле (1.1.1) первое и последнее слагаемые определяют соответственно пустое и универсальное множества, сумма которых, как очевидно, равна двум.

Следует отметить, что значения ЛФ могут быть представлены не на всех 2^K возможных наборах значений аргументов, и тогда на некоторых из них они не определены. Названия таких ЛФ и их «объемов» – неполностью определенные или частичные.

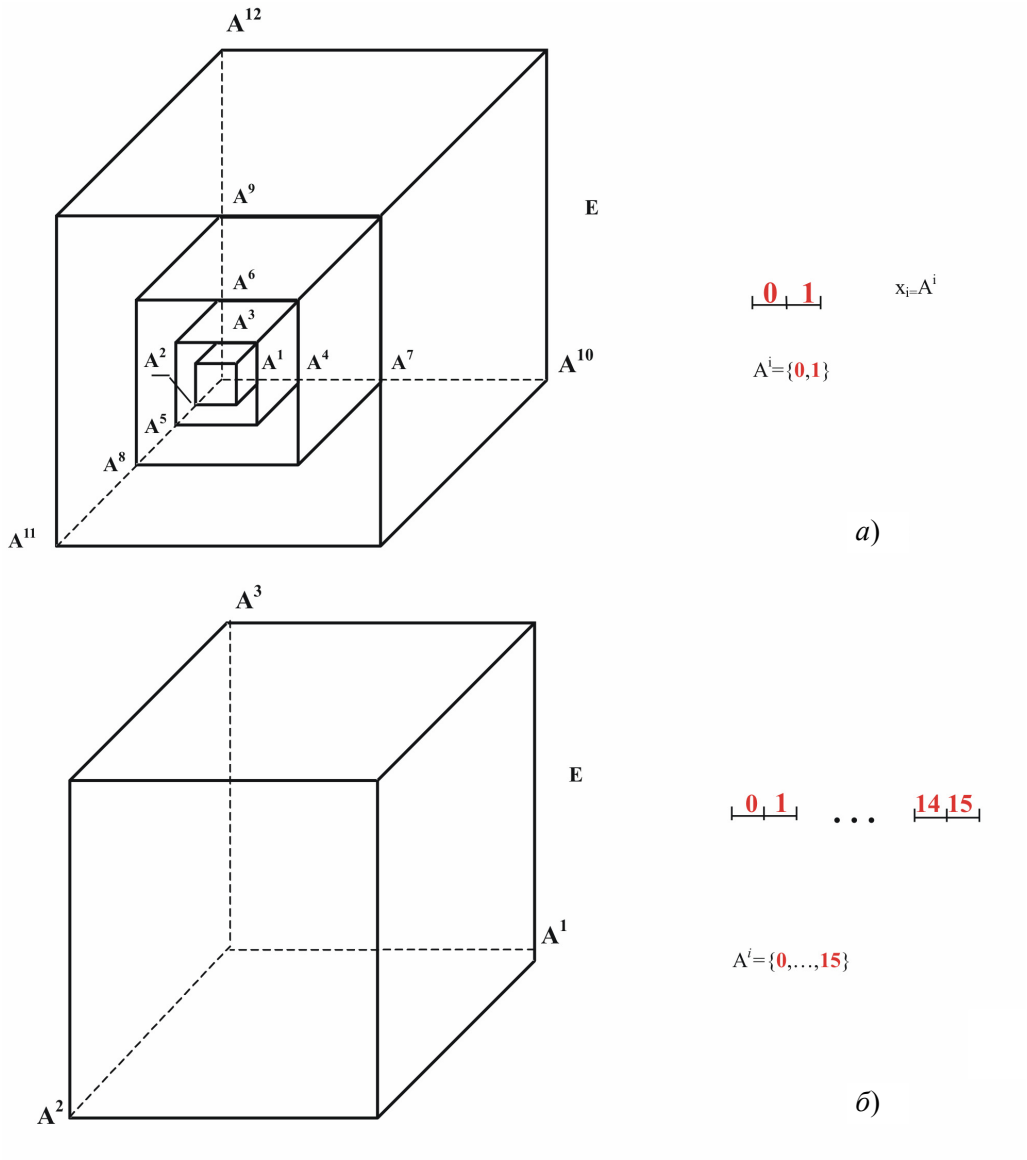


Рис. 1.4 (начало)

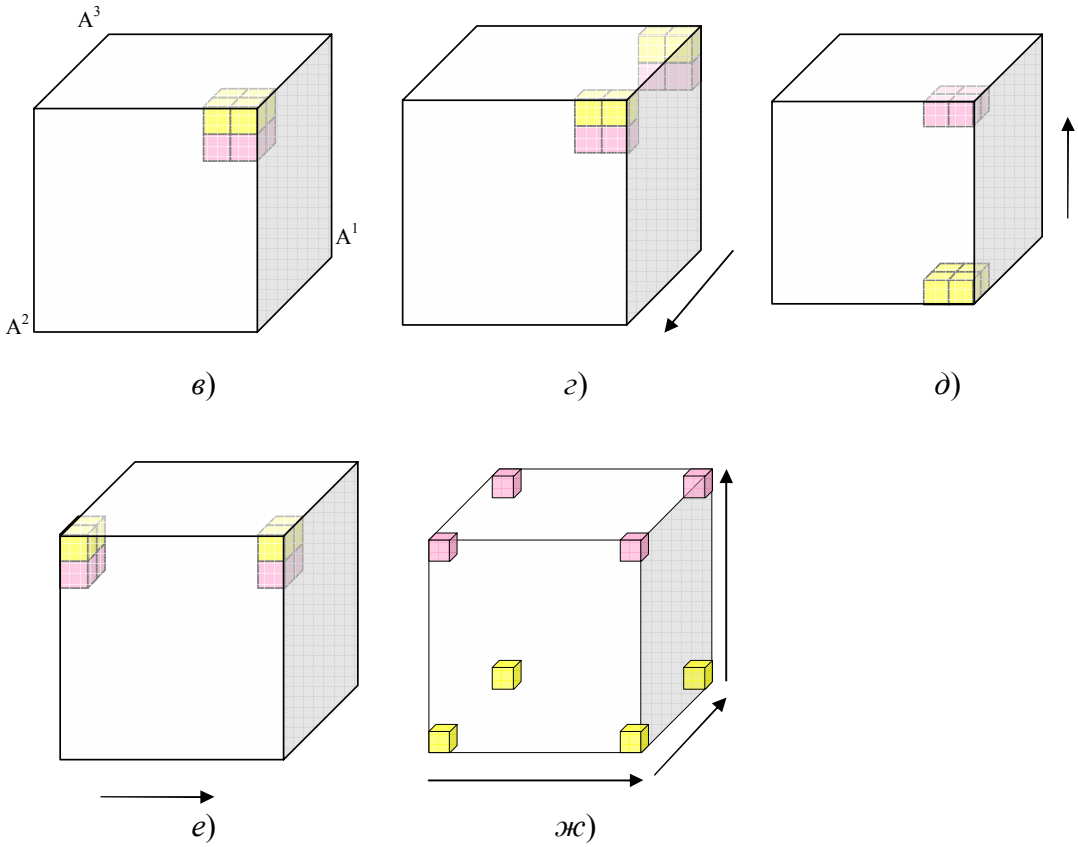


Рис. 1.4 (продолжение)

Для более четкого представления многомерного цифрового пространства, использующего декартову систему координат, на рис. 1.4, *а* изображено двенадцатимерное цифровое пространство, где каждый разряд имеет основание системы счисления 2 (цифры **0**, **1**), а на рис. 1.4, *б* – эквивалентное ему трехмерное цифровое множество, где каждый разряд имеет основание системы счисления 16 (цифры **00**, ... , **15**), которые могут быть, в свою очередь, закодированы различными способами, например четырьмя разрядами двоичного кода.

Таким образом, для геометрического представления ЛФ может быть выбрано цифровое пространство мерности от 1 до K, где все пространство упаковывается элементарными фигурами от **0** до **($2^K - 1$)** (нумерованное цифровое пространство). Объединение всех этих 2^K элементарных фигур представляет собой универсальное цифровое множество E, включающее все возможные наборы значений K аргументов. Выделение в универсальном множестве E подмножеств элементарных фигур, соответствующих наборам аргументов, при которых ЛФ $y = f(x_1, \ldots, x_K)$ равна 1*, образует ее геометрический образ.

В качестве измерения мерности пространства нами выбрано число разрядов соответствующего основания системы счисления, из которых может быть составлено любое действительное натуральное число N_0, нумерующее 2^K на-

боров аргументов ЛФ. Поэтому геометрический образ многомерного цифрового пространства должен обладать аналогично цифрам разряда свойством замкнутости этих цифр на себя. Представить физическое существование одномерного либо двухмерного цифрового пространства, где цифры разрядов, определяющих мерность пространства, замкнуты на себя, не сложно. Для одномерного цифрового пространства, «склеивая» между собой свободную грань элементарного кубика, соответствующего цифре **0** и перпендикулярного свободной гранью к цифровой оси, со свободной гранью такого же кубика, соответствующего цифре $(2^K - 1)$ и также перпендикулярного свободной гранью к цифровой оси, получим «полый» тор. Проводя аналогичные «склеивания» цифр разрядов двухмерного цифрового пространства, получим тор с внутренней полостью и т.д.

Таким путем может быть получено достаточно сложное тороидальное многомерное цифровое пространство.

Однако здесь необходимо остановиться и сделать следующее замечание.

Несмотря на то, что декартова система координат и тороидальная являются ортогональными и имеют взаимно однозначные соответствия [18], обращаться к более сложной и менее изученной тороидальной системе координат нет необходимости. Такой переход только «затуманит» рассмотрение, а представить упомянутую выше замкнутость цифр разрядов, т.е. координат на себя, можно и в декартовой системе координат.

В самом деле, обратимся к трехмерному цифровому пространству декартовой системы координат (см. рис. 1.4, *в*), где на границе этого пространства расположена геометрическая фигура, состоящая, например, для простоты изложения из восьми элементарных кубиков. Для наглядности дальнейшего изложения половина кубиков имеет более темный цвет.

При сдвиге этой фигуры, которая, как мы условились ранее, является геометрическим образом ЛФ, вдоль координаты A_1 на одну цифру (см. рис. 1.4, *г*) в большую сторону она появится одной своей половинкой в начале координаты A_1. Аналогичные положения займет эта фигура при сдвиге на одну цифру относительно координат A_2 и A_3 (см. рис. 1.4, *д*, *е*).

Одновременный сдвиг этой фигуры, например, по всем трем координатам A_1, A_2, A_3 (см. рис. 1.4, *ж*), приведет к тому, что она будет присутствовать своими частями в начале и конце всех трех координат.

Здесь следует помнить, что исходная геометрическая фигура (см. рис. 1.4, *а*) при любом сдвиге её вдоль координат преобразуется в аналогичные образы (см. рис. 1.4, *г – ж*), которые все эквивалентны между собой, т.е. всегда сохраняют свою форму (не дробятся и не объединяются в более крупные образования) и всегда при этом находятся внутри многомерного цифрового пространства.

Если при этом многомерное цифровое пространство не бесконечно, а для основания системы счисления используется циклический код, то отмеченное выше свойство пространства позволяет сразу сделать важный вывод: число всех возможных логических функций, представленных формулой (1.1.1), по-

требует при наших условиях для своей реализации в n^k раз меньше принципиальных схем, чем число логических функций формулы (1.1.1).

Это доказывается тем, что циклические перестановки символов (аргументов) кодового слова вновь приведут к этому кодовому слову. Следовательно, одна принципиальная схема, эквивалентная исходной логической функции, при циклической перестановке сигналов на её входах будет эквивалентна n^k логическим функциям.

Существуют и другие методы для уменьшения числа принципиальных схем, реализующих логические функции многомерного цифрового пространства, но это является предметом нашего дальнейшего рассмотрения.

Уже в начале изложения основ теории многомерных цифровых множеств, где размещаются геометрические образы логических функций, сделаны два серьёзных вывода. Эти выводы, по нашему мнению, не могли быть получены, исходя из чисто аналитических логических рассуждений. Это полностью подтверждает высказывание Ф. Клейна (F.C. Klein), критикующего чисто формальную теорию чисел: «Попытка совершенно изгнать созерцание и удержать только логическое исследование представляется мне в полной мере неосуществимой. Некоторый остаток, некоторый минимум интуиции всегда должен сохраниться…». И далее: «…совершенно невозможно чисто логическим путем показать, что законы, в которых мы обнаружили отсутствие логического противоречия, действительно имеют силу по отношению к числам, столь хорошо нам известным эмпирически, что неопределенные объекты могут быть отождествлены с реальными числами, а выкладки, которые мы над ними производим, – с реальными эмпирическими процессами» [16].

1.2. Позиционные системы счисления

Позиционные системы счисления занимают основную, главную роль во всей математике, и их использование при двоичном принципе кодирования оснований имеет в машинной арифметике также всеобщий характер. Тем не менее использование двоичного кодирования не может определить все возможные варианты кодирования оснований систем счисления. Нетрудно показать, что число кодов, эквивалентных сигналам ОЦК от **0** до (**2^K – 1**) , равно 2^K!.

В самом деле, каждой цифре от **0** до (**2^K – 1**) основания системы счисления 2^K сопоставляется его код – слово из алфавита $\{x_1, x_2, \ldots, x_k\}$. При двоичном неизбыточном законе кодирования, например для К = 3, это будут следующие слова (кодовые комбинации): **0** ↔ $\underline{x}_1\ \underline{x}_2\ \underline{x}_3$, **1** ↔ $x_1\ \underline{x}_2\ \underline{x}_3$, … , **7** ↔ $x_1\ x_2\ x_3$. Здесь цифры **0**, … , (**2^K** – **1**) и конъюнкции К-го ранга находятся во взаимно однозначном соответствии, что определяется постановкой между ними соответствующего знака (**↔**). Другие взаимно однозначные соответствия могут быть получены перестановкой конъюнкций К-го ранга.

Обозначим конъюнкции К-го ранга простыми цифрами 0, … , (2^K – 1), совпадающими с двоичным принципом кодирования. Тогда взаимно однозначные

соответствия двоичного принципа кодирования представятся записью: **0** ↔ 0, **1** ↔ 1, **2** ↔ 2, **3** ↔ 3 , **4** ↔ 4, **5** ↔ 5, **6** ↔ 6, **7** ↔ 7, а другой вариант кодирования будет представлен, например, иной записью соответствия: **0** ↔ 0, **1** ↔ 2, **2** ↔ 1, **3** ↔ 3 , **4** ↔ 4, **5** ↔ 5, **6** ↔ 6, **7** ↔ 7 и т.д.

Поставив на первое место в ряду кодов двоичный код, представляет определенный интерес определить место в этом ряду каждому коду. Для этого необходимо принять определенное правило для формирования остальных типов кодов.

Для основания n=2 все просто: число различных перестановок равно двум (0 1) и (1 0). Первая перестановка определяет двоичный код, вторая – обратный ему код. В матричной форме записи это будет выглядеть следующим образом:

$$\begin{vmatrix} (01) \\ (10) \end{vmatrix}_{n=2} . \qquad (1.2.1)$$

Для основания n=3 число различных перестановок представляется также весьма просто матрицей размерами 3 × 2

$$\begin{vmatrix} (012) & (021) \\ (102) & (120) \\ (210) & (201) \end{vmatrix}_{n=3} . \qquad (1.2.2)$$

В (1.2.2) содержится шесть перестановок, а дальнейшее представление принципов кодирования оснований n = 4, 5, 6, ... может быть сформулировано следующим правилом.

Первый столбец матрицы перестановок i-го основания системы счисления образуется из элементов первого столбца матрицы перестановок (i – 1)-го основания системы счисления, где в каждой перестановке добавляется цифра (i – 1), а последний i элемент первого столбца матрицы перестановок i-го основания состоит из цифр в последовательности (i – 1) (i – 2) ... 0.

В матрице перестановок i-го основания число столбцов совпадает с числом элементов матрицы перестановок (i – 1)-го основания, где в матрице i-го основания каждый первый элемент столбца содержит первую цифру 0, а остальные цифры этого элемента равны цифрам элементов матрицы перестановок (i – 1)-го основания при их прохождении сверху вниз и слева направо, увеличенным на единицу.

Все последующие цифры элементов в каждом из столбцов задаются порядком расположения цифр первого столбца матрицы перестановок искомого i-го основания.

Тогда следующая матрица размерами 4 × 6 определяет все коды основания n = 4:

$$\begin{vmatrix} (0123) & (0213) & (0321) & (0132) & (0231) & (0312) \\ (1023) & (1203) & (1320) & (1032) & (1230) & (1302) \\ (2103) & (2013) & (2301) & (2130) & (2031) & (2310) \\ (3210) & (3120) & (3012) & (3201) & (3102) & (3021) \end{vmatrix} \quad n = 4 \qquad (1.2.3)$$

и т.д.

Для неизбыточных кодов, где основание системы счисления кратно двум $n = 2^K$, матрицы перестановок непосредственно определяют все типы возможных кодов. Если первая перестановка и ее цифровые эквиваленты от **0** до ($\mathbf{2^K - 1}$) соответствуют кодовым комбинациям двоичного принципа кодирования, то все остальные перестановки этих цифр дают только порядок расположения этих кодовых комбинаций в других принципах кодирования, а эквивалентные им сигналы ОЦК должны также располагаться в последовательности от **0** до ($\mathbf{2^K - 1}$).

Принимая для любых типов кодов неизменность кодового слова для цифры **0**, равенство нулю всех сигналов кода, число кодов для основания n уменьшается до значения $(n - 1)!$, а с учетом необходимости иметь каждому коду его обратный код число кодов равно $2[(n - 1)!]$. Так, для основания n = 4 число прямых кодов будет определяться первой строкой матрицы (1.2.3).

Кроме понятия «нулевых» кодов введем понятие «дружественных» кодов, в которых, дополнительно к постоянству кодовой комбинации сигналов цифры **0**, устанавливается такое же постоянство комбинации сигналов для цифры ($\mathbf{n - 1}$) – равенство их всех логической единице. Число «дружественных» кодов равно $(n - 2)!$.

Число эквивалентных кодов S_K, т.е. кодов одинаковой структуры, определяет число классов неизбыточного кода мерности K. Очевидно, что число классов неизбыточного кода мерности K равно $n!/ S_K$.

Число «дружественных» кодов каждого класса определяется числом перестановок K аргументов кода, в образовании которых будем также придерживаться представленного выше порядка.

Для избыточных кодов $n < 2^K$ их общее число кодов определяется зависимостью $C^n_{2K} \times (n!)$, а с учетом равенства нулю всех сигналов кода, эквивалентных цифре **0** основания системы счисления, число кодов уменьшается до величины $2C^{(n-1)}_{(2K-1)} \times [(n - 1)!]$. Число избыточных кодов огромно и не поддается счету.

Поэтому остановим пока внимание только на неизбыточных кодах. В этом случае каждый код основания n имеет свой порядковый номер в последовательности перестановок $П_{(n-1)}$. Все двоичные коды при таком подходе имеют нулевой номер для любого основания системы счисления.

В этих кодах любая цифра основания позиционной системы счисления определяется как конъюнкция всех K аргументов кода. Поэтому знание числа конъюнкций определенного ранга и общего числа конъюнкций всех рангов, а также определенного порядка их образования требует уделить этому вопросу некоторое внимание.

Определение конъюнкций первого ранга тривиально – в качестве конъюнкций здесь выступают сами аргументы x_i и их инверсии $\underline{x}_i$. При K аргументах число таких конъюнкций 2K. Это значение конъюнкций может быть представлено так же, как число сочетаний из 2K по 1, т.е. $C^1_{2K} = 2K$.

Для нахождения конъюнкций второго ранга можно определить число сочетаний из 2K по 2 и в этом числе сочетаний вычеркнуть такие, где содержатся произведения $x_i\,\underline{x}_i$. Таким же образом можно определить все конъюнкции j-го ранга, находя все сочетания из 2K по j и убирая нулевые произведения. Этот способ определения конъюнкций не продуктивен, не имеет определенного порядка их образования, не дает их общего числа и каждого числа конъюнкций определенного ранга.

Изложение предложенной процедуры определения числа конъюнкций начнем на примере шести аргументов ($x_1, x_2, \dots, x_6$) кода, который затем распространим на их любое число.

Матричная форма записи числа конъюнкций второго ранга, где элементарная матрица содержит четыре конъюнкции второго ранга

$$\left|\begin{matrix} x_i x_j & x_i \underline{x}_j \\ \underline{x}_i x_j & \underline{x}_i \underline{x}_j \end{matrix}\right|_{a_{ij}},$$

позволяет конъюнкции второго ранга дать в табличном представлении следующим образом:

$$\begin{array}{c|cccccc|} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ x_2 & & & a_{23} & a_{24} & a_{25} & a_{26} \\ x_3 & & & & a_{34} & a_{35} & a_{36} \\ x_4 & & & & & a_{45} & a_{46} \\ x_5 & & & & & & a_{56} \\ x_6 & & & & & & \\ \hline \end{array}\,. \qquad a_{ij} \qquad (1.2.4)$$

Из этой записи следует, что число конъюнкций второго ранга здесь равно 60. В самом деле, каждая элементарная матрица содержит четыре конъюнкции второго ранга, например:

$$\left| a_{14} \right| = \left|\begin{matrix} x_1 x_4 & x_1 \underline{x}_4 \\ \underline{x}_1 x_4 & \underline{x}_1 \underline{x}_4 \end{matrix}\right|,$$

а число таких элементарных матриц равно 15.

Элементарная матрица для конъюнкций третьего ранга a_{ijs}, которая состоит из восьми конъюнкций третьего ранга, записывается через элементарные матрицы второго порядка:

$$\left| \begin{matrix} x_i\, x_j\, x_s & x_i\, \underline{x}_j\, x_s & x_i\, x_j \underline{x}_s & x_i\, \underline{x}_j\, \underline{x}_s \\ \underline{x}_i\, x_j\, x_s & \underline{x}_i\, \underline{x}_j\, x_s & \underline{x}_i\, x_j\, \underline{x}_s & \underline{x}_i\, \underline{x}_j\, \underline{x}_s \end{matrix} \right|_{a_{ijs}}$$

или

$$\left| a_{ij}\, x_s \quad a_{ij}\, \underline{x}_s \right|_{a_{ijs}} . \tag{1.2.5}$$

Тогда конъюнкции третьего ранга в координатах прямых и инверсных сигналов аргументов второго ранга имеют вид

$$\begin{array}{c} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{23} \\ a_{24} \\ a_{25} \\ a_{34} \\ a_{35} \\ a_{45} \end{array} \left| \begin{array}{cccc} x_3 & x_4 & x_5 & x_6 \\ a_{123} & a_{124} & a_{125} & a_{126} \\ & a_{134} & a_{135} & a_{136} \\ & & a_{145} & a_{146} \\ & & & a_{156} \\ & a_{234} & a_{235} & a_{236} \\ & & a_{245} & a_{246} \\ & & & a_{256} \\ & & a_{345} & a_{346} \\ & & & a_{356} \\ & & & a_{456} \end{array} \right|_{a_{ijs}} . \tag{1.2.6}$$

Элементарная матрица конъюнкций четвертого ранга a_{ijst}, которая состоит из шестнадцати конъюнкций четвертого ранга, равна

$$\left| a_{ijs}\, x_t \quad a_{ijs}\, \underline{x}_t \right|_{a_{ijst}} . \tag{1.2.7}$$

Конъюнкции четвертого ранга в координатах прямых и инверсных сигналов аргументов и конъюнкций третьего ранга записываются следующим образом:

$$
\begin{array}{c|ccc|l}
 & x_4 & x_5 & x_6 & \\
a_{123} & a_{1234} & a_{1235} & a_{1236} & \\
a_{124} & & a_{1245} & a_{1246} & \\
a_{125} & & & a_{1256} & \\
a_{134} & & a_{1345} & a_{1346} & \\
a_{135} & & & a_{1356} & \\
a_{145} & & & a_{1456} & \\
a_{234} & & a_{2345} & a_{2346} & \\
a_{235} & & & a_{2356} & \\
a_{245} & & & a_{2456} & \\
a_{345} & & & a_{3456} & a_{ijst}
\end{array}
\quad . \qquad (1.2.8)
$$

Элементарная матрица конъюнкций пятого ранга состоит из тридцати двух конъюнкций пятого ранга

$$
\left| \begin{array}{cc} a_{ijst}\, x_l & a_{ijst}\underline{x}_l \end{array} \right|_{a_{ijstl}} . \qquad (1.2.9)
$$

Конъюнкции пятого ранга в координатах прямых и инверсных сигналов аргументов и конъюнкций четвертого ранга представляются следующим образом:

$$
\begin{array}{c|cc|l}
 & x_5 & x_6 & \\
a_{1234} & a_{12345} & a_{12346} & \\
a_{1235} & & a_{12356} & \\
a_{1245} & & a_{12456} & \\
a_{1345} & & a_{13456} & \\
a_{2345} & & a_{23456} & \\
 & & & a_{ijstl}
\end{array}
\quad . \qquad (1.2.10)
$$

Элементарная матрица конъюнкций шестого ранга состоит из шестидесяти четырех конъюнкций шестого ранга и равна

$$
\left| \begin{array}{cc} a_{12345}x_6 & a_{12345}\underline{x}_6 \end{array} \right|_{a_{123456}} . \qquad (1.2.11)
$$

Приведенные выражения для шести аргументов позволяют представить общую зависимость числа конъюнкций любого ранга при неограниченном числе аргументов К.

Число сочетаний из 2К по j , где j = 1, 2, …, К, определяется следующими простыми выражениями:

$C^1_{2K} = 2K = 2C^1_K$,

$C^2_{2K}\,(x_i\,\underline{x}_i=0) = 2^2\,[(K-1) + (K-2) + \ldots + 1] = 2^2\,C^2_K$,

$C^3_{2K}\,(x_i\,\underline{x}_i=0) = 2^3\,[\{(K-2) + \ldots + 1\} + \{(K-3) + \ldots + 1\} + \ldots + 1] = 2^3 C^3_K$,

$C^4_{2K}\,(x_i\,\underline{x}_i=0) = 2^4\,[\{(K-3) + \ldots + 1\} + \{(K-4) + \ldots + 1\} + \ldots + 1] = 2^4 C^4_K$,

…

$$C^K_{2K}\,(x_i\,\underline{x}_i=0) = 2^K C^K_K \quad . \qquad (1.2.12)$$

Поскольку конъюнкцию любого ранга можно записать как произведение K сомножителей, каждый из которых может принимать значение x, $\underline{x}$ либо отсутствовать, то общее число конъюнкций всех рангов, включая и нулевой ранг ($C^0_{2K} = 0$), равно 3^K. В этом легко убедиться непосредственно из (1.2.12) для конкретного значения K.

1.3. Основные логические операции над подмножествами многомерного цифрового пространства

Создатель основ теории множеств выдающийся немецкий математик Г. Кантор рассматривал множество как объединение в одно целое объектов, различимых нашей интуицией или мыслью. Понятие «множество» относится к неопределенным понятиям науки, но в нашем случае под множеством будем понимать, как отмечено выше, числа натурального ряда от **0** до ($\mathbf{2^n - 1}$), представленные в любой позиционной системе счисления. Очевидно, что все известные правила действия над множествами [14] полностью распространяются на эти цифровые множества. Поэтому здесь приведем только те из них, которые определяются нашими ограничениями.

Принадлежность цифры i множеству **A** обозначается $i \in \mathbf{A}$ (i принадлежит **A**); непринадлежность обозначается $i \notin \mathbf{A}$ (i не принадлежит **A**).

Если $\mathbf{A^i}$ (i – разряд числа **A**), то запись $\mathbf{A} = \{\mathbf{A^1}, \ldots, \mathbf{A^i}\}$ означает разряды $\mathbf{A^1}$, $\mathbf{A^2}$, ... суть элементы множества (числа) **A** .

Запись $\mathbf{A^i} = \{\mathbf{O^i}_a, \mathbf{1^i}_a, \ldots, \mathbf{(n-1)^i}_a\} = \{\mathbf{O}, \mathbf{1}, \ldots, \mathbf{(n-1)}\}^i_a$ означает цифры разряда i (0, 1, ... , (n – 1)), где n – основание системы счисления, суть элементы подмножества (разряда) $\mathbf{A^i}$.

Одноэлементное множество – множество, состоящее из одного элемента, например $\{\mathbf{O^i}_a\}$ или $\{\mathbf{A^i}\}$, где в первом случае элементом является цифра разряда $\mathbf{A^i}$, а во втором случае – это i-й разряд числа **A** (иерархия элементов, подмножеств).

Пустое множество ∅ – множество, не содержащее элементов (цифр) вообще. Множества **A** и ∅ называются несобственными подмножествами множества **A**, а все остальные подмножества **A** – его собственными подмножествами. Например, множества натуральных чисел являются подмножеством всех це-

лых чисел, а последние, в свою очередь, – подмножеством множества всех рациональных чисел.

Известно [14], что «множества, встречающиеся в рассуждениях той или иной математической теории, являются подмножествами некоторого фиксированного множества E, называемого универсальным для данной теории». Универсальным для нашего случая является множество, где во всех ячейках многомерного цифрового пространства расположены логические единицы 1*.

Множества **A** и **B** называются равными, **A** = **B**, если они состоят из одних и тех же элементов на всех ступенях иерархии.

Если все элементы множества **A** являются также элементами множества **B**, то **A** – подмножество **B**, или **A** включается в **B**, что обозначается как $\mathbf{A} \subseteq \mathbf{B}$. Если $\mathbf{A} \subseteq \mathbf{B}$, причем $\mathbf{A} \neq \mathbf{B}$, то записывается $\mathbf{A} \subset \mathbf{B}$. Знак $\subset$ в, отличие от знака $\subseteq$, называется знаком строгого включения.

Для любого множества **A** выполняется включение $\varnothing \subset \mathbf{A}$, $\mathbf{A} \subset \mathbf{A}$ (рефлексность).

Множества **A** и $\varnothing$ называются несобственными подмножествами множества **A**, а все остальные подмножества **A** – его собственные подмножества.

Если $\mathbf{A} \subseteq \mathbf{B}$ и $\mathbf{B} \subseteq \mathbf{C}$, то $\mathbf{A} \subseteq \mathbf{C}$ (транзитивность).

Если в качестве элементов множества используются другие множества, например $\mathbf{A} = \{\mathbf{A}^1, \mathbf{A}^2, ..., \mathbf{A}^i\}$, то это семейство множеств $\{\mathbf{A}^i \mid i \in \mathbf{I}\}$, где $\mathbf{A}^i$ – некоторое множество, например, i-го разряда натурального числа **A**, а **I** – множество индексов, которое может быть конечным или бесконечным.

Очевидно, что множества, состоящие из конечного числа элементов, называются конечными; в противном случае говорят о бесконечных множествах. Бесконечным множеством является множеств цифр расширенного натурального ряда, а множество цифр конкретного основания **n** позиционной системы счисления является конечным.

Говорят, что «элементы множеств **A** и **B** находятся во взаимно однозначном соответствии, если каждому элементу **a** множества **A** сопоставлен по некоторому закону единственный элемент в множестве **B**, причем каждый элемент $\mathbf{b} \in \mathbf{B}$ оказывается сопоставленным одному и только одному элементу $\mathbf{a} \in \mathbf{A}$». Взаимно однозначное соответствие элементов множеств **A** и **B** обозначается как $\mathbf{a} \leftrightarrow \mathbf{b}$.

Множества **A** и **B** называются эквивалентными или равномощными ($\mathbf{A} \sim \mathbf{B}$), если можно установить взаимно однозначное соответствие их элементов. Это определение распространяется на конечные и бесконечные множества. Очевидно, что конечные множества **A** и **B** эквивалентны лишь тогда, когда они содержат одинаковое число элементов. Причем если $\mathbf{A} \sim \mathbf{B}'$, где $\mathbf{B}' \subset \mathbf{B}$, то множество **A** содержит меньше элементов, чем множество **B**. Таким образом, понятие числа элементов множества эквивалентно понятию мощности множества.

Общеизвестно, что непротиворечивость любой теории заключается в том, что в ней не может быть получено противоречие, т.е. не может быть доказано некоторое утверждение **A** и его отрицание $\underline{\mathbf{A}}$.

Выдающийся математик Д. Гильберт полагал, что «парадоксы возникают тогда, когда пользуются недопустимыми и бессмысленными образованиями понятий». При этом необходимо различать *действительные* и *идеальные* предложения классической т.е. «чистой математики». Первые должны включать содержательный смысл, а вторые могут его и не включать.

Например, предложения, употребляющие понятия бесконечности, идеальны, но их соединение с действительными понятиями применимы и к рассуждениям о бесконечных множествах.

Для бесконечных множеств можно также задать вопрос: нельзя ли «бесконечность» одного множества считать большей, равной или меньшей, чем «бесконечность» другого множества? Здесь ответ заключается в понятии одинаковой мощности этих множеств, если между их элементами можно установить взаимно однозначное соответствие, т.е. если одно множество можно также отобразить на другое, что каждому элементу первого взаимно однозначно соответствует некоторый элемент второго. Если же подобное отображение неосуществимо, то множества имеют различную мощность; при этом оказывается, что в последнем случае, как бы мы ни пытались привести в соответствие элементы обоих множеств, всегда останутся лишние элементы и притом постоянно от одного и того же множества, которое имеет поэтому «большую мощность».

Бесконечное множество чисел натурального ряда N_0 и бесконечное множество ячеек многомерного цифрового пространства при любом числе мерности этого пространства имеют одинаковую мощность и, следовательно, по определению эквивалентны.

Фиксированное множество **E**, определяющее все многомерное цифровое пространство, где множества **A**, **B**,... являются его подмножествами, как уже отмечалось выше, называется универсальным.

Каждому подмножеству $\mathbf{A} \subseteq \mathbf{E}$ сопоставим функцию y_a такую, что $y_a = 1^*$, если подмножество **A** не пустое $(\mathbf{A} \neq \varnothing)$, и $\mathbf{y_a} = 0^*$, если $\mathbf{A} = \varnothing$. Тем самым достигается полное соответствие: между операцией объединения $\cup$ и логическим сложением $\vee$, между операцией пересечения $\cap$ и логическим умножением &, между операцией дополнения $\mathbf{A_E}$ и логической операцией отрицания $\underline{\mathbf{A}}$.

Для перехода от конкретной фигуры цифрового множества к логической функции, по которой строится принципиальная схема логического блока, необходимо дать определение покрытия цифрового множества: покрытием данного цифрового множества называется конечная совокупность его подмножеств, дающая в своей сумме всё это цифровое множество.

Классической постановкой задачи определения покрытия «длины», «площади», «объема» цифровых множеств является поиск нормальной минимальной дизъюнктивной формы (ДНФ), когда лучшей считается логическая схема, соответствующая этой ДНФ и занимающая меньшую площадь кристалла большой интегральной схемы (БИС). Представленный критерий оптимизации в тесном сочетании с технологическими требованиями регулярности структуры БИС привел к отображению на плоскости либо объеме кристалла ДНФ ло-

гической функции или системы функций в виде матричных схем, которые получили название программируемых логических матриц (ПЛМ), а также других микросхем программируемой матричной логики (ПМЛ).

Предлагаемая теория многомерных цифровых множеств является инструментом, наилучшим образом ориентированным на автоматизацию синтеза таких БИС, когда конструкция каждой из них имеет не только аналитическую запись в ДНФ, но и физический геометрический образ цифрового множества многомерного цифрового пространства.

Это есть стремление к наглядности, которая, по мнению Д. Гилберта, «стремится к живому пониманию объектов и их внутренних соотношений».

Именно наглядность играет в теории многомерных цифровых множеств первостепенную роль как обладающая большой доказательной силой для понимания и оценки результатов автоматизированного синтеза ЛФ, что будет доказано ниже.

Особое значение здесь придается изучению симметрии цифрового пространства и его подмножеств, как свойству содержать в себе равные и однообразно расположенные части, где законы преобразования фигур в пространстве являются следствием симметрии законов самого многомерного цифрового пространства, обладающего таким же богатством, как окружающий нас мир.

1.4. Бинарные логические операции в двухмерном цифровом пространстве

Общепринятое аналитическое рассмотрение бинарных логических операций, как правило, начинается с представления ЛФ двух аргументов x_1,x_2, что представлено табл. 1.4.1.

Таблица 1.4.1

оцк	x_2	x_1	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}	y_{16}
0	0^*	0^*	0^*	1^*	0^*	1^*	0^*	1^*	0^*	1^*	0^*	1^*	0^*	1^*	0^*	1^*	0^*	1^*
1	0^*	1^*	0^*	0^*	1^*	1^*	0^*	0^*	1^*	1^*	0^*	0^*	1^*	1^*	0^*	0^*	1^*	1^*
2	1^*	0^*	0^*	0^*	0^*	0^*	1^*	1^*	1^*	1^*	0^*	0^*	0^*	0^*	1^*	1^*	1^*	1^*
3	1^*	1^*	0^*	0^*	0^*	0^*	0^*	0^*	0^*	0^*	1^*	1^*	1^*	1^*	1^*	1^*	1^*	1^*

Не будем перечислять известные из учебной литературы названия и обозначения логических операций, представленных в этой таблице.

Поскольку концепция эквивалентности вариантов покрытия «длины», «площади», «объема» лежит в основе конкретной реализации логического блока и его геометрического образа, обратимся непосредственно к изучению этих фигур.

На рис. 1.5 приведены геометрические образы всех шестнадцати бинарных ЛФ в двухмерном пространстве $\mathbf{E_2}$, которое имеет пять осей симметрии *1–5*. Допуская, что входы и выходы логических блоков имеют шины прямых и инверсных сигналов, число рассматриваемых фигур в пространстве любой кратности значительно сокращается. Для нашего случая $y_1 = \underline{y}_{16}$, $y_2 = \underline{y}_{15}$, $y_3 = \underline{y}_{14}$, $y_9 = \underline{y}_8$, $y_5 = \underline{y}_{12}$, $y_4 = \underline{y}_{13}$, $y_{11} = \underline{y}_6$, $y_{10} = \underline{y}_7$.

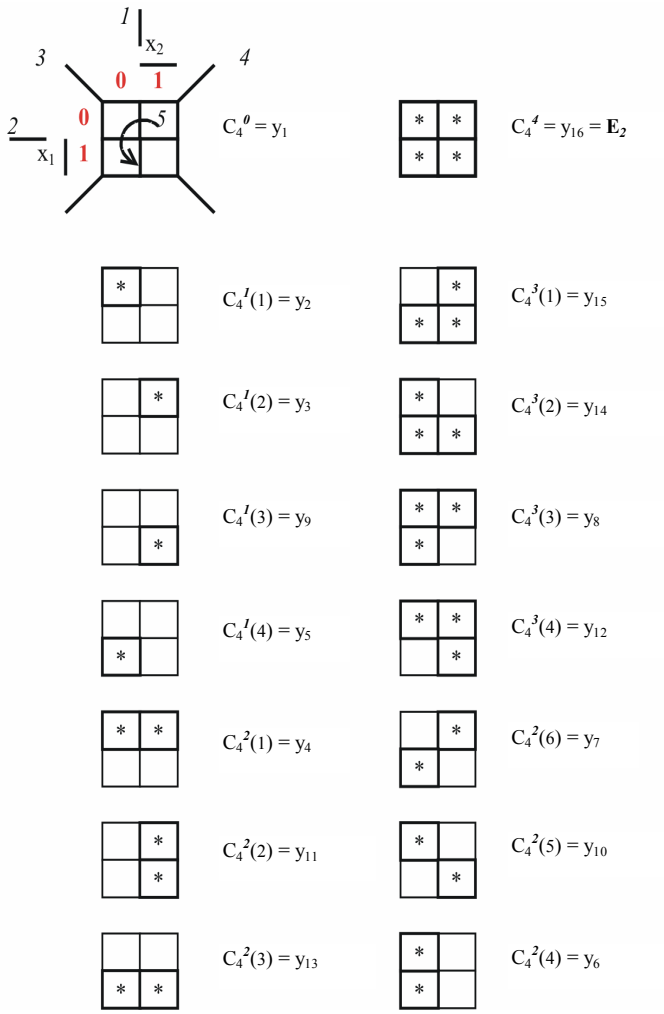


Рис. 1.5

Дальнейшее сокращение количества рассматриваемых фигур связано с симметрией пространства $\mathbf{E_2}$.

Мысленный поворот вокруг оси *1* приводит к записи аргумента x_1 ЛФ в обратном коде $x_1 \to \underline{x}_1$; поворот вокруг оси *2* – к записи аргумента x_2 в исходной ЛФ в обратном коде $x_2 \to \underline{x}_2$; поворот вокруг оси *3* приводит к смене местами аргументов $(x_1, x_2) \to (x_2, x_1)$; поворот вокруг оси *4* приводит к смене местами аргументов, а также записи их в обратном коде $(x_1, x_2) \to (\underline{x}_2, \underline{x}_1)$; каждый последовательный мысленный поворот вокруг оси *5* на 90° против часовой стрелки приводит к смене местами аргументов с одновременной записью первого из них в обратном коде, что иллюстрируется следующим образом: $\to (x_1, x_2) \to \to (x_2, \underline{x}_1) \to (\underline{x}_1, \underline{x}_2) \to (\underline{x}_2, x_1) \to (x_1, x_2)$, а при вращении в противоположном направлении – к смене аргументов в обратном направлении.

Исходя из симметрии двухмерного цифрового пространства и убрав из рассмотрения тривиальные функции $y_1 = \varnothing = 0^*$, $y_{16} = \mathbf{E_2} = 1^*$, конструкцию любо-

го логического блока двух аргументов определяем только одной из первообразных фигур y_2 , y_4 , y_{10} , а остальные фигуры получаются соответствующими мысленными поворотами вокруг осей симметрии.

Условимся в качестве первообразных геометрических образов выбирать такую фигуру, которая включает ячейку пространства под номером цифры **0**.

Из фигур, получаемых сочетанием из четырех по одному, это функция $y_2 = C^1_4(1)$, а остальным, которые получаются соответствующим мысленным поворотом, присвоим следующие порядковые номера: $y_3 = C^1_4(2)$, $y_9 = C^1_4(3)$, $y_5 = C^1_4(4)$.

Из фигур, получаемых сочетанием из четырех по два, первая первообразная – $y_4 = C^2_4(1)$, а остальные будут иметь следующие обозначения: $y_{11} = C^2_4(2)$, $y_{13} = C^2_4(3)$, $y_6 = C^2_4(4)$; вторая первообразная – сочетания из четырех по два – $y_{10} = C^2_4(5)$, а ее образующая – $y_4 = C^2_4(1)$.

Фигурам, получаемым сочетанием из четырех по три, присвоим порядковые номера их дополнений: $y_{15} = C^3_4(1)$, $y_{14} = C^3_4(2)$, $y_8 = C^3_4(3)$, $y_{12} = C^3_4(4)$.

1.5. Сложные логические функции в двухмерном цифровом пространстве

При рассмотрении арифметических и логических операций нами используется финитная точка зрения, которая совершается «в виде мысленных экспериментов над наглядно представляемыми объектами и не зависит от предложений аксиоматического характера. Рассуждения такого рода называются финитными, а методическая установка, лежащая в основе таких рассуждений, называется финитной установкой или финитной точкой зрения».

Известно, что любые арифметические операции совершаются над конкретными значениями цифровых данных, которые обычно называются операндами. Для связи логических функций с арифметическими операциями введем понятие сложной логической функции, под ней будем понимать логическую функцию, аргументами которой выступают операнды.

Пусть логическая функция состоит из двух аргументов **A**, **B**, которые, в свою очередь, эквивалентны цифрам натурального ряда **0**, **1**, ... , (**n – 1**) и закодированы любым известным способом.

Прямой порядок следования цифр натурального ряда может быть представлен вектором, в начале которого расположен «элементарный кубик», нумерованный цифрой **0**, а в конце вектора – «элементарный кубик» под номером (**n –1**). Исходя из векторного представления аргументов, на рис. 1.6 графически показаны преобразования аргументов **A**, **B** логических функций при соответствующих мысленных поворотах вокруг осей симметрии *1–5*. В них исходные положения векторов аргументов записываются в виде **F (A**, **B)**, а при соответствующем повороте изменяется порядок их следования либо (и) происходит преобразование кода аргументов из прямого в обратный, когда новое положение вектора противоположно исходному вектору в функции **F (A**, **B)**.

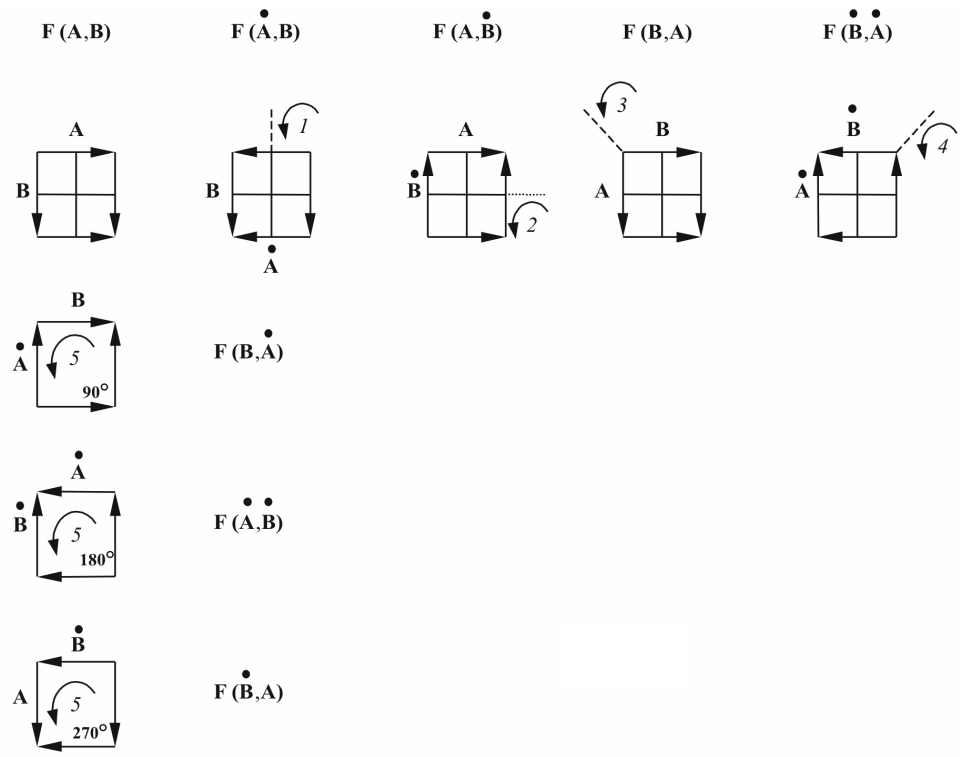


Рис. 1.6

Мысленный поворот вокруг оси *1* производит перевод аргумента **A** из прямого кода в обратный, когда новое положение фигуры логической функции и ее покрытие будут определяться первообразной логической функцией, где аргумент **A** взят в обратном коде, что запишется следующим образом: **F (A•, B)**.

На рисунке, соответствующем этому повороту вокруг оси *1*, новое положение вектора аргумента **A** противоположно исходному положению этого же вектора и обозначено **A•**, при этом вектор **B** сохраняет свое первоначальное направление.

Еще один пример мысленного поворота относительно оси *5*, из которого станут более понятны все преобразования рис. 1.6.

При мысленном повороте вокруг оси *5* против часовой стрелки новое положение будет представлено следующим образом: вектор аргумента **A** расположится на месте исходного вектора **B** в противоположном ему направлении, а новое положение вектора **B** – на месте исходного вектора **A** и совпадает с ним по направлению. Поэтому новое положение фигуры логической функции, а также ее покрытие будут определяться первообразной логической функцией, где аргументы поменяны местами и аргумент **A** взят в обратном коде, т.е. **F (B ,A•)** и т.д.

При симметрии фигуры относительно данной оси любой мысленный поворот относительно этой оси не изменяет фигуру и логическую функцию, определяющую ее покрытие и конструкцию блока. Если фигура асимметрична относительно данной оси, то при соответствующем мысленном повороте вокруг нее она займет новое положение в координатах аргументов **A**, **B**. Принципиальная же схема логического блока, реализующая покрытие этой новой фигуры, совпадает с прежней схемой, где на входе произведены соответствующие перестановки аргументов, а также перевод их из прямого кода в обратный.

Мысленные повороты относительно осей *1–4* – это простые повороты, а поворот относительно оси *5* – сложный поворот, который может быть осуществлен двумя последовательными поворотами вокруг осей *1 – 4*.

На рис. 1.7 представлены все возможные положения аргументов **A**, **B**, которые достигаются соответствующими мысленными поворотами относительно осей симметрии пространства.

Следует отметить, что использование в логических функциях аргументов с циклическим принципом кодирования, когда каждая кодовая комбинация, эквивалентная цифре натурального ряда **0**, **1**, ... , (**n – 1**), получается циклической перестановкой их составляющих сигналов, расширяет зону применения первообразной функции. При мысленном перемещении фигуры в многомерном цифровом пространстве вдоль оси с циклическим принципом кодирования сложного аргумента принципиальная схема логического блока, реализующая покрытие этой новой фигуры, будет совпадать с первообразной фигурой. В этой принципиальной исходной схеме необходимо будет произвести в соответствующих аргументах определенные циклические перестановки сигналов.

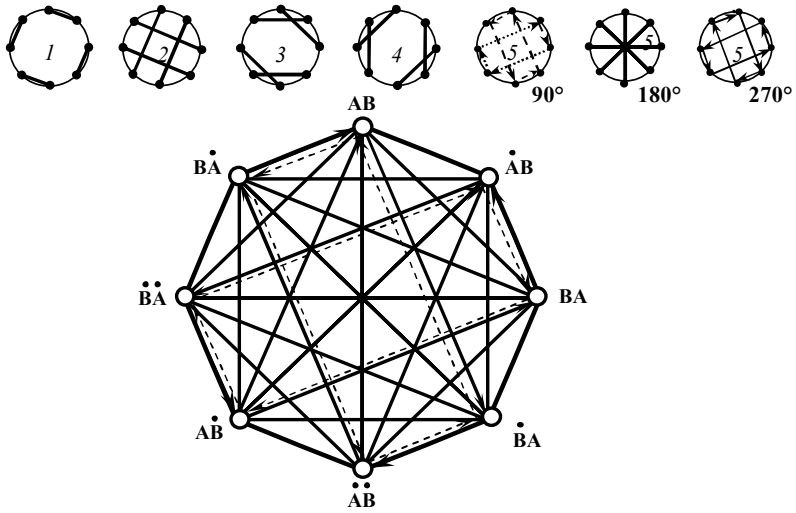


Рис. 1.7

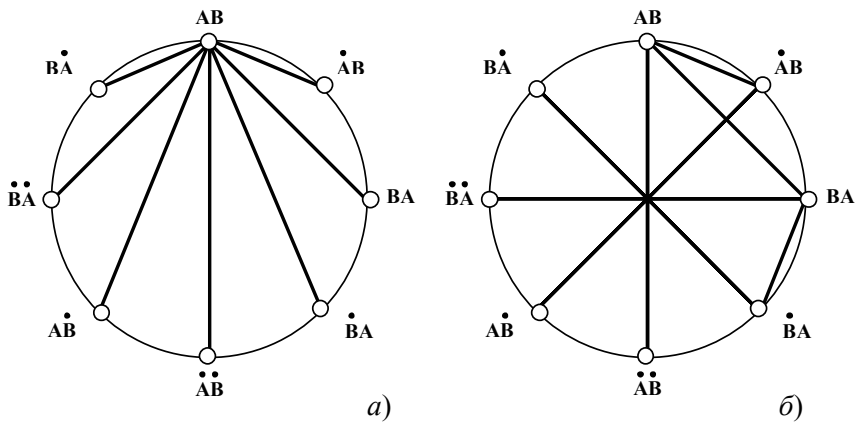


Рис. 1.8

К числу таких циклических кодов относятся обычный цифровой код, который составляет основу многозначной логики, а также многофазные коды, лежащие в основе большинства устройств электротехники. Многофазные коды относятся также к числу так называемых геометрических кодов, где под термином «геометрический код» понимается то, что эквивалентная цифра натурального ряда в этих типах кодов может определяться не только соответствующей конъюнкцией максимального ранга, но и взаимным геометрическим положением их сигналов. К слову сказать, алгебра Буля не применима для синтеза и анализа устройств, использующих геометрические коды.

Для выполнения всех приведенных выше мысленных поворотов и их аналитического представления наиболее удобно ограничиться последовательным использованием только поворотов относительно осей *1–3*. Поворот относительно осей *1* и *2* приводит к переводу соответствующего вектора (аргумента) из прямого кода в обратный и наоборот. Аналитически это записывается следующим образом: $\mathbf{A}^{\circlearrowleft}$ либо $\mathbf{B}^{\circlearrowleft}$, а поворот вокруг оси *3*, который соответствует смене местами векторов, запишется как $(\mathbf{AB})^{\circlearrowleft}$.

На рис. 1.8, *а* показаны все возможные мысленные положения аргументов **A** и **B**, которые реализуются соответствующим мысленным поворотом относительно осей *1–5*. В отличие от предыдущего (см. рис. 1.7), здесь показаны только переходы из одного мысленного положения системы координат в другие, а на рис. 1.8, *б* – то же самое, но с использованием таких поворотов относительно только осей *1–3*.

Для большей прозрачности и наглядности преобразований геометрического образа логической функции в двухмерном цифровом пространстве при мысленных поворотах этого образа относительно осей *1–5* представим, например, аргументы **A** и **B** в системе счисления основания n = 4. Тогда исходная нумерация ячеек двухмерного цифрового пространства, которая всегда остаётся неизменной, будет совпадать с нумерацией кодовых слов, определяющих каждое еди-

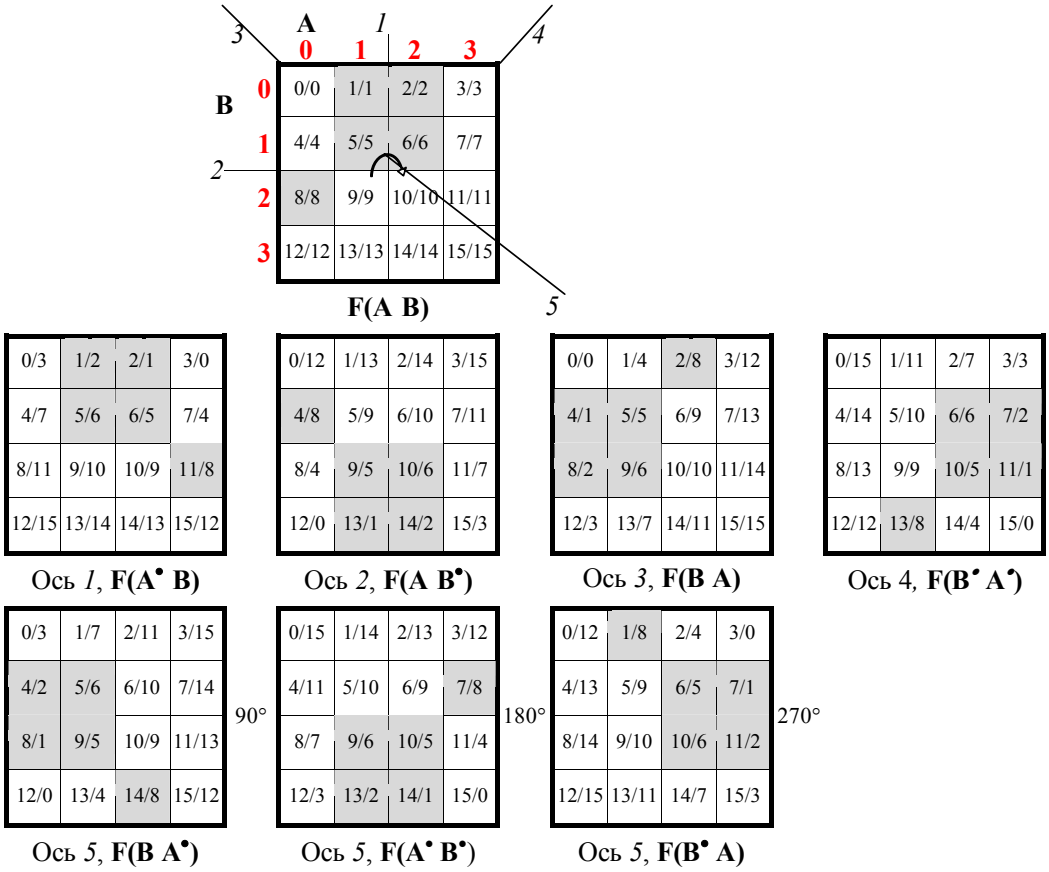


Рис. 1.9а

ничное множество геометрического образа логической функции. Это положение подтверждено рис. 1.9а, где исходная логическая функция занимает место в ячейках пространства, например, с номерами 1, 2, 5, 6, 8. Следовательно, в этих ячейках цифрового пространства должны быть помещены, как мы условились выше, знаки логической функции 1*. Пусть, например, цифры натурального ряда **0 – 3** представлены двумя вариантами кодирования: в двоичном a_1, a_2 и двухфазном a_1, a_2 кодах. На рис. 1.9б показаны соотношения между сигналами для прямого **A** и обратного **A•** следования цифр этого операнда.

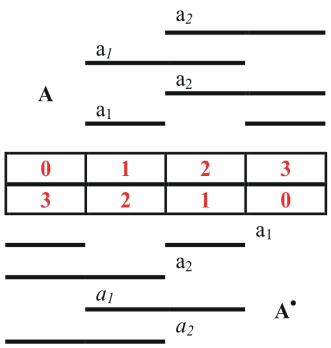


Рис. 1.9б

Прямое (**0 – 3**) и обратное (**3 – 0**) следование цифр операнда **A** определяет взаимно однозначные соотношения между аргументами этих кодов: для двоичного принципа кодирования это общеизвестные соотношения $a_1 = \underline{a}_1$, $a_2 = \underline{a}_2$; для двухфазного – это еще более простое соотношение: $a_2 = \underline{a}_2$. На данном примере очевиден принцип получения взаимно

однозначных соотношений между аргументами любых типов кодов для любого основания системы счисления при переводе кода из прямого в обратный.

Теперь вернёмся к преобразованиям геометрического образа выбранной нами логической функции при мысленных поворотах этого образа относительно осей симметрии двухмерного цифрового пространства. Мысленными поворотами они называются потому, что само цифровое пространство и его координаты (аргументы) всегда остаются неизменными, а воображаемые перемещения позволяют определить только то место цифрового пространства, куда должен будет переместиться геометрический образ логической функции при этих мысленных поворотах.

Геометрический образ исходной логической функции **F(A B)** размещается в ячейках пространства, где номер ячейки (числитель дроби) всегда совпадает с номером кодового слова (знаменатель дроби), определяющего при выбранном способе кодирования определенную конъюнкцию соответствующего ранга.

При мысленном повороте геометрического образа исходной логической функции относительно какой-либо оси пространства, что происходит путем соответствующего преобразования операндов, такого совпадения уже не будет. Это изменение представлено числителем дроби в соответствующих ячейках пространства.

С учетом этого можно представить соответствующие логические функции при двоичном принципе кодирования следующим образом:

$$
\begin{aligned}
\mathbf{F(A\ B)} &= a_1\,\underline{a_2}\,\underline{b_2} \vee \underline{a_1}\,a_2\,\underline{b_2} \vee \underline{a_1}\,\underline{a_2}\,\underline{b_1}\,b_2,\\
\mathbf{F(A^{\bullet}\ B)} &= \underline{a_1}\,a_2\,\underline{b_2} \vee a_1\,\underline{a_2}\,\underline{b_2} \vee a_1\,a_2\,\underline{b_1}\,b_2,\\
\mathbf{F(A\ B^{\bullet})} &= a_1\,\underline{a_2}\,b_2 \vee \underline{a_1}\,a_2\,b_2 \vee \underline{a_1}\,\underline{a_2}\,b_1\,\underline{b_2},\\
\mathbf{F(A^{\bullet}\ B^{\bullet})} &= \underline{a_1}\,a_2\,b_2 \vee a_1\,\underline{a_2}\,b_2 \vee a_1\,a_2\,b_1\,\underline{b_2},\\
\mathbf{F(B\ A)} &= b_1\,\underline{b_2}\,\underline{a_2} \vee \underline{b_1}\,b_2\,\underline{a_2} \vee \underline{b_1}\,\underline{b_2}\,\underline{a_1}\,a_2,\\
\mathbf{F(B^{\bullet}\ A)} &= \underline{b_1}\,b_2\,\underline{a_2} \vee b_1\,\underline{b_2}\,\underline{a_2} \vee b_1\,b_2\,\underline{a_1}\,a_2,\\
\mathbf{F(B\ A^{\bullet})} &= b_1\,\underline{b_2}\,a_2 \vee \underline{b_1}\,b_2\,a_2 \vee \underline{b_1}\,\underline{b_2}\,a_1\,\underline{a_2},\\
\mathbf{F(B^{\bullet}\ A^{\bullet})} &= \underline{b_1}\,b_2\,a_2 \vee b_1\,\underline{b_2}\,a_2 \vee b_1\,b_2\,a_1\,\underline{a_2}.
\end{aligned}
\qquad (1.5.1)
$$

Аналогично при двухфазном принципе кодирования эти логические функции будут проще:

$$
\begin{aligned}
\mathbf{F(A\ B)} &= a_1\,\underline{b_2} \vee \underline{a_1}\,\underline{a_2}\,b_1\,b_2,\\
\mathbf{F(A^{\bullet}\ B)} &= a_1\,\underline{b_2} \vee \underline{a_1}\,a_2\,b_1\,b_2,\\
\mathbf{F(A\ B^{\bullet})} &= a_1\,b_2 \vee \underline{a_1}\,\underline{a_2}\,b_1\,\underline{b_2},\\
\mathbf{F(A^{\bullet}\ B^{\bullet})} &= a_1\,b_2 \vee \underline{a_1}\,a_2\,b_1\,\underline{b_2},\\
\mathbf{F(B\ A)} &= b_1\,\underline{a_2} \vee \underline{b_1}\,\underline{b_2}\,a_1\,a_2,\\
\mathbf{F(B^{\bullet}\ A)} &= b_1\,\underline{a_2} \vee \underline{b_1}\,b_2\,a_1\,a_2,\\
\mathbf{F(B\ A^{\bullet})} &= b_1\,a_2 \vee \underline{b_1}\,\underline{b_2}\,a_1\,\underline{a_2},\\
\mathbf{F(B^{\bullet}\ A^{\bullet})} &= b_1\,a_2 \vee \underline{b_1}\,b_2\,a_1\,\underline{a_2}.
\end{aligned}
\qquad (1.5.2)
$$

Каждая логическая функция в (1.5.1) , а также в (1.5.2) реализуется одной и той же принципиальной схемой, где на её входных шинах меняется только порядок их соединения с сигналами операндов.

Теперь представим, что в операндах использованы различные принципы кодирования оснований системы счисления: операнд **A** поступает в двухфазном коде, а операнд **B** – в двоичном коде. В этом случае рассматриваемые нами логические функции будут иметь иной вид

$$
\begin{aligned}
\mathbf{F(A\ B)} &= a_1\,\underline{b_2} \vee \underline{a_1}\,\underline{a_2}\,\underline{b_1}\,b_2,\\
\mathbf{F(A^{\bullet}\ B)} &= a_1\,\underline{b_2} \vee \underline{a_1}\,a_2\,\underline{b_1}\,b_2,\\
\mathbf{F(A\ B^{\bullet})} &= a_1\,b_2 \vee \underline{a_1}\,\underline{a_2}\,b_1\,\underline{b_2},\\
\mathbf{F(A^{\bullet}\ B^{\bullet})} &= a_1\,b_2 \vee \underline{a_1}\,a_2\,b_1\,\underline{b_2},\\
\mathbf{F(B\ A)} &= \underline{a_2}\,b_1\,\underline{b_2} \vee \underline{a_2}\,\underline{b_1}\,b_2 \vee a_1\,a_2\,\underline{b_1}\,\underline{b_2},\\
\mathbf{F(B^{\bullet}\ A)} &= \underline{a_2}\,\underline{b_1}\,b_2 \vee \underline{a_2}\,b_1\,\underline{b_2} \vee a_1\,a_2\,b_1\,b_2,\\
\mathbf{F(B\ A^{\bullet})} &= a_2\,b_1\,\underline{b_2} \vee a_2\,\underline{b_1}\,b_2 \vee a_1\,\underline{a_2}\,\underline{b_1}\,\underline{b_2},\\
\mathbf{F(B^{\bullet}\ A^{\bullet})} &= a_2\,\underline{b_1}\,b_2 \vee a_2\,b_1\,\underline{b_2} \vee a_1\,\underline{a_2}\,b_1\,b_2.
\end{aligned}
\qquad (1.5.3)
$$

Из выражений (1.5.3) следует, что при использовании операндов с одинаковым значением основания системы счисления, но разным способом его кодирования число принципиальных схем, реализующих покрытие подобных фигур, будет равно числу операндов.

Простой пример реализации покрытия геометрического образа логической функции в двухмерном цифровом пространстве позволяет сделать следующие выводы:

1) геометрический образ конкретной логической функции может иметь «лучшее покрытие», т.е. минимальные затраты оборудования для определенного принципа кодирования этого основания системы счисления;

2) алгебра Буля не позволяет анализировать и синтезировать логические функции, находящиеся вне правил двоичного принципа кодирования;

3) мысленный поворот геометрического образа логической функции вокруг осей симметрии двумерного цифрового пространства позволяет определить общее число логических функций, которые используют одну принципиальную схему реализации.

1.6. Тернарные логические операции в трехмерном цифровом пространстве

Число тернарных логических операций весьма значительно ($2^8 = 256$) и поэтому они, в отличие от бинарных, в литературе отдельно не рассматриваются. Вместе с тем, используя геометрические аналоги логических функций в трехмерном цифровом пространстве, все геометрические образы этих логических

функций могут быть представлены не 256 фигурами, а только 22. Эти геометрические образы логических функций приведены на рис. 1.10.

На рис. 1.10 показаны только такие представители геометрического образа ЛФ, где отчетливо видны все элементарные кубики.

С учетом того, что входы и выходы логических блоков являются бинарными, можно записать: $C^0_8(1) = \underline{C}^8_8(1)$, $C^1_8(1) = \underline{C}^7_8(1)$, $C^2_8(1) = \underline{C}^6_8(1)$, $C^2_8(2) = \underline{C}^6_8(2)$, $C^2_8(3) = \underline{C}^6_8(3)$, $C^3_8(1) = \underline{C}^5_8(1)$, $C^3_8(2) = \underline{C}^5_8(2)$, $C^3_8(3) = \underline{C}^5_8(3)$.

При этом, если не учитывать пустое и универсальное множества, число этих функций сокращается до 13.

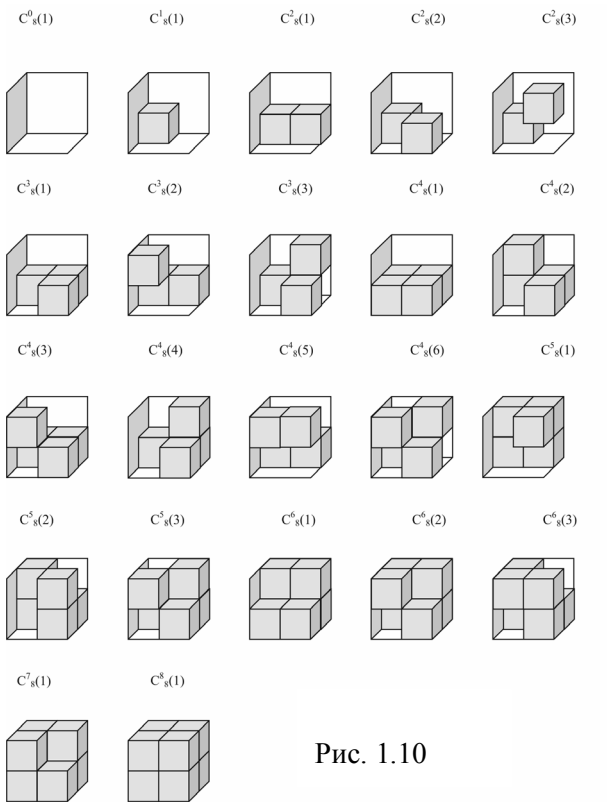


Рис. 1.10

Это следующие функции: $y_1 = C^1_8(1) = \underline{x}_1\,\underline{x}_2\,\underline{x}_3$; $y_2 = C^2_8(1) = \underline{x}_2\,\underline{x}_3$; $y_3 = C^2_8(2) = \underline{x}_1\,\underline{x}_2\,\underline{x}_3 \vee x_1\,x_2\,\underline{x}_3$; $y_4 = C^2_8(3) = \underline{x}_1\,\underline{x}_2\,\underline{x}_3 \vee x_1\,x_2\,x_3$; $y_5 = C^3_8(1) = \underline{x}_2\,\underline{x}_3 \vee x_1\,\underline{x}_3$; $y_6 = C^3_8(2) = \underline{x}_2\,\underline{x}_3 \vee \underline{x}_1\,x_2\,x_3$; $y_7 = C^3_8(3) = \underline{x}_1\,\underline{x}_2\,\underline{x}_3 \vee x_1\,\underline{x}_2\,x_3 \vee x_1\,x_2\,\underline{x}_3$; $y_8 = C^4_8(1) = \underline{x}_3$; $y_9 = C^2_8(2) = \underline{x}_2\,\underline{x}_1 \vee x_1\,\underline{x}_3$; $y_{10} = C^4_8(3) = \underline{x}_2\,\underline{x}_3 \vee x_1\,\underline{x}_3 \vee \underline{x}_1\,x_3\,x_2$; $y_{11} = C^4_8(4) = \underline{x}_2\,\underline{x}_3 \vee x_1\,\underline{x}_3 \vee x_1\,\underline{x}_2$; $y_{12} = C^4_8(5) = \underline{x}_2\,\underline{x}_3 \vee x_2\,x_3$; $y_{13} = C^4_8(6) = \underline{x}_1\,\underline{x}_2\,\underline{x}_3 \vee x_1\,x_2\,\underline{x}_3 \vee x_1\,\underline{x}_2\,x_3 \vee \underline{x}_1\,x_2\,x_3$.

Представленные здесь оптимальные покрытия «объемов» этих 13 фигур определяют также все 254 тернарные логические операции. Правила пользования этими 13 фигурами для определения всех остальных фигур будут понятны после изучения симметрии трехмерного цифрового пространства.

1.7. Сложные логические функции в многомерном цифровом пространстве

В полном соответствии с определениями п. 1.5 рассмотрим изображения сложных логических функций от трех аргументов, определяемых их цифровыми векторами **A**, **B**, **C**, в трехмерном цифровом пространстве. При этом исследуем преобразование фигур логических функций в трехмерном цифровом пространстве, осуществляемое соответствующим мысленным поворотом либо несколькими такими поворотами относительно осей симметрии этого пространства. Из кристаллографии известно 13 осей симметрии кубика, которыми, однако,

не исчерпываются все оси симметрии рассматриваемого цифрового пространства, поскольку в нем возможно еще и «послойное», т.е. двухмерное выполнение соответствующих мысленных поворотов.

Число возможных взаимных положений векторов аргументов можно определить чисто аналитически путем соответствующего перебора всех размещений из шести (**A**, **B**, **C**, **A**$^{\bullet}$, **B**$^{\bullet}$, **C**$^{\bullet}$) по три, из которых необходимо будет исключить такие, где встречаются взаимно исключающие сочетания (**A**, **A**$^{\bullet}$; **B**, **B**$^{\bullet}$; **C**, **C**$^{\bullet}$). Однако такое определение весьма трудоёмко и особенно для большой мерности пространства. Оно не обладает необходимой прозрачностью для дальнейшего использования.

На рис. 1.11, *а*, *б* приведены исходное положение векторов аргументов трехмерного цифрового пространства и новые положения этих векторов, которые получаются соответствующими простыми поворотами.

Повороты относительно осей симметрии под номерами *1–9* выполняются одновременными операциями над всеми «слоями» элементарных кубиков цифрового пространства. Остальные повороты осуществляются относительно известных из кристаллографии 13 осей симметрии кубика. Повороты с номерами *10 –12* выполняются последовательно через 90° (три поворота), повороты с номерами от *13* до *18* – одиночные через 180°, а с номерами от *19* до *22* – последовательные двойные повороты через 120° .

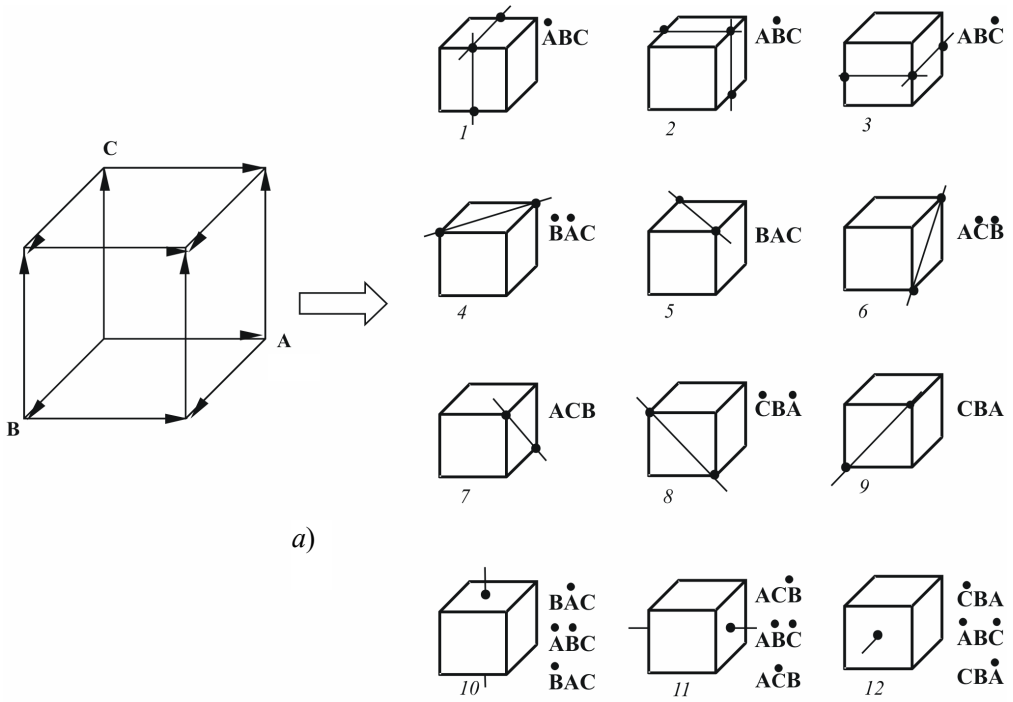
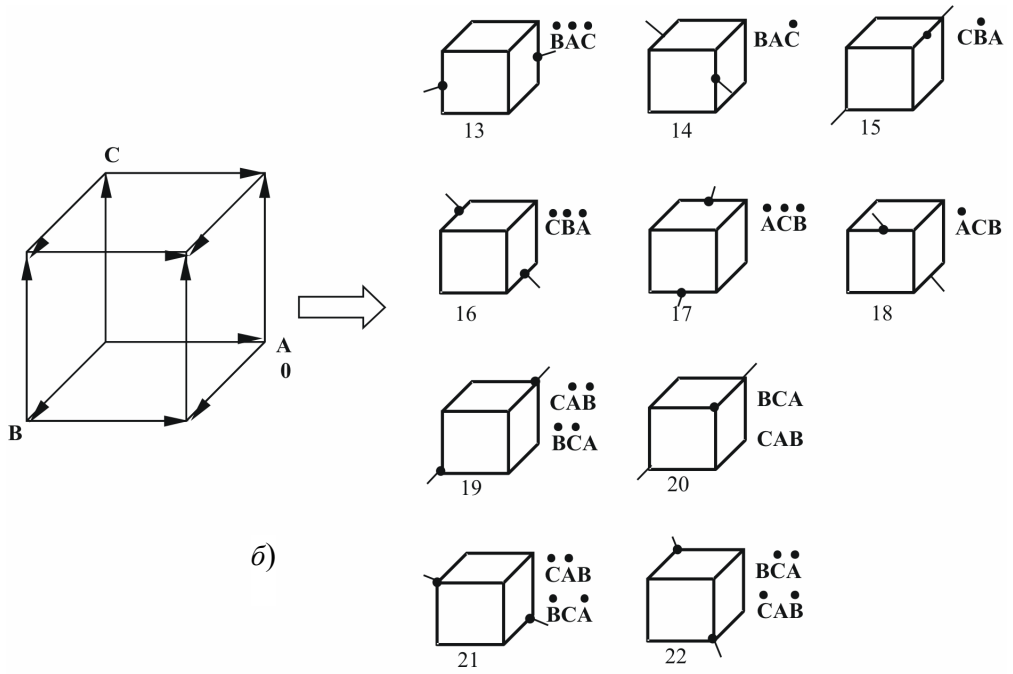


Рис. 1.11 (начало)



б)

Рис. 1.11 (продолжение)

Взаимные расположения векторов аргументов (их вместе с исходным 33), представленных выше, не определяют всего их числа. Другие 15 положений векторов аргументов образуются сложными поворотами, например, новое положение векторов $\mathbf{A^{\bullet}B^{\bullet}C^{\bullet}}$ может быть получено следующими поворотами из первоначального положения: **ABC** 1→ $\mathbf{A^{\bullet}BC}$ 2→ $\mathbf{A^{\bullet}B^{\bullet}C}$ 3→ $\mathbf{A^{\bullet}B^{\bullet}C^{\bullet}}$ либо **ABC** 17→ $\mathbf{A^{\bullet}C^{\bullet}B^{\bullet}}$ 7→ $\mathbf{A^{\bullet}B^{\bullet}C^{\bullet}}$ и т.д. Здесь, например, запись **ABC** 1→ $\mathbf{A^{\bullet}BC}$ означает, что исходное положение координат аргументов **ABC** при осуществлении поворота относительно оси *1* приведет к смене аргумента **A** в исходной системе координат из прямого кода в обратный, что в этом случае будет записано как $\mathbf{A^{\bullet}BC}$, и т.д.

Приведем графическое представление изменений исходной системы координат **ABC** при осуществлении соответствующих поворотов относительно приведенных выше осей симметрии трехмерного цифрового пространства.

С целью сокращения использования всех указанных выше поворотов и удобства аналитической записи преобразований аналогично тому, как это было сделано при рассмотрении двухмерного цифрового пространства, введем минимальные базисы осей поворотов.

Если для одномерного цифрового пространства **A** – это один поворот $\mathbf{A}^{\circlearrowright}$, то для двухмерного пространства **AB** к нему необходимо добавить поворот второго вектора $\mathbf{B}^{\circlearrowright}$ и поворот $(\mathbf{AB})^{\circlearrowright}$ (иное обозначение: $(\mathbf{AB})^{\circlearrowright}$ 180°).

Для трехмерного цифрового пространства **ABC** к этим поворотам необходимо добавить поворот **C** ↻ и два поворота относительно оси под номером 20 (см. рис. 1.11, *б*), которые представляются следующим образом: (**ABC**) ↻ 120°, (**ABC**) ↻ 240°.

Исходя из выбранного базиса осей поворотов трехмерного цифрового пространства, взаимное расположение векторов аргументов сложной логической функции с обозначением соответствующей операции поворота приведено на рис. 1.12. Это весьма прозрачно показывают все 48 взаимных положений векторов аргументов этого пространства.

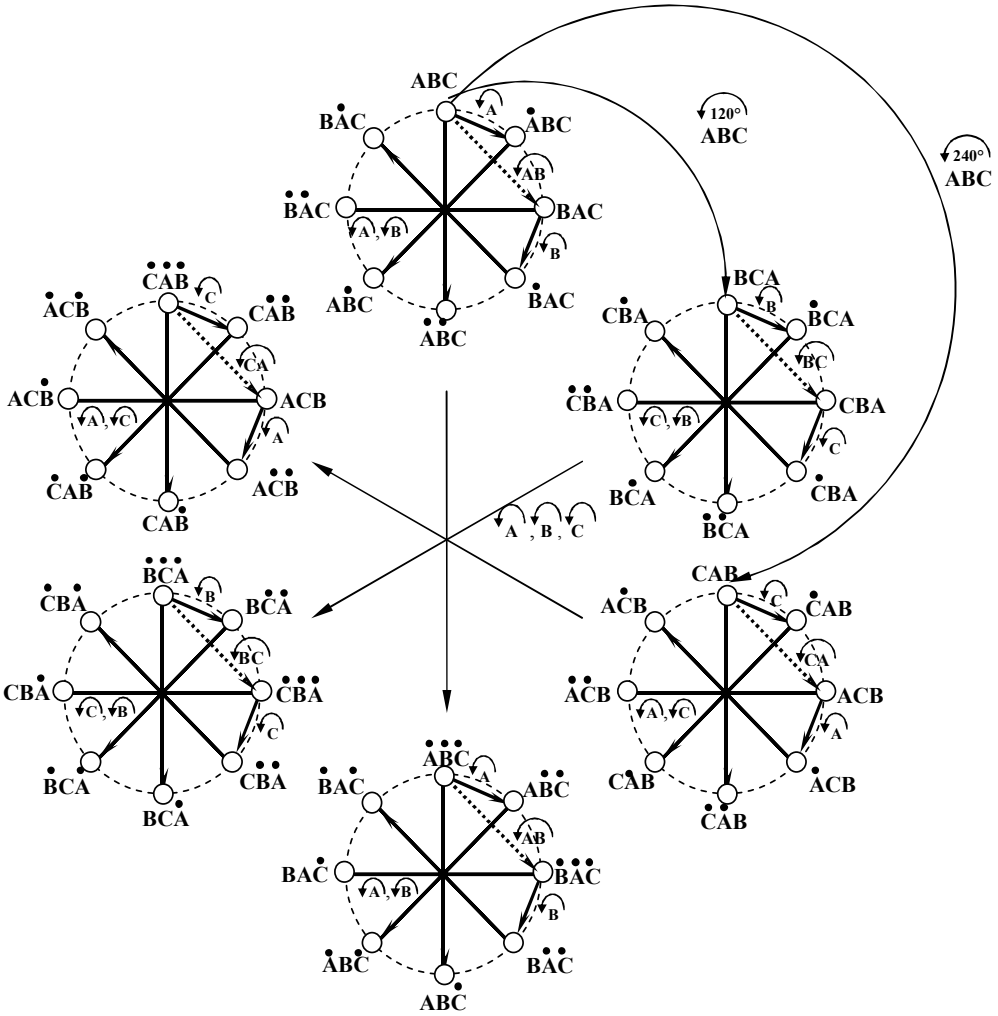


Рис. 1.12

В общем случае можно утверждать, что гипотетическая асимметричная относительно всех осей симметрии фигура сложной логической функции трехмерного цифрового пространства может занять в пространстве 48 положений.

Принципиальная схема логического блока, реализующая покрытие каждой новой (в смысле ее расположения в пространстве) фигуры, совпадает с принципиальной первообразной схемой блока, где на входе произведены соответствующие перестановки аргументов и перевод их из прямого кода в обратный, как это показано на рис. 1.12.

В том случае, если фигура симметрична относительно какой-либо оси, то поворот относительно этой оси не изменит положения фигуры в цифровом пространстве. При этом логическая схема блока, несмотря на соответствующие замены аргументов и перевод их из прямого кода в обратный, останется неизменной.

Здесь необходимо уточнить, что наличие симметрии может быть и в каждом аргументе, т.е. между множествами сигналов его составляющих, а также между множествами сигналов с другими аргументами. В этом случае также логическая схема блока, несмотря на замены аргументов и перевод их из прямого кода в обратный, останется неизменной.

Для рассмотрения симметрии цифрового пространства большей мерности продолжим расширение минимального базиса осей поворотов и представим его следующей табличной схемой:

A AB	A↻ A↻	B↻	AB↻						
ABC	A↻	B↻	C↻	AB↻	BC↻	CA↻	ABC ↻ 120	ABC ↻ 240	
ABCD		A↻	B↻	C↻	D↻	AB↻	BC↻	CA↻	DA↻
		ABC ↻ 120		BCD↻ 120		CDA ↻ 120		DAB↻ 120	
		ABC↻ 240		BCD↻ 240		CDA↻ 240		DAB↻ 240	
		ABCD↻ 90		ABCD ↻180		ABCD↻ 270		и т.д.	

Из неё следует, что если известны все взаимные положения векторов аргументов сложной функции для (k – 1)-мерного цифрового пространства, например с числом $S_{(K-1)}$, то для k-й мерности цифрового пространства взаимное положение векторов будет определяться соответствующими поворотами (k – 1)-мерности пространства. Число различных положений векторов аргументов сложной функции определяется следующей зависимостью: $S_K = S_{(K-1)}\,2K = 2^K(K!)$.

Для одномерного пространства это значение равно 2, двухмерного – 8, трехмерного – 48, четырехмерного – 384, пятимерного – 3840 и т.д. Значение этих чисел резко возрастает с увеличением мерности пространства, но они с такой же интенсивностью уменьшают и общее число логических функций многомерного цифрового пространства, которые нам необходимо изучать.

Для большей прозрачности представленных в этом разделе материалов рассмотрим трехмерное цифровое пространство ABC, где основание системы счисления равно 2, а в его ячейках размещены некоторые цифровые подмножества – геометрические образы логических функций $F_0 – F_7$. Тогда на основании изложенного выше покажем новые положения этих подмножеств цифрового пространства при выполнении мысленных преобразований координат (см. рис. 1.12), что представляется следующим образом:

A B C			
F_0	F_1	F_2	F_3
F_4	F_5	F_6	F_7

,

$A^\bullet$ B C			
F_1	F_0	F_3	F_2
F_5	F_4	F_7	F_6

,

B A C			
F_0	F_2	F_1	F_3
F_4	F_6	F_5	F_7

,

$B^\bullet$ A C			
F_2	F_0	F_3	F_1
F_6	F_4	F_7	F_5

,

$A^\bullet$ $B^\bullet$ C			
F_3	F_2	F_1	F_0
F_7	F_6	F_5	F_4

,

A $B^\bullet$ C			
F_2	F_3	F_0	F_1
F_2	F_7	F_4	F_5

,

$B^\bullet$ $A^\bullet$ C			
F_3	F_1	F_2	F_0
F_7	F_5	F_6	F_4

,

B $A^\bullet$ C			
F_1	F_3	F_0	F_2
F_5	F_7	F_4	F_6

,

$A^\bullet$ $B^\bullet$ $C^\bullet$			
F_7	F_6	F_5	F_4
F_3	F_2	F_1	F_0

,

A $B^\bullet$ $C^\bullet$			
F_6	F_7	F_4	F_5
F_2	F_3	F_0	F_1

,

$B^\bullet$ $A^\bullet$ $C^\bullet$			
F_7	F_5	F_6	F_4
F_3	F_1	F_2	F_0

,

B $A^\bullet$ $C^\bullet$			
F_5	F_7	F_4	F_6
F_1	F_3	F_0	F_2

,

A B $C^\bullet$			
F_4	F_5	F_6	F_7
F_0	F_1	F_2	F_3

,

$A^\bullet$ B $C^\bullet$			
F_5	F_4	F_7	F_6
F_1	F_0	F_3	F_2

,

B A $C^\bullet$			
F_4	F_6	F_5	F_7
F_0	F_2	F_1	F_3

,

$B^\bullet$ A $C^\bullet$			
F_6	F_4	F_7	F_5
F_2	F_0	F_3	F_1

,

B C A			
F_0	F_2	F_4	F_6
F_1	F_3	F_5	F_7

,

$B^\bullet$ C A			
F_2	F_0	F_6	F_4
F_3	F_1	F_7	F_5

,

C B A			
F_0	F_4	F_2	F_6
F_1	F_5	F_3	F_7

,

$C^\bullet$ B A			
F_4	F_0	F_6	F_2
F_5	F_1	F_7	F_3

,

$B^\bullet$ $C^\bullet$ A			
F_6	F_4	F_2	F_0
F_7	F_5	F_3	F_1

,

B $C^\bullet$ A			
F_4	F_6	F_0	F_2
F_5	F_7	F_1	F_3

,

$C^\bullet$ $B^\bullet$ A			
F_6	F_2	F_4	F_0
F_7	F_3	F_5	F_1

,

C $B^\bullet$ A			
F_2	F_6	F_0	F_4
F_3	F_7	F_1	F_5

,

$B^\bullet$ $C^\bullet$ $A^\bullet$			
F_7	F_5	F_3	F_1
F_6	F_4	F_2	F_0

,

B $C^\bullet$ $A^\bullet$			
F_5	F_7	F_1	F_3
F_4	F_6	F_0	F_2

,

$C^\bullet$ $B^\bullet$ $A^\bullet$			
F_7	F_3	F_5	F_1
F_6	F_2	F_4	F_0

,

C $B^\bullet$ $A^\bullet$			
F_3	F_7	F_1	F_5
F_2	F_6	F_0	F_4

,

B C $A^\bullet$			
F_1	F_3	F_5	F_7
F_0	F_2	F_4	F_6

,

$B^\bullet$ C $A^\bullet$			
F_3	F_1	F_7	F_5
F_2	F_0	F_6	F_4

,

C B $A^\bullet$			
F_1	F_5	F_3	F_7
F_0	F_4	F_2	F_6

,

$C^\bullet$ B $A^\bullet$			
F_5	F_1	F_7	F_3
F_4	F_0	F_6	F_2

,

C A B			
F_0	F_4	F_1	F_5
F_2	F_6	F_3	F_7

,

$C^\bullet$ A B			
F_4	F_0	F_5	F_1
F_6	F_2	F_7	F_3

,

A C B			
F_0	F_1	F_4	F_5
F_2	F_3	F_6	F_7

,

$A^\bullet$ C B			
F_1	F_0	F_5	F_4
F_3	F_2	F_7	F_6

,

$C^\bullet$ $A^\bullet$ B			
F_5	F_1	F_4	F_0
F_7	F_3	F_6	F_2

,

C $A^\bullet$ B			
F_1	F_5	F_0	F_4
F_3	F_7	F_2	F_6

,

$A^\bullet$ $C^\bullet$ B			
F_5	F_4	F_1	F_0
F_7	F_6	F_3	F_2

,

A $C^\bullet$ B			
F_4	F_5	F_0	F_1
F_7	F_7	F_2	F_3

,

$C^\bullet$ $A^\bullet$ $B^\bullet$			
F_7	F_3	F_6	F_2
F_5	F_1	F_4	F_0

,

C $A^\bullet$ $B^\bullet$			
F_3	F_7	F_2	F_6
F_1	F_5	F_0	F_4

,

$A^\bullet$ $C^\bullet$ $B^\bullet$			
F_7	F_6	F_3	F_2
F_5	F_4	F_1	F_0

,

A $C^\bullet$ $B^\bullet$			
F_6	F_7	F_2	F_3
F_4	F_5	F_0	F_1

,

C A $B^\bullet$			
F_2	F_6	F_3	F_7
F_0	F_4	F_1	F_5

,

$C^\bullet$ A $B^\bullet$			
F_6	F_2	F_7	F_3
F_4	F_0	F_5	F_1

,

A C $B^\bullet$			
F_2	F_3	F_6	F_7
F_0	F_1	F_4	F_5

,

$A^\bullet$ C $B^\bullet$			
F_3	F_2	F_7	F_6
F_1	F_0	F_5	F_4

.

Из этого пояснения становится ясно, каким образом изложенные в п. 1.5 оптимальные покрытия «объемов» 13 первообразных фигур трехмерного цифрового пространства определяют все 254 варианта выполнения трехвходового логического блока.

Как известно, по определению эквивалентные множества имеют одинаковую мощность и каждому элементу одного из них можно поставить взаимно однозначно некоторый элемент другого. Исходя из этого определения, все геометрические фигуры многомерного цифрового пространства, которые мысленными поворотами перемещаются в другое место этого пространства, можно назвать эквивалентными.

Однако здесь имеется более глубокая эквивалентность цифровых множеств, поскольку геометрические образы этих множеств сохраняют свою форму, т.е. подмножества этих множеств в результате таких мысленных операций не разбиваются на более мелкие и не объединяются в более крупные подмножества. Следовательно, в нашем случае цифровое множество (геометрический образ) одного из них и цифровое множество другого имеют одинаковую структуру подмножеств, составленных из цифр натурального ряда, которые отличаются в каждом подмножестве на единицу, и число таких подмножеств в этих эквивалентных множествах одинаково.

1.8. Непрерывное множество ряда натуральных чисел в многомерном цифровом пространстве

Из предыдущего рассмотрения нетрудно представить физический образ непрерывного множества натуральных чисел от 0 до (N – 1) для любых оснований систем счисления и принципов их кодирования. Этот физический образ с увеличением N непрерывно изменяет свою форму. Тем не менее можно записать оптимальное покрытие этой многомерной фигуры, т.е. определить логическую тупиковую функцию в базисе ДНФ [3].

Логическая функция этого непрерывного множества, определяемого числом его элементарных кубиков и нулевым началом отсчета вектора, это его «длина» L. Например, для четырехразрядного представления «длины» числа $a_4a_3a_2a_1$ она может быть записана следующим образом:

$$L = \boldsymbol{\mu}(a_4-1) \vee \mathbf{M}(a_4)\boldsymbol{\mu}(a_3-1) \vee \mathbf{M}(a_4)\mathbf{M}(a_3)\mu(a_2-1) \vee \\ \vee \mathbf{M}(a_4)\mathbf{M}(a_3)\mathbf{M}(a_2)\boldsymbol{\mu}(a_1-1). \qquad (1.8.1)$$

Здесь приняты следующие обозначения:

$\boldsymbol{\mu}(i-1)$ – цифровое непрерывное множество в пределах цифр разряда ($i = a_4, a_3, a_2, a_1$), которое содержит все цифры от **0** до (**i – 1**), т.е. $\boldsymbol{\mu}(i-1) = \mathbf{0} \vee \ldots \vee (\mathbf{i-1})$;

$\mathbf{M}(i)$ – цифровое множество, не обязательно непрерывное в пределах цифр разряда, но обязательно содержащее цифру i, т.е. $\mathbf{M}(i) = \mathbf{0} \vee \ldots \vee \mathbf{i} = \mathbf{1} \vee \ldots \vee \mathbf{i} = \ldots = \mathbf{i}$, где конкретный выбор этого множества определяется только принципом кодирования цифр основания системы счисления и весьма прозрачен.

При этом если i = 0, то $\boldsymbol{\mu}(i-1) = 0^*$ (логический нуль), если i = (n–1), то $\mathbf{M}(i) = 1^*$ (логическая единица).

Для большей наглядности изложения рассмотрим наиболее простые, но весьма важные примеры записи «длины» непрерывного множества для двоичного четырехразрядного пространства с номерами ячеек от **0** до **15** и представим все непрерывные множества, что в одномерном измерении изображено на рис. 1.13, а логические функции в базисе ДНФ в соответствии с (1.8.1) записываются следующим образом:

L(0001) =	0* ∨	00* ∨	000* ∨	0000,
L(0010) =	0* ∨	00* ∨	000 ∨	001*0*,
L(0011) =	0* ∨	00* ∨	000 ∨	001*0,
L(0100) =	0* ∨	00 ∨	01*0* ∨	01*00*,
L(0101) =	0* ∨	00 ∨	01*0* ∨	01*00,
L(0110) =	0* ∨	00 ∨	01*0 ∨	01*1*0*,
L(0111) =	0* ∨	00 ∨	01*0 ∨	01*1*0,
L(1000) =	0 ∨	1*0* ∨	1*00* ∨	1*000*,
L(1010) =	0 ∨	1*0* ∨	1*00 ∨	1*01*0,
L(1011) =	0 ∨	1*0* ∨	1*00 ∨	1*01*0,
L(1100) =	0 ∨	1*0 ∨	1*10* ∨	1*1*00*,
L(1101) =	0 ∨	1*0 ∨	1*1*0* ∨	1*1*00,
L(1110) =	0 ∨	1*0 ∨	1*1*0 ∨	1*1*1*0*,
L(1111) =	0 ∨	1*0 ∨	1*1*0 ∨	1*1*1*0.

Учитывая, что $0_i = \underline{i}$, эти зависимости можно представить в более привычном виде:

L(0001) =				$\underline{a_4}\,\underline{a_3}\,\underline{a_2}\,\underline{a_1}$,
L(0010) =			$\underline{a_4}\,\underline{a_3}\,\underline{a_2}$,	
L(0011) =			$\underline{a_4}\,\underline{a_3}\,\underline{a_2}$ ∨	$\underline{a_4}\,\underline{a_3}\,\underline{a_1}$,
L(0100) =		$\underline{a_4}\,\underline{a_3}$,		
L(0101) =		$\underline{a_4}\,\underline{a_3}$ ∨		$\underline{a_4}\,\underline{a_2}\,\underline{a_1}$,
L(0110) =		$\underline{a_4}\,\underline{a_3}$ ∨	$\underline{a_4}\,\underline{a_2}$,	
L(0111) =		$\underline{a_4}\,\underline{a_3}$ ∨	$\underline{a_4}\,\underline{a_2}$ ∨	$\underline{a_4}\,\underline{a_1}$,
L(1000) =	$\underline{a_4}$,			
L(1001) =	$\underline{a_4}$ ∨			$\underline{a_3}\,\underline{a_2}\,\underline{a_1}$,
L(1010) =	$\underline{a_4}$ ∨		$\underline{a_3}\,\underline{a_2}$,	
L(1011) =	$\underline{a_4}$ ∨		$\underline{a_3}\,\underline{a_2}$ ∨	$\underline{a_3}\,\underline{a_1}$,
L(1100) =	$\underline{a_4}$ ∨	$\underline{a_3}$,		
L(1101) =	$\underline{a_4}$ ∨	$\underline{a_3}$ ∨		$\underline{a_2}\,\underline{a_1}$,
L(1110) =	$\underline{a_4}$ ∨	$\underline{a_3}$ ∨	$\underline{a_2}$,	
L(1111) =	$\underline{a_4}$ ∨	$\underline{a_3}$ ∨	$\underline{a_2}$ ∨	$\underline{a_1}$.

(1.8.2)

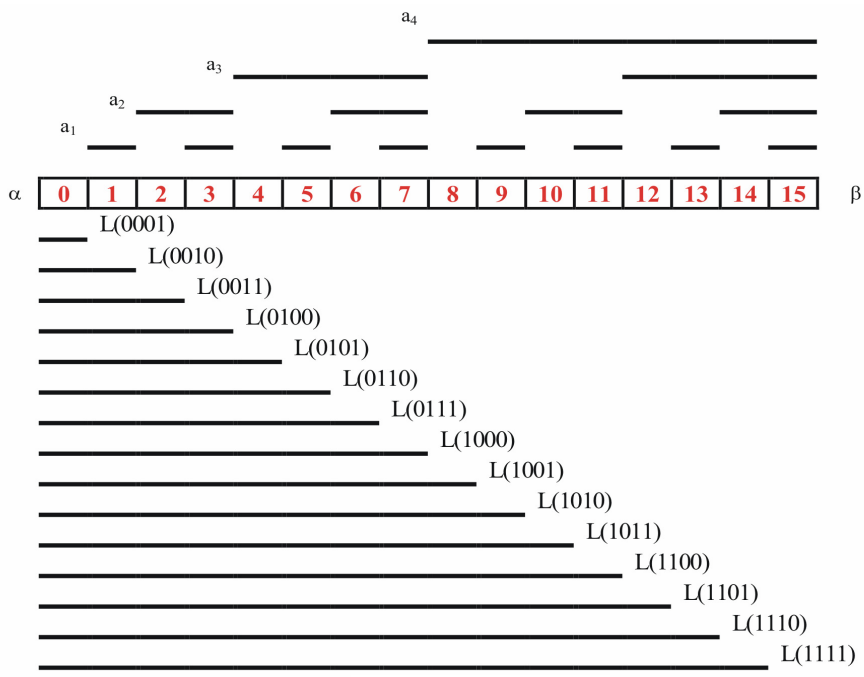


Рис. 1.13

Очевидно, что если «длина» непрерывного множества задана с началом отсчета от максимального номера кубика пространства, то все аргументы в (1.8.1) должны быть изменены на обратные (для двоичного кода – инвертированы).

Следовательно, если начало вектора непрерывного цифрового множества задано в цифровом пространстве номером ячейки N_1, а конец вектора – номером ячейки N_2, то, определяя непрерывное множество с нулевым началом отсчета вектора $L^{\alpha}(N_2)$ и непрерывное множество с началом отсчета вектора от максимального номера ячейки цифрового пространства $L^{\beta}(N_1)$, искомое непрерывное множество будет представлено следующей логической зависимостью:

$$L(N_1,N_2) = L^{\alpha}(N_2)\, L^{\beta}(N_1). \qquad (1.8.3)$$

С увеличением количества двоичных разрядов число слагаемых в (1.8.1) увеличивается и возникают аппаратурные трудности ее реализации. В этом случае наиболее целесообразно перейти на большие основания систем счисления, например n= 16. Тогда цифровые и непрерывные в пределах разряда множества **μ(**i – 1**)** полностью определяются логическим выражением (1.8.2), а оптимальные цифровые множества **M(**i**)**, которые содержат в пределах разряда только цифру i, определяются при минимальных аппаратурных затратах последними слагаемыми в (1.8.2). Эти слагаемые в (1.8.2) выделены светлым шрифтом.

Оптимальные цифровые множества **M(**i**) (**i =**0**, ... , **14**)** в пределах разряда приведены на рис. 1.14.

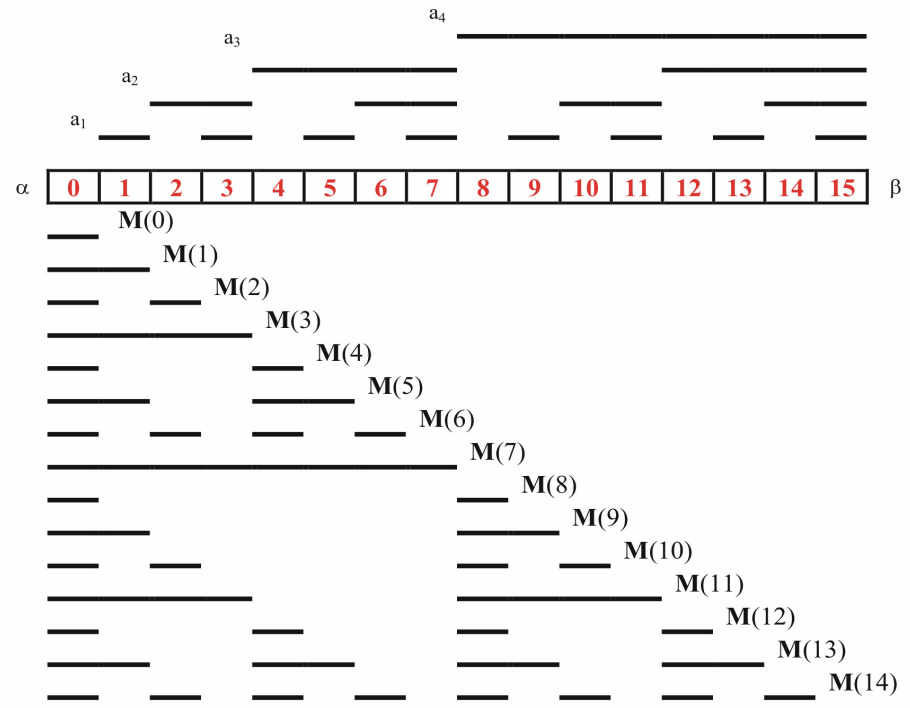


Рис. 1.14

После формирования в каждом разряде сигналов **μ(**i–1**)** и **M(**i**)** они поступают в логический блок, конструкция которого определяется выражением (1.8.1).

Необходимость хотя бы такого весьма краткого рассмотрения непрерывных множеств натурального ряда чисел связана с их широким использованием при формировании выходных и внутренних сигналов инверторов и конверторов напряжений, а также в системах регулирования электроприводов и в устройствах машинной арифметики.

В дальнейшем, при рассмотрении конкретных устройств в этих областях техники, сведения об этих типах множеств будут расширяться и дополняться.

Несколько слов относительно терминологии для этих цифровых множеств. В тех случаях, когда один из концов цифрового вектора множества совпадает с полюсом α (начало системы координат) либо противоположным ему полюсом β (конец системы координат), эти цифровые множества будут называться одномерными угловыми либо сигналами интегрального кода. В литературе можно встретить и иное определение таких множеств, например «ртутный столбик».

1.9. Многозначные логические функции в многомерном цифровом пространстве

Известно, что при создании первой цифровой вычислительной машины ее авторами было принято решение использовать десятичную систему счисления. В 1943 г. в США была начата и выполнена такая разработка электронной машины ENIAC для расчета баллистических таблиц по заказу военного ведомства. В 1945 г. к этой работе в качестве консультанта был приглашен Дж. фон Нейман. В результате дискуссий трех главных идеологов построения вычислительных машин Дж. фон Неймана, Г. Гольдстайна, Ф. Беркса возникла идея использования в дальнейшем при создании вычислительных машин двоичной системы счисления.

Эти ученые изложили (1946) основные принципы построения вычислительных машин нового поколения в ставшей теперь классической статье «Предварительное рассмотрение логической конструкции электронного устройства» (Кибернетический сборник. М.: Мир, 1964. № 9).

Главные положения этой статьи – обоснование использования двоичной системы счислений для представления чисел и принцип «хранимой программы». Наряду с обоснованием применения двоичной системы счислений и «хранимой программы» в работе предлагался ряд важных рекомендаций по конструированию машин и методике программирования.

Ряд отечественных исследователей в своих публикациях [23–25] обращают внимание на необходимость использования не двоичной, а многозначной логики, в частности трехзначной [20], как основы построения высокопроизводительных цифровых систем.

В МГУ была создана машина «Сетунь» – единственная в мире ЭВМ, в которой использована такая троичная система счисления, и, как утверждали авторы разработки, самая экономичная с точки зрения использования аппаратных средств.

Неудачный опыт использования трехзначной логики в вычислительной машине «Сетунь» вызвал критику такого подхода в ряде зарубежных публикаций, из которых наиболее острой является статья американского ученого Б. Байцера, в которой он пишет: «В течение многих лет близорукие теоретики, не давая себе труда проанализировать ошибочные допущения, на которых базировалось это “доказательство”, усиленно пропагандировали скрытые преимущества основания 3 как наилучшего приближения к **e**. Машина, использующая систему счисления с основанием 3, была построена в СССР, но она оказалась не столь уж хорошей. Пусть это послужит нам напоминанием того, что такое изящное доказательство, основанное на недействительных предпосылках, представляет собой некоторый курьез; ошибочными оказываются обычно допущения, положенные в основу некоторого исследования, а не сам ход этого исследования» [13].

Несмотря на эту жесткую критику, изучение многозначной логики все же необходимо для того, чтобы избежать таких ошибок при синтезе цифровых

устройств, а также понять причины все же удачного использования многозначной логики человеком в обычной жизни.

Исследования в предыдущих разделах логических функций в многомерном пространстве, когда их сложные аргументы эквивалентны цифрам натурального ряда и представлены в ОЦК, могут рассматриваться как начало изучения многозначной логики.

В самом деле, если двухзначная логика эквивалентна операциям над цифрами ОЦК основания n = 2, то многозначной логике можно поставить в соответствие операции над цифрами обычного кода большего основания. К изложенному в упомянутых выше разделах необходимо добавить некоторые пояснения.

Обычной p-значной логической функцией, в дальнейшем логической функцией $y_p = f(x_1, \dots, x_K)$, называется функция, принимающая, как и ее аргументы $x_1, \dots, x_K$, значения, которые обозначим символами 0*, 1*, ... , (p – 1)*.

Поскольку число различных наборов значений аргументов логической функции конечно и равно p^K, любая логическая функция может быть задана таблицей истинности с p^K строками, например для p=3 (табл. 1.9.1).

Таблица 1.9.1

ОЦК	x_1	x_2	$y_{p=3}$
0	0*	0*	0*
1	1*	0*	0*
2	2*	0*	2*
3	0*	1*	2*
4	1*	1*	0*
5	2*	1*	0*
6	0*	2*	1*
7	1*	2*	0*
8	2*	2*	0*

Если всем наборам значений аргументов функции $y_p = f(x_1, \dots, x_K)$ поставить в соответствие сигналы ОЦК от **0** до ($\mathbf{p^K - 1}$), то каждый аргумент x_{ij} (i = 0, ... , K; j = 0*, ... , (p – 1)*) и сама функция будут являть собой определенные цифровые множества, которые могут быть представлены как на цифровой «прямой», так и в многомерном цифровом пространстве, где каждая координата определяется соответствующим разрядом либо группой разрядов.

Число различных логических функций определяется зависимостью p^{p^k}.

Эта зависимость (аналогично двухзначным функциям) может быть представлена через все возможные сочетания. Перед тем как представить эти зависимости, сделаем некоторые пояснения. Если значения двухзначных логических функций в многомерном цифровом нумерованном пространстве заполняют все его ячейки символами 0* и 1*, где одно крайнее заполнение соответствует «пустому» множеству, т.е. пустому для значений 1*, а другое – «универсальному» множеству, т.е. полному для 1*, которое может рассматриваться как «пустое»

для символов 0*, то многозначная логическая функция будет иметь p таких крайних значений, каждое из которых будет «универсальным» для одного конкретного значения p и «пустым» для остальных (p – 1) значений логической функции.

Число различных двухзначных логических функций, определяемое зависимостью (1.1.1), может быть записано в несколько ином виде

$$2^{3^{K}} = \sum_{i=0}^{i=2^{K}} C^{i}_{(2)^{k}} \Big|_{0^*=(2^K-1)}^{1^*=i} . \qquad (1.9.1)$$

Тогда число трехзначных логических функций может быть записано следующим образом:

$$3^{3^{K}} = \sum_{i=0}^{i=3^{K}} \left(C^{i}_{(2)^{k}} \Big|_{1^*,\ 0^*=(2^K-1)}^{2^*=i} \sum_{j=0}^{i=(3^K-i)} C^{j}_{(3^K-i)} \Big|_{0^*=(3^K-i-j)}^{1^*=i} \right) \qquad (1.9.2)$$

и т.д.

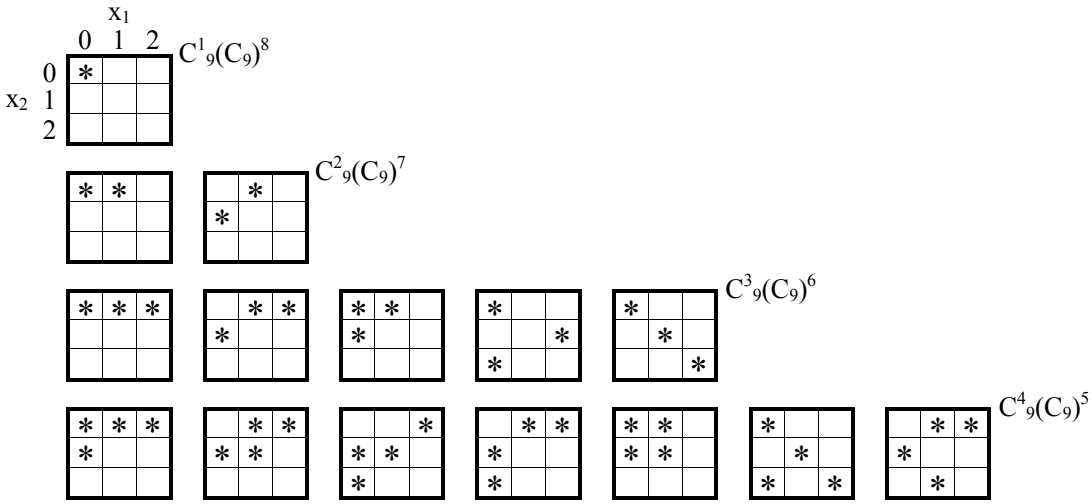


Рис. 1.15

Приведем пример перечисления всех трехзначных логических функций для K = 2 и будем размещать символы 0*, 1*, 2* в двухмерном цифровом пространстве. В соответствии с (1.9.2) получим:

$$
\begin{aligned}
&C_9^0 \Big|_{1^*,0^*=9}^{2^*=0} \Big(C_9^0 \Big|_{0^*=9}^{1^*=0} \quad C_9^1 \Big|_{0^*=8}^{1^*=1} \quad C_9^2 \Big|_{0^*=7}^{1^*=2} \quad C_9^3 \Big|_{0^*=6}^{1^*=3} \quad C_9^4 \Big|_{0^*=5}^{1^*=4} \; + \\
&\qquad + C_9^9 \Big|_{0^*=9}^{1^*=9} \quad C_9^8 \Big|_{0^*=1}^{1^*=8} \quad C_9^7 \Big|_{0^*=2}^{1^*=7} \quad C_9^6 \Big|_{0^*=3}^{1^*=6} \quad C_9^5 \Big|_{0^*=4}^{1^*=5} \Big) + \\
&+C_9^1 \Big|_{1^*,0^*=8}^{2^*=1} \Big(C_8^0 \Big|_{0^*=8}^{1^*=0} \quad C_8^1 \Big|_{0^*=7}^{1^*=1} \quad C_8^2 \Big|_{0^*=6}^{1^*=2} \quad C_8^3 \Big|_{0^*=5}^{1^*=3} \quad C_8^4 \Big|_{0^*=4}^{1^*=4} \; + \\
&\qquad + C_8^8 \Big|_{0^*=0}^{1^*=8} \quad C_8^7 \Big|_{0^*=1}^{1^*=7} \quad C_8^6 \Big|_{0^*=2}^{1^*=6} \quad C_8^5 \Big|_{0^*=3}^{1^*=5} \Big) + \\
&+C_9^2 \Big|_{1^*,0^*=7}^{2^*=2} \Big(C_7^0 \Big|_{0^*=7}^{1^*=0} \quad C_7^1 \Big|_{0^*=6}^{1^*=1} \quad C_7^2 \Big|_{0^*=5}^{1^*=2} \quad C_7^3 \Big|_{0^*=4}^{1^*=3} \; + \\
&\qquad + C_7^7 \Big|_{0^*=0}^{1^*=7} \quad C_7^6 \Big|_{0^*=1}^{1^*=6} \quad C_7^5 \Big|_{0^*=2}^{1^*=5} \quad C_7^4 \Big|_{0^*=3}^{1^*=4} \Big) + \\
&+C_9^3 \Big|_{1^*,0^*=6}^{2^*=3} \Big(C_6^0 \Big|_{0^*=6}^{1^*=0} \quad C_6^1 \Big|_{0^*=5}^{1^*=1} \quad C_6^2 \Big|_{0^*=4}^{1^*=2} \quad C_6^3 \Big|_{0^*=3}^{1^*=3} \; + \\
&\qquad + C_6^6 \Big|_{0^*=0}^{1^*=6} \quad C_6^5 \Big|_{0^*=1}^{1^*=5} \quad C_6^4 \Big|_{0^*=2}^{1^*=4} \Big) \quad + \; . \; . \; . \; + \\
&+C_9^8 \Big|_{1^*,0^*=1}^{2^*=8} \Big(C_1^0 \Big|_{0^*=1}^{1^*=0} + \quad C_1^1 \Big|_{0^*=0}^{1^*=1} \Big) + \quad C_9^9 \Big|_{1^*,0^*=0}^{2^*=9} = \quad 19683.
\end{aligned}
$$

Здесь первое подмножество множества сочетаний определяется отсутствием в девяти ячейках пространства символов 2* и размещением во всех этих ячейках только символов 1*,0*, что выражается суммой сочетаний $(C^0_9 + \ldots + C^9_9) = 2^9$; второе подмножество определяется наличием в девяти ячейках пространства одного символа 2* и размещением во всех остальных восьми ячейках символов 1*,0*, что также представлено суммированием сочетаний $(C^0_8 + \ldots + C^8_8) = 2^8$ и т.д.

Общее число всех трехзначных логических функций для двух аргументов весьма значительно, что, на первый взгляд, сулит большие сложности их представления и обозрения. Однако даже краткое изложение основ теории многомерных цифровых множеств, симметрии поворотов и переносов фигур в двухмерном пространстве позволяет представить все первообразные (без учета «пустого» и «универсального» множеств) только 15 фигурами. Эти фигуры приведены на рис. 1.15.

Очевидно, что для еще более значной логики, например для четырехзначной при К =2, где общее число всех логических функций равно 4 294 967 296, невозможно обойтись без применения приведенной выше методики их представления.

1.10. Алгоритмы синтеза двухзначных логических функций

Идея синтеза логических функций в базисе ДНФ с помощью вычислительных устройств всегда привлекала внимание многих исследователей и сулила им большие надежды, которые возрастали с появлением нового более быстродействующего поколения ЭВМ. Однако «победить» огромную трудоемкость, главным образом временную, получения оптимальных решений синтеза любых ЛФ с достаточно большим числом аргументов до настоящего времени не удалось. Очевидно, эта идея может быть реализована при весьма ограниченном числе аргументов.

В настоящее время создано несколько комплексов алгоритмов, каждый из которых имеет свое назначение, использует свой набор эвристических приемов и свои языки программирования [1, 2, 15].

Большинство из этих известных алгоритмов является тем или иным переложением на язык вычислительных машин широко известного метода синтеза, предложенного Квайном [27] и модифицированного Мак-Класки [26]. Метод Квайна – Мак-Класки предполагает, что минимизируемая ЛФ задана в совершенной ДНФ и все дальнейшие преобразования основаны на одном и том же математическом приеме алгебры логики: преобразование ДНФ достигается путем применения операции «склеивания» пар интервалов, на которых значения ЛФ отличны от нуля (0*).

Поиск этих интервалов осуществляется перебором ряда вариантов, где с помощью эвристических приемов стараются управлять перебором с ограничением их числа.

Многие исследователи считают, что этот аналитический путь поиска единственный, что он исчерпывает все возможности математического аппарата и поэтому усилия следует направить по пути создания подходящих языков программирования и определения лучших эвристик.

По нашему мнению, этот вывод ошибочен и, руководствуясь только этим чисто аналитическим подходом, ожидать больших успехов в автоматизации синтеза ЛФ не приходится, тем более что методы алгебры Буля не применимы для геометрических кодов, которые широко используются в рассматриваемых нами областях техники.

Вернемся к двухзначным ЛФ и их геометрическому образу в многомерном нумерованном цифровом пространстве.

Метод геометрического синтеза на основе построения геометрических образов логических и арифметических функций в многомерном цифро-векторном пространстве, а также изучение симметрии цифрового пространства и симметрии геометрических образов ЛФ позволяют успешно решать задачи автоматизированного синтеза логических блоков при любой значности и любом числе аргументов ЛФ.

При этом поиск оптимальных решений ведется с использованием ЭВМ в диалоговом режиме, когда на экране дисплея строится в многомерном цифро-

вом пространстве геометрический образ ЛФ и путем соответствующих операций в интерактивном режиме при взаимодействии с этим геометрическим образом находятся оптимальные его покрытия, которые определяют конструкцию синтезируемого логического блока.

Ограниченный объем данного раздела не позволяет детально остановиться на рассмотрении всех возможных вариантов выполнения алгоритмов оптимального синтеза логических и арифметических блоков, и основное внимание этому предмету будет перенесено нами в практическое русло рассмотрения примеров такого синтеза в отдельной главе. Тем не менее для доказательства вышеизложенного приведем самый простой, но достаточно эффективный алгоритм многоуровневого синтеза.

Сущность рассматриваемого способа синтеза и реализующего его алгоритма заключается в последовательном рассмотрении заполнения клеток нумерованного цифрового пространства, начиная с клеток старших, чем элементарные клетки, заполненные символами 0* и 1*, и определения всех двухзначных ЛФ.

После этого (второй шаг программы) рассматриваются следующие по старшинству ячейки пространства, в которых располагаются не символы 1* и 0*, а непосредственно геометрические образы ЛФ, определенные на первом шаге программы, и находятся все четырехзначные ЛФ цифрового пространства.

Очередным шагом синтеза является рассмотрение следующих по старшинству ячеек пространства, где расположены геометрические образы четырехзначных ЛФ, и т.д. вплоть до последней единственной ячейки пространства, представляющей все нумерованное цифровое пространство.

Определение покрытия этой ячейки пространства и является искомой ЛФ в ДНФ.

На каждом шаге программы поиск ЛФ, определяющих покрытие фигуры ячейки пространства, исходя из фигур младших на единицу ячеек, выполняется в соответствии с геометрическим образом:

		x_i
	$\mathbf{F}_o$	$\mathbf{F}_1$
x_{i-1}	$\mathbf{F}_2$	$\mathbf{F}_3$

по логическому выражению, представленному в общем виде

$$Q = \mathbf{F_o}\, \underline{x}_i\, \underline{x}_{i+1} \vee \mathbf{F_1}\, x_i\, \underline{x}_{i+1} \vee \mathbf{F_2}\, \underline{x}_i\, x_{i+1} \vee \mathbf{F_3}\, x_i\, x_{i+1}.$$

Это выражение может быть уменьшено в том случае, если функции $\mathbf{F_0}$, $\mathbf{F_1}$, $\mathbf{F_2}$, $\mathbf{F_3}$ принимают значения 0* или 1*, а также, если некоторые из них являются подмножествами других.

Например, если $\mathbf{F_o}$= 0*, а $\mathbf{F_3} \supset \mathbf{F_2}$ и $\mathbf{F_3} \supset \mathbf{F_1}$,то $\mathbf{Q} = \mathbf{F_1}\, x_i \vee \mathbf{F_2}\, x_{i+1} \vee \mathbf{F_3}\, x_i\, x_{i+1}$; если $\mathbf{F_o}$ = 0*, а $\mathbf{F_1} = \mathbf{F_2} = \mathbf{F_3} = \mathbf{F}$, то $\mathbf{Q} = \mathbf{F}\, x_i \vee \mathbf{F}\, x_{i+1}$, и т.д.

Определение включений одной фигуры в другую не вызывает каких-либо сложностей и осуществляется простым совмещением фигур на экране дисплея.

Представленная программа поясняется на примере синтеза шестизначной ЛФ, геометрический образ которой приведен в двухмерном цифровом пространстве на рис. 1.16.

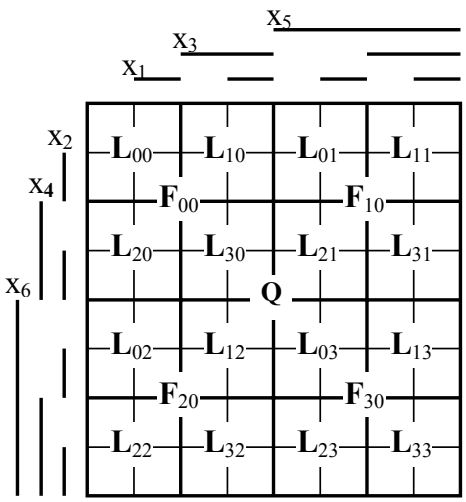


Рис. 1.16

На первом шаге программы определяется 16 элементарных двухзначных ЛФ ($\mathbf{L}_{00}$, ... , $\mathbf{L}_{30}$), ($\mathbf{L}_{01}$, ... , $\mathbf{L}_{31}$), ($\mathbf{L}_{02}$, ... , $\mathbf{L}_{32}$), ($\mathbf{L}_{03}$, ... , $\mathbf{L}_{33}$).

На втором шаге программы с учетом взаимного включения фигур в каждой группе из четырех двухзначных функций определяются четыре четырехзначные функции $\mathbf{F}_{00}$, ... , F_{30}.

На третьем, заключительном для данного примера шаге программы, определяется, также с учетом взаимного включения фигур в этой группе, шестизначная логическая функция $\mathbf{Q}_{00}$, которая и определяет покрытие соответствующей фигуры цифрового пространства в базисе ДНФ.

Представленная программа синтеза «жесткой» конструкции логического блока является универсальной и может служить основой для построения программируемого асинхронного устройства, реализующего все 2^{64} варианта шестизначных ЛФ.

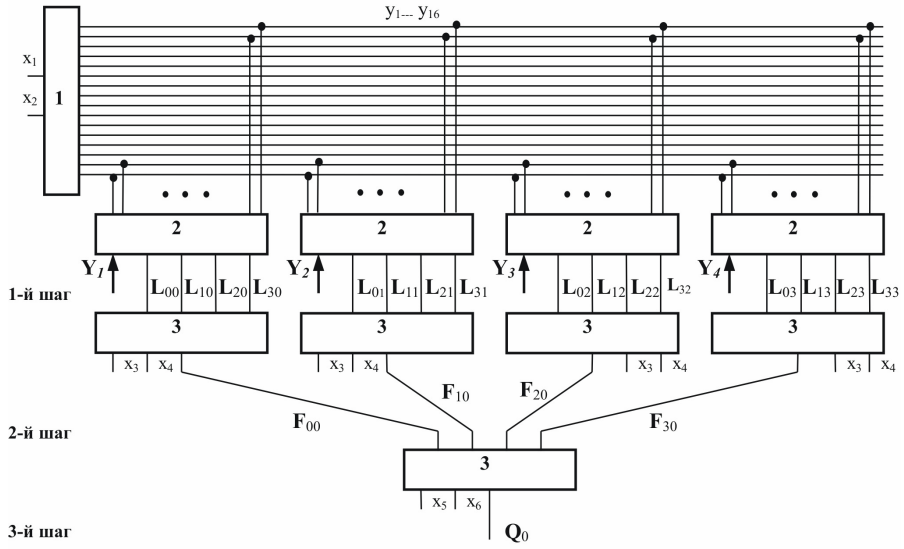


Рис. 1.17

Здесь следует, однако, заметить, что когда среди логических функций $\mathbf{L}_{ii}$, $\mathbf{F}_{ii}$ имеются одинаковые, т.е. их геометрические образы совпадают, либо они преобразуются одна в другую соответствующими поворотами относительно осей симметрии многомерного пространства, то программы синтеза «жесткой» конструкции могут претерпеть существенные упрощения.

Однако предусмотреть все эти варианты не представляется возможным, и поэтому на рис. 1.17 приведена обобщенная структурная схема реализации этого устройства, которая не претендует на минимальные аппаратурные затраты, поскольку не учитывает всех оговоренных выше условий.

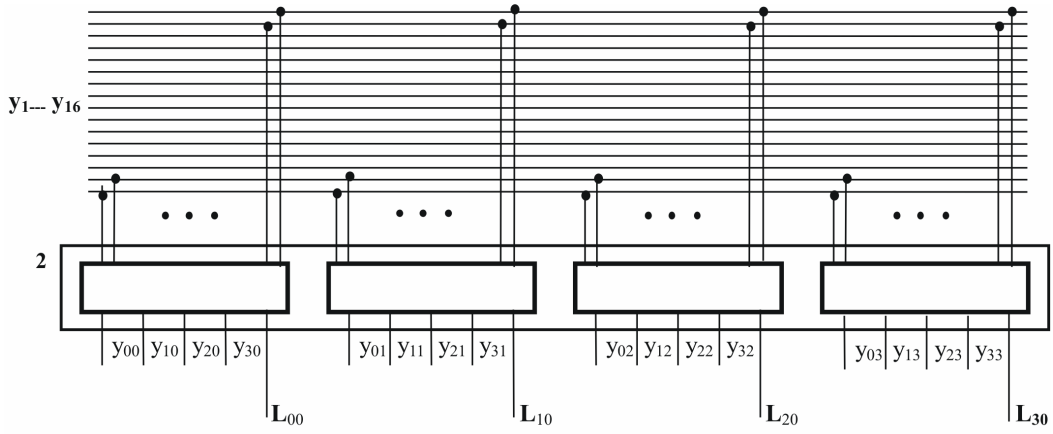


Рис. 1.18

В схеме имеются генератор двухзначных ЛФ 1 с шестнадцатью выходными шинами $y_1(x_1,x_2) - y_{16}(x_1,x_2)$ и четыре однотипных мультиплексных блока *2*, осуществляющих коммутацию сигналов функций $y_1(x_1,x_2) - y_{16}(x_1,x_2)$ на соответствующие шины $\mathbf{L_{0i}}$, ... , $\mathbf{L_{3i}}$ (первый шаг программы). Эта коммутация выполняется по сигналам управляющих команд $\mathbf{Y_1}$, $\mathbf{Y_2}$, $\mathbf{Y_3}$, $\mathbf{Y_4}$. Схема блока *2*, состоящая из четырех стандартных мультиплексоров, приведена на рис. 1.18. На втором шаге программы формируются на выходе логических схем функции $\mathbf{F}_{00}$, $\mathbf{F}_{10}$, $\mathbf{F}_{20}$, $\mathbf{F}_{30}$, а на третьем – логическая функция $\mathbf{Q}_{00}$.

Рассматриваемая теория многомерных цифровых множеств может быть успешно применена для любых многозначных логических функций как при синтезе «жестких» логических конструкций, так и при синтезе программных логических блоков. Это подтверждается широко известной концепцией об эквивалентности аппаратных и программных средств, которые лежат в основе методов программирования и позволяют «какой-нибудь специальный прием, используемый при проектировании аппаратуры, использовать и при программировании» [13]. При этом можно утверждать, что если имеется геометрический образ логической функции любой значности в многомерном цифровом пространстве, то задача синтеза как предмета изобретательской деятельности не существует, а эта работа может быть выполнена при помощи вычислительной техники или даже ручным способом грамотным специалистом. Достигнуть этого стало возможным потому, что получен четкий ответ на вопросы «как» и «почему».

Ответ на вопрос об известности геометрических образов логических функций либо известности законов их построения формулируется следующим образом: это известно для всех устройств машинной арифметики, основанных на

общеизвестных принципах выполнения операций (суммирования, вычитания, поразрядных деления и умножения), принципиально для любых устройств, реализующих известные операции в поле натуральных чисел. Причем это распространяется на любое число разрядов устройств, любые значения оснований систем счисления и любые принципы их кодирования, а также на все операции, попадающие под действие теории нумераций. Поэтому представленная выше теория многомерных цифровых множеств является геометрическим аналогом теории нумераций и дальнейшее развитие теории многомерных цифровых множеств как «чистой» теории должно происходить с учетом этой аналогии. Однако нас интересует практическое применение этой теории, и ее краткое изложение является, по нашему мнению, достаточным для решения задач синтеза логических и цифровых устройств в рассматриваемых областях техники.

В дальнейшем изложение материала в работе будет носить прикладной характер для решения конкретных задач в области преобразовательной техники, электропривода, обслуживающих их цифровых устройств, а также для построения систем в этих областях техники с повышенными показателями надежности и помехозащищенности. Создание таких цифровых систем управления непосредственно связано с использованием в них контролеспособных кодов с возможностью не только обнаружения, но, что наиболее сложно, и исправления ошибок в режиме реального времени. Эта весьма актуальная задача до настоящего времени никем не решена, поскольку при попытках ее решения известным классическим способом, где используется соответствующий математический аппарат, возникают сложности, связанные со значительными временными затратами. Ниже практически во всех разделах будут приведены способы решения этой задачи в рассматриваемых областях техники в режиме реального времени.

1.11. Примеры синтеза одноразрядного устройства суммирования и вычитания

С целью более четкого представления принципа построения геометрического образа ЛФ в многомерном цифровом пространстве рассмотрим примеры синтеза одноразрядного суммирующего и вычитающего устройства для основания системы счисления n = 8. Это выполним для двух вариантов кодирования данного основания: двоичного и многофазного.

Основой построения геометрического образа сигналов результата этих операций являются общеизвестные таблицы Пифагора. В табл. 1.11.1 представлен результат арифметической операции суммирования (**A+B**) при отсутствии сигнала переноса с предыдущего младшего разряда, а в табл. 1.11.2 – при наличии такого сигнала.

В этих таблицах выделены подмножества результата сложения в разряде, где произошло переполнение и получен перенос для единицы старшего разряда.

Таблица 1.11.1

B \ A	0	1	2	3	4	5	6	7
0	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0
2	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1
3	0.3	0.4	0.5	0.6	0.7	1.0	1.1	1.2
4	0.4	0.5	0.6	0.7	1.0	1.1	1.2	1.3
5	0.5	0.6	0.7	1.0	1.1	1.2	1.3	1.4
6	0.6	0.7	1.0	1.1	1.2	1.3	1.4	1.5
7	0.7	1.0	1.1	1.2	1.3	1.4	1.5	1.6

Таблица 1.11.2

B \ A	0	1	2	3	4	5	6	7
0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0
1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1
2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	1.2
3	0.4	0.5	0.6	0.7	1.0	1.1	1.2	1.3
4	0.5	0.6	0.7	1.0	1.1	1.2	1.3	1.4
5	0.6	0.7	1.0	1.1	1.2	1.3	1.4	1.5
6	0.7	1.0	1.1	1.2	1.3	1.4	1.5	1.6
7	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7

Для двоичного кодирования цифр 0–7 используются на примере первого операнда три сигнала $a_1 = 1 \vee 3 \vee 5 \vee 7$, $a_2 = 2 \vee 3 \vee 6 \vee 7$, $a_3 = 4 \vee 5 \vee 6 \vee 7$. В соответствии с представлением сигналов как множества цифр натурального ряда от **0** до 7 в ячейках нумерованного двухмерного цифрового пространства содержатся цифры, которые есть элементы подмножества сигналов разряда операндов. Для каждого сигнала результата операции в ячейках пространства располагается логическая единица 1* (звездочка), а остальные ячейки заполнены логическими нулями 0* (пустая ячейка).

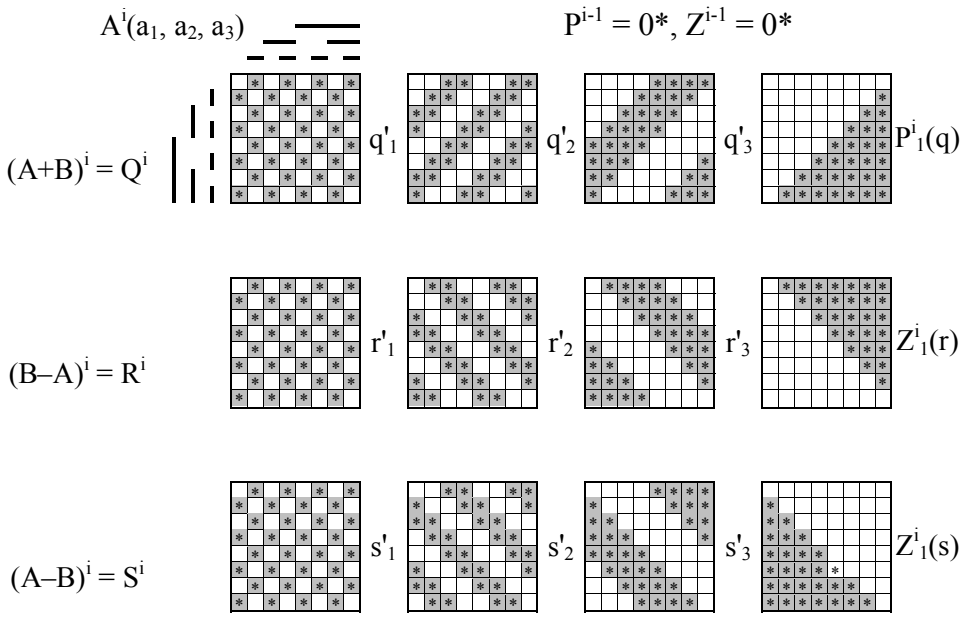


Рис. 1.19

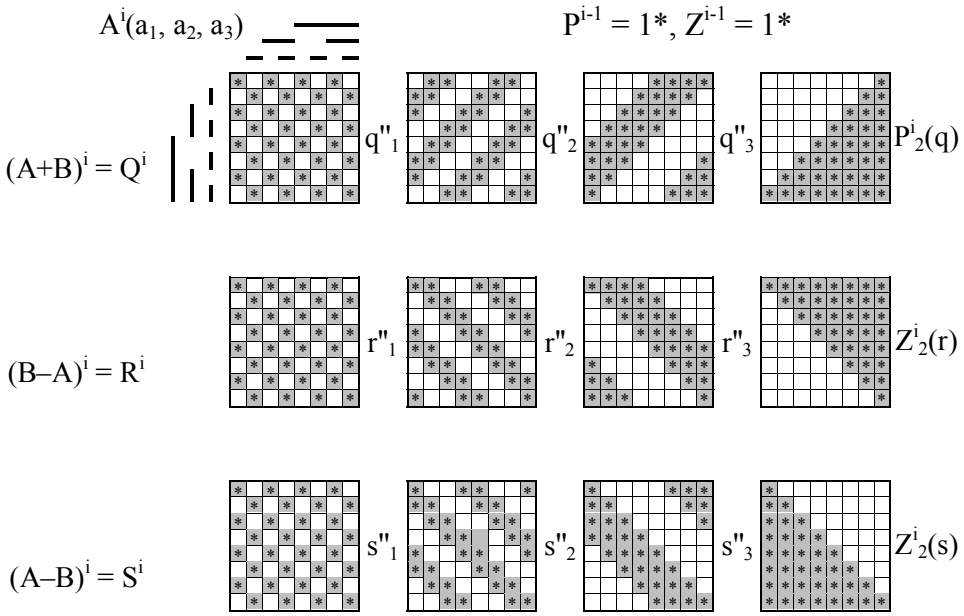


Рис. 1.20

В соответствии с этим на рис. 1.19 показаны геометрические образы сигналов i-го разряда q'_1, q'_2, q'_3 и переноса в старший разряд $P^i_1(q)$ для операции суммирования $\mathbf{(A + B)}^i$. Эти геометрические образы получаются при отсутствии сигнала переноса результата операции в младшем разряде $P^{i-1}(q)=0^*$, а на рис. 1.20 приведены аналогичные сигналы q''_1, q''_2, q''_3 , $P^i_2(q)$ при наличии такого сигнала переноса $P^{i-1}(q)=1^*$.

Проводя аналогичные действия, начиная с таблиц Пифагора, для операций вычитания $\mathbf{(B – A)}^i$, $\mathbf{(A – B)}^i$ можем получить соответствующие им геометрические образы. Эти фигуры приведены на этих же рисунках соответственно для операции вычитания $\mathbf{(B – A)}^i$ [сигналы i-го разряда r'_1, r'_2, r'_3, $Z^i_1(r)$ при отсутствии сигнала заёма из старшего разряда $Z^{i-1}=0^*$ и сигналы r''_1, r''_2, r''_3, $Z^i_2(r)$ при наличии сигнала заема для младшего разряда $Z^{i-1}=1^*$] и операции вычитания $\mathbf{(A – B)}^i$ [сигналы i-го разряда s'_1, s'_2, s'_3, $Z^i_1(s)$ при отсутствии сигнала заема для младшего разряда $Z^{i-1}=0^*$ и сигналы s''_1, s''_2, s''_3, $Z^i_2(s)$ при наличии сигнала заема для младшего разряда $Z^{i-1}=1^*$].

Таким образом, геометрические фигуры всех ЛФ со сложными аргументами рассматриваемых арифметических операций нами представлены и остается только их определить, т.е. выполнить процедуру покрытия этих фигур сигналами операнд, что приводилось в предыдущих разделах.

Поскольку выполнение всех арифметических операций для двоичного кодирования оснований систем счисления широко известно и без знания теории многомерных цифровых множеств, остановимся только на связях выходных

сигналов с их базовыми геометрическими образами, которыми в данном случае будут образы сигналов $q'_1(\mathbf{AB})$, $q'_2(\mathbf{AB})$, $q'_3(\mathbf{AB})$, $P^i(\mathbf{AB})$.

Остальные фигуры получаем соответствующими мысленными поворотами относительно осей симметрии двухмерного цифрового пространства, а именно:

$$q''_1 = \underline{q}'_1(\mathbf{AB}) = \underline{q}'_1(\underline{\mathbf{A}}^{\bullet}\mathbf{B}^{\bullet}) = q'_1(\mathbf{AB}^{\bullet}) = q'_1(\mathbf{A}^{\bullet}\mathbf{B}),\ q''_2 = \underline{q}'_2(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet}),\ q''_3 = \underline{q}'_3(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet});$$

$$r'_1 = q'_1(\mathbf{AB}) = q'_1(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet}) = \underline{q}'_1(\mathbf{AB}^{\bullet}) = \underline{q}'_1(\mathbf{A}^{\bullet}\mathbf{B}),\ r'_2 = \underline{q}'_2(\mathbf{AB}^{\bullet}),\ r'_3 = \underline{q}'_3(\mathbf{AB}^{\bullet});$$

$$r''_1 = \underline{q}'_1(\mathbf{AB}) = \underline{q}'_1(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet}) = q'_1(\mathbf{AB}^{\bullet}) = q'_1(\mathbf{A}^{\bullet}\mathbf{B}),\ r''_2 = q'_2(\mathbf{A}^{\bullet}\mathbf{B}),\ r''_3 = q'_3(\mathbf{A}^{\bullet}\mathbf{B});$$

$$s'_1 = \underline{q}'_1(\mathbf{AB}) = q'_1(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet}) = \underline{q}'_1(\mathbf{AB}^{\bullet}) = \underline{q}'_1(\mathbf{A}^{\bullet}\mathbf{B}),\ s''_2 = \underline{q}'_2(\mathbf{A}^{\bullet}\mathbf{B}),\ r''_3 = \underline{q}'_3(\mathbf{A}^{\bullet}\mathbf{B});$$

$$s''_1 = \underline{q}'_1(\mathbf{AB}) = \underline{q}'_1(\mathbf{A}^{\bullet}\mathbf{B}^{\bullet}) = q'_1(\mathbf{AB}^{\bullet}) = q'_1(\mathbf{AB}^{\bullet}),\ s''_2 = q'_2(\mathbf{AB}^{\bullet}),\ s''_3 = q'_3(\mathbf{AB}^{\bullet});$$

$$Z^i(r) = P^i(\mathbf{AB}^{\bullet}),\ Z^i(s) = P^i(\mathbf{A}^{\bullet}\mathbf{B}). \qquad (1.11.1)$$

Следовательно, зная покрытие базовых фигур логических функций операции суммирования $q'_1(\mathbf{AB})$, $q'_2(\mathbf{AB})$, $q'_3(\mathbf{AB})$ при отсутствии сигнала переноса с младшего разряда $P^{i-1}(\mathbf{AB}) = 0^*$, не представляет сложности получить аналогичные сигналы для операции суммирования при наличии такого переноса ($P^{i-1}(\mathbf{AB}) = 1^*$). Это справедливо также для операции вычитания как для случая, когда **A>B** , так и для **A<B**, при любом значении сигналов заёма со старшего разряда сумматора. Замечательным свойством при этом обладают сигналы первого разряда сумматора, которые имеют полную симметрию геометрического образа и при различных мысленных поворотах относительно осей симметрии пространства остаются неизменными.

Очевидно, что для любого i-го разряда базовой операции

$$q_j = q_j'\,\underline{P}^{i-1} + q_j''\,P^{i-1}.$$

Классическое построение вычислительных устройств, работающих в двоичной системе счисления, когда операция суммирования принимается базовой для всех остальных, общеизвестно, но получить большую прозрачность выполнения алгоритмов операций вычитания, чем те, которые представлены здесь, невозможно.

Для четных оснований систем счисления в сигналах кода операнд имеется сигнал, который является непрерывным множеством второй половины цифр основания системы счисления. В приведенном выше примере это – сигналы a_3, b_3, q_3, которые позволяют определить сигнал переноса P^i по следующему логическому выражению:

$$P^i = \underline{q}^i_3\, a^i_3 \vee \underline{q}^i_3\, b^i_3 \vee a^i_3\, b^i_3\,. \qquad (1.11.2)$$

В (1.11.2) нет сигнала переноса с младшего (i – 1)-го разряда, но он присутствует в формировании переноса P^i и интегрирован в сигнале $\underline{q}^i_3$.

На рис. 1.21, *а*, *б* показаны соответственно геометрические образы сигнала $\underline{q}^i_3$ при $P^{i-1} = 0^*$ и $P^{i-1} = 1^*$, из которых ясен принцип формирования цифровых подмножеств $a^i_3\, b^i_3$ и $\underline{q}^i_3\, a^i_3 \vee \underline{q}^i_3\, b^i_3$, составляющих цифровое множество P^i.

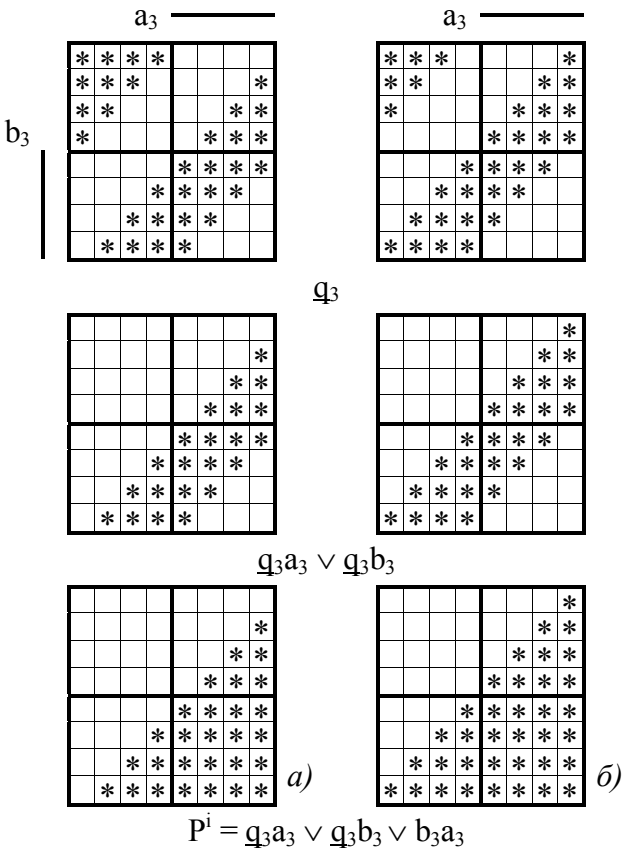


Рис. 1.21

Формула (1.11.2) – общая для любого принципа кодирования четных оснований систем счисления, где операнды содержат в коде разряда сигнал, являющийся непрерывным множеством цифр второй половины этого основания. Тогда выражение для любых оснований (1.11.2) будет преобразовано к виду

$$P^i = \underline{q}^i_k\, a^i_k \vee \underline{q}^i_k\, b^i_k \vee a^i_k\, b^i_k\,, \qquad (1.11.3)$$

где q^i_k, a^i_k, b^i_k – непрерывные цифровые множества операндов, содержащие цифры второй половины основания системы счисления.

Значение выражения (1.11.3) заключается не только в его простоте, а и в том, что если в них сигналы $\underline{q}^i_k$, a^i_k, b^i_k безошибочны, то также будет безошибочен сигнал переноса P^i i-го разряда.

При многофазном принципе кодирования цифр **0**–**7**, когда, на примере сигналов первого операнда, a_1 = **1** ∨ **…** ∨ **4**, a_2 = **2** ∨ **…** ∨ **5**, a_3 = **3** ∨ **…** ∨ **6**, a_4 = **4** ∨ **…** ∨ **7**, геометрические образы сигналов результата сложения $\mathbf{(A+B)}^i$ и вычитания $\mathbf{(B-A)}^i$ и $\mathbf{(A-B)}^i$ приведены на рис. 1.22 и 1.23. Эти рисунки анало-

гичны рис. 1.19 и 1.20 двоичного принципа кодирования этого же основания системы счисления.

На рис. 1.22 построены в двухмерном цифровом пространстве координат операндов **A**, **B** геометрические образы сигналов i-го разряда $q'_1,\ldots, q'_4$ и переноса в старший разряд P^i_1**(AB)** операции суммирования $(\mathbf{A+B})^i$, когда P^{i-1}_1**(AB)** = 0*, а на рис. 1.23 приведены такие же сигналы $q''_1,\ldots, q''_4$ и P^i_2**(AB)** при наличии сигнала переноса с младшего разряда P^{i-1}_1**(AB)** =1*.

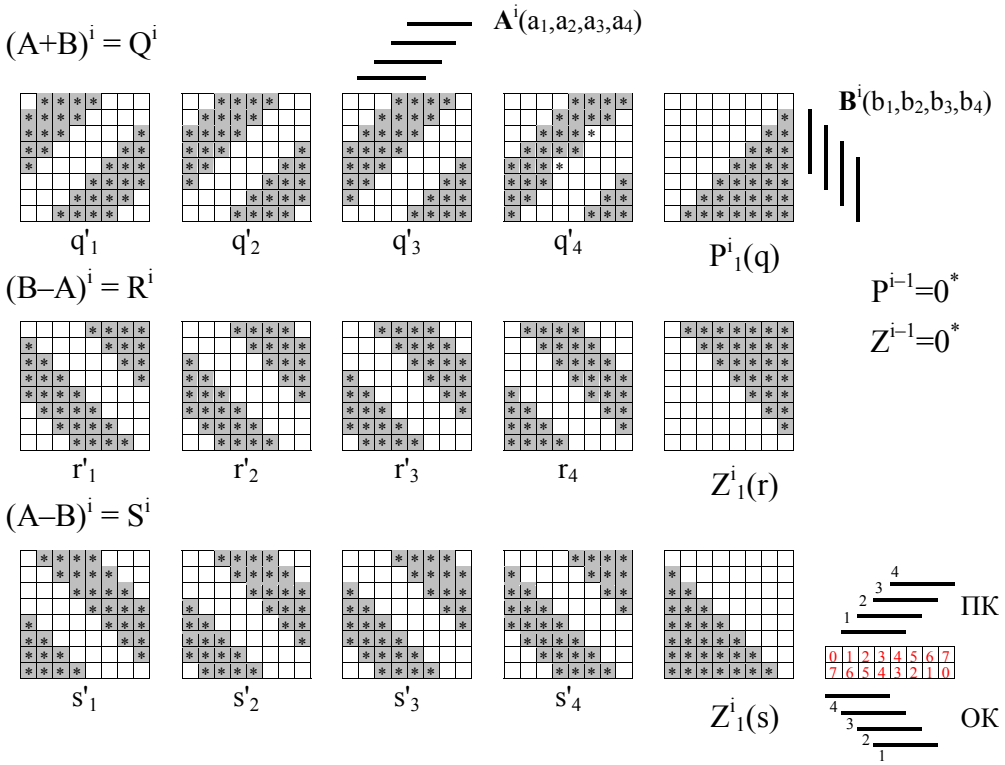


Рис. 1.22

На этих же рисунках показаны геометрические образы сигналов для операций вычитания. Эти фигуры приведены соответственно для операции вычитания $(\mathbf{B–A})^i$ [сигналы i-го разряда $r'_1, \ldots, r'_4$, $Z^i_1(r)$ при отсутствии сигнала заема для младшего разряда Z^{i-1}=0* и сигналы $r''_1, \ldots, r''_4$, $Z^i_2(r)$ при наличии сигнала заема для младшего разряда Z^{i-1}=1*] и для операции вычитания $(\mathbf{A–B})^i$ [сигналы i-го разряда $s'_1, \ldots, s'_4$, $Z^i_1(s)$ при отсутствии сигнала заема для младшего разряда Z^{i-1}=0* и сигналы $s''_1, \ldots, s''_4$, $Z^i_2(s)$ при наличии сигнала заема для младшего разряда Z^{i-1}=1*].

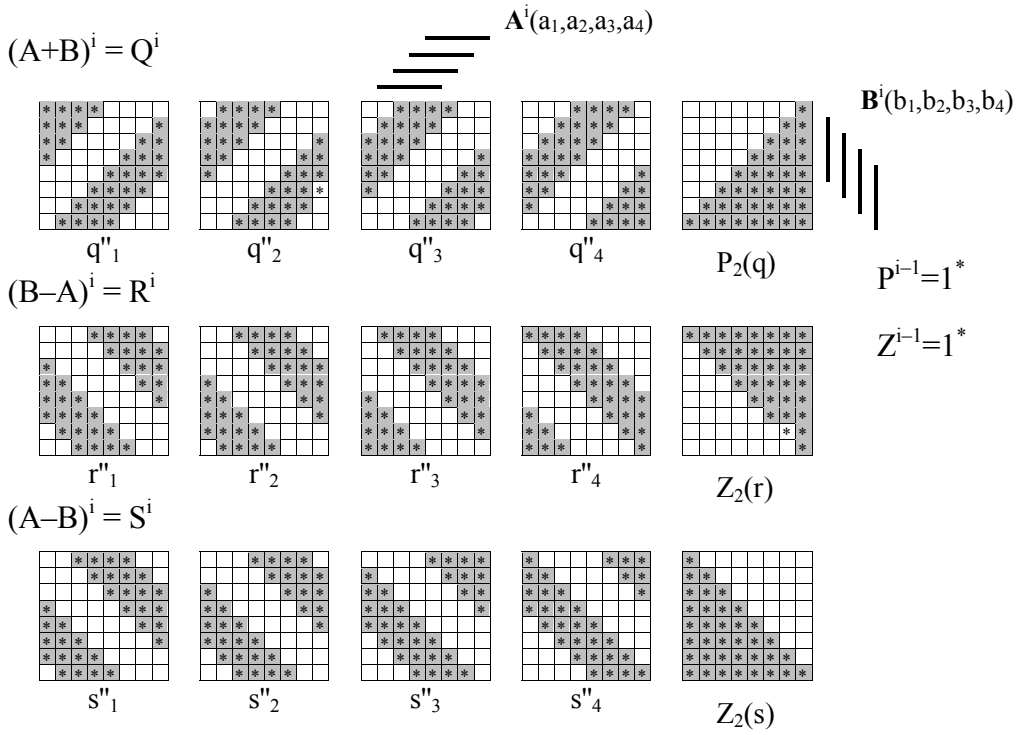


Рис. 1.23

Из геометрических образов логических функций операций суммирования и вычитания видно, что, принимая в качестве базовых фигуры сигналов $q'_1(\mathbf{AB})$ и $q'_3(\mathbf{AB})$, сигналы второй и четвертой фаз результата суммирования запишутся следующим образом: $q'_2(\mathbf{AB}) = q'_1(\mathbf{A^\bullet B^\bullet})$, $q'_4(\mathbf{AB}) = \underline{q}'_3(\mathbf{A^\bullet B^\bullet})$, а все другие сигналы фаз операций суммирования и вычитания, а также сигналы заема определяются выражениями:

$$r'_1 = q'_3(\mathbf{AB^\bullet}),\quad r'_2 = q'_1(\mathbf{A^\bullet B}),\ r'_3 = q'_1(\mathbf{AB^\bullet}),\ r'_4 = q'_3(\mathbf{A^\bullet B});$$

$$s'_1 = q'_3(\mathbf{A^\bullet B}),\quad s'_2 = q'_1(\mathbf{AB^\bullet}),\ s'_3 = q'_1(\mathbf{A^\bullet B}),\ s'_4 = q'_3(\mathbf{AB^\bullet});$$

$$q''_1 = q'_3(\mathbf{A^\bullet B^\bullet}),\quad q''_2 = q'_1(\mathbf{AB}),\ q''_3 = q'_1(\mathbf{A^\bullet B^\bullet}),\ q''_4 = q'_3(\mathbf{AB});$$

$$r''_1 = q'_1(\mathbf{A^\bullet B}),\quad r''_2 = q'_1(\mathbf{A^\bullet B^\bullet}),\ r''_3 = q'_3(\mathbf{A^\bullet B}),\ r''_4 = \underline{q}'_3(\mathbf{AB^\bullet});$$

$$s''_1 = q'_3(\mathbf{AB^\bullet}),\quad s''_2 = q'_1(\mathbf{A^\bullet B}),\ s''_3 = q'_3(\mathbf{AB^\bullet}),\ s''_4 = \underline{q}'_3(\mathbf{A^\bullet B});$$

$$Z^i(r) = P^i(q)(\mathbf{AB^\bullet}),\ Z^i(s) = P^i(q)(\mathbf{A^\bullet B}). \qquad (1.11.4)$$

Подведем итог синтеза выполнения операции суммирования и вычитания для двоичного и многофазного принципов кодирования основания системы счисления. При этом будем считать базовым вариантом операцию суммирования, а операцию вычитания будем выполнять соответствующим переключением сигналов на входных и выходных шинах устройства.

Для двоичного принципа кодирования (см. рис. 1.19, 1.20) при переходе с режима суммирования (**A+B**) на режим вычитания (**B–A**) необходимо выполнить следующие операции: код числа **B** перевести из прямого кода в обратный (поворот вокруг оси *2*), что для двоичного принципа кодирования соответствует инвертированию сигналов этого операнда, а также инвертировать выходные сигналы q'_1, q'_2, q'_3 (q''_1, q''_2, q''), которые станут совпадать с необходимыми в этом случае выходными сигналами r'_1, r'_2, r'_3 (r''_1, r''_2, r''). При этом инвертировать выходной сигнал переноса P(q) не нужно: при переводе кода числа **B** из прямого кода в обратный он сразу будет совпадать с сигналом заема Z(r).

Для выполнения операции (**A–B**) необходимо код числа **A** перевести из прямого кода в обратный (поворот вокруг оси *1*), а выходные сигналы следует инвертировать. При этом очевидно, что для любых оснований систем счисления n = 2, 4, 8, … эти операции остаются неизменными.

При многофазном принципе кодирования основания системы счисления перевод числа разряда из прямого кода (ПК) в обратный код (ОК) осуществляется на примере операнда **A** по следующему правилу: $a_m \leftrightarrow \underline{a}_m$, $a_{m-1} \leftrightarrow a_1$, $a_{m-2} \leftrightarrow a_2$, …, где m = n/2 – число фаз в коде операнда. Из этого представления видно, что при четном числе фаз в коде всегда есть такой сигнал, который при переводе кода из прямого в обратный остается неизменным.

Для нашего примера это сигнал a_2, а остальные сигналы изменяются по следующему правилу: $a_4 \leftrightarrow \underline{a}_4$, $a_3 \leftrightarrow a_1$. Переход от операции суммирования к операции вычитания при многофазном принципе кодирования имеет многочисленные варианты реализации, и один из этих вариантов полностью совпадает с двоичным принципом кодирования. Например, для операции (**B–A**), когда осуществляется соответствующее преобразование выходного кода и кода **B** (см. рис. 1.22, 1.23), происходят следующие коммутации: выход сигнала q_1' станет сигналом r_3', выход сигнала q_3' – сигналом r_1', а выход сигнала q_4' – сигналом $\underline{r}_4'$. Из представленного также полностью очевидна программа реализации операции вычитания (**A–B**), когда происходят аналогичные коммутации: выход сигнала q_1' – сигналом s_3', выход сигнала q_3' – сигналом s_1', а выход сигнала q_4' – сигналом $\underline{s}_4'$.

Причем формирование сигналов переноса и заема при любом принципе кодирования оснований систем счисления и переключении режимов работы от суммирования к вычитанию реализуется одинаковым образом и не требует дополнительных преобразований на выходной шине этого сигнала устройства.

В матричной записи ДНФ для операции суммирования с одновременной индексной записью матриц, когда вместо a_1 будем записывать только индекс 1, а вместо $\underline{a}_1$ – индекс $\underline{1}$ и т.д., логическая функция двухвходового сумматора будет представлена выражением (1.11.5).

Это выражение реализует только однократное покрытие ячеек пространства, занятого геометрическим образом ЛФ для операции суммирования, что не исключает проблемы «иголок», т.е. возможности появления коротких ложных сигналов на границах участков покрытия, в этой записи для операнда $\mathbf{B}^i$:

$$\mathbf{Q} = \left(\begin{vmatrix} 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 & 3 & 2 \\ 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 & 3 \\ 3 & 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 \\ 4 & 3 & 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} \end{vmatrix}_{\mathbf{A}^i} \underline{P}^{i-1} \vee \begin{vmatrix} 4 & 3 & 2 & 1 & 4 & 3 & 2 & 1 \\ 1 & 4 & 3 & 2 & 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 & 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 & 3 & 2 & 1 & 4 \end{vmatrix}_{\mathbf{A}^i} P^{i-1} \right) \begin{vmatrix} \underline{4} & \underline{1} \\ 1 & \underline{2} \\ 2 & \underline{3} \\ 3 & \underline{4} \\ 4 & 1 \\ \underline{1} & 2 \\ \underline{2} & 3 \\ \underline{3} & 4 \end{vmatrix}_{\mathbf{B}^i} . \qquad (1.11.5)$$

Для исключения «иголок» необходимо в (1.11.5) все вторые либо все первые сомножители в матрицах операндов циклически изменить: для операнда **A** в последовательности $1 \to 2 \to 3 \to 4 \to \underline{1} \to \underline{2} \to \underline{3} \to \underline{4} \to 1$, а для операнда **B** – в обратной последовательности. Первый такой сдвиг для операнда **A** даст следующую запись ЛФ:

$$\mathbf{Q}^i = \left(\begin{vmatrix} 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 & 3\,2 & 2\,1 \\ 2\,1 & 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 & 3\,2 \\ 3\,2 & 2\,1 & 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 \\ 4\,3 & 3\,2 & 2\,1 & 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 \end{vmatrix}_{\mathbf{A}^i} \underline{P}^{i-1} \vee \begin{vmatrix} \underline{4}\,\underline{2} \\ 1\,\underline{3} \\ 2\,\underline{4} \\ 3\,1 \\ 4\,2 \\ \underline{1}\,3 \\ \underline{2}\,4 \\ \underline{3}\,1 \end{vmatrix}_{\mathbf{B}^i}$$

$$\vee \begin{vmatrix} \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 & 3\,2 & 2\,1 & 1\,\underline{4} \\ 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 & 3\,2 & 2\,1 \\ 2\,1 & 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 & 3\,2 \\ 3\,2 & 2\,1 & 1\,\underline{4} & \underline{4}\,\underline{3} & \underline{3}\,\underline{2} & \underline{2}\,\underline{1} & \underline{1}\,4 & 4\,3 \end{vmatrix}_{\mathbf{A}^i} P^{i-1}) \qquad (1.11.6)$$

Следующий сдвиг в этой зависимости приведет эти выражения к виду

$$\mathbf{Q}^i = \left(\begin{vmatrix} 1\,\underline{3} & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & \underline{1}\,3 & 4\,2 & 3\,1 & 2\,\underline{4} \\ 2\,\underline{4} & 1\,\underline{3} & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & \underline{1}\,3 & 4\,2 & 3\,1 \\ 3\,1 & 2\,\underline{4} & 1\,\underline{3} & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & \underline{1}\,3 & 4\,2 \\ 4\,2 & 3\,1 & 2\,\underline{4} & 1\,\underline{3} & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & \underline{1}\,3 \end{vmatrix}_{\mathbf{A}^i} \underline{P}^{i-1} \vee \begin{vmatrix} \underline{4}\,\underline{3} \\ 1\,\underline{4} \\ 2\,1 \\ 3\,2 \\ 4\,3 \\ \underline{1}\,4 \\ \underline{2}\,1 \\ \underline{3}\,2 \end{vmatrix}_{\mathbf{B}^i}$$

$$\vee \begin{vmatrix} \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & 1\,\underline{3} & 4\,2 & 3\,1 & 2\,\underline{4} & 1\,\underline{3} \\ 1\,3 & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & 1\,\underline{3} & 4\,2 & 3\,1 & 2\,\underline{4} \\ 2\,4 & 1\,3 & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & 1\,\underline{3} & 4\,2 & 3\,1 \\ 3\,1 & 2\,4 & 1\,3 & \underline{4}\,\underline{2} & \underline{3}\,\underline{1} & \underline{2}\,4 & 1\,\underline{3} & 4\,2 \end{vmatrix}_{\mathbf{A}^i} P^{i-1}) \qquad (1.11.7)$$

Выражение (1.11.5) определяет покрытие геометрических образов ЛФ сигналов $q_1 - q_4$ непересекающимися прямоугольниками размерами 1×4, а выражения (1.11.6), (1.11.7) – пересекающимися прямоугольниками соответственно размерами 2×3 и 3×2. Это показано на примере четвертой фазы q_4 для $\underline{P}^{i-1}$ на рис. 1.24, где для большей наглядности приведены только два прямоугольника из восьми.

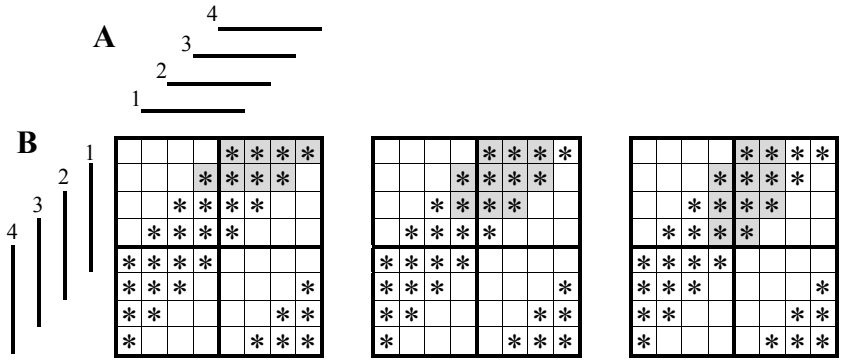


Рис. 1.24

На рис. 1.25 приведен геометрический образ сигнала первой фазы q_1 результата суммирования в трехмерном цифровом пространстве сложных аргументов **A**, **B**, **Q**. Все геометрические образы остальных сигналов фаз для операций суммирования и вычитания могут быть получены, исходя из образа сигнала q_1 соответствующим мысленным поворотом относительно осей симметрии пространства, а также параллельным перемещением этой фигуры. Это возможно выполнить, поскольку здесь используется циклический принцип кодирования сигналов операндов. Следовательно, при многофазном принципе кодирования число базовых фигур может быть уменьшено здесь до одной, например q_1.

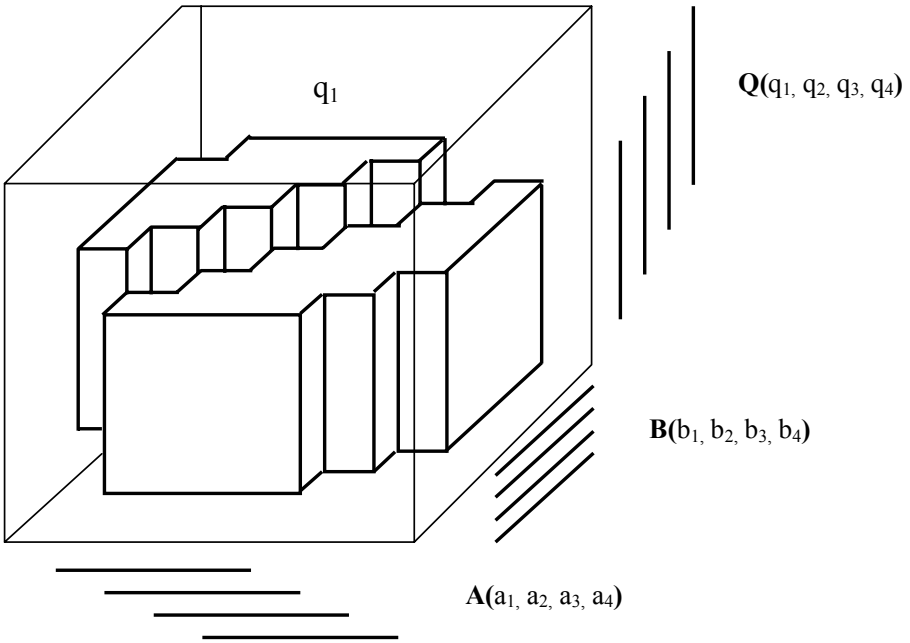


Рис. 1.25

Традиционно в вычислительной технике используется двоичный принцип кодирования оснований систем счисления, а многофазный является естественным для устройств преобразовательной техники и электропривода. При управлении такими устройствами от ЦВМ необходимо рассмотреть возможность их работы не только в многофазных кодах [5–11], но и для применения смешанного кодирования, если один из операндов представлен, например, в двоичном коде, а другой – в многофазном.

Если результат операции суммирования или вычитания реализуется в многофазном коде, что обычно и требуется в изучаемых нами разделах техники, то ЛФ этих операций по аналогии с (1.11.5) и задании, например, операнда $\mathbf{A}^i$ в многофазном коде, а операнда $\mathbf{B}^i$ – в двоичном, будут следующими:

$$\mathbf{Q}^i = \left(\begin{vmatrix} 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 & 3 & 2 \\ 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 & 3 \\ 3 & 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} & 4 \\ 4 & 3 & 2 & 1 & \underline{4} & \underline{3} & \underline{2} & \underline{1} \end{vmatrix}_{\mathbf{A}^i}^{\underline{P}^{i-1}} \vee \begin{vmatrix} 4 & 3 & 2 & 1 & 4 & 3 & 2 & 1 \\ 1 & 4 & 3 & 2 & 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 & 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 & 3 & 2 & 1 & 4 \end{vmatrix}_{\mathbf{A}^i}^{P^{i-1}} \right) \begin{vmatrix} \underline{3} & \underline{2} & \underline{1} \\ \underline{3} & \underline{2} & 1 \\ \underline{3} & 2 & \underline{1} \\ \underline{3} & 2 & 1 \\ 3 & \underline{2} & \underline{1} \\ 3 & \underline{2} & 1 \\ 3 & 2 & \underline{1} \\ 3 & 2 & 1 \end{vmatrix}_{\mathbf{B}^i}. \qquad (1.11.8)$$

При этом геометрические образы сигналов операций суммирования и вычитания $\mathbf{Q}^i$, $\mathbf{R}^i$, $\mathbf{S}^i$ полностью определяются рис. 1.22 и 1.23.

Необходимо отметить, что для реализации каждой из ЛФ в (1.11.9) можно использовать интегральную схему мультиплексора, которая представляет собой многовходовый логический элемент с одним выходом, где входы подразделены на информационные и управляющие. При подключении к управляющим входам двоичных сигналов операнда $\mathbf{B}(b_1,b_2,b_3)$ и переноса P^{i-1}, а к информационным входам в определенной последовательности – многофазных сигналов операнда $\mathbf{A}$ к выходу мультиплексора типа (1 из 16) подключается один из его информационных входов, и, следовательно, на выходе будет в соответствии с (1.11.8) сформирован сигнал одной из фаз.

При параллельном формировании сигналов всех фаз необходимо использовать такое же количество мультиплексоров, а при последовательном определении сигналов фаз потребуется одна схема мультиплексора.

1.12. Пример анализа неизбыточных кодов позиционной системы счисления

После краткого обзора синтеза устройств машинной арифметики в позиционных системах счисления возникает необходимость вернуться к началу гл. 1, где сделана попытка систематизации кодов, представляющих позиционную систему счисления. Это было выполнено для того, чтобы присвоить неизбыточному коду конкретного основания системы счисления свой порядковый номер. При этом номер один оставлен за двоичным принципом кодирования, который будем называть основным двоичным кодом. Очевидно, что число неизбыточных кодов для конкретной системы счисления n конечно и равно n!.

Примем для любых типов кодов позиционной системы счисления неизменность кодового слова для цифры **0** (**0** $\leftrightarrow \underline{a}_1\, \underline{a}_2 \ldots \underline{a}_i$.). В этом случае число кодов будет уменьшено до значения (n – 1)!. Дальнейшим решением, что следует из предыдущих рассмотрений задач синтеза устройств машинной арифметики, является использование только четных оснований систем счисления. Причем в коде этих систем счисления обязательное условие – наличие сигнала, который является непрерывным множеством цифр второй половины основания системы счисления.

При этом оставшиеся (i – 1) комбинации сигналов кода могут быть одинаковыми на первой и второй половине цифр основания либо иметь симметричное расположение комбинаций на второй половине относительно первой половины цифр основания системы счисления.

Представленные выше ограничения на использование кодов в позиционных системах счисления позволяют весьма просто, например в суммирующем либо вычитающем устройствах, формировать сигналы переноса и заёма непосредственно по выходным и входным сигналам старших разрядов кода. Это можно реализовать, не используя отдельного полного логического блока для формирования сигнала переноса. Тогда каждому виду симметрии оставшихся (i – 1) кодовых слов будет соответствовать (n/2 – 1)! кодов.

Для пояснения сказанного обратимся к основанию системы счисления, например n=8.

На рис. 1.26, *а* под №1 представлен двоичный принцип кодирования этого основания, где сигнал a_3 = **4** $\vee$ **5** $\vee$ **6** $\vee$ **7** и будет оставаться неизменным при любых принципах кодирования этого основания системы счисления. Цифра **0** определяется логической зависимостью **0** = $\underline{a}_1\underline{a}_2\, \underline{a}_3$, а сигнал цифры **4** будет определяться при других принципах кодирования выбранным соответствием кодов на первой и второй половине цифр основания. Так же в зависимости от этого выбора будут различным образом кодироваться цифры **1**, **5**; **2**, **6**; **3**, **7**. Если в исходном двоичном коде для цифр **1**, **5** принять обозначение состояния сигналов кода $A \leftrightarrow a_1\underline{a}_2$; для цифр **2**, **6** – $B \leftrightarrow \underline{a}_1 a_2$; для цифр **3**, **7** – $C \leftrightarrow a_1 a_2$, то в соответствии с выражением (1.2.2) число перестановок, определяющих число кодов, равно шести: *ABC, ACB, BAC, BCA, CAB, CBA*.

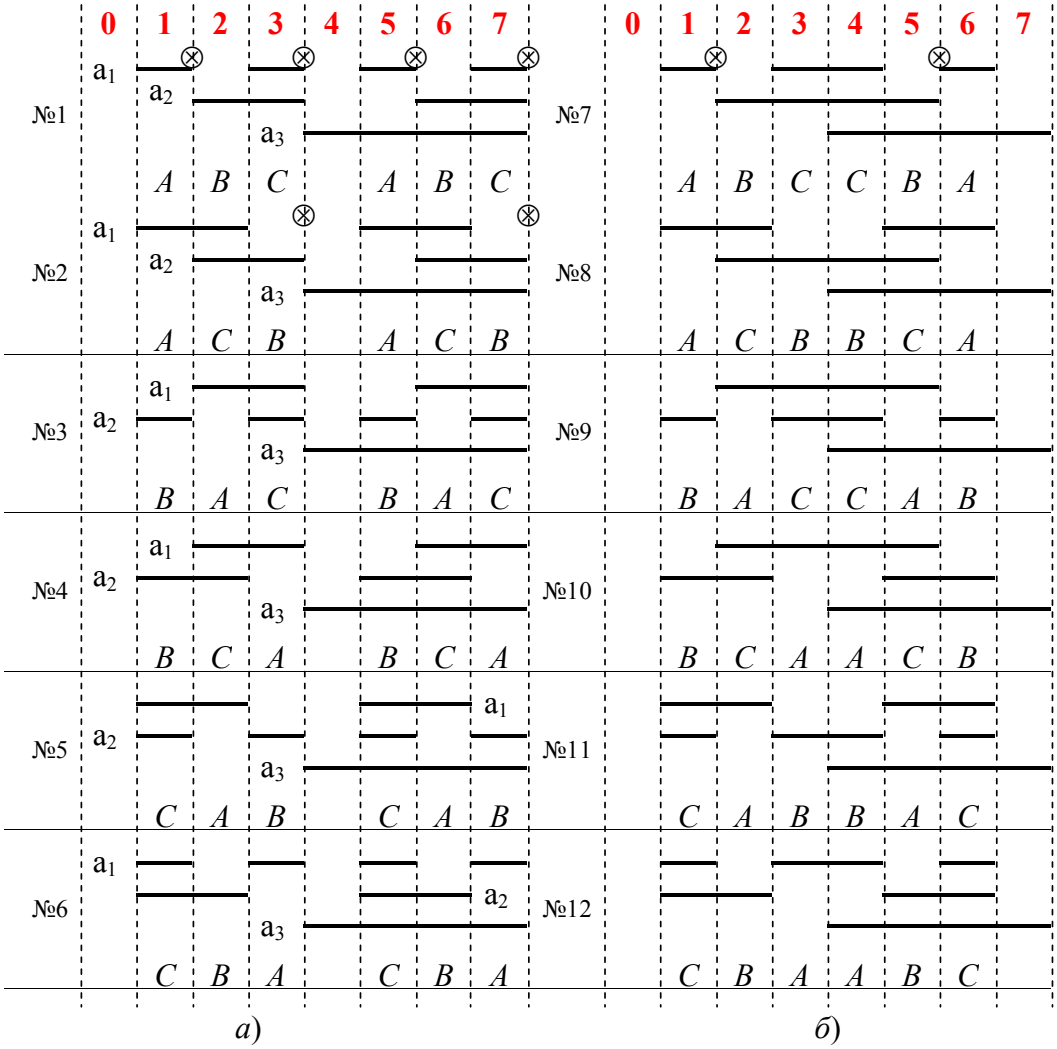


Рис. 1.26

Все эти коды приведены под своими номерами на рис. 1.26, *а*, а на рис. 1.26, *б* изображены коды для симметричного расположения (i – 1) сигналов на второй половине цифр основания системы счисления.

Таким образом, мы получили двенадцать кодов, которые, как все возможные коды, могут быть использованы для выполнения любых арифметических операций. Причем пары кодов №1, №3; №2, №4; №5, №6, а также №7, №9; №8, №10; №11, №12 имеют одинаковые возможности по выполнению арифметических операций. Поэтому будем рассматривать только первые из них: №1, №2, №5, №7, №8, №11. Из этих шести кодов только два имеют официальное название: №1 – двоичный код, №8 – код Грея.

Поскольку позиционные системы счисления в системах электропривода используются не только в арифметических блоках, но и, например, в цифровых преобразователях угла, то к ним предъявляются дополнительные требования.

В этих устройствах возникает неопределенность считывания информации за счет того, что переход от одного числа к другому сопровождается переменой сигналов кода в нескольких разрядах.

На рис. 1.26 переход от одной цифры к другой, когда это происходит изменением сигналов более чем в одном разряде, отмечен знаком ⊗. При этом коды №5, №11, которые не имеют перед другими кодами каких-либо преимуществ по выполнению арифметических операций и к тому же обладают большим числом нежелательных переходов, исключаются из рассмотрения.

Этот недостаток можно ликвидировать, применив такой код, при котором переход от одного числа к другому сопровождается переменой кода только в одном сигнале. Этому требованию отвечает только код Грея. Достоинством этого кода также является простота перевода кода из прямого в обратный, что выполняется инвертированием только одного сигнала a_3 и упрощает выполнение арифметических операций суммирования и вычитания в блоке каждого разряда.

Общеизвестно, что большим достижением индийской математики было создание системы счисления с нулем. Основным здесь было предложение или, вернее сказать, изобретение обозначать особым знаком (нуль) соответствующий (пустой) разряд. Разряда нет, но само его отсутствие было представлено как наличие «ничего», хотя его нет, но оно имеет обозначение. Нетривиальность этого изобретения, по словам А.Д. Александрова, блестяще выражена в афоризме Дирака, высказанного им в связи с «теорией дырок» (теория позитрона): «Ничто, помещенное во что-нибудь, вполне эквивалентно чему-нибудь, помещенному в ничто».

В нашем случае есть не только обозначение, есть для каждого разряда реально существующий, например, кубик для цифры **0** в многомерном цифровом пространстве. Однако при выполнении каких-либо арифметических операций или других функциональных действий с цифровыми данными по общепринятым законам математики, результат которых затем может отражаться в многомерном цифровом пространстве, цифра нуль, как и положено, есть «ничто».

Поскольку геометрия исходит из наглядных представлений, и в этом ее главная сила, то «она позволяет ухватить в одном представлении то, к чему алгебра и анализ приходят последовательными шагами». Следовательно, там, «где удается провести геометрическую точку зрения и геометрический метод, всегда достигается успех настолько существенный, насколько геометрический взгляд соответствует сущности предмета». В этом нам представится возможность убедиться в последующих разделах книги.

1.13. Синтез суммирующих и вычитающих устройств в нетрадиционных двоичных кодах

Использование больших оснований систем счисления, кратных двум, при двоичном неизбыточном принципе их кодирования, когда это основной двоичный код, позволяет считать их совершенными для арифметических операций суммирования и вычитания. Это означает, что результат этих операций для каждого из разрядов, составляющих основание системы счисления, определяется, с учетом входного сигнала переноса или заема, сигналами операндов этого конкретного разряда основания, а также всех младших разрядов операндов в этом основании.

При всех других двоичных принципах кодирования подобных оснований систем счисления результат аналогичных арифметических операций в каждом из разрядов основания системы счисления будет определяться сигналами всех без исключения разрядов этого основания системы счисления.

Несмотря на то, что основной двоичный код является лучшим для выполнения таких арифметических операций и ему нет в этом плане достойной альтернативы, если не выдвигаются какие-либо иные ограничения на его применение, а они нередко выдвигаются, поэтому представляет интерес рассмотреть выполнение суммирующих и вычитающих устройств в иных двоичных кодах. Принципиально это могут быть любые двоичные коды, которые при представлении цифр оснований систем счисления отличаются от основного двоичного кода перестановками его кодовых комбинаций. Эти кодовые комбинации наиболее удобно представлять суммой весовых W_i (1, 2, 4, 8 …) значений их двоичных разрядов, которые для краткости будем называть весовыми значениями. Для основного двоичного кода его весовые значения W_a совпадают со значением цифр оснований систем счисления, и, следовательно, в нем каждую кодовую комбинацию можно представлять только цифрами основания системы счисления, что мы выполняли до сих пор. При других двоичных принципах кодирования оснований систем счисления необходимо для каждой цифры этого основания представлять и ее весовое значение в сигналах разрядов основного двоичного кода.

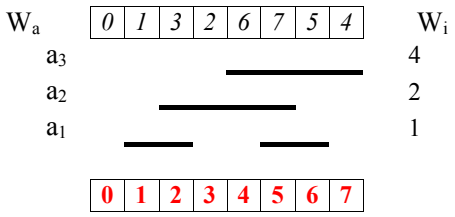


Рис. 1.27

Будем в обозначении цифр оснований систем счисления использовать шрифт красного цвета, а для весовых значений кодовых комбинаций – шрифт черного цвета и курсив. В соответствии с этим на рис. 1.27 в качестве одного из примеров такого кода представлен код Грея для основания системы счисления

n = 8, где одновременно с сигналами разрядов кода (a_1, a_2, a_3) приведены цифровые **0–7** и весовые *0–7* значения его кодовых комбинаций.

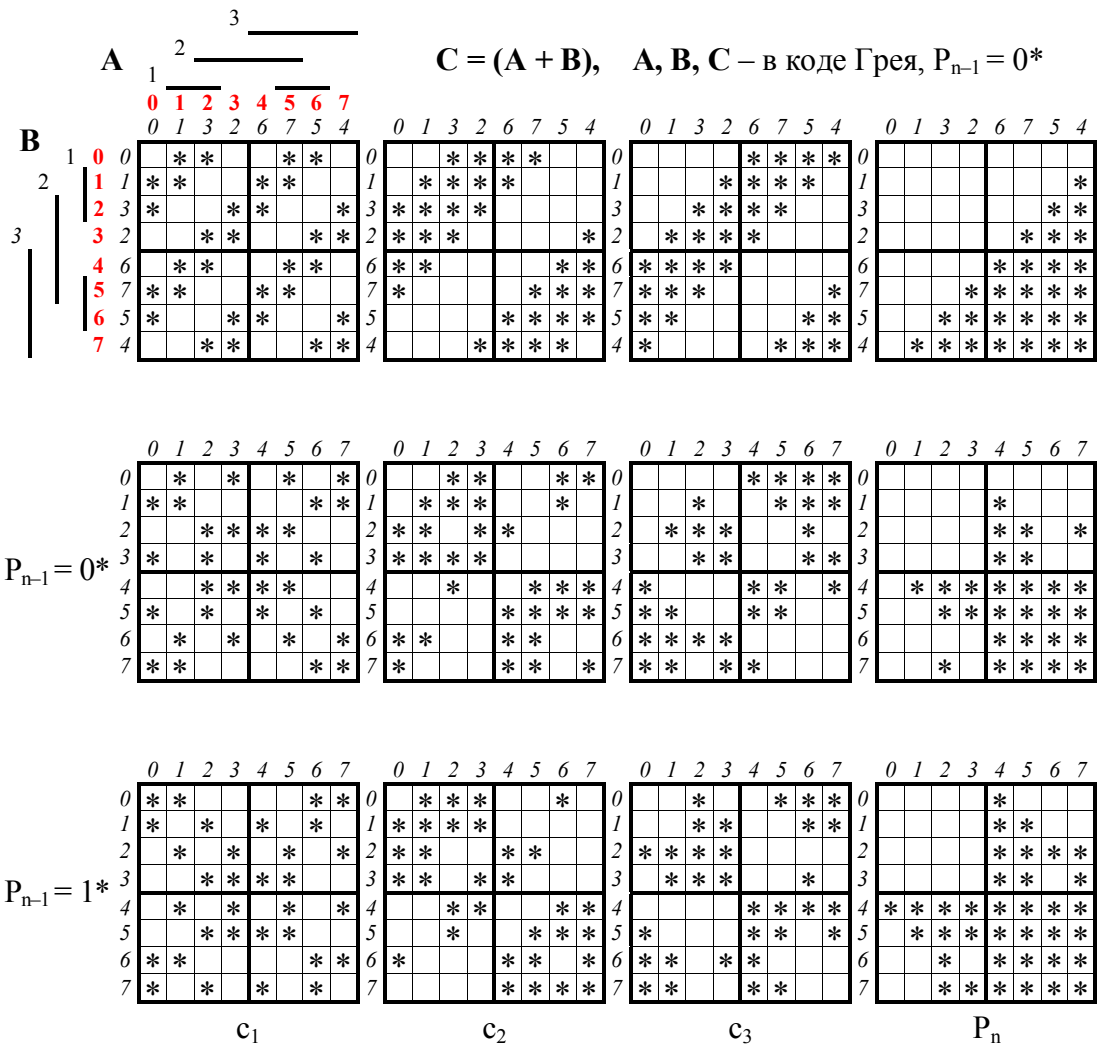


Рис. 1.28

Цифровые значения кодовых комбинаций всегда используются для построения геометрических образов сигналов разрядов на основании соответствующих арифметических таблиц. На основании такой таблицы суммирования на рис. 1.28 построены геометрические образы выходных сигналов сумматора **C = (A + B),** когда во входных и выходных сигналах применяется код Грея, при входных сигналах переносов P_{n-1} = 0* и P_{n-1} = 1*.

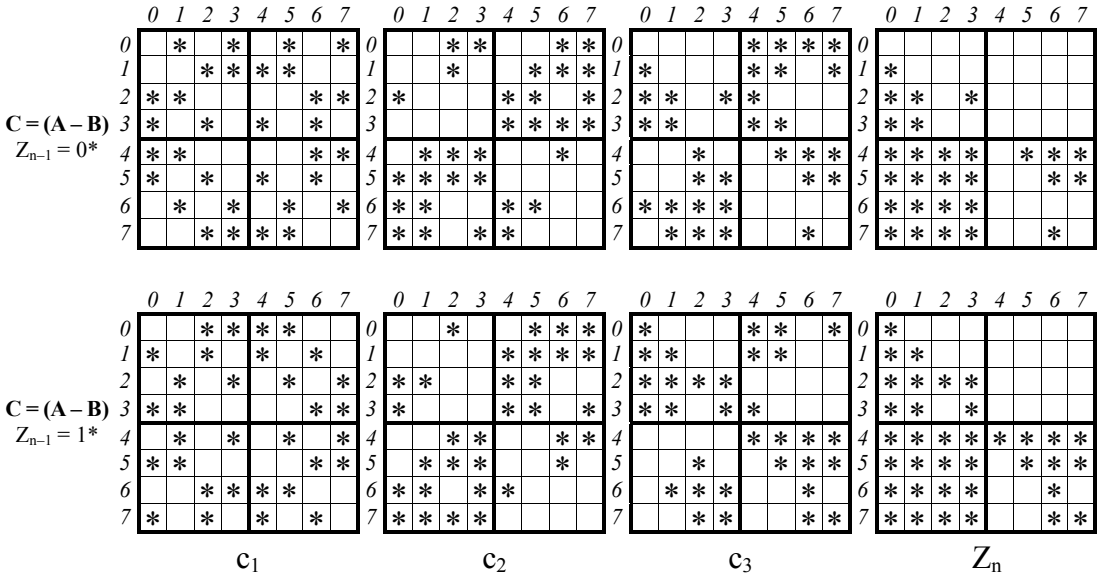


Рис. 1.29

Для входного сигнала переноса P_{n-1} = 0* построение геометрических образов выходных сигналов c_1, c_2, c_3, P_n первоначально выполнено в цифровых значениях координат входных сигналов **A, B** (первый ряд геометрических фигур этих сигналов на рис. 1.28), а в дальнейшем представлены изменения этих фигур при последовательном переходе к весовым значениям координат основного двоичного кода.

На этом же рисунке в четвертом ряду фигур изображены геометрические фигуры выходных сигналов сумматора при P_{n-1} = 1*, минуя промежуточные преобразования, сразу в координатах основного двоичного кода.

Для режима вычитания **C = (A – B)** аналогичные геометрические фигуры приведены, минуя представление промежуточных геометрических преобразований, на рис. 1.29 соответственно при Z_{n-1} = 0*, Z_{n-1} = 1*.

Применение в суммирующих блоках иных возможных сочетаний кодов представлено, также минуя представление промежуточных геометрических преобразований, в двух вариантах: на рис. 1.30, когда операнды **A, B** – в коде Грея, а выходной сигнал **C = (A ± B)** – в основном двоичном коде; на рис. 1.31, когда операнд **A** – в коде Грея, а операнд **B** и выходной сигнал **C = (A ± B)** – в основном двоичном коде.

Дальнейшие этапы синтеза логических функций сумматоров C = (A ± B), определяющие их схемотехнические решения, будут выполняться на основании покрытия выходных сигналов их геометрических образов. Это покрытие должно выполняться в соответствии с требованием быстродействия сумматоров, что определяет необходимый уровень схемного решения. Очевидно, предельное быстродействие будет получено в двухуровневой схеме сумматора.

A, B – в коде Грея, **C** – в основном двоичном коде

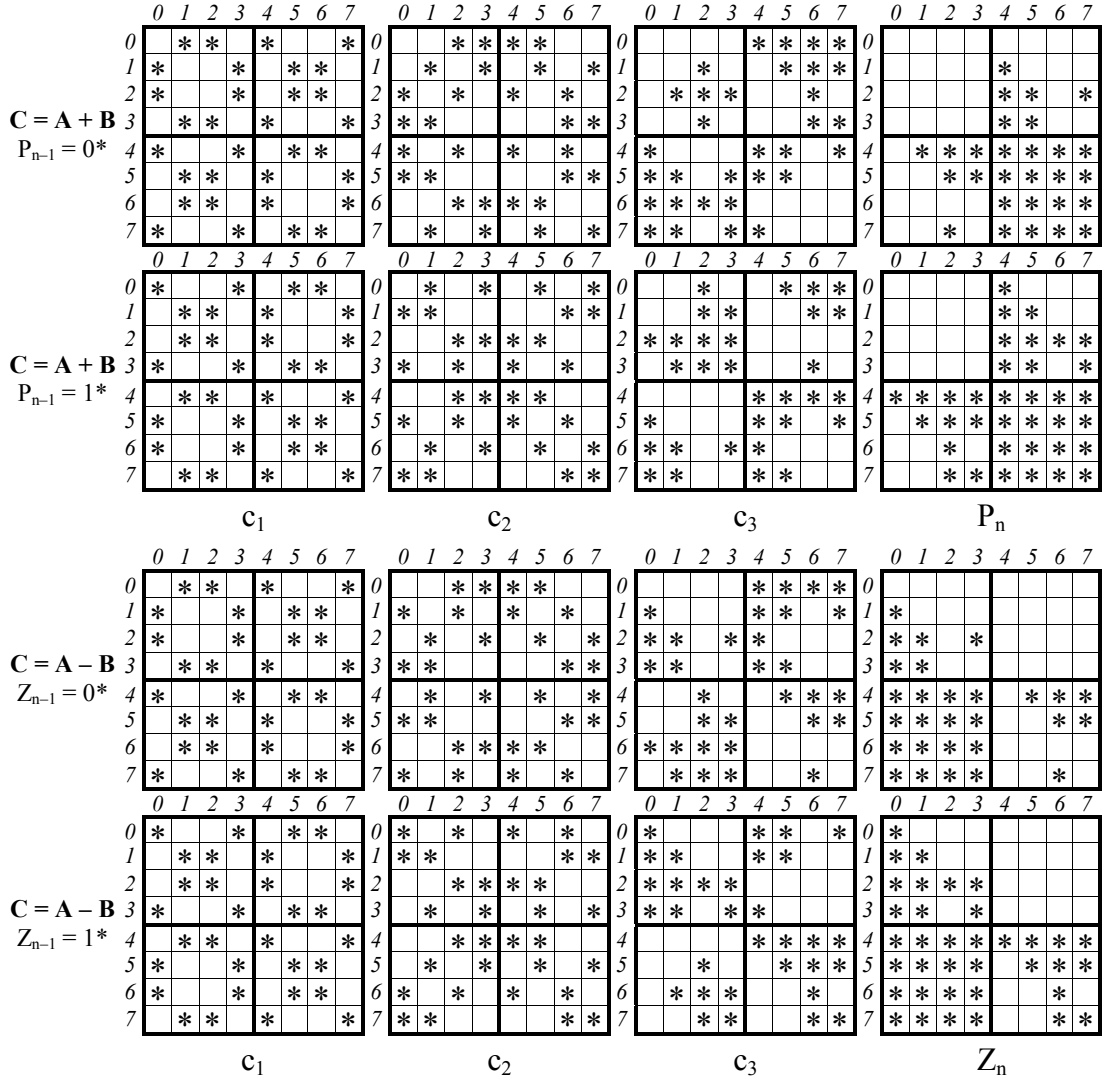


Рис. 1.30

Первоначально в качестве примера многоуровневого синтеза сумматора **C = (A ± B)** рассмотрим такой вариант схемного решения, когда все входные сигналы и результат операции представляются в коде Грея. Для этого обратимся к геометрическим образам (см. рис. 1.28, 1.29) его выходных сигналов c_1, c_2, c_3, P_n, Z_n и рассмотрим их подмножества в каждой четверти пространства координат основного двоичного кода.

A – в коде Грея; **B, C** – в основном двоичном коде

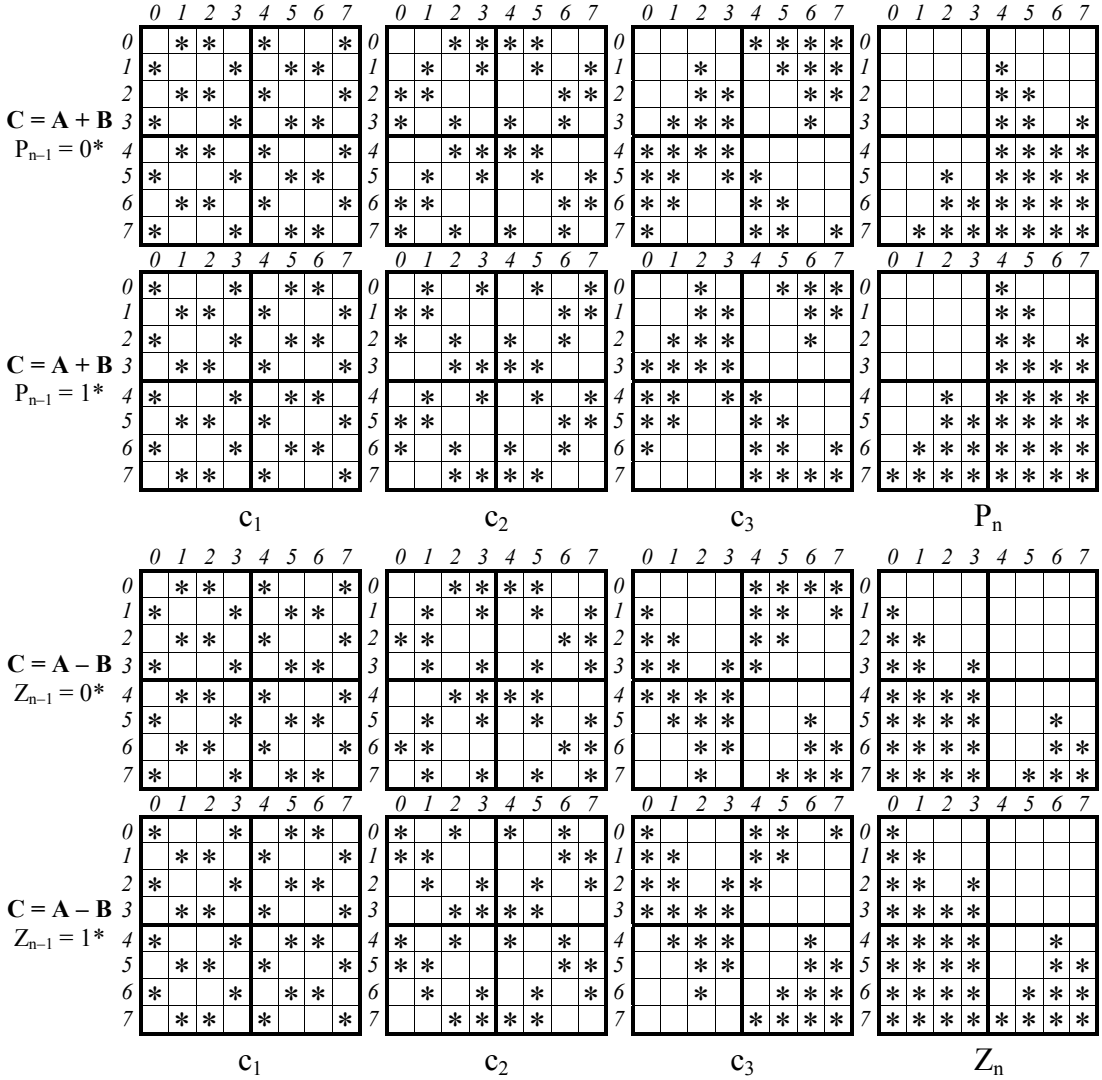
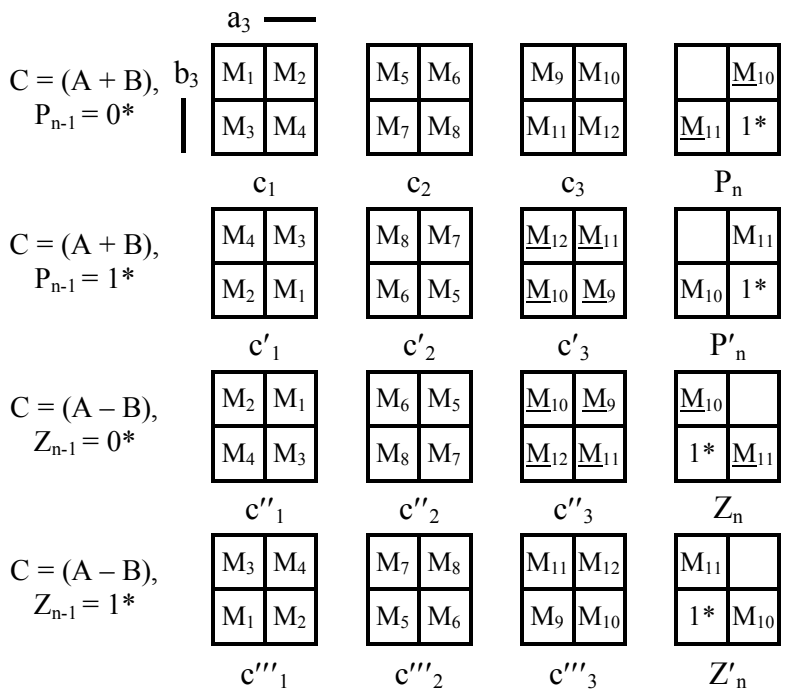
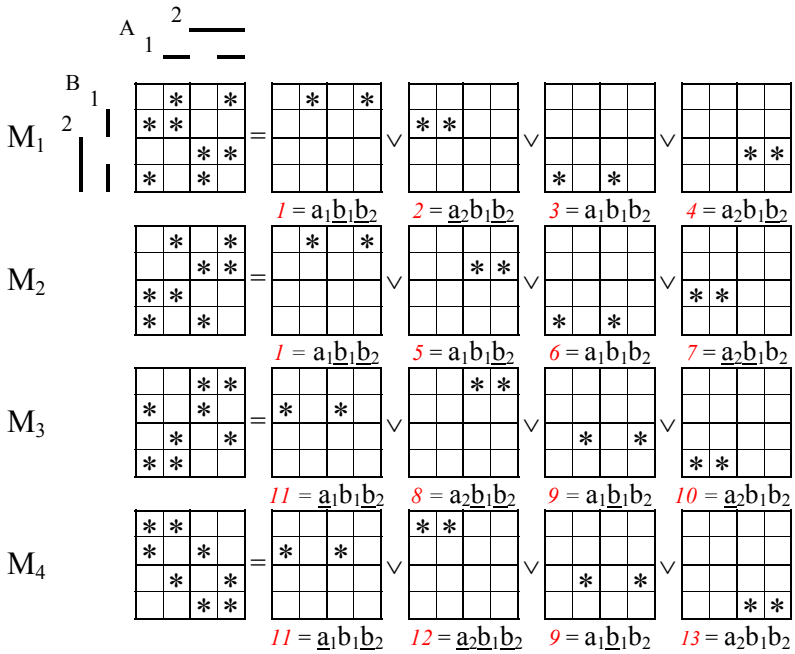


Рис. 1.31

На рис. 1.32 изображены эти подмножества $M_1 - M_{12}$, а на рис. 1.32, *а – б* приведены более мелкие подмножества, их составляющие, которые представлены соответствующими логическими выражениями и пронумерованы курсивом цифрами красного цвета.

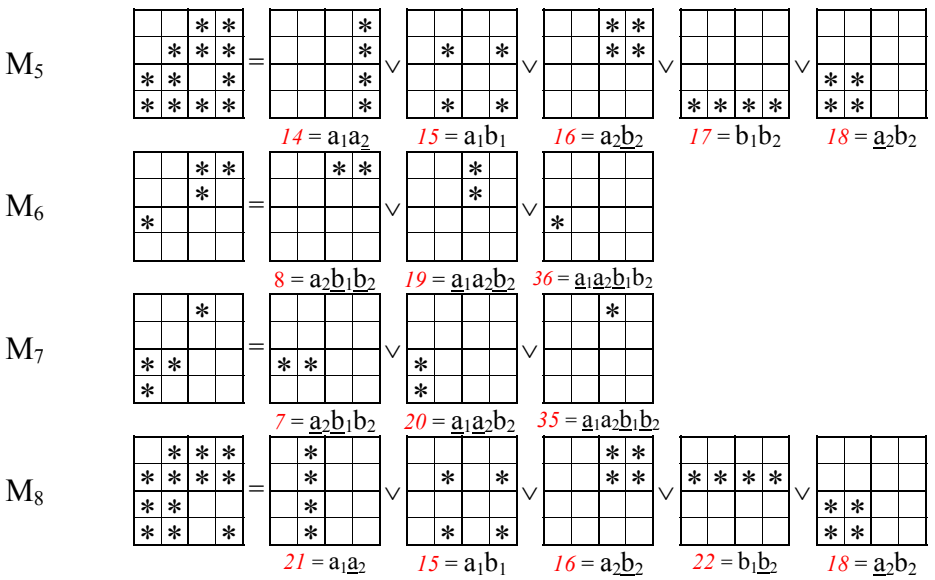


а)

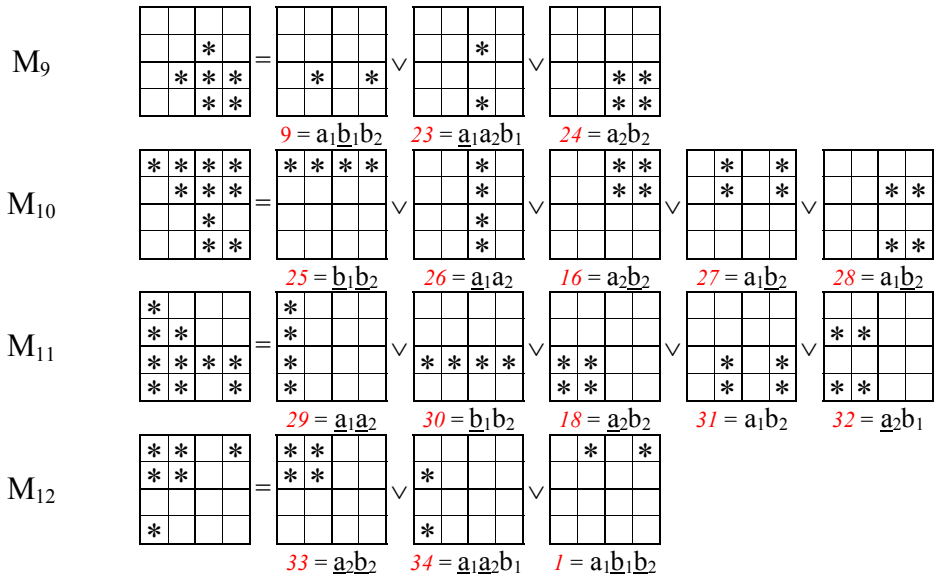


б)

Рис. 1.32 (начало)



в)



г)

Рис. 1.32 (продолжение)

В соответствии с рис. 1.32, *а* – *г*, учитывая, что $M_5 \supset M_6$, $M_5 \supset M_7$, $M_8 \supset M_6$, $M_8 \supset M_7$, можно записать логические выражения, определяющие построение сумматора **C = (A + B)** при $P_{n-1} = 0$*:

$c_1 = M_1\underline{a}_3\underline{b}_3 \vee M_2a_3\underline{b}_3 \vee M_3\underline{a}_3b_3 \vee M_4a_3b_3,$
$c_2 = M_5\underline{a}_3\underline{b}_3 \vee M_6a_3$ (или $M_6\underline{b}_3$)$\vee M_7b_3$ (или $M_7\underline{a}_3$)$\vee \vee M_8a_3b_3,$
$c_3 = M_9\underline{a}_3\underline{b}_3 \vee M_{10}a_3\underline{b}_3 \vee M_{11}\underline{a}_3b_3 \vee M_{12}a_3b_3,$
$P_n = \underline{M}_{10}a_3 \vee \underline{M}_{11}b_3 \vee a_3b_3\,.$ (1.13.1)

Принимая в качестве базового варианта построения этого блока логические выражения (1.13.1), остальные режимы работы сумматора могут быть получены путем соответствующих поворотов относительно осей симметрии двухмерного пространства их геометрических образов в координатах основного двоичного кода:

$c'_1 = c_1(\mathbf{A}^\bullet\,\mathbf{B}^\bullet)$, $c'_2 = c_2\,(\mathbf{A}^\bullet\,\mathbf{B}^\bullet)$, $c'_3 = \underline{c}_3\,(\mathbf{A}^\bullet\,\mathbf{B}^\bullet)$, $P'_n = \underline{P}_n\,(\mathbf{BA})$;
$c''_1 = c_1(\mathbf{A}^\bullet\,\mathbf{B})$, $c''_2 = c_2\,(\mathbf{A}^\bullet\,\mathbf{B})$, $c''_3 = \underline{c}_3\,(\mathbf{A}^\bullet\,\mathbf{B})$, $Z_n = P_n\,(\mathbf{A}^\bullet\,\mathbf{B})$;
$c_1 = c_1(\mathbf{A}\,\mathbf{B}^\bullet)$, $c'''_2 = c_2\,(\mathbf{A}\,\mathbf{B}^\bullet)$, $c'''_3 = \underline{c}_3\,(\mathbf{A}\,\mathbf{B}^\bullet)$, $Z'_n = \underline{P}_n\,(\mathbf{A}\,\mathbf{B}^\bullet)$. (1.13.2)

Выходные сигналы сумматора соответственно для каждого из этих режимов, отличных от базового, подчиняются соответствующим двум поворотам для сигналов $c'_1 - c'_3$ либо одному повороту для сигналов $c''_1 - c''_3$, $c'''_1 - c'''_3$, которые совершаются относительно осей симметрии этого пространства. Это правило не относится к сигналу переноса P'_n, где требуется поменять местами входы операндов **A** и **B**. Поэтому базовый блок для сигнала переноса P_n должен будет выполняться автономным со своими входами для операндов **A** и **B**, а также своими схемами формирования подмножеств M_{10}, M_{11} внутри этого автономного блока.

Можно предложить иной вариант формирования сигналов переноса и заема, который не требует использования такого автономного блока, а именно, когда сигнал переноса формируется по сигналам старших разрядов операндов a_3, b_3 и аналогичного выходного сигнала c_3 сумматора по логическому выражению

$$P_n = a_3\,b_3 \vee \underline{a}_3\,c_3 \vee \underline{b}_3\,c_3.$$

В этом варианте суммирующего устройства при выполнении необходимых переключений на его входных шинах, которые изменяют его режимы работы с целью получения выходных сигналов $c'_1 - c'_3$, $c''_1 - c''_3$, $c'''_1 - c'''_3$, он будет автоматически сопровождаться получением необходимого сигнала переноса либо заема.

Для синтеза суммирующего блока максимального быстродействия необходимо реализовать покрытие геометрических образов каждого отдельно сигналов c_1, c_2 ,c_3, начиная «сверху вниз», т.е. рассматривая старшие ячейки многомерного пространства, в которых определяются первоначально элементарные ячейки, общие для всех разрядов, если такие имеются. Затем рассматриваются ячейки более младших разрядов и т.д. вплоть до полного покрытия всех геометрических образов.

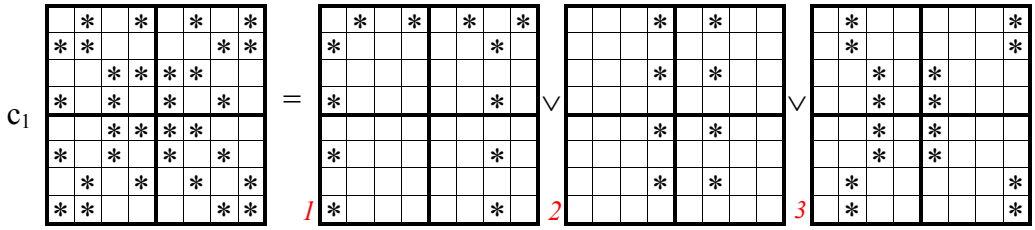


Рис. 1.33

$$1 = a_1\,\underline{b_1}\,\underline{b_2}\,\underline{b_3} \vee \underline{a_1}\,\underline{a_2}\,b_1\,\underline{a_3} \vee \underline{a_1}\,a_2\,b_1\,a_3;\quad 2 = a_1\,a_2\,\underline{b_1}\,\underline{a_3} \vee a_1\,\underline{a_2}\,\underline{b_1}\,a_3;$$

$$3 = a_1\,\underline{a_2}\,\underline{a_3}\,\underline{b_3}\,\underline{b_2} \vee \underline{a_1}\,a_2\,\underline{a_3}\,\underline{b_3}\,b_2 \vee a_1\,a_2\,a_3\,\underline{b_3}\,\underline{b_2} \vee \underline{a_1}\,\underline{a_2}\,\underline{a_3}\,\underline{b_3}\,b_2 \vee$$

$$\vee\, \underline{a_1}\,a_2\,\underline{a_3}\,b_3\,\underline{b_2} \vee a_1\,\underline{a_2}\,\underline{a_3}\,b_3\,b_2 \vee \underline{a_1}\,\underline{a_2}\,a_3\,b_3\,\underline{b_2} \vee a_1\,a_2\,a_3\,b_3\,b_2. \qquad (1.13.3)$$

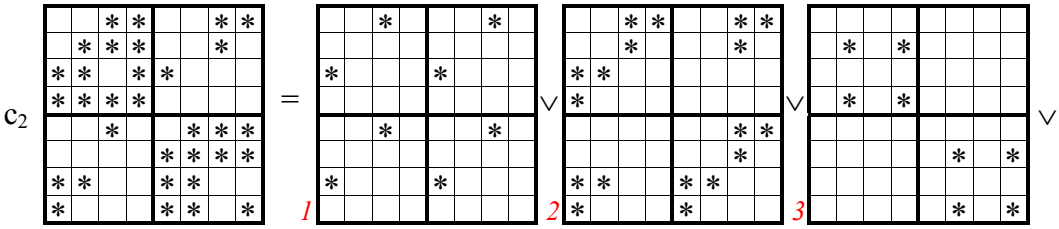
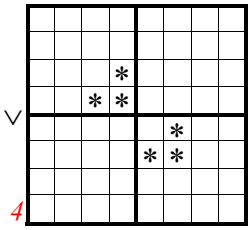


Рис. 1.34



$$1 = \underline{a_1}\,a_2\,\underline{b_1}\,\underline{b_2} \vee \underline{a_1}\,\underline{a_2}\,\underline{b_1}\,b_2\,;\quad 3 = a_1\,b_1\,\underline{a_3}\,\underline{b_3} \vee a_1\,b_1\,a_3\,b_2\,;$$

$$2 = a_2\,\underline{b_1}\,\underline{b_2}\,\underline{b_3} \vee a_2\,\underline{b_1}\,\underline{b_2}\,a_3 \vee \underline{a_1}\,a_2\,\underline{b_2}\,\underline{b_3} \vee \underline{a_1}\,a_2\,\underline{b_2}\,a_3 \vee \underline{a_2}\,\underline{b_1}\,b_2\,\underline{a_3} \vee$$

$$\vee\, \underline{a_2}\,\underline{b_1}\,b_2\,b_3 \vee \underline{a_1}\,\underline{a_2}\,b_2\,\underline{a_3} \vee \underline{a_1}\,\underline{a_2}\,b_2\,b_3\,;$$

$$4 = a_1 a_2\,b_2\,\underline{a_3}\,\underline{b_3} \vee a_2\,b_1\,b_2\,\underline{a_3}\,\underline{b_3} \vee a_1\underline{a_2}\,\underline{b_2}\,a_3\,b_3 \vee \underline{a_2}\,b_1\,\underline{b_2}\,a_3\,b_3. \qquad (1.13.4)$$

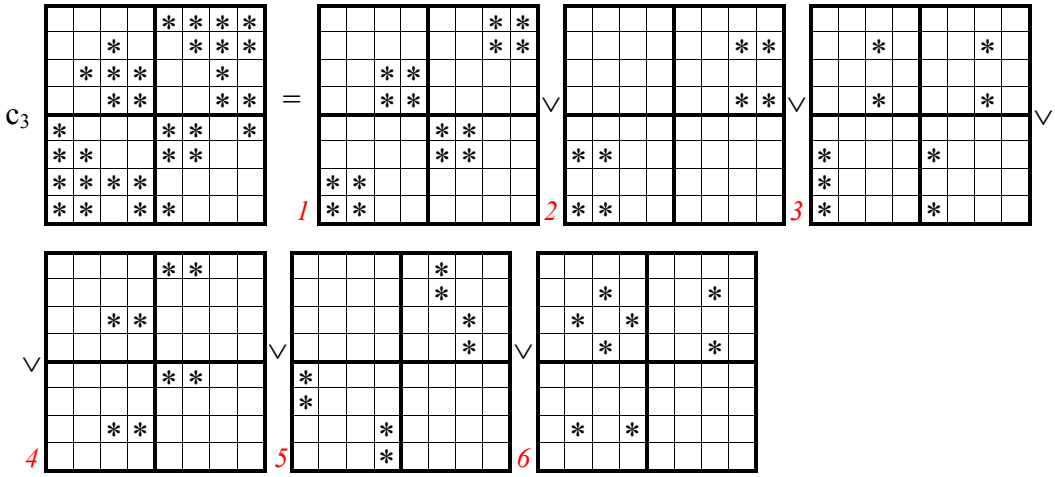


Рис. 1.35

$$1 = a_2\,\underline{b_2}\,a_3\,\underline{b_3} \vee a_2\,b_2\,\underline{a_3}\,\underline{b_3} \vee \underline{a_2}\,b_2\,a_3\,b_3 \vee \underline{a_2}\,\underline{b_2}\,a_3\,b_3;\ 2 = a_2\,b_1\,a_3\,\underline{b_3} \vee \underline{a_2}\,b_1\,\underline{a_3}\,b_3;$$

$$3 = \underline{a_1}\,a_2\,b_1\,\underline{b_3} \vee \underline{a_1}\,\underline{a_2}\,b_1\,b_3;\ 4 = a_2\,\underline{b_1}\,b_2\,\underline{a_3} \vee \underline{a_2}\,\underline{b_1}\,\underline{b_2}\,a_3;$$

$$5 = a_1\,\underline{a_2}\,\underline{b_2}\,a_3\,\underline{b_3} \vee \underline{a_1}\,a_2\,b_2\,a_3\,\underline{b_3} \vee \underline{a_1}\,\underline{a_2}\,\underline{b_2}\,\underline{a_3}\,b_3 \vee a_1\,a_2\,b_2\,\underline{a_3}\,b_3;$$

$$6 = a_1\,\underline{b_1}\,b_2\,\underline{a_3} \vee \underline{a_1}\,a_2\,b_1\,\underline{b_3}. \qquad (1.13.5)$$

Процедура определения геометрических подмножеств, составляющих множества геометрических образов сигналов c_1, c_2 ,c_3, представлена в графическом виде соответственно на рис.1.33 – 1.35. Эти геометрические подмножества в каждом основном множестве изображены на рисунках для большей прозрачности записи их логических выражений отдельными подгруппами, логические зависимости для которых приведены ниже рисунков и обозначены курсивом цифрами красного цвета. Логические суммы этих подмножеств и определяют схемное выполнение основного логического блока сумматора, а режимы работы сумматора при переходе от суммирования к вычитанию меняются также в зависимости от сигналов заема и переноса P_{n-1} (Z_{n-1}) и реализуются соответствующим инвертированием сигналов операндов и выходного сигнала c_3 по данным таблицы истинности (табл. 1.13.1), в которой режим работы логический нуль (0*) определяет операцию суммирования **C = (A + B)**, а логическая единица (1*) – операцию вычитания **C = (A – B).**

Таблица 1.13.1

Режим	P_{n-1} (Z_{n-1})	**A**	**B**	c_3
0*	0*	**A**	**B**•	c_3
0*	1*	**A**•	**B**	$\underline{c}_3$
1*	0*	**A**•	**B**	$\underline{c}_3$
1*	1*	**A**	**B**•	$\underline{c}_3$

1.14. Синтез одноразрядных умножителей

Для синтеза одноразрядных умножителей больших оснований систем счисления необходимо использовать таблицы умножения в цифровых координатах его операндов. Построение таких одноразрядных умножителей для оснований систем счисления, кратных двум, даже когда в операндах применяется основной двоичный код, исследовано недостаточно, а синтез систем счисления таких оснований при нетрадиционных двоичных кодах, необходимость применения которых возникает в цифровых системах с обнаружением и исправлени-

ем различного типа ошибок, в литературе отсутствует. Это положение можно объяснить тем, что нетрадиционные двоичные коды до настоящего времени считались непригодными для выполнения любых арифметических операций.

На примере табл. 1.14.1 системы счисления основания n = 8 проведем синтез одноразрядного умножителя **C = A × B**, когда его операнды и выходные сигналы представляются в различных комбинациях их кодов на входных и выходных шинах одноразрядного умножителя.

Таблица 1.14.1

C = A×B

	A	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
B	**0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
	2	0.0	0.2	0.4	0.6	1.0	1.2	1.4	1.6
	3	0.0	0.3	0.6	1.1	1.4	1.7	2.2	2.5
	4	0.0	0.4	1.0	1.4	2.0	2.4	3.0	3.4
	5	0.0	0.5	1.2	1.7	2.4	3.1	3.6	4.3
	6	0.0	0.6	1.4	2.2	3.0	3.6	4.4	5.2
	7	0.0	0.7	1.6	2.5	3.4	4.3	5.2	6.1

Результат операции умножения представляется в этой таблице двухразрядным числом, где цифры первого разряда числа определяют выходной сигнал конкретного разряда основания системы счисления, а цифры второго разряда – сигнал переноса в следующий старший разряд. Последующий порядок синтеза геометрических образов выходных сигналов разрядов c_1, c_2 ,c_3 и сигналов переноса P_1, P_2, P_3, который основывается на данных табл. 1.14.1, полностью аналогичен синтезу устройства сложения. Поэтому, опуская очевидные из вышеизложенного промежуточные преобразования, на рис. 1.36, *а – г* приведены геометрические образы этих сигналов соответственно для следующих вариантов применения кодов в операндах и выходных сигналах устройства:

а) **A, B, C** – в основном двоичном коде;
б) **A, B, C** – в коде Грея;
в) **A, B** – в коде Грея, **C** – в основном двоичном коде;
г) **A** – в коде Грея, **B** и **C** – в основном двоичном коде.

A, **B**, **C** = **A** × **B** – в основном двоичном коде

	0	1	2	3	4	5	6	7
0								
1		*		*		*		*
2								
3		*		*		*		*
4								
5		*		*		*		*
6								
7		*		*		*		*

c_1

	0	1	2	3	4	5	6	7
0								
1			*	*			*	*
2		*		*		*		*
3		*	*			*	*	
4								
5			*	*			*	*
6		*		*		*		*
7		*	*			*	*	

c_2

	0	1	2	3	4	5	6	7
0								
1					*	*	*	*
2			*	*			*	*
3			*		*	*		*
4		*		*		*		*
5		*		*	*		*	
6		*	*			*	*	
7		*	*	*	*			

c_3

	0	1	2	3	4	5	6	7
0								
1								
2					*	*	*	*
3					*	*		
4			*	*			*	*
5			*	*		*	*	
6			*		*	*		*
7			*		*		*	

P_1

	0	1	2	3	4	5	6	7
0								
1								
2								
3							*	*
4					*	*	*	*
5					*	*	*	
6				*	*	*		
7				*	*			*

P_2

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								*
6							*	*
7						*	*	*

P_3

б)

A, B, C – в коде Грея

	0	1	2	3	4	5	6	7
0								
1		*		*		*		*
2			*	*	*	*		
3		*	*		*			*
4			*	*	*	*		
5		*	*		*			*
6								
7		*		*		*		*

c_1

	0	1	2	3	4	5	6	7
0								
1			*	*		*		*
2		*		*	*	*		
3		*	*		*			*
4			*	*		*		*
5		*	*		*		*	
6								
7		*		*	*	*		

c_2

	0	1	2	3	4	5	6	7
0								
1					*	*	*	*
2				*	*		*	*
3			*	*	*	*		
4		*	*	*			*	
5		*		*		*		*
6		*	*		*			*
7		*	*			*	*	

c_3

	0	1	2	3	4	5	6	7
0								
1								
2					*	*		
3								
4			*	*	*	*		
5			*	*	*			
6			*	*			*	*
7			*	*			*	

P_1

	0	1	2	3	4	5	6	7
0								
1								
2					*	*		
3								
4			*			*	*	*
5			*		*	*	*	*
6					*	*	*	*
7					*	*	*	*

P_2

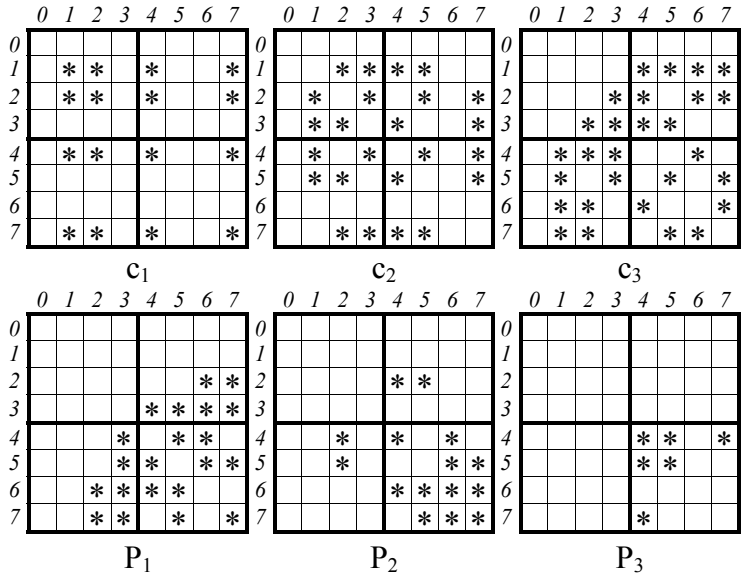
	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4					*	*		*
5					*	*		
6								
7					*			

P_3

б)

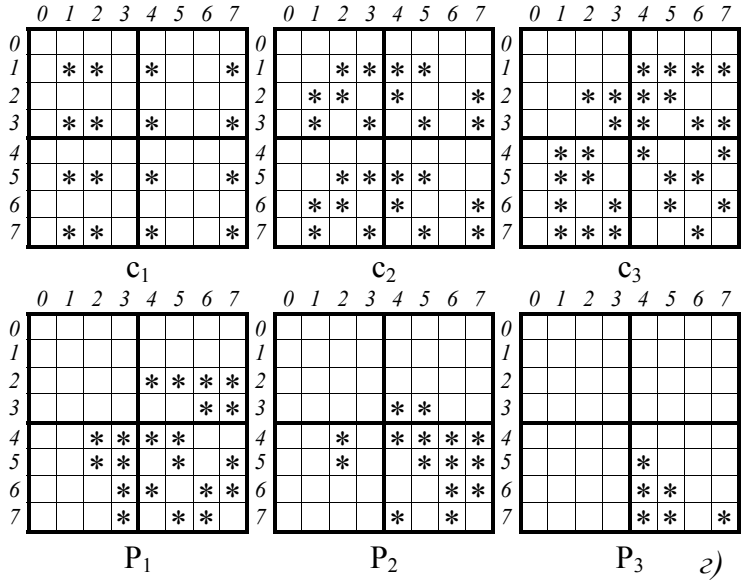
Рис. 1.36 (начало)

A, B – в коде Грея, **C** – в основном двоичном коде



в)

A – в коде Грея, **B** и **C** – в основном двоичном коде



г)

Рис. 1.36 (продолжение)

Все эти геометрические образы изображены в многомерном цифро-векторном пространстве координат основного двоичного кода, что позволяет применять для их покрытия геометрический алгоритм основного двоичного кода.

Умножители играют значительную роль в различных устройствах цифровой обработки сигналов, где обычно требуется повышенное быстродействие и поэтому желательно применение в них двухуровневых логических схем. Двухуровневый алгоритм покрытия геометрических образов позволяет получить необходимые логические зависимости необходимого быстродействия.

Геометрический алгоритм основного двоичного кода для синтеза двухуровневых логических схем поясняется рис. 1.37, *а – к*.

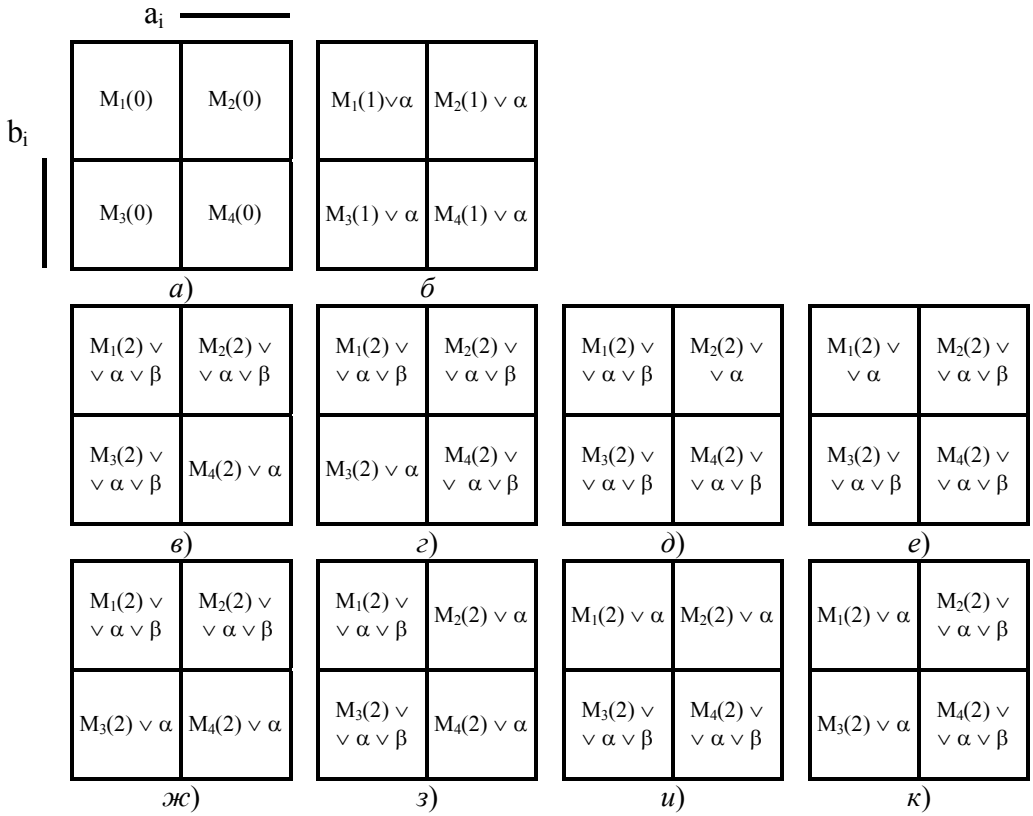


Рис. 1.37

Этот синтез начинается с рассмотрения геометрических образов логических функций двухмерного цифрового пространства, начиная со старших разрядов операндов a_n, b_n. В четвертях этого пространства (см. рис. 1.37, *а*) располагаются соответственно геометрические образы множеств $M_1(0)$ – $M_4(0)$. При отсутствии в этих множествах общих для них подмножеств их покрытие определяется логическим выражением

$$F = M_1(0)\,\underline{a_i}\,\underline{b_i} \vee M_2(0)\,a_i\,\underline{b_i} \vee M_3(0)\,\underline{a_i}\,b_i \vee M_4(0)\,a_i\,b_i \qquad (1.14.1)$$

и дальнейший этап синтеза будет заключаться в последовательном рассмотрении каждого из множеств $M_1(0)$ – $M_4(0)$.

Если в каждом из этих множеств имеется подмножество α (рис. 1.37, *б*), то выражение (1.14.1) будет преобразовано к виду

$$F = M_1(1)\,\underline{a}_i\,\underline{b}_i \vee M_2(1)\,a_i\,\underline{b}_i \vee M_3(1)\,\underline{a}_i\,b_i \vee M_4(1)\,a_i\,b_i \;\vee\; \alpha\,. \qquad (1.14.2)$$

При этом если в множествах $M_1(1)$ – $M_4(1)$ нет общих подмножеств, то дальнейший этап синтеза будет заключаться в последовательном рассмотрении каждого из них.

Когда множества $M_1(1)$ – $M_4(1)$ содержат в различных сочетаниях подмножества β (см. рис.1.37, *в – к*), то логические выражения, определяющие покрытие этих геометрических образов, будут соответственно следующими:

$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta\underline{a}_i \vee \beta\underline{b}_i \;\vee \alpha. \qquad (1.14.3)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta a_i \vee \beta\underline{b}_i \;\vee \alpha. \qquad (1.14.4)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta\underline{a}_i \vee \beta b_i \;\vee \alpha. \qquad (1.14.5)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta a_i \vee \beta b_i \;\vee \alpha. \qquad (1.14.6)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \vee \beta\underline{b}_i \;\vee \alpha. \qquad (1.14.7)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta\underline{a}_i \vee \alpha. \qquad (1.14.8)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \vee \beta b_i \;\vee \alpha. \qquad (1.14.9)$$
$$F = M_1(2)\,\underline{a}_i\,\underline{b}_i \vee M_2(2)\,a_i\,\underline{b}_i \vee M_3(2)\,\underline{a}_i\,b_i \vee M_4(2)\,a_i\,b_i \;\vee \beta a_i \vee \alpha. \qquad (1.14.10)$$

Этот этап синтеза может начинаться и заканчиваться на одном из этих логических выражений. Следующие этапы синтеза заключаются в последовательном рассмотрении геометрических образов множеств $M_1(2)$ – $M_4(2)$ в координатах операнд a_{n-1}, b_{n-1} по правилам, описанным выше, и т.д.

При этом следует учитывать, что множество β может содержать в себе подмножества α и $M_i(2)$ и т.д.

Применяя этот геометрической алгоритм двухуровневого синтеза основного двоичного кода для первого варианта (**A, B, C** – в основном двоичном коде) схемной реализации устройства, его логические выражения можно представить в виде

$$c_1 = a_1 b_1;\; c_2 = a_1\,\underline{a}_2\,b_2 \vee a_2\,b_1\,\underline{b}_2 \vee \underline{a}_1\,a_2\,b_1 \vee a_1\,\underline{b}_1\,b_2;$$
$$c_3 = a_1\,b_1\,a_3\,\underline{b}_3 \vee a_1\,b_1\,\underline{a}_3\,b_3 \vee a_2\,\underline{b}_1\,b_2\,\underline{b}_3 \vee \underline{a}_1\,a_3\,b_1\,\underline{b}_2 \vee \underline{a}_1\,a_2\,\underline{a}_3\,b_2 \vee$$
$$\vee\, a_1\,\underline{b}_1\,\underline{b}_2\,b_3 \vee a_1\,\underline{a}_2\,\underline{b}_1\,b_3 \vee \underline{a}_1\,\underline{a}_2\,a_3\,b_1 \vee \underline{a}_1\,a_2\,\underline{b}_1 b_2;$$
$$P_1 = \underline{a}_2\,a_3\,\underline{b}_1\,b_2 \vee a_2\,\underline{b}_1\,\underline{b}_2\,b_3 \vee a_1\,a_3\,\underline{b}_1\,b_2 \vee \underline{a}_1\,a_2\,b_1\,b_3 \vee a_2\,\underline{b}_1\,b_2\,a_3\,\underline{b}_3 \vee$$
$$\vee\, \underline{a}_2\,a_3\underline{b}_1\,\underline{b}_2\,\underline{b}_3 \vee a_1\,a_2\,\underline{a}_3\underline{b}_2\,b_3 \vee \underline{a}_1\,a_2\,\underline{a}_3\,b_2\,b_3 \vee \underline{a}_1\,\underline{a}_2\,a_3 b_2\,b_3 \vee a_1\,\underline{a}_2\,a_3 b_1\,\underline{b}_2\,b_3;$$
$$P_2 = a_2\,a_3 b_1\,b_2\,\underline{b}_3 \vee a_1\,a_2\,\underline{a}_3 b_2\,b_3 \;\vee\; \underline{a}_2\,a_3\,\underline{b}_2\,b_3 \;\vee a_1\,a_2\,a_3\,\underline{b}_1\,b_2 \vee a_3\,\underline{b}_1\,\underline{b}_2\,b_3 \vee$$
$$\vee\, \underline{a}_1\,\underline{a}_2\,a_3\,b_3 \vee \;\underline{a}_2\,a_3\underline{b}_1\,b_3 \vee \;\underline{a}_1\,a_3\,\underline{b}_2\,b_3;$$
$$P_3 = a_2\,a_3 b_2\,b_3 \vee a_1\,a_3\,b_1\,b_2\,b_3 \;\vee\; a_1\,a_2\,a_3\,b_1\,b_3. \qquad (1.14.11)$$

Для второго варианта (**A, B, C** – в коде Грея) схемной реализации устройства логические выражения запишутся:

$$c_1 = a_1\,b_1\,\underline{b}_2\,\underline{b}_3 \vee a_1\,b_1\,b_2\,b_3 \vee a_1\,a_2\,a_3 b_1 \vee \underline{a}_1\,a_2\,\underline{a}_3\,b_2\,\underline{b}_3 \vee \underline{a}_1\,a_2\,\underline{a}_3\,\underline{b}_2\,b_3 \vee \underline{a}_1\,\underline{a}_2\,a_3\,b_2\,\underline{b}_3 \vee$$
$$\vee\; \underline{a}_1\,\underline{a}_2\,a_3\,\underline{b}_2\,b_3 \vee a_2\,\underline{a}_3\,\underline{b}_1\,b_2\,\underline{b}_3 \vee \underline{a}_2\,a_3\,\underline{b}_1\,b_2\,\underline{b}_3 \vee a_2\,\underline{a}_3\,\underline{b}_1\,\underline{b}_2\,b_3 \vee \underline{a}_2\,a_3\,\underline{b}_1\,\underline{b}_2\,b_3 \vee a_1\,\underline{a}_2\,\underline{a}_3\,b_1;$$

$$c_2 = a_1 a_3 b_1 \underline{b_2} \underline{b_3} \vee a_1 \underline{a_3} \underline{b_1} b_2 \underline{b_3} \vee a_1 a_3 \underline{b_1} b_2 b_3 \vee \underline{a_1} a_3 b_1 \underline{b_2} b_3 \vee a_1 \underline{a_3} b_1 b_2 b_3 \vee \underline{a_1} a_2 \underline{a_3} b_1 \underline{b_3} \vee$$
$$\vee a_1 a_2 a_3 b_1 \underline{b_3} \vee a_1 \underline{a_2} \underline{a_3} b_1 b_3 \vee a_1 \underline{a_2} \underline{a_3} b_1 b_3 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_3 \vee a_2 \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee a_1 \underline{a_2} \underline{a_3} b_2 \underline{b_3} \vee$$
$$\vee \underline{a_1} \underline{a_2} a_3 b_2 \underline{b_3} \vee \underline{a_2} a_3 \underline{b_1} b_2 \underline{b_3} \vee a_2 \underline{a_3} \underline{b_1} \underline{b_2} b_3 \vee a_2 \underline{a_3} \underline{b_1} \underline{b_2} b_3 \vee a_1 \underline{a_2} \underline{a_3} b_1 b_2 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_2;$$
$$c_3 = a_1 b_1 \underline{b_2} b_3 \vee a_1 \underline{a_2} a_3 \underline{b_1} b_3 \vee \underline{a_1} a_2 \underline{a_3} \underline{b_1} b_3 \vee \underline{a_2} a_3 b_1 \underline{b_3} \vee \underline{a_1} a_3 \underline{b_1} b_2 b_3 \vee a_1 a_2 \underline{b_1} b_2 \underline{b_3} \vee$$
$$\vee a_2 a_3 b_1 \underline{b_2} \underline{b_3} \vee a_2 a_3 \underline{b_1} b_2 \underline{b_3} \vee a_2 \underline{a_3} b_1 b_2 \underline{b_3} \vee a_1 a_2 \underline{a_3} \underline{b_2} b_3 \vee \underline{a_1} a_2 \underline{b_1} \underline{b_2} b_3 \vee a_1 \underline{a_2} b_1 b_2 b_3 \vee$$
$$\vee \underline{a_1} a_2 b_1 b_2 b_3 \vee \underline{a_1} \underline{a_2} a_3 \underline{b_1} b_2 \vee a_1 a_2 a_3 \underline{b_1} b_2;$$
$$P_1 = a_3 b_2 \underline{b_3} \vee a_2 \underline{a_3} b_3 \vee \underline{a_2} a_3 \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} \underline{a_2} a_3 \underline{b_2} b_3 \vee a_2 a_3 \underline{b_1} b_2 \text{ (или } a_2 \underline{b_1} b_2 b_3) \vee$$
$$\vee \underline{a_1} a_2 a_3 b_2 \text{ (или } \underline{a_1} a_1 b_2 b_3);$$
$$P_2 = \underline{a_1} a_2 \underline{b_2} b_3 \vee \underline{a_2} a_3 \underline{b_1} b_2 \vee a_2 a_3 b_3 \vee a_3 b_2 b_3 \vee a_1 a_3 b_3 \vee b_1 b_3 a_3;$$
$$P_3 = \underline{a_2} a_3 \underline{b_2} b_3 \vee a_1 a_3 \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_3. \qquad (1.14.12)$$

Для третьего варианта (**A, B** – в коде Грея, **C** – в основном двоичном коде) логические выражения принимают вид

$$c_1 = a_1 \underline{a_2} \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee \underline{a_1} a_2 \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee \underline{a_1} \underline{a_2} a_3 b_1 \underline{b_2} \underline{b_3} \vee a_1 a_2 a_3 b_1 \underline{b_2} \underline{b_3} \vee$$
$$\vee a_1 \underline{a_2} \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee \underline{a_1} a_2 \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee \underline{a_1} \underline{a_2} a_3 b_1 \underline{b_2} \underline{b_3} \vee a_1 a_2 a_3 b_1 \underline{b_2} \underline{b_3} \vee$$
$$\vee a_1 \underline{a_2} \underline{a_3} \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} a_2 \underline{a_3} \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} \underline{a_2} a_3 \underline{b_1} \underline{b_2} b_3 \vee a_1 a_2 a_3 \underline{b_1} \underline{b_2} \underline{b_3} \vee$$
$$\vee a_1 \underline{a_2} \underline{a_3} b_1 b_2 b_3 \vee \underline{a_1} a_2 \underline{a_3} b_1 b_2 b_3 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_2 b_3 \vee a_1 a_2 a_3 b_1 b_2 b_3;$$
$$c_2 = \underline{a_1} a_2 \underline{a_3} b_1 \vee \underline{a_1} \underline{a_2} \underline{a_3} b_1 \vee a_1 \underline{b_1} b_2 \underline{b_3} \vee a_1 \underline{b_1} \underline{b_2} b_3 \vee a_2 \underline{a_3} b_1 \underline{b_2} \underline{b_3} \vee \underline{a_2} a_3 b_1 \underline{b_2} \underline{b_3} \vee$$
$$\vee a_1 \underline{a_2} \underline{a_3} b_2 \underline{b_3} \vee a_1 a_2 a_3 b_2 \underline{b_3} \vee a_1 \underline{a_2} \underline{a_3} \underline{b_2} b_3 \vee a_1 a_2 a_3 \underline{b_2} b_3 \vee a_2 \underline{a_3} b_1 b_2 b_3 \vee \underline{a_2} a_3 b_1 b_2 b_3;$$
$$c_3 = a_1 \underline{a_3} \underline{b_2} b_3 \vee a_1 a_3 b_1 \underline{b_2} \vee \underline{a_1} a_3 \underline{b_1} b_2 \underline{b_3} \vee a_3 b_1 \underline{b_2} b_3 \vee \underline{a_2} a_3 b_1 \underline{b_3} \vee a_2 \underline{a_3} b_1 b_2 \underline{b_3} \vee$$
$$\vee a_1 a_2 \underline{a_3} b_2 \underline{b_3} \vee a_1 \underline{a_2} a_3 b_2 b_3 \vee \underline{a_1} a_2 \underline{a_3} b_2 b_3 \vee a_1 \underline{a_2} b_1 b_3 \vee \underline{a_1} \underline{a_2} a_3 \underline{b_1} b_2 \vee a_1 a_2 a_3 \underline{b_1} b_2 \vee$$
$$\vee \underline{a_1} a_2 \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} a_2 b_1 b_2 b_3;$$
$$P_1 = a_1 a_3 b_1 b_2 \vee a_1 a_2 b_1 b_3 \vee a_3 b_1 b_2 \underline{b_3} \vee a_1 a_2 \underline{a_3} b_3 \vee a_1 \underline{a_2} a_3 \underline{b_1} \vee \underline{a_1} \underline{a_2} a_3 \underline{b_1} b_3;$$
$$P_2 = \underline{a_1} a_2 \underline{b_2} b_3 \vee \underline{a_2} a_3 \underline{b_1} b_2 \vee \underline{a_1} a_3 \underline{b_1} b_3 \vee a_1 a_2 a_3 b_1 b_3 \vee b_1 b_2 b_3 a_1 a_3 \vee a_2 a_3 b_2 b_3;$$
$$P_3 = \underline{a_2} a_3 \underline{b_2} b_3 \vee a_1 a_3 \underline{b_1} \underline{b_2} b_3 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_3. \qquad (1.14.13)$$

Для четвертого варианта (**A** – в коде Грея, **B** и **C** – в основном двоичном коде) логические выражения будут следующими:

$$c_1 = a_1 \underline{a_2} \underline{a_3} b_1 \underline{b_2} \vee \underline{a_1} a_2 \underline{a_3} b_1 b_2 \vee \underline{a_1} \underline{a_2} a_3 b_1 \underline{b_2} \vee a_1 a_2 a_3 b_1 \underline{b_2} \vee a_1 \underline{a_2} \underline{a_3} b_1 b_2 \vee$$
$$\vee \underline{a_1} a_2 \underline{a_3} b_1 b_2 \vee \underline{a_1} \underline{a_2} a_3 b_1 b_2 \vee a_1 a_2 a_3 b_1 b_2;$$
$$c_2 = a_2 \underline{a_3} b_1 \underline{b_2} \vee \underline{a_2} \underline{a_3} b_1 \underline{b_2} \vee a_1 b_1 b_2 \vee a_1 \underline{a_2} a_3 b_2 \vee a_1 a_2 a_3 b_2 \vee \underline{a_1} a_2 \underline{a_3} b_1 \underline{b_2} \vee$$
$$\vee \underline{a_1} \underline{a_2} a_3 b_1 \underline{b_2} \vee \underline{a_1} a_2 a_3 \underline{b_1} b_2 \vee \underline{a_1} \underline{a_2} \underline{a_3} \underline{b_1} b_2;$$
$$c_3 = \underline{a_1} \underline{a_3} b_1 \underline{b_3} \vee a_1 \underline{b_1} b_2 b_3 \vee a_1 \underline{a_2} a_3 b_1 \underline{b_2} \vee \underline{a_1} a_2 a_3 b_1 b_3 \vee a_1 a_2 \underline{a_3} b_2 \vee a_2 \underline{a_3} b_1 b_2 b_3 \vee$$
$$\vee a_1 \underline{a_2} \underline{a_3} \underline{b_2} b_3 \vee \underline{a_1} a_2 \underline{a_3} \underline{b_2} b_3 \vee a_2 a_3 b_1 \underline{b_2} \underline{b_3} \vee a_2 \underline{a_3} \underline{b_1} b_2 \underline{b_3} \vee \underline{a_2} a_3 \underline{b_1} b_2 \underline{b_3} \vee$$
$$\vee a_2 a_3 b_1 b_2 \underline{b_3} \vee a_1 a_2 \underline{a_3} \underline{b_1} \underline{b_2} b_3 \vee a_1 a_2 a_3 \underline{b_1} \underline{b_2} b_3;$$
$$P_1 = \underline{a_1} a_3 \underline{b_1} b_2 \vee a_3 \underline{b_1} b_2 \underline{b_3} \vee a_1 a_3 b_1 \underline{b_2} b_3 \vee a_2 a_3 b_2 \underline{b_3} \vee a_2 \underline{a_3} \underline{b_2} b_3 \vee a_1 a_2 \underline{a_3} b_3 \vee$$
$$\vee \underline{a_1} a_2 a_3 b_2 \vee \underline{a_2} a_3 \underline{b_1} \underline{b_2} b_3 \vee a_1 \underline{a_2} a_3 \underline{b_2} b_3 \vee a_1 \underline{a_2} a_3 b_1 b_3 \vee a_2 a_3 \underline{b_1} b_2 b_3;$$
$$P_2 = \underline{a_1} a_2 \underline{b_2} b_3 \vee \underline{a_2} a_3 b_1 b_2 \underline{b_3} \vee \underline{a_1} a_3 b_1 b_2 b_3 \vee a_1 a_3 \underline{b_2} b_3 \vee a_3 \underline{b_1} \underline{b_2} b_3 \vee a_2 a_3 \underline{b_1} b_3;$$
$$P_3 = \underline{a_2} \underline{a_3} b_2 b_3 \vee \underline{a_1} \underline{a_2} \underline{a_3} b_1 b_3 \vee a_1 \underline{a_3} b_1 b_2 b_3. \qquad (1.14.14)$$

Схемная реализация двухуровневого одноразрядного умножителя (1.14.11), когда все операнды и результат операции представляются в основном двоичном коде, содержит меньше оборудования.

Все последующие варианты (1.14.12) – (1.14.14) более затратны, но если операнды и результат операции представлены в различных кодах и одновременно требуется максимальное быстродействие от схемы одноразрядного умножителя, то их использование становится оправданным.

При реализации менее быстродействующей многоуровневой схемы одноразрядного умножителя, при любых сочетаниях кодов операндов, первый вариант схемы предпочтительней. В этом случае все коды операндов преобразуются в основной двоичный код, а затем поступают на входные шины одноразрядного умножителя.

Едва ли кто-нибудь из нематематиков в состоянии освоиться с мыслью, что цифры могут представлять собой культурную и эстетическую ценность или иметь какое-нибудь отношение к таким понятиям, как красота, сила, вдохновение.

Н. Винер

Глава 2

СИНТЕЗ УМНОЖИТЕЛЕЙ, ДЕЛИТЕЛЕЙ И СЧЕТЧИКОВ

2.1. Синтез устройств умножения

Устройства умножения широко используются в цифровой технике и в значительной степени определяют быстродействие математических операций, выполняемых в этих системах. В общеизвестных устройствах двоичной системы счисления [26, 27, 21] результат операции умножения формируется как сумма частичных произведений множимого или его частей на цифры множителя. В этих устройствах основные аппаратные средства для реализации алгоритмов умножения состоят из блоков, осуществляющих сложение, и блоков, осуществляющих хранение чисел операндов и результата сложения, а также их сдвига [26], либо из блоков, осуществляющих параллельное сложение во всех разрядах результата умножения [21]. Отличительной особенностью этих устройств является то, что они имеют большие аппаратурные затраты и, несмотря на меры по ускорению операции умножения, невысокое быстродействие. Непосредственное применение таких устройств в системах, где применяются многофазные коды, потребует установки на входе и выходе устройства умножения преобразователей из одного кода в другой, что не способствует снижению аппаратурных затрат и повышению быстродействия. При этом также очевидно, что устройства с большим основанием систем счисления более быстродействующие и переход к системе счисления с основанием два нецелесообразен.

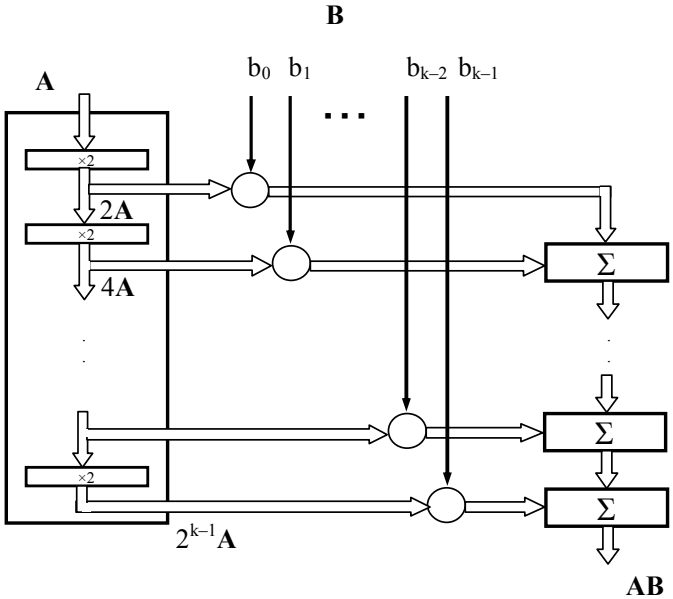


Рис. 2.1

Поскольку все вычислительные машины выполняются в двоичной системе счисления, а многофазный код является принадлежностью электроприводов, то существует потребность объединения составных частей такого комплекса в единую систему без промежуточных преобразований одного кода в другой. Эта задача была решена в [1, 2], где предложена схема блока умножения (рис. 2.1), которая не требует преобразования кодов. В этой схеме одно число, например **A,** поступает в многофазном коде, а другое **B** – в двоичном коде, что позволяет управлять электроприводом непосредственно от внешнего двоичного сигнала.

Первый операнд **A** в многофазном коде, число разрядов которого S, проходит через последовательно соединенные блоки умножения на два (×2), число которых равно разрядности k второго операнда **B** (b_0, ... , b_{k-1}).

Сигналы второго операнда пропускают через соответствующие вентили выходные сигналы 2**A,** 4**A,** ... **,** 2^{k-1}**A** блоков умножения на входы последовательно соединенных многоразрядных блоков суммирования Σ, число которых также равно k. Таким образом, все входные и выходные сигналы блоков суммирования будут представлены в многофазном коде.

Выполнение одноразрядного умножителя на два будет отличаться для каждого из значений числа фаз m. Однако сам принцип синтеза этих умножителей одинаков и может быть пояснен на примере одного из них. Пусть это будет представлено для m=4.

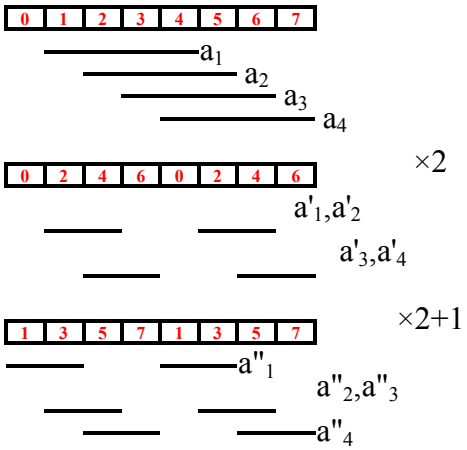


Рис. 2.2

На рис. 2.2 приведены соотношения одноразрядных сигналов этого кода a_1, ... , a_4, умноженных на два – a'_1, ... , a'_4, а также увеличенных на единицу при наличии сигнала переноса p с предыдущего разряда результата умножения – a''_1, ... , a''_4. Связь выходных сигналов a''_1, ... , a''_4 с входными a_1, ... , a_4; p определяется простыми логическими выражениями:

$$a''_1=(a_1 \otimes a_3)\,\underline{p} \vee \underline{(a_1 \otimes a_3)}\,p;$$
$$a''_2=(a_1 \otimes a_3);$$
$$a''_3=(a_2 \otimes a_4)\,\underline{p} \vee (a_1 \otimes a_3)\,p;$$
$$a''_4=(a_2 \otimes a_4). \qquad (2.1.1)$$

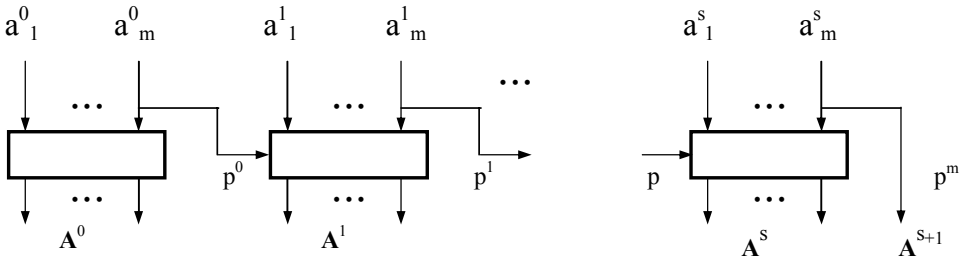


Рис. 2.3

Первый разряд каждого многоразрядного умножителя на два (рис. 2.3) не имеет входного сигнала переноса p, и его выполнение определяется первыми составляющими (2.1.1). Прохождение сигнала переноса по последующим разрядам здесь не происходит, и его действие заканчивается на следующем разряде после исходного. Число разрядов по мере спуска по дереву умножителей на два постепенно возрастает. Причем последний разряд в этой цепочке также постепенно усложняется до появления сигнала m фазы, когда возникает сигнал переноса в последующий разряд. Увеличение разрядов умножителей на два должно отслеживаться в схемах многоразрядных сумматоров по мере синхронного спуска по дереву сумматоров. Выполнение таких сумматоров очевидно из предыдущих разделов.

Более высокое быстродействие может быть получено в матричных блоках умножения [19], где используются одноразрядные матричные умножители и сумматоры, работающие с обычным цифровым кодом основания больше двух, например n =10.

Для использования такого умножения блока в нашем случае необходимо преобразование многофазного кода в обычный цифровой код, а после проведения операции умножения требуется выполнить обратное преобразование. Следует отметить, что преобразование кодов здесь проще, чем при использовании двоичного представления больших оснований систем счисления. Одноразрядные матричные умножители и сумматоры в этих устройствах содержат квадратную матрицу размерами $n \times n$, в узлах которой расположены двухвходовые элементы И, выходные шины которых соединены с выходными шинами через элементы ИЛИ. Отличительной особенностью одноразрядного сумматора и умножителя здесь является высокое быстродействие, что определяет быстродействие всего устройства умножения. Вместе с тем его реализация сопровождается значительными аппаратурными затратами, поскольку все операции производятся с единичными цифровыми множествами.

Принципы построения матричных устройств умножения [19] были использованы нами в устройствах, работающих непосредственно в многофазных кодах [16]. Благодаря симметрии сигналов многофазного кода, использованию множеств, охватывающих большие цифровые области пространства, здесь удалось при высоком быстродействии уменьшить затраты оборудования примерно в два раза. Еще больших успехов в сокращении оборудования можно достигнуть при использовании в устройствах умножения генератора кратностей.

Сущность хорошо известного метода кратных множимого [22, 27] заключается в том, что при умножении на цифру b_i в n-ичной системе счисления частичное произведение формируется как сумма сложений, где имеет смысл заранее подготовить все возможные произведения x×a, где x=1, 2, ... , (n – 1), и, проанализировав текущее значение b_i, выбрать одну из (n – 1) заготовок.

Синтез устройства умножения, которое применимо для любой системы счисления, поясняется структурной схемой рис. 2.4 на примере системы счисления основания n=8 при его представлении в многофазном коде.

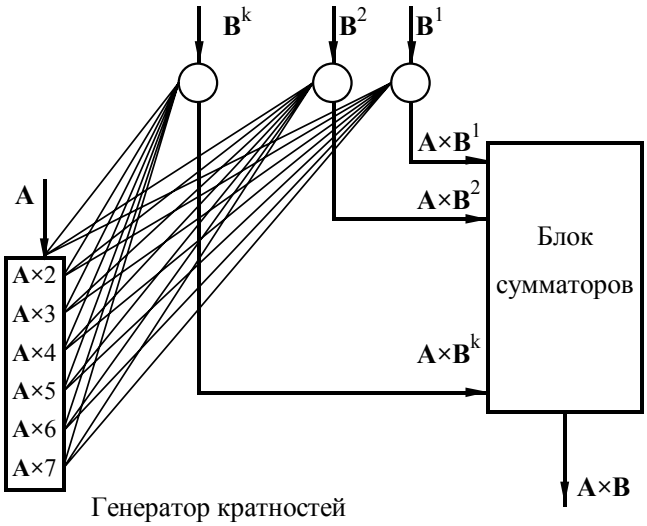


Рис. 2.4

Генератор кратностей осуществляет заготовку всех произведений операнда **A** на x=1, 2, ... , 7, а каждый разряд второго операнда **B** в соответствии со своим значением x посылает в блок сумматоров произведение **A**×x. Оптимизация всего устройства умножения заключается здесь лишь в оптимизации генератора кратностей и многовходовых сумматоров.

Примем следующие обозначения. Операнды обозначаются заглавными буквами **A** и **B**. Если у заглавной буквы установлен индекс, например $\mathbf{A}^k$, то это означает k-й разряд числа **A**. Строчные буквы с индексами внизу обозначают, как и прежде, сигналы многофазного кода в пределах одного разряда. Результат умножения на константу «два» будем обозначать $\mathbf{C}^k$, на «три» – $\mathbf{D}^k$, на «четыре» – $\mathbf{H}^k$, на «пять» – $\mathbf{F}^k$, на «шесть» – $\mathbf{R}^k$, на «семь» – $\mathbf{S}^k$.

Для синтеза устройства необходимо знать таблицы умножения и сложения разряда для конкретной системы счисления.

Результат умножения для первого разряда очевиден из таблицы умножения для всех констант и представлен графически на рис. 2.5.

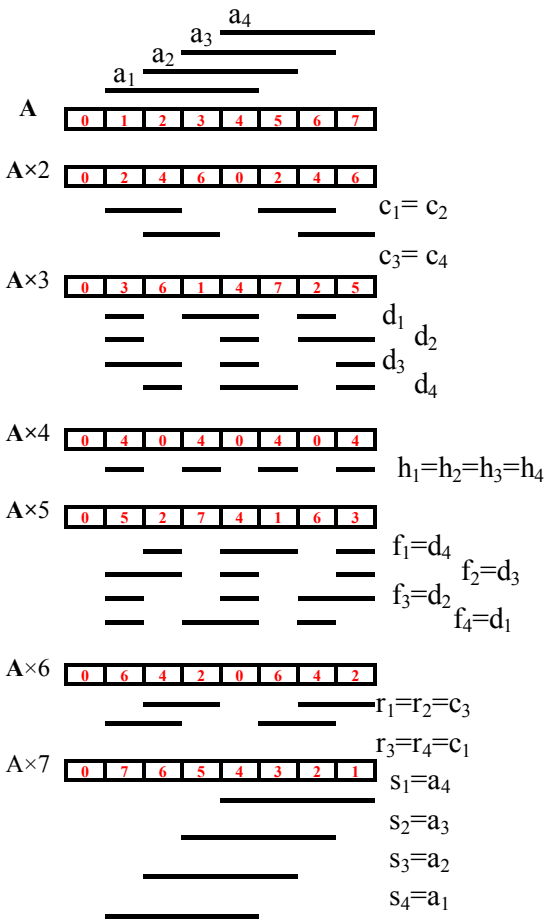


Рис. 2.5

Сигналы, представленные на рис. 2.5, определяются по следующим логическим зависимостям:

$$
\begin{gathered}
c_1= c_2= (a_1 \otimes a_3);\\
c_3= c_4= (a_2 \otimes a_4);\\
d_1= a_1\underline{a}_2 \vee a_1a_3 \vee \underline{a}_2a_3;\\
d_2= a_1\underline{a}_2 \vee a_1a_4 \vee \underline{a}_2a_4;\\
d_3= a_1\underline{a}_3 \vee a_1a_4 \vee \underline{a}_3a_4;\\
d_4= a_2\underline{a}_3 \vee a_2a_4 \vee \underline{a}_3a_4;\\
h_1= \ldots= h_4= (a_1 \otimes a_2)\,(a_3 \otimes a_4);\\
f_1=d_4,\ f_2=d_3,\ f_3=d_2,\ f_4=d_1;\\
r_1=r_2=c_3;\ r_3=r_4=c_1;\\
s_1=a_4,\ s_2=a_3,\ s_3=a_2,\ s_4=a_1.
\end{gathered}
\tag{2.1.2}
$$

Выполнение операции умножения на константу «два» было рассмотрено выше и не требует дополнительных пояснений.

Умножение на константу «три» для второго разряда результата умножения представляется табл. 2.1.1, где «единицы» результата умножения во втором разряде суммируются с «восьмерками» (переносами) с первого разряда этой же операции умножения.

Таблица 2.1.1

	Перенос с $\mathbf{A}^1$							
$\mathbf{A}^2\times3$	**0**	**0**	**0**	**1**	**1**	**1**	**2**	**2**
00	0	0	0	1	1	1	2	2
03	3	3	3	4	4	4	5	5
06	6	6	6	7	7	7	10	10
11	11	11	11	12	12	12	13	13
14	14	14	14	15	15	15	16	16
17	17	17	17	20	20	20	21	21
22	22	22	22	23	23	23	24	24
25	25	25	25	26	26	26	27	27

Результат этой двойной операции (умножения и суммирования) представлен графически на рис. 2.6а для сигналов $d^2_1 - d^2_4$ и цифр переноса P^2_d в третий разряд устройства.

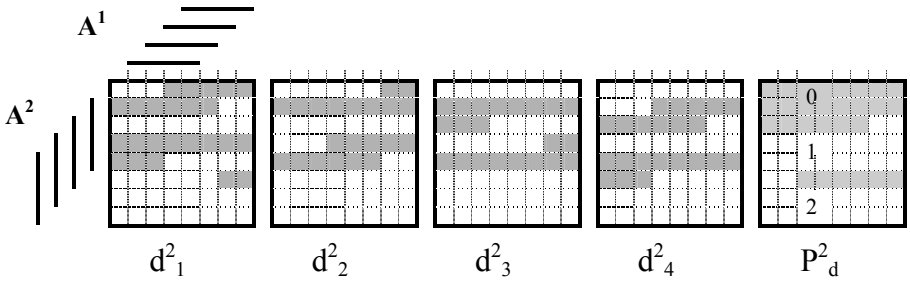


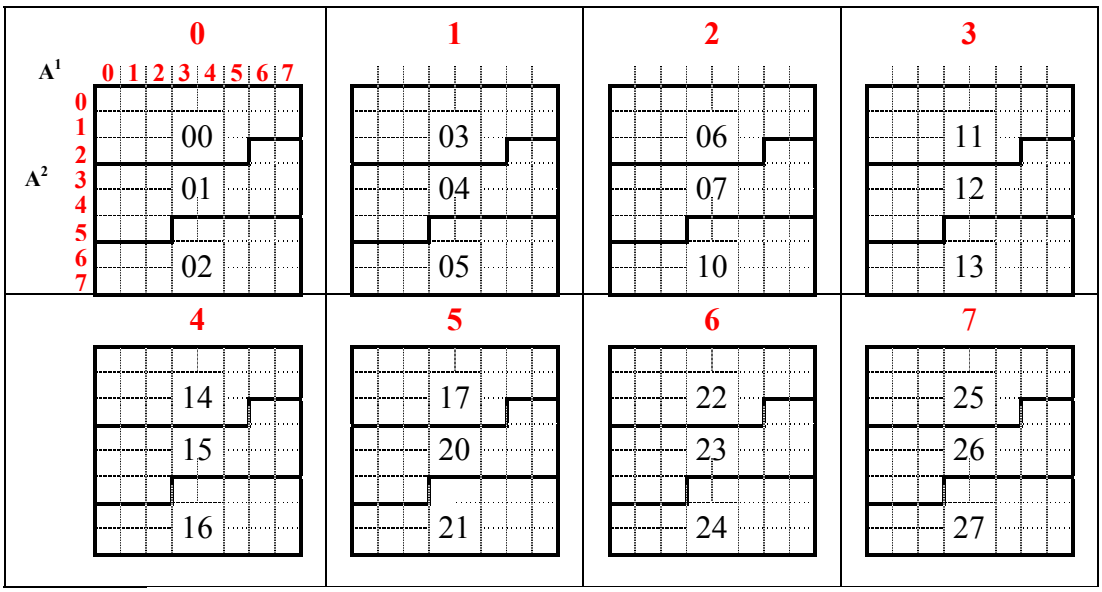
Рис. 2.6а

Из рис. 2.6а видно, что геометрические образы сигналов $d^2_1 - d^2_4$ на второй половине двухмерного цифрового пространства равны инверсии этих образов первой половины пространства. Следовательно, в дальнейшем можно изображать только одну половину геометрических образов этих сигналов, поскольку вторая половина этих образов очевидна.

Проделав эти же двойные операции в третьем разряде, можно получить цифровые множества сигналов $d^3_1 - d^3_4$ и переноса P^3_d в четвертый разряд устройства.

Для этого построим трехмерную табл. 2.1.1', где «единицы» результата умножения в третьем разряде суммируются с «восьмерками» P^2_d, поступающими со второго разряда этой операции умножения.

Таблица 2.1.1



Эта процедура построения таблицы и на ее основе геометрических образов соответствующих сигналов выполняется следующим образом.

Геометрические образы сигналов d^3_1 – d^3_4 будем представлять только на первой половине трехмерного цифрового пространства (цифры **0–3** третьего разряда $\mathbf{A}^3$), а образ сигнала переноса P^3_d, поскольку он не имеет симметрии в этом пространстве, будем представлять полностью.

Рассмотрим вначале геометрический образ сигнала d^3_1. Для цифры **0** ($\mathbf{A}^3$) умножение на «три» остается без последствий, и сигнал переноса P^2_d (см. рис. 2.6а) полностью определяет в таблице результат умножения на константу «три» для третьего разряда генератора кратностей.

Для цифры **1** ($\mathbf{A}^3$) умножение на «три» дает результат **03** и сигнал переноса P^2_d, который с ней суммируется и полностью определяет результат умножения на константу «три» в третьем разряде генератора кратностей, и т.д.

Учитывая, что $d_1 = \mathbf{1} \vee \mathbf{2} \vee \mathbf{3} \vee \mathbf{4}$, $d_2 = \mathbf{2} \vee \mathbf{3} \vee \mathbf{4} \vee \mathbf{5}$, $d_3 = \mathbf{3} \vee \mathbf{4} \vee \mathbf{5} \vee \mathbf{6}$, $d_4 = \mathbf{4} \vee \mathbf{5} \vee \mathbf{6} \vee \mathbf{7}$, геометрические образы всех этих сигналов изображаются на рис. 2.6б в трехмерном цифровом пространстве ($\mathbf{A}^1$, $\mathbf{A}^2$, $\mathbf{A}^3$).

Аналогично из этой таблицы определяется геометрический образ сигнала переноса P^3_d, который также изображен на рис. 2.6б.

В дальнейшем описании работы генератора кратностей для мерности пространства больше трех подобные детали в построении геометрических образов выходных сигналов генератора кратностей из-за их очевидности будем опускать.

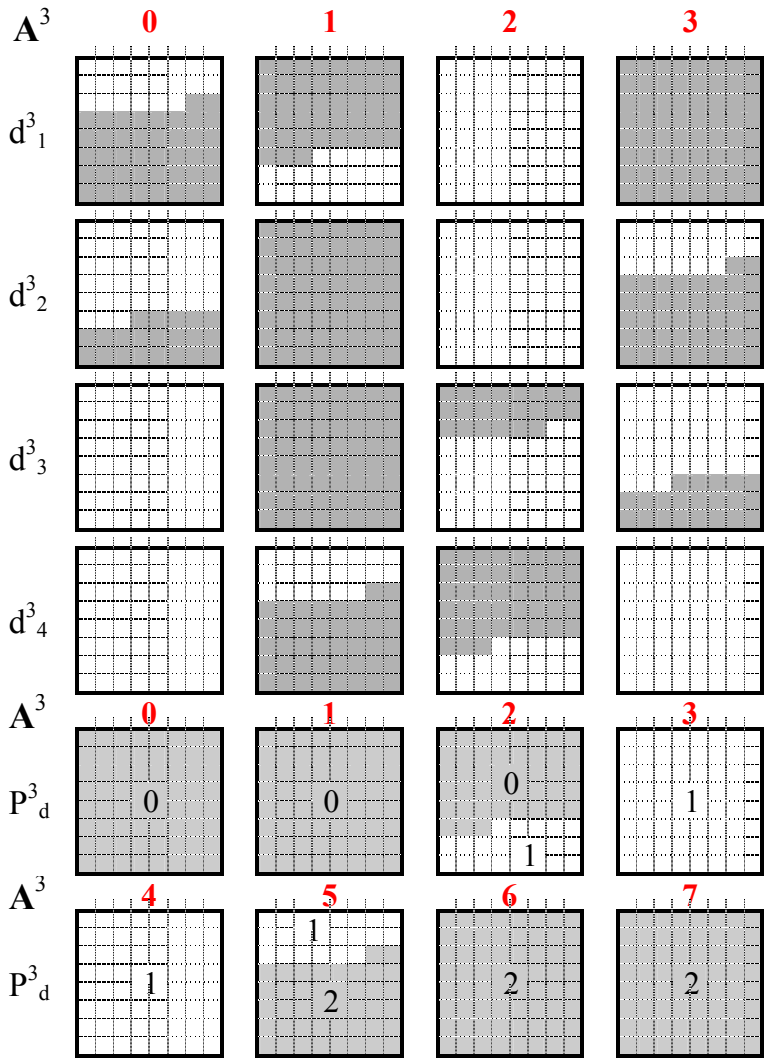


Рис. 2.6б

При этом геометрические образы всех выходных сигналов генератора кратностей в трехмерном цифровом пространстве **(A^1, A^2, A^3)** будем представлять также послойно и термин «послойно» в дальнейшем упоминать не будем.

Результаты этих операций для k-го разряда устройства в матричной форме запишутся следующим образом:

$$\mathbf{D}^k = \mathbf{M}^k_d \left| \begin{matrix} \mathbf{0} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{7} \end{matrix} \right| \begin{matrix} k \\ \\ \\ \\ \mathbf{A} \end{matrix},$$

где

$$\mathbf{M}^{k}_{D} = \left| \begin{array}{cccccccc} m_1 & m_2 & 0^* & 1^* & \underline{m_1} & \underline{m_2} & 1^* & 0^* \\ \underline{m_2} & 1^* & 0^* & m_1 & m_2 & 0^* & 1^* & \underline{m_1} \\ 0^* & 1^* & \underline{m_1} & \underline{m_2} & 1^* & 0^* & m_1 & m_2 \\ 0^* & m_1 & m_2 & 0^* & 1^* & \underline{m_1} & \underline{m_2} & 1^* \end{array} \right| \begin{array}{l} k \\ \\ \\ \mathbf{D} \end{array} \qquad (2.1.3)$$

Матрица цифровых множеств $\mathbf{M}^{k}_{D}$ определяется по следующей записи: для k= 2 $m_1 = (\mathbf{3} \vee ... \vee \mathbf{7})^1$; $m_2 = (\mathbf{0} \vee ... \vee \mathbf{5})^1$, а для k= 3 $m_1 = (\mathbf{3} \vee ... \vee \mathbf{7})^2 \vee (\mathbf{6} \vee \mathbf{7})^1 a^2_2$; $m_2 = (\mathbf{0} \vee ... \vee \mathbf{5})^2 \vee (\mathbf{0} \vee \mathbf{1} \vee \mathbf{2})^1 a^2_2$ и т.д.

Таблица 2.1.2

	Перенос с $\mathbf{A}^1$							
$\mathbf{A}^2 \times 4$	**0**	**0**	**1**	**1**	**2**	**2**	**3**	**3**
00	0	0	1	1	2	2	3	3
04	4	4	5	5	6	6	7	7
10	10	10	11	11	12	12	13	13
14	14	14	15	15	16	16	17	17
20	20	20	21	21	22	22	23	23
24	24	24	25	25	26	26	27	27
30	30	30	31	31	32	32	33	33
34	34	34	35	35	36	36	37	37

Умножение на константу «четыре» для второго разряда результата умножения представлено в табл. 2.1.2.

Простота этой таблицы и симметрия расположения цифровых данных в двухмерном цифровом пространстве координат первого и второго разрядов определяют простые геометрические образы сигналов фаз и сигналов переноса в третий разряд генератора кратностей.

Результат этой двойной операции показан графически на рис. 2.7 для сигналов $h^2_1 - h^2_4$ и цифр переноса P^2_H в третий разряд устройства.

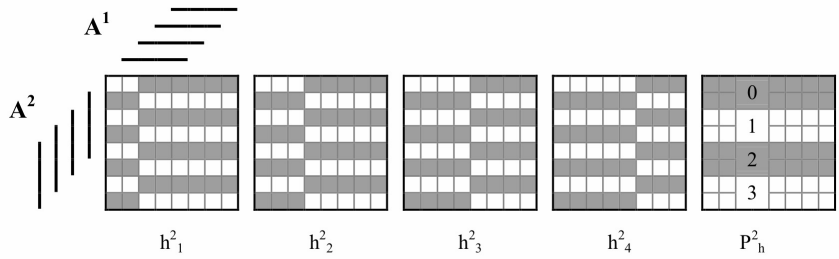


Рис. 2.7

Поскольку для умножения на константу «четыре» нет сквозного переноса через разряд, то можно сразу записать результат операции для k-го разряда устройства:

$$\mathbf{H}^k = \mathbf{M}^k_h \left| \begin{matrix} \mathbf{0} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{7} \end{matrix} \right|^{k}_{\mathbf{A}},$$

где

$$\mathbf{M}^k_H = \left| \begin{matrix} m_1 & \underline{m}_1 & m_1 & \underline{m}_1 & m_1 & \underline{m}_1 & m_1 & \underline{m}_1 \\ m_2 & \underline{m}_2 & m_2 & \underline{m}_2 & m_2 & \underline{m}_2 & m_2 & \underline{m}_2 \\ a_4 & \underline{a}_4 & a_4 & \underline{a}_4 & a_4 & \underline{a}_4 & a_4 & \underline{a}_4 \\ m_3 & \underline{m}_3 & m_3 & \underline{m}_3 & m_3 & \underline{m}_3 & m_3 & \underline{m}_3 \end{matrix} \right|^{k}_{\mathbf{H}}, \qquad (2.1.4)$$

а $m^k_1 = (\mathbf{2} \vee \ldots \vee \mathbf{7})^k$, $m^k_2 = (\mathbf{3} \vee \ldots \vee \mathbf{7})^k$, $m^k_3 = (\mathbf{5} \vee \ldots \vee \mathbf{7})^k$ и $h^{k+1}_1 = m^k_1 = (\mathbf{2} \vee \ldots \vee \mathbf{7})^k$, $h^{k+1}_2 = a^k_4$, $h^{k+1}_3 = (\mathbf{6} \vee \mathbf{7})^k$.

Умножение на константу «пять» для второго разряда умножения представляется данными табл. 2.1.3.

Таблица 2.1.3

	Перенос с $\mathbf{A}^1$							
$\mathbf{A}^2 \times 5$	**0**	**0**	**1**	**1**	**2**	**3**	**3**	**4**
00	0	0	1	1	2	3	3	4
05	5	5	6	6	7	10	10	11
12	12	12	13	13	14	15	15	16
17	17	17	20	20	21	22	22	23
24	24	24	25	25	26	27	27	30
31	31	31	32	32	33	34	34	35
36	36	36	37	37	40	41	41	42
43	43	43	44	44	45	46	46	47

Результат этой двойной операции показан графически на рис. 2.8, *а* для сигналов $f^2_1 - f^2_4$ и цифр сигнала переноса P^2_F в третий разряд устройства.

Выполнив такие же двойные операции в третьем разряде устройства, можно представить цифровые множества сигналов $f^2_1 - f^2_4$ и цифр сигнала переноса P^2_F в четвертый разряд устройства, что и показано на рис. 2.8, *б* в трехмерном цифровом пространстве.

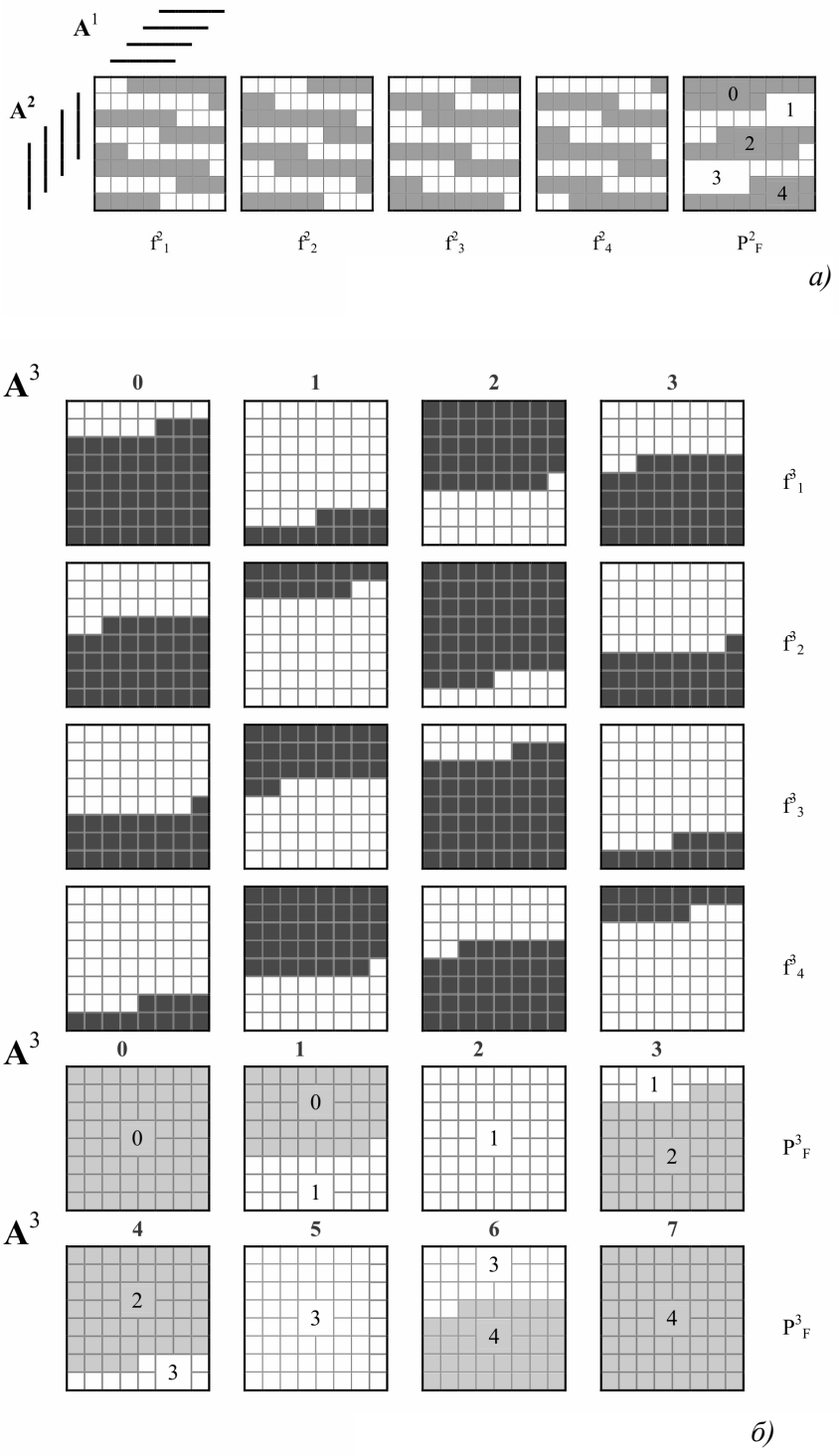

A1
A2
0
1
2
3
4
f2 1
f2 2
f2 3
f2 4
P2 F
а)
A3
0
1
2
3
f3 1
f3 2
f3 3
f3 4
A3
0
1
2
3
P3 F
A3
4
5
6
7
P3 F
б)

Рис. 2.8

Результат умножения для k-го разряда устройства в матричной форме записи:

$$\mathbf{F}^k = \mathbf{M}^k{}_f \left| \begin{matrix} \mathbf{0} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{7} \end{matrix} \right|^{k}_{\mathbf{A}},$$

где

$$\mathbf{M}^k{}_F = \left| \begin{matrix} m_1 & m_2 & m_3 & m_4 & \underline{m}_1 & \underline{m}_2 & \underline{m}_3 & \underline{m}_4 \\ m_4 & \underline{m}_1 & \underline{m}_2 & \underline{m}_3 & \underline{m}_4 & m_1 & m_2 & m_3 \\ \underline{m}_3 & \underline{m}_4 & m_1 & m_2 & m_3 & m_4 & \underline{m}_1 & \underline{m}_2 \\ m_2 & m_3 & m_4 & \underline{m}_1 & \underline{m}_2 & \underline{m}_3 & \underline{m}_4 & m_1 \end{matrix} \right|^{k}_{F}. \qquad (2.1.5)$$

Для k=2 $m_1 = (\mathbf{2} \vee \ldots \vee \mathbf{7})^1$, $m_2 = \mathbf{7}^1$, $m_3 = (\mathbf{0} \vee \ldots \vee \mathbf{4})^1$, $m_4 = a^1{}_4$; а для k=3 $m_1 = (\mathbf{2} \vee \ldots \vee \mathbf{7})^2 \vee (\mathbf{5} \vee \mathbf{6} \vee \mathbf{7})^1 a^2{}_1$, $m_2 = \mathbf{7}^2 \vee a^1{}_4 (\mathbf{5} \vee \mathbf{6})^2$, $m_3 = \underline{a}^2{}_4 \vee (\mathbf{0} \vee \ldots \vee \mathbf{6})^1 a^2{}_1$, $m_4 = a^2{}_4 \vee (\mathbf{2} \vee \ldots \vee \mathbf{7})^1 a^2{}_3$ и т.д.

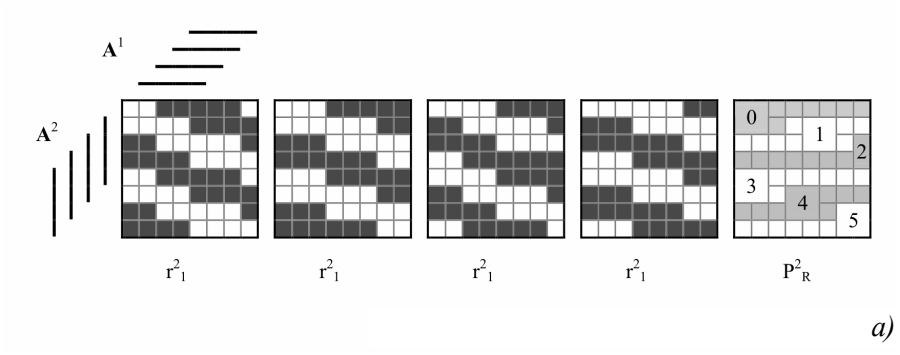
Результаты умножения на константу «шесть» для второго разряда определяются данными табл. 2.1.4.

Таблица 2.1.4

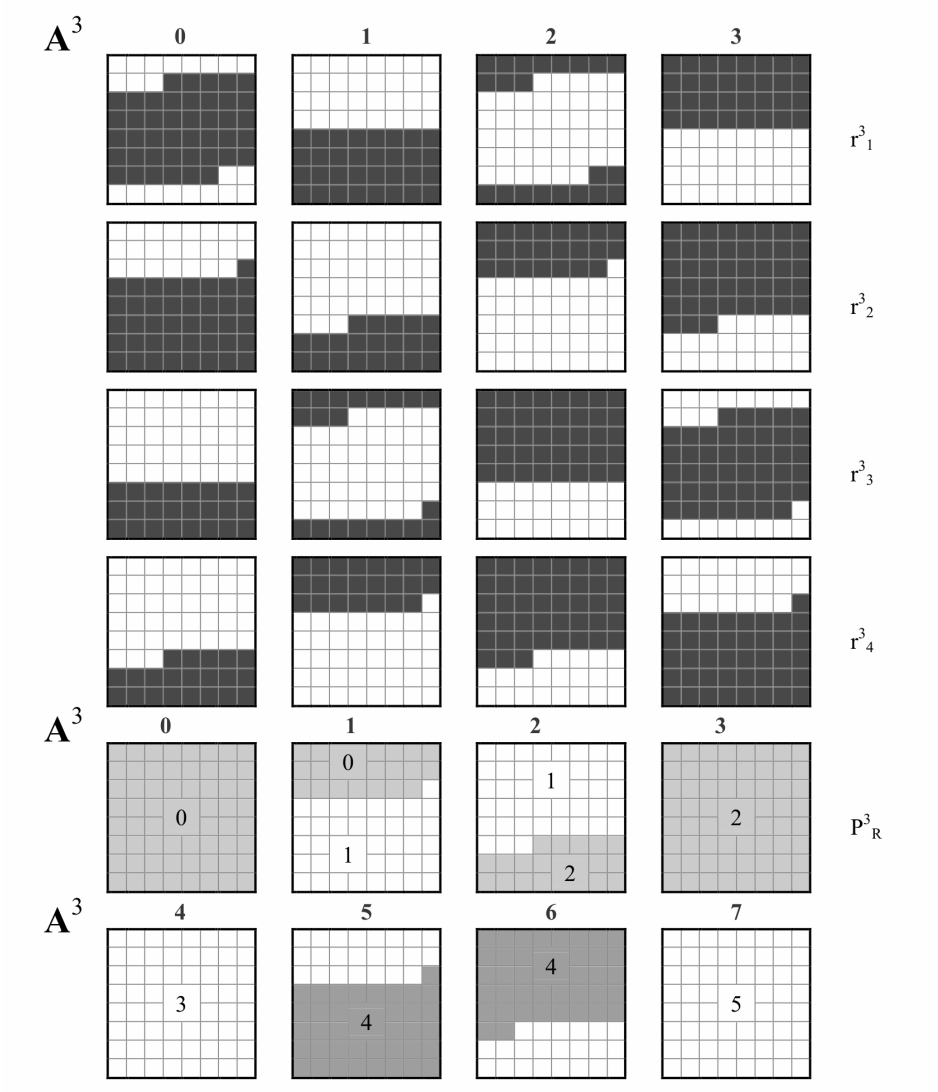
	Перенос с $\mathbf{A}^1$							
$\mathbf{A}^2 \times 6$	**0**	**0**	**1**	**2**	**3**	**3**	**4**	**5**
00	0	0	1	2	3	3	4	5
06	6	6	7	10	11	11	12	13
14	14	14	15	16	17	17	20	21
22	22	22	23	24	25	25	26	27
30	30	30	31	32	33	33	34	35
36	36	36	37	40	41	41	42	43
44	44	44	45	46	47	47	50	51
52	52	52	53	54	55	55	56	57

Результат этой двойной операции изображен графически на рис. 2.9, *а* для сигналов $r^2{}_1 - r^2{}_4$ и цифр сигнала переноса $P^3{}_R$ в третий разряд устройства.

Выполнив аналогичные двойные операции в третьем разряде, можно представить цифровые множества сигналов $r^2{}_1 - r^2{}_4$ и цифр сигналов переноса $P^3{}_R$ в четвертый разряд устройства, что приведено на рис. 2.9, *б* в трехмерном цифровом пространстве.


A1
A2
0
1
2
3
4
5
r21
r21
r21
r21
P2R

а)


A3
0
1
2
3
r31
r32
r33
r34
A3
0
1
2
3
P3R
A3
4
5
6
7

б)

Рис. 2.9

Результат умножения для k-го разряда устройства в матричной форме записи:

$$\mathbf{R}^k = \mathbf{M}^k_{\mathbf{R}} \begin{matrix} \mathbf{0} & k \\ \vdots & \\ \mathbf{7} & \mathbf{A} \end{matrix},$$

где

$$\mathbf{M}^k_{\mathbf{R}} = \left| \begin{matrix} m_1 & a^{k-1}{}_4 & \underline{m_1} & \underline{a}^{k-1}{}_4 & m_1 & a^{k-1}{}_4 & \underline{m_1} & \underline{a}^{k-1}{}_4 \\ m_2 & m_3 & \underline{m_2} & \underline{m_3} & m_2 & m_3 & \underline{m_2} & \underline{m_3} \\ a^{k-1}{}_4 & \underline{m_1} & \underline{a}^{k-1}{}_4 & m_1 & a^{k-1}{}_4 & \underline{m_1} & \underline{a}^{k-1}{}_4 & m_1 \\ m_3 & \underline{m_2} & \underline{m_3} & m_2 & m_3 & \underline{m_2} & \underline{m_3} & m_2 \end{matrix} \right| \begin{matrix} k \\ \\ \\ \mathbf{R} \end{matrix}. \qquad (2.1.6)$$

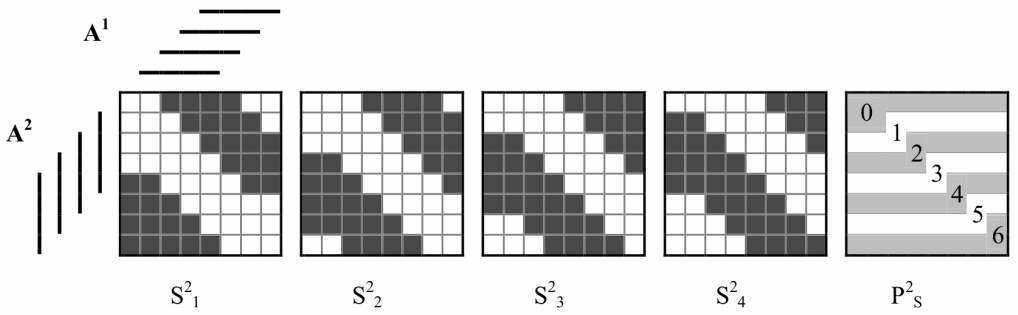
Для k= 2 $m_1 = (\mathbf{2} \vee \ldots \vee \mathbf{6})^1$, $m_2 = (\mathbf{3} \vee \ldots \vee \mathbf{7})^1$, $m_3 = (\mathbf{6} \vee \mathbf{7})^1$; а для k= 3 $m_1 = (\mathbf{2} \vee \ldots \vee \mathbf{6})^2 (\mathbf{1} \vee \ldots \vee \mathbf{5})^1 \vee (\mathbf{1} \vee \ldots \vee \mathbf{5})^2 (\mathbf{3} \vee \ldots \vee \mathbf{7})^1$, $m_2 = (\mathbf{3} \vee \ldots \vee \mathbf{7})^2 \vee \mathbf{7}^1 a^2{}_2$, $m_3 = (\mathbf{7} \vee \mathbf{6})^2 \vee (\mathbf{3} \vee \ldots \vee \mathbf{7})^1 (\mathbf{5} \vee \mathbf{6} \vee \mathbf{7})^2$ и т.д.

Умножение на константу «семь» для второго разряда умножения представляется данными табл. 2.1.5.

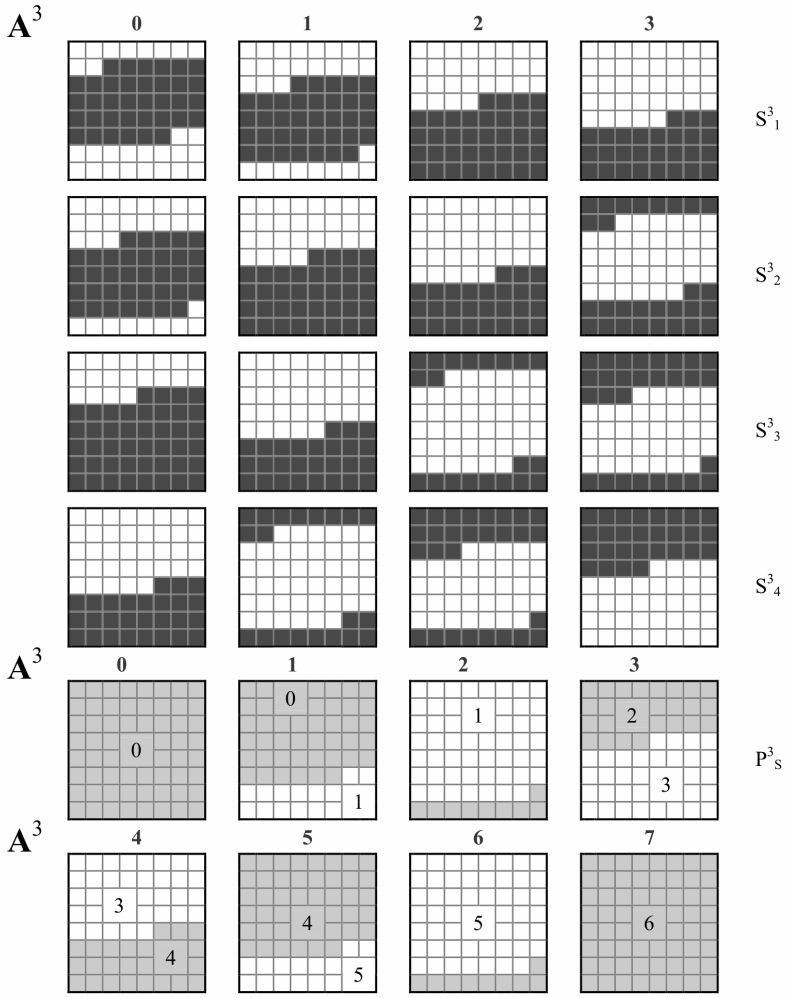
Таблица 2.1.5

	Перенос с A^1							
$\mathbf{A}^2 \times 7$	**0**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
00	0	0	1	2	3	4	5	6
07	7	7	10	11	12	13	14	15
16	16	16	17	20	21	22	23	24
25	25	25	26	27	30	31	32	33
34	34	34	35	36	37	40	41	42
43	43	43	44	45	46	47	50	51
52	52	52	53	54	55	56	57	60
61	61	61	62	63	64	65	66	67

Результат этой двойной операции изображен графически на рис. 2.10, *а* для сигналов $s^2{}_1 - s^2{}_4$ и цифр переноса $P^2{}_S$ в третий разряд устройства.


A1
A2
0
1
2
3
4
5
6
S21
S22
S23
S24
P2S

а)


A3
0
1
2
3
S31
S32
S33
S34
A3
P3S
A3
4
5
6
7

б)

Рис. 2.10

Выполняя такие же операции в третьем разряде, получаем цифровые множества сигналов $s^3_1 - s^3_4$ и переноса P^3_S в четвертый разряд устройства, что изображено на рис. 2.10, *б* в трехмерном цифровом пространстве.

Результат аналогичных действий для k-го разряда устройства можно записать в матричной форме:

$$\mathbf{S}^{\mathbf{k}} = \mathbf{M}^{\mathbf{k}}_{\mathbf{S}} \begin{matrix} \mathbf{0} & k \\ \vdots & \\ \mathbf{7} & A \end{matrix},$$

где

$$\mathbf{M}^k_S = \left| \begin{matrix} m_1 & m_2 & m_3 & m_4 & \underline{m_1} & \underline{m_2} & \underline{m_3} & \underline{m_4} \\ m_2 & m_3 & m_4 & \underline{m_1} & \underline{m_2} & \underline{m_3} & \underline{m_4} & m_1 \\ m_3 & m_4 & \underline{m_1} & \underline{m_2} & \underline{m_3} & \underline{m_4} & m_1 & m_2 \\ m_4 & \underline{m_1} & \underline{m_2} & \underline{m_3} & \underline{m_4} & m_1 & m_2 & m_3 \end{matrix} \right| \begin{matrix} k \\ \\ \\ S \end{matrix}. \qquad (2.1.7)$$

Для k= 2 $m_1=a^1_2$, $m_2=a^1_3$, $m_3=a^1_4$, $m_4=(\mathbf{5} \vee \mathbf{6} \vee \mathbf{7})^1$; а для k= 3 $m_1= a^2_2\,(\mathbf{0} \vee \ldots \vee \mathbf{5})^1 \vee a^2_1\,(\mathbf{2} \vee \ldots \vee \mathbf{7})^1$, $m_2= a^2_3\,(\mathbf{0} \vee \ldots \vee \mathbf{6})^1 \vee a^2_2\,(\mathbf{3} \vee \ldots \vee \mathbf{7})^1$, $m_3= a^2_4 \vee a^1_4\, a^2_3$, $m_4= (\mathbf{5} \vee \mathbf{6} \vee \mathbf{7})^2 \vee (\mathbf{5} \vee \mathbf{6} \vee \mathbf{7})^2\, a^2_4$ и т.д.

Анализ геометрических образов выходных сигналов синтезированного нами генератора кратностей при умножении входного операнда на нечетные числа «три», «пять», «семь» показывает, что эти образы на второй половине многомерного цифрового пространства равны инверсии этих образов первой половины этого пространства.

При умножении входного операнда на четные числа «два», «четыре», «шесть» геометрические образы выходных сигналов на первой и второй половине пространства одинаковые.

Это нашло отражение в формулах (2.1.2) – (2.1.7), определяющих покрытия этих геометрических образов логических функций выходных сигналов генератора кратностей.

При синтезе генератора кратностей с другими основаниями систем счисления будет меняться ансамбль цифровых множеств, формирующих выходные сигналы разрядов, но принципы подхода к его синтезу, предложенные в [3], останутся такими же, как представлено выше. При этом возможно не только двухступенчатое формирование выходных сигналов устройства, но и одноступенчатое, когда в формулах (2.1.3) – (2.1.7) цифровые множества m_i определяются непосредственно из входных сигналов разрядов операнда **A**, что дает возможность получить максимальное быстродействие генератора кратностей.

В связи с тем, что метод кратных множимого обладает максимально возможным быстродействием [26], применение в нем быстродействующих генератора кратностей и многовходовых сумматоров позволяет достигнуть предельного быстродействия устройств умножения. Таким же предельным быстродействием обладает предложенный нами метод кратных делимого, который будет рассмотрен в следующем разделе, и в нем также используются генератор кратностей и многовходовые суммирующие (вычитающие) устройства. Поэтому его быстродействие непосредственно зависит от быстродействия этих устройств.

Использование при этом бóльших оснований систем счисления, чем приведено в нашем примере, позволяет не только увеличить быстродействие операции умножения и деления, но и повысить контролеспособность цифровых блоков, реализующих эти операции.

Контролеспособность рассматриваемых нами устройств обеспечивается использованием в них многофазных кодов либо кодов многомерных угловых цифровых множеств, которые, кроме того, позволяют создавать многовходовые быстродействующие сумматоры. Эти многовходовые сумматоры и генераторы кратностей являются неотъемлемой частью устройств умножения и деления и поэтому определяют их контролеспособность и быстродействие.

После прочтения этого раздела заинтересованный читатель может удовлетворить свою любознательность с помощью самостоятельного решения задач по синтезу таких устройств, где используются различные основания систем счисления и принципы их кодирования. При этом он оценит истину и красоту предмета синтеза устройств умножения, состоящих из генератора кратностей и многовходовых сумматоров.

2.2. Синтез устройств деления

Хотя операция деления не является столь распространенной в системах управления электроприводами, как суммирование или умножение, ее освещение представляет не только теоретический интерес.

Из известных технических решений можно выделить устройства для деления, работающие в двоичном коде и реализующие методы деления с восстановлением и без восстановления остатка, и устройства, реализующие деление «обходным путем» с использованием умножения или иной процедуры [23].

Все эти устройства деления двоичного кода основания $n = 2$ не позволяют получить высокого быстродействия и значительно уступают быстродействию, например операции умножения, которая представляет собой завершенную логическую схему без элементов памяти и сдвига. Применение этих устройств в системах с основанием больше двух потребует преобразований из одного кода в другой, что также уменьшает быстродействие операции деления. Кроме того, такой принцип построения устройств деления нецелесообразен из-за перехода к неконтролеспособной системе счисления, какой является двоичная система счисления.

Устройство для деления с основанием системы счисления больше двух [19], где деление выполняется через умножение делимого на число, обратное делителю, которое находится итерационным способом, может быть использовано и для многофазного принципа кодирования основания системы счисления. Однако оно отличается значительной сложностью и невысоким быстродействием вследствие необходимости выполнения операции полного деления в несколько циклов.

Классический метод деления без восстановления остатка изложен в [17]. В этой работе авторов с мировыми именами описан принцип построения делителя для системы счисления с основанием **n**, где сформулирован принцип обработки остатка ($\mathbf{B_i}$): если $\mathbf{B_i} \geq 0$ ($\mathbf{B_i} < 0$), то из $\mathbf{B_i}$n многократно вычитают (прибавляют) делитель **C** до тех пор, пока знак разности (суммы) станет противоположным знаку величины $\mathbf{B_i}$ или пока количество произведенных сложений (вычитаний) не сделается равным (n – 1).

Устройство [29], реализующее этот способ деления для системы счисления n=**2**, выполнено в виде итеративной сети матрицы, составленной из одинаковых блоков, образующих L строк по числу разрядов частного, где в каждой строке содержатся m блоков по числу разрядов делителя, блок сигнала опережающего переноса и логический блок. Для ускорения операции сложения или вычитания каждый остаток $\mathbf{B_i}$ в каждой строке вычисляется в виде двух чисел: числа **S**, составленного из поразрядных сумм $\mathbf{s_j}$, и числа E, составленного из поразрядных переносов e_j, где перенос в знаковый разряд строки, очередной разряд частного и управляющий сигнал для следующей строки (сложить или вычесть) формируются при помощи схемы опережения переносов. По сигналу блока переноса осуществляется операция сложения либо вычитания в блоках строк матрицы, где переход от операции вычитания к операции сложения осуществляется переводом делителя из прямого кода в дополнительный.

Аппаратурная сложность устройства при его реализации для систем счисления с основанием больше двух приведет к еще большему усложнению и потере быстродействия. Это связано с тем, что в каждой строке необходимо многократно осуществлять операцию сложения либо вычитания (для двоичного кода это выполняется один раз).

Вместе с тем классическое, известное из общеобразовательной школы деление «углом», где элементарная операция деления выполняется последовательным подбором результата, может быть реализована без этого перебора, а непосредственно сразу с максимальным быстродействием [4, 5]. Структурная схема такого устройства приведена на рис. 2.11.

Синтез устройства проведем на примере деления двух чисел **A** и **B** (**A** – делитель, **B** – делимое), представленных в пятифазном коде (n=10). Примем, что оба операнда заданы в нормализованном виде, когда в старших разрядах содержится сигнал, отличающийся от цифры 0. Пусть делимое содержит семь разрядов, делитель содержит три разряда, а результат деления выдается в семи разрядах.

Входные шины трех разрядов делителя $\mathbf{A(A^1, A^2, A^3)}$ соединены с входом генератора кратностей, выходные шины которого содержат кратные делителя: **A**×1, **A**×2=**C**, **A**×3=**D**, **A**×4=**E**, **A**×5=**F**, **A**×6=**G**, **A**×7=**H**, **A**×8=**J**, **A**×9=**L**.

Цифровой блок первой строки осуществляет деление числа, заданного старшими разрядами делимого $\mathbf{B^7}$, $\mathbf{B^6}$, $\mathbf{B^5}$, на делитель и результат деления, который

является целым числом, выдается на первый выход $\mathbf{X}^7$ цифрового блока строки, а остаток от деления поступает на второй трехразрядный выход этого блока.

Второй блок второй строки осуществляет деление числа, старшие разряды которого задаются остатком от предыдущего деления, а младший разряд является числом следующего разряда $\mathbf{B}^4$ делимого, на делитель, и результат деления выдается на второй выход $\mathbf{X}^6$ цифрового блока строки, а остаток от деления поступает на второй трехразрядный выход этого блока и т.д.

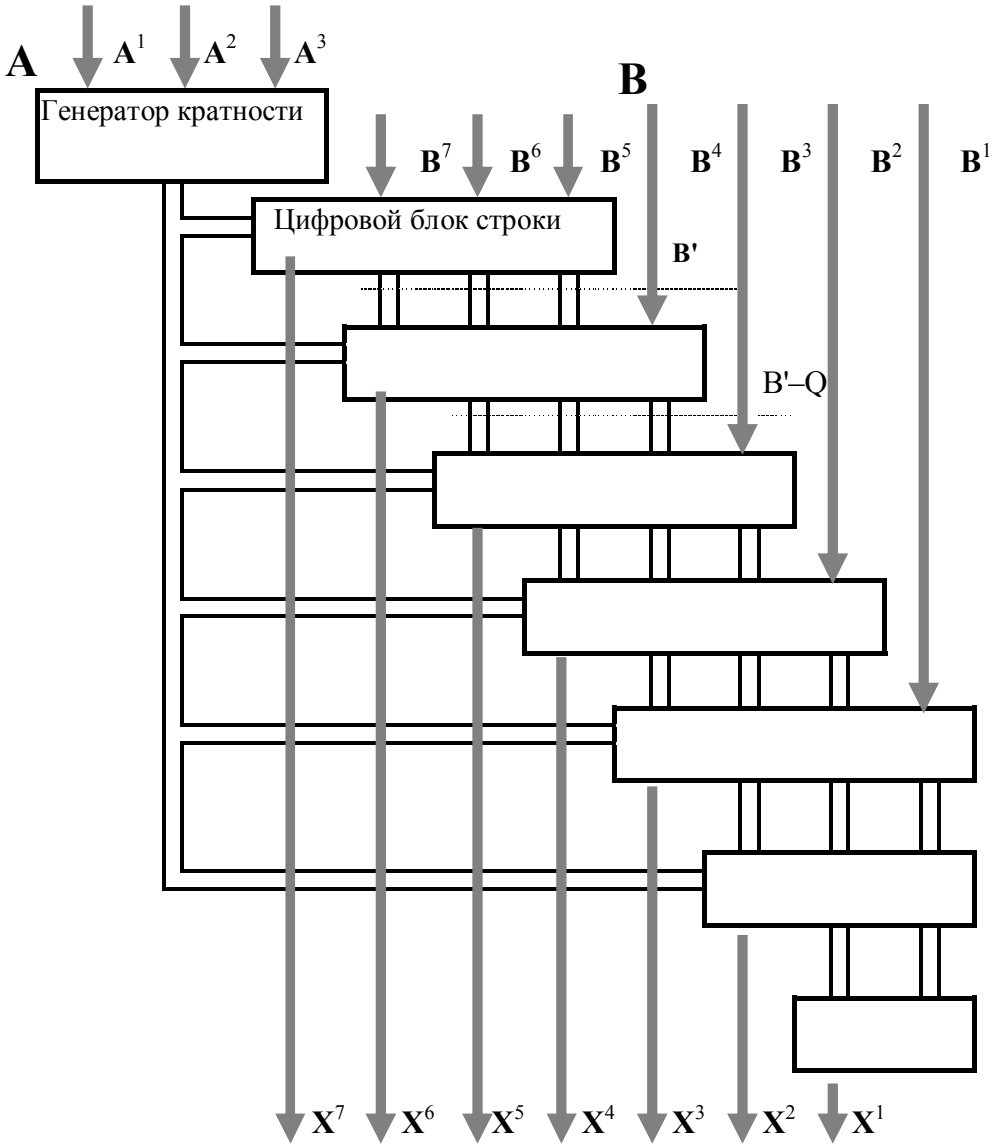


Рис. 2.11

В представленной части работа схемы принципиально не отличается от общеизвестного деления «углом». Отличие здесь в осуществлении элементарной операции деления в цифровом блоке строки, где деление производится не последовательным подбором результата, а выдается сразу.

На рис. 2.12 приведена структурная схема цифрового блока строки. Число **B'** на первом входе цифрового блока строки сравнивается в первом узле сравнения с числом **A** и **A×6** и при выполнении неравенства **A ≤ B'** < **A×6** [6] на выходной шине x_1 появляется сигнал x_1=1*. Во втором узле сравнения число **B'** сравнивается с числами **A×2** и **A×7**, и при выполнении неравенства **A×2 ≤ B'** < **A×7** на выходной шине $\mathbf{x_2}$ появляется сигнал x_2=1* и т.д. вплоть до пятого узла сравнения, когда при выполнении соотношения **A×5 ≤ B'** появляется сигнал x_5=1*.

На рис. 2.13 приведены соотношения между цифровыми значениями выходных сигналов **A×1**, … , **A×9** генератора кратности, пятифазными сигналами **X** (x_1, … ,x_5) и их цифровыми эквивалентами **0–9**.

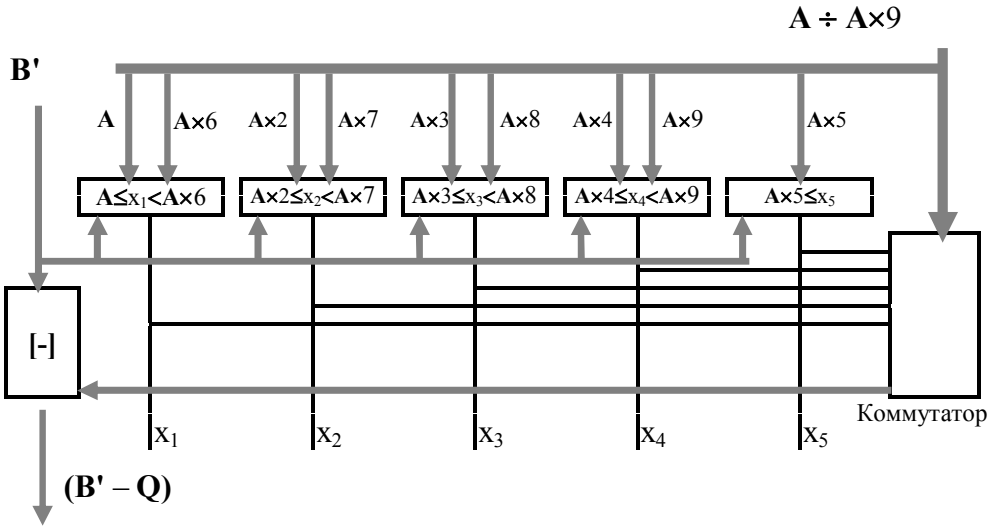


Рис. 2.12

Следовательно, на первом выходе цифрового блока строки формируется результат деления числа **B** на делитель **A**($\mathbf{A^1}$, $\mathbf{A^2}$, $\mathbf{A^3}$). Цифровой эквивалент этого деления $\mathbf{X_i}$ через коммутатор пропускает число **Q=A×**$\mathbf{X_i}$ на второй вход узла вычитания, где определяется разность чисел (**B'** – **Q**), которая является вторым выходом цифрового блока строки, и т.д.

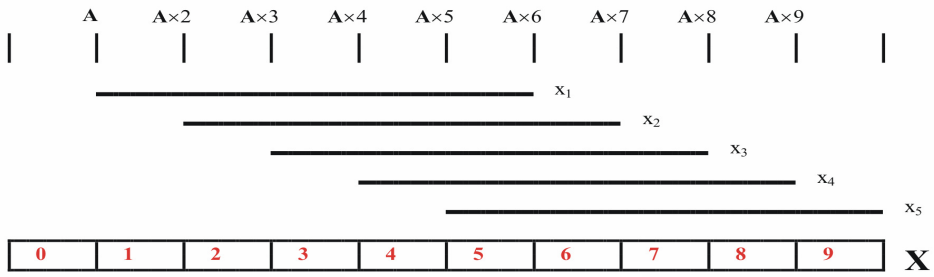


Рис. 2.13

Таким образом, синтезированное устройство для деления благодаря параллельному выполнению операции элементарного деления в цифровом блоке строки, где результат операции формируется без итеративного подбора, позволяет получить высокое быстродействие. При этом быстродействие операции деления эквивалентно операции умножения, реализующее метод кратных множимого, который, как известно [27], является одним из наиболее сильных аппаратных методов.

Выполняя все представленные выше процедуры синтеза, можно создать устройство деления для любых оснований систем счисления и любых принципов кодирования этих оснований.

Одним из таких устройств является схема [5], где принят двоичный принцип кодирования основания n=8. Работа этой схемы поясняется рис. 2.14.

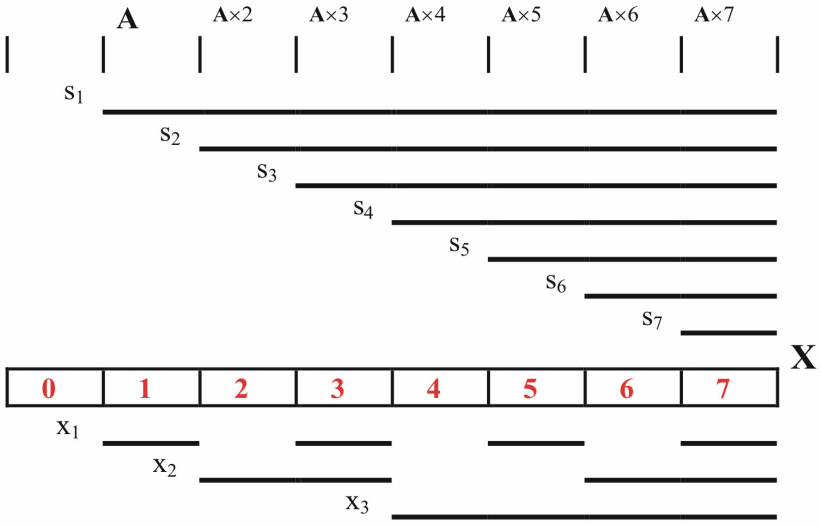


Рис. 2.14

Здесь в цифровом блоке строки используется промежуточный «интегральный» код **S**(s_1, ... , s_7), сигналы которого определяются в соответствующих узлах сравнения по следующим неравенствам: $\mathbf{A} \le s_1$, $\mathbf{A} \times 2 \le s_2$, ... , $\mathbf{A} \times 7 \le s_7$, а связь между этими сигналами и выходными определяется простыми зависимостями:

$$x_1 = s_1 \underline{s_3} \vee s_3 \underline{s_4} \vee s_5 \underline{s_6} \vee s_7; \quad x_2 = s_2 \underline{s_4} \vee s_6; \; x_3 = s_4.$$

2.3. Синтез реверсивных счетчиков

Схемы счетчиков и делителей частоты находят самое широкое применение в электроприводах и системах энергоснабжения, где они используются при формировании многофазных напряжений, обработке сигналов с датчиков обратных связей и в целом ряде других цифровых узлов.

Структура многофазного кода позволяет создавать достаточно простые схемы счетчиков, которые могут быть построены на различных типах триггеров: RS, JK, D, DV и других. Схемы делителей-счетчиков отличаются многообразием вариантов и названий [28, 30, 20, 18], из которых наиболее часто встречаются названия «счетчик лента Мебиуса», «счетчик с кодом Либау – Крейга», «счетчик на регистре с перекрестными связями», «кольцевой счетчик»,

«кодовое кольцо», т.е. названию счетчика приписывают фамилию автора счетчика либо автора кода, либо сам код.

Здесь необходимо внести ясность. Многофазный код ведет свою историю с начала XIX в., когда появились в электротехнике многофазные напряжения, питающие электродвигатели переменного тока. Если в многофазных напряжениях положительную полуволну принять за логическую единицу, а отрицательную – за логический нуль, то при нечетном числе фаз сдвиг между сигналами многофазного кода равен $2\pi/m$, а при четном числе фаз – π/m.

W.H. Libaw и L.J. Craig в предложенной ими схеме преобразователя угол – код применили сдвиг фаз между сигналами для нечетного числа фаз при m=5 таким же, как для четного числа фаз. Таким образом, код Либау – Крейга отличается от многофазного кода и кодов счетчиков [7–13] только сдвигом фаз между его сигналами: код Либау – Крейга – π/m, многофазный код – $2\pi/m$, а код, предложенный в [7] для применения в кольцевых управляемых по внешнему цифровому сигналу делителях-счетчиках, имеет сдвиг фаз – $2\pi(m–1)/m$.

На рис. 2.15 приведены соотношения между сигналами этих кодов соответственно для m=3, m=5 и m=4. Любой из этих кодов исторически правильно называть многофазным, а при необходимости уточнения указывать угол между фазами. При этом следует отметить, что почти во всех литературных источниках не делается отличий в выполнении счетчиков многофазного кода при четном и нечетном числе фаз. Однако счетчики на четном и нечетном числе триггеров имеют различную структуру и достоверность функционирования. Например, счетчик трехфазного кода (m=3) имеет $(2^m – 2)=2$ нерабочих состояния, в которых он может находиться сколь угодно долго, а счетчик четырехфазного кода, у которого восемь нерабочих состояний, не имеет устойчивого нерабочего цикла.

Основная проблема, которую приходится решать при синтезе делителей-счетчиков многофазного кода, – устранение нерабочих состояний. Число этих состояний зависит от основания системы счисления и совпадает с избыточностью кода, в котором работает счетчик. Чем больше фазность делителя-счетчика, тем больше у него устойчивых нерабочих состояний и тем сложнее обеспечить его устойчивость.

На рис. 2.16 приведена схема трехфазного делителя-счетчика, выполненного на RS-триггерах, которые соединены между собой вентилями в два кольца. Первое кольцо образовано вентилями, которые соединяют левые плечи триггеров с входами последующих триггеров. Второе кольцо образуют вентили связи, которые соединяют правые плечи триггеров с входами последующих триггеров. На управляющие входы вентилей связи первого и второго кольца подаются соответственно тактовые импульсы α и β. Если триггеры счетчика находятся в состоянии, соответствующем цифре **0** (см. рис. 2.16), то тактовый импульс α установит первый триггер в противоположное положение с третьим триггером, что соответствует цифре **1**, следующий тактовый импульс β установит второй триггер в противоположное положение с первым триггером, что соответствует цифре **2**, и т.д.

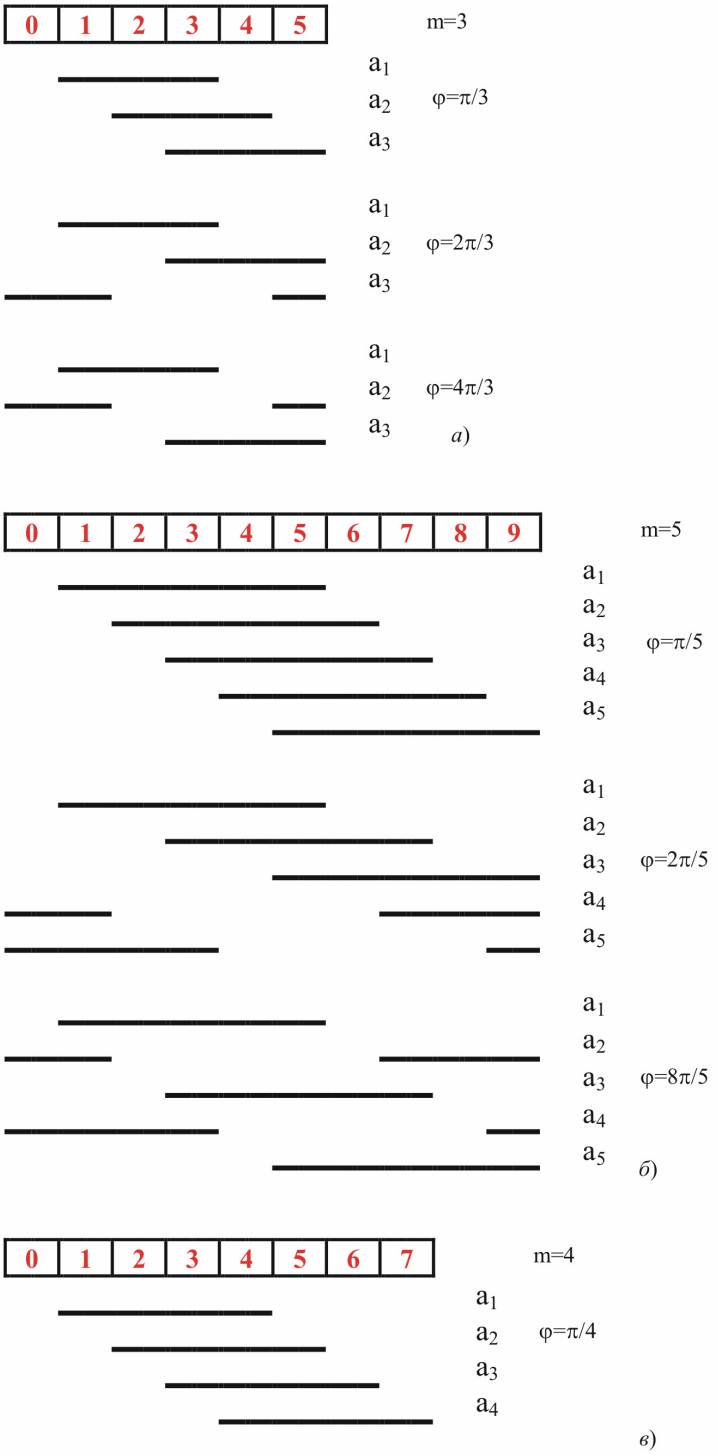

0 1 2 3 4 5
m=3
a1
a2
a3
φ=π/3
a1
a2
a3
φ=2π/3
a1
a2
a3
φ=4π/3
а)
0 1 2 3 4 5 6 7 8 9
m=5
a1
a2
a3
a4
a5
φ=π/5
a1
a2
a3
a4
a5
φ=2π/5
a1
a2
a3
a4
a5
φ=8π/5
б)
0 1 2 3 4 5 6 7
m=4
a1
a2
a3
a4
φ=π/4
в)

Рис. 2.15

Этот трехфазный делитель-счетчик имеет два устойчивых нерабочих состояния: первое состояние $a_1=a_2=a_3=0^*$ соответствует одновременному появлению на выходе четных цифровых сигналов **0**, **2**, **4**, а второе состояние $a_1=a_2=a_3=1^*$ – одновременному появлению нечетных цифр **1**, **3**, **5**.

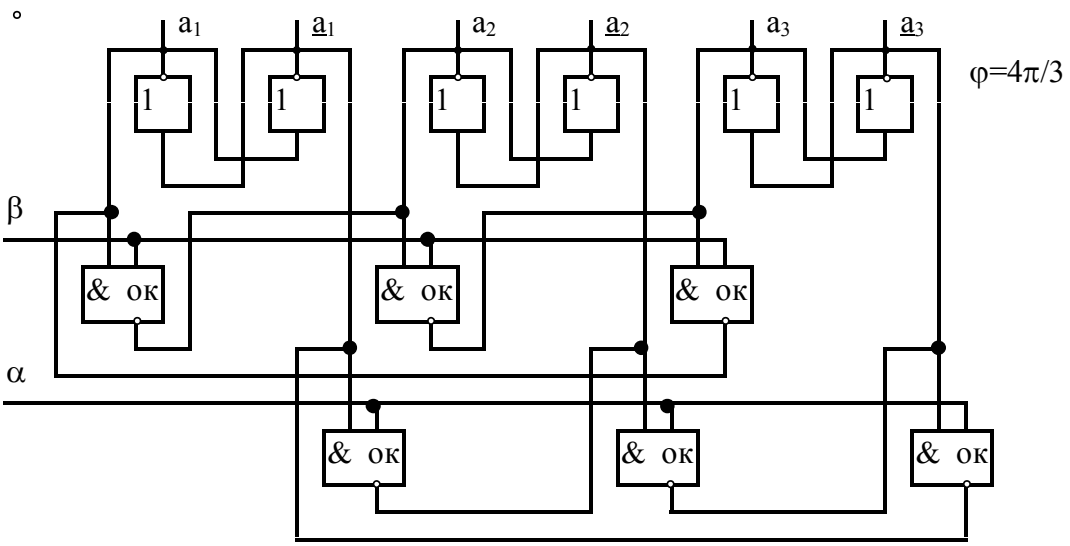


Рис. 2.16

Переход из нерабочих состояний в рабочие, а также установление устойчивых нерабочих циклов определяются временными соотношениями между переключениями триггеров. Если обозначить времена переключения триггеров соответственно t_1, t_2, t_3, предполагая, что время переключения каждого триггера из 0* в 1* и из 1* в 0* одинаково, то можно выделить следующие тринадцать соотношений: $\tau_1(t_1< t_2< t_3)$, $\tau_2(t_2< t_3< t_1)$, $\tau_3(t_3< t_1< t_2)$, $\tau_4(t_2< t_3< t_1)$, $\tau_5(t_2< t_1< t_3)$, $\tau_6(t_1< t_3< t_2)$, $\tau_7(t_1< t_2< t_3)$, $\tau_8(t_1=t_2< t_3)$, $\tau_9(t_2= t_3< t_1)$, $\tau_{10}(t_1< t_2=t_3)$, $\tau_{11}(t_2< t_3= t_1)$, $\tau_{12}(t_3< t_2= t_1)$, $\tau_{13}(t_1= t_2= t_3)$.

На рис. 2.17 приведен граф переходов трехфазного делителя-счетчика, на котором состояния счетчика обозначены эквивалентными им цифрами **0**–**5**. Из графа переходов видно, что при определенных временных соотношениях делитель-счетчик имеет устойчивый нерабочий цикл.

С увеличением числа фаз растут количество устойчивых нерабочих циклов и число состояний в этих циклах. Для устранения нерабочих состояний требуется применение специальных мер.

Рассмотрим построение делителей-счетчиков на четном числе триггеров.

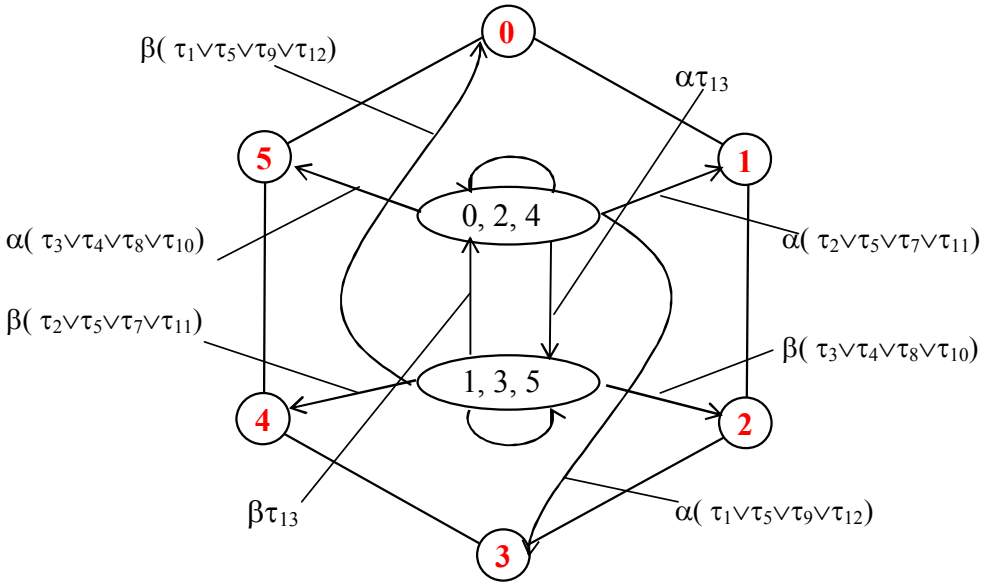


Рис. 2.17

На рис. 2.18 приведена схема четырехфазного делителя-счетчика, которая содержит четыре RS-триггера, соединенных между собой вентилями связи.

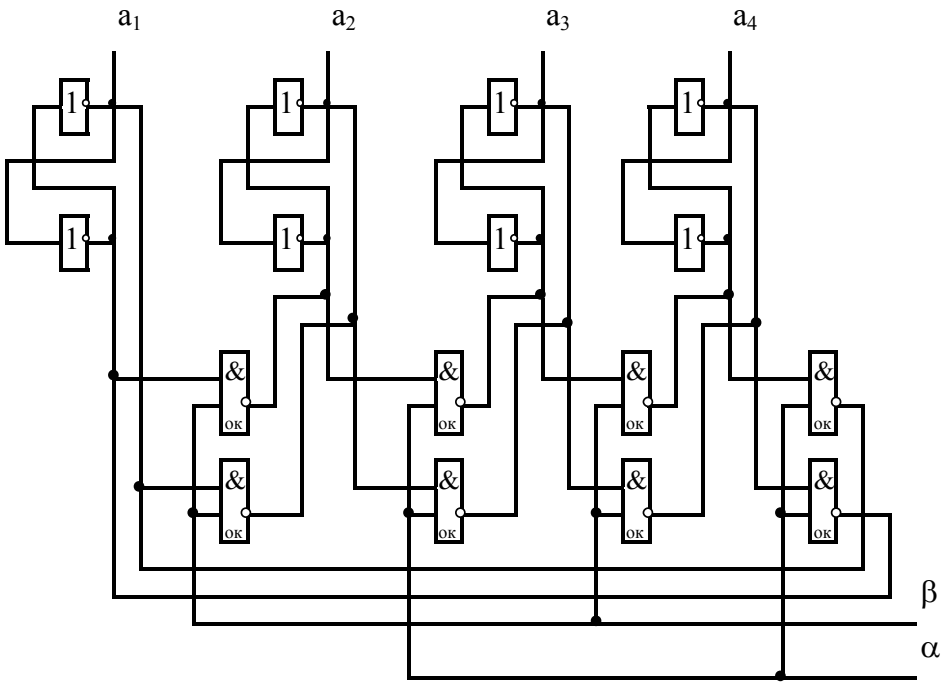


Рис. 2.18

Представленный счетчик не имеет замкнутых колец, а вентили образуют две группы: на управляющие входы вентилей первой группы подается тактовый импульс α, а на управляющие входы второй группы – импульс β. Если триггеры находились в состоянии, соответствующем цифре **0** (см. рис. 2.15, *в*), то тактовый импульс α установит первый триггер в положение, противоположное четвертому триггеру, что соответствует цифре **1**. Следующий тактовый импульс β установит второй триггер в одинаковое положение с первым, что соответствует цифре 2, и т.д.

Каждый тактовый импульс устанавливает один из триггеров в одинаковое положение с предыдущим триггером. Исключение составляет первый триггер, который устанавливается в противоположное положение с четвертым триггером.

Из графа переходов четырехфазного делителя-счетчика (рис. 2.19) видно, что схема не имеет устойчивых нерабочих состояний и переход из нерабочих состояний в рабочее состояние не зависит от соотношения времен переключения триггеров.

Однако уже следующая по количеству фаз схема на четном числе триггеров (шестифазный делитель-счетчик) имеет устойчивые нерабочие состояния и сложный граф состояния [24], где выход из нерабочего состояния без принятия специальных мер невозможен.

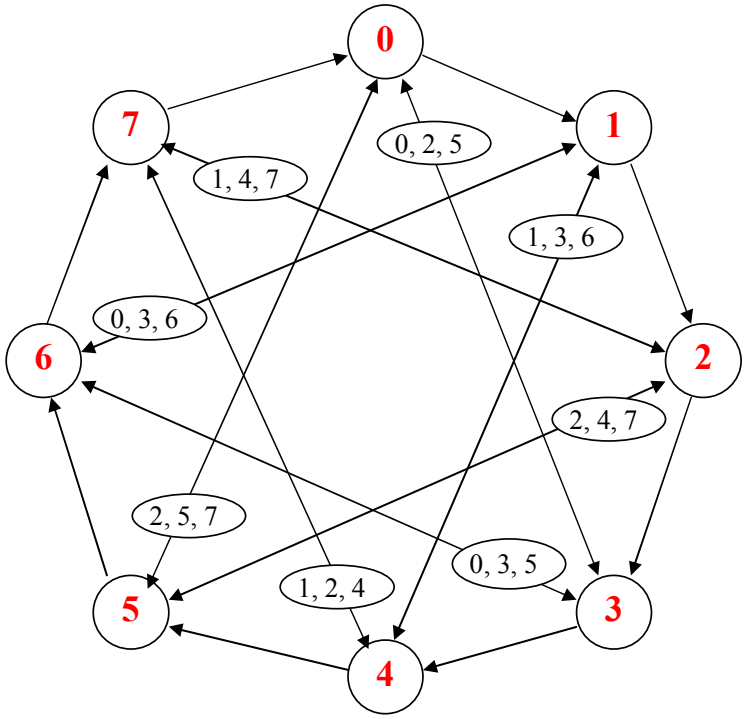


Рис. 2.19

Устранить устойчивые нерабочие состояния делителей-счетчиков можно увеличением числа тактовых импульсов. Например, шестифазный делитель-счетчик при трехтактовых импульсах не имеет устойчивых нерабочих циклов [25]. При таком способе построения структуры выход из нерабочего состояния осуществляется в несколько тактовых импульсов, что не всегда приемлемо.

Другой предложенный в [13] способ ликвидации нештатных состояний – использование блокировки триггеров делителя-счетчика определенными цифровыми сигналами. Сущность работы такой схемы поясняется рис. 2.20, *а*, где представлены трехфазные сигналы a_1, a_2, a_3 делителя-счетчика и соответствующие им цифры **0**–**5**, а звездочками отмечены соответствующие блокировки триггеров.

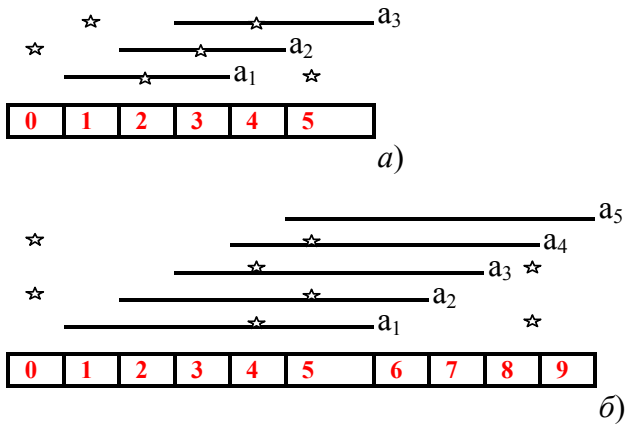


Рис. 2.20

При цифре **0** второй триггер устанавливается в положение 0*, при цифре **1** третий триггер устанавливается в положение 0*, а при цифре **2** первый триггер устанавливается в положение 1* и т.д. Граф переходов такого делителя-счетчика содержит только рабочие состояния, соответствующие цифрам **0**–**5**.

При увеличении числа фаз блокировки триггеров от всех цифр использовать нельзя, поскольку это приводит к возникновению состязаний в схеме делителя-счетчика. Здесь возможно использовать только частичную блокировку.

На рис. 2.20, *б* показаны сигналы a_1, ..., a_5 триггеров пятифазного делителя-счетчика и соответствующие им цифры **0**–**9**. При цифре **0** второй и четвертый триггеры устанавливаются в положение 0*, а при цифре **4** – первый и третий триггеры устанавливаются в положение 1* и т.д.

Использование таких частичных блокировок исключает опасные состязания в схеме пятифазного делителя-счетчика, что иллюстрируется графом переходов

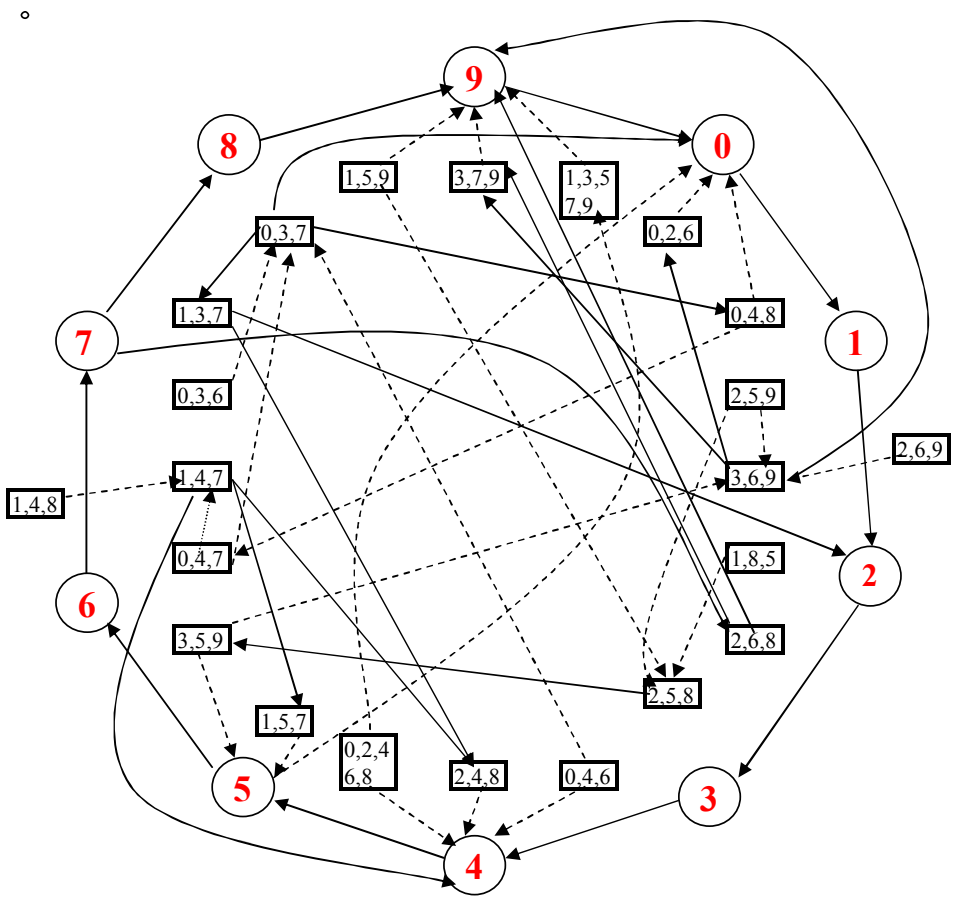


Рис. 2.21

на рис. 2.21, где пунктирными линиями обозначены возможные переходы при описанных блокировках. Введение блокировок ликвидирует устойчивые нерабочие состояния, но переход из нерабочих состояний в рабочие может осуществляться в течение нескольких тактов входных импульсов, что является недостатком схемы.

Ни один из рассмотренных методов не обеспечивает решения задачи устойчивости делителей-счетчиков многофазного кода в общем случае. Решение проблемы устойчивости заключается во включении в обратную связь блока исправления ошибок, где выходы триггеров в m-фазном делителе-счетчике соединяются с блоком исправления ошибок на входе устройства [14]. В пределах корректирующей способности блок исправления может исправлять ошибки определенной кратности, а также комбинации ошибок. Если имеется информация о вероятностях возникновения ошибок разной кратности, то этот блок целесообразно ориентировать на наиболее вероятный тип ошибок, поскольку выполнить его для возможного максимального количества ошибок достаточно сложно. Принципы синтеза блоков исправления ошибок будут рассмотрены в следующей главе.

В электроприводах широко используются реверсивные делители-счетчики, поэтому необходимо рассмотреть построение таких устройств в многофазном коде.

Имеются три основных способа реверсирования счетчиков в цифровой технике. Первый из них заключается в использовании счетчика с двумя входами. Подаче тактовых импульсов на один вход соответствует одно направление счета, а изменение направления счета происходит при подаче тактовой частоты на другой вход. При втором способе счетчик имеет шину тактовой частоты и шины суммирования и вычитания. На первую шину подаются тактовые импульсы, а на одну из остальных шин – разрешающий сигнал.

Оба способа имеют невысокую помехоустойчивость, поскольку при переключении тактовых сигналов с одного входа на другой, а также переключении сигнала с шины суммирования на шину вычитания либо наоборот в счетчике имеется вероятность сбоя. По этой причине нежелательно использование таких реверсивных делителей-счетчиков в системах управления электроприводами, где переключение с режима на режим происходит довольно часто.

Более надежным в этом отношении является разработанный нами [7–9, 11, 12, 15] третий способ осуществления реверса, который заключается в изменении порядка следования тактовых импульсов. Пусть имеются четыре последовательности тактовых импульсов α, β, α', β', диаграммы которых показаны на рис. 2.22, *а*, *б*.

Они подаются на делитель-счетчик по четырем шинам. Построение такого делителя-счетчика на любом числе триггеров не вызывает затруднений. Для этого он должен иметь не два, а четыре рабочих кольца, причем при порядке следования тактовых импульсов (α, β, α', β', α и т.д.), показанном на рис. 2.22, *а*, рабочими являются кольца, в которые первыми поступают импульсы α и α'. Переключение триггеров в делителе-счетчике осуществляется в порядке возрастания эквивалентных цифр. При изменении порядка следования импульсов, показанном на рис. 2.22, *б*, рабочими становятся кольца, в которых подаются импульсы β и β', и переключение триггеров осуществляется в порядке убывания цифр.

Примером реверсивного делителя-счетчика, работающего по описанному выше принципу, является двухфазный делитель-счетчик на двух RS триггерах, схема которого приведена на рис. 2.23, а диаграммы входных сигналов и сигналов триггеров a_1 и a_2 показаны на предыдущем рисунке.

Такой делитель-счетчик имеет четыре устойчивых состояния, закодированных цифрами **0**–**3**, а переключение триггеров осуществляется импульсами α и α' при суммировании и импульсами β и β' – при вычитании, как это показано на рис. 2.22, где в режиме суммирования выходные цифровые сигналы изменяются в прямой последовательности **0**–**3**, а в режиме вычитания – в обратной последовательности **3**–**0**. При смене режима работы происходит смена кода сигнала второй фазы по закону $a_2 \leftrightarrow \underline{a}_2$, а сигнал первой фазы a_1 не меняется.

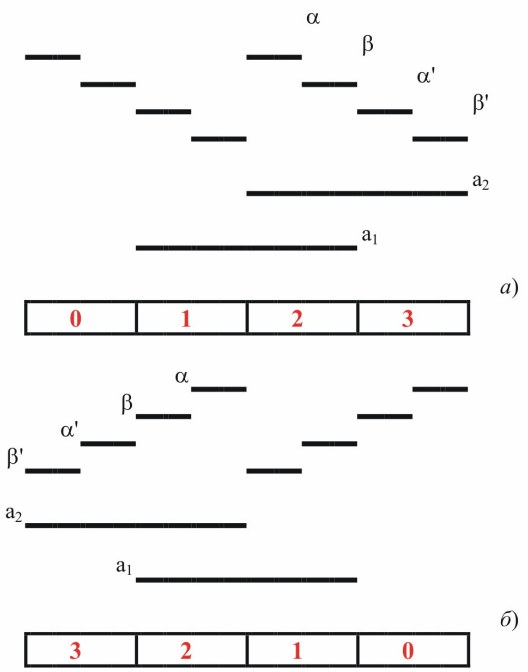


Рис. 2.22

Анализ работы предложенных реверсивных делителей-счетчиков показывает, что они обладают такой же устойчивостью, как и их нереверсивные варианты, имеющие вдвое меньшее количество шин. Поэтому при проектировании реверсивного делителя-счетчика достаточно знать его нереверсивный вариант, а дальнейшие действия будут заключаться в устранении нерабочих состояний делителя-счетчика и установке средств для реверсирования.

Различные области применения делителей-счетчиков многофазного кода обусловили многообразие их структур и оснований кода в отдельных разрядах. Примеры построения таких структур для инверторов с цифровым формированием и регулированием их выходных напряжений, цифровых преобразователей угла и скорости, а также других устройств электроприводов постоянного и переменного тока будут рассмотрены в следующих разделах.

Все применяемые здесь многоразрядные делители-счетчики представляют собой последовательную цепочку различных делителей в общем случае с разным основанием системы счисления, чаще всего соединенных при помощи согласующих делителей на двух триггерах.

Быстродействие одного многофазного делителя-счетчика, как отмечалось выше, больше, чем у двоичного счетчика равного основания. Однако последовательное соединение разрядов и здесь снижает быстродействие многоразрядного делителя-счетчика многофазного кода, что в ряде случаев нежелательно.

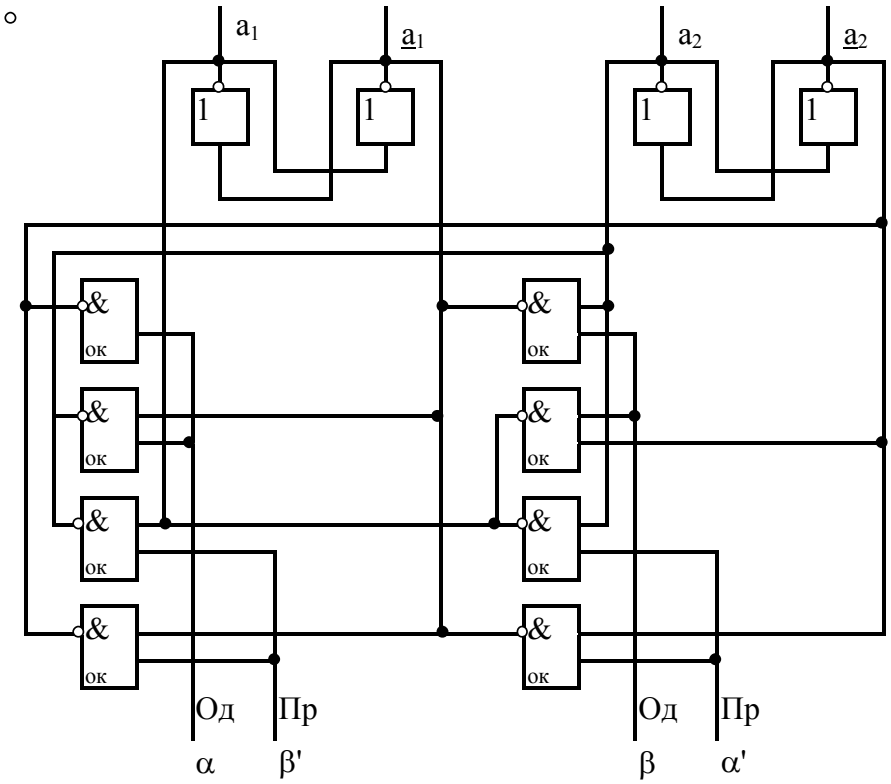


Рис. 2.23

Рассмотрим в связи с этим организацию параллельного переноса в делителях-счетчиках многофазного кода. На рис. 2.24 приведена структурная схема многоразрядного делителя-счетчика и показана цепь параллельного переноса. В таком счетчике переключение во всех триггерах происходит практически одновременно. Соседнее кодирование многофазного кода позволяет упростить цепь параллельного переноса: для организации переноса по одной шине в каждом разряде используется один трехвходовой элемент И.

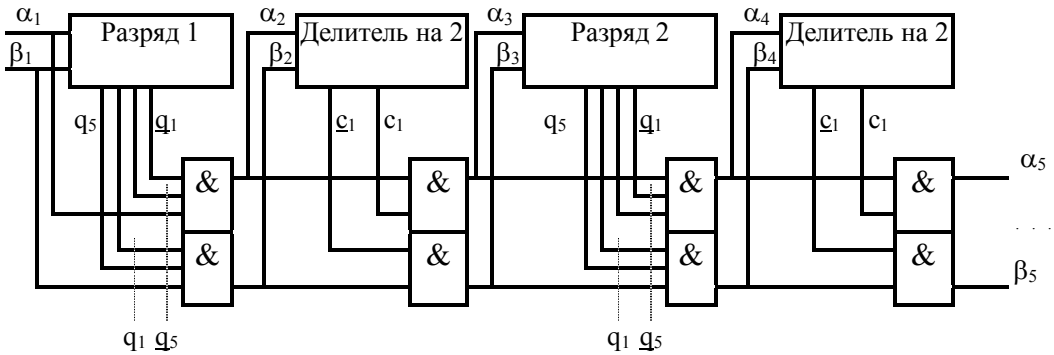


Рис. 2.24

У реверсивных делителей-счетчиков и счетчиков, имеющих большее количество входных шин, цепи переноса выполняются аналогичным образом.

2.4. Синтез управляемых делителей-счетчиков

Требования к управляемому от внешнего цифрового сигнала делителю-счетчику в устройствах электропривода и системах энергоснабжения – это высокая помехозащищенность, необходимость выходного частотного сигнала с неизменным для данного коэффициента деления периодом. Следовательно, сигнал не должен быть «гребенкой с поломанными зубьями», а цифровой выходной код должен менять основание системы счисления в соответствии с изменением коэффициента деления.

Этим условиям полностью удовлетворяет управляемый делитель-счетчик, который был предложен в [7] на нечетном числе RS триггеров и двух кольцевых схемах, составленных из вторичных обмоток входного импульсного трансформатора и такого же количества диодов.

На рис. 2.25 приведен фрагмент этой схемы из двух i и (i+1) RS триггеров, но выполненных на интегральных схемах, что не меняет существа его работы.

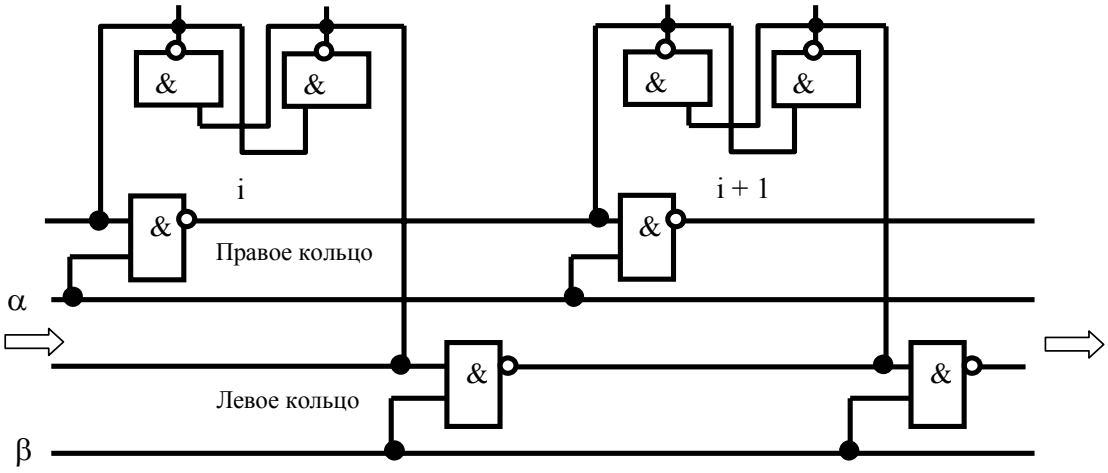


Рис. 2.25

В схеме содержится m нечетных RS триггеров, которые соединены между собой двумя кольцами, составленными, например, из двухвходовых элементов И-НЕ с открытым коллекторным выходом. Правое кольцо соединяет между собой правые плечи триггеров, а левое кольцо – левые плечи этих триггеров. В схеме имеются две входные шины, которые объединяют соответствующие входы элементов И-НЕ этих колец. На эти входные шины поочередно поступают импульсы, соответствующие положительной α и отрицательной $\underline{\alpha}$ полуволнам входного частотного сигнала делителя-счетчика.

В табл. 2.4.1 на примере делителя-счетчика из девяти RS триггеров представлена его работа при делении входной частоты на девять.

Таблица 2.4.1

№	Триггеры									Кольцо
	1	2	3	4	5	6	7	8	9	
0	0	0	1	0	1	0	1	0	1	Л
1	0	1	1	0	1	0	1	0	1	Пр
2	0	1	0	0	1	0	1	0	1	Л
3	0	1	0	1	1	0	1	0	1	Пр
4	0	1	0	1	0	0	1	0	1	Л
5	0	1	0	1	0	1	1	0	1	Пр
6	0	1	0	1	0	1	0	0	1	Л
7	0	1	0	1	0	1	0	1	1	Пр
8	0	1	0	1	0	1	0	1	0	Л
9	1	1	0	1	0	1	0	1	0	Пр
10	1	0	0	1	0	1	0	1	0	Л
11	1	0	1	1	0	1	0	1	0	Пр
12	1	0	1	0	0	1	0	1	0	Л
13	1	0	1	0	1	1	0	1	0	Пр
14	1	0	1	0	1	0	0	1	0	Л
15	1	0	1	0	1	0	1	1	0	Пр
16	1	0	1	0	1	0	1	0	0	Л
17	1	0	1	0	1	0	1	0	1	Пр

При нечетном числе триггеров и приведенном соединении только два рядом стоящих по порядку обхода триггера могут находиться в одинаковом состоянии 0*, 0* или 1*, 1*. Состояние всех остальных триггеров чередуется.

В первой строке этой таблицы показано состояние RS триггеров, когда был подан сигнал α в правое кольцо, и каждый по порядку обхода RS триггер, находящийся в положении 1*, установил следующий по кольцу за ним триггер в противоположное себе положение. Этому состоянию триггеров делителя-счетчика соответствует цифра 0.

При подаче следующего импульса $\underline{\alpha}$ на входную шину левого кольца каждый RS триггер, находящийся в положении 0*, устанавливает следующий по кольцу триггер в противоположное себе положение. Этому состоянию соответствует вторая строка таблицы и т.д.

При каждом поступлении чередующихся импульсов α, $\underline{\alpha}$ входной частоты только один триггер изменяет свое состояние. Это происходит по кольцу, и все триггеры делителя-счетчика переключаются с частотой, в m = 9 раз меньшей входной частоты.

Для большей наглядности на рис. 2.26 показаны графические соотношения между входными сигналами α и сигналами выходного кода $a_1 - a_9$, представляющие основание системы счисления $n = 2m = 18$. Сигналы этого кода являются разновидностью многофазного кода и имеют между собой фазовый сдвиг в $2\pi(m - 1)/m$ радиан.

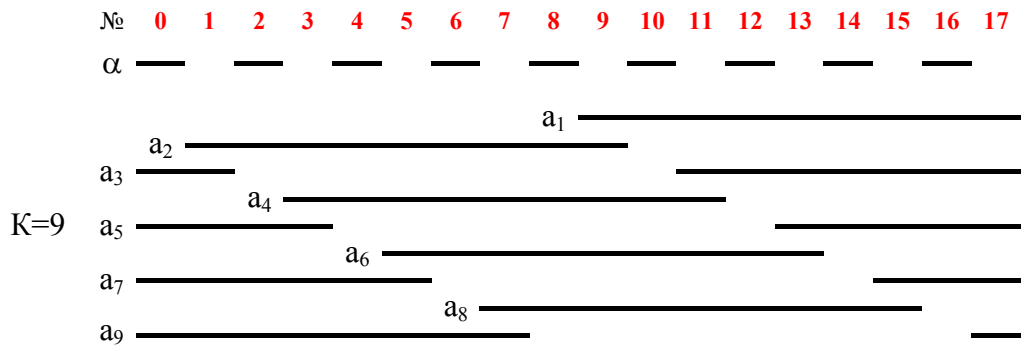


Рис. 2.26

Очевидно, что здесь выходной частотный сигнал $\alpha_9 = \underline{a}_1$.

Поскольку число 9 кратно 3, то не представляет труда сформировать сигналы многофазного кода для деления на три непосредственно из сигналов девятифазного кода, например в записи Либау – Крейга для основания $n = 6$. Это следующие сигналы:

$$a'_1 = a_2\, a_5 \vee a_2\, a_8 \vee a_8\, a_5;$$
$$a'_2 = \underline{a}_3\, \underline{a}_6 \vee \underline{a}_9\, \underline{a}_3 \vee \underline{a}_6\, \underline{a}_9;$$
$$a'_3 = a_4\, a_7 \vee a_1\, a_4 \vee a_1\, a_7,$$

а выходной частотный сигнал такого делителя-счетчика определяется выражением

$$\alpha_3 = \underline{a}'_3 = a'_3 = \underline{a}_4\, \underline{a}_7 \vee \underline{a}_1\, \underline{a}_4 \vee \underline{a}_1\, \underline{a}_7.$$

Для деления входной частоты на $(m - \gamma)$ из ансамбля RS триггеров должно быть исключено их четное число γ.

На рис. 2.27 приведен фрагмент принципиальной схемы делителя-счетчика для уменьшения его коэффициента деления на γ.

При наличии сигнала 1* на вторых входах элементов И-НЕ, например с открытым коллекторным выходом, RS триггер с номером γ будет дублировать все переключения RS триггера перед этой группой триггеров. Последний триггер будем называть полностью ведущим для триггера с номером γ и обозначать, например, i $(1^*, 0^*) \to \gamma$.

Остальные RS триггеры группы γ будут переключаться в определенном порядке, задаваемом алгоритмом переключения делителя-счетчика.

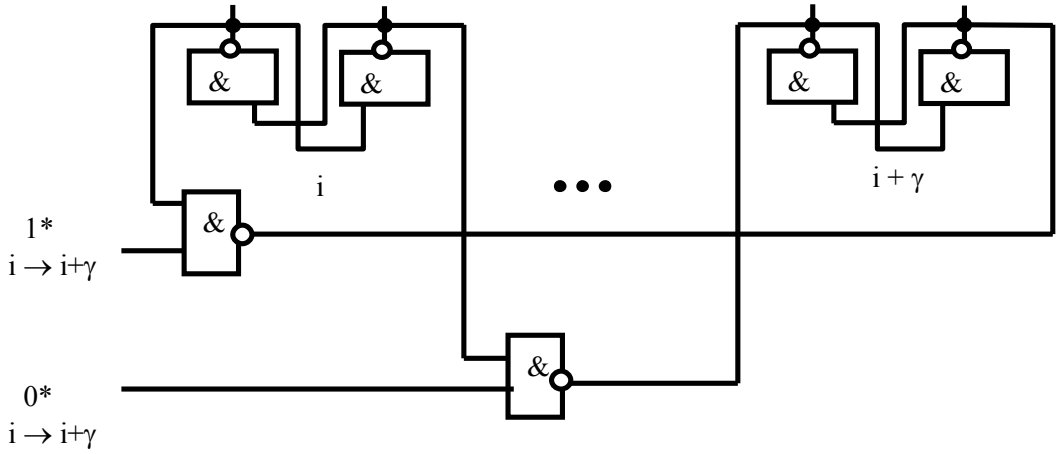


Рис. 2.27

Таким образом могут быть реализованы все следующие нечетные коэффициенты деления m = 7, 5, 3.

Пропустив процедуру составления таблиц переключения RS триггеров управляемого делителя-счетчика, которые выполняются аналогично табл. 2.4.1, покажем графически соотношения между входными сигналами α и сигналами $a_1 - a_9$.

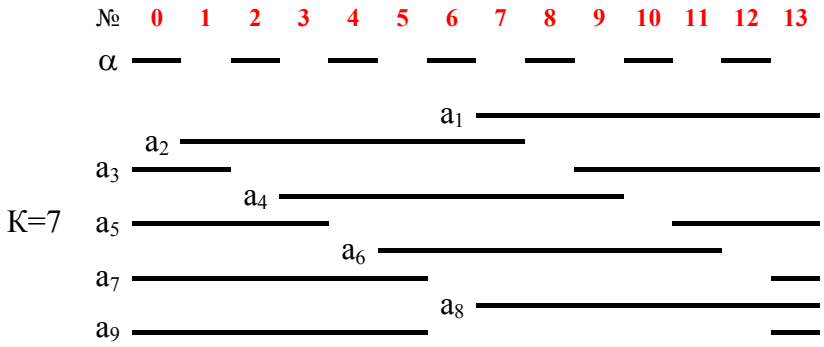


Рис. 2.28

Для деления на семь (рис. 2.28) сигналы $a_1 - a_7$ представляют цифры основания n = 14, а другие сигналы определяются зависимостями $a_9 = a_7$, $a_8 = a_1$. При этом выходной частотный сигнал $\alpha_7 = \underline{a}_1$.

Для деления на пять (рис. 2.29) сигналы $a_1 - a_5$ представляют цифры основания n = 10, а другие сигналы определяются зависимостями $a_9 = a_5$, $a_8 = a_3$, $a_7 = a_2$. При этом выходной частотный сигнал $\alpha_5 = \underline{a}_1$.

Для деления на четное число (m – γ/2) в схеме рис. 2.27 необходимо убрать одну из связей i-го RS триггера с триггером под номером γ. Например, триггер γ будет всегда совпадать с частично ведущим триггером только тогда, когда он находится в положении 1*. Это состояние записывается следующим образом: i (1*) → γ. Очевидно, что в схеме возможно использовать и другое состояние

i (0*) → γ. В дальнейшем будем рассматривать только первый вариант соединения.

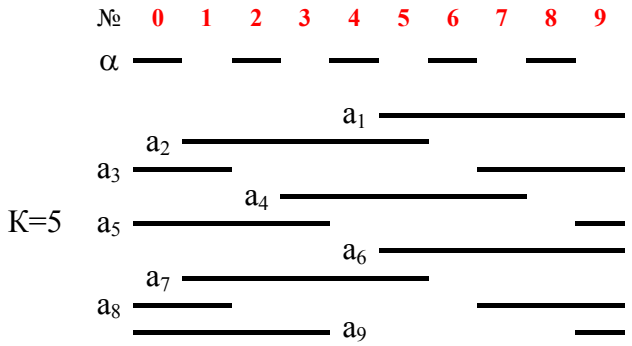


Рис. 2.29

Для деления на восемь в схеме необходимо установить соединение 7 (1*) → 9, тогда режим работы делителя-счетчика будет определяться табл. 2.4.2.

Таблица 2.4.2

№	Триггеры									Кольцо
	1	2	3	4	5	6	7	8	9	
0	0	0	1	0	1	0	1	0	1	Л
1	0	1	1	0	1	0	1	0	1	Пр
2	0	1	0	0	1	0	1	0	1	Л
3	0	1	0	1	1	0	1	0	1	Пр
4	0	1	0	1	0	0	1	0	1	Л
5	0	1	0	1	0	1	1	0	1	Пр
6	0	1	0	1	0	1	0	0	1	Л
7	0	1	0	1	0	1	0	1	1	Пр
8	0	1	0	1	0	1	0	1	0	Л
9	1	1	0	1	0	1	0	1	0	Пр
10	1	0	0	1	0	1	0	1	0	Л
11	1	0	1	1	0	1	0	1	0	Пр
12	1	0	1	0	0	1	0	1	0	Л
13	1	0	1	0	1	1	0	1	0	Пр
14	1	0	1	0	1	0	0	1	0	Л
15	1	0	1	0	1	0	1	1	1	Пр

Графические соотношения между входными сигналами и сигналами кода a_1 – a_9, определяющие основание системы счисления n = 16, приведены на рис. 2.30.

В этом случае все сигналы синтезированного кода a_1 – a_9 определяют основание системы счисления n = 16. В записи Либау – Крейга эти многофазные сигналы представляются следующим образом:

$a'_1 = \underline{a}_1\, a_2$, $a'_2 = \underline{a}_3\, a_2$, $a'_3 = \underline{a}_3\, a_4$, $a'_4 = \underline{a}_5\, a_4$, $a'_5 = \underline{a}_5\, a_6$, $a'_6 = \underline{a}_7\, a_6$, $a'_7 = \underline{a}_7\, a_8$, $\underline{a}'_8 = \underline{a}_1 a_9$.

При этом выходной частотный сигнал делителя-счетчика определяется зависимостью

$\alpha_8 = \underline{a}'_8 = = \underline{a}_1 a_9$.

Поскольку число 8 кратно 4 и 2, то несложно сформировать сигналы систем счисления оснований n =8 и n =4.

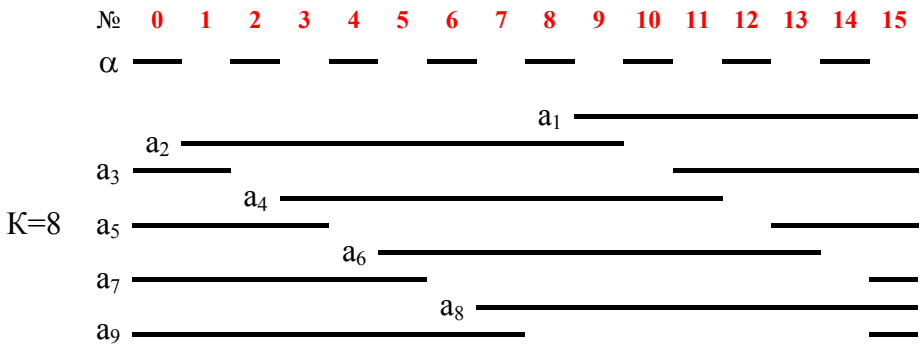


Рис. 2.30

Для деления на четыре можно непосредственно из сигналов девятифазного кода получить четырехфазный код, например в записи Либау – Крейга для основания n = 8. Это следующие выражения:

$$a'_1 = a_2\,\underline{a}_6 \vee a_1\,\underline{a}_5,\ a'_2 = a_7\,\underline{a}_3 \vee a_6\,\underline{a}_2,\ a'_3 = a_4\,\underline{a}_8 \vee a_3\,\underline{a}_7,\ a'_4 = a_9\,\underline{a}_5 \vee a_8\,\underline{a}_4,$$

а частотный выходной сигнал делителя-счетчика определяется выражением

$$\alpha_4 = \underline{a}'_3 = a_5\,\underline{a}_1 \vee a_4\,\underline{a}_8.$$

Для деления на два можно также непосредственно из сигналов девятифазного кода получить двухфазный код, например в записи Либау – Крейга для основания n = 4. Это выражения:

$$a'_1 = a_2\,\underline{a}_4 \vee a_6\,\underline{a}_8 \vee a_1\,\underline{a}_3 \vee a_5\,\underline{a}_7, \quad a'_2 = a_5\,\underline{a}_3 \vee a_9\,\underline{a}_7 \vee a_4\,\underline{a}_2 \vee a_8\,\underline{a}_6,$$

а частотный выходной сигнал делителя-счетчика определяется выражением

$$\alpha_2 = \underline{a}'_2 = a_3\,\underline{a}_1 \vee a_7\,\underline{a}_5 \vee a_2\,\underline{a}_9 \vee a_6\,\underline{a}_4.$$

Получение коэффициента деления на шесть можно реализовать двумя вариантами перестройки схемы делителя-счетчика. Первый вариант выполнения определяется установкой соединения 3 (1*) → 9, а второй – двумя соединениями 7 (1*, 0*) → 9, 5 (1*) → 7. Для реализации выберем первый вариант соединения.

На рис. 2.31 приведены соотношения между входными сигналами α и сигналами кода a_1 – a_9, определяющие основание системы счисления n = 12.

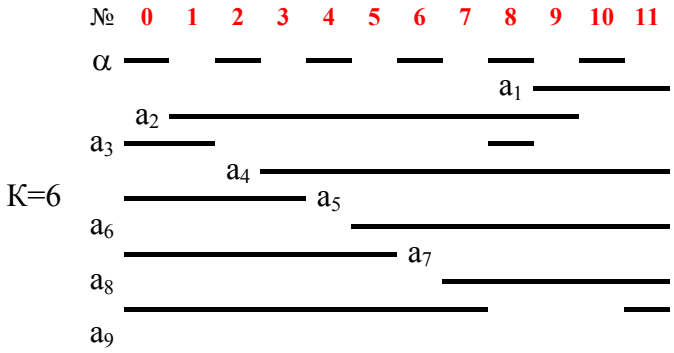


Рис. 2.31

Тогда сигналы шестифазного кода в записи Либау – Крейга представляются следующими выражениями:

$$a'_1 = a_2\,\underline{a}_8,\ a'_2 = a_9\,\underline{a}_3,\ a'_3 = a_4\,\underline{a}_1,\ a'_4 = a_2\,\underline{a}_5,\ a'_5 = a_6\,\underline{a}_3,\ a'_6 = \underline{a}_7,$$

а частотный выходной сигнал делителя-счетчика определяется выражением

$$\alpha_6 = \underline{a}'_7 = a_7.$$

Алгоритм выполнения логического блока, который должен будет по внешним цифровым сигналам управлять режимом работы делителя-счетчика, т.е. изменять его коэффициент деления, ясен из изложенного выше и определяется табл. 2.4.3.

Таблица 2.4.3

Входной код	$K_д$	Соединение в схеме делителя-счетчика				
		3 (1*) → 9	5 (1*) → 9	5 (0*) → 9	7 (1*) → 9	7 (0*) → 9
1	2					
2	3					
3	4					
4	5					
5	6					
6	7					
7	8					

Из данных этой таблицы ясно, как реализуется принципиальная схема логического блока, управляющего режимом работы делителя-счетчика.

При этом необходимо отметить, что число режимов работы управляемого делителя-счетчика, охватывающее все коэффициенты деления от единицы до m, определяется зависимостью (m + 1)/2. Так, для представленного выше делителя-счетчика (m = 9) это пять режимов: 9(3), 8(4,2), 7, 6, 5. Для одиннадцатифазного делителя-счетчика (m = 11) это шесть режимов работы: 11, 10(5, 2), 9(3), 8(4,2), 7, 6 и т.д., где кратные частоты получаются без изменения основного режима работы.

Остановимся на выполнении коммутаций выходных цифровых сигналов кода при изменении коэффициентов деления и соответственно оснований систем счисления разрядов многоразрядного делителя-счетчика многофазного кода. Пусть при этом цифровой код разрядов представляется в многофазном коде записью Либау – Крейга.

При делении на нечетное число это будут непосредственно сигналы RS триггеров, записанные в определенной последовательности.

При делении на четное число это будут синтезированные сигналы, определяемые логическими схемами по приведенным выше выражениям. Эти логические выражения приведены в табл. 2.4.4, где последние сигналы каждой колонки определяют также сигнал $\underline{\alpha}$ выходной частоты делителя-счетчика.

При делении входной частоты на девять (код управляющего цифрового сигнала соответствует цифре 7_b) все девять выходных шин определяют код числа делителя-счетчика.

При делении входной частоты на восемь (код управляющего цифрового сигнала соответствует цифре 6_b) только восемь выходных шин определяют код

числа делителя-счетчика и т.д. вплоть до деления на два, когда две выходные шины определяют этот код.

Принципиальная схема логического блока, реализующего алгоритм, представленный табл. 2.4.4, приведена на рис. 2.32.

Таблица 2.4.4

Выходной код		Входной код							
		0_b	**1_b**	**2_b**	**3_b**	**4_b**	**5_b**	**6_b**	**7_b**
	a'_1	$a_2\underline{a_4} \vee a_6\underline{a_8} \vee a_1\underline{a_3} \vee a_5\underline{a_7}$ 1	a_2	$a_2\underline{a_6} \vee a_1\underline{a_5}$ 2	a_2	a_2a_8 3	a_2	$a_2\underline{a_1}$ 4	a_2
	a'_2	$a_5\underline{a_3} \vee a_9\underline{a_7} \vee a_4\underline{a_2} \vee a_8\underline{a_6}$ 5	$\underline{a_3}$	$a_7\underline{a_3} \vee a_6\underline{a_2}$ 6	$\underline{a_3}$	$a_9\underline{a_3}$ 7	$\underline{a_3}$	$a_2\underline{a_3}$ 8	$\underline{a_3}$
	a'_3		a_4	$a_4\underline{a_8} \vee a_3\underline{a_7}$ 9	a_4	$a_4\underline{a_1}$ 10	a_4	$a_4\underline{a_3}$ 11	a_4
	a'_4			$a_9\underline{a_5} \vee a_8\underline{a_4}$ 12	$\underline{a_5}$	$a_2\underline{a_5}$ 13	$\underline{a_5}$	$a_4\underline{a_5}$ 14	$\underline{a_5}$
	a'_5				a_6	$a_6\underline{a_3}$ 15	a_6	$a_6\underline{a_5}$ 16	a_6
	a'_6					$\underline{a_7}$	$\underline{a_7}$	$a_6\underline{a_7}$ 17	$\underline{a_7}$
	a'_7						a_8	$a_8\underline{a_7}$ 18	a_8
	a'_8							$a_9\underline{a_1}$ 19	$\underline{a_9}$
	a'_9								a_1

В схеме блока используются десять мультиплексоров MS_1 – MS_{10} и преобразователь входного кода ПК.

Мультиплексоры MS_1 – MS_5, MS_{10} являются коммутаторами из 8 в 1; MS_6, MS_7 – коммутаторами из 4 в 1; MS_8, MS_9 – коммутаторами из 2 в 1. Причем мультиплексоры MS_3 – MS_9 имеют каждый выходную шину с тремя устойчивыми состояниями.

Рядом с входными информационными шинами первых девяти мультиплексоров написаны светлым шрифтом взятые из табл. 2.4.4 номера сигналов в коде Либау – Крейга, а информационные сигналы, совпадающие с сигналами RS триггеров делителя-счетчика, представлены непосредственно.

На выходных шинах преобразователя кодов входной двоичный код $B(b_1, b_2, b_3)$ преобразуется в сигналы интегрального кода $L_1 = \mathbf{0} \vee ... \vee \mathbf{6}$, $L_2 = \mathbf{0} \vee ... \vee \mathbf{5}$, ... , $L_6 = \mathbf{0} \vee 1$, $L_7 = \mathbf{0}$. При наличии этих сигналов на управляющих входах соответствующих мультиплексоров MS_3 – MS_9 их выходные шины отключаются.

Таким образом, при входном цифровом коде цифры $\mathbf{0_b}$ будут использованы только выходные сигналы MS_1, MS_2 соответственно на выходных шинах 1 и 2; при входном цифровом коде цифры $\mathbf{1_b}$ будут использованы выходные сигналы MS_1, MS_2 , MS_3 соответственно на выходных шинах 1, 2 и 3 и т.д. вплоть до цифрового входного кода цифры $\mathbf{7_b}$, когда будут подключены к выходным шинам 1–9 сигналы мультиплексоров MS_1 – MS_9.

Десятый мультиплексор MS_{10}, где его информационные входы соединены с выходными шинами мультиплексоров MS_2 – MS_9, а входы выбора данных – с входами шин двоичного кода $B(b_1, b_2, b_3)$, служит для формирования частотного выходного сигнала $\underline{\alpha}$ делителя-счетчика.

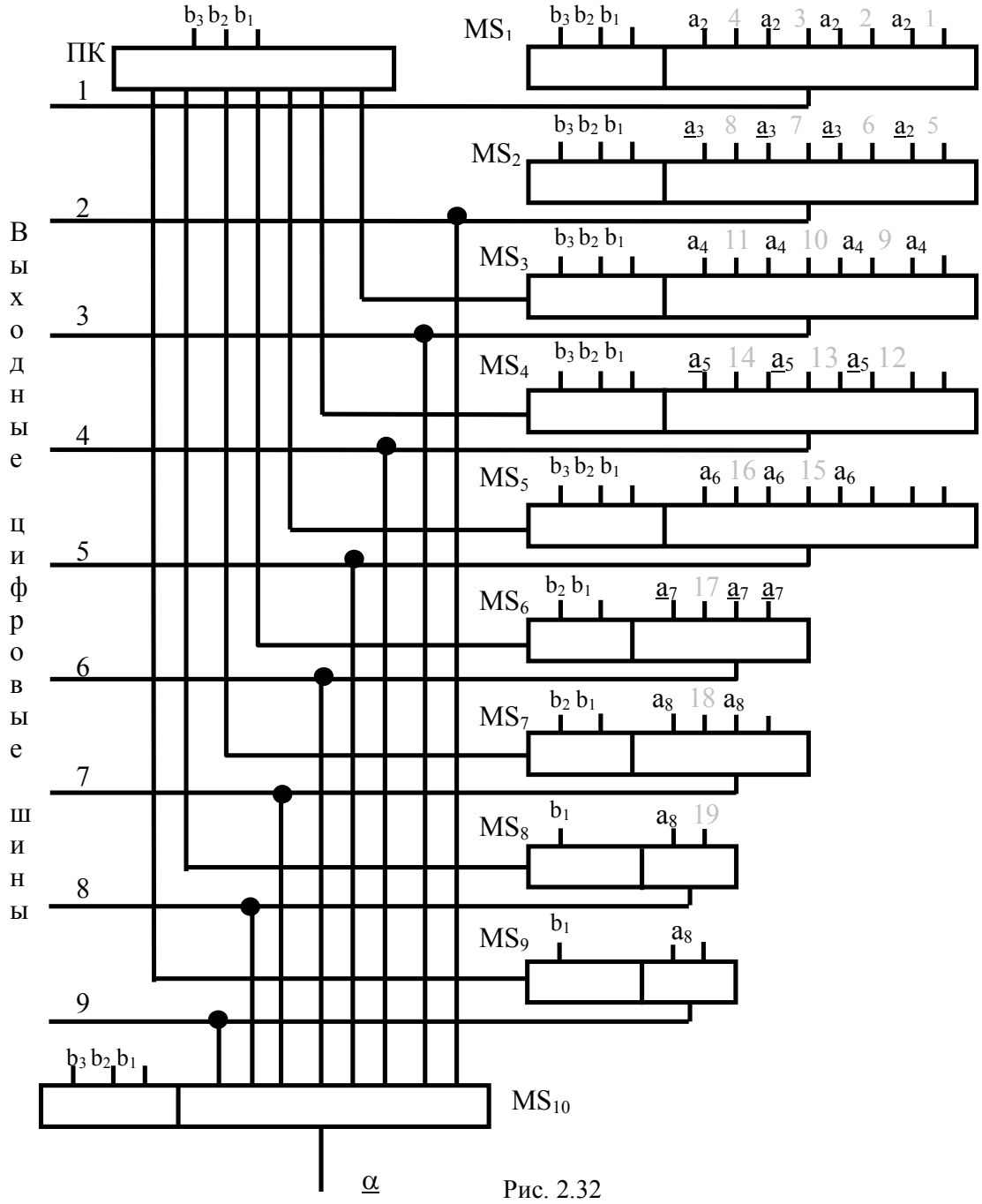


Рис. 2.32

Таким образом, проведен синтез управляемого делителя-счетчика, который позволяет получить выходной частотный сигнал с неизменным для заданного коэффициента деления периодом и цифровым выходным кодом, изменяющим соответствующим образом основание системы счисления, оставаясь при этом во всех режимах многофазным.

Ничто не является для нас более наглядным, чем фигура, ибо ее можно осязать и видеть.

Декарт

Числа повторяют пространство, хотя так от него отличны.

Паскаль

Глава 3

КОНТРОЛЕСПОСОБНОСТЬ ПОЗИЦИОННЫХ СИСТЕМ СЧИСЛЕНИЯ

Проблема синтеза высоконадежных систем управления электроприводами и комплексами энергоснабжения, состоящими из большого количества аналоговых и цифровых электронных блоков, приобретает все большее значение.

Одним из основных и перспективных путей достижения высоких показателей надежности таких комплексов является их синтез на базе использования самопроверяемых электронных блоков. Причем электронные блоки современных электроприводов становятся все более сложными.

Самопроверяемость в этом случае есть свойство обнаружить, а в лучшем случае и устранить неисправности как в основной, так и во встроенной аппаратуре контроля. Причем эти операции должны выполняться в режиме реального времени без прогонки специальных тестов или имитации неисправных состояний. Такой подход не исключает прогонки специальных тестов до момента включения электропривода для определения его готовности к выполнению необходимых функций.

Аналогичные задачи стоят достаточно давно и перед разработчиками ЭВМ, где использование самопроверяемых схем встроенного контроля позволяет избежать проблемы «контролера над контролером» и свести в идеале неконтролируемое ядро системы к нулю [9].

Решение этой проблемы для ЭВМ далеко до завершения, хотя ограничивается в основном пока только задачей обнаружения неисправностей. Это объясняется большой сложностью решения задач самоконтроля, где средства достижения отказоустойчивости очень разнообразны. Выбор средств обеспечения отказоустойчивости зависит и от условий применения, и от требований к показателям надежности систем, а обоснование выбора средств весьма затруднительно. При этом следует учитывать, что обычно ЭВМ работают в «тепличных» условиях внешних помех, а использование электронных блоков, например, в электроприводах далеко от таких «тепличных» условий.

Отказоустойчивость электронных блоков и системы в целом достижима не только с помощью обеспечения самопроверяемости, а также самым распространенным способом – резервированием, в частности использованием мажоритарных структур либо полным поэлементным резервированием блоков.

Однако резервирование, которое устраняет последствия полного выхода из строя какого-либо элемента схемы, не может служить гарантией защиты от всевозможных видов кратковременных сбоев в схеме. Поэтому создание высоконадежных схем заключается в сочетании резервирования и использовании избыточных кодов, позволяющих обнаруживать и даже исправлять ошибки при передаче, хранении и выполнении определенных логических и арифметических операций над сигналами.

Самым простым решением задачи контролеспособности цифровых устройств является использование двухпроводных кодов. Именно в применении этих кодов ряд исследователей видит решение вопроса контролеспособности ЭВМ [12, 5].

В двухпроводном коде каждый информационный разряд дополняется еще одним разрядом, причем значение сигнала в дополнительном разряде противоположно значению сигнала в информационном разряде, т.е. он является его инверсией. Следовательно, для двухпроводного кода кодовыми словами **A** являются слова, у которых каждый информационный разряд и соответствующий ему дополнительный разряд имеют противоположные значения, т.е. 0*1* или 1*0*. Ошибочным является слово, у которого найдется по крайней мере одна такая пара разрядов, где информационный и дополнительный разряды совпадают. Для определения работоспособности какого-либо вычислительного блока либо комплекса в целом необходимы два условия. Первое из них заключается в том, что информационные и дополнительные разряды существуют независимо один от другого на всем протяжении от входа устройства до его выхода, а не образуются простым инвертированием сигналов при их передаче между узлами или блоками схемы. Второе обусловлено тем, что каждый функциональный узел выполняется для получения раздельно информационных и дополнительных разрядов.

Отмеченное поясняется рис. 3.1, *а*, где приведены информационные **A**, **B** и дополнительные $\underline{\mathbf{A}}$, $\underline{\mathbf{B}}$ сигналы, поступающие в два функциональных блока: $\mathbf{\Phi_1}$, $\mathbf{\Phi_2}$. Блоки формируют соответственно функционалы информационных **F** и дополнительных $\underline{\mathbf{F}}$ сигналов. Связь между сигналами **F** и $\underline{\mathbf{F}}$ поясняется их геометрическим образом (рис. 3.1, *б*) в многомерном цифровом пространстве, например операнд **A** и **B**.

Правильность работы любого функционального блока и комплекса в целом будет определяться несовпадением соответствующих информационных и дополнительных разрядов на входе и выходе этого блока. Такое построение любого цифрового устройства требует дополнительной установки оборудования, но дает полную гарантию его работоспособности.

Несмотря на кажущуюся тривиальность представленного решения самоконтроля, оно весьма продуктивно, а в сочетании с другими контролеспособными кодами позволяет не только обнаруживать, но даже исправлять большой класс ошибок.

В самом деле, двухпроводность может быть применима для любого типа кода, как контролеспособного, так и неконтролеспособного, например двоичного.

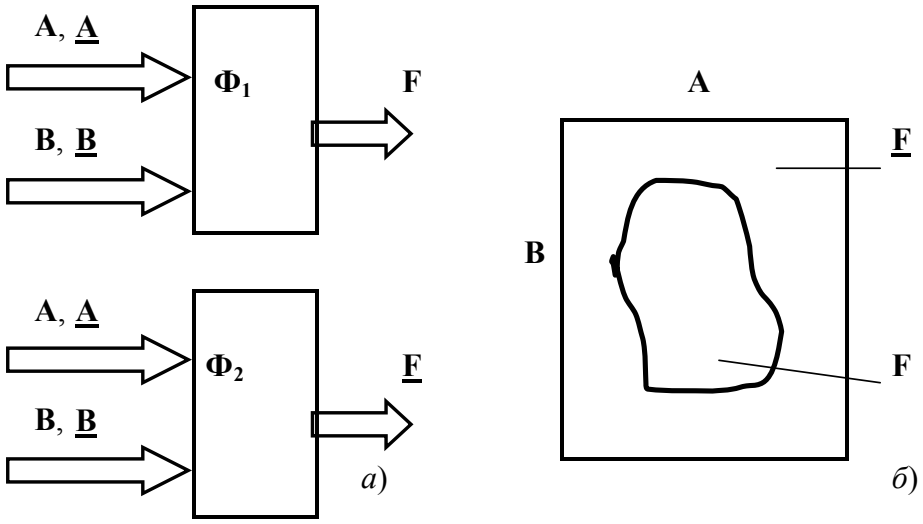


Рис. 3.1

В случае применения двухпроводности для контролеспособного кода, например многофазного, она увеличивает его возможности по исправлению и обнаружению ошибок. Именно этот принцип сочетания кодов используется нами в дальнейшем исследовании.

При этом структура любого самоконтролируемого блока будет представляться в виде рис. 3.2, где на его входные шины подаются операнды в двухпроводном коде, например **A**, **A**; **B**, **B**; ... , а выходные шины содержат результат соответствующего функционала **F**, **F** также в двухпроводном коде и сигнал λ, определяющий исправность блока.

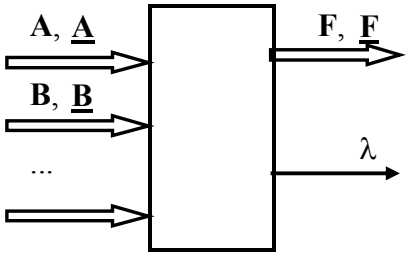


Рис. 3.2

Причем в дальнейшем с целью упрощения материала не указываются на структурных схемах сигналы дополнительных разрядов **A**, **B**, ...

В тех случаях, когда линии связи между сигналами коротки или практически отсутствуют, что исключает возникновение помех в них, возможно получе-

ние на входах и выходах блоков дополнительных сигналов разрядов простым инвертированием информационных сигналов.

Использование для представления позиционной системы счисления так называемых корректирующих кодов, предназначенных для обнаружения и исправления ошибок, является основным методом аппаратного контроля.

Корректирующие свойства кода зависят от его избыточности, проявляющейся в том, что для представления позиционной системы используются не все возможные кодовые комбинации универсального цифрового пространства. Действительно, если все 2^k кодовые комбинации, где k – длина кода, использовать для записи информации, то сбой любого вида переводит одну информационную кодовую комбинацию в другую, и этот перевод обнаружить и тем более исправить не удастся. Поэтому для обнаружения и исправления ошибок из 2^k кодовых комбинаций необходимо использовать для представления, например, цифр обычной позиционной системы счисления только часть. Обнаружение ошибок в этом случае сводится к выявлению нерабочей области универсального цифрового пространства, а исправление – к определению наиболее вероятной кодовой комбинации в рабочей области пространства, которая соответствует нерабочей области.

При этом выделение в k разрядах отдельно информационных и контрольных, что обычно выполняется в работах по вычислительной технике, не обязательно и является только одним из вариантов построения корректирующих кодов.

3.1. Расположение ошибок позиционных систем счисления в многомерном пространстве

Для определения возможностей по обнаружению и исправлению ошибок необходимо принять следующие допущения:

1. Все разряды кода равноценны, т.е. появление ошибок в любом разряде возможно с равной вероятностью.
2. Вероятность появления ошибок с большей кратностью уменьшается с увеличением кратности.
3. Вероятности появления ошибок одной кратности независимы друг от друга.

Если избыточность кода заранее определена, то необходимым и достаточным условием обнаружения всех ошибок определенного типа является невозможность перевода одного штатного положения в другое при этом типе ошибок, а условием исправления ошибок – отсутствие перекрытия геометрических образов кодовых комбинаций соответствующего типа ошибок различных штатных кодовых комбинаций.

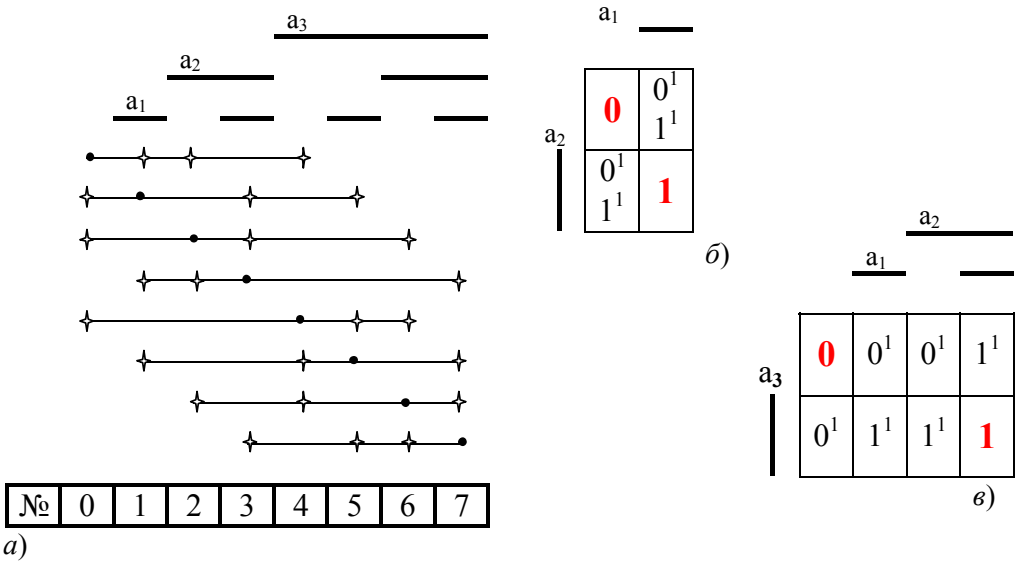


Рис. 3.3

Для пояснения этого обратимся к рис. 3.3, *а*, где для трехразрядного кода, задающего восемь позиций, показаны переходы каждой из позиций в новые ошибочные при воздействии, например, одиночных ошибок.

При кодировании основания системы счисления $n = 2$ необходимым и достаточным условием обнаружения одиночных ошибок является расположение, например, цифры **0** на первой позиции, а цифры **1** – на четвертой позиции, т.е. использование двух разрядов, когда одиночные ошибки будут переводить кодовые комбинации для цифр **0** и **1** в перекрывающиеся кодовые комбинации на второй и четвертой позиции (см. рис. 3.3, *б*).

Необходимым и достаточным условием исправления одиночных ошибок для этого основания является расположение цифры **0** на первой позиции, а цифры **1** – на восьмой позиции, т.е. необходимо использование трех разрядов, когда одиночные ошибки будут переводить кодовую комбинацию цифры **0** во вторую, третью и пятую позиции, а цифру **1** – в седьмую, шестую и четвертую позиции (см. рис. 3.3, *в*).

Аналогично могут быть синтезированы коды, обнаруживающие и исправляющие любой тип ошибок либо их комбинацию, но такой метод синтеза весьма трудоемок и не обладает необходимой «прозрачностью».

Наиболее наглядно и менее трудоемко применение для анализа и синтеза корректирующих кодов метода многомерных цифровых множеств.

Перед изложением этого метода для решения вопросов контролеспособности позиционных систем счисления введем обозначение штатных кодовых комбинаций сигналами обычного цифрового кода, т.е. цифрами, а кодовые комбинации, появляющиеся от ошибок кратности k, – этими же цифрами с индексом k. Например, **0** – штатная кодовая комбинация, соответствующая цифре 0, 0^k –

кодовая комбинация, возникающая из штатной в результате ошибки кратности k. В дальнейшем будем пользоваться терминами соответствия «цифре i» и «цифре i кратности k» , подразумевая изложенное выше.

Определение ошибочных кодовых комбинаций соответствующей кратности с помощью средств вычислительной техники весьма просто, но их графическое представление обладает большей наглядностью и удобством использования для дальнейшего изложения. Это представление изображено на рис. 3.4, где для ошибки кратности один представлены для каждой позиции (черная заливка) их новые ошибочные положения (светлая заливка).

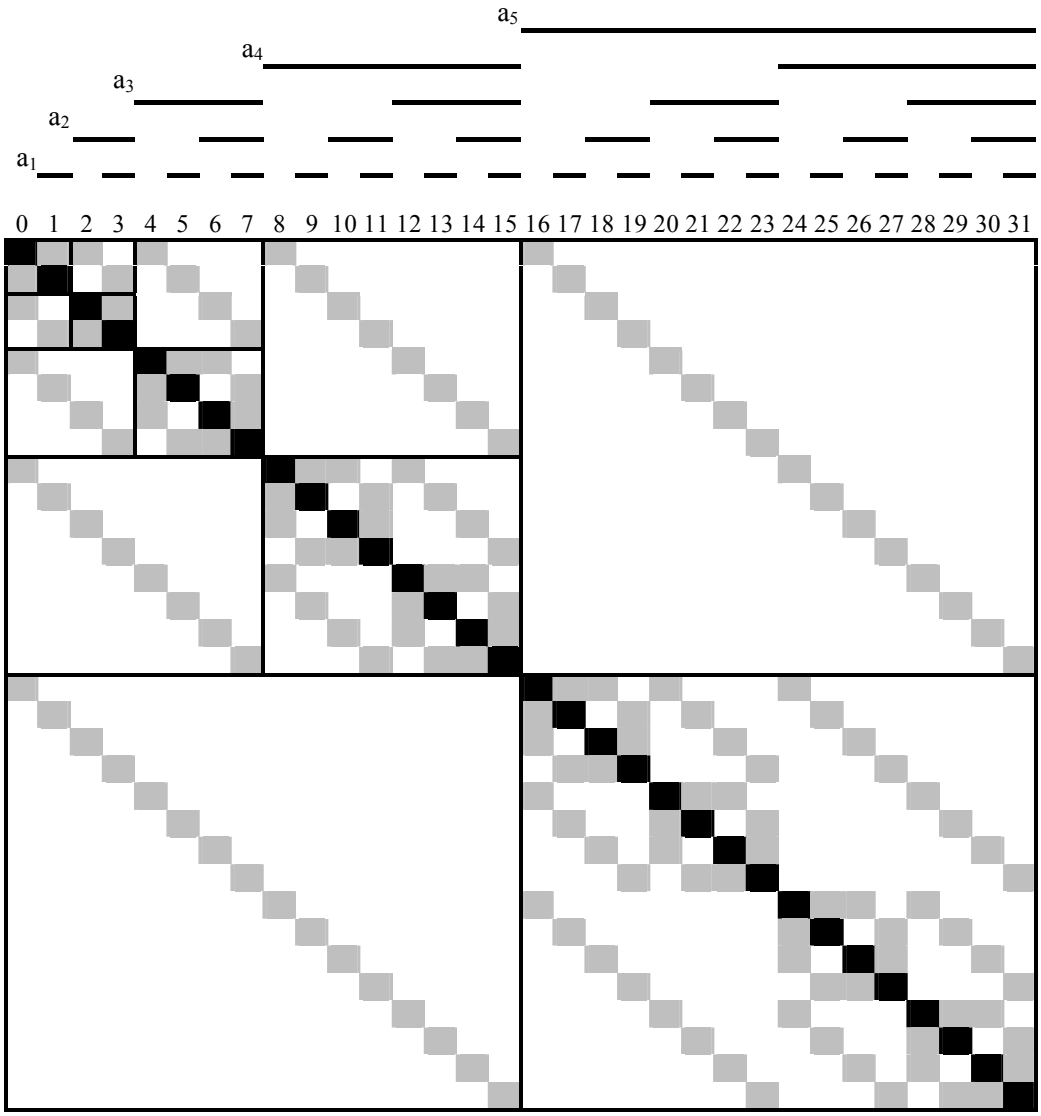


Рис. 3.4

Обратимся к рис. 3.4, где кодовая комбинация под номером 1 (первая строка) в случае одиночных ошибок может перейти в одну из позиций под номерами 2, 3 или 17. Позиция 2 (вторая строка) может перейти соответственно в позиции 1, 4, 6, 10 или 18 и т.д. вплоть до 32-й строки, где кодовая позиция с этим номером может перейти в позиции 16, 24, 28, 30 или 31.

Теперь получим полную картину одиночных ошибок в кодовых комбинациях из пяти разрядов. Очевидная симметрия представленной фигуры позволяет продолжить ее построение для любого количества разрядов.

Образующей фигурой для одиночных ошибок является квадрат размерами $2^1 \times 2^1$, где в ячейках по главной диагонали осуществлена черная заливка, а на второй диагонали – более светлая. Следующий больший квадрат размерами $2^2 \times \times 2^2$ составляется из элементов квадрата $2^1 \times 2^1$: по главной диагонали располагаются упомянутые выше без каких-либо изменений квадраты размерами $2^1 \times \times 2^1$, на другой диагонали – эти квадраты, где вместо черной заливки светлая, а светлая заливка первичного квадрата отсутствует. Следующий квадрат размерами $2^3 \times 2^3$ формируется из элементов квадрата $2^2 \times 2^2$ по такому же принципу и т.д.

Исходя из геометрического представления обнаружения ошибок основания n = 2, легко выделить в двухмерном пространстве кодовых комбинаций рис. 3.3, *б* сигнал одиночных ошибок, который определяется покрытием «площади», в которую входят кодовые комбинации 0^1, 1^1. Этот сигнал представляется элементарным логическим выражением

$$\underline{\lambda} = a_1\underline{a}_2 \vee \underline{a}_1 a_2. \qquad (3.1.1)$$

Исправление одиночных ошибок для этого основания определяется из двухмерного представления кодовых комбинаций рис. 3.3, *в*, где сигнал цифры **0** определяется «площадью», в которую входят кодовые комбинации **0**, 0^1, а сигнал цифры **1** – площадью, в которую входят кодовые комбинации **1**, 1^1, и задается простым логическим выражением

$$\mathbf{1} = a_2a_3 \vee a_1a_2 \vee a_1a_3 . \qquad (3.1.2)$$

Без наложения ограничений на число разрядов задача обнаружения и исправления любого типа ошибок и их комбинаций несложна. Также проста задача обнаружения и исправления одиночных ошибок любого основания системы счисления при ее сведении к отдельным разрядам. Например, для $n = 2^k$ каждый разряд может рассматриваться отдельно и для обнаружения одиночных ошибок этого основания потребуется 2k разрядов, а для исправления одиночных ошибок должно быть 3k разрядов.

Здесь уместен вопрос: правильно ли мы распорядились этим количеством разрядов? Нельзя ли более рационально расположить штатные кодовые комбинации в многомерном цифровом пространстве для повышения контролеспособности?

Проведенные исследования позволяют дать положительный ответ на поставленные вопросы. Таким образом, задача обнаружения и исправления ошибок без наложения ограничения на число использованных разрядов сводится к обнаружению и исправлению ошибок каждого разряда.

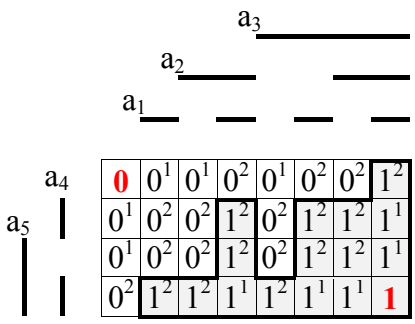


Рис. 3.5

Таким образом, нетрудно представить в двухмерном измерении «площади», определяющие одновременное исправление одиночных и двойных (рис. 3.5), а также одиночных, двойных и тройных ошибок (рис. 3.6) и т.д.

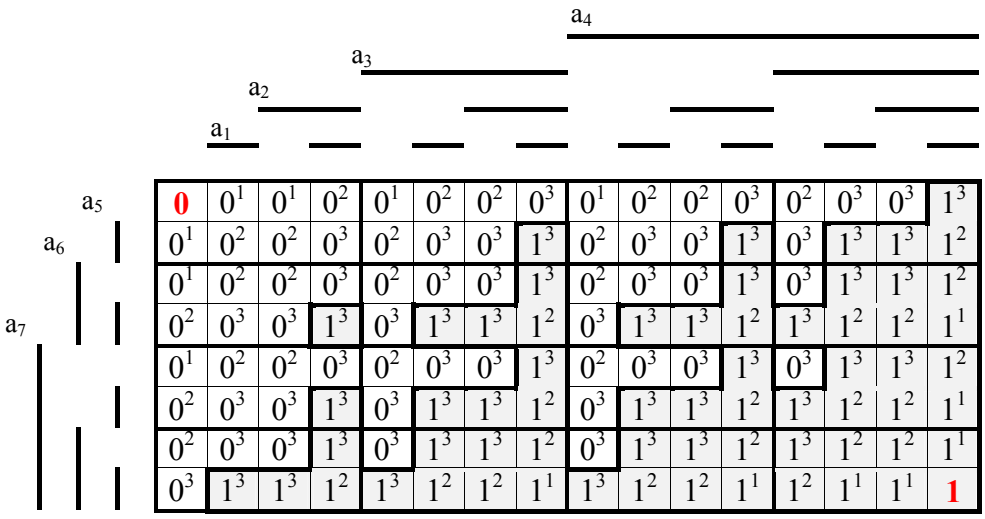
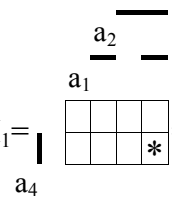
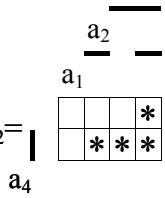
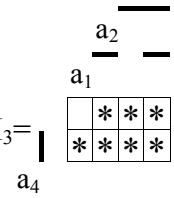


Рис. 3.6

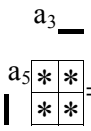
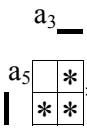
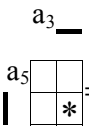
Для получения логического выражения, определяющего одновременное исправление одиночных и двойных ошибок, обратимся к рис. 3.5. Представим цифровые подмножества, составляющие геометрический образ логической функции с исправлением этих ошибок для цифры **1**, в координатах a_1, a_2, a_4. Число этих подмножеств равно трем:

$M_1=$  $= a_1\, a_2\, a_4,$

$M_2=$  $= a_1a_2 \vee a_2a_4 \vee a_1a_4,$

$M_3=$  $= a_1 \vee a_2 \vee a_4.$ (3.1.3)

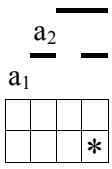
Учитывая, что $M_3 \supset M_2 \supset M_1$, эти цифровые подмножества могут быть изображены в координатах a_3,a_5 и соответственно записаны следующим образом:

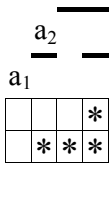
 $= M_1$,  $= M_1(a_3 \vee a_5)$,  $= M_1(a_3\, a_5)$. (3.1.4)

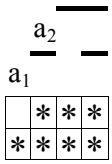
Из (3.1.3), (3.1.4) непосредственно следует логическое выражение для определения сигнала цифры **1**, где исправляются все одиночные и двойные ошибки

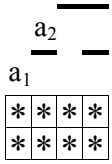
1 $= a_1a_2\, a_4 \vee a_1a_2\, a_3 \vee a_2a_4a_3 \vee a_1a_4\, a_3 \vee a_1a_2\, a_5 \vee a_2a_4a_5 \vee a_1a_4\, a_5 \vee a_3a_5\, a_1 \vee a_3a_5a_2 \vee a_3a_5\, a_4.$ (3.1.5)

Аналогичным образом одновременное исправление всех одиночных, двойных и тройных ошибок в соответствии с рис. 3.6 будет определяться подмножествами в координатах a_1, a_2, a_5, которые в свою очередь размещаются в ячейках пространства с координатами a_3, a_4, a_6, a_7.

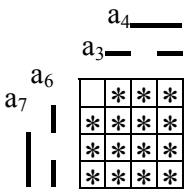
$M_1=$  $= a_1\, a_2\, a_5,$

$M_2=$  $= a_1a_2 \vee a_2a_5 \vee a_1a_5,$

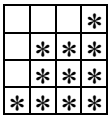
$M_3=$  $= a_1 \vee a_2 \vee a_5,$

$M_4=$  $= 1*.$ (3.1.6)

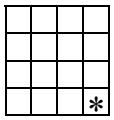
Учитывая, что $M_4 \supset M_3 \supset M_2 \supset M_1$, эти цифровые подмножества могут быть изображены в этих координатах и записаны следующим образом:



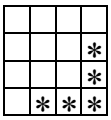
$= a_3 \vee a_4 \vee a_6 \vee a_7,$



$= a_4a_7 \vee a_3a_4 \vee a_6a_7 \vee a_3a_6 \vee a_3a_7,$



$= a_4a_3a_6a_7,$



$= a_3a_4a_6 \vee a_3a_4a_7 \vee a_6a_7a_3 \vee a_6a_7a_4.$

(3.1.7)

Из (3.1.6), (3.1.7) непосредственно следует вывод логического выражения для определения сигнала цифры **1**, в котором исправляются все одиночные, двойные и тройные ошибки:

$$
\begin{aligned}
\mathbf{1} = \; & a_1a_2a_5a_3 \vee a_1a_2a_5a_4 \vee a_1a_2a_5a_6 \vee a_1a_2a_5a_7 \vee a_1a_2a_4a_7 \vee \\
& \vee a_1a_5a_4a_7 \vee a_2a_5a_7a_7 \vee a_1a_2a_3a_4 \vee a_1a_5a_3a_4 \vee a_2a_5a_3a_4 \vee \\
& \vee a_1a_2a_6a_7 \vee a_1a_5a_6a_7 \vee a_2a_5a_6a_7 \vee a_1a_2a_3a_6 \vee a_1a_5a_3a_6 \vee \\
& \vee a_2a_5a_3a_6 \vee a_1a_2a_3a_7 \vee a_1a_5a_3a_7 \vee a_2a_5a_3a_7 \vee a_3a_4a_6a_1 \vee \\
& \vee a_3a_4a_7a_1 \vee a_6a_7a_3a_1 \vee a_6a_7a_4a_1 \vee a_3a_4a_6a_2 \vee a_3a_4a_7a_2 \vee \\
& \vee a_6a_7a_3a_2 \vee a_6a_7a_4a_2 \vee a_3a_4a_6a_5 \vee a_3a_4a_7a_5 \vee a_6a_7a_3a_5 \vee \\
& \vee a_6a_7a_4a_5 \vee a_4a_3a_6a_7 .
\end{aligned}
\qquad (3.1.8)
$$

Представленные на рис. 3.3, *в*, 3.5, 3.6 геометрические образы кодов, с возможностью исправления ошибок определенной кратности, могут быть названы суперсовершенными для основания n = 2. Определение суперсовершенного кода конкретного основания системы счисления, исходя из геометрического образа этого кода, состоит из трех компонентов.

Во-первых, в ячейках многомерного цифрового пространства координат кода должны располагаться только цифровые сигналы определенной заданной нами кратности ошибок. При этом в каждой ячейке может располагаться только цифра с одной ошибкой определенной кратности.

Во-вторых, многомерное цифровое пространство должно использоваться для размещения ошибок определенных кратностей на 100 процентов.

В-третьих, исправление ошибок контрольных разрядов кода должно выполняться такими же логическими схемами, которые исправляют ошибки в информационных разрядах кода, посредством соответствующего изменения входных сигналов, которые производятся мысленными поворотами относительно осей симметрии этого цифрового пространства.

При отсутствии третьего компонента в этом определении совершенного кода такой код будем считать совершенным, а при отсутствии второго и третьего – квазисовершенным.

Совершенные коды позволяют не только исправлять ошибки, но одновременно выполняют функцию резервирования логических и цифровых систем управления, где они используются. По этой причине их значение сложно переоценить.

Следует отметить, что синтез совершенных кодов для больших оснований систем счисления n > 2 представляет собой сложную задачу, которая до настоящего времени была полностью не решена.

Таблица 3.1.1

0	1	2	4	8	16
0	**1**	3	5	9	17
0	**2**	3	6	10	18
1	2	**3**	7	11	19
0	**4**	5	6	12	20
1	4	**5**	7	13	21
2	4	**6**	7	14	22
3	5	6	**7**	15	23
0	**8**	9	10	12	24
1	8	**9**	11	13	25
2	8	**10**	11	14	26
3	9	10	**11**	15	27
4	8	**12**	13	14	28
5	9	12	**13**	15	29
6	10	12	**14**	15	30
7	11	13	14	**15**	31
0	**16**	17	18	20	24
1	16	**17**	19	21	25
2	16	**18**	19	22	26
3	17	18	**19**	23	27
4	16	**20**	21	22	28
5	17	20	**21**	23	29
6	18	20	**22**	23	30
7	19	20	22	**23**	31
8	16	**24**	25	26	28
9	17	24	**25**	27	29
10	18	24	**26**	27	30
11	19	25	26	**27**	31
12	20	24	**28**	29	30
13	21	25	28	**29**	31
14	22	26	28	**30**	31
15	23	27	29	30	**31**

≡

0	1	2	4	8	16
1	0	3	5	9	17
2	0	3	6	10	18
3	1	2	7	11	19
4	0	5	6	12	20
5	1	4	7	13	21
6	2	4	7	14	22
7	3	5	6	15	23
8	0	9	10	12	24
9	1	8	11	13	25
10	2	8	11	14	26
11	3	9	10	15	27
12	4	8	13	14	28
13	5	9	12	15	29
14	6	10	12	15	30
15	7	11	13	14	31
16	0	17	18	20	24
17	1	16	19	21	25
18	2	16	19	22	26
19	3	17	18	23	27
20	4	16	21	22	28
21	5	17	20	23	29
22	6	18	20	23	30
23	7	19	21	22	31
24	8	16	25	26	28
25	9	17	24	27	29
26	10	18	24	27	30
27	11	19	25	26	31
28	12	20	24	29	30
29	13	21	25	28	31
30	14	22	26	28	31
31	15	23	27	29	30

При наличии меньшего количества разрядов, чем рассмотрено выше, для кодирования позиционных систем счисления синтезировать оптимальный код по критерию максимального исправления ошибок весьма сложно, и любое техническое решение является компромиссом в ущерб каким-либо штатным состояниям.

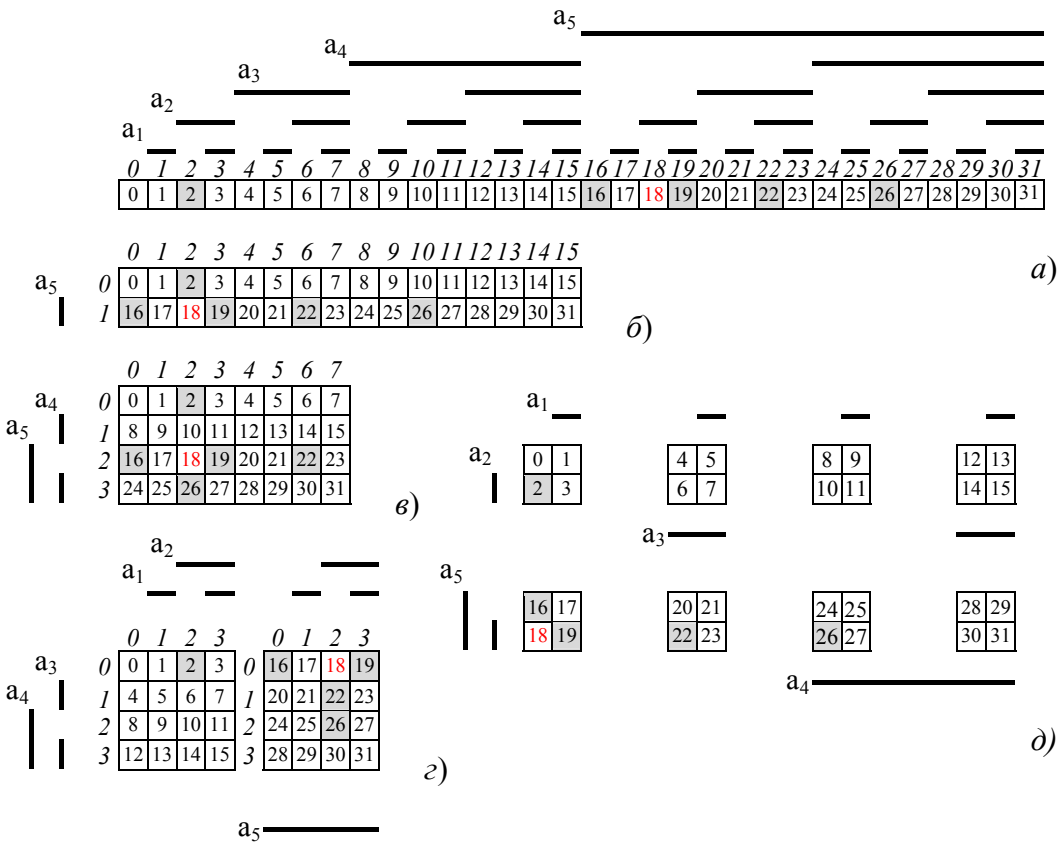
Поэтому здесь необходимо идти от простой задачи к более сложному ее варианту. Начнем этот путь с анализа существующих кодов оснований систем счисления, обладающих частичной контролеспособностью.

Перед началом такого рассмотрения представим аналитическую запись одиночных ошибок, которая следует из рис. 3.4. Эта запись может быть представлена простой табличной схемой, где безошибочные номера кодовых комбинаций выделены жирным шрифтом, а номера кодовых комбинаций их эквивалентов при одиночной ошибке – светлым шрифтом и меньшим размером.

Рис. 3.7

Левая часть табл. 3.1.1 определяет процедуру формирования этих номеров, а правая часть, которая полностью соответствует ее левой части, но где безошибочные номера кодовых комбинаций записаны в каждой строке первыми, наиболее удобна при построении геометрических образов соответствующих логических функций и будет использована нами в дальнейшем. Эту таблицу можно продолжить и далее, но в этом нет практической необходимости. В самом деле, таблица представляет номера ячеек с одиночными ошибками в одномерном цифровом пространстве, что возможно представить графически на листе 11 формата только для n = 5. На рис. 3.7, *а* показано такое одномерное цифровое пространство для этого случая. Здесь, например, для цифры **18** одиночные ошибки в соответствии с табл. 3.1.1 будут располагаться в ячейках с номерами 02, 16, 19, 22 и 26.

В эквивалентном исходному цифровому пространству двухмерном пространстве (см. рис. 3.7, *б*) координат $A_1(a_1, a_2, a_3, a_4)$, $A_5(a_5)$ цифра **18**, которая здесь является уже числом, расположена на пересечении координат A_1 – 02, A_2 – 01.



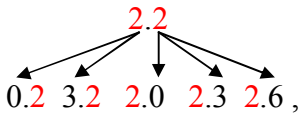
Тогда одиночные ошибки по координате A_1 (строка 02 таблицы) попадают в ячейки по координате A_1 в номера 0, 3, 6, 10, что соответствует цифрам ячеек 16, 19, 22, 26 , а по координате A_2 (строка 01 таблицы) – цифре 2. Это полностью совпадает с результатами исходного одномерного пространства.

Можно продолжить рассмотрение: в двухмерном цифровом пространстве координат $A_1(a_1, a_2, a_3)$, $A_5(a_4, a_5)$, для этого используется одна 02-я строка таблицы; для трехмерного цифрового пространства координат $A_1(a_1, a_2)$, $A_2(a_3, a_4)$, $A_3(a_5)$ это будут соответственно строки 2, 0, 1; и, наконец, для эквивалентного пятимерного цифрового пространства координат a_1, a_2, a_3, a_4, a_5 это будут только две строки 0 и 1 таблицы ($a_1 – 0$, $a_2 – 1$, $a_3 – 0$, $a_4 – 0$, $a_5 – 1$).

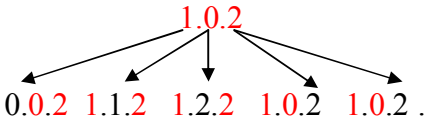
Очевидно, что в любом из этих эквивалентных многомерных цифровых пространств одиночные ошибки для любой из цифр одномерного цифрового пространства всегда попадают в номера ячеек, которые определяются одномерным цифровым пространством.

Поскольку координаты числа в многомерном цифровом пространстве являются цифрами соответствующих оснований систем счисления, то можно перейти от номеров ячеек в десятичной системе счисления к номерам ячеек в этих смешанных системах счисления.

Например, в двухмерном цифровом пространстве координат $A_1(a_1, a_2, a_3)$, $A_5(a_4, a_5)$ одиночные ошибки числа 18 десятичной системы счисления представляются следующим образом:



а в трехмерном цифровом пространстве координат $A_1(a_1, a_2)$, $A_2(a_3, a_4)$, $A_3(a_5)$ координаты одиночных ошибок этого числа определяются диаграммой



Для пятимерного цифрового пространства диаграмма определения координат одиночных ошибок определяется только первой и второй строками табл. 3.1.1, что приведет к следующему преобразованию кодовых комбинаций и соответствующим им цифровым значениям:

a_5	a_4	a_3	a_2	a_1		
1	0	0	1	0	=	18
1	0	0	1	1	=	19
1	0	0	0	0	=	16
1	0	1	1	0	=	22
1	1	0	1	0	=	26
0	0	0	1	0	=	2.

Для любого числа координат многомерного цифрового пространства при двоичном принципе кодирования их оснований систем счисления алгоритм определения координат одиночных ошибок, представленный выше, остается неизменным.

Последовательное применение данного алгоритма к координатам многомерного цифрового пространства позволяет определить координаты двойных, тройных и т.д. ошибок исходного числа любых оснований систем счисления.

Например, число 19, которое является одиночной ошибкой для числа 18, при повторной ошибке даст числа, определяемые следующим образом:

a_5	a_4	a_3	a_2	a_1		
1	0	0	1	1	=	19
1	0	0	1	0	=	18
1	0	0	0	1	=	17
1	0	1	1	1	=	23
1	1	0	1	1	=	27
0	0	0	1	1	=	3

и т.д.

Необходимо отметить, что наиболее удобно использовать представление одиночных ошибок в системах счисления больших оснований с двоичным принципом их кодирования, в цифрах десятичного кода, как это дано в табл. 3.1.1, поскольку при этом для их записи требуются небольшие размеры геометрических ячеек многомерного цифрового пространства.

При определении двойных, тройных и т.д. ошибок данные этой таблицы используются многократно: для двойных ошибок они применяются к каждой одиночной ошибке, для тройных ошибок – к каждой двойной ошибке и т.д.

Именно эти принципы были использованы нами при построении геометрических образов двойных и тройных ошибок, элементарной двоичной (n = 2) системы счисления, представленных на рис. 3.5, 3.6.

Вариант геометрического представления одиночных ошибок, заданных табл. 3.4.1, можно выполнить, используя трехмерное цифровое пространство координат рис. 3.8, где в вершинах «куба» размещены элементарные ячейки пространства, обозначенные кодовыми комбинациями 0–7. Линии, параллельные осям координат пространства и соединяющие вершины этого «куба», определяют изменения этих комбинаций при одиночных ошибках в их двоичных разрядах. Например, кодовая комбинация 0 при одиночной ошибке может стать 1 или 2 или 4; кодовая комбинация 1 при одиночной ошибке может стать 0 или 3 или 5 и т.д.

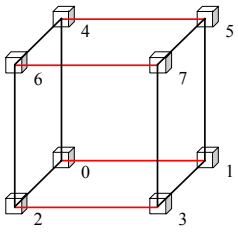


Рис. 3.8

Для четырехмерного цифрового пространства, где ячейки нумеруются кодовыми комбинациями 0–15, используется понятие «гиперкуба», где мерность пространства более трех.

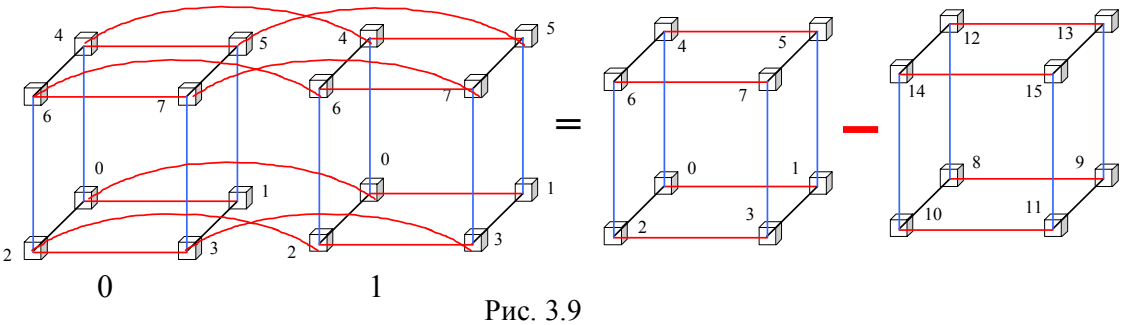


Рис. 3.9

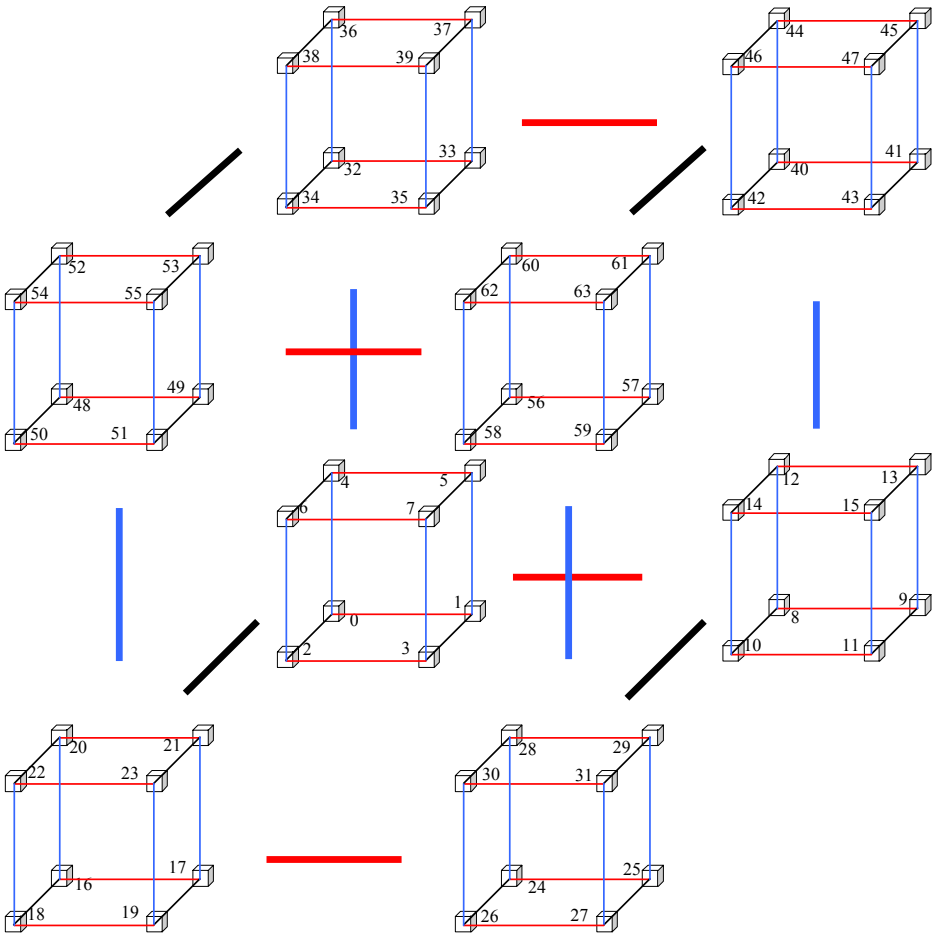


Рис. 3.10

На рис. 3.9 изображен «гиперкуб», составленный из двух «кубов», каждый из которых представляет собой основание системы счисления $n = 2^3$ (0–7), а вместе они представляют собой основание системы счисления $n = 2^4$ (0–15). Здесь приведены два эквивалентных варианта изображения этого «гиперкуба».

В первом варианте показаны все связи, определяющие одиночные ошибки, которые переводят одни кодовые комбинации в другие, где натуральные числа представлены двумя разрядами: первый разряд содержит цифры (0–7), второй – цифры 0, 1. Следовательно, первый «куб» представляет цифру 0 второго разряда, а второй куб – цифру 1 этого же второго разряда. Все одноименные элементарные ячейки этих «кубов» соединены линиями, определяющими переход кодовых комбинаций при одиночных ошибках.

Второй вариант представления «гиперкуба» более простой: в нем номера элементарных ячеек представлены цифрами одного разряда (0–15), а все соединения между двумя трехмерными «кубами» представляются общей жирной чертой, которая параллельна здесь оси координат x_1. Именно такой вариант представления «гиперкубов» будет использоваться нами в дальнейшем.

Данные табл. 3.1.1 отражаются в половине «гиперкуба» шестимерного пространства рис. 3.10. Этот «гиперкуб» состоит из восьми трехмерных «кубов», каждый из которых определяется и может задаваться рядом последовательных натуральных чисел (0–7), (8–15), (16–23), (24–31), …, (56–63). Жирные линии, параллельные осям координат x_1, x_2, x_3, соединяют соответствующие «кубы» и определяют одиночные ошибки, которые переводят натуральные числа одного «куба» или «гиперкуба» в другие числа одинаковых «кубов» или «гиперкубов». Например, все числа «куба» (0–7) при одиночных ошибках могут переходить в соответствующие числа внутри этого «куба», а также в числа «кубов» (8–15), (16–23), (32– 9) и т.д.

Очевидно, что не представляет какого-либо труда изобразить «гиперкуб» любой мерности, поскольку увеличение мерности «гиперкуба» выполняется простым изображением ему подобного, соединенного с первым жирной чертой, параллельной соответствующей координате трехмерного цифрового пространства. Причем в качестве исходного может быть принят не «куб», а «гиперкуб» любой мерности. Например, если в качестве исходного принять «гиперкуб» мерности четыре (рис. 3.9), определяемый цифрами (0–15), то «гиперкуб» мерности семь может быть изображен рис. 3.11 и т.д.

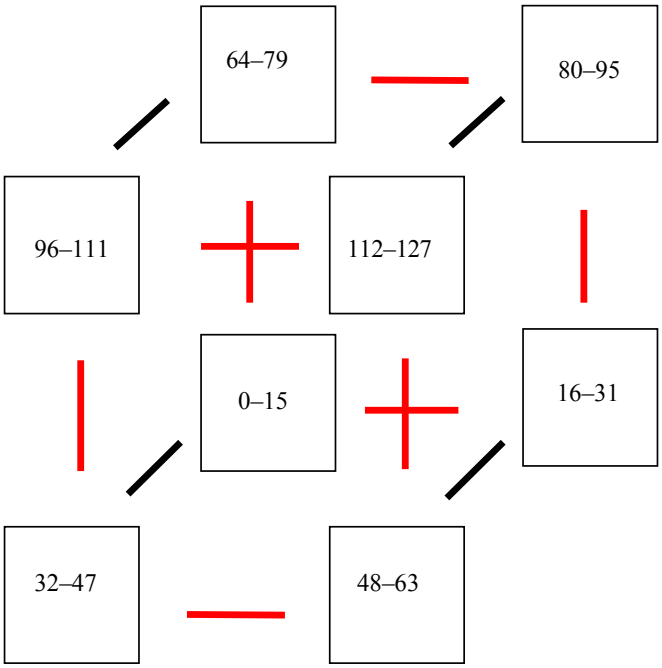


Рис. 3.11

3.2. Многофазный код

Важное место в системах электропривода и электропитания занимает многофазный код, являющийся естественным кодом целого ряда их устройств, где используются многофазные напряжения.

Целесообразно начать анализ с многофазного кода, поскольку этот тип кода наиболее исследован и для него решена задача обнаружения и исправления ошибок на основе особой физической структуры кода. Эта особенность кода заключается в возможности представления любого штатного состояния как составной части двух непрерывных множеств: логических нулей и единиц.

Известная процедура обнаружения и исправления ошибок [2] заключается здесь именно в контроле непрерывности этих множеств и восстановлении этой непрерывности при любых ее нарушениях. Следовательно, имеется возможность сравнить два способа контроля многофазного кода, отмеченные выше.

В качестве примера для рассмотрения выберем пятифазный код (n=10). Из 2^k возможных кодовых комбинаций многофазный код использует 2k штатные комбинации, которые для пятифазного кода (k=5) соответствуют десяти сигналам обычного цифрового кода (ОЦК) **0**–**9**, а остальные кодовые комбинации являются запрещенными и каждая из них представляется как ошибочная выдача одновременно нескольких сигналов ОЦК.

Последовательно записанные штатные кодовые комбинации, соответствующие цифрам от **0** до **9**, и образуют сигналы пятифазного кода $a_1 – a_5$.

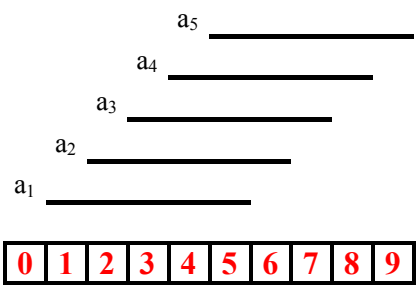


Рис. 3.12

Связь между сигналами $a_1 – a_5$ и эквивалентными им сигналами ОЦК приведена на рис. 3.12.

На рис. 3.13 приведены все 32 кодовые комбинации в двухмерном цифровом пространстве, где, например, по одной координате отложены позиции сигналов первых трех разрядов a_1, a_2, a_3, а по второй – остальных двух разрядов a_4, a_5. На пересечении координат здесь записаны эквивалентные кодовым комбинациям сигналы ОЦК – как штатные (выделены цветом), так и ошибочные.

Следует отметить, что эти сигналы соответствуют преобразованиям пятифазного кода в сигналы ОЦК по алгоритму

$$\mathbf{0} = \underline{a_1}\underline{a_5},\ \mathbf{1} = a_1\underline{a_2},\ \mathbf{2} = a_2\underline{a_3},\ \mathbf{3} = a_1\underline{a_4},\ \mathbf{4} = a_4\underline{a_5},$$
$$\mathbf{5} = a_1a_5,\ \mathbf{6} = \underline{a_1}a_2,\ \mathbf{7} = \underline{a_2}a_3,\ \mathbf{8} = \underline{a_3}a_4,\ \mathbf{9} = \underline{a_4}a_5$$

и их не следует путать с ошибочными сигналами, обозначение которых было приведено выше.

Покрытие «площади», ограниченной этими ошибочными сигналами ОЦК, определяет все возможные варианты построения логической схемы обнаружения «немногофазности», а покрытие «площади», ограниченной штатными сигналами для цифр **0**–**9**, определяет «многофазность» кода, одним из вариантов которой является логическая функция

$$\lambda = \underline{a_2}\underline{a_3}\underline{a_4}\underline{a_5} \vee a_1a_2\underline{a_4}\underline{a_5} \vee a_1a_2a_3a_4 \vee \underline{a_1}a_3a_4a_5 \vee \underline{a_1}\underline{a_2}\underline{a_3}a_5\,, \qquad (3.2.1)$$

где первое слагаемое покрывает штатные сигналы **0**, **1**, второе – **2**, **3** и т.д.

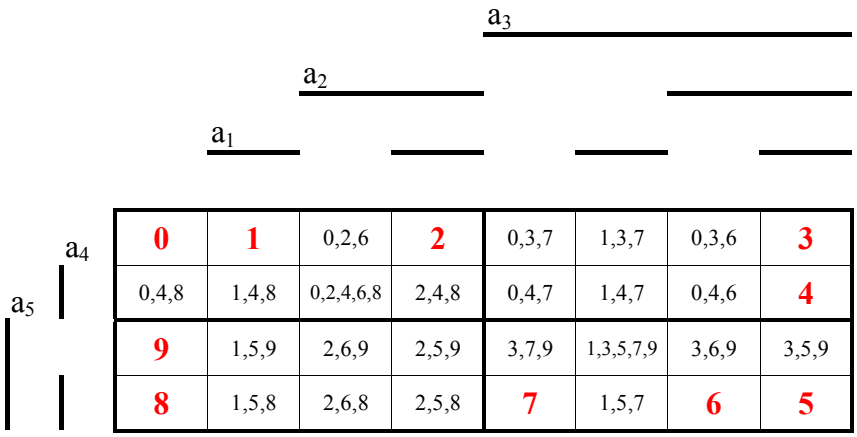


Рис. 3.13

Для обеспечения устойчивости работы счетчиков многофазного кода [3], когда при каких-либо сбоях и выходе счетчика из устойчивых рабочих режимов необходимо вернуть его в одно из устойчивых состояний, например начальное, используется именно сигнал «немногофазности» $\underline{\lambda}$, хотя наиболее правильным решением было бы возвращение счетчика в то рабочее состояние, из которого он был только что выведен. Решить эту задачу положительно можно только после анализа контролеспособности многофазного кода.

3.3. Анализ контролеспособности многофазного кода методом многомерных цифровых множеств

Этот анализ проведем, как ранее условились, на примере основания системы счисления n = 10.

В соответствии с распределением эквивалентных пятифазному коду сигналов ОЦК в ячейках двухмерного пространства, а также одиночных и двойных ошибок на рис. 3.14 приведено заполнение ячеек этого пространства указанными сигналами.

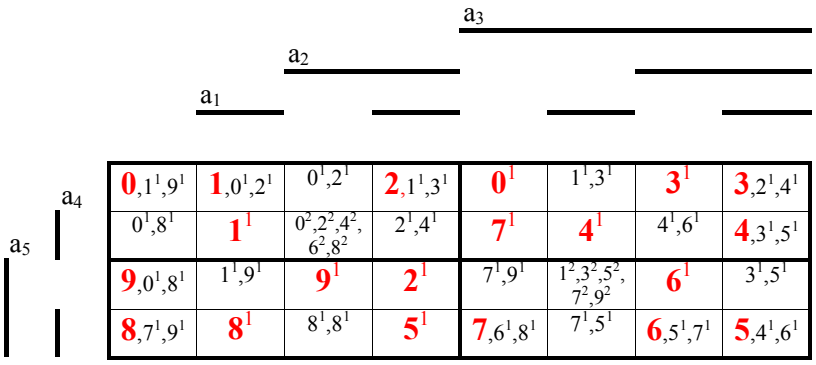


Рис. 3.14

В ячейках, которые уже заняты сигналами одиночных ошибок, сигналы двойных ошибок не приведены.

Из этого представления видно, что только одиночные ошибки имеют возможность исправления (один из пяти) и обнаружения (четыре из пяти). Двойные же ошибки занимают только две оставшиеся от размещения более вероятностных сигналов ячейки и не могут быть исправлены.

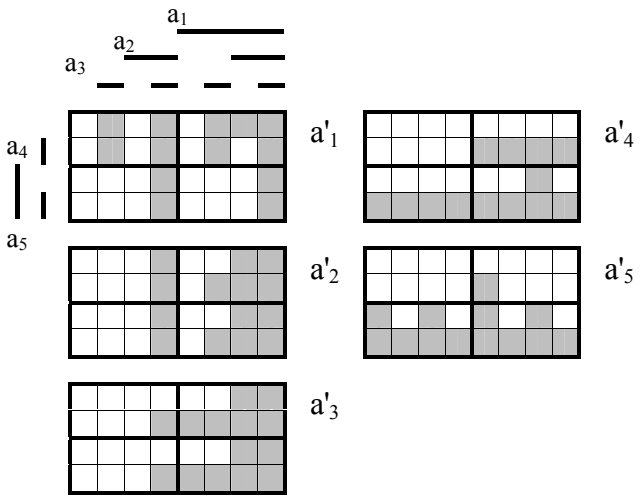


Рис. 3.15

В соответствии с рис. 3.14 и зависимостью сигналов многофазного кода a_1 – a_5 от сигналов ОЦК (см. рис. 3.12) для каждого геометрического образа сигналов a'_1 – a'_5 выберем ту ячейку пространства, которая однозначно включается в эти сигналы. Эта процедура позволяет получить в многомерном пространстве (рис. 3.15) пять фигур для сигналов многофазного кода, покрытие которых определяет логическую функцию блока исправления ошибок.

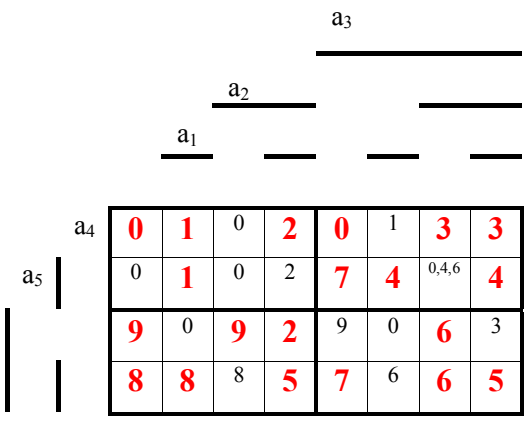


Рис. 3.16

Пропуск исправленных сигналов $a'_1 - a'_5$ через блок формирования сигналов ОЦК даст распределение сигналов ОЦК в ячейках пространства, как приведено на рис. 3.16. Это распределение полностью выполняет функции исправления ошибок наибольшей вероятности, а остальные одиночные ошибки исправляются стихийно в пользу каких-либо альтернативных вариантов равной вероятности. Причем здесь имеются ячейки под номерами 17, 29, 21, где установка сигналов ОЦК произведена не в альтернативном, а в худшем варианте, чем это предлагается на рис. 3.14.

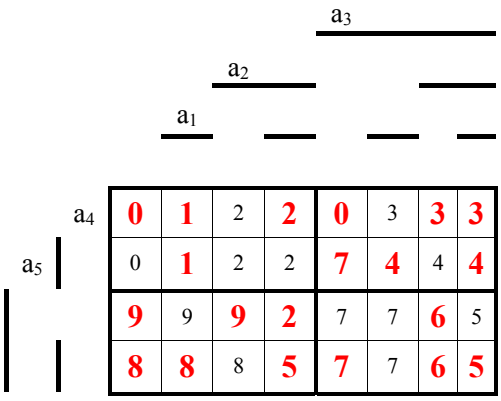


Рис. 3.17

Для исправления этого не совсем удачного размещения сигналов ОЦК можно заранее принять волевое решение о выборе конкретной цифры в ячейке в соответствии с альтернативами рис. 3.14.

Такое распределение приведено на рис. 3.17, а соответствующие ему геометрические образы сигналов $a'_1 - a'_5$ – на рис. 3.18. Логические функции, определяющие покрытие фигур этих сигналов, более просты, чем покрытие фигур рис. 3.15, а результат исправления предпочтительней.

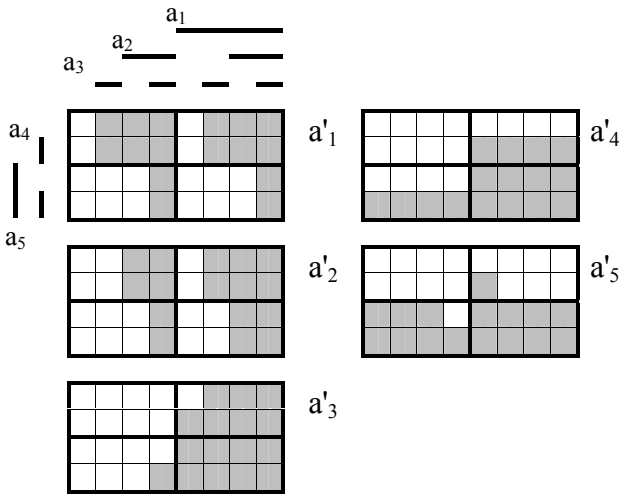


Рис. 3.18

Именно такой подход наиболее прост и эффективен для обеспечения устойчивости счетчиков многофазного кода.

В блоках машинной арифметики, где выбор каких-либо альтернативных вариантов распределения сигналов ОЦК не допустим, поскольку приводит к аварийным ситуациям, рабочими режимами являются только штатные состояния кода и те состояния, которые непрерывно исправляются контролирующими устройствами.

Для этого варианта построения блока исправления ошибок в многомерном пространстве заполняются только ячейки штатного состояния и ячейки, где располагаются исправленные кодовые комбинации.

Геометрические образы исправленных сигналов (рис. 3.19) для этого варианта более сложные для покрытия.

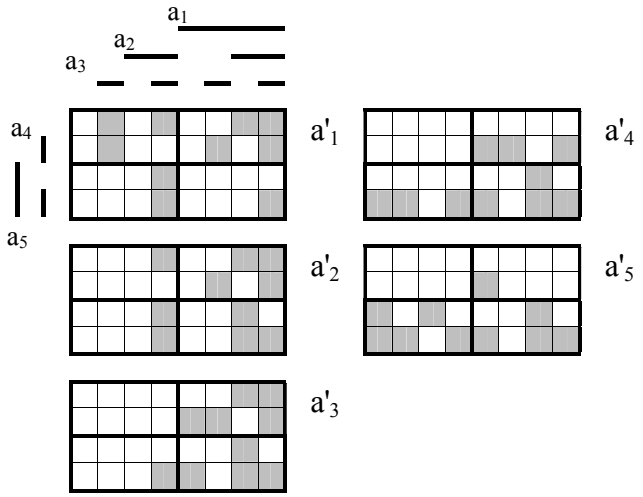


Рис. 3.19

В этом варианте обязательно необходимо определение бесконтрольной нерабочей области пространства $\underline{\lambda}$, появление которой требует остановки вычислительного процесса, поскольку в этой области пространства сигналы кода недействительны. При этом в нерабочую область пространства не должны входить нулевые значения сигналов кода, которые эквивалентны цифре **0**.

Известность сигнала $\underline{\lambda}$ позволяет использовать более простые геометрические образы сигналов $a'_1 - a'_5$, чем такие же сигналы рис. 3.19, например сигналы рис. 3.18.

Определение оптимальных, т.е. более простых для покрытия геометрических образов сигналов $a'_1 - a'_5$, будет сформулировано следующим образом: обязательной частью этих оптимальных образов должны быть фигуры, например рис. 3.19, а поиск более простых фигур должен производиться добавлением к ним любых ячеек пространства, геометрический образ которого представлен на рис. 3.20.

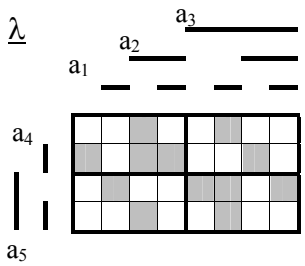


Рис. 3.20

Весьма прост по реализации и наиболее эффективен по быстродействию алгоритм исправления ошибок, например, многофазных кодов, в записи Либау–Крейга, который основан на контроле и сохранении непрерывности множеств логических нулей и единиц этого типа кодов [2]. Эти непрерывные множества всегда имеются при последовательной записи прямых и инверсных сигналов этих кодов. При этом здесь появляется возможность исправления не только одиночных, но и двойных, тройных и т.д. ошибок, а также различных пачек таких ошибок. Возможности этого алгоритма возрастают с увеличением числа фаз кода, но наиболее ценным является исправление именно одиночных ошибок.

В соответствии с этим алгоритмом исправление, например, одиночных ошибок пятифазного кода будет производиться по следующим логическим формулам:

$$
\begin{aligned}
a'_1 &= a_1\underline{a}_5 \vee a_1a_2 \vee \underline{a}_5a_2;\\
a'_2 &= a_1a_2 \vee a_2a_3 \vee a_1a_3;\\
a'_3 &= a_3a_2 \vee a_3a_4 \vee a_2a_4;\\
a'_4 &= a_4a_3 \vee a_4a_5 \vee a_3a_5;\\
a'_5 &= a_5a_4 \vee a_5\underline{a}_1 \vee a_4\underline{a}_1.
\end{aligned}
\qquad (3.3.1)
$$

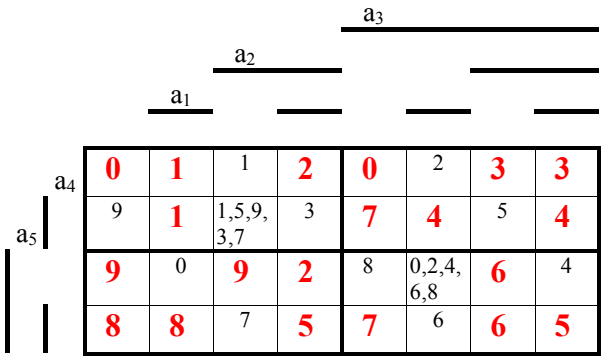


Рис. 3.21

На основании (3.3.1) и рис. 3.13 распределение сигналов ОЦК в ячейках многомерного пространства для этого варианта реализации приведено на рис. 3.21, а геометрические образы исправленных сигналов (a'_1– a'_5) – на рис. 3.22.

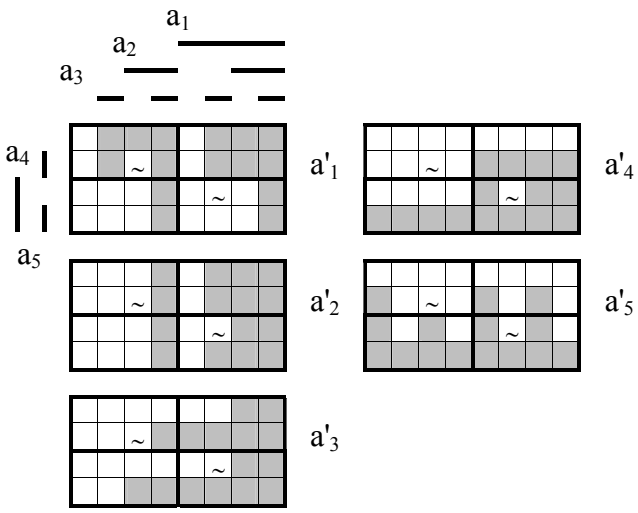


Рис. 3.22

Несмотря на то, что данный алгоритм не следует альтернативам распределения сигналов рис. 3.14, все исправляемые одиночные ошибки он действительно исправляет, а при наличии сигнала $\underline{\lambda}$ (рис. 3.20), что требуется при любых вариантах, здесь действительно аппаратурные затраты малы и осуществляется полное выполнение функций контроля и исправления ошибок.

3.4. Анализ контролеспособности кодов Хемминга методом многомерных цифровых множеств

Одним из широко известных в технике передачи информации является код Хемминга, предназначенный, например, для исправления одной ошибки или обнаружения двух и исправления одной ошибки [11].

Коды Хемминга имеют несколько контрольных разрядов, каждый из которых контролирует по четности (нечетности) свою группу разрядов. Эти группы формируются таким образом, чтобы по данным контрольных разрядов можно было бы с определенной вероятностью указать сбой соответствующего разряда.

С этой целью первый контрольный разряд контролирует по четности либо нечетности все нечетные информационные разряды кода. В нем при передаче информации записывается такая цифра, чтобы число единиц в контролируемых разрядах совместно с этим контрольным разрядом было нечетным (четным). Если при приеме и проверке кода окажется, что число единиц в нечетных разрядах и первом контрольном разряде четное (нечетное), то это с определенной вероятностью будет свидетельствовать об ошибке в нечетных разрядах. В противном случае имеется ошибка в четных разрядах или ее вообще нет.

Следовательно, сигнал о наличии ошибки в нечетных разрядах является младшим разрядом двоичного номера разряда с ошибкой.

Второй контрольный разряд контролирует по четности (нечетности) все разряды кода, которые имеют одинаковые цифры во втором разряде своего номера, третий контрольный разряд – все разряды, имеющие одинаковые цифры в третьем разряде своего номера, и т.д.

Исходя из изложенного, для определения следующего вероятного номера разряда с ошибкой необходимо сформировать аналогичную функцию от разрядов с номерами 2, 3, 6, 7, ... и вторым контрольным разрядом и т.д.

Все это можно наглядно представить в табл. 3.4.1, где обозначены номера информационных разрядов, которые контролируются соответствующими контрольными разрядами.

Таблица 3.4.1

Контрольные разряды	Информационные разряды														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															

В литературе по теории информации высоко оцениваются возможности кодов Хемминга по контролеспособности при приеме и передаче информации, а в публикациях по вычислительной технике эти достоинства кодов Хемминга обесцениваются утверждением о непригодности этих кодов для выполнения

операций машинной арифметики [10]. Как первое, так и второе утверждения, далеки от истинных свойств этого кода.

Поскольку метод многомерных цифровых множеств является универсальным инструментом для анализа любых типов корректирующих кодов, воспользуемся его достоинствами для исследования кодов Хемминга применительно к выполнению арифметических операций.

Выберем в качестве первого примера для исследований основание системы счисления n = 8 (a_1, a_2, a_3), что по табл. 3.4.1 определяет два контрольных разряда x_1, x_2.

В соответствии с алгоритмом Хемминга представим в многомерном пространстве сигналов a_1, a_2, a_3, x_1, x_2 штатные сигналы ОЦК **0** – **7**, а также их однократные ошибочные представления, которые все приведены на рис. 3.23.

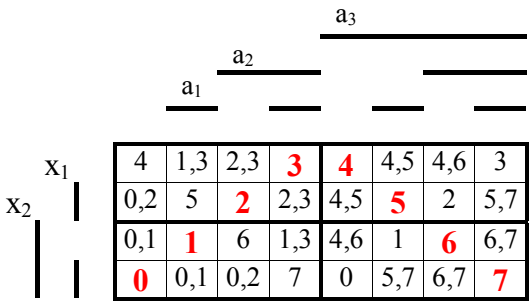


Рис. 3.23

Достоинством кода является, как это видно из рис. 3.23, оптимальное размещение сигналов **0** –**7** в ячейках пространства, когда в них нет одиночных ошибок. Все одиночные ошибки кода располагаются в пространстве нештатных состояний. Необходимо отметить, что если число контрольных разрядов обозначить через k, число кодов, эквивалентных коду Хемминга, будет (2^k**!**).

Из рис. 3.23 также очевидна связь между цифрами **0** –**7** основания и всеми сигналами кода Хемминга, что для большей наглядности приведено на рис. 3.24.

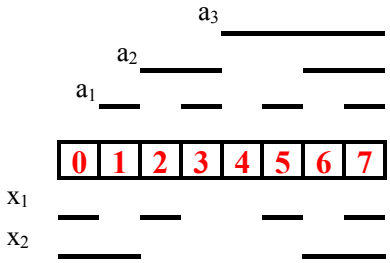


Рис. 3.24

Для того чтобы оценить достоверность предложенной Хеммингом аналитической методики исправления ошибок, представим на рис. 3.25 результаты такого исправления в многомерном цифровом пространстве.

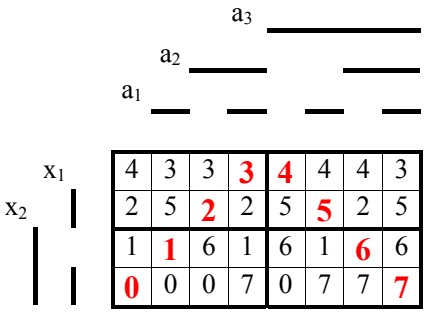


Рис. 3.25

Сравнивая результаты такого исправления с фактическим расположением одиночных ошибок (см. рис. 3.23), можно видеть, что все безальтернативные одиночные ошибки действительно исправляются, а в остальных случаях принимается «жесткий» вариант для одной из альтернатив подобно тому, как это выполнялось в одном из вариантов исправления многофазного кода.

Геометрические образы информационных a'_1, a'_2, a'_3 и контрольных x'_1, x'_2 сигналов такого «исправления», а также образ сигнала достоверного рабочего пространства λ, который в методике Хемминга не определяется, приведены на рис. 3.26.

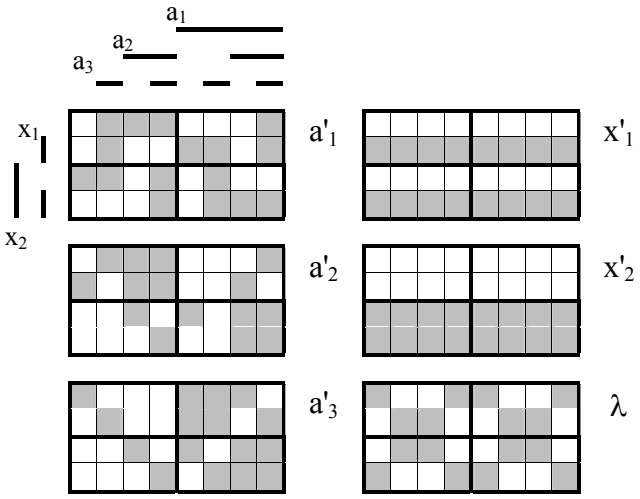


Рис. 3.26

Очевидно, что для возможности выполнения арифметических операций в кодах Хемминга знание сигнала λ обязательно и его равенство нулю будет оз-

начать неисправность и невозможность дальнейшего выполнения каких-либо арифметических операций.

Геометрические образы исправленных информационных a'_1, a'_2, a'_3 и контрольных x'_1, x'_2 сигналов при этом показаны на рис. 3.27. В эти образы, с целью их упрощения без потери информации, включены ячейки нерабочего пространства $\underline{\lambda}$.

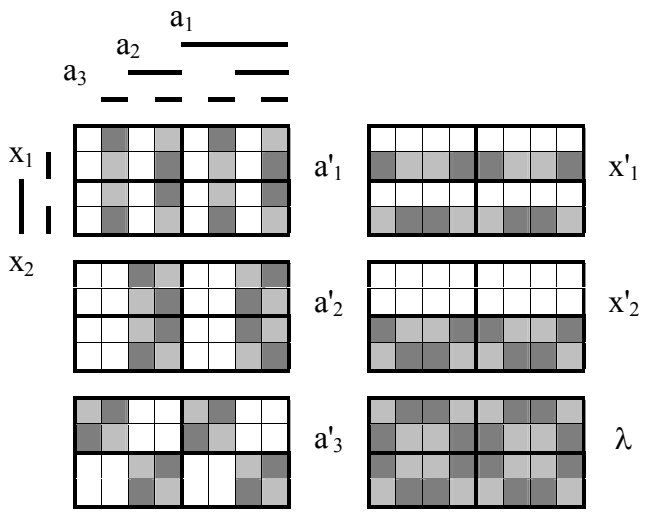


Рис. 3.27

Таким образом, представленный вариант кода Хемминга для кодирования основания n = 8 имеет весьма простую и быстродействующую процедуру исправления одиночных ошибок и полностью пригоден для выполнения любых арифметических операций. Особый интерес представляет код Хемминга, который рационально использует все многомерное пространство сигналов, т.е. где все пространство рабочее и всегда $\lambda = 1^*$. Такими кодами в соответствии с табл. 3.4.1 будут те, где общая длина кода информационных и контрольных разрядов равна 3, 7, 15, ...

В качестве второго примера рассмотрим именно такой код при кодировании основания n = 16, где имеются четыре информационных разряда a'_1, ... , a'_4 и три контрольных x_1, x_2, x_3.

Распределение в многомерном пространстве штатных сигналов **0–15**, а также сигналов одиночных ошибок для этого кода приведено на рис. 3.28, где штатные сигналы выделены.

Из этого представления очевидны зависимости между цифрами **0–15** и всеми сигналами кода, а также все геометрические образы действительно исправленных сигналов при наличии одной одиночной ошибки как в информационных, так и в контрольных разрядах. Первые зависимости приведены на рис. 3.29, а геометрические образы исправляемых сигналов кода – на рис. 3.30. Поскольку в этом случае $\lambda = 0^*$, то геометрические образы сигналов не могут подвергаться каким-либо упрощениям.

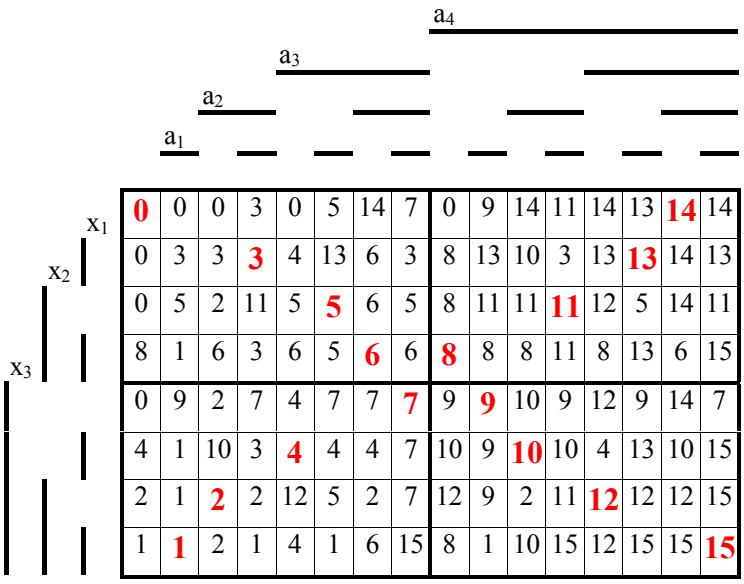


Рис. 3.28

Анализ кодов Хемминга показывает, что его процедура исправления ошибок требует больших аппаратурных затрат и поэтому мало приспособлена для включения в структуру арифметических устройств.

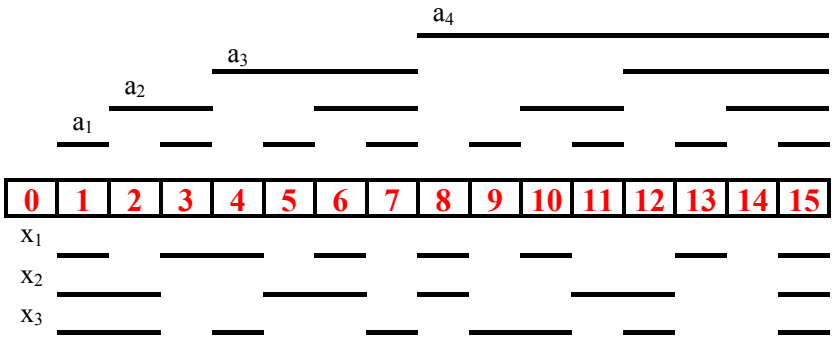


Рис. 3.29

Тем не менее в кодах Хемминга принципиально возможно создание быстродействующего логического устройства, например на ПЛМ, реализующего покрытие всех геометрических образов сигналов a'_1, a'_2, a'_3, a'_4, x'_1, x'_2, x'_3, λ.

Следовательно, вопрос об использовании кода Хемминга в том или ином арифметическом блоке определяется только аппаратурными затратами, исходя из требуемого быстродействия устройства.

Выполнение устройств машинной арифметики с использованием контролеспособных кодов, в том числе и кодов Хемминга, находится в стадии развития.

Их систематическое изложение требует большого объема, что будет рассмотрено в дальнейшем. Однако остановимся на положениях, которые могут быть положены в основу синтеза таких устройств. Для этого сделаем небольшой экскурс в проблему исправления ошибок в теории связи и ее отличия при построении помехоустойчивых цифровых устройств, составляющих предмет нашего анализа.

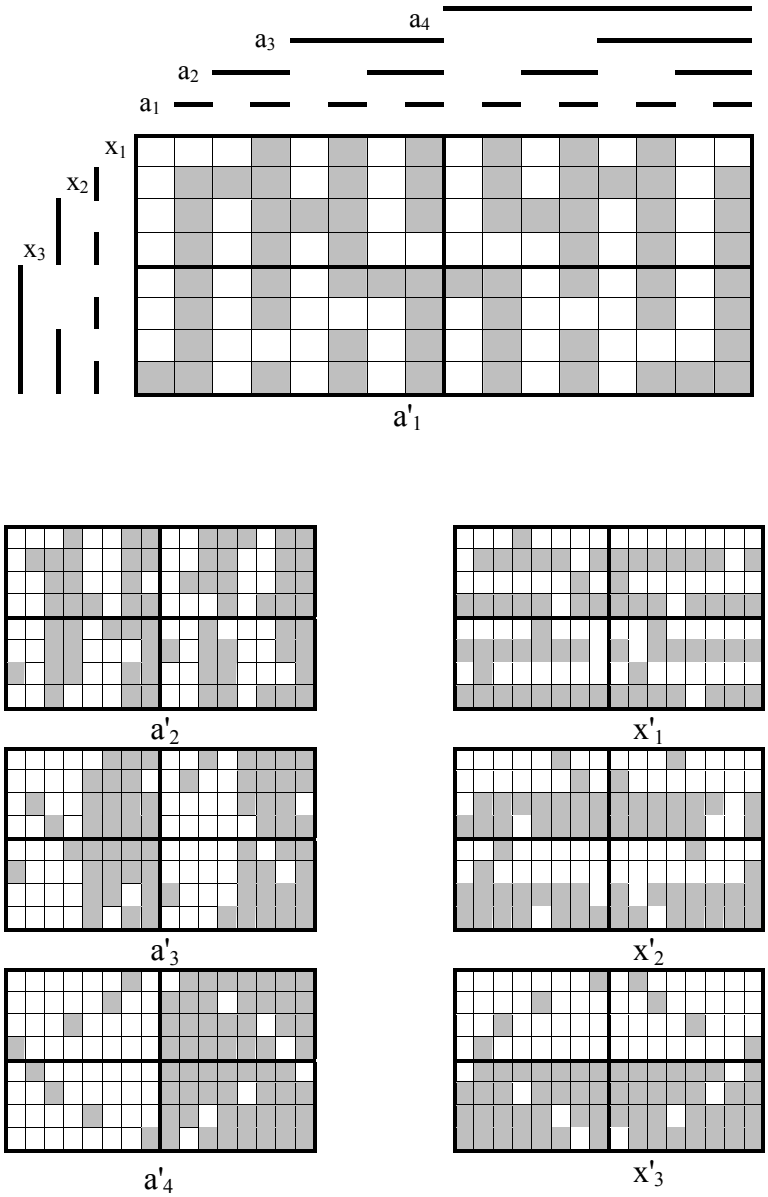


Рис. 3.30

Классическое решение задачи помехозащищенности в теории связи [6] заключается в посылке через линию связи с помехами, кроме информационных сигналов, соответствующим образом сформированных в кодирующем блоке контрольных сигналов (рис. 3.31, *а*).

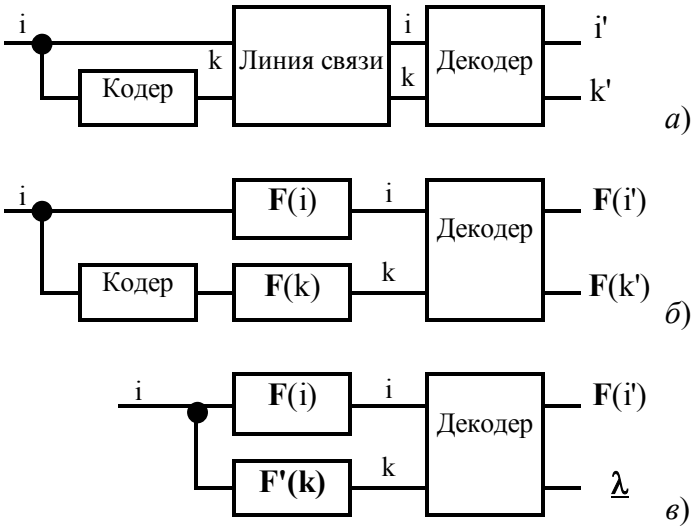


Рис. 3.31

На выходном конце линии связи эти две части кода поступают в декодирующий блок, который исправляет ошибки в обеих частях синтезирующего кода, где при успешном выполнении этой операции возможно активное использование информационной части кода либо дальнейшая совместная ретрансляция сигнала обеих его частей.

Формальный перенос из теории связи операции исправления ошибок на цифровые устройства, что выполняется практически во всех работах, представлен на структурной схеме рис. 3.31, *б*. В ней линия связи предыдущей схемы заменена на соответствующие блоки, реализующие определенные арифметические функционалы над информационной частью кода **F**(i) и его контрольной частью **F**(k), а кодирующие и декодирующие блоки сохранены без изменения. При попадании в нерабочую область цифрового пространства декодирующий блок выдает сигнал о невозможности исправления ошибок $\underline{\lambda}$ и необходимости остановить дальнейшее вычисление.

В описанной выше структурной схеме задача исправления ошибок в цифровом устройстве неоправданно усложнена. Без ущерба для решения задачи исправления ошибок в арифметическом либо логическом блоке устройства кодирующий блок и блок выполнения определенной операции над контрольной частью кода могут быть заменены одним блоком без промежуточного преобразования входной информационной части во входную контрольную часть кода.

Кроме того, декодирующий блок может быть также упрощен, поскольку нет необходимости иметь на его выходе контрольную часть кода выполняемой операции. Приведенные изменения структурной схемы изображены на рис. 3.31, *в*. Эта схема и может быть положена в основу синтеза арифметических и логических устройств, где основной задачей является создание нового блока, реализующего функционал **F'**(k).

3.5. Анализ контролеспособности обычного цифрового кода методом многомерных цифровых множеств

Название этого позиционного кода происходит от его повседневного использования для основания системы счисления n = 10 в обычной жизни всеми людьми. Напомним, что обычным цифровым кодом названо позиционное представление цифр счисления основания n ≥ 2, где в каждом разряде имеется n позиций с номерами **0**, **1**, ... , **(n – 1)** и лишь один символ (сигнал), вес которого равен номеру позиции, в которой символ в данный момент расположен. Иногда в литературе этот тип кода называют унитарным, однорядным, кодом с активным нулем, просто позиционным кодом и т.д.

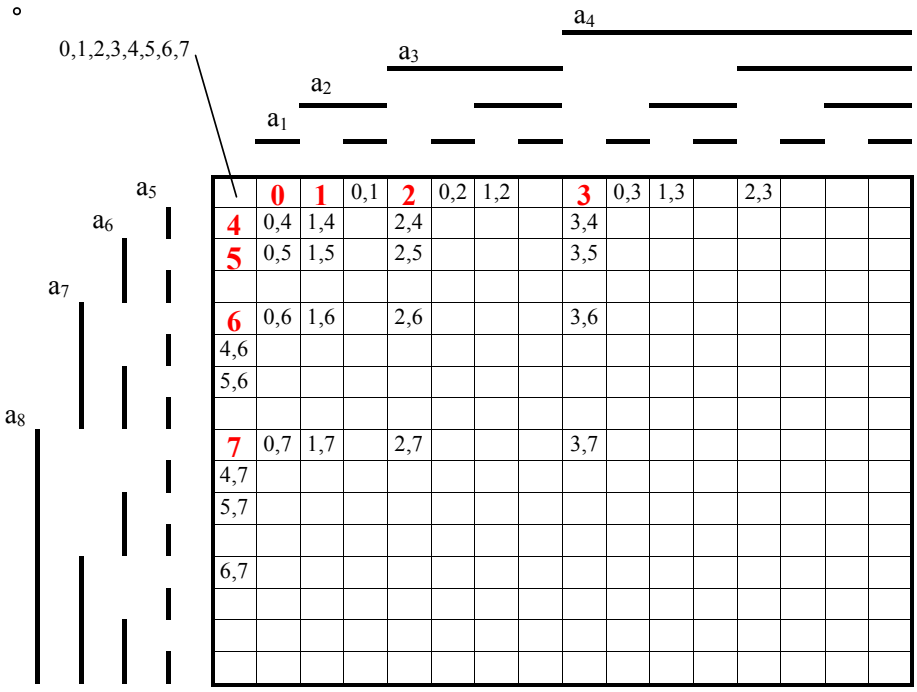


Рис. 3.32

С первых шагов создания вычислительных устройств предпринимались и предпринимаются не очень удачные попытки их выполнения в коде ОЦК [9]. Причем, учитывая огромную избыточность ОЦК, равную (2^n – n), ряд исследо-

вателей [5] видят в нем решение проблемы контролеспособности вычислительных устройств, забывая при этом, что этот код используется только для сокращенной записи цифрового вектора, где сообщается только старшая цифра вектора. Поэтому ОЦК должен рассматриваться как составляющая внешнего языка для ЭВМ и от него не следует ожидать каких-либо положительных эффектов при построении устройств, например машинной арифметики.

Докажем это утверждение на примере системы счисления основания $n = 8$. Обратимся к рис. 3.32, где изображено многомерное цифровое пространство, в ячейках которого расположены штатные сигналы **0–7** ОЦК, а также сигналы одиночных ошибок. Штатные сигналы выделены соответствующим образом.

Из представленного видно, что этот код не позволяет исправлять какие-либо ошибки, а схема обнаружения ошибок довольно громоздка:

$$\lambda = a_1\, \underline{a}_2\, \underline{a}_3 \ldots \underline{a}_8 \vee \underline{a}_1\, a_2\, \underline{a}_3 \ldots \underline{a}_8 \vee \underline{a}_1\, \underline{a}_2\, a_3 \ldots \underline{a}_8 \vee \ldots \vee \underline{a}_1\, \underline{a}_2\, \underline{a}_3 \ldots a_8 .$$

Поэтому если предположить, что этот код будет использован во внутреннем языке ЭВМ, то необходимо будет с каждой ошибкой в коде ее останавливать и устранять ошибку. Здесь уместно обратиться к Дж. фон Нейману, который писал: «Весьма вероятно, что если бы каждую ошибку надо было немедленно обнаружить, объяснить и исправлять, то система такой сложности, как живой организм, не смогла бы проработать и миллисекунды. Система такого типа так хорошо скомпонована, что может работать и тогда, когда в ней возникают отдельные ошибки» [7].

Поэтому необходимо стремиться к идеалу – живому организму и в этом стремлении наиболее подходит другой тип кода, который действительно может быть назван обычным цифровым кодом внутреннего языка ЭВМ. Этот код включает все цифры вектора, а в предыдущем рассмотрении был представлен как код одномерных угловых множеств, который, обладая даже несколько меньшей избыточностью (2^{n-1}) по сравнению с ОЦК, имеет огромные возможности по исправлению и обнаружению ошибок.

Этот тип кода, очевидно, и является языком человеческого общения. Если мы имеем, например, 6 рублей, то это значит, что у нас есть все составляющие этой суммы. Следовательно, рассматривать одиночное множество 6 не следует: это всегда сумма всех единичных множеств. Поэтому такой код можно также называть интегральным. Его отличие от языка человеческого общения заключается лишь в том, что цифра **0** равноценна здесь со всеми другими цифрами кода соответствующего основания системы счисления.

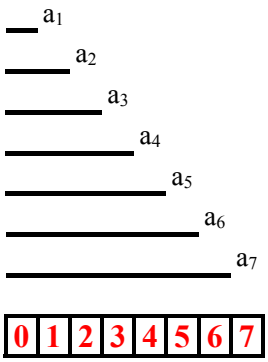


Рис. 3.33

Для основания n = 8 сигналы ОЦК относительно полюса α весьма просты a_1 = **0**, a_2 = **0** ∨ **1**, ... , a_7 = **0** ∨ ... ∨ 6 (рис. 3.33), а расположение в ячейках многомерного цифрового пространства штатных сигналов ОЦК, что показано на рис. 3.34, позволяет обнаруживать и исправлять, не останавливая ЭВМ, не только одиночные ошибки, но также ряд двойных и даже тройных ошибок.

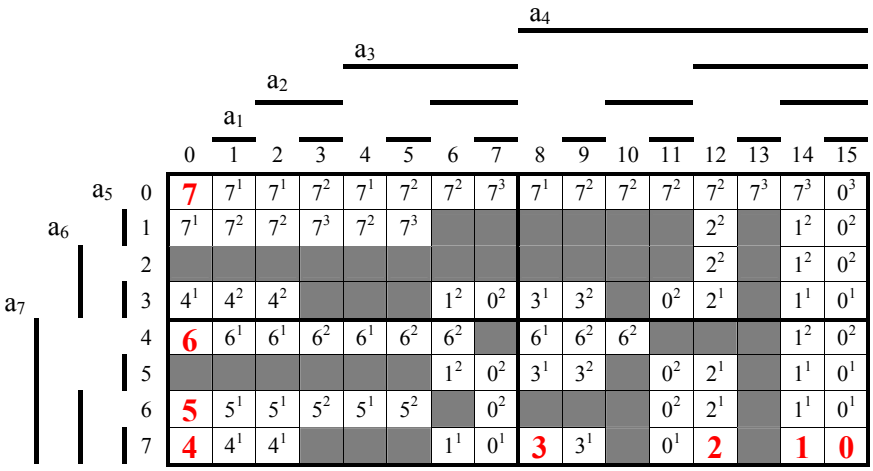


Рис. 3.34

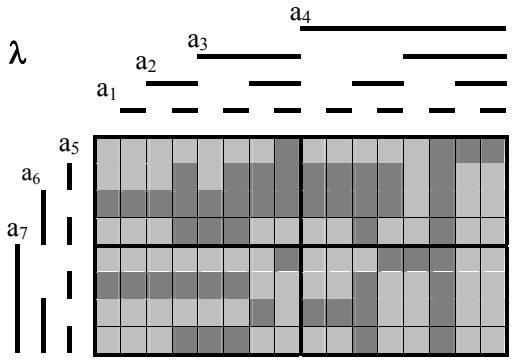


Рис. 3.35

Без учета сигналов тройных ошибок область рабочего пространства данного кода представляется фигурой рис. 3.35, а геометрические образы исправленных сигналов a'_1 – a'_7, дополненные с целью упрощения ячейками нерабочего пространства, приведены на рис. 3.36.

Если еще более упростить задачу и исправлять, например, только одиночные ошибки кода и при этом использовать сигналы относительно полюсов и α, β, то логические блоки определения сигнала λ и исправленных сигналов a'_1 – a'_7 будут еще более просты.

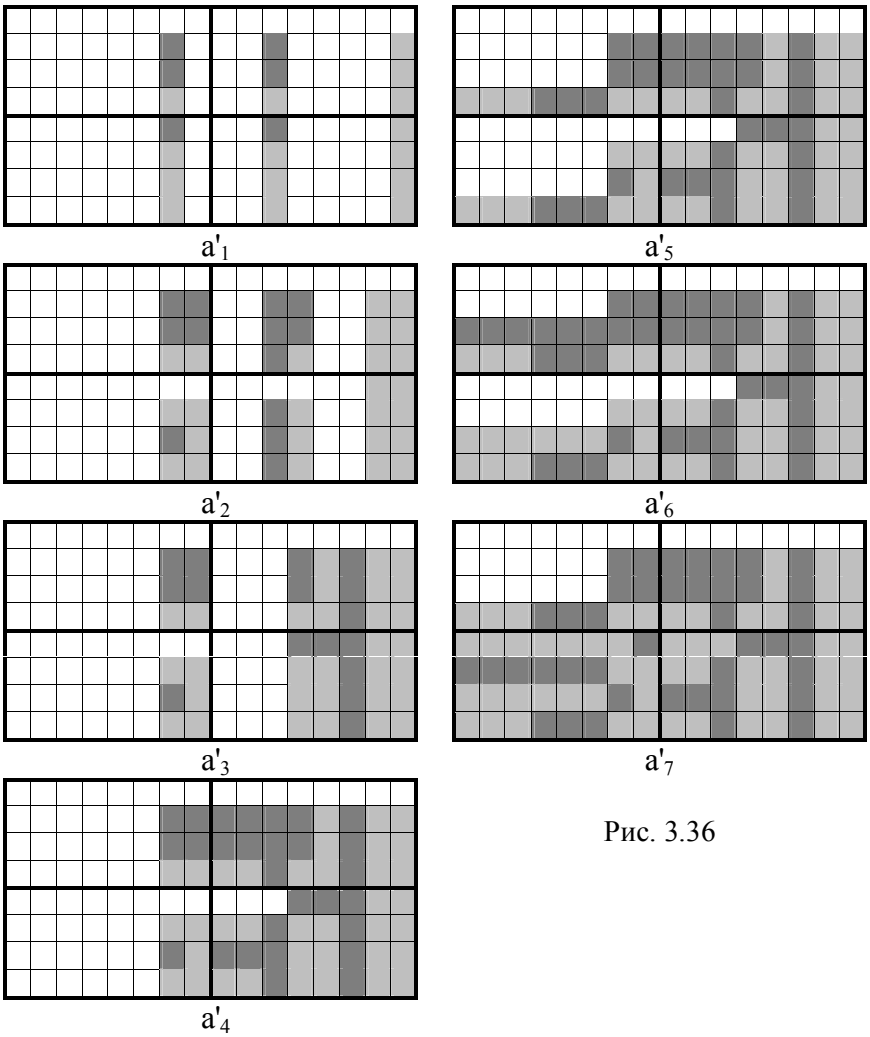


Рис. 3.36

Этот вариант кодирования сигналов с основанием $n = 8$ приведен на рис. 3.37, а многомерное пространство с обозначением штатных ячеек и ячеек пространства с одиночными ошибками и геометрические образы сигналов λ, $a'_1 - a'_7$ – соответственно на рис. 3.38 и 3.39.

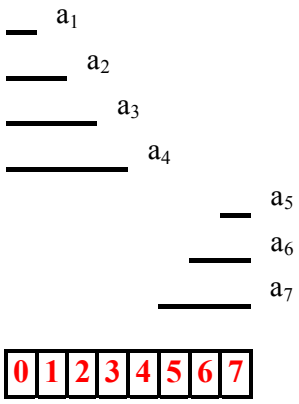


Рис. 3.37

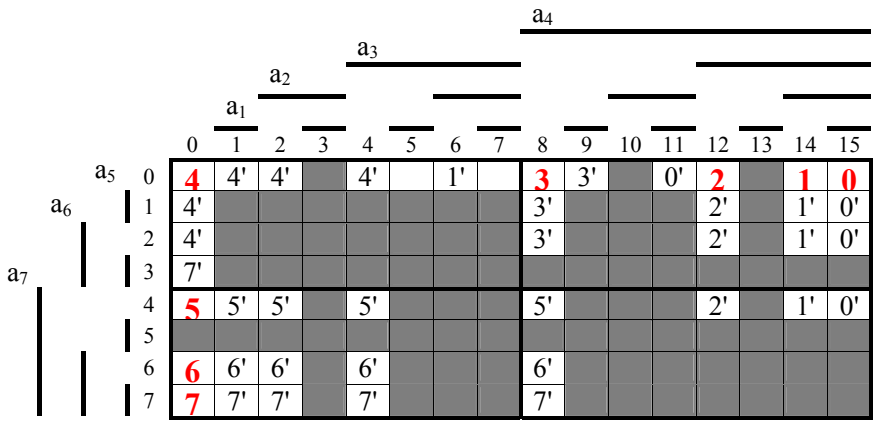


Рис. 3.38

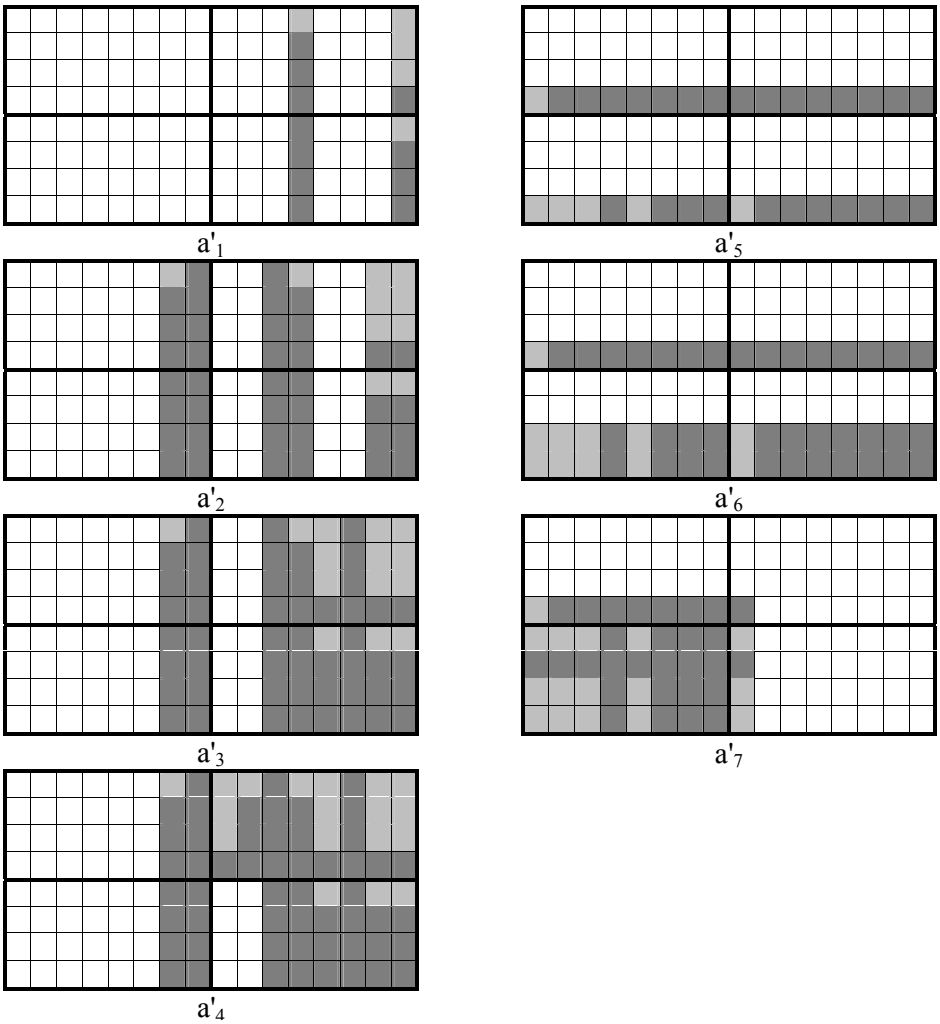


Рис. 3.39

Здесь необходимо отметить, что код одномерных угловых множеств так же, как многофазный код, имеет замечательное свойство, заключающееся в том, что в его каждой кодовой комбинации замкнутый ряд сигналов $\underline{a}_m\, a_1 \ldots a_m\, \underline{a}_1 \ldots \underline{a}_m$ содержит два непрерывных множества сигналов: множество нулей 0* и множество единиц 1*.

Это качество кода позволяет применять весьма простую по технической реализации процедуру обнаружения и исправления любых типов ошибок: одиночных, двойных и т.д., а также их комбинаций и пачек ошибок подобно тому, как это выполняется в многофазном коде [2]. При этом следует заметить, что если в цифровой системе либо отдельном цифровом блоке непрерывно осуществляется обнаружение с одновременным исправлением одиночных ошибок с высоким быстродействием, например за два уровня срабатывания одиночного вентиля, то в такой системе практически невозможно появление ошибок большей кратности, тем более пачек ошибок. Поскольку здесь будем рассматривать полностью контролеспособные цифровые системы, то станем в дальнейшем уделять больше внимания исправлению одиночных ошибок в режиме реального времени.

3.6. Анализ контролеспособности кода реверсивного двоичного делителя-счетчика

В реверсивном делителе-счетчике [4], где режим работы определяется порядком следования входных сигналов, имеются на выходных шинах два жестко связанных между собой кода: двоичный код i разрядности и код Грея разрядности (i – 1).

Сигналы такого синтезированного кода, предназначенного для представления сигналов ОЦК основания $n = 2^i$, содержат большую избыточность $2^i(2^{i-1} - 1)$. Поскольку делитель-счетчик с этим типом кода широко используется в цифровых устройствах электропривода, преобразователях угла, генераторах многофазного кода и т.д., целесообразно рассмотреть возможности этого кода по обнаружению и исправлению различных типов ошибок.

Для основания n = 4 соотношения между сигналами ОЦК **0**–**3**, двоичного кода a_1, a_2 и кода Грея q_1 (рис. 3.40) весьма просты. В качестве информационных сигналов в этом синтезированном коде могут быть приняты сигналы a_1, a_2 либо q_1, a_2, тогда контрольными сигналами будут соответственно сигналы q_1 либо a_1.

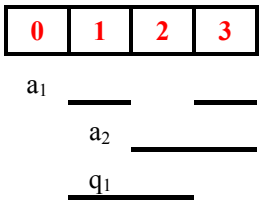


Рис. 3.40

Для дальнейшего исследования контролеспособности кода в данном параграфе примем для любых оснований систем счисления выполнение информационных сигналов в двоичном коде. Распределение в многомерном цифровом пространстве сигналов, кодируемых ими штатных сигналов ОЦК, и их одиночных ошибок приведено на рис. 3.41, из которого очевидна возможность

только обнаружения этих ошибок. Здесь сигнал безошибочности кода, т.е. сигнал рабочей области пространства, следует из его геометрического образа (рис. 3.42), а исправления информационных и контрольных сигналов здесь нет.

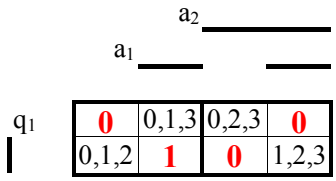


Рис. 3.41

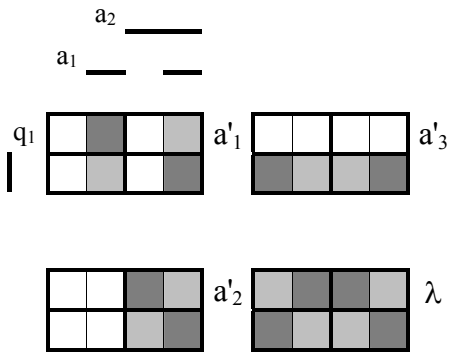


Рис. 3.42

Для следующего основания n = 8 на рис. 3.43 приведены соотношения между сигналами ОЦК **0**–**7**, сигналами двоичного кода a_1, a_2, a_3 и сигналами кода Грея q_1, q_2.

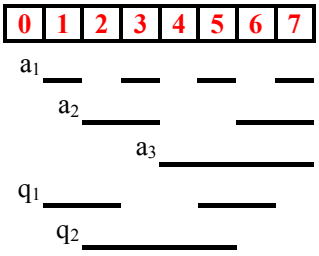


Рис. 3.43

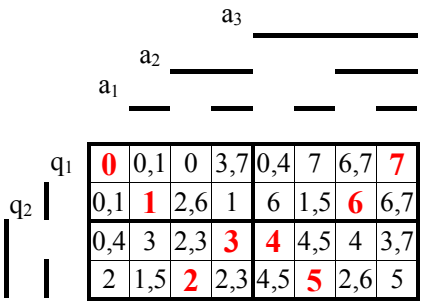


Рис. 3.44

В качестве информационных сигналов в этом случае можно принять следующие группы сигналов: a_1, a_2, a_3; q_1, q_2, a_3; q_1, a_2, a_3, тогда контрольными будут соответственно следующие сигналы: q_1, q_2; a_1, a_2; a_1, q_2.

Из распределения штатных сигналов ОЦК и их одиночных ошибок в ячейках многомерного цифрового пространства, которое приведено на рис. 3.44, видно, что этот код является разновидностью кода Хемминга с равноценной с ним контролеспособностью. Поэтому геометрический образ сигнала λ полностью совпадает с кодом Хемминга, а информационные и контрольные сигналы с упрощением их за счет ячеек нерабочего пространства (рис. 3.45) весьма просты:

$$a'_1 = a_1,\ a'_2 = a_3\underline{q_2} \vee \underline{a_3}q_2,\ a'_3 = a_3,\ q'_1 = q_1,\ q'_2 = q_2\,. \qquad (3.6.1)$$

Совпадение кода Хемминга с нашим синтезированным кодом начинается и заканчивается на основании n = 8.

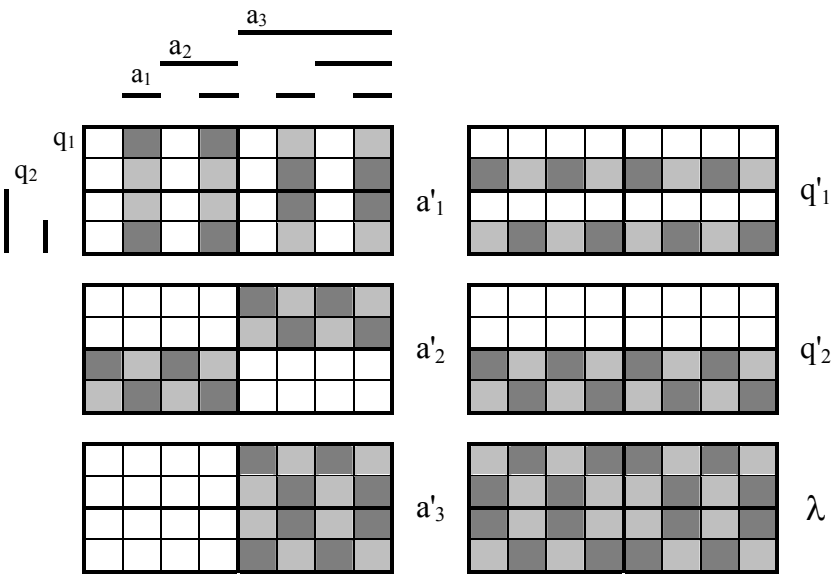


Рис. 3.45

Для основания n = 16, в котором код Хемминга рационально использует все многомерное цифровое пространство (λ =1*), код реверсивного делителя-счетчика не имеет таких возможностей.

Для пояснения этого обратимся к рис. 3.46, где приведены соотношения между сигналами синтезированного кода a_1, ... , a_4, q_1, q_2, q_3 и эквивалентными им сигналами ОЦК **0–15**.

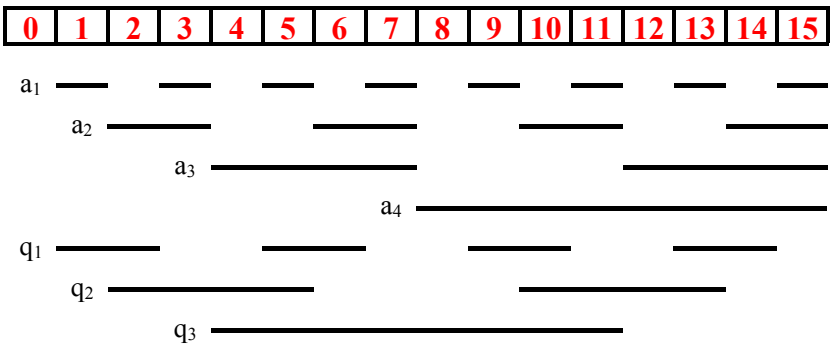


Рис. 3.46

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
q_1	0	**0**	0,1	0	3	0			7,15	0,8			15	12	15	14,15	**15**
q_2	1	0,1	**1**	2	1		1	6,14			1,9	14		14	13	**14**	14,15
	2	0	3	2,3	**3**	4,12			3	12			3,11	**12**	12,13	12	15
q_3	3	2	1	**2**	2,3		5,13	2			13	2,10		12,13	**13**	14	13
	4	0,8			7	4	7	6,7	**7**	**8**	8,9	8	11	8			7,15
	5		1,9	6		6	5	**6**	6,7	8,9	**9**	10	9		9	6,14	
	6	4			3,11	**4**	4,5	4	7	8	11	10,11	**11**	4,12			11
	7		5	2,10		4,5	**5**	6	5	10	9	**10**	10,11		5,13	10	

Рис. 3.47

В данном случае в качестве информационных могут быть выбраны следующие группы сигналов: $a_1, \ldots, a_4$; q_1, q_2, q_3, a_4; q_1, a_1, q_3, a_4, а контрольными сигналами для каждой из групп будут оставшиеся сигналы синтезированного кода.

Распределение штатных сигналов **0–15** и их одиночных ошибок в ячейках многомерного пространства приведено на рис. 3.47, из которого видно, что одиночные ошибки могут быть обнаружены, но только часть из них может быть исправлена. Незаполненные ячейки многомерного пространства определяют часть двукратных ошибок, которые могут быть только обнаружены.

Из 128 ячеек многомерного цифрового пространства в 64 ячейках располагаются штатные цифровые сигналы **0–15** и исправленные сигналы 0–15, а остальные 64 ячейки пространства делят поровну между собой неисправляемые одиночные и двойные ошибки.

В соответствии с известной из предыдущего рассмотрения методикой анализа на рис. 3.48 приведены геометрические образы безошибочности кода λ и всех сигналов упрощенного исправления ошибок

$$a'_1 = a_1,\ a'_2 = a_1\underline{q}_1 \vee \underline{a}_1 q_1,\ a'_3 = a_4\underline{q}_3 \vee \underline{a}_4 q_3,\ a'_4 = a_4,\ q'_1 = q_1;$$
$$q'_2 = (a_1\underline{q}_1 \vee \underline{a}_1 q_1)\,(\underline{a}_4\underline{q}_4 \vee a_4 q_4) \vee (a_1 q_1 \vee \underline{a}_1\underline{q}_1)\,(\underline{a}_4 q_3 \vee a_4\underline{q}_3);$$
$$q'_3 = q_3. \qquad (3.6.2)$$

Исходя из известности геометрических образов сигналов синтезированного кода $a'_1, \ldots, a'_i, q'_1, \ldots, q'_{i-1}$, λ основания $n = 2^i$, сформулируем общие правила построения геометрических образов сигналов для следующего старшего основания $n = 2^{i+1}$. С этой целью выделим в $(2i - 1)$-мерном цифровом пространстве, где расположены геометрические образы сигналов основания $n=2^i$, восемь неэлементарных ячеек. Для геометрического образа каждого из названных выше сигналов эти ячейки заполняются четырьмя геометрическими фигурами множеств $\mathbf{M}_{1j}, \ldots, \mathbf{M}_{4j}$, как это показано на рис. 3.49.

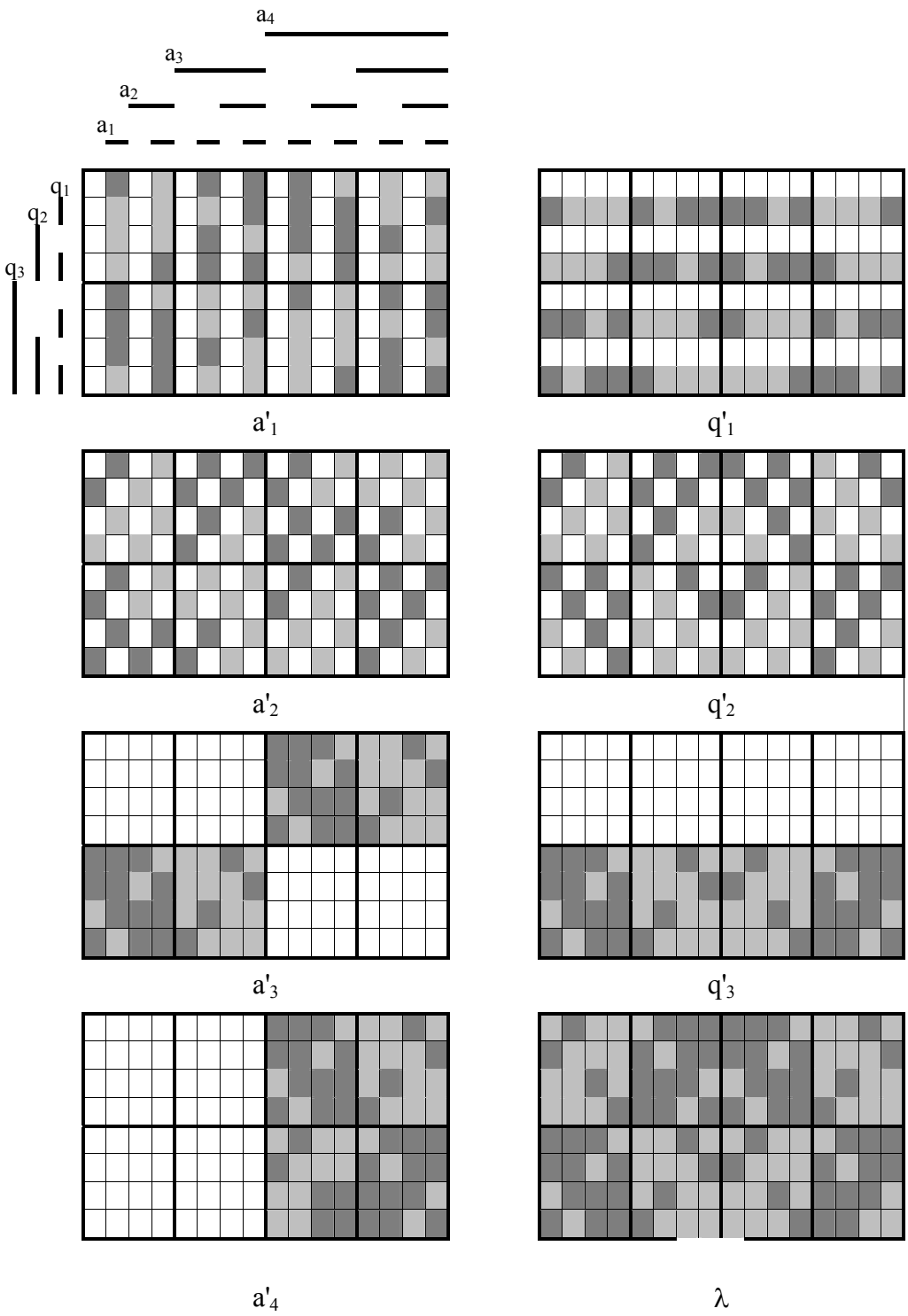
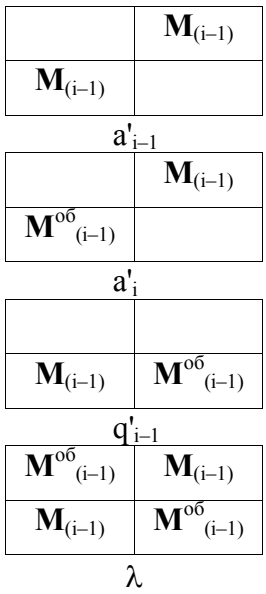


Рис. 3.48



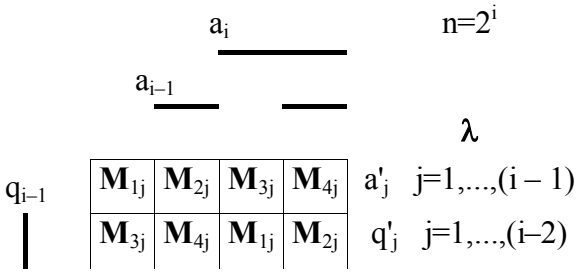


Рис. 3.49

Причем для геометрических образов сигналов a'_{i-1}, a'_i, q'_{i-1} многомерное цифровое пространство может быть разбито и на более крупные четыре ячейки, которые заполняются геометрическими фигурами $\mathbf{M}_{(i-1)}$, $\mathbf{M}^{об}_{(i-1)}$, связанные между собой простыми соотношениями:

$$\mathbf{M}^{об}_{(i-1)}(\mathbf{AQ}) = \mathbf{M}_{(i-1)}(\mathbf{AQ}^{\bullet}). \qquad (3.6.3)$$

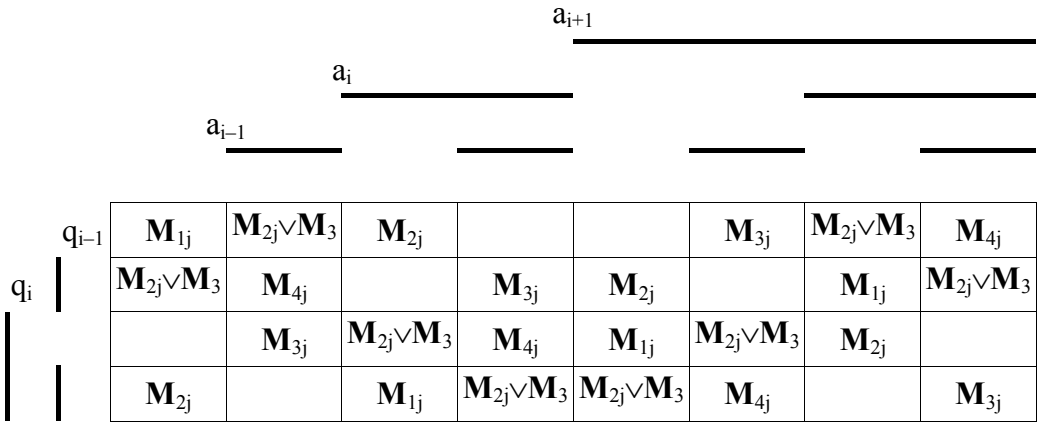


Рис. 3.50

Любой геометрический образ сигналов $a'_1, \ldots, a'_i, q'_1, \ldots, q'_{i-1}$, λ основания $n=2^{i+1}$ может быть получен из одноименных сигналов основания $n=2^i$ по алгоритму, представленному в пространстве $(2i + 1)$, как показано на рис. 3.50.

Причем знание геометрического образа сигнала λ либо a'_i основания $n=2^{i+1}$ определяет непосредственно геометрические образы сигналов a'_{i+1}, q'_i этого основания аналогично тому, как это показано для основания $n=2^i$ на рис. 3.49.

Геометрические образы составляющих $\mathbf{M}^{\lambda}_{1j}$, $\mathbf{M}^{\lambda}_{2j}$, $\mathbf{M}^{\lambda}_{3j}$, $\mathbf{M}^{\lambda}_{4j}$ сигнала λ связаны между собой следующими соотношениями:

$$\mathbf{M}^{\lambda}_{1j}(\mathbf{AQ}) = \mathbf{M}^{\lambda}_{4j}(\mathbf{A}^{\bullet}\mathbf{Q}),\ \mathbf{M}^{\lambda}_{2j}(\mathbf{AQ}) = \mathbf{M}^{\lambda}_{3j}(\mathbf{A}^{\bullet}\mathbf{Q}). \qquad (3.6.4)$$

В соответствии с представленным алгоритмом на рис. 3.51 приведен геометрический образ сигнала λ для основания $n = 32$, который синтезирован непосредственно из геометрического образа сигнала рис. 3.48 основания $n = 16$.

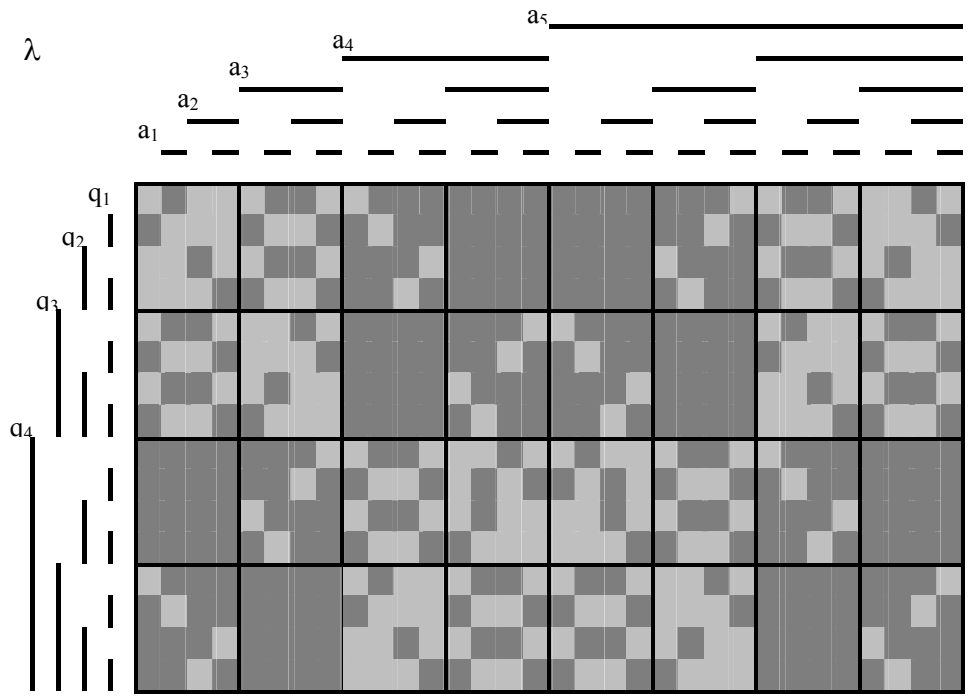


Рис. 3.51

Первообразными для определения геометрических образов исправляемых сигналов всех последующих оснований систем счисления являются сигналы основания $n = 8$, а не $n = 4$, поскольку в последнем нет возможности для исправления одиночных ошибок.

Если же в качестве первообразного принять сигнал λ основания n = 4, то, синтезируя по предложенному выше алгоритму аналогичные сигналы для оснований n = 8, 16, 32, ... , получаем геометрические образы сигналов λ, включающие в себя только штатные сигналы ОЦК этих оснований, в чем можно непосредственно убедиться, проведя соответствующий синтез.

Таким образом, код реверсивного двоичного делителя-счетчика при любой его протяженности обладает возможностью обнаружения и исправления ошибок. Структурная схема для обнаружения и исправления ошибок этого кода приведена на рис. 3.52, где блок *1* выполняет задачу определения рабочей зоны многомерного цифрового пространства кода, а блок *2* непосредственно исправляет ошибки, используя процедуру синтеза более простых геометрических образов исправляемых сигналов введением в них клеток нерабочего цифрового пространства $\underline{\lambda}$.

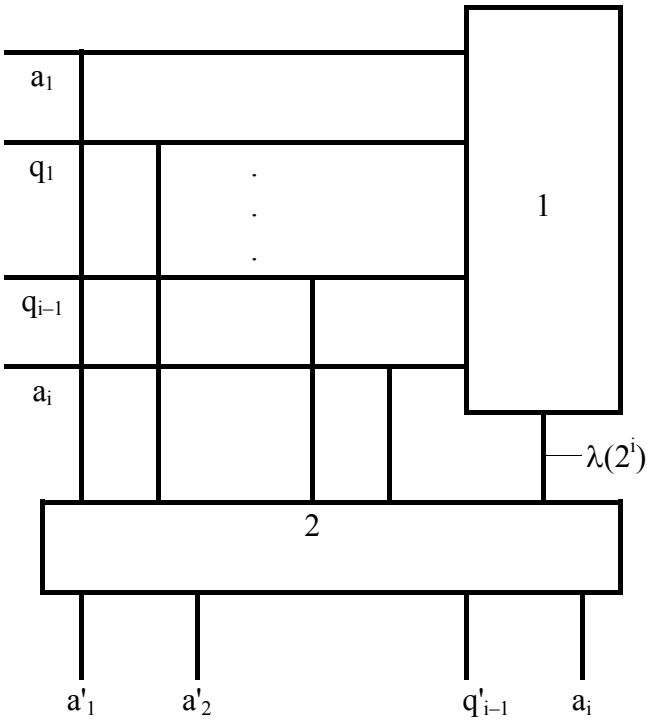


Рис. 3.52

Для быстродействующего устройства обнаружения и исправления ошибок блоки *1* и *2* выполняются по логическим функциям в ДНФ, что при значительной протяженности кода требует больших аппаратурных затрат.

3.7. Геометрический синтез контролеспособных кодов позиционных систем счисления

До настоящего времени мы в основном занимались только анализом контролеспособности известных кодов. Здесь поставим более сложную задачу – синтез кодов с заданной контролеспособностью, решение которой опирается на принятую концепцию обнаружения и исправления ошибок как процедуру поиска аварийного перехода чисел натурального ряда из отведенных им структурой кода ячеек многомерного цифрового пространства в нештатные ячейки этого пространства. Эта процедура, как показано выше, позволяет обнаружить все виды и комбинации ошибок, определить наиболее вероятную штатную ячейку пространства, из которой была получена ошибочная кодовая комбинация информационных и контрольных сигналов.

Структура кода и все его разряды (информационные и контрольные), составляющие все универсальное цифровое пространство, определяют штатное $\lambda(0)$ и нештатное $\underline{\lambda}(0)$ многомерные подмножества (фигуры) этого пространства, где в рабочей части пространства расположены ячейки подмножеств с одиночными, двойными и т.д. безальтернативными ошибками, т.е. нештатное подмножество представляет собой, в свою очередь, множества подмножеств безальтернативных одиночных $\lambda(1)$, двойных $\lambda(2)$ и т.д. ошибок ($\underline{\lambda}(0) \supset \lambda(1)$, $\underline{\lambda}(0) \supset \lambda(2)$, ...).

В нештатном подмножестве могут располагаться также ячейки с альтернативными ошибками, которые могут быть только обнаружены, но не исправлены. Это подмножество цифрового пространства является нерабочим, и естественно стремление свести его к нулю. Такие коды с нулевым нерабочим цифровым пространством будем называть полностью контролеспособными для определенного вида ошибок.

Контролеспособность кода, как это следует из предыдущего, зависит только от конфигурации штатной фигуры $\lambda(0)$ и ее соотношения со всем многомерным пространством.

В этом параграфе в основном развивается эвристический подход к рассматриваемой проблеме, что связано со стремлением дать достаточно простые, универсальные и прозрачные методы синтеза, не исключая и более строгих подходов к решению поставленных задач. При этом ограничимся только двоичным представлением информационной части кода, рассмотрение которой дает в то же время полную ясность и для любого другого принципа кодирования.

Предрасположением к успешному решению задачи синтеза контролеспособных кодов геометрическим способом является геометрически прозрачное распределение любого типа ошибок кодовых комбинаций на цифровой прямой емкости 2^k. Эта прозрачность была продемонстрирована на примерах одиночных ошибок кодовых комбинаций (см. рис. 3.4).

Обращает на себя внимание то, что при симметричном расположении штатных ячеек относительно центра цифровой прямой совместного двоичного представления всех информационных и контрольных кодовых комбинаций также соблюдается строгая симметрия расположения их ошибок. Из этого следует правило распределения штатных ячеек для цифр натурального ряда на цифровой прямой, исключающее заполнение остальных ячеек пространства несколькими ошибками одинаковой кратности. Это правило свидетельствует не только о необходимости сохранения определенного кодового расстояния между штатными ячейками на цифровой прямой, но и об их обязательном симметричном расположении относительно центра цифровой прямой.

Например, если первая штатная ячейка располагается на позиции с номером 1, то симметричная ей ячейка должна быть расположена на позиции с номером 2^k, что для нашего примера – 32. Для исправления, например, одиночных ошибок необходимо сохранить определенное кодовое расстояние между штатными ячейками, и поэтому следующая штатная ячейка должна быть в соответствии с рис. 3.4 расположена на позиции с номером 8, а симметричная ей ячейка – на позиции с номером 25 и т.д.

3.7.1. Коды с обнаружением одиночных ошибок

В п. 3.1 мы рассмотрели принципы синтеза контролеспособного кода основания n= 2. Сейчас поставим задачу распространить этот подход на системы счисления большего основания, предполагая, что построение фигур $\lambda(0)$ больших оснований систем счисления должно производиться на основе сочетания аналогичных фигур именно основания n = 2.

На рис. 3.53 показано построение таких фигур с обнаружением одиночных ошибок соответственно для оснований n = 2, 4, 8, 16 и 32.

Каждая из фигур $\lambda_j(0)$ основания $n = 2^i (j=i+1)$ строится из аналогичных фигур $\lambda_{j-1}(0)$, $\underline{\lambda}_{j-1}(0)$ меньшего на две единицы основания по следующим логическим выражениям:

$$\lambda_j(0) = \left| \underline{x}_j \, x_j \right| \begin{vmatrix} \lambda_{j-1}(0) \\ \\ \underline{\lambda}_{j-1}(0) \end{vmatrix}, \qquad \underline{\lambda}_j(0) = \left| \underline{x}_j \, x_j \right| \begin{vmatrix} \underline{\lambda}_{j-1}(0) \\ \\ \lambda_{j-1}(0) \end{vmatrix}. \qquad (3.7.1)$$

Очевидно, что в данном случае для любого n требуется только один контрольный разряд, в качестве которого может быть принят любой из аргументов x_i (i= 1, ... , j). Тогда остальные аргументы, взятые в последовательности двоичного принципа кодирования либо иного принципа, определяют конкретное распределение цифр **0 – (n –1)** в ячейках цифрового пространства, отмеченных соответствующей заливкой.

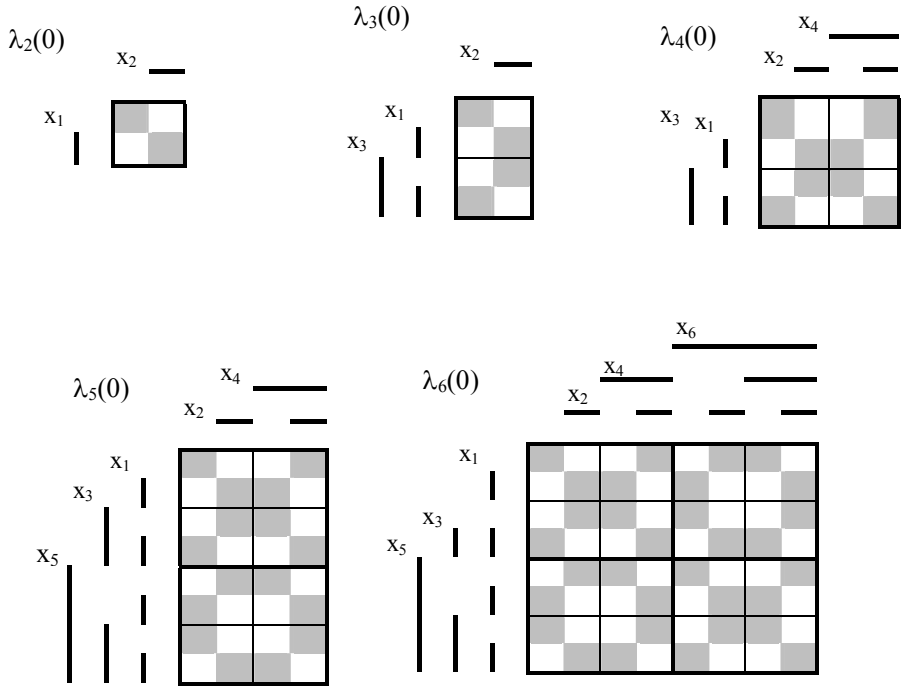


Рис. 3.53

Несмотря на тривиальность полученного решения, которое заключается во введении в код разряда проверки на четность (нечетность), эти фигуры обладают свойством симметрии, когда любая перестановка сигналов разрядов кода не меняет фигуру $\lambda_j(0)$, а логическая функция, связывающая между собой сигналы $\lambda_j(0)$ и x_j, при выборе в качестве функции любого аргумента этой зависимости остается постоянной, где новая функция и аргумент поменяются местами.

Например, для основания $n = 4$ можно записать:

$$\lambda_3(0) = \lambda_2(0)\underline{x}_3 \vee \underline{\lambda}_2(0)x_3 = x_1\, x_2\, \underline{x}_3 \vee \underline{x}_1\, \underline{x}_2\, \underline{x}_3 \vee x_1\, \underline{x}_2\, x_3 \vee \underline{x}_1\, x_2\, x_3\,;$$

$$\underline{\lambda}_3(0) = \underline{\lambda}_2(0)\underline{x}_3 \vee \lambda_2(0)x_3 = x_1\, \underline{x}_2\, \underline{x}_3 \vee \underline{x}_1\, x_2\, \underline{x}_3 \vee x_1\, x_2\, x_3 \vee \underline{x}_1\, \underline{x}_2\, x_3\,;$$

$$x_1 = \lambda_3(0)x_2\,\underline{x}_3 \vee \underline{\lambda}_3(0)\underline{x}_2\,\underline{x}_3 \vee \lambda_3(0)\underline{x}_2\,x_3 \vee \underline{\lambda}_3(0)x_2\,x_3\,;$$

$$\underline{x}_1 = \lambda_3(0)\underline{x}_2\,\underline{x}_3 \vee \underline{\lambda}_3(0)x_2\,\underline{x}_3 \vee \lambda_3(0)x_2\,x_3 \vee \underline{\lambda}_3(0)\underline{x}_2\,x_3.$$

Такие логические функции будем называть обратимыми. При переводе из прямого кода в обратный всех аргументов этой функции она остается неизменной при их четном числе, а при нечетном числе аргументов эта операция приводит к инвертированию функции, в чем можно убедиться, проводя соответствующие преобразования над фигурами рис. 3.53.

3.7.2. Коды с исправлением одиночных ошибок

Синтез кодов больших оснований систем счисления с обнаружением и исправлением одиночных ошибок можно, очевидно, производить на основании комбинации фигур, например, основания n = 2 трехмерного цифрового пространства, определяющих исправление одиночных ошибок этого основания, с последующим размещением их в ячейках других пространств, содержащих в качестве элементарных эти ячейки трехмерного пространства. Причем заполняемые элементарные ячейки этих пространств должны образовывать фигуры цифровых множеств, характеризующие обнаружение одиночных ошибок.

На рис. 3.54 приведены эти элементарные ячейки трехмерного цифрового пространства, которые совместно с размещенными в них фигурами обозначены соответственно μ_1 – μ_4. Здесь же представлены цифровые пространства большей мерности, которые с размещенными в них фигурами обозначены λ_2, λ_3, λ_4 , где светлой заливкой отмечены элементарные ячейки для помещения фигур μ_1 – μ_4.

Фигуры μ_1 – μ_4 охватывают все возможные варианты размещения цифр **0**, **1** основания n = 2 в трехмерном цифровом пространстве, где исправляются одиночные ошибки. Построение этих фигур определяется последовательным размещением этих цифр на цифровой прямой емкостью 2^k с сохранением симметрии размещения относительно центра этой прямой и неизменностью кодового расстояния между ними, обеспечивающего возможность исправления одиночных ошибок.

Для основания n = 4 необходимо в соответствующие ячейки пространства λ_2 размещать фигуры цифровых множеств μ_1 – μ_4, где $a_i = a_1$.

Причем для обеспечения равенства нулю информационных и контрольных сигналов синтезируемого кода для эквивалентной информационной цифры **0** будем всегда при любых основаниях систем счисления в начале координат размещать фигуру μ_1.

Оставшаяся при этом незаполненная ячейка пространства λ_2 может в принципе включать в себя любую из фигур μ_1 – μ_4 . Рассмотрим все варианты размещения, которые показаны соответственно на рис. 3.55–3.58.

На этих рисунках в ячейках пространства λ_2 в соответствии с двоичным принципом кодирования информационной части кода выделены штатные информационные цифры **0**–**3** этого основания и приведены их одиночные безальтернативные ошибки. Здесь также на основании представленного размещения штатных цифр приведено одномерное графическое соответствие информационным сигналам a_1, a_2 их эквивалентных цифр, а также контрольных сигналов x_1, x_2, x_3.

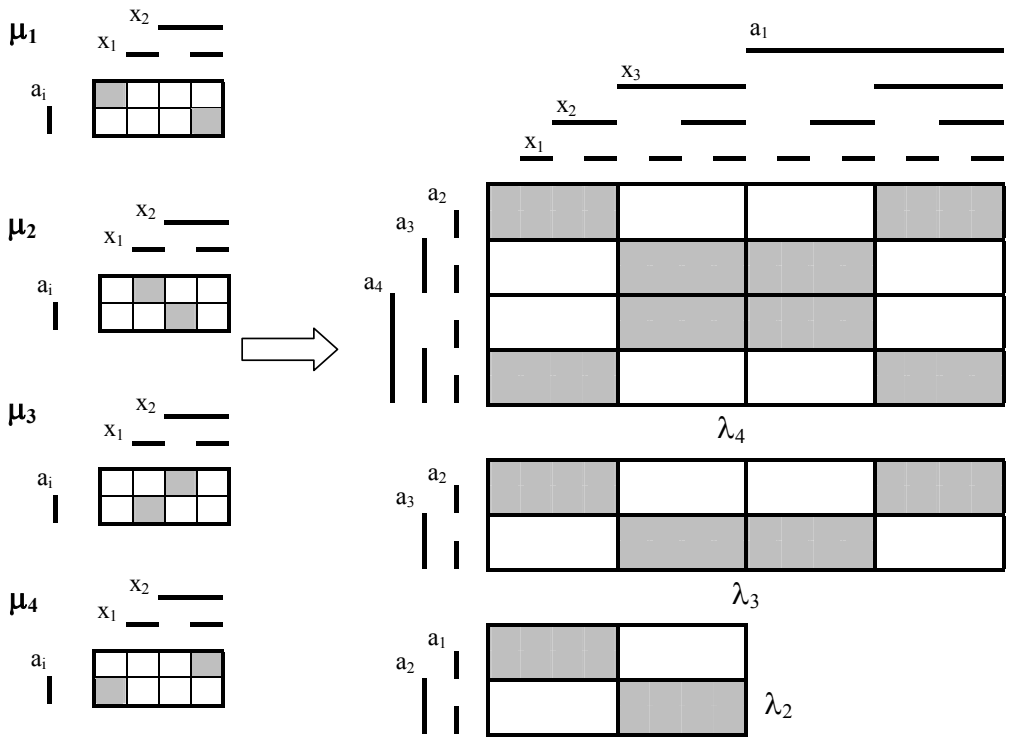


Рис. 3.54

Эти соотношения позволяют разместить в ячейках многомерного пространства информационных сигналов кода, которые сохраняют стандартную нумерацию его ячеек от **0** до **3**, эквивалентные цифры 0–7 контрольных разрядов синтезированного кода. Эти эквивалентные контрольные цифры также соответствуют их двоичному принципу кодирования сигналами x_1, x_2, x_3.

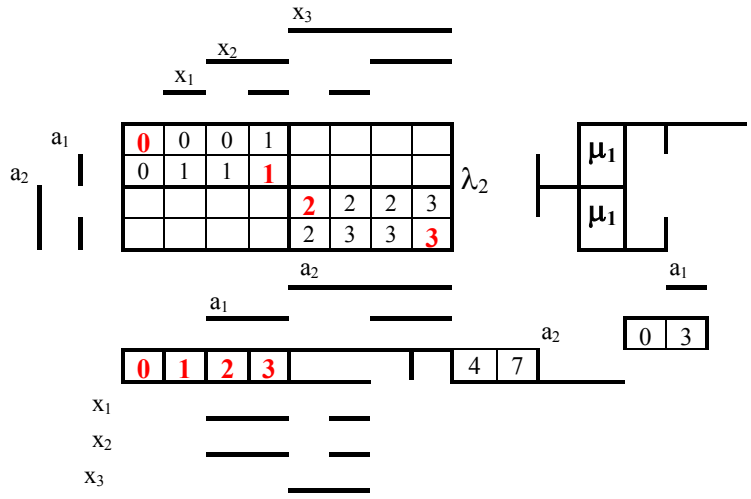


Рис. 3.55

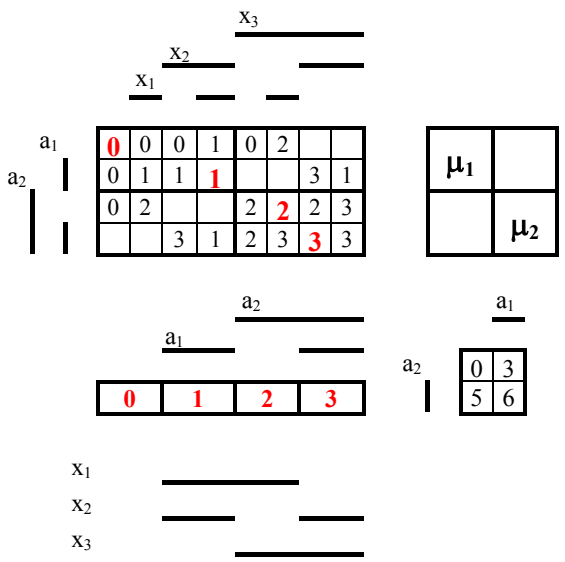


Рис. 3.56

Из этих графических соотношений непосредственно следует определение нерабочей области многомерного цифрового пространства синтезируемого кода для этих вариантов размещения:

$$\underline{\lambda}(\mu_1/\mu_4) = x_1\underline{x}_2\underline{x}_3a_2 \vee \underline{x}_1x_2\underline{x}_3a_2 \vee x_1\underline{x}_2x_3\underline{a}_2 \vee \underline{x}_1x_2x_3\underline{a}_2\,;$$

$$\underline{\lambda}(\mu_1/\mu_1) = a_2\underline{x}_3 \vee \underline{a}_2x_3\,;$$

$$\underline{\lambda}(\mu_1/\mu_2) = x_2x_3\underline{a}_1\underline{a}_2 \vee \underline{x}_2x_3a_1\underline{a}_2 \vee x_2\underline{x}_3\underline{a}_1a_2 \vee \underline{x}_2\underline{x}_3a_1a_2\,;$$

$$\underline{\lambda}(\mu_1/\mu_3) = x_1x_3\underline{a}_1\underline{a}_2 \vee \underline{x}_1x_3a_1\underline{a}_2 \vee x_1\underline{x}_3\underline{a}_1a_2 \vee \underline{x}_1\underline{x}_3a_1a_2\,. \qquad (3.7.2)$$

Из (3.7.2) следует, что определение нерабочей области цифрового пространства первого варианта размещения наиболее просто, но использование многомерного пространства здесь составляет только 50%. Остальные варианты используют более рационально ячейки многомерного пространства (75%), но в отличие от полного совпадения в первом варианте сигналов контрольных и информационных разрядов здесь только один контрольный разряд совпадает с информационным ($x_3 = a_2$).

Очевидно, что при большем совпадении контрольных и информационных сигналов кода выполнение каких-либо арифметических или логических операций в таком синтезированном коде упрощается. Однако это не означает, что данный код лучше других по обеспечению контролеспособности в конкретной арифметической или логической операции, и для ответа на этот вопрос необходимо провести отдельные исследования.

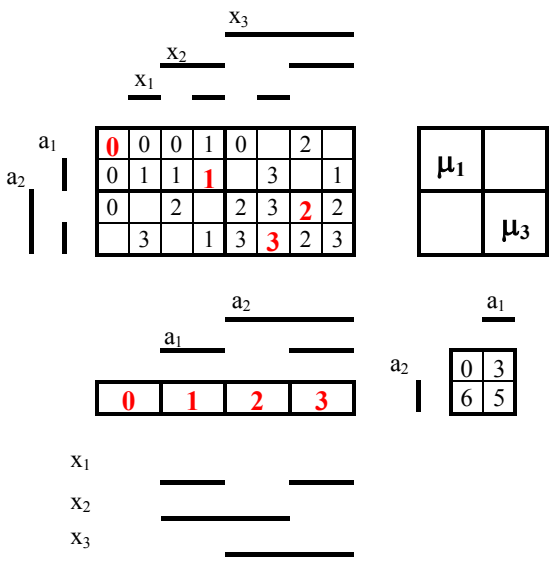


Рис. 3.57

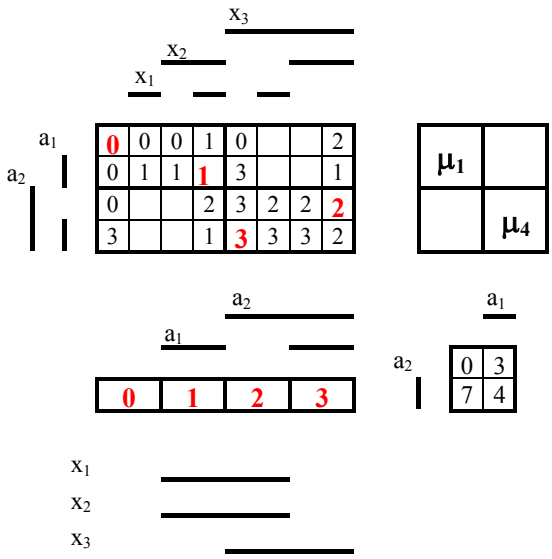


Рис. 3.58

Из симметрии многомерного цифрового пространства очевидно, что число эквивалентных синтезированных контролеспособных кодов, при выбранном принципе кодирования его информационной части, определяется только мерностью контрольной части кода. Ограничиваясь при этом только такими типами кодов, число кодов одного класса будет определяться количеством перестановок контрольных разрядов. Из этого следует, что второй и третий вариант размещения (рис. 3.56 и 3.57) образуют два эквивалентных кода.

Обратимся к синтезу кода основания n = 8, для чего в соответствующих рис. 3.54 ячейках пространства λ_3 разместим фигуры $\mu_1 - \mu_4$, где $a_i = a_2$.

Этот вариант синтезированного кода, в полной аналогии с рассмотренным выше для основания n = 4, представлен на рис. 3.59.

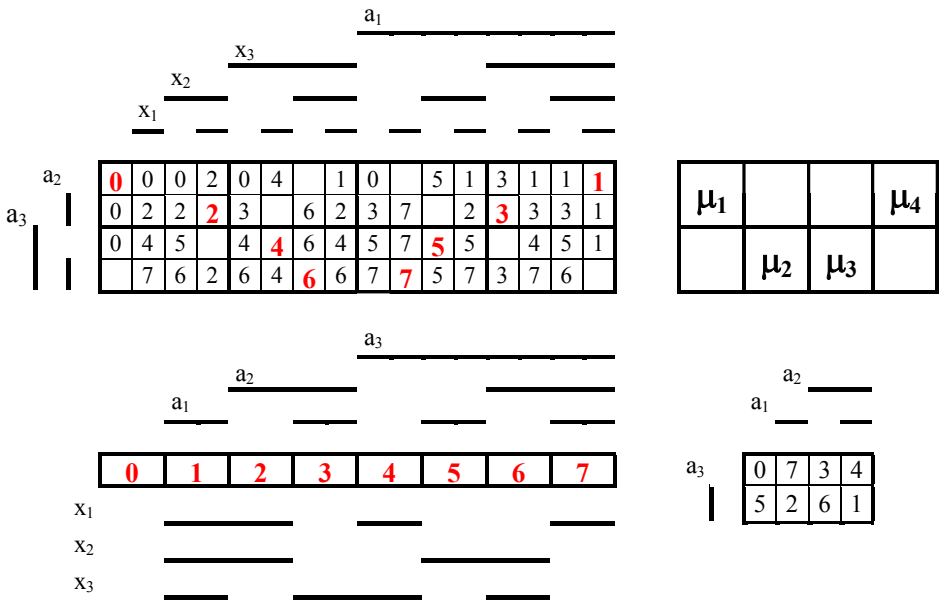


Рис. 3.59

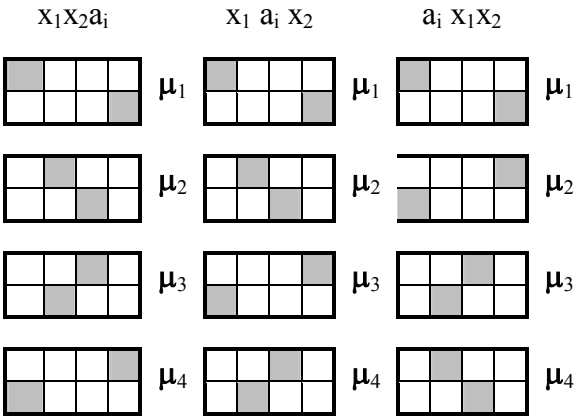


Рис. 3.60

Он является представителем класса синтезированного контролеспособного кода, где остальные получаются перестановкой сигналов контрольных разрядов.

Таким образом, в каждом классе будет содержаться шесть кодов, которые связаны между собой следующей перестановкой эквивалентных цифр контрольных разрядов:

$$x_1\ x_2\ x_3 \rightarrow 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7;$$
$$x_1\ x_3\ x_2 \rightarrow 0\ 1\ 4\ 5\ 2\ 3\ 6\ 7;$$
$$x_2\ x_1\ x_3 \rightarrow 0\ 2\ 1\ 3\ 4\ 6\ 5\ 7;$$
$$x_2\ x_3\ x_1 \rightarrow 0\ 2\ 4\ 6\ 1\ 3\ 5\ 7;$$
$$x_3\ x_2\ x_1 \rightarrow 0\ 4\ 2\ 6\ 1\ 5\ 3\ 7;$$
$$x_3\ x_1\ x_2 \rightarrow 0\ 4\ 1\ 5\ 2\ 6\ 3\ 7. \qquad (3.7.3$$

Очевидно, что остальные представители классов синтезированного кода могут быть определены из рассмотрения перестановок сигналов x_1 x_2 a_i , где следует взять только перестановки, отличающиеся расположением сигнала a_i. Таким образом, для исправления одиночных ошибок может быть выделено три класса контролеспособных кодов, которые характеризуются различным размещением фигур $\mu_1 - \mu_4$ в соответствующих ячейках пространства. Эти подстановки должны выполняться по рис. 3.60.

Тогда типопредставители второго и третьего класса синтезированных кодов могут быть показаны соответствующим образом на рис. 3.61 и 3.62.

Связь между контрольными и информационными сигналами для трех классов этих кодов на основании рис. 3.59, 3.61 и 3.62 определяется соответственно следующими логическими зависимостями:

$$x_1 = \underline{a}_1\underline{a}_2a_3 \vee a_1\underline{a}_2\underline{a}_3 \vee \underline{a}_1a_2\underline{a}_3 \vee a_1a_2a_3\,,\ x_2 = \underline{a}_1a_2 \vee a_1\underline{a}_2\,,\ x_3 = \underline{a}_1a_3 \vee a_1\underline{a}_3; \qquad (3.7.4)$$

$$x_1 = \underline{a}_2a_3 \vee a_2\underline{a}_3\,,\ x_2 = \underline{a}_1a_2 \vee a_1\underline{a}_2\,,\ x_3 = \underline{a}_1a_3 \vee a_1\underline{a}_3; \qquad (3.7.5)$$

$$x_1 = \underline{a}_1\underline{a}_2a_3 \vee a_1\underline{a}_2\underline{a}_3 \vee \underline{a}_1a_2\underline{a}_3 \vee a_1a_2a_3\,,\ x_2 = \underline{a}_2a_3 \vee a_2\underline{a}_3\,,\ x_3 = \underline{a}_1a_3 \vee a_1\underline{a}_3. \qquad (3.7.6)$$

Все приведенные здесь коды основания n =8 используют 87,5% многомерного цифрового пространства.

Теперь перейдем к синтезу кода основания n =16. Для этого необходимо в соответствующих восьми ячейках пространства λ_4 разместить четыре фигуры $\mu_1 - \mu_4$. Выполненный выше синтез кода меньшего основания и закон симметрии расположения штатных цифр на цифровой прямой позволяют выполнить эти размещения сразу. Эти варианты размещения, определяющие три класса кодов, в полной аналогии с рассмотренными выше синтезированными кодами представлены соответственно на рис. 3.63–3.65.

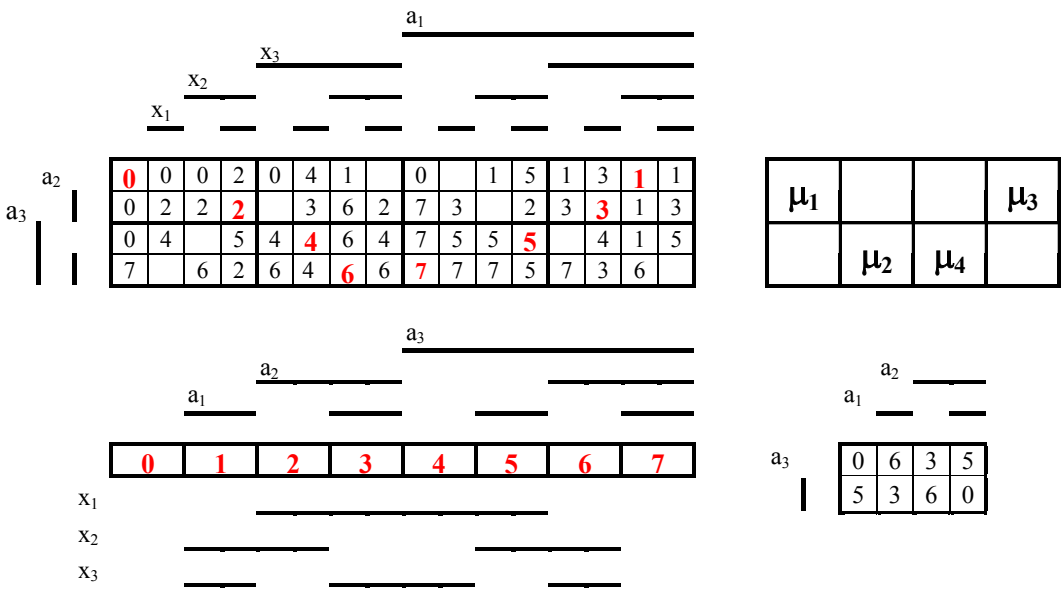


Рис. 3.61

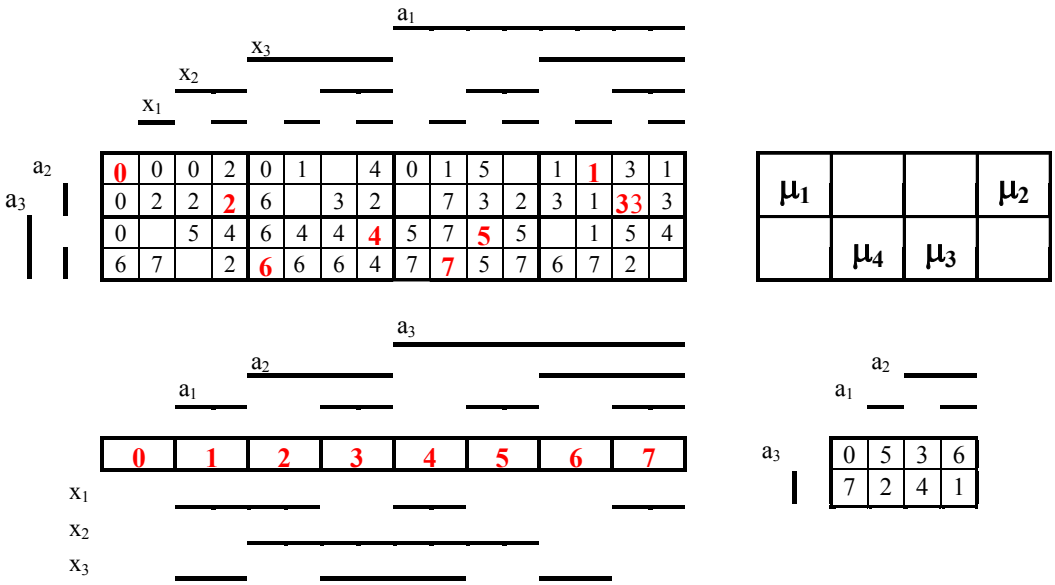


Рис. 3.62

Приведенные здесь синтезированные коды основания n = 16 используют на 100% все многомерное цифровое пространство относительно одиночных ошибок.

Связь между контрольным разрядом и определенной группой информационных сигналов разрядов для всех кодов всех классов одна и та же. Эта связь полностью определяется логическим выражением (3.7.1) для $\underline{\lambda}_3(0)$.

Все восемнадцать синтезированных кодов основания n = 16, представленных распределением цифр контрольных разрядов в многомерном пространстве информационной части кода, приведены на рис. 3.66.

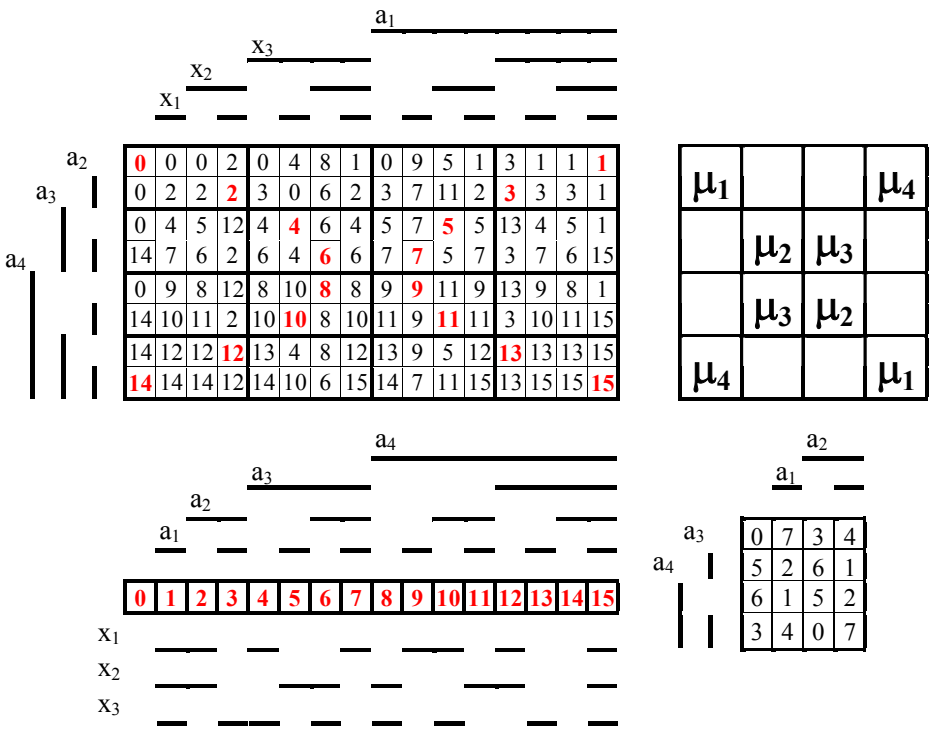


Рис. 3.63

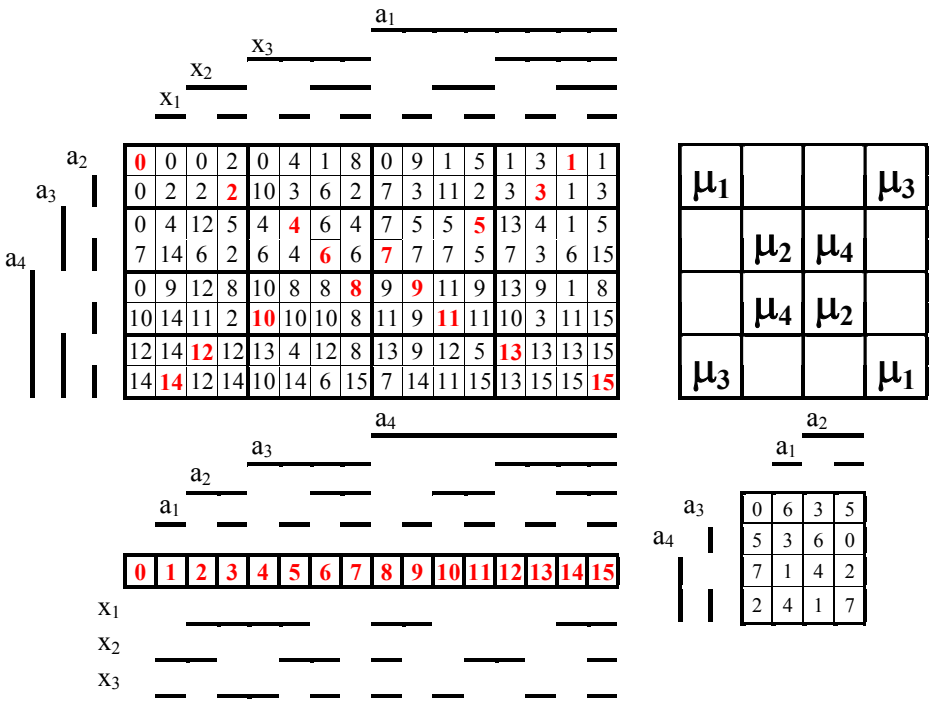


Рис. 3.64

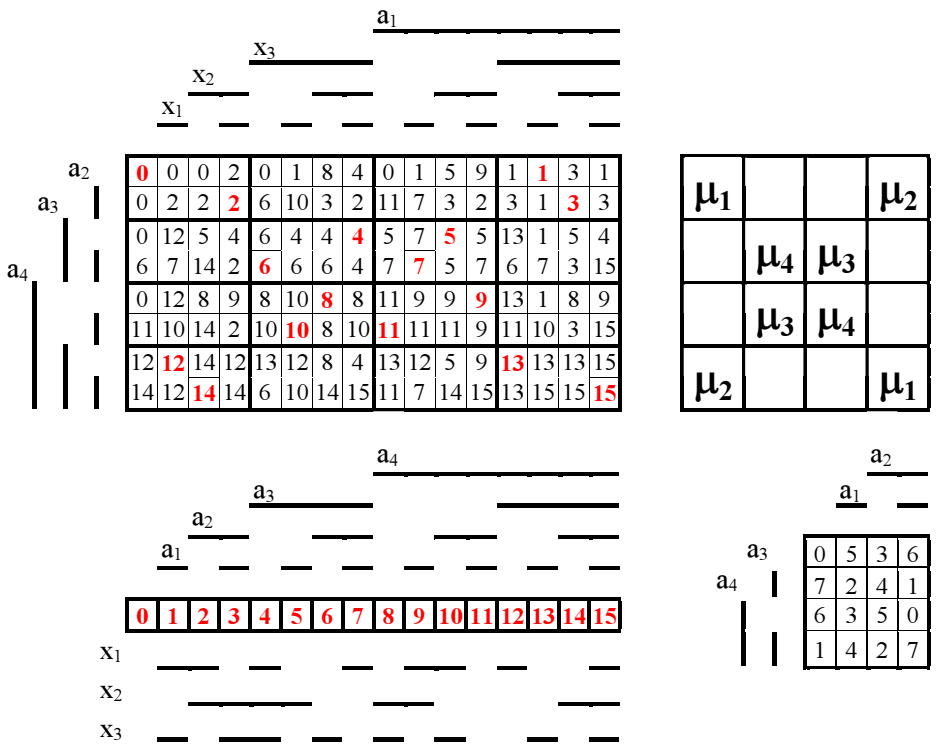


Рис. 3.65

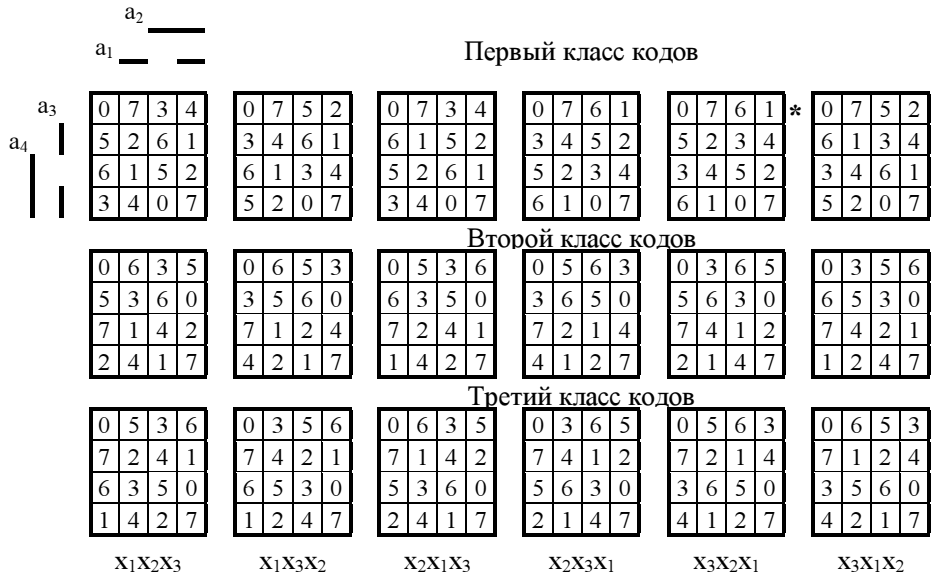


Рис. 3.66

Сравнение распределения цифр 0–7 контрольных разрядов в многомерном цифровом пространстве информационной части кода с аналогичным распределением кодов оснований n = 4, 8 показывает, что полностью контролеспособный код основания n = 16 определяет все соотношения между информационной и контрольной частями в кодах меньших оснований. Поэтому в дальнейшем будем рассматривать только такие основания систем счисления, которые позволяют синтезировать полностью контролеспособные коды.

3.7.3. Коды больших оснований систем счисления с исправлением одиночных ошибок

Синтез кодов больших оснований систем счисления можно выполнить, опираясь на представленную выше систему счисления основания n = 16. При этом процедура операции синтеза будет так же, как рассмотренная выше, заключаться в размещении в многомерном цифровом пространстве информационных и одного контрольного разряда фигур основания n = 16 семимерного цифрового пространства (см. рис. 3.53), где реализовано исправление одиночных ошибок.

Для сокращения графического материала ограничим рассмотрение только представлением синтеза кода одного класса.

На рис. 3.67 приведены эти восемь ячеек семимерного пространства, которые вместе с размещенными в них фигурами обозначены σ_1 – σ_8. Эти фигуры охватывают все варианты размещения цифр **0–15** основания n = 16 в семимерном (k = 3, i = 4) цифровом пространстве, где исправляются все одиночные ошибки. Построение этих фигур, выполненное аналогично основанию n = 2, определяется последовательным размещением этих цифр на цифровой прямой емкостью 2^7 с сохранением симметрии размещения относительно центра этой прямой и неизменности кодового расстояния между ними, обеспечивающим исправление одиночных ошибок.

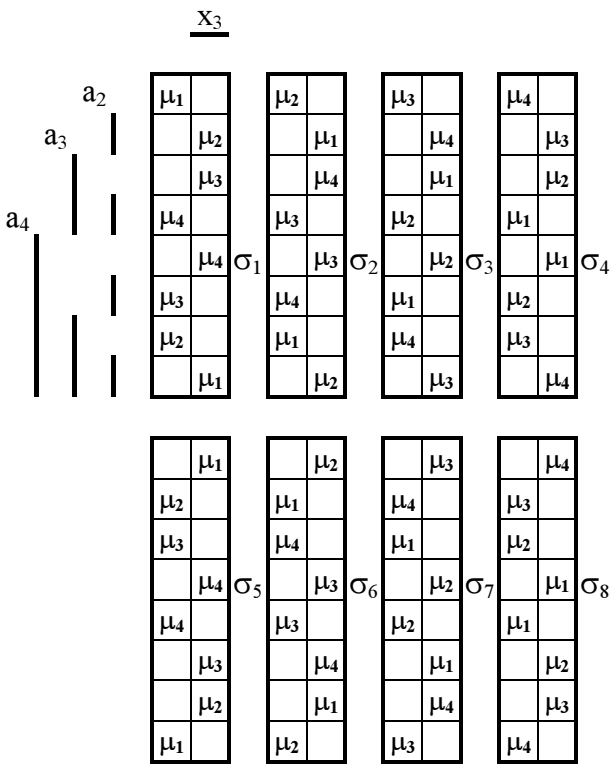


Рис. 3.67

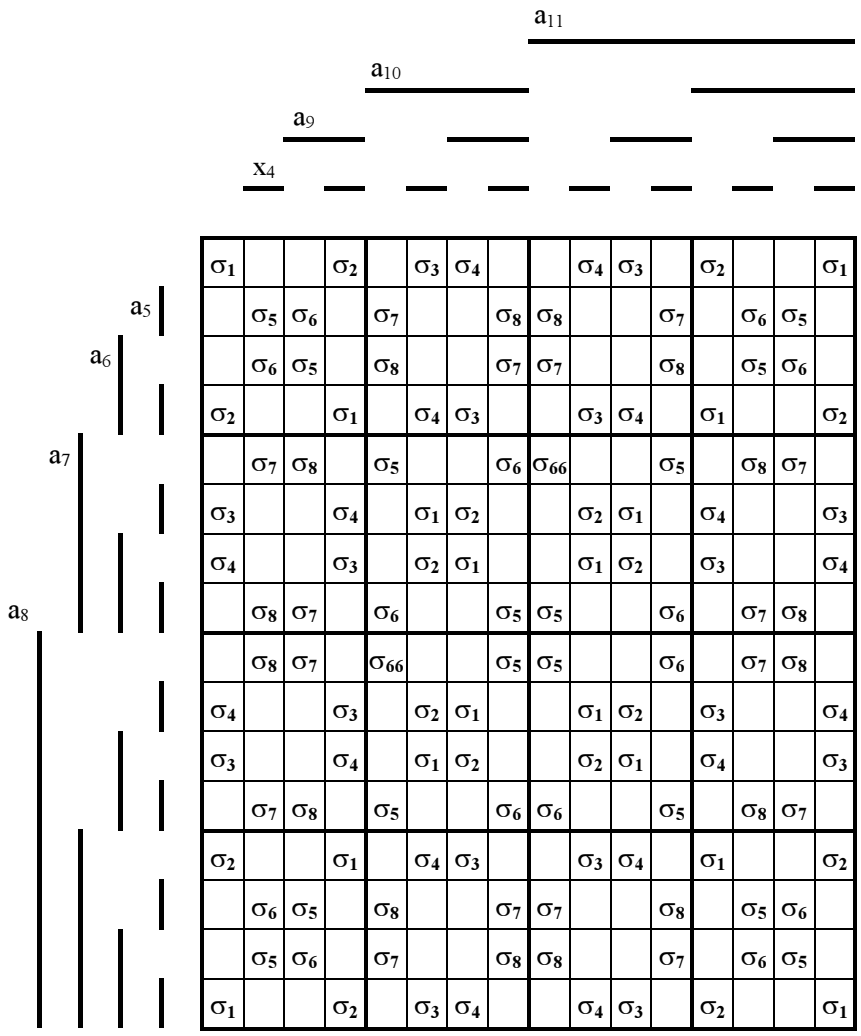


Рис. 3.68

Опустив достаточно прозрачные из изложенного выше операции подстановки фигур $\boldsymbol{\sigma_1}$ – $\boldsymbol{\sigma_8}$ в ячейки пространства большей мерности (см. рис. 3.53), на рис. 3.68 приведен один из многочисленных вариантов такого размещения в пространстве $\lambda_8(0)$, что определяет полностью контролеспособный код для основания $n = 2^{11}$. В соответствии с этой процедурой не представит какого-либо труда разместить в ячейках многомерного цифрового пространства информационной части кода эквивалентные контрольным сигналам разрядов $\mathbf{x}_1$ – $\mathbf{x}_4$ цифры 0–15.

Для компактного графического представления одного из вариантов синтеза на рис. 3.69,*а* показаны большие ячейки информационного пространства, которые обозначены соответственно $\mathbf{s}_1$ – $\mathbf{s}_4$ и $\boldsymbol{s_1}$ – $\boldsymbol{s_4}$, а на рис. 3.69, *б* приведена ячейка $\mathbf{s}_1$ с заполненными элементарными ячейками информационного пространства эквивалентными цифрами контрольной части синтезированного кода.

Заполнение остальных больших ячеек информационного цифрового пространства, которые имеют такие же координаты, как $\mathbf{s_1}$, подчиняется следующим взаимным перестановкам эквивалентных цифр контрольной части кода:

$$\left|\begin{array}{l} \mathbf{s_1} \Leftrightarrow \mathbf{s_2} \\ \\ \mathbf{s_3} \Leftrightarrow \mathbf{s_4} \end{array}\right| \equiv \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 8 & 9 & 10 & 11 \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 7 & 6 & 5 & 4 & 15 & 14 & 13 & 12, \end{array}$$

$$\left|\begin{array}{l} \mathbf{s_1} \Leftrightarrow \mathbf{s_4} \\ \\ \mathbf{s_2} \Leftrightarrow \mathbf{s_3} \end{array}\right| \equiv \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 14 & 15 & 12 & 13 & 10 & 11 & 8 & 9, \end{array}$$

$$\left|\begin{array}{l} \mathbf{s_1} \Leftrightarrow \mathbf{s_1} \\ \mathbf{s_2} \Leftrightarrow \mathbf{s_2} \\ \\ \mathbf{s_3} \Leftrightarrow \mathbf{s_3} \\ \mathbf{s_4} \Leftrightarrow \mathbf{s_4} \end{array}\right| \equiv \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8. \end{array}$$

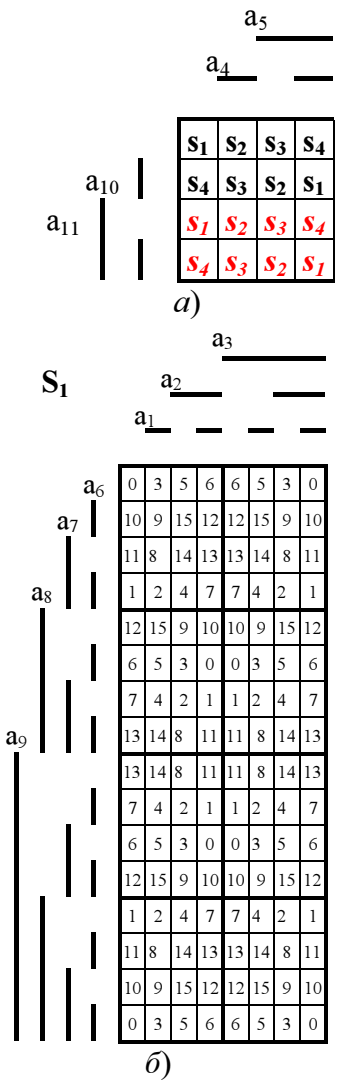


Рис. 3.69

На основании зависимостей рис. 3.69, выполняя определенные преобразования, аналогичные тем, которые проводились выше, можно получить графические образы логических функций, связывающие между собой контрольные и информационные сигналы разрядов.

На рис. 3.70 приведена такая зависимость между сигналом контрольного разряда x_1 и информационными разрядами $x_1 = \mathbf{F}(a_1, a_2, a_4, a_5, a_7, a_9, a_{11})$, которая полностью определяется выражением $\underline{\lambda}_7(0)$.

Аналогичные зависимости представляются для сигналов других контрольных разрядов, которые отличаются иными группами сигналов информационных разрядов, а именно:

$$x_2 = \mathbf{F}(a_1, a_2, a_3, a_4, a_5);$$

$$x_3 = \mathbf{F}(a_2, a_3, a_4, a_8, a_9);$$

$$x_4 = \mathbf{F}(a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}).$$

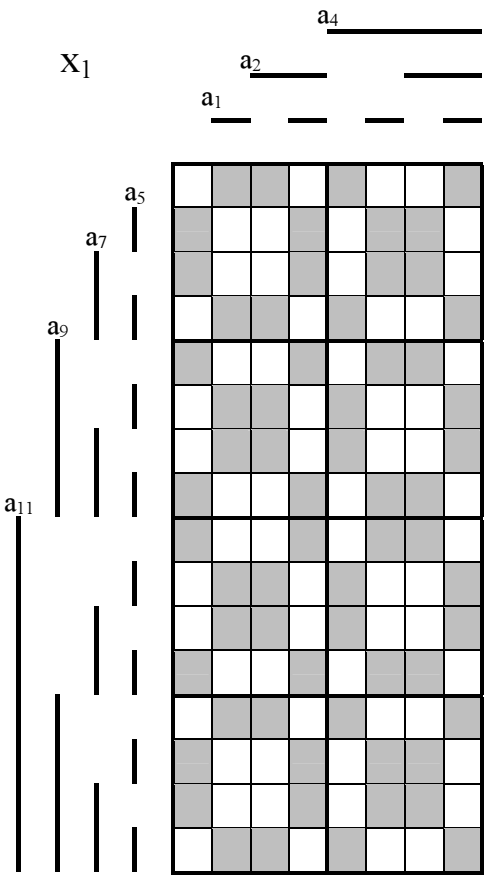


Рис. 3.70

В представленных нами примерах синтеза полностью контролеспособных кодов относительно одиночных ошибок, тем более охватывающих все их классы, должны содержаться все коды, синтезированные каким-либо иным способом. Например, замечательная процедура синтеза контролеспособных кодов Хемминга для исправления одиночных ошибок представлена на рис. 3.66 в первом классе и отмечена звездочкой, а только что синтезированный код основания $n = 2^{11}$ также является кодом Хемминга.

Связь между контрольными k и информационными i сигналами разрядов полностью контролеспособных кодов определяется зависимостью $i = [2^k - (k+1)]$, что для оснований $n = 2^1, 2^4, 2^{11}, 2^{26}, ...$ позволяет исправлять все одиночные ошибки.

Отмеченное здесь не означает, что для этих оснований и любых других нет возможности обнаружения и исправления большего количества комбинаций ошибок, чем рассмотрено выше. Увеличение числа контрольных разрядов сверх представленного приведет к тому, что при геометрическом способе синтеза кодов необходимо будет в ячейках пространства рис. 3.53 размещать, например, фигуры основания n = 2, позволяющие исправлять одиночные, двойные и т.д. ошибки, частично рассмотренные в п. 3.1.

Именно эти достаточно сложные задачи будут детально рассмотрены в последующих главах книги, а представленный в этом параграфе синтез совершенных кодов и определение числа этих кодов носят предварительный, вводный характер и будут уточнены в дальнейшем изложении.

3.8. Решение задачи повышения надежности логических и цифровых устройств в режиме реального времени

Известно, что надежность логических и цифровых устройств можно повысить, не используя резервирования их элементов. Для достижения этой цели необходимо применять высоконадежные элементы и схемы с большим запасом надежности, где уделяется повышенное внимание технологии их изготовления и сборки. Однако этот правильный путь не может обеспечить надежные вычислительные системы многолетней длительной эксплуатации без человеческого сопровождения и хранения, используемые для проведения, например, исследований космического пространства и управления ответственными бортовыми системами реального масштаба времени [13].

С целью повышения надежности таких систем нередко используется их резервирование со схемой голосования. Такой путь повышения надежности был предложен Дж. фон Нейманом [14], который разработал и проанализировал схему тройного резервирования элементов с мажоритарной функцией голосования. Геометрический вариант подобного подхода к проблеме исправления ошибок был представлен нами в предыдущих разделах. Этот подход заключается в исправлении всех ошибок конечной двоичной системы счисления, например, всех одиночных ошибок или одновременно всех одиночных и двойных ошибок; одновременно всех одиночных, двойных и тройных ошибок и т.д. Под термином «всех» понимается, что конкретная ошибка может существовать все время, и она должна при этом непременно исправляться. Следовательно, исправление ошибок и резервирование, например, элементов цифровой системы здесь будут совпадать при условии, что исправление ошибок каждого разряда систематического кода осуществляется отдельной схемой, не имеющей общих элементов со схемами исправления ошибок других разрядов.

Повышение надежности с мажоритарной функцией голосования Дж. фон Неймана, где равные одновременные сигналы F(a) одинаковых элементов *1* (рис. 3.71) объединяются мажоритарной схемой голосования *2*, имеет много общего с задачей исправления всех одиночных ошибок в системе счисления основания два (см. рис. 3.3).

В самом деле, исправление всех одиночных ошибок в каждом из разрядов любой системы счисления является резервированием канала связи, что равноценно резервированию выходных сигналов F(a) рис. 3.71.

Здесь следует учитывать, что при несовершенстве мажоритарной схемы вероятность безотказной работы резервированной схемы определяется [15] зависимостью

$$P_p = P^2(3 - 2P)P_m,$$

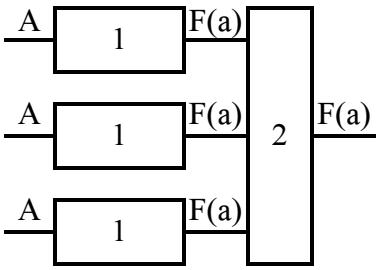


Рис. 3.71

где

P – вероятность безотказной работы отдельной нерезервированной схемы;

P_m – вероятность безотказной работы отдельной мажоритарной схемы.

Из анализа этого выражения следует, что в резервированной таким образом схеме вероятность безотказной работы выше, чем в нерезервированной, но до определенного момента времени, после которого она становится меньше нерезервированной схемы. Поэтому средняя наработка на отказ становится недостаточной для работы в течение требуемого интервала времени и для получения необходимого значения средней наработки на отказ вся схема должна будет строиться с многократным мажоритарным резервированием (см. рис. 3.5, рис. 3.6) либо необходимо будет выполнить резервирование мажоритарного элемента.

Построение логических и цифровых устройств, в систематических кодах с исправлением всех заданных заранее ошибок, вполне реально, но требует значительных аппаратурных затрат. При этом наиболее целесообразно использовать систематические коды с большей разрядностью его информационной части, что позволяет уменьшать аппаратурные затраты.

Правильность этого заключения легко показать на примере выполнения комбинационных схем бинарных логических функций при исправлении всех одиночных ошибок двоичной системы счисления, где систематический код первого операнда будет содержать один информационный разряд – a_1 и два контрольных разряда – x_1, x_2. Для второго операнда это будут соответственно разряды b_1, z_1, z_2.

Геометрические образы бинарных логических функций y_1 – y_{16} с исправлением одиночных ошибок и аналогичные им функции без исправления ошибок приведены на рис. 3.72 в многомерном цифровом пространстве координат a_1, x_1, x_2; b_1, z_1, z_2. Звездочками в ячейках этого пространства представляются геометрические образы выходных функций, где цветом выделены безошибочные значения входных сигналов.

Все мысленные повороты относительно осей симметрии *1–5* этого пространства, сопровождаемые соответствующим изменением входных сигналов координат, переводят геометрические образы этих функций в новые положения так же, как это имеет место в двоичном коде без исправления ошибок. Поэтому в каждой из групп одинаковых геометрических образов (y_7, y_{10}; y_2, y_3, y_9, y_5; y_{15}, y_{14}, y_{12}, y_8; y_4, y_{11}, y_{13}, y_6) можно представить только одну, а остальные будут определяться соответствующими поворотами относительно осей симметрии этого цифрового пространства.

Пусть это будут функции y_7, y_2, y_{15}, y_4, а также функция универсального цифрового пространства – y_{16}, геометрический образ которой при всех мысленных поворотах относительно осей симметрии цифрового пространства так же, как и прежде, не меняет своего положения в координатах этого пространства.

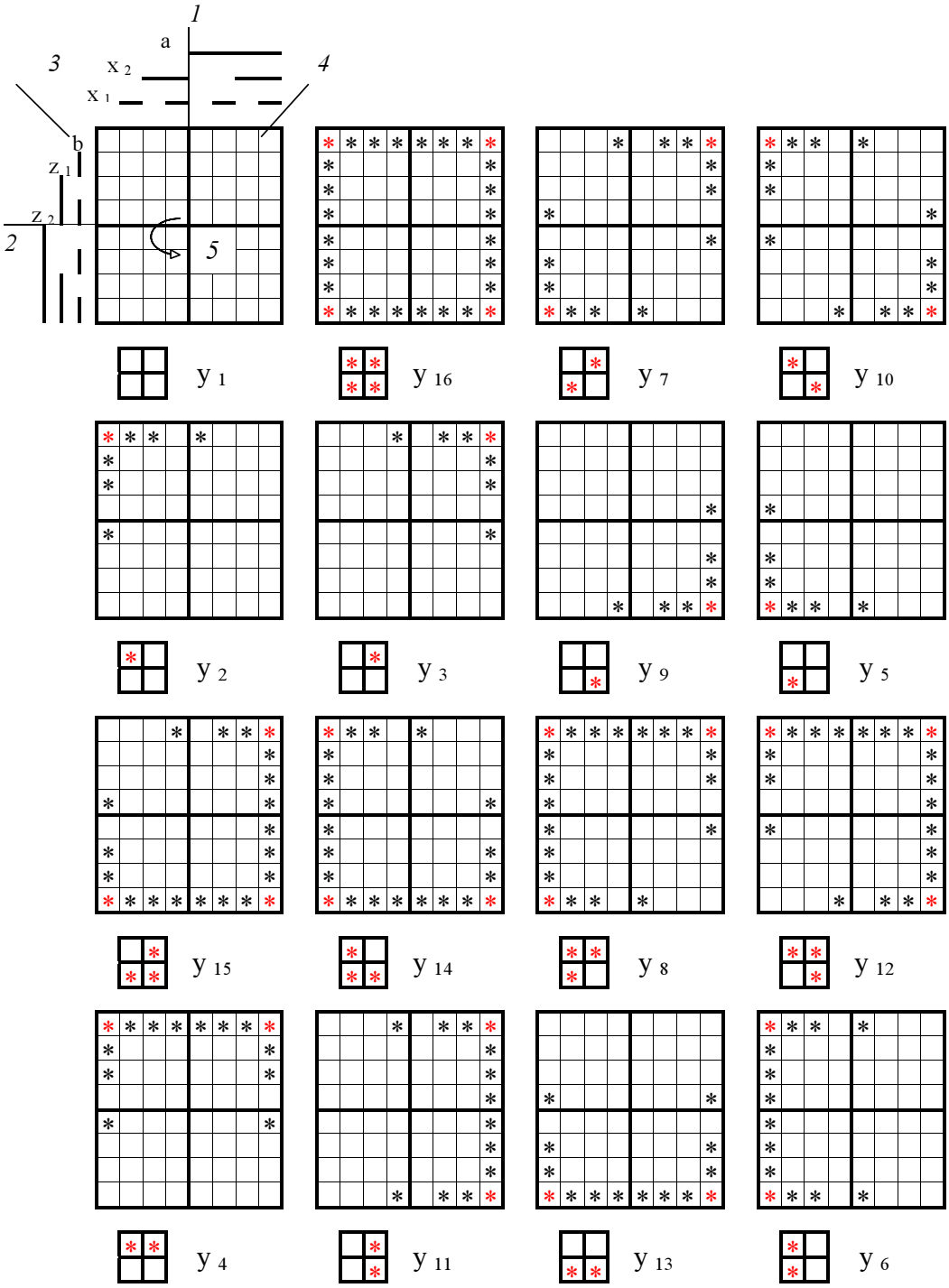

1
a
3
x 2
x 1
4
b
z 1
z 2
2
5
y 1
y 16
y 7
y 10
y 2
y 3
y 9
y 5
y 15
y 14
y 8
y 12
y 4
y 11
y 13
y 6

Рис. 3.72

$$y_7 = x_1 x_2 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee x_1 a_1 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee x_2 a_1 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee$$
$$\vee \underline{z}_1 \underline{z}_2 a_1 x_1 x_2 \vee \underline{z}_1 \underline{b}_2 a_1 x_1 x_2 \vee \underline{z}_2 \underline{b}_2 a_1 x_1 x_2 \vee$$
$$\vee z_1 z_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee z_1 b_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee \underline{z}_2 b_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee$$
$$\vee \underline{x}_1 \underline{x}_2 b_1 z_1 z_2 \vee \underline{x}_1 \underline{a}_1 b_1 z_1 z_2 \vee \underline{x}_2 \underline{a}_1 b_1 z_1 z_2;$$
$$y_2 = \underline{x}_1 \underline{x}_2 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee \underline{x}_2 \underline{a}_1 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee \underline{x}_1 \underline{a}_1 \underline{b_1} \underline{z}_1 \underline{z}_2 \vee$$
$$\vee \underline{z}_1 \underline{z}_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee \underline{z}_2 \underline{b}_1 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee \underline{z}_1 \underline{b}_1 \underline{a_1} \underline{x}_1 \underline{x}_2;$$
$$y_{15} = x_1 x_2 a_1 \vee z_1 z_2 b_1;$$
$$y_4 = \underline{z_1} \underline{z}_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee \underline{z}_2 \underline{b}_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee \underline{z}_1 \underline{b}_2 \underline{a_1} \underline{x}_1 \underline{x}_2 \vee$$
$$\vee \underline{z}_1 \underline{z}_2 a_1 x_1 x_2 \vee \underline{z}_2 \underline{b}_2 a_1 x_1 x_2 \vee \underline{z}_1 \underline{b}_2 a_1 x_1 x_2 \vee$$
$$\vee \underline{z}_1 \underline{z}_2 \underline{b_1};$$
$$Y_{16} = \underline{x}_1 \underline{x}_2 \underline{a_1} \vee x_1 x_2 a_1 \vee \underline{z}_1 \underline{z}_2 \underline{b_1} \vee z_1 z_2 b_1.$$

Нерациональное использование многомерного цифрового пространства, где универсальное пространство использует только его «оболочку», и сложность при этом логических выражений даже для простых бинарных функций свидетельствуют о неперспективности использования этого кода для синтеза логических и цифровых устройств еще больших оснований систем счисления.

Следовательно, необходимо использовать систематические коды с большой разрядностью информационной части, где общее число разрядов систематического кода определяется выражением

$$(i + k) = (2^{k-1}),$$

где

i – число информационных разрядов кода;

k – число контрольных разрядов кода.

Уже для k = 9 число информационных разрядов настолько большое, что емкость такого многомерного цифрового пространства, которое является контролеспособным относительно одиночных ошибок, превосходит единицу измерения Гугол (10^{100}), название которой предложил американский математик Кастнер. Хотя натуральный ряд чисел и бесконечен, Гугол – это граница исчисляемого мира. Обратимся к просторам космоса и выразим расстояние между звездами в ангстремах, который равен одной десятимиллионной части миллиметра. Тогда световой год, который определяет расстояние, проходимое солнечным лучом в течение года, представляется только числом 10^{26} Å. Даже расстояние до самых удаленных галактик не превышает 6×10^{27} Å. Возраст Вселенной, выраженный в световых годах, составляет менее 10^{40} Å. Примерно столько же составляет вся энергия, излучаемая всеми звездами Вселенной, выраженная в микроваттах. Поэтому в практической деятельности мы всегда встречаемся только с конечными логическими и цифровыми устройствами, что позволяет говорить о конечности исследуемого нами цифрового пространства, где содержимое каждой его ячейки может контролироваться, т.е. исправляться при соответствующем соотношении информационных и контрольных частей систематического кода.

Перед тем как приступить к применению синтезированных нами геометрическим способом систематических кодов, предназначенных для исправления, например, всех одиночных ошибок, в решении задач резервирования логических и цифровых устройств без памяти (комбинационные схемы) и с памятью (автоматы), необходимо еще раз уточнить связь расширенного натурального ряда чисел с логическими и цифровыми функциями и схемами, их реализующими.

Входным аргументом любой логической функции и сигналом схемы, ее реализующей, является натуральное число, которое определяет номер ячейки цифрового пространства, где для выходной двухзначной функции записываются логический нуль или единица, которые также соответствуют натуральному числу. Входное натуральное число поступает на входные шины схемы в основном двоичном эквиваленте в одном из значений числа от 0 до $(2^n - 1)$.

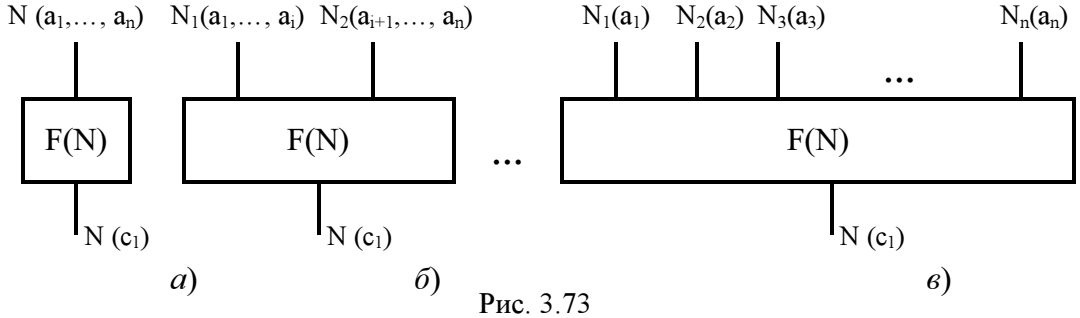


Рис. 3.73

При этом натуральное входное число может представляться, как одноразрядным (рис. 3.73, *а*) в одномерном цифровом пространстве, двухразрядным (см. рис. 3.73, *б*) в двухмерном пространстве и т.д. вплоть до максимальной разрядности (см. рис. 3.73, *в*) n-мерного пространства.

Если выходная функция многозначная, а такой может быть представлена любая выходная многоразрядная функция двоичного кода, то в ячейки многомерного цифрового пространства, номер которых определяется натуральным входным числом, записывается также натуральное число, определяемое этой значностью. Например, для четырехзначной выходной логической функции – это двоичное число от 0 до 3, для восьмизначной – двоичное число от 0 до 7 и т.д. Здесь разрядность выходной функции может быть равной или различной входной, при этом совпадать или различаться с ней по числу разрядов, а также неограниченно превосходить ее по числу разрядов.

Все комбинационные схемы машинной арифметики, которые используют при кодировании цифр разрядов основной двоичный код, являются также логическими схемами, где логическая связь между натуральным входным числом или цифрой в пределах одного разряда и выходным натуральным определяется законами арифметики.

В машинной арифметике, где при кодировании цифр разрядов используется неосновной двоичный код, т.е. «весовые» значения цифр этого кода не совпадают с «весовыми» значениями основного двоичного кода, эквивалентность логических

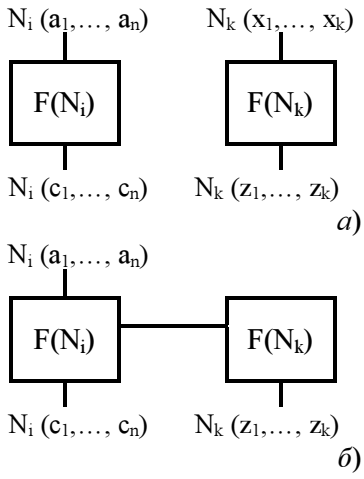


Рис. 3.74

схем и схем машинной арифметики также очевидна. Эта эквивалентность достигается преобразованием геометрических образов сигналов выходных разрядов соответствующего арифметического устройства путем расстановки цифровых сигналов координат цифрового пространства соответственно нумерации цифр основного двоичного кода.

В многовходовых логических устройствах и устройствах машинной арифметики каждый из операндов, начиная со второго, состоит из аргументов натуральных чисел более старших разрядов, чем предшествующие ему. Таким образом, разряды натурального входного числа охватывают все возможное количество таких многовходовых устройств.

В дальнейшем будем считать, что информационные разряды систематического кода, предназначенного исправлять определенные типы ошибок, заданы в основном двоичном коде и, следовательно, двоичный код контрольных разрядов этого систематического кода не является основным. Кроме того, под расширенным определением функции логического устройства будем представлять все возможные функции, а также соответствующие им схемы машинной арифметики. Структурная схема логического устройства (рис. 3.74, *а*), входным сигналом которой является систематический код N_i (a_1,…, a_n), N_k (x_1,…, x_k), позволяющий исправлять определенный тип ошибок, состоит из двух блоков, реализующих соответственно функции F(N_i), F(N_k), где вторая функция синхронна с функцией информационной части кода F(N_i), поскольку априорно задана жесткая связь между содержимым ячеек этих двух частей общего цифрового пространства.

В случае безошибочности входного кода в системе используется только информационная часть, но если при этом необходимо исключить определенного типа ошибки в логическом блоке (см. рис. 3.74, *б*), то контрольная часть логического блока должна будет формироваться в ячейках информационной части многомерного цифрового пространства в соответствии с заданной кодом зависимостью между «весовыми» значениями кодовых комбинаций информационной и контрольной его частей, а также логической функцией, реализуемой этим блоком. При этом резервирование внутри схем рис. 3.74 гарантируется только от таких неисправностей, которые приводят к ошибкам в определенном выбранном заранее количестве разрядов, предусматривающем их исправление в этом систематическом коде.

Для исправления всех одиночных ошибок в выходных сигналах логического или цифрового блока каждый из его разрядов должен выполняться отдельной схемой (рис. 3.75), которая не имеет общих логических узлов с другими разрядами, как для информационной, так и для контрольной частей систематического кода.

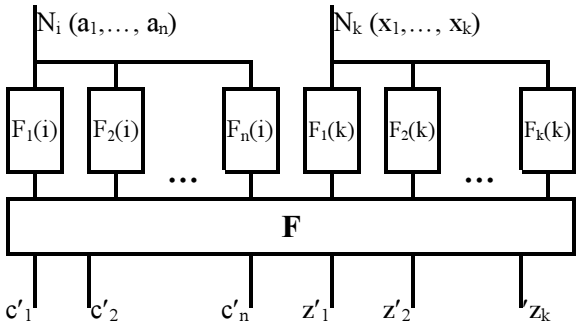


Рис. 3.75

Если при этом блок не проходной в системе, а конечный, то необходимо на его выходных шинах установить схему исправления одиночных ошибок **F**.

Другим вариантом построения аналогичного выходного блока может служить схема (рис. 3.76), где исправление ошибок, например, происходит непосредственно в логических схемах каждого разряда информационной части кода.

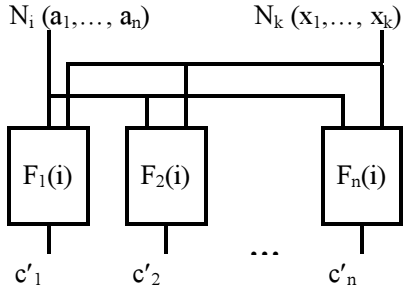


Рис. 3.76

С этой целью в многомерном цифровом пространстве координат N_i (a_1,…, a_n), N_k (x_1,…, x_k) строятся геометрические образы исправленных выходных сигналов, покрытие которых и определяет выходные сигналы c'_1 – c'_n. Эта схема более быстродействующая, чем предыдущая, но в ней отсутствует возможность поблочного резервирования, которое имеет место в предыдущей схеме.

Очевидно, что в схеме рис. 3.75 может выйти из строя любой из блоков $F_1(i)$ – $F_n(i)$ или $F_1(k)$ – $F_k(k)$ без ущерба для её правильного функционирования и поэтому она может рассматриваться как обобщение общеизвестной мажоритарной схемы рис. 3.71, но с одновременным исправлением всех ошибок определенного типа. Дальнейшее повышение вероятности безотказной работы будет заключаться в поэлементном резервировании блока исправления ошибок.

Все рассматриваемые здесь устройства являются конечными автоматами. Структурно конечный автомат может быть представлен в обобщенном виде «черным ящиком», обозначенным как система (рис. 3.77). Система имеет p входов и m выходов. Входом также является сигнал на временной тактовой шине, где такты отсчитываются от нулевого значения (t =0, 1, 2, …). Входные сигналы – A_x, B_x, C_x , … . Выходные сигналы – A_y, B_y, C_y , … . Каждый из входных и выходных сигналов может содержать свои символы конечного алфавита. В качестве этих символов выступают цифры натурального ряда {0, 1, 2, …} соответствующего основания системы счисления, закодированные определенным образом.

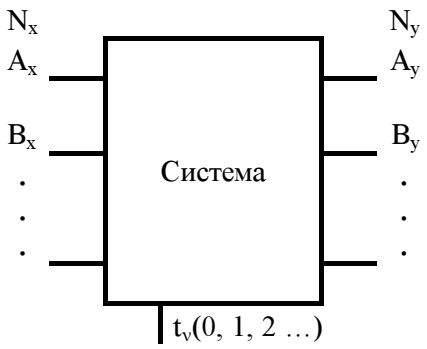


Рис. 3.77

Принципиально все основания систем счисления входных (n_{ax}, n_{bx}, n_{cx}, …) и выходных (n_{ay}, n_{by}, n_{cy}, …) сигналов могут быть различными и даже иметь принципиально разные принципы кодирования этих оснований. Тогда можно считать, что вход «черного ящика» имеет основание системы счисления $n_x = n_{ax}\, n_{bx}\, n_{cx} \ldots$, а выход – основание системы счисления $n_y = n_{ay}\, n_{by}\, n_{cy} \ldots$. Следовательно, входной и выходной сигналы автомата являются многоразрядными соответственно для чисел N_x и N_y, где разрядность входа p, выхода – m.

Тогда для каждого такта конечного автомата существует функциональная связь входа и выхода $N_x = F(N_y)$, где $N_x(A_x, B_x, C_x, \ldots)$, $N_y(A_y, B_y, C_y, \ldots)$, $N_x \geq N_y$ или $N_x \leq N_y$, а каждый разряд выхода автомата является функцией сигналов всех входных разрядов:

$$A_y = F_a\,(A_x, B_x, C_x, \ldots),\ B_y = F_b\,(A_x, B_x, C_x, \ldots),\ C_y = F_c\,(A_x, B_x, C_x, \ldots),\ \ldots\ .$$

Справедливо считается , что в то время как в качестве входов и выходов выбираются переменные N_x и N_y, которые мы можем наблюдать и измерять, природа промежуточных переменных может оставаться неизменной, а их измерение просто невозможным. Значение промежуточных переменных заключается не в характере изменения каждой из них, а скорее в их комбинированном действии на зависимости между входными и выходными переменными «черного ящика», в качестве которого может представляться конечный автомат. Комбинированное действие, так же как и переменные, вызывающие его, подчинено дискретностью времени и конечностью алфавита. Описанное действие называется *состоянием* системы. Состояние системы в момент t_v будем обозначать через S_v.

Набор всех возможных состояний системы, которые ей присущи, называется множеством состояний и обозначается через S.

При этом переход от одного состояния к другому определяется следующим: выходной сигнал в данный момент времени однозначно определяется входным сигналом и состоянием в данный момент времени; состояние в следующий момент времени однозначно определяется входным сигналом и состоянием в настоящий момент времени. Следовательно, существующее состояние автомата и входные сигналы однозначно определяют выходные сигналы, а при определенных комбинациях входных сигналов и существующем состоянии реализуется переход к новому состоянию конечного автомата и т.д.

В частном случае, что рассматривается практически во всех теоретических работах по конечным автоматам, входные и выходные символы конечного алфавита автомата это – {0, 1}, что подразумевает использование здесь только основной двоичной системы счисления, но также, следовательно, и однородность всех конечных автоматов. Для этого варианта построения автомата p его входов задают натуральное число от 0 до $(2^p - 1)$ в двоичном коде, а m выходов – натуральное число от 0 до $(2^m - 1)$ также в двоичном коде.

Таким образом, независимо от принципов кодирования цифр в позиционных системах счисления, каждому натуральному числу N_x ставится в соответствие определенное натуральное число N_y. В комбинационных устройствах эта связь жесткая, когда в многомерном цифро-векторном пространстве входных координат ей соответствует определенный геометрический образ для выходного сигнала каждого элементарного разряда автомата. Все возможные варианты покрытия этих геометрических образов определяют внутреннюю структуру такого дискретного устройства.

При этом сумма различных сочетаний из N_x по цифрам от 0 до N_x, как это представлено в гл. 1, это – множество состояний конечного автомата

$$S = C^{0}_{Nx} + C^{1}_{Nx} + C^{2}_{Nx} + \ldots + C^{Nx}_{Nx} = 2^{Nx}.$$

Сочетания $C^{0}_{Nx} = 1$ и $C^{Nx}_{Nx} = 1$ представляют здесь соответственно пустое (все ячейки пространства пусты) и универсальное множества (все ячейки пространства заполнены логическими единицами). Остальные сочетания представляют определенный класс логических функций.

Следовательно, любой геометрический образ логической или арифметической функции каждого выходного разряда определенного состояния конечного автомата содержится в одном из (Nx – 1) классов этого представления, а изменение состояния автомата заключается в переходе на иной его геометрический образ внутри одного класса либо переход к образу в следующем классе функций.

Решение задачи повышения надежности и помехоустойчивости комбинационных устройств заключается в синтезе систем с самовосстановлением.

Эта реализация может производиться с использованием методов теории цифро-векторных множеств. Сущность этих методов заключается в применении систематических совершенных кодов, позволяющих не только исправлять кратковременные ошибки определенной кратности, т.е. сбои в системе, но и осуществить резервирование отдельных элементов, блоков и всей системы.

Такое самовосстановление является кульминацией применения методов теории цифро-векторных множеств, а системы, где реализуются такие возможности, – это системы с самосознанием. Примером самообнаружения неполадок в системе являются болевые сигналы в биологических системах, каким является человек, где болевые сигналы требуют вмешательства и определенного лечения. Аналогом болевого сигнала для конечных автоматов является операция обнаружения и исправления ошибок. При ошибках, которые вызваны кратковременными помехами, они в системе с самосознанием постоянно исправляются. Если происходит выход из строя элементов в такой системе, то здесь возможны два варианта. Первый вариант поведения – это исправление длительно существующих ошибок – система больна, и она лечится. Второй вариант – ошибки не исправляются, и это «летальный» исход.

Теории приходят и уходят, а примеры остаются

И.М. Гельфанд

Глава 4

ПРИМЕРЫ СИНТЕЗА КОМБИНАЦИОННЫХ СХЕМ

В предыдущих главах приводились примеры синтеза комбинационных схем, но они в основном касались систем счисления, где используются многофазные и интегральные коды. В этих типах контролеспособных кодов нет деления на информационную и контрольную части, и поэтому синтез таких устройств одновременно решает проблему обнаружения и исправления ошибок. Однако эти коды, которые являются естественными для электроприводов и многофазных систем энергоснабжения, пока не применяются в традиционных вычислительных устройствах. Поскольку теория многомерных цифро-векторных множеств является, как показано выше, универсальной для синтеза любых комбинационных схем, то лучшим подтверждением этого теоретически доказанного факта могут дополнительно служить примеры синтеза таких схем. Для большей достоверности этого утверждения необходимо представить примеры, которые не решаются другими известными методами. Часть таких примеров приведена нами в [1–6].

В этих примерах показана, например, принципиальная возможность создания многовходовых, с числом входов до (n + 1), где n – основание системы счисления, арифметических сумматоров и логических устройств управляемого быстродействия, вплоть до предельного возможного, как в двоичной, так и любой нетрадиционной позиционной системе счисления, а также то, что они могут строиться по принципу исправления одиночных ошибок как внешнего, так и внутреннего происхождения при сочетании исправления ошибок с построением резервированных цифровых и логических систем.

Представленные в [1–6] примеры не в состоянии охватить всего спектра методов синтеза теории многомерных цифро-векторных множеств, что требует дополнительных подтверждений.

Углубляя и расширяя эти подтверждения, рассмотрим ещё более сложные примеры синтеза подобных устройств, что позволит дополнительно раскрыть все возможности представленных в первой главе алгоритмов синтеза двухзначных логических функций.

Нумерация рисунков в этой главе не сквозная, а относится к конкретному примеру.

Пример 1. Синтез устройства исправления одиночных ошибок двоичной системы счисления основания n =16

Для решения этой задачи выберем синтезированный в третьей главе систематический код, где информационные разряды кода **A** (a_1, a_2, a_3, a_4) связаны с эквивалентными цифрами контрольной части кода **X** (x_1, x_2, x_3) зависимостью, показанной на рис. 4.1.1. В соответствии с этим распределение эквивалентных цифр (**0**–**15**) информационной части кода, а также этих же цифр (0–15) при одиночных ошибках в информационных и контрольных разрядах кода будет определяться рис. 4.1.2.

a_1 a_2

a_3	0	6	5	3
a_4	7	1	2	4
	3	5	6	0
	4	2	1	7

Рис. 4.1.1

A a_1 a_2 a_3 a_4

X x_1 x_2 x_3

0	0	0	11	0	5	6	7	0	11	11	**11**	12	13	14	11
0	5	2	3	5	**5**	14	5	8	9	14	11	14	5	**14**	14
0	1	6	3	6	13	**6**	6	8	13	10	11	13	**13**	6	13
8	3	3	**3**	4	5	6	3	**8**	8	8	3	8	13	14	15
0	1	2	7	12	7	7	**7**	12	9	10	11	**12**	12	12	7
2	9	**2**	2	4	5	2	7	9	**9**	2	9	12	9	14	15
1	**1**	10	1	4	1	6	7	10	1	**10**	10	12	13	10	15
4	1	2	3	**4**	4	4	15	8	9	10	15	4	15	15	**15**

Рис. 4.1.1

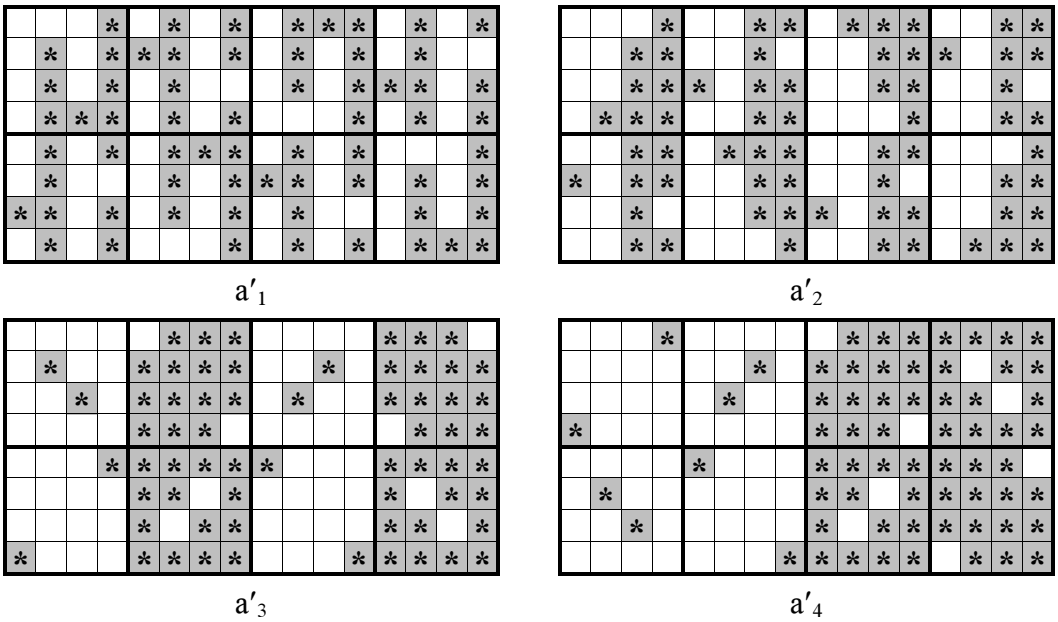
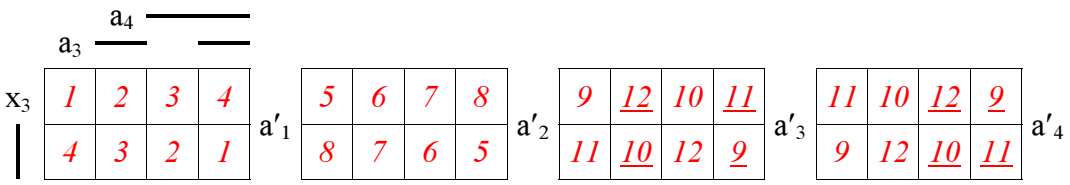


Рис. 4.1.3

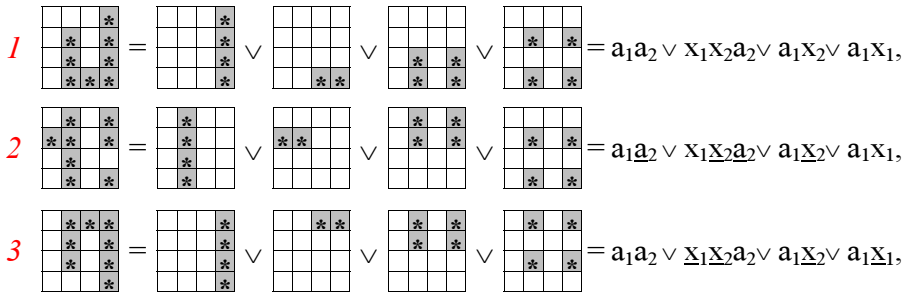
Учитывая, что $a_1 = 1 \vee 3 \vee \ldots \vee 15$, $a_2 = (2 \vee 3) \vee (6 \vee 7) \vee (10 \vee 11) \vee (14 \vee 15)$, $a_3 = (4 \vee \ldots \vee 7) \vee (12 \vee \ldots \vee 15)$, $a_4 = (8 \vee \ldots \vee 15)$, покрытие геометрических образов цифровых множеств, подчиненных этим соотношениям в пространстве координат **A** (a_1, a_2, a_3, a_4) **X** (x_1, x_2, x_3), определяет исправленные информационные сигналы a'_1, a'_2, a'_3, a'_4 (рис.4.1.3). Отдельно выделяя цифровые подмножества этих геометрических образов в координатах a_1, a_2, x_1, x_2, можно представить геометрические образы информационных разрядов в системе координат a_3, a_4, x_3. Геометрические образы сигналов a'_1, a'_2, a'_3, a'_4 в системе координат a_3, a_4, x_3 приведены на рис.4.1.4, где эти подмножества размещены в ячейках этого пространства и обозначены цифрами *1* – *12*.



Подмножества *1–4* сигнала a'_1 и подмножества *5–8* сигнала a'_2 независимые в пределах своих сигналов, а подмножества сигналов a'_3, a'_4 – зависимые (*12* ⊃ *11*, *9* ⊃ *10*, *10* ⊃ *9*, *11* ⊃ *12*; *12* ⊃ *9*, *11* ⊃ *10*, *10* ⊃ *11*, *9* ⊃ *12*). С учетом этого несложно представить логические выражения исправленных сигналов:

$$
\begin{aligned}
a'_1 &= 1\, \underline{a}_3\, \underline{a}_4\, \underline{x}_3 \vee 1\, a_3 a_4\, x_3 \vee 2\, a_3 \underline{a}_4\, \underline{x}_3 \vee 2\, \underline{a}_3 a_4 x_3 \vee 3\, \underline{a}_3 a_4\, \underline{x}_3 \vee 3\, a_3 \underline{a}_4 x_3 \vee \\
&\qquad \vee\, 4\, a_3 a_4\, \underline{x}_3 \vee 4\, \underline{a}_3 \underline{a}_4 x_3, \\
a'_2 &= 5\, \underline{a}_3\, \underline{a}_4\, \underline{x}_3 \vee 5\, a_3 a_4\, x_3 \vee 6\, a_3 \underline{a}_4\, \underline{x}_3 \vee 6\, \underline{a}_3 a_4 x_3 \vee 7 \underline{a}_3 a_4\, \underline{x}_3 \vee 7\, a_3 \underline{a}_4 x_3 \vee \\
&\qquad \vee\, 8\, a_3 a_4\, \underline{x}_3 \vee 8\, \underline{a}_3 \underline{a}_4 x_3, \\
a'_3 &= 9\, \underline{a}_4\, \underline{x}_3 \vee \underline{12}\, a_3 \underline{a}_4\, \underline{x}_3 \vee 10\, a_4\, \underline{x}_3 \vee \underline{11}\, a_3 a_4 \underline{x}_3 \vee 11\, a_4\, x_3 \vee \underline{10}\, a_3 \underline{a}_4 x_3 \vee \\
&\qquad \vee\, 12\, a_4\, x_3 \vee \underline{9}\, a_3 a_4 x_3, \\
a'_4 &= 11\, \underline{a}_3\, \underline{x}_3 \vee 10\, a_3 \underline{x}_3 \vee \underline{12}\, \underline{a}_3 a_4 \underline{x}_3 \vee \underline{9}\, a_3 a_4 \underline{x}_3 \vee 9\, \underline{a}_3 x_3 \vee 12\, a_3 x_3 \vee \\
&\qquad \vee\, \underline{10}\, \underline{a}_3\, a_4\, x_3 \vee \underline{11}\, a_3 a_4 x_3. \qquad (4.1.1)
\end{aligned}
$$

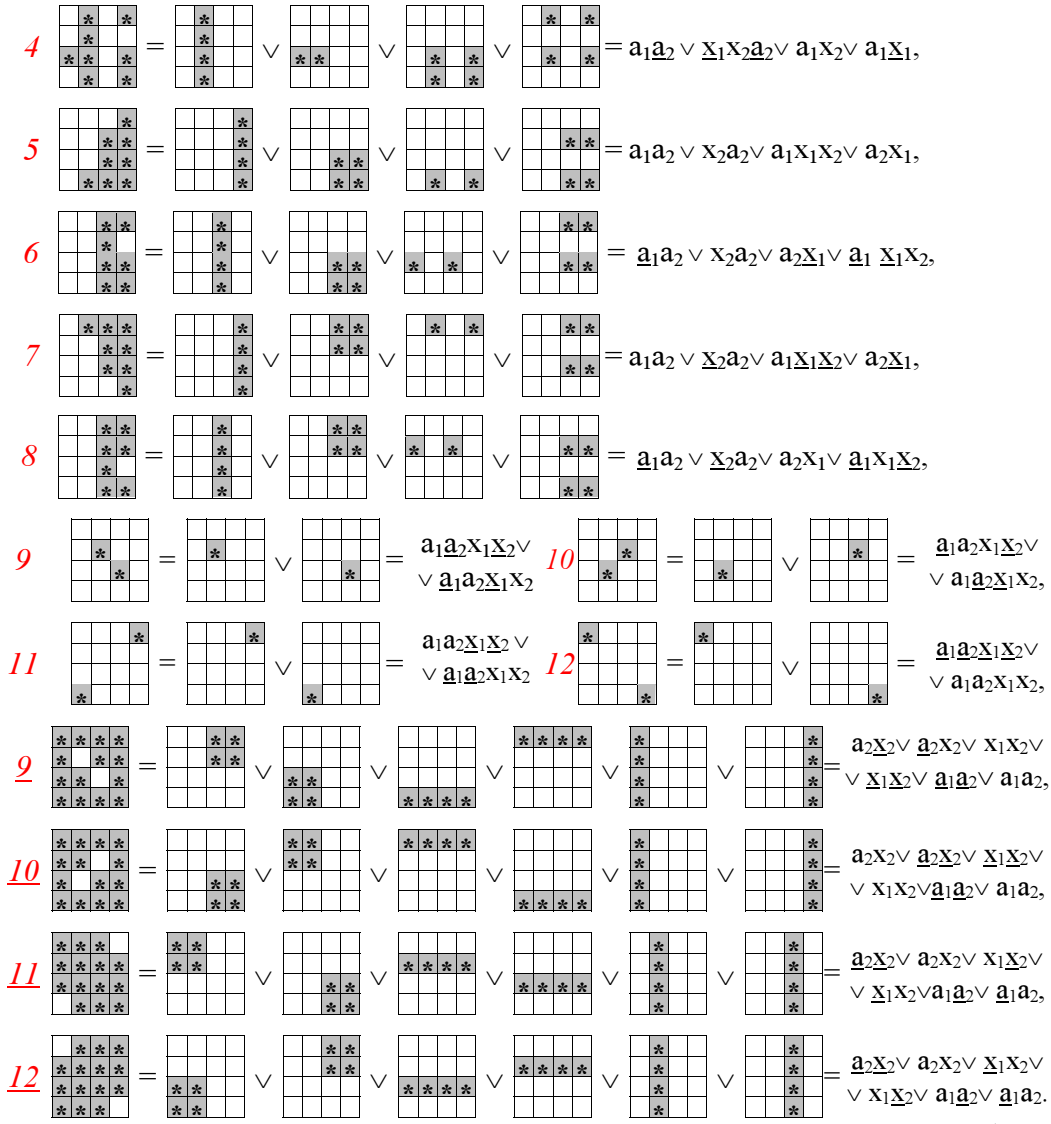
Представим подмножества *1–12* и *9–12* составленными из более мелких подмножеств, логическая запись которых проста и очевидна из их геометрических образов:



$$1 = a_1 a_2 \vee x_1 x_2 a_2 \vee a_1 x_2 \vee a_1 x_1,$$

$$2 = a_1 \underline{a}_2 \vee x_1 \underline{x}_2 \underline{a}_2 \vee a_1 \underline{x}_2 \vee a_1 x_1,$$

$$3 = a_1 a_2 \vee \underline{x}_1 \underline{x}_2 a_2 \vee a_1 \underline{x}_2 \vee a_1 \underline{x}_1,$$



(4.1.2)

Поскольку переход от геометрических образов элементарных подмножеств к записи их логических выражений прост и не требует дополнительных пояснений, в дальнейшем не будем их приводить.

Представленный здесь вариант исправления ошибок на примере информационной части систематического кода использует алгоритм многоуровневого синтеза, когда покрытие геометрических образов начинается с рассмотрения подмножеств в младших ячейках цифрового пространства с последовательным переходом к его старшим ячейкам, учитывая при этом соответствующие поглощения одних подмножеств другими. Такой вариант алгоритма синтеза позволяет минимизировать аппаратурные затраты за счет определенного снижения быстродействия логической операции исправления ошибок.

Для максимального быстродействия этой операции необходимо использовать вариант двухуровневого синтеза, когда покрытие геометрических образов начинается с рассмотрения подмножеств в самых старших ячейках пространства с последующим переходом к младшим ячейкам.

Продолжим или точнее начнем сначала рассмотрение нашего примера для двухуровневого синтеза, но прежде покажем, как необходимо переходить к геометрическим образам исправленных разрядов систематического кода непосредственно от известных соотношений цифр его информационной и контрольной частей.

Для этого обратимся к рис. 4.1.4, где показан переход от соотношений информационных и контрольных цифр систематического кода непосредственно к геометрическим образам сигналов кодирующих и декодирующих блоков.

Учитывая, что $x_1 = 1 \vee 3 \vee 5 \vee 7$, $x_2 = 2 \vee 3 \vee 6 \vee 7$, $x_3 = 4 \vee 5 \vee 6 \vee 7$, непосредственно строятся их геометрические образы (рис. 4.1.4, *а*) в двухмерном пространстве координат $A_1(a_1, a_2)$, $A_2(a_3, a_4)$, а по этим образам определяются логические зависимости

$$x_1 = a_2\underline{a_3}\underline{a_4} \vee a_2a_3a_4 \vee \underline{a_2}a_3\underline{a_4} \vee \underline{a_2}\underline{a_3}a_4;$$

$$x_2 = a_1\underline{a_3}\underline{a_4} \vee a_1a_3a_4 \vee \underline{a_1}a_3\underline{a_4} \vee \underline{a_1}\underline{a_3}a_4;$$

$$x_3 = a_3\underline{a_1}\underline{a_2} \vee a_3a_1a_2 \vee \underline{a_3}a_1\underline{a_2} \vee \underline{a_3}\underline{a_1}a_2, \qquad (4.1.3)$$

которые определяют выполнение кодирующего устройства для нашего систематического кода.

Первый этап синтеза заключается в том, что развернутые в двухмерном цифровом пространстве координат $A(a_1 – a_4)$, $X(x_1, x_2, x_3)$ соотношения информационных и контрольных цифровых сигналов (см. рис. 4.1.4, *б*) позволяют определить ячейки этого пространства, где размещаются штатные цифровые сигналы (безошибочные) для каждого из разрядов x_1, x_2, x_3, a_1, a_2, a_3, a_4 (см. рис. 4.1.4, *в – и*).

Эти ячейки определяются в нашем случае для разрядов x_1, x_2, x_3 мысленными горизонтальными линиями, а для разрядов a_1, a_2, a_3, a_4 – вертикальными, которые соединяют сигналы этих разрядов с ячейками, определяющими соотношения информационных и контрольных цифровых сигналов систематического кода.

На рис. 4.1.4, *б – в* эти соединения показаны пунктиром на примере разряда x_1, где горизонтальные линии соединяют цифровые сигналы разряда x_1 с соответствующими ячейками цифрового пространства.

После определения штатных ячеек для всех цифровых сигналов разрядов систематического кода, которые выделены красным цветом, наступает второй этап синтеза, где определяются ячейки с одиночными ошибками и одновременно выполняется процедура покрытия геометрических образов исправленных сигналов разрядов.

На этом этапе используются данные табл. 3.1.1, где достаточно подробно изложена процедура определения ячеек пространства с одиночными ошибками. Эти ячейки на рис. 4.1.4, *в – и* отмечены звездочками черного цвета.

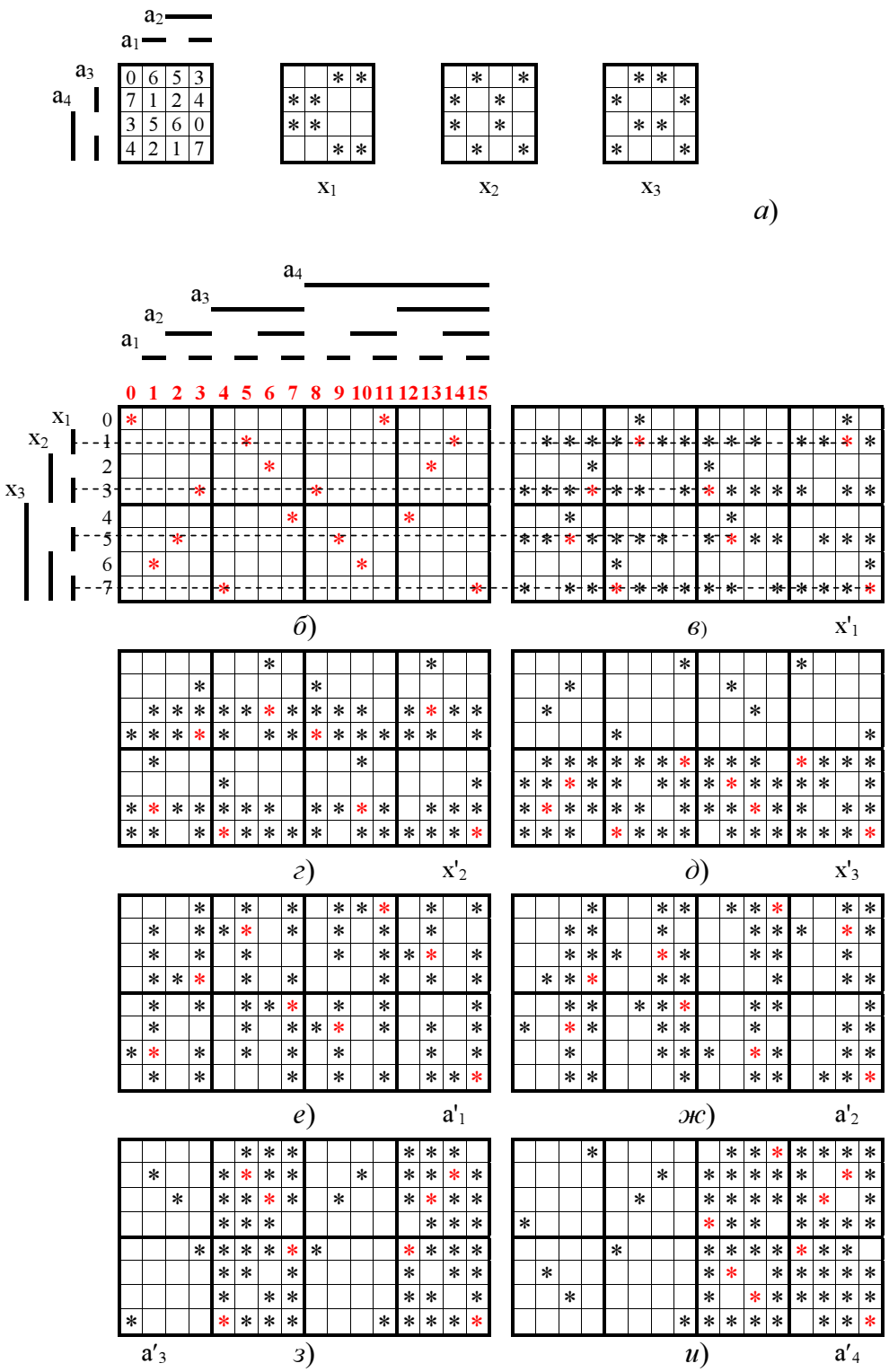

a2
a1
a3
a4
0 6 5 3
7 1 2 4
3 5 6 0
4 2 1 7
x1
x2
x3
а)
a4
a3
a2
a1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
x1
x2
x3
б)
в)
x'1
г)
x'2
д)
x'3
е)
a'1
ж)
a'2
a'3
з)
и)
a'4

Рис. 4.1.4

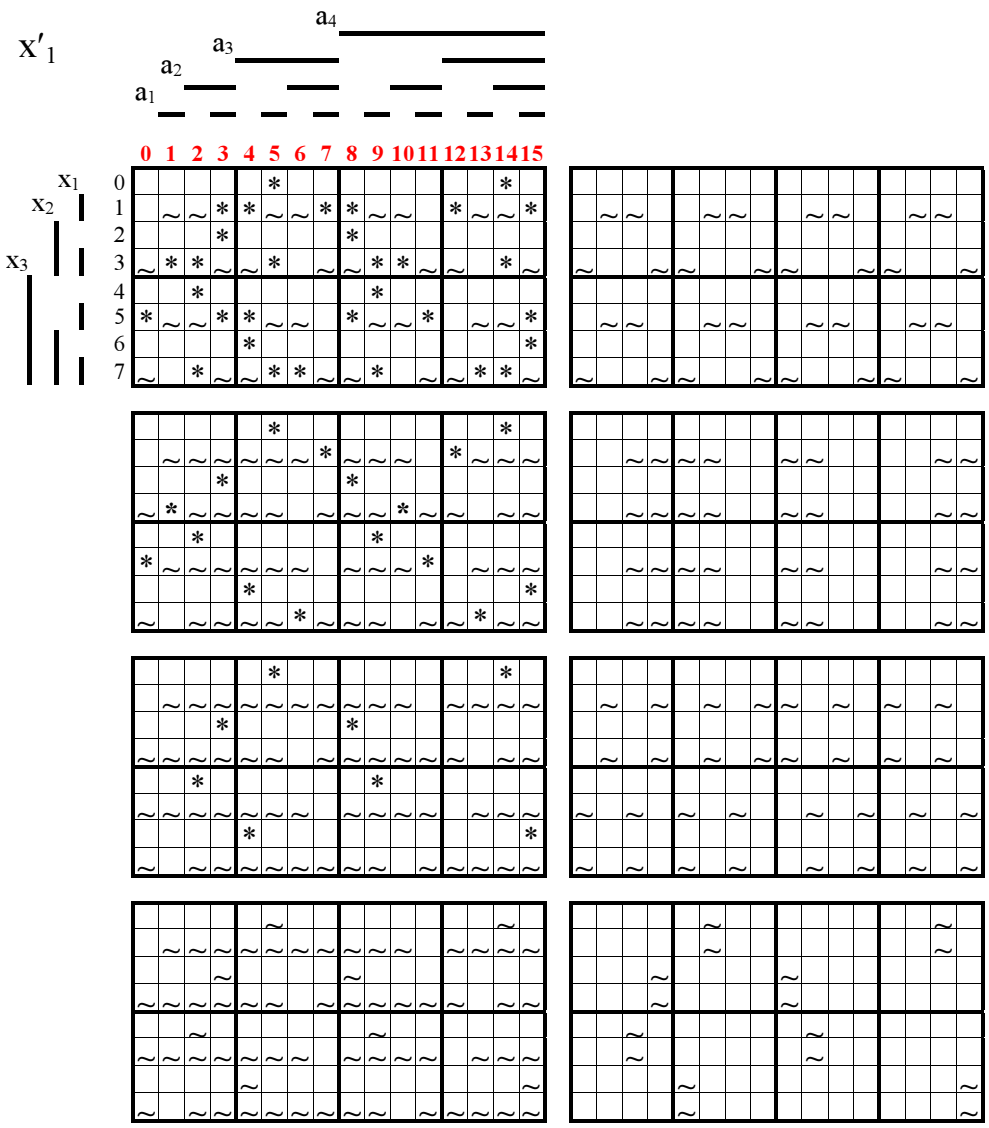


Рис. 4.1.5

$$
\begin{aligned}
x'_1 = {} & \underline{a_1}\underline{a_1}x_1x_2 \vee a_1\underline{a_2}x_1\underline{x_2} \vee \underline{a_1}a_2x_1\underline{x_2} \vee a_1a_2x_1x_2 \vee \\
& \vee a_2x_1\underline{a_3}\underline{a_4} \vee \underline{a_2}x_1a_3\underline{a_4} \vee \underline{a_2}x_1\underline{a_3}a_4 \vee a_2x_1a_3a_4 \vee \\
& \vee a_1x_1\underline{x_3}\underline{a_4} \vee \underline{a_1}x_1\underline{x_3}a_4 \vee \underline{a_1}x_1x_3\underline{a_4} \vee a_1x_1x_3a_4 \vee \\
& \vee a_1a_2x_2\underline{x_3}\underline{a_3}\underline{a_4} \vee a_1\underline{a_2}\underline{x_2}\underline{x_3}a_3\underline{a_4} \vee \underline{a_1}\underline{a_2}x_2\underline{x_3}\underline{a_3}\underline{a_4} \vee \underline{a_1}a_2\underline{x_2}\underline{x_3}a_3a_4 \vee \\
& \vee \underline{a_1}\underline{a_2}x_2x_3\underline{a_3}\underline{a_4} \vee \underline{a_1}\underline{a_2}x_2x_3a_3\underline{a_4} \vee a_1\underline{a_2}\underline{x_2}x_3\underline{a_3}a_4 \vee a_1a_2x_2x_3a_3a_4 . \qquad (4.1.4)
\end{aligned}
$$

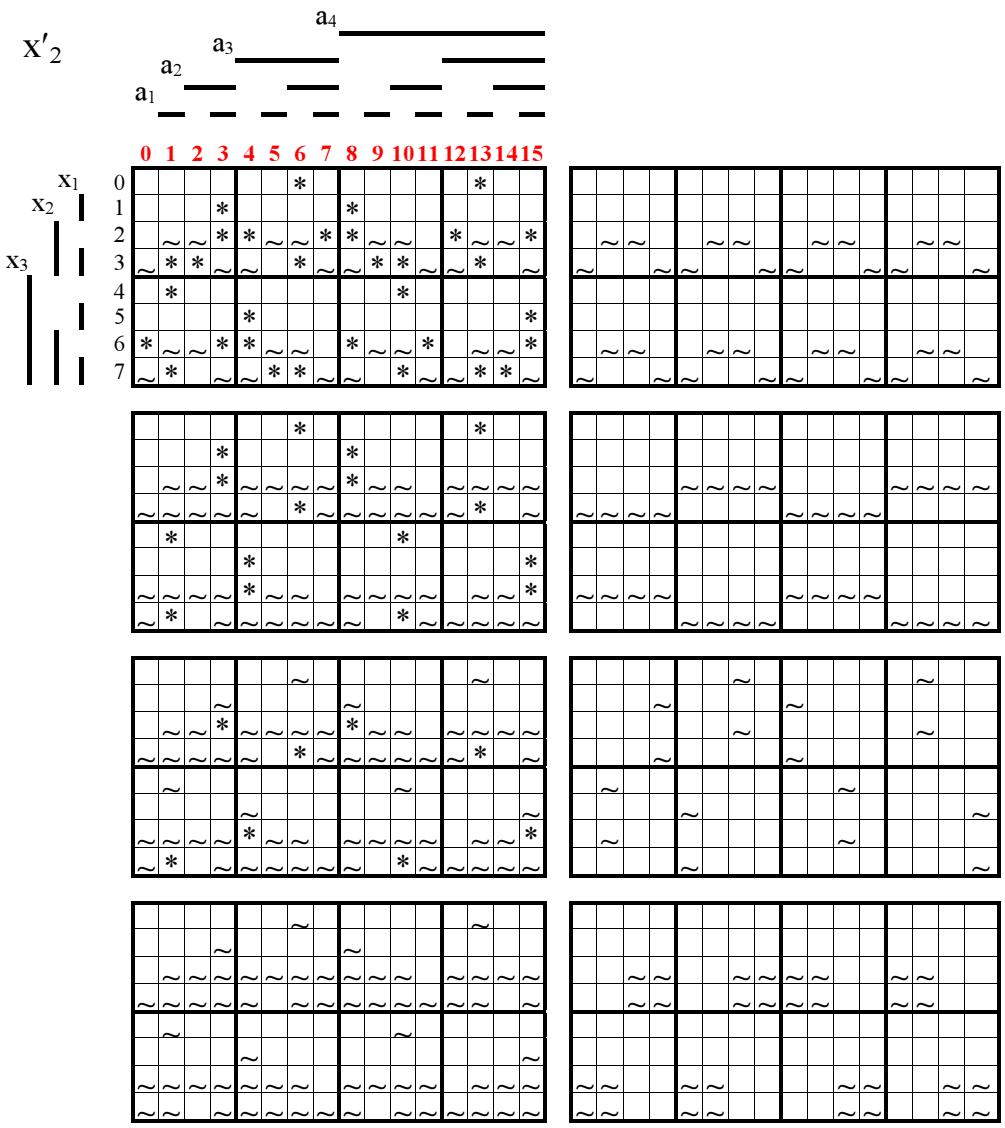


Рис. 4.1.6

$$
\begin{aligned}
x'_2 = {} & \underline{a_1}\underline{a_2}x_1x_2 \vee \underline{a_1}a_2\underline{x_1}x_2 \vee \underline{a_1}a_2\underline{x_1}x_2 \vee a_1a_2x_1x_2 \vee \\
& \vee x_1x_2\,\underline{a_3}\underline{x_3} \vee \underline{x_1}x_2\,a_3\underline{x_3} \vee \underline{x_1}x_2\,\underline{a_3}x_3 \vee x_1x_2\,a_3x_3 \vee \\
& \vee a_1a_2x_1\underline{x_3}\underline{a_3}\underline{a_4} \vee \underline{a_1}a_2\underline{x_1}\underline{x_3}a_3\underline{a_4} \vee \underline{a_1}\underline{a_2}x_1\underline{x_3}\underline{a_3}a_4 \vee a_1\underline{a_2}\underline{x_1}\underline{x_3}a_3a_4 \vee \\
& \vee a_1\underline{a_2}\underline{x_1}x_3\underline{a_3}\underline{a_4} \vee \underline{a_1}\underline{a_2}x_1x_3a_3\underline{a_4} \vee \underline{a_1}a_2\underline{x_1}x_3\underline{a_3}a_4 \vee a_1a_2x_1x_3a_3a_4 \vee \\
& \vee a_2x_2\underline{x_3}\underline{a_4} \vee \underline{a_2}x_2\underline{x_3}\underline{a_4} \vee \underline{a_2}x_2x_3\underline{a_4} \vee a_2x_2x_3a_4 . \qquad (4.1.5)
\end{aligned}
$$

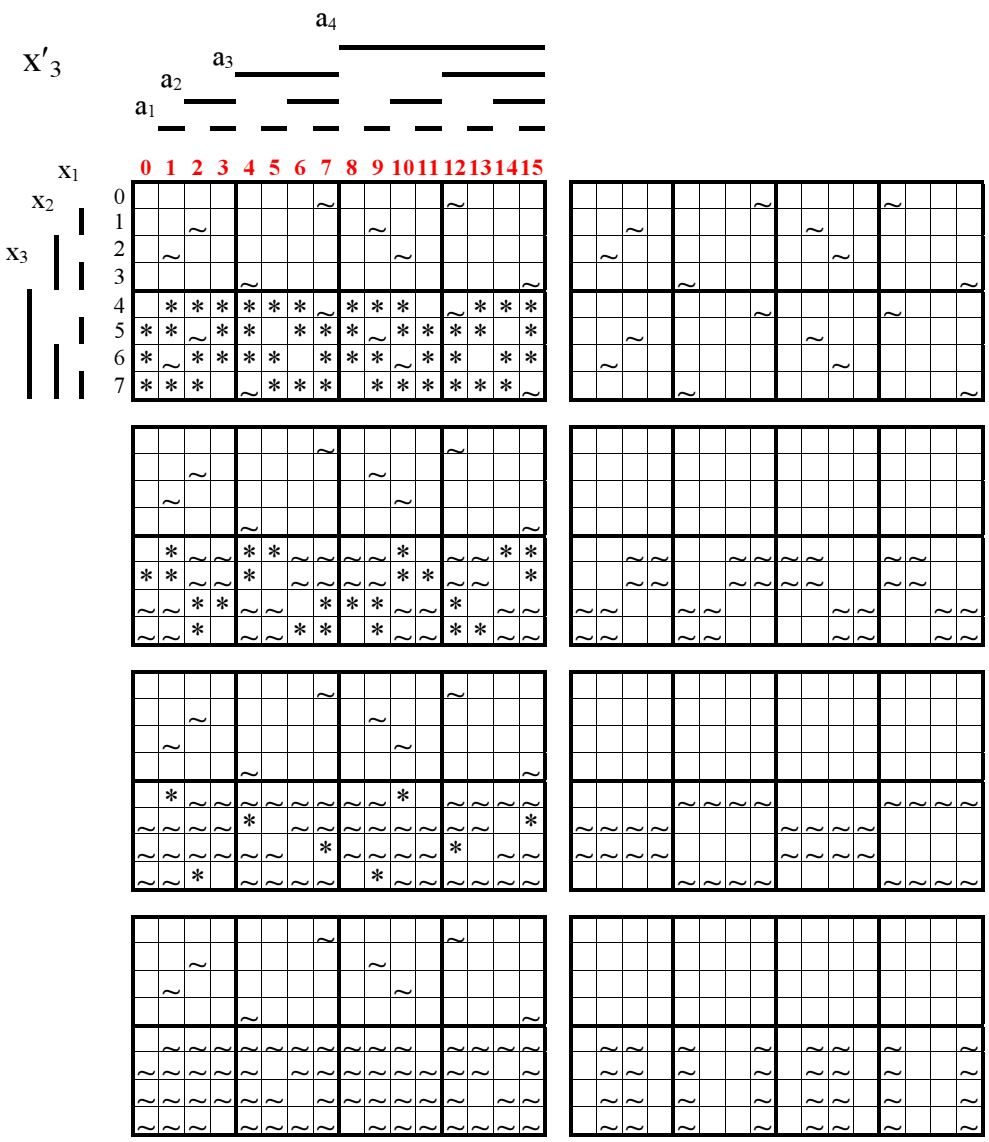


Рис. 4.1.7

$$
\begin{aligned}
x'_3 = {} & a_1\underline{a}_2x_1\underline{x}_2\underline{a}_3\underline{a}_4 \vee \underline{a}_1a_2x_1\underline{x}_2\underline{a}_3\underline{a}_4 \vee \underline{a}_1\underline{a}_2x_1x_2a_3\underline{a}_4 \vee a_1a_2\underline{x}_1\underline{x}_2a_3\underline{a}_4 \vee \\
& \vee a_1\underline{a}_2x_1\underline{x}_2\underline{a}_3a_4 \vee \underline{a}_1a_2\underline{x}_1\underline{x}_2\underline{a}_3a_4 \vee \underline{a}_1\underline{a}_2\underline{x}_1\underline{x}_2a_3a_4 \vee a_1a_2x_1x_2a_3a_4 \vee \\
& \vee a_2x_2x_3\underline{a}_4 \vee \underline{a}_2x_2x_3a_4 \vee \underline{a}_2\underline{x}_2x_3\underline{a}_4 \vee a_2x_2x_3a_4 \vee \\
& \vee x_1\underline{x}_2\,x_3\underline{a}_3 \vee \underline{x}_1\underline{x}_2\,x_3a_3 \vee \underline{x}_1\underline{x}_2\,x_3\underline{a}_3 \vee x_1x_2\,x_3a_3 \vee \\
& \vee a_1\underline{a}_2x_3\underline{a}_3 \vee \underline{a}_1a_2x_3\underline{a}_3 \vee \underline{a}_1\underline{a}_2x_3a_3 \vee a_1a_2x_3a_3. \qquad (4.1.6)
\end{aligned}
$$

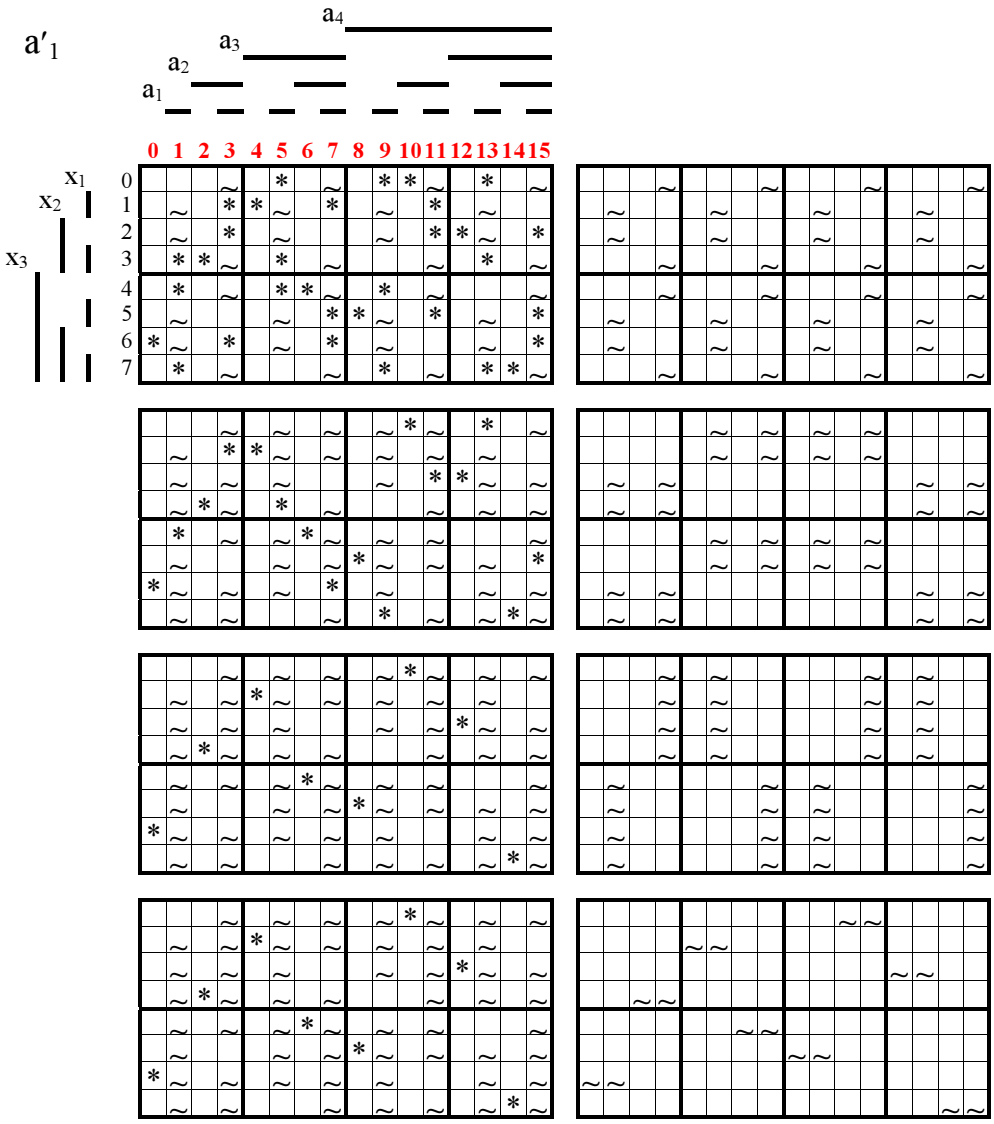


Рис. 4.1.8

$$
\begin{aligned}
a'_1 = {} & a_1\underline{a}_2x_1x_2 \vee a_1a_2\underline{x}_1\underline{x}_2 \vee a_1\underline{a}_2\underline{x}_1x_2 \vee a_1a_2x_1x_2 \vee \\
& a_1x_2\underline{a}_3a_4 \vee a_1\underline{x}_2\underline{a}_3\underline{a}_4 \vee a_1\underline{x}_2\underline{a}_3a_4 \vee a_1x_2a_3a_4 \vee \\
& a_1a_2\underline{x}_3\underline{a}_3 \vee a_1\underline{a}_2\underline{x}_3a_3 \vee a_1\underline{a}_2x_3\underline{a}_3 \vee a_1a_2x_3a_3 \vee \\
& \vee a_2x_1x_2\underline{x}_3\underline{a}_3\underline{a}_4 \vee \underline{a}_2x_1\underline{x}_2\underline{x}_3a_3\underline{a}_4 \vee a_2x_1\underline{x}_2\underline{x}_3\underline{a}_3a_4 \vee \underline{a}_2\underline{x}_1x_2\underline{x}_3a_3a_4 \vee \\
& \vee \underline{a}_2\underline{x}_1x_2x_3\underline{a}_3\underline{a}_4 \vee a_2\underline{x}_1\underline{x}_2x_3a_3\underline{a}_4 \vee \underline{a}_2x_1\underline{x}_2x_3\underline{a}_3a_4 \vee a_2x_1x_2x_3a_3a_4. \qquad (4.1.7)
\end{aligned}
$$

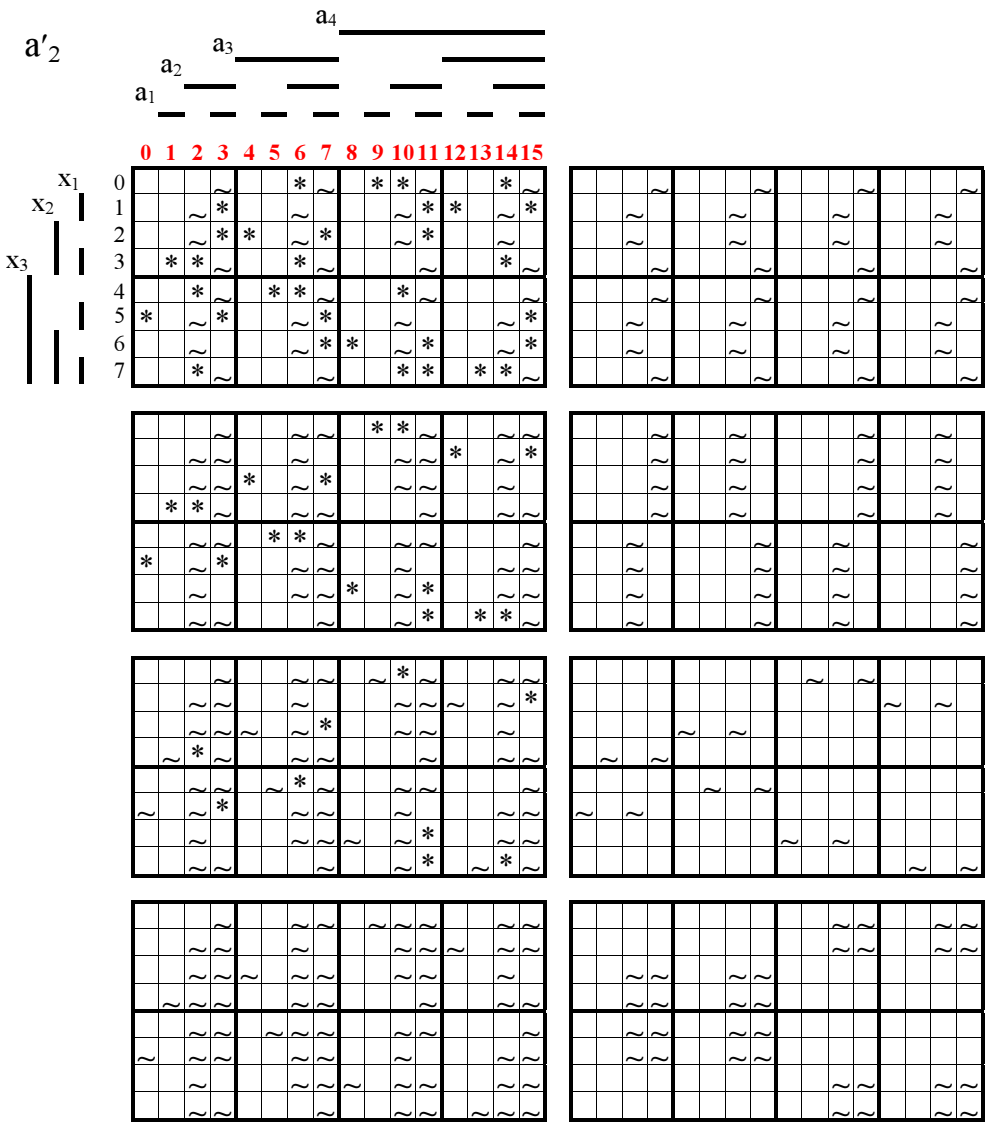


Рис. 4.1.9

$$
\begin{aligned}
a'_2 = {} & a_1a_2\underline{x_1}\underline{x_2} \vee \underline{a_1}a_2x_1\underline{x_2} \vee \underline{a_1}a_2\underline{x_1}x_2 \vee a_1a_2x_1x_2 \vee \\
& \vee a_1a_2\underline{x_3}\underline{a_3} \vee \underline{a_1}a_2x_3a_3 \vee \underline{a_1}a_2x_3\underline{a_4} \vee \underline{a_1}a_2x_3a_4 \vee \\
& \vee a_1x_1x_2\underline{x_3}\underline{a_3}\underline{a_4} \vee \underline{a_1}\underline{x_1}x_2\underline{x_3}\underline{a_3}\underline{a_4} \vee a_1\underline{x_1}\underline{x_2}\underline{x_3}\underline{a_3}a_4 \vee \underline{a_1}x_1\underline{x_2}\underline{x_3}a_3a_4 \vee \\
& \vee \underline{a_1}x_1\underline{x_2}x_3\underline{a_3}\underline{a_4} \vee a_1\underline{x_1}\underline{x_2}x_3a_3\underline{a_4} \vee \underline{a_1}\underline{x_1}x_2x_3\underline{a_3}a_4 \vee a_1x_1x_2x_3a_3a_4 \vee \\
& \vee a_2x_2\underline{x_3}\underline{a_4} \vee a_2\underline{x_2}\underline{x_3}a_4 \vee a_2\underline{x_2}x_3\underline{a_4} \vee a_2x_2x_3a_4. \qquad (4.1.8)
\end{aligned}
$$

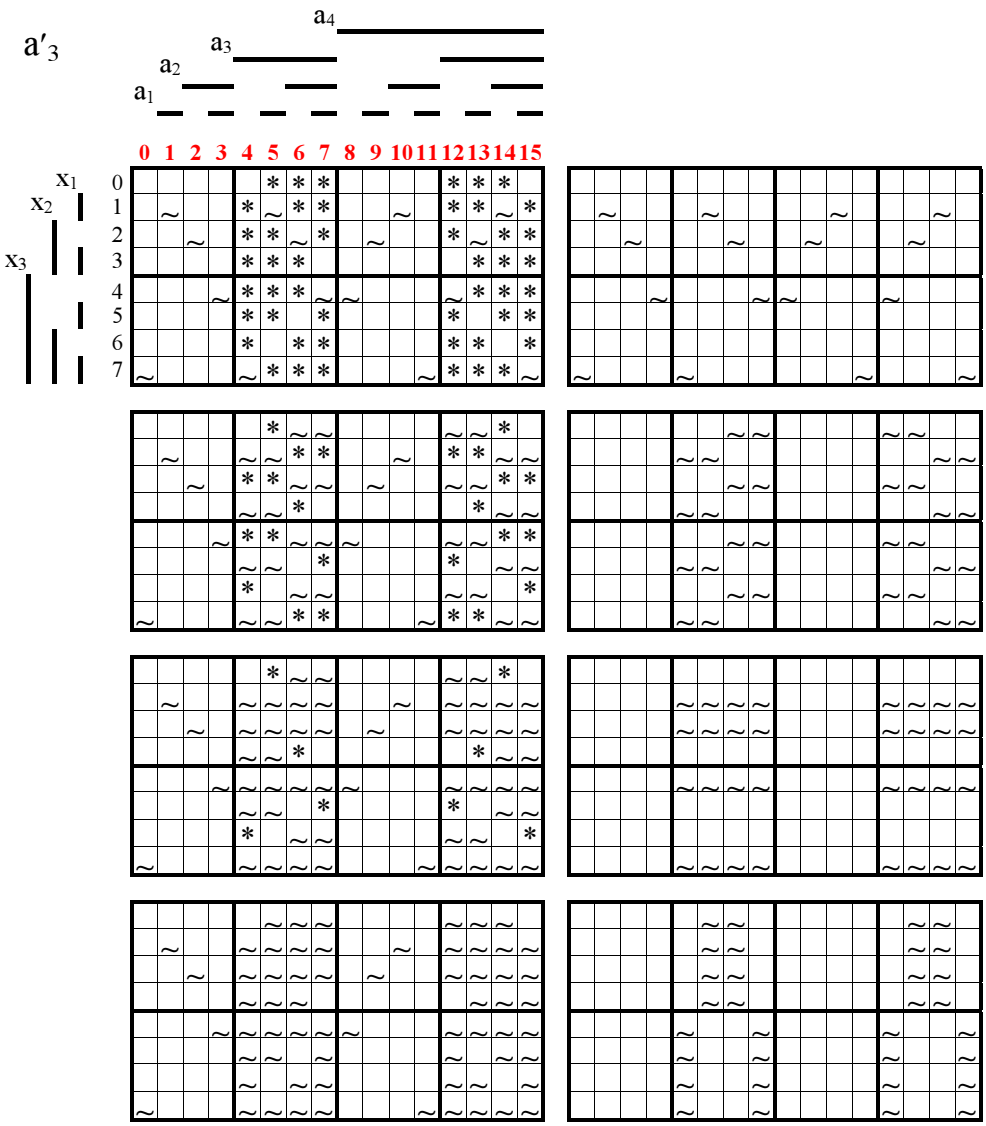


Рис. 4.1.10

$$
\begin{aligned}
a'_3 = {} & a_1\underline{a}_2x_1\underline{x}_2\underline{x}_3\underline{a}_4 \vee \underline{a}_1a_2\underline{x}_1x_2\underline{x}_3\underline{a}_4 \vee a_1\underline{a}_2\underline{x}_1x_2\underline{x}_3a_4 \vee \underline{a}_1a_2x_1\underline{x}_2\underline{x}_3a_4 \vee \\
& \vee \underline{a}_1\underline{a}_2x_1x_2x_3\underline{a}_4 \vee a_1a_2\underline{x}_1\underline{x}_2x_3\underline{a}_4 \vee \underline{a}_1\underline{a}_2\underline{x}_1\underline{x}_2x_3a_4 \vee a_1a_2x_1x_2x_3a_4 \vee \\
& \vee \underline{a}_2x_1a_3\underline{a}_4 \vee a_2\underline{x}_1a_3\underline{a}_4 \vee \underline{a}_2\underline{x}_1a_3a_4 \vee a_2x_1a_3a_4 \vee \\
& \vee x_1\underline{x}_2\underline{x}_3a_3 \vee \underline{x}_1x_2\underline{x}_3a_3 \vee x_1\underline{x}_2x_3a_3 \vee x_1x_2x_3a_3 \vee \\
& \vee a_1\underline{a}_2\underline{x}_3a_3 \vee \underline{a}_1a_2\underline{x}_3a_3 \vee \underline{a}_1\underline{a}_2x_3a_3 \vee a_1a_2x_3a_3. \qquad (4.1.9)
\end{aligned}
$$

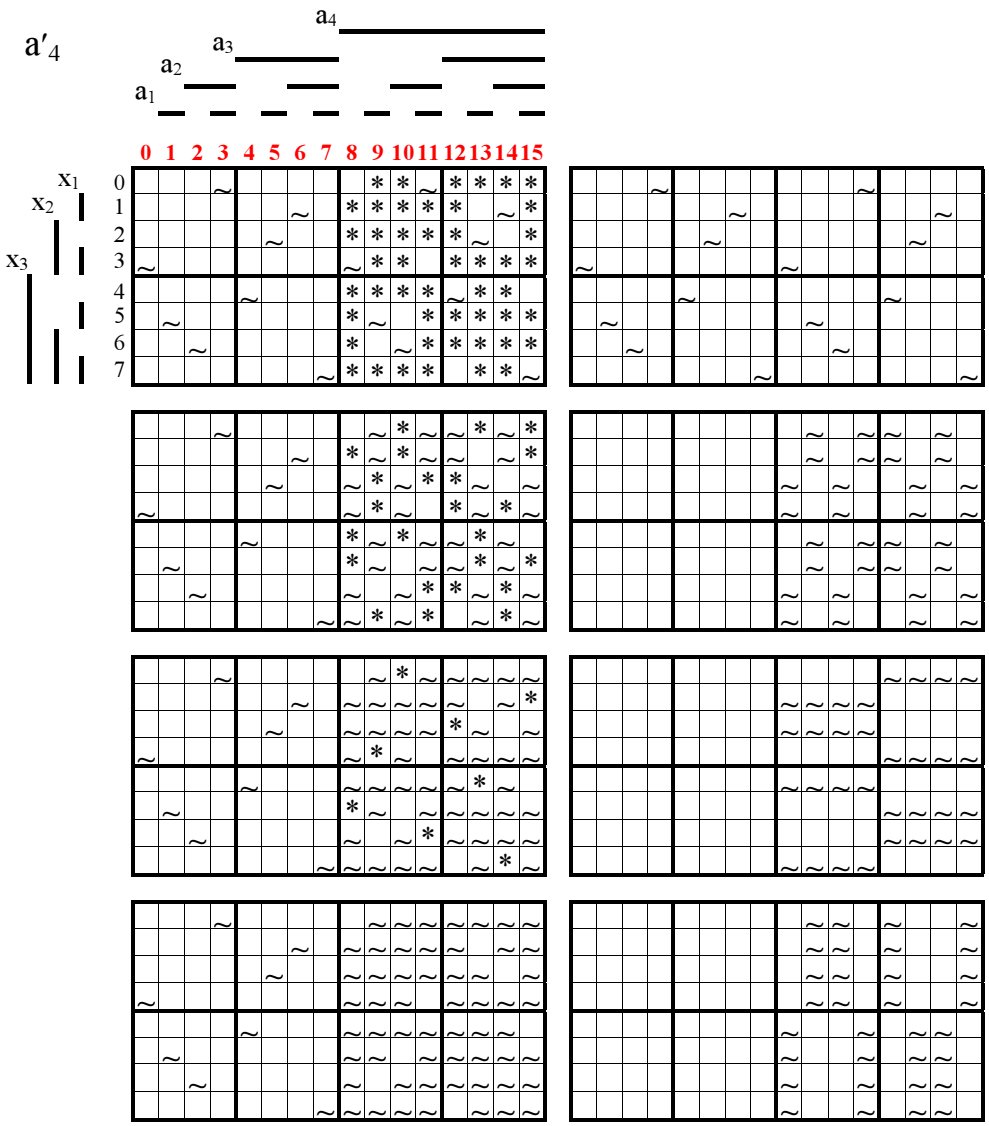


Рис. 4.1.11

$$
\begin{aligned}
a'_4 = {} & \underline{a_1}\underline{a_2}x_1x_2\underline{x_3}\underline{a_3} \vee a_1a_2\underline{x_1}x_2\underline{x_3}\underline{a_3} \vee a_1\underline{a_2}\underline{x_1}x_2\underline{x_3}a_3 \vee \underline{a_1}a_2x_1\underline{x_2}\underline{x_3}a_3 \vee \\
& \vee a_1\underline{a_2}x_1\underline{x_2}x_3\underline{a_3} \vee \underline{a_1}a_2\underline{x_1}x_2x_3\underline{a_3} \vee \underline{a_1}\underline{a_2}\underline{x_1}x_2x_3\underline{a_3} \vee a_1a_2x_1x_2x_3a_3 \vee \\
& \vee a_1\underline{x_2}\underline{a_3}a_4 \vee \underline{a_1}x_2\underline{a_3}a_4 \vee \underline{a_1}\underline{x_2}a_3a_4 \vee a_1x_2a_3a_4 \vee \\
& \vee x_1\underline{x_2}\underline{x_3}a_3 \vee \underline{x_1}x_2\underline{x_3}a_3 \vee x_1\underline{x_2}x_3a_3 \vee x_1x_2x_3a_3 \vee \\
& \vee a_1\underline{a_2}\underline{x_3}a_3 \vee \underline{a_1}a_2\underline{x_3}a_3 \vee \underline{a_1}\underline{a_2}x_3a_3 \vee a_1a_2x_3a_3. \qquad (4.1.10)
\end{aligned}
$$

Покрытие геометрических образов цифровых сигналов разрядов x'_1, x'_2, x'_3, a'_1, a'_2, a'_3, a'_4 представлено на рис. 4.1.5 – 4.1.11, где это реализовано движением «сверху вниз». Для большей прозрачности синтеза в каждом из этих рисунков приведено два столбца геометрических образов. В первом левом столбце показано последовательное покрытие подмножествами конкретного геометрического образа цифровых сигналов разрядов систематического кода, а в правом – изображены геометрические образы покрывающих их подмножеств, где их ячейки отмечены знаком ~. В результате покрытия конкретного геометрического образа подмножеством, расположенным с ним на одной горизонтали, знак звездочки в геометрическом образе заменяется знаком ~, а измененный таким образом геометрический образ переходит во все нижние его представления первого столбца. Это движение осуществляется до тех пор, пока в геометрическом образе все звездочки не будут заменены знаком ~.

В первом столбце всех подмножеств, которые в общем случае являются элементарными и поэтому весьма прозрачными для записи в аналитическом виде, представлены элементы логических сумм для аналитической записи конкретного геометрического образа цифровых сигналов исправленных разрядов систематического кода.

Логические выражения (4.1.4) – (4.1.10) полностью определяют выполнение двухуровневого декодирующего устройства, обладающего максимально возможным быстродействием.

Рассмотрев этот пример, видим что его можно значительно упростить. Из геометрических образов исправленных сигналов разрядов a'_1, a'_2, a'_3 , представленных на с. 206, видно, что соответствующими мысленными поворотами относительно конкретных осей симметрии цифрового пространства можно получить геометрические образы сигналов разрядов x'_1, x'_2, x'_3.

Для разрядов a'_1, a'_2 однократные повороты относительно оси *2* двухмерного цифрового пространства координат $A(a_1, a_2)$, $X(x_1, x_2)$ позволяют получить геометрические образы сигналов контрольных разрядов x'_1, x'_2. Эти преобразования распространяются и на соответствующие логические функции, где должны быть осуществлены следующие замены:

$$a'_1(a_1, a_2, a_3, a_4, x_1, x_2, x_3) \rightarrow x'_1 = a'_1(x_1, x_2, a_3, a_4, a_1, a_2, x_3);$$

$$a'_2(a_1, a_2, a_3, a_4, x_1, x_2, x_3) \rightarrow x'_2 = a'_2(x_1, x_2, a_3, a_4, a_1, a_2, x_3).$$

Для разряда a'_3 последовательные повороты первоначально относительно оси *2* двухмерного цифрового пространства координат $A(a_1, a_2, a_3)$, $X(x_1, x_2, a_3)$ выполняются следующей заменой:

$$a'_3(a_1, a_2, a_3, a_4, x_1, x_2, x_3) \rightarrow x^*_3 = a'_3(x_1, x_2, x_3, a_4, a_1, a_2, a_3)$$

и дают промежуточный геометрический образ логической функции, а второй поворот относительно оси *2* двухмерного цифрового пространства координат a_1, a_2 либо координат x_1, x_2 позволяет получить геометрический образ исправленного сигнала контрольного разряда x'_3.

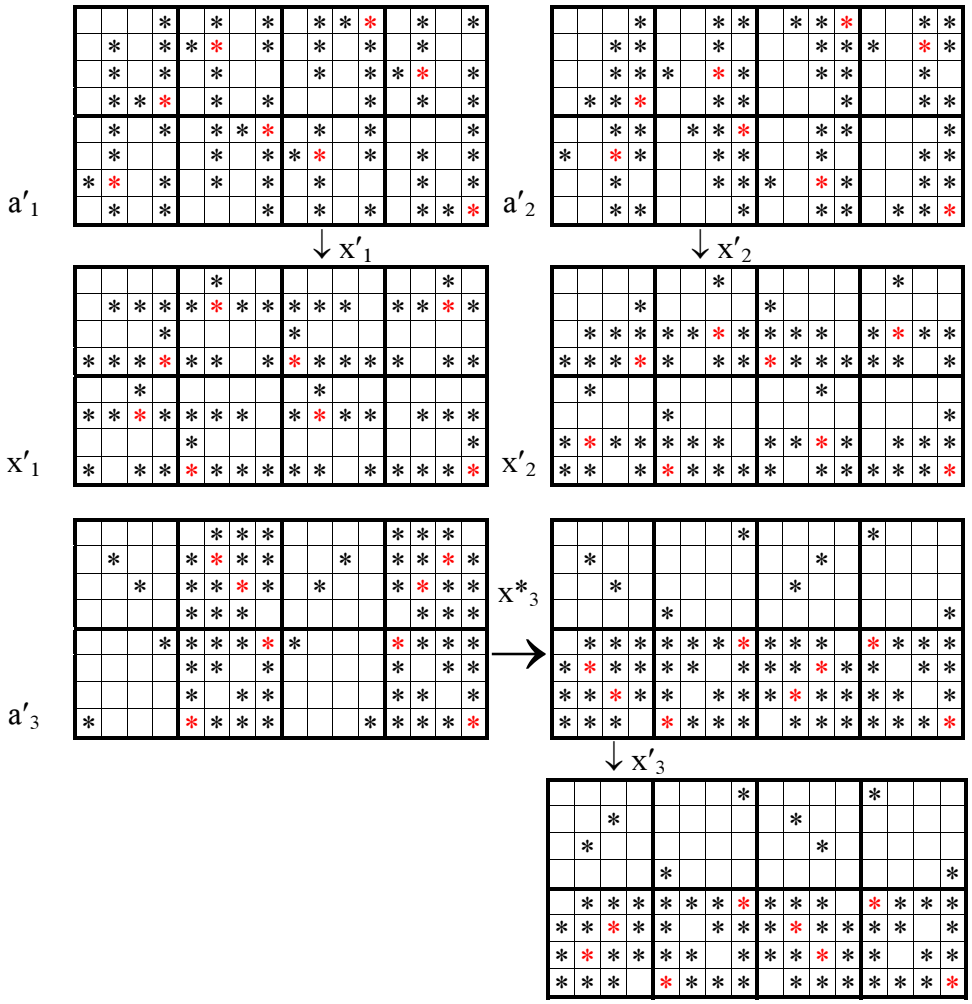
Эти двойные преобразования представляются следующими схемами замен координат:

$a'_3(a_1, a_2, a_3, a_4, x_1, x_2, x_3) \to x^*_3 = a'_3\ (x_1, x_2, x_3, a_4, a_1, a_2, a_3) \to\ x'_3 = x^*_3\ (x_2, x_1, x_3, a_4, a_1, a_2, a_3)$ либо $a'_3(a_1, a_2, a_3, a_4, x_1, x_2, x_3) \to x^*_3 = a'_3\ (x_1, x_2, x_3, a_4, a_1, a_2, a_3) \to$
$\to x'_3 = x^*_3\ (x_1, x_2, x_3, a_4, a_2, a_1, a_3)$,

что комплексно представляется двумя вариантами:

Вариант 1	$a_1 \to x_2$	$a_2 \to x_1$	$a_3 \leftrightarrow x_3$	$x_1 \to a_1$	$x_2 \to a_2$
Вариант 2	$a_1 \to x_1$	$a_2 \to x_2$		$x_1 \to a_2$	$x_2 \to a_1$

Эти геометрические преобразования выглядят следующим образом:



Пример 2. Синтез устройства исправления одиночных ошибок двоичной системы счисления для любых синтезированных кодов основания $n = 2^4$

Примеры синтеза быстродействующих арифметических и логических устройств, использующие методы многомерных цифровых множеств и представленные в [1– 3], относятся к методам точных математических моделей и не требуют дополнительных пояснений. Другие примеры в [4–6], решающие задачи исправления всех одиночных ошибок в системах счисления основания n = 16, опираются на эвристический метод решения задачи, который обычно противопоставляется в математике формальным методам решения, опирающимся на точные математические модели.

Известно, что использование эвристик сокращает время решения задачи по сравнению с методом полного направленного перебора возможных алгоритмов и относится к множеству допустимых решений. Подобным образом нами было синтезировано геометрическим эвристическим способом 18 кодов, исправляющих одиночные ошибки в любых логических устройствах и блоках машинной арифметики. Несмотря на принятые нами в этом синтезе ограничения, которые заключаются в использовании здесь только дружественных кодов в информационной и контрольной его частях, на основе этих примеров можно сделать следующие обобщающие выводы: для устройств исправления одиночных ошибок все 18 кодов полностью равноценны в смысле затрат оборудования и быстродействия; при схемной реализации конкретной логической или математической функции в этих кодах имеется один или несколько кодов, позволяющих получить решения, оптимальные по затратам оборудования.

Следует заметить, что в этих синтезированных нами кодах использована многозначная логика, когда в ячейках многомерного цифрового пространства для информационной части кода располагаются не логические значения нуль (0*) или единица (1*), а цифровые значения 0–7 контрольной его части. Тем самым получается геометрический образ логической функции синтезированного систематического кода в четырехмерном цифро-векторном пространстве.

Для решения задачи синтеза устройства исправления одиночных ошибок любого систематического кода основания $n = 2^4$ нужно определить все эти коды. Необходимо отметить, что уже на начальных этапах исследований совершенных кодов высказывалось мнение [7], что «имеются некоторые результаты, показывающие, что совершенных кодов мало, и кажется вполне правдоподобным, что не существует других совершенных кодов», кроме известных в это время. Ошибочность этого утверждения представим в данном примере.

Очевидно, что к геометрическому образу любого из уже известных кодов можно применить мысленные повороты относительно осей симметрии цифровекторного пространства и получить, таким образом, все другие эквивалентные коды, исправляющие, например, все одиночные ошибки.

В первой главе было выведено число возможных положений геометрических образов логических функций $s_i = 2^i\, i!$, где i – мерность цифрового пространства, совпадающая с числом разрядов двоичного кода.

Для поворотов относительно осей симметрии только информационной части систематического кода i = 4 получается число возможных положений геометрического образа s_4 = 384, что вполне обозримо, тогда как для следующего кода, исправляющего одиночные ошибки, i = 11 это число s_{11} = 81 749 606 400 – огромно и необозримо. При наличии симметрии исходной фигуры, что в нашем случае здесь имеет место, число этих геометрических образов будет уменьшено до значения $s_i = 2^k i!$, где k – число разрядов контрольной части систематического кода. Тогда s_4 = 192, s_{11} = 638 668 800, что также для второго случая необозримо.

Дополнительно выполняя повороты относительно осей симметрии контрольной части кода, число кодов, исправляющих одиночные ошибки, будет увеличено в $2^k k!$ раз, что не представляется возможным все их рассмотреть.

Поэтому представим коды, исправляющие одиночные ошибки для основания системы счисления n = 16, получаемые только поворотами относительно осей симметрии информационной части систематического кода.

Обратимся к первой главе, где в п. 1.2 представлена процедура формирования перестановок, в том числе и для четырех операндов, в качестве которых будем рассматривать разряды a_1, a_2, a_3, a_4 информационной части систематического кода, и эти разряды будем записывать их индексами 1, 2, 3, 4, т.е. $a_1, a_2, a_3, a_4 \equiv 1, 2, 3, 4$.

В соответствии с этой процедурой в табл. 4.2.1 представлены все положения координат четырехмерного цифрового пространства, которые получаются мысленными поворотами относительно осей его симметрии, где первая половина первого столбца этой таблицы составлена, исходя только из перестановок его координат (разрядов 1, 2, 3, 4), а другие столбцы первой половины образуются переводом одного или двух координат из прямого кода в обратный. При этом столбцы этой таблицы обозначены цифрами 0–7, строки – 0–47. Вторая часть всех столбцов (строки под номерами 23–47) таблицы получается переводом координат пространства из прямого кода в обратный, а для двоичного кода это – простое инвертирование всех его разрядов.

Выбрав в качестве исходного первый код, например с координатами 0, 15, табл. 4.2.1, 4.2.2 и осуществляя мысленные повороты относительно информационных координат, получим ряд кодов, исправляющих одиночные ошибки основания n = 16. Цветной шрифт кодов в этой таблице соответствует 18 кодам, полученным нами эвристическим методом.

Таблица 4.2.1

	0	1	2	3	4	5	6	7
0	1234	1234	1234	1234	1234	1234	1234	1234
1	2134	2134	2134	2134	2134	2134	2134	2134
2	3214	3214	3214	3214	3214	3214	3214	3214
3	4321	4321	4321	4321	4321	4321	4321	4321
4	1324	1324	1324	1324	1324	1324	1324	1324
5	2314	2314	2314	2314	2314	2314	2314	2314
6	3124	3124	3124	3124	3124	3124	3124	3124
7	4231	4231	4231	4231	4231	4231	4231	4231
8	1432	1432	1432	1432	1432	1432	1432	1432
9	2431	2431	2431	2431	2431	2431	2431	2431
10	3412	3412	3412	3412	3412	3412	3412	3412
11	4123	4123	4123	4123	4123	4123	4123	4123
12	1243	1243	1243	1243	1243	1243	1243	1243
13	2143	2143	2143	2143	2143	2143	2143	2143
14	3241	3241	3241	3241	3241	3241	3241	3241
15	4312	4312	4312	4312	4312	4312	4312	4312
16	1342	1342	1342	1342	1342	1342	1342	1342
17	2341	2341	2341	2341	2341	2341	2341	2341
18	3142	3142	3142	3142	3142	3142	3142	3142
19	4213	4213	4213	4213	4213	4213	4213	4213
20	1423	1423	1423	1423	1423	1423	1423	1423
21	2413	2413	2413	2413	2413	2413	2413	2413
22	3421	3421	3421	3421	3421	3421	3421	3421
23	4132	4132	4132	4132	4132	4132	4132	4132
24	1234	1234	1234	1234	1234	1234	1234	1234
25	2134	2134	2134	2134	2134	2134	2134	2134
26	3214	3214	3214	3214	3214	3214	3214	3214
27	4321	4321	4321	4321	4321	4321	4321	4321
28	1324	1324	1324	1324	1324	1324	1324	1324
29	2314	2314	2314	2314	2314	2314	2314	2314
30	3124	3124	3124	3124	3124	3124	3124	3124
31	4231	4231	4231	4231	4231	4231	4231	4231
32	1432	1432	1432	1432	1432	1432	1432	1432
33	2431	2431	2431	2431	2431	2431	2431	2431
34	3412	3412	3412	3412	3412	3412	3412	3412
35	4123	4123	4123	4123	4123	4123	4123	4123
36	1243	1243	1243	1243	1243	1243	1243	1243
37	2143	2143	2143	2143	2143	2143	2143	2143
38	3241	3241	3241	3241	3241	3241	3241	3241
39	4312	4312	4312	4312	4312	4312	4312	4312
40	1342	1342	1342	1342	1342	1342	1342	1342
41	2341	2341	2341	2341	2341	2341	2341	2341
42	3142	3142	3142	3142	3142	3142	3142	3142
43	4213	4213	4213	4213	4213	4213	4213	4213
44	1423	1423	1423	1423	1423	1423	1423	1423
45	2413	2413	2413	2413	2413	2413	2413	2413
46	3421	3421	3421	3421	3421	3421	3421	3421
47	4132	4132	4132	4132	4132	4132	4132	4132

Результаты этих преобразований сведены в табл. 4.2.2, где повторное получение кодов в результате соответствующего поворота отмечено затемненной клеткой цифрового пространства координат таблицы. Анало-

гично отмечены координаты ячеек табл. 4.2.1, совершаемые после заполнения табл. 4.2.2.

Из табл. 4.2.2 следует, что число кодов, исправляющих одиночные ошибки, в нашем случае равно 192.

Таблица 4.2.2

	0	1	2	3	4	5	6	7
0	0 7 3 4 5 2 6 1 6 1 5 2 3 4 0 7	7 0 4 3 2 5 1 6 1 6 2 5 4 3 7 0	3 4 0 7 6 1 5 2 5 2 6 1 0 7 3 4	5 2 6 1 0 7 3 4 3 4 0 7 6 1 5 2	6 1 5 2 3 4 0 7 0 7 3 4 5 2 6 1	4 3 7 0 1 6 2 5 2 5 1 6 7 0 4 3	2 5 1 6 7 0 4 3 4 3 7 0 1 6 2 5	1 6 2 5 4 3 7 0 7 0 4 3 2 5 1 6
1	0 3 7 4 5 6 2 1 6 5 1 2 3 0 4 7	3 0 4 7 6 5 1 2 5 6 2 1 0 3 7 4	7 4 0 3 2 1 5 6 1 2 6 5 4 7 3 0	5 6 2 1 0 3 7 4 3 0 4 7 6 5 1 2	6 5 1 2 3 0 4 7 0 3 7 4 5 6 2 1	4 7 3 0 1 2 6 5 2 1 5 6 7 4 0 3	6 5 1 2 3 0 4 7 0 3 7 4 5 6 2 1	5 6 2 1 0 3 7 4 3 0 4 7 6 5 1 2
2	0 5 3 6 7 2 4 1 6 3 5 0 1 4 2 7	5 0 6 3 2 7 1 4 3 6 0 5 4 1 7 2	3 6 0 5 4 1 7 2 5 0 6 3 2 7 1 4	7 2 4 1 0 5 3 6 1 4 2 7 6 3 5 0	6 3 5 0 1 4 2 7 0 5 3 6 7 2 4 1	6 3 5 0 1 4 2 7 0 5 3 6 7 2 4 1	2 7 1 4 5 0 6 3 4 1 7 2 3 6 0 5	3 6 0 5 4 1 7 2 5 0 6 3 2 7 1 4
3	0 6 5 3 3 5 6 0 7 1 2 4 4 2 1 7	6 0 3 5 5 3 0 6 1 7 4 2 2 4 7 1	5 3 0 6 6 0 3 5 2 4 7 1 1 7 4 2	3 5 6 0 0 6 5 3 4 2 1 7 7 1 2 4	7 1 2 4 4 2 1 7 0 6 5 3 3 5 6 0	3 5 6 0 0 6 5 3 4 2 1 7 7 1 2 4	5 3 0 6 6 0 3 5 2 4 7 1 1 7 4 2	1 7 4 2 2 4 7 1 6 0 3 5 5 3 0 6
4	0 7 5 2 3 4 6 1 6 1 3 4 5 2 0 7	7 0 2 5 4 3 1 6 1 6 4 3 2 5 7 0	5 2 0 7 6 1 3 4 3 4 6 1 0 7 5 2	3 4 6 1 0 7 5 2 5 2 0 7 6 1 3 4	6 1 3 4 5 2 0 7 0 7 5 2 3 4 6 1	2 5 7 0 1 6 4 3 4 3 1 6 7 0 2 5	4 3 1 6 7 0 2 5 2 5 7 0 1 6 4 3	1 6 4 3 2 5 7 0 7 0 2 5 4 3 1 6
5	0 3 5 6 7 4 2 1 6 5 3 0 1 2 4 7	3 0 6 5 4 7 1 2 5 6 0 3 2 1 7 4	5 6 0 3 2 1 7 4 3 0 6 5 4 7 1 2	7 4 2 1 0 3 5 6 1 2 4 7 6 5 3 0	6 5 3 0 1 2 4 7 0 3 5 6 7 4 2 1	6 5 3 0 1 2 4 7 0 3 5 6 7 4 2 1	4 7 1 2 3 0 6 5 2 1 7 4 5 6 0 3	5 6 0 3 2 1 7 4 3 0 6 5 4 7 1 2
6	0 5 7 2 3 6 4 1 6 3 1 4 5 0 2 7	5 0 2 7 6 3 1 4 3 6 4 1 0 5 7 2	7 2 0 5 4 1 3 6 1 4 6 3 2 7 5 0	3 6 4 1 0 5 7 2 5 0 2 7 6 3 1 4	6 3 1 4 5 0 2 7 0 5 7 2 3 6 4 1	2 7 5 0 1 4 6 3 4 1 3 6 7 2 0 5	6 3 1 4 5 0 2 7 0 5 7 2 3 6 4 1	3 6 4 1 0 5 7 2 5 0 2 7 6 3 1 4
7	0 6 3 5 5 3 6 0 7 1 4 2 2 4 1 7	6 0 5 3 3 5 0 6 1 7 2 4 4 2 7 1	3 5 0 6 6 0 5 3 4 2 7 1 1 7 2 4	5 3 6 0 0 6 3 5 2 4 1 7 7 1 4 2	7 1 4 2 2 4 1 7 0 6 3 5 5 3 6 0	5 3 6 0 0 6 3 5 2 4 1 7 7 1 4 2	3 5 0 6 6 0 5 3 4 2 7 1 1 7 2 4	1 7 2 4 4 2 7 1 6 0 5 3 3 5 0 6
8	0 7 6 1 5 2 3 4 3 4 5 2 6 1 0 7	7 0 1 6 2 5 4 3 4 3 2 5 1 6 7 0	6 1 0 7 3 4 5 2 5 2 3 4 0 7 6 1	5 2 3 4 0 7 6 1 6 1 0 7 3 4 5 2	3 4 5 2 6 1 0 7 0 7 6 1 5 2 3 4	1 6 7 0 4 3 2 5 2 5 4 3 7 0 1 6	2 5 4 3 7 0 1 6 1 6 7 0 4 3 2 5	4 3 2 5 1 6 7 0 7 0 1 6 2 5 4 3
9	0 3 6 5 5 6 3 0 7 4 1 2 2 1 4 7	3 0 5 6 6 5 0 3 4 7 2 1 1 2 7 4	6 5 0 3 3 0 5 6 1 2 7 4 4 7 2 1	5 6 3 0 0 3 6 5 2 1 4 7 7 4 1 2	7 4 1 2 2 1 4 7 0 3 6 5 5 6 3 0	5 6 3 0 0 3 6 5 2 1 4 7 7 4 1 2	6 5 0 3 3 0 5 6 1 2 7 4 4 7 2 1	4 7 2 1 1 2 7 4 3 0 5 6 6 5 0 3
10	0 5 6 3 7 2 1 4 3 6 5 0 4 1 2 7	5 0 3 6 2 7 4 1 6 3 0 5 1 4 7 2	6 3 0 5 1 4 7 2 5 0 3 6 2 7 4 1	7 2 1 4 0 5 6 3 4 1 2 7 3 6 5 0	3 6 5 0 4 1 2 7 0 5 6 3 7 2 1 4	3 6 5 0 4 1 2 7 0 5 6 3 7 2 1 4	2 7 4 1 5 0 3 6 1 4 7 2 6 3 0 5	6 3 0 5 1 4 7 2 5 0 3 6 2 7 4 1
11	0 6 7 1 3 5 4 2 5 3 2 4 6 0 1 7	6 0 1 7 5 3 2 4 3 5 4 2 0 6 7 1	7 1 0 6 4 2 3 5 2 4 5 3 1 7 6 0	3 5 4 2 0 6 7 1 6 0 1 7 5 3 2 4	5 3 2 4 6 0 1 7 0 6 7 1 3 5 4 2	1 7 6 0 2 4 5 3 4 2 3 5 7 1 0 6	5 3 2 4 6 0 1 7 0 6 7 1 3 5 4 2	3 5 4 2 0 6 7 1 6 0 1 7 5 3 2 4

Продолжение табл. 4.2.2

	0	1	2	3	4	5	6	7
12	0 7 3 4 6 1 5 2 5 2 6 1 3 4 0 7	7 0 4 3 1 6 2 5 2 5 1 6 4 3 7 0	3 4 0 7 5 2 6 1 6 1 5 2 0 7 3 4	6 1 5 2 0 7 3 4 3 4 0 7 5 2 6 1	5 2 6 1 3 4 0 7 0 7 3 4 6 1 5 2	4 3 7 0 2 5 1 6 1 6 2 5 7 0 4 3	1 6 2 5 7 0 4 3 4 3 7 0 2 5 1 6	2 5 1 6 4 3 7 0 7 0 4 3 1 6 2 5
13	0 3 7 4 6 5 1 2 5 6 2 1 3 0 4 7	3 0 4 7 5 6 2 1 6 5 1 2 0 3 7 4	7 4 0 3 1 2 6 5 2 1 5 6 4 7 3 0	6 5 1 2 0 3 7 4 3 0 4 7 5 6 2 1	5 6 2 1 3 0 4 7 0 3 7 4 6 5 1 2	4 7 3 0 2 1 5 6 1 2 6 5 7 4 0 3	5 6 2 1 3 0 4 7 0 3 7 4 6 5 1 2	6 5 1 2 0 3 7 4 3 0 4 7 5 6 2 1
14	0 5 3 6 6 3 5 0 7 2 4 1 1 4 2 7	5 0 6 3 3 6 0 5 2 7 1 4 4 1 7 2	3 6 0 5 5 0 6 3 4 1 7 2 2 7 1 4	6 3 5 0 0 5 3 6 1 4 2 7 7 2 4 1	7 2 4 1 1 4 2 7 0 5 3 6 6 3 5 0	6 3 5 0 0 5 3 6 1 4 2 7 7 2 4 1	3 6 0 5 5 0 6 3 4 1 7 2 2 7 1 4	2 7 1 4 4 1 7 2 5 0 6 3 3 6 0 5
15	0 6 5 3 7 1 2 4 3 5 6 0 4 2 1 7	6 0 3 5 1 7 4 2 5 3 0 6 2 4 7 1	5 3 0 6 2 4 7 1 6 0 3 5 1 7 4 2	7 1 2 4 0 6 5 3 4 2 1 7 3 5 6 0	3 5 6 0 4 2 1 7 0 6 5 3 7 1 2 4	3 5 6 0 4 2 1 7 0 6 5 3 7 1 2 4	1 7 4 2 6 0 3 5 2 4 7 1 5 3 0 6	5 3 0 6 2 4 7 1 6 0 3 5 1 7 4 2
16	0 7 5 2 6 1 3 4 3 4 6 1 5 2 0 7	7 0 2 5 1 6 4 3 4 3 1 6 2 5 7 0	5 2 0 7 3 4 6 1 6 1 3 4 0 7 5 2	6 1 3 4 0 7 5 2 5 2 0 7 3 4 6 1	3 4 6 1 5 2 0 7 0 7 5 2 6 1 3 4	2 5 7 0 4 3 1 6 1 6 4 3 7 0 2 5	1 6 4 3 7 0 2 5 2 5 7 0 4 3 1 6	4 3 1 6 2 5 7 0 7 0 2 5 1 6 4 3
17	0 3 5 6 6 5 3 0 7 4 2 1 1 2 4 7	3 0 6 5 5 6 0 3 4 7 1 2 2 1 7 4	5 6 0 3 3 0 6 5 2 1 7 4 4 7 1 2	6 5 3 0 0 3 5 6 1 2 4 7 7 4 2 1	7 4 2 1 1 2 4 7 0 3 5 6 6 5 3 0	6 5 3 0 0 3 5 6 1 2 4 7 7 4 2 1	5 6 0 3 3 0 6 5 2 1 7 4 4 7 1 2	4 7 1 2 2 1 7 4 3 0 6 5 5 6 0 3
18	0 5 7 2 6 3 1 4 3 6 4 1 5 0 2 7	5 0 2 7 3 6 4 1 6 3 1 4 0 5 7 2	7 2 0 5 1 4 6 3 4 1 3 6 2 7 5 0	6 3 1 4 0 5 7 2 5 0 2 7 3 6 4 1	3 6 4 1 5 0 2 7 0 5 7 2 6 3 1 4	2 7 5 0 4 1 3 6 1 4 6 3 7 2 0 5	3 6 4 1 5 0 2 7 0 5 7 2 6 3 1 4	6 3 1 4 0 5 7 2 5 0 2 7 3 6 4 1
19	0 6 3 5 7 1 4 2 5 3 6 0 2 4 1 7	6 0 5 3 1 7 2 4 3 5 0 6 4 2 7 1	3 5 0 6 4 2 7 1 6 0 5 3 1 7 2 4	7 1 4 2 0 6 3 5 2 4 1 7 5 3 6 0	5 3 6 0 2 4 1 7 0 6 3 5 7 1 4 2	5 3 6 0 2 4 1 7 0 6 3 5 7 1 4 2	1 7 2 4 6 0 5 3 4 2 7 1 3 5 0 6	3 5 0 6 4 2 7 1 6 0 5 3 1 7 2 4
20	0 7 6 1 3 4 5 2 5 2 3 4 6 1 0 7	7 0 1 6 4 3 2 5 2 5 4 3 1 6 7 0	6 1 0 7 5 2 3 4 3 4 5 2 0 7 6 1	3 4 5 2 0 7 6 1 6 1 0 7 5 2 3 4	5 2 3 4 6 1 0 7 0 7 6 1 3 4 5 2	1 6 7 0 2 5 4 3 4 3 2 5 7 0 1 6	4 3 2 5 7 0 1 6 1 6 7 0 2 5 4 3	2 5 4 3 1 6 7 0 7 0 1 6 4 3 2 5
21	0 3 6 5 7 4 1 2 5 6 3 0 2 1 4 7	3 0 5 6 4 7 2 1 6 5 0 3 1 2 7 4	6 5 0 3 1 2 7 4 3 0 5 6 4 7 2 1	7 4 1 2 0 3 6 5 2 1 4 7 5 6 3 0	5 6 3 0 2 1 4 7 0 3 6 5 7 4 1 2	5 6 3 0 2 1 4 7 0 3 6 5 7 4 1 2	4 7 2 1 3 0 5 6 1 2 7 4 6 5 0 3	6 5 0 3 1 2 7 4 3 0 5 6 4 7 2 1
22	0 5 6 3 3 6 5 0 7 2 1 4 4 1 2 7	5 0 3 6 6 3 0 5 2 7 4 1 1 4 7 2	6 3 0 5 5 0 3 6 1 4 7 2 2 7 4 1	3 6 5 0 0 5 6 3 4 1 2 7 7 2 1 4	7 2 1 4 4 1 2 7 0 5 6 3 3 6 5 0	3 6 5 0 0 5 6 3 4 1 2 7 7 2 1 4	6 3 0 5 5 0 3 6 1 4 7 2 2 7 4 1	2 7 4 1 1 4 7 2 5 0 3 6 6 3 0 5
23	0 6 7 1 5 3 2 4 3 5 4 2 6 0 1 7	6 0 1 7 3 5 4 2 5 3 2 4 0 6 7 1	7 1 0 6 2 4 5 3 4 2 3 5 1 7 6 0	5 3 2 4 0 6 7 1 6 0 1 7 3 5 4 2	3 5 4 2 6 0 1 7 0 6 7 1 5 3 2 4	1 7 6 0 4 2 3 5 2 4 5 3 7 1 0 6	3 5 4 2 6 0 1 7 0 6 7 1 5 3 2 4	5 3 2 4 0 6 7 1 6 0 1 7 3 5 4 2

Продолжение табл. 4.2.2

	0	1	2	3	4	5	6	7
24	7 0 4 3 2 5 1 6 1 6 2 5 4 3 7 0	0 7 3 4 5 2 6 1 6 1 5 2 3 4 0 7	4 3 7 0 1 6 2 5 2 5 1 6 7 0 4 3	2 5 1 6 7 0 4 3 4 3 7 0 1 6 2 5	1 6 2 5 4 3 7 0 7 0 4 3 2 5 1 6	3 4 0 7 6 1 5 2 5 2 6 1 0 7 3 4	5 2 6 1 0 7 3 4 3 4 0 7 6 1 5 2	6 1 5 2 3 4 0 7 0 7 3 4 5 2 6 1
25	7 4 0 3 2 1 5 6 1 2 6 5 4 7 3 0	4 7 3 0 1 2 6 5 2 1 5 6 7 4 0 3	0 3 7 4 5 6 2 1 6 5 1 2 3 0 4 7	2 1 5 6 7 4 0 3 4 7 3 0 1 2 6 5	1 2 6 5 4 7 3 0 7 4 0 3 2 1 5 6	3 0 4 7 6 5 1 2 5 6 2 1 0 3 7 4	1 2 6 5 4 7 3 0 7 4 0 3 2 1 5 6	2 1 5 6 7 4 0 3 4 7 3 0 1 2 6 5
26	7 2 4 1 0 5 3 6 1 4 2 7 6 3 5 0	2 7 1 4 5 0 6 3 4 1 7 2 3 6 0 5	4 1 7 2 3 6 0 5 2 7 1 4 5 0 6 3	0 5 3 6 7 2 4 1 6 3 5 0 1 4 2 7	1 4 2 7 6 3 5 0 7 2 4 1 0 5 3 6	1 4 2 7 6 3 5 0 7 2 4 1 0 5 3 6	5 0 6 3 2 7 1 4 3 6 0 5 4 1 7 2	4 1 7 2 3 6 0 5 2 7 1 4 5 0 6 3
27	7 1 2 4 4 2 1 7 0 6 5 3 3 5 6 0	1 7 4 2 2 4 7 1 6 0 3 5 5 3 0 6	2 4 7 1 1 7 4 2 5 3 0 6 6 0 3 5	4 2 1 7 7 1 2 4 3 5 6 0 0 6 5 3	0 6 5 3 3 5 6 0 7 1 2 4 4 2 1 7	4 2 1 7 7 1 2 4 3 5 6 0 0 6 5 3	2 4 7 1 1 7 4 2 5 3 0 6 6 0 3 5	6 0 3 5 5 3 0 6 1 7 4 2 2 4 7 1
28	7 0 2 5 4 3 1 6 1 6 4 3 2 5 7 0	0 7 5 2 3 4 6 1 6 1 3 4 5 2 0 7	2 5 7 0 1 6 4 3 4 3 1 6 7 0 2 5	4 3 1 6 7 0 2 5 2 5 7 0 1 6 4 3	1 6 4 3 2 5 7 0 7 0 2 5 4 3 1 6	5 2 0 7 6 1 3 4 3 4 6 1 0 7 5 2	3 4 6 1 0 7 5 2 5 2 0 7 6 1 3 4	6 1 3 4 5 2 0 7 0 7 5 2 3 4 6 1
29	7 4 2 1 0 3 5 6 1 2 4 7 6 5 3 0	4 7 1 2 3 0 6 5 2 1 7 4 5 6 0 3	2 1 7 4 5 6 0 3 4 7 1 2 3 0 6 5	0 3 5 6 7 4 2 1 6 5 3 0 1 2 4 7	1 2 4 7 6 5 3 0 7 4 2 1 0 3 5 6	1 2 4 7 6 5 3 0 7 4 2 1 0 3 5 6	0 3 5 6 7 4 2 1 6 5 3 0 1 2 4 7	1 2 4 7 6 5 3 0 7 4 2 1 0 3 5 6
30	7 2 0 5 4 1 3 6 1 4 6 3 2 7 5 0	2 7 5 0 1 4 6 3 4 1 3 6 7 2 0 5	0 5 7 2 3 6 4 1 6 3 1 4 5 0 2 7	4 1 3 6 7 2 0 5 2 7 5 0 1 4 6 3	1 4 6 3 2 7 5 0 7 2 0 5 4 1 3 6	5 0 2 7 6 3 1 4 3 6 4 1 0 5 7 2	1 4 6 3 2 7 5 0 7 2 0 5 4 1 3 6	4 1 3 6 7 2 0 5 2 7 5 0 1 4 6 3
31	7 1 4 2 2 4 1 7 0 6 3 5 5 3 6 0	1 7 2 4 4 2 7 1 6 0 5 3 3 5 0 6	4 2 7 1 1 7 2 4 3 5 0 6 6 0 5 3	2 4 1 7 7 1 4 2 5 3 6 0 0 6 3 5	0 6 3 5 5 3 6 0 7 1 4 2 2 4 1 7	2 4 1 7 7 1 4 2 5 3 6 0 0 6 3 5	4 2 7 1 1 7 2 4 3 5 0 6 6 0 5 3	6 0 5 3 3 5 0 6 7 1 4 2 2 4 1 7
32	7 0 1 6 2 5 4 3 4 3 2 5 1 6 7 0	0 7 6 1 5 2 3 4 3 4 5 2 6 1 0 7	1 6 7 0 4 3 2 5 2 5 4 3 7 0 1 6	2 5 4 3 7 0 1 6 1 6 7 0 4 3 2 5	4 3 2 5 1 6 7 0 7 0 1 6 2 5 4 3	6 1 0 7 3 4 5 2 5 2 3 4 0 7 6 1	5 2 3 4 0 7 6 1 6 1 0 7 3 4 5 2	3 4 5 2 6 1 0 7 0 7 6 1 5 2 3 4
33	7 4 1 2 2 1 4 7 0 3 6 5 5 6 3 0	4 7 2 1 1 2 7 4 3 0 5 6 6 5 0 3	1 2 7 4 4 7 2 1 6 5 0 3 3 0 5 6	2 1 4 7 7 4 1 2 5 6 3 0 0 3 6 5	0 3 6 5 5 6 3 0 7 4 1 2 2 1 4 7	2 1 4 7 7 4 1 2 5 6 3 0 0 3 6 5	1 2 7 4 4 7 2 1 6 5 0 3 3 0 5 6	3 0 5 6 6 5 0 3 7 4 1 2 2 1 4 7
34	7 2 1 4 0 5 6 3 4 1 2 7 3 6 5 0	2 7 4 1 5 0 3 6 1 4 7 2 6 3 0 5	1 4 7 2 6 3 0 5 2 7 4 1 5 0 3 6	0 5 6 3 7 2 1 4 3 6 5 0 4 1 2 7	4 1 2 7 3 6 5 0 7 2 1 4 0 5 6 3	4 1 2 7 3 6 5 0 7 2 1 4 0 5 6 3	5 0 3 6 2 7 4 1 6 3 0 5 1 4 7 2	1 4 7 2 6 3 0 5 2 7 4 1 5 0 3 6
35	7 1 0 6 4 2 3 5 2 4 5 3 1 7 6 0	1 7 6 0 2 4 5 3 4 2 3 5 7 1 0 6	0 6 7 1 3 5 4 2 5 3 2 4 6 0 1 7	4 2 3 5 7 1 0 6 1 7 6 0 2 4 5 3	2 4 5 3 1 7 6 0 7 1 0 6 4 2 3 5	6 0 1 7 5 3 2 4 3 5 4 2 0 6 7 1	2 4 5 3 1 7 6 0 7 1 0 6 4 2 3 5	4 2 3 5 7 1 0 6 1 7 6 0 2 4 5 3

Окончание табл. 4.2.2

	0	1	2	3	4	5	6	7
36	7 0 4 3	0 7 3 4	4 3 7 0	1 6 2 5	2 5 1 6	3 4 0 7	6 1 5 2	5 2 6 1
	1 6 2 5	6 1 5 2	2 5 1 6	7 0 4 3	4 3 7 0	5 2 6 1	0 7 3 4	3 4 0 7
	2 5 1 6	5 2 6 1	1 6 2 5	4 3 7 0	7 0 4 3	6 1 5 2	3 4 0 7	0 7 3 4
	4 3 7 0	3 4 0 7	7 0 4 3	2 5 1 6	1 6 2 5	0 7 3 4	5 2 6 1	6 1 5 2
37	7 4 0 3	4 7 3 0	0 3 7 4	1 2 6 5	2 1 5 6	3 0 4 7	2 1 5 6	1 2 6 5
	1 2 6 5	2 1 5 6	6 5 1 2	7 4 0 3	4 7 3 0	5 6 2 1	4 7 3 0	7 4 0 3
	2 1 5 6	1 2 6 5	5 6 2 1	4 7 3 0	7 4 0 3	6 5 1 2	7 4 0 3	4 7 3 0
	4 7 3 0	7 4 0 3	3 0 4 7	2 1 5 6	1 2 6 5	0 3 7 4	1 2 6 5	2 1 5 6
38	7 2 4 1	2 7 1 4	4 1 7 2	1 4 2 7	0 5 3 6	1 4 2 7	4 1 7 2	5 0 6 3
	1 4 2 7	4 1 7 2	2 7 1 4	7 2 4 1	6 3 5 0	7 2 4 1	2 7 1 4	3 6 0 5
	0 5 3 6	5 0 6 3	3 6 0 5	6 3 5 0	7 2 4 1	6 3 5 0	3 6 0 5	2 7 1 4
	6 3 5 0	3 6 0 5	5 0 6 3	0 5 3 6	1 4 2 7	0 5 3 6	5 0 6 3	4 1 7 2
39	7 1 2 4	1 7 4 2	2 4 7 1	0 6 5 3	4 2 1 7	4 2 1 7	2 4 7 1	6 0 3 5
	0 6 5 3	6 0 3 5	5 3 0 6	7 1 2 4	3 5 6 0	3 5 6 0	5 3 0 6	1 7 4 2
	4 2 1 7	2 4 7 1	1 7 4 2	3 5 6 0	7 1 2 4	7 1 2 4	1 7 4 2	5 3 0 6
	3 5 6 0	5 3 0 6	6 0 3 5	4 2 1 7	0 6 5 3	0 6 5 3	6 0 3 5	2 4 7 1
40	7 0 2 5	0 7 5 2	2 5 7 0	1 6 4 3	4 3 1 6	5 2 0 7	6 1 3 4	3 4 6 1
	1 6 4 3	6 1 3 4	4 3 1 6	7 0 2 5	2 5 7 0	3 4 6 1	0 7 5 2	5 2 0 7
	4 3 1 6	3 4 6 1	1 6 4 3	2 5 7 0	7 0 2 5	6 1 3 4	5 2 0 7	0 7 5 2
	2 5 7 0	5 2 0 7	7 0 2 5	4 3 1 6	1 6 4 3	0 7 5 2	3 4 6 1	6 1 3 4
41	7 4 2 1	4 7 1 2	2 1 7 4	1 2 4 7	0 3 5 6	1 2 4 7	2 1 7 4	3 0 6 5
	1 2 4 7	2 1 7 4	4 7 1 2	7 4 2 1	6 5 3 0	7 4 2 1	4 7 1 2	5 6 0 3
	0 3 5 6	3 0 6 5	5 6 0 3	6 5 3 0	7 4 2 1	6 5 3 0	5 6 0 3	4 7 1 2
	6 5 3 0	5 6 0 3	3 0 6 5	0 3 5 6	1 2 4 7	0 3 5 6	3 0 6 5	2 1 7 4
42	7 2 0 5	2 7 5 0	0 5 7 2	1 4 6 3	4 1 3 6	5 0 2 7	4 1 3 6	1 4 6 3
	1 4 6 3	4 1 3 6	6 3 1 4	7 2 0 5	2 7 5 0	3 6 4 1	2 7 5 0	7 2 0 5
	4 1 3 6	1 4 6 3	3 6 4 1	2 7 5 0	7 2 0 5	6 3 1 4	7 2 0 5	2 7 5 0
	2 7 5 0	7 2 0 5	5 0 2 7	4 1 3 6	1 4 6 3	0 5 7 2	1 4 6 3	4 1 3 6
43	7 1 4 2	1 7 2 4	4 2 7 1	0 6 3 5	2 4 1 7	2 4 1 7	6 0 5 3	4 2 7 1
	0 6 3 5	6 0 5 3	3 5 0 6	7 1 4 2	5 3 6 0	5 3 6 0	1 7 2 4	3 5 0 6
	2 4 1 7	4 2 7 1	1 7 2 4	5 3 6 0	7 1 4 2	7 1 4 2	3 5 0 6	1 7 2 4
	5 3 6 0	3 5 0 6	6 0 5 3	2 4 1 7	0 6 3 5	0 6 3 5	4 2 7 1	6 0 5 3
44	7 0 1 6	0 7 6 1	1 6 7 0	4 3 2 5	2 5 4 3	6 1 0 7	3 4 5 2	5 2 3 4
	4 3 2 5	3 4 5 2	2 5 4 3	7 0 1 6	1 6 7 0	5 2 3 4	0 7 6 1	6 1 0 7
	2 5 4 3	5 2 3 4	4 3 2 5	1 6 7 0	7 0 1 6	3 4 5 2	6 1 0 7	0 7 6 1
	1 6 7 0	6 1 0 7	7 0 1 6	2 5 4 3	4 3 2 5	0 7 6 1	5 2 3 4	3 4 5 2
45	7 4 1 2	4 7 2 1	1 2 7 4	0 3 6 5	2 1 4 7	2 1 4 7	3 0 5 6	1 2 7 4
	0 3 6 5	3 0 5 6	6 5 0 3	7 4 1 2	5 6 3 0	5 6 3 0	4 7 2 1	6 5 0 3
	2 1 4 7	1 2 7 4	4 7 2 1	5 6 3 0	7 4 1 2	7 4 1 2	6 5 0 3	4 7 2 1
	5 6 3 0	6 5 0 3	3 0 5 6	2 1 4 7	0 3 6 5	0 3 6 5	1 2 7 4	3 0 5 6
46	7 2 1 4	2 7 4 1	1 4 7 2	4 1 2 7	0 5 6 3	4 1 2 7	1 4 7 2	5 0 3 6
	4 1 2 7	1 4 7 2	2 7 4 1	7 2 1 4	3 6 5 0	7 2 1 4	2 7 4 1	6 3 0 5
	0 5 6 3	5 0 3 6	6 3 0 5	3 6 5 0	7 2 1 4	3 6 5 0	6 3 0 5	2 7 4 1
	3 6 5 0	6 3 0 5	5 0 3 6	0 5 6 3	4 1 2 7	0 5 6 3	5 0 3 6	1 4 7 2
47	7 1 0 6	1 7 6 0	0 6 7 1	2 4 5 3	4 2 3 5	6 0 1 7	4 2 3 5	2 4 5 3
	2 4 5 3	4 2 3 5	5 3 2 4	7 1 0 6	1 7 6 0	3 5 4 2	1 7 6 0	7 1 0 6
	4 2 3 5	2 4 5 3	3 5 4 2	1 7 6 0	7 1 0 6	5 3 2 4	7 1 0 6	1 7 6 0
	1 7 6 0	7 1 0 6	6 0 1 7	4 2 3 5	2 4 5 3	0 6 7 1	2 4 5 3	4 2 3 5

Обратимся, например, к коду с координатами 15, 0 табл. 4.2.1, 4.2.2 и представим геометрические образы исправленных разрядов $a'_1 - a'_4$ этого кода в четырехмерном пространстве координат $a_1 - a_4$.

Координаты кода 15,0

0	6	5	3
7	1	2	4
3	5	6	0
4	2	1	7

a'_1

6	$\underline{0}$	3	$\underline{5}$
1	$\underline{7}$	4	$\underline{2}$
5	$\underline{3}$	0	$\underline{6}$
2	$\underline{4}$	7	$\underline{1}$

a'_2

5	3	$\underline{0}$	$\underline{6}$
2	4	$\underline{7}$	$\underline{1}$
6	0	$\underline{3}$	$\underline{5}$
1	7	$\underline{4}$	$\underline{2}$

a'_3

7	1	2	4
$\underline{0}$	$\underline{6}$	$\underline{5}$	$\underline{3}$
4	2	1	7
$\underline{3}$	$\underline{5}$	$\underline{6}$	$\underline{0}$

a'_4

3	5	6	0
4	2	1	7
$\underline{0}$	$\underline{6}$	$\underline{5}$	$\underline{3}$
$\underline{7}$	$\underline{1}$	$\underline{2}$	$\underline{4}$

В ячейках этого пространства записаны эквивалентные цифровые сигналы 0_x – 7_x контрольных разрядов x_1 – x_3 систематического кода АХ и инверсии этих цифр $\underline{0}_x$ – $\underline{7}_x$.

Учитывая, что в представленных зависимостях значения цифровых множеств в клетках пространства подчиняются простым соотношениям $\underline{i} \supset j$ ($\underline{i} \neq j$), логические выражения для исправленных информационных разрядов этого кода могут быть записаны в двухуровневом исполнении следующим образом:

$$
\begin{aligned}
a'_1 = {} & 6_x\,\underline{a}_3\,\underline{a}_4\,\underline{a}_2 \vee 3_x\,\underline{a}_3\,\underline{a}_4\,a_2 \vee \underline{0}_x\,\underline{a}_3\,\underline{a}_4\,a_1\,\underline{a}_2 \vee \underline{5}_x\,\underline{a}_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 1_x\,a_3\,\underline{a}_4\,\underline{a}_2 \vee 4_x\,a_3\,\underline{a}_4\,a_2 \vee \underline{7}_x\,a_3\,\underline{a}_4\,a_1\,\underline{a}_2 \vee \underline{2}_x\,a_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 5_x\,\underline{a}_3\,a_4\,\underline{a}_2 \vee 0_x\,\underline{a}_3\,a_4\,a_2 \vee \underline{3}_x\,\underline{a}_3\,a_4\,a_1\,\underline{a}_2 \vee \underline{6}_x\,\underline{a}_3\,a_4\,a_1\,a_2 \vee \\
& \vee 2_x\,a_3\,a_4\,\underline{a}_2 \vee 7_x\,a_3\,a_4\,a_2 \vee \underline{4}_x\,a_3\,a_4\,a_1\,\underline{a}_2 \vee \underline{1}_x\,a_3\,a_4\,a_1\,a_2,
\end{aligned}
$$

$$
\begin{aligned}
a'_2 = {} & 5_x\,\underline{a}_3\,\underline{a}_4\,\underline{a}_1 \vee 3_x\,\underline{a}_3\,\underline{a}_4\,a_1 \vee \underline{0}_x\,\underline{a}_3\,\underline{a}_4\,\underline{a}_1\,a_2 \vee \underline{6}_x\,\underline{a}_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 2_x\,a_3\,\underline{a}_4\,\underline{a}_1 \vee 4_x\,a_3\,\underline{a}_4\,a_1 \vee \underline{7}_x\,a_3\,\underline{a}_4\,\underline{a}_1\,a_2 \vee \underline{1}_x\,a_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 6_x\,\underline{a}_3\,a_4\,\underline{a}_1 \vee 0_x\,\underline{a}_3\,a_4\,a_1 \vee \underline{3}_x\,\underline{a}_3\,a_4\,\underline{a}_1\,a_2 \vee \underline{5}_x\,\underline{a}_3\,a_4\,a_1\,a_2 \vee \\
& \vee 1_x\,a_3\,a_4\,\underline{a}_1 \vee 7_x\,a_3\,a_4\,a_1 \vee \underline{4}_x\,a_3\,a_4\,\underline{a}_1\,a_2 \vee \underline{2}_x\,a_3\,a_4\,a_1\,a_2,
\end{aligned}
$$

$$
\begin{aligned}
a'_3 = {} & 7_x\,\underline{a}_4\,\underline{a}_1\,\underline{a}_2 \vee 1_x\,\underline{a}_4\,a_1\,\underline{a}_2 \vee 2_x\,\underline{a}_4\,\underline{a}_1\,a_2 \vee 4_x\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 4_x\,a_4\,\underline{a}_1\,\underline{a}_2 \vee 2_x\,a_4\,a_1\,\underline{a}_2 \vee 1_x\,a_4\,\underline{a}_1\,a_2 \vee 7_x\,a_4\,a_1\,a_2 \vee \\
& \vee \underline{0}_x\,a_3\,\underline{a}_4\,\underline{a}_1\,\underline{a}_2 \vee \underline{6}_x\,a_3\,\underline{a}_4\,a_1\,\underline{a}_2 \vee \underline{5}_x\,a_3\,\underline{a}_4\,\underline{a}_1\,a_2 \vee \underline{3}_x\,a_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee \underline{3}_x\,a_3\,a_4\,\underline{a}_1\,\underline{a}_2 \vee \underline{5}_x\,a_3\,a_4\,a_1\,\underline{a}_2 \vee \underline{6}_x\,a_3\,a_4\,\underline{a}_1\,a_2 \vee \underline{0}_x\,a_3\,a_4\,a_1\,a_2,
\end{aligned}
$$

$$
\begin{aligned}
a'_4 = {} & 3_x\,\underline{a}_3\,\underline{a}_1\,\underline{a}_2 \vee 1_x\,\underline{a}_4\,a_1\,\underline{a}_2 \vee 2_x\,\underline{a}_4\,\underline{a}_1\,a_2 \vee 4_x\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee 4_x\,a_3\,\underline{a}_1\,\underline{a}_2 \vee 2_x\,a_4\,a_1\,\underline{a}_2 \vee 1_x\,a_4\,\underline{a}_1\,a_2 \vee 7_x\,a_4\,a_1\,a_2 \vee \\
& \vee \underline{0}_x\,\underline{a}_3\,a_4\,\underline{a}_1\,\underline{a}_2 \vee \underline{6}_x\,a_3\,\underline{a}_4\,a_1\,\underline{a}_2 \vee \underline{5}_x\,a_3\,\underline{a}_4\,\underline{a}_1\,a_2 \vee \underline{3}_x\,a_3\,\underline{a}_4\,a_1\,a_2 \vee \\
& \vee \underline{7}_x\,a_3\,a_4\,\underline{a}_1\,\underline{a}_2 \vee \underline{1}_x\,a_3\,a_4\,a_1\,\underline{a}_2 \vee \underline{2}_x\,a_3\,a_4\,\underline{a}_1\,a_2 \vee \underline{4}_x\,a_3\,a_4\,a_1\,a_2.
\end{aligned}
$$

Из геометрических образов этих логических функций реализация многоуровневой схемы вполне очевидна и здесь приводиться не будет.

Аналогичные выражения могут быть получены для всех кодов, представленных в табл. 4.2.2, но все они при этом различны и даже в пределах одного кода геометрические образы сигналов информационных разрядов a'_1 – a'_4 не могут мысленными поворотами относительно осей симметрии этого пространства быть совмещены, но при этом необходимо отметить, что если исключить инверсии цифр сигналов контрольных разрядов в геометрических образах сигналов исправленных информационных разрядов a'_1 – a'_4 , то видоизмененные таким образом геометрические фигуры могут переходить одна в другую мыс-

ленными поворотами относительно осей симметрии пространства, на примере кода с координатами 15, 0, по следующему алгоритму:

$$a_1\ a_2\ a_3\ a_4 \begin{array}{|cccc|} 0 & 6 & 5 & 3 \\ 7 & 1 & 2 & 4 \\ 3 & 5 & 6 & 0 \\ 4 & 2 & 1 & 7 \end{array}$$

$$\text{Поворот}\ \underline{a_1}\ a_2\ a_3\ a_4 \begin{array}{|cccc|} 6 & 0 & 3 & 5 \\ 1 & 7 & 4 & 2 \\ 5 & 3 & 0 & 6 \\ 2 & 4 & 7 & 1 \end{array}\ \text{Поворот}\ a_1\ \underline{a_2}\ a_3\ a_4 \begin{array}{|cccc|} 5 & 3 & 0 & 6 \\ 2 & 4 & 7 & 1 \\ 6 & 0 & 3 & 5 \\ 1 & 7 & 4 & 2 \end{array}\ \text{Поворот}\ a_1\ a_2\ \underline{a_3}\ a_4 \begin{array}{|cccc|} 7 & 1 & 2 & 4 \\ 0 & 6 & 5 & 3 \\ 4 & 2 & 1 & 7 \\ 3 & 5 & 6 & 0 \end{array}\ \text{Поворот}\ a_1\ a_2\ a_3\ \underline{a_4} \begin{array}{|cccc|} 3 & 5 & 6 & 0 \\ 4 & 2 & 1 & 7 \\ 0 & 6 & 5 & 3 \\ 7 & 1 & 2 & 4 \end{array}.$$

Из этого геометрического представления непосредственно синтезируется структурная схема для параллельного исправления всех одиночных ошибок в информационных разрядах системы счисления основания n = 16 (рис. 4.2.1).

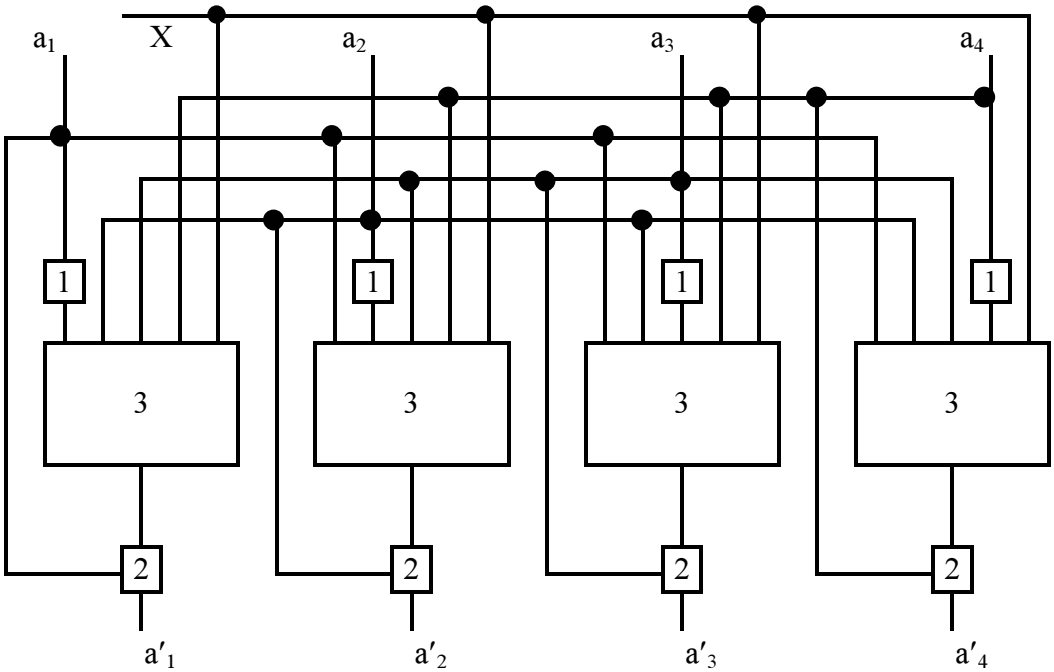


Рис. 4.2.1

В этой схеме номерами *1* и *2* обозначены элементарные инверторы – соответственно простой и управляемый, а номером *3* – логический блок, функционально определяемый таблично-матричной формой представления любого из

кодов табл. 4.2.2, который на примере кода с координатами 15, 0 может быть записан следующим образом:

$$
\begin{aligned}
C = {} & 0_a\, 0_x \vee \quad 1_a\, 6_x \vee \quad 2_a\, 5_x \vee \quad 3_a\, 3_x \vee \\
\vee {} & 4_a\, 7_x \vee \quad 5_a\, 1_x \vee \quad 6_a\, 2_x \vee \quad 7_a\, 4_x \vee \\
\vee {} & 8_a\, 3_x \vee \quad 9_a\, 5_x \vee \quad 10_a\, 6_x \vee \quad 11_a\, 0_x \vee \\
\vee {} & 12_a\, 4_x \vee \quad 13_a\, 2_x \vee \quad 14_a\, 1_x \vee \quad 15_a\, 7_x .
\end{aligned}
\qquad (4.2.1)
$$

В этом выражении вторые составляющие логических произведений являются цифровыми сигналами контрольной части кода, взятыми из табл. 4.2.2, а первые составляющие – сигналами информационной части этого систематического кода.

Из 192 совершенных кодов табл. 4.2.2 можно выделить 16 суперсовершенных кодов, которые полностью попадают под это определение: в ячейках многомерного цифрового пространства координат кода располагаются только цифровые сигналы определенной одиночной кратности ошибок, а в каждой такой ячейке находится цифра с одной ошибкой этой кратности; многомерное цифровое пространство используется для размещения однократных ошибок на 100 %; исправление ошибок контрольных разрядов кода выполняется логическими схемами, которые исправляют ошибки в информационных разрядах совершенного кода посредством соответствующего изменения входных сигналов, осуществляемых мысленными поворотами относительно осей симметрии этого цифрового пространства. Это коды с координатами таблицы 15, 0; 15, 3; 3, 0; 3, 3; 10, 0; 10, 3; 22, 0; 22, 3; 15, 4; 39, 5; 3, 4; 27, 3; 10, 4; 34, 4; 22, 4; 46, 3.

Первый из этих кодов был исследован нами выше, где было показано, что логические схемы, исправляющие ошибки в разрядах информационной части систематического кода, могут быть использованы также для исправления аналогичных ошибок в разрядах его контрольной части. При последовательной обработке систематического кода, когда первоначально выполняется, например, исправление его ошибок в информационной части с последующей передачей исправленных сигналов в регистр памяти, на втором этапе исправления ошибок эти же логические схемы, при соответствующем изменении сигналов на их входных шинах, позволяют исправить ошибки в контрольных разрядах с передачей их исправленных сигналов также в регистр памяти.

В остальных типах совершенных кодов табл. 4.2.2 этого выполнить не удается: информационные и контрольные разряды систематического кода будут обрабатываться разными принципиальными логическими схемами.

Пример 3. Синтез одноразрядного сумматора (**A** ± **B**) с исправлением одиночных ошибок в системе счисления основания n = 16

Из теоретических исследований второй главы вытекает, что алгоритм построения геометрического образа одноразрядной арифметической функции для операций сложения и вычитания весьма прост и даже не требует составления исходной таблицы результата этой операции в цифровых данных.

Для его построения необходимо иметь только геометрические образы сигналов кода операндов, которые заполняют собой первый столбец или первую строку двухмерного цифрового пространства посредством размещения в его ячейках символов, соответствующих логической единице (1* или *). Дальнейшее заполнение остальных ячеек этого цифрового пространства такими же символами осуществляется для операции суммирования, рассматривая, например, заполнение строк путем сдвига этих строк сверху вниз при одновременном синхронном смещении символов справа налево, а для операции вычитания также сдвигом строк сверху вниз, но при смещении этих символов в противоположном направлении – слева направо.

При всех этих сдвигах символов предполагается циклическая замкнутость цифровых значений сигналов разряда операндов, т.е. циклическая замкнутость символов каждой строки и столбца.

Предыдущий пример устанавливает зависимость между кодовыми комбинациями информационной (**0** – **15**) и контрольной (0–7) частями систематического кода всех его типов, исправляющих одиночные ошибки для основания системы счисления n = 16, и представляет алгоритм получения геометрических образов исправленных сигналов разрядов $a'_1 - a'_4$, покрытие которых определяет функциональную схему декодирующего устройства. Каждый из этих геометрических образов задается двухмерной таблицей, где в информационных ячейках пространства соответственно под номерами **0**–**15** размещаются прямые и инверсные значения контрольных кодовых комбинаций под номерами 0–7.

При этом прямые и инверсные значения контрольных кодовых комбинаций 0–7 могут рассматриваться нами как геометрические образы логических функций в координатах разрядов контрольной части систематического кода либо представляются с позиций многозначной логики, где в качестве логических сигналов многозначной логики выступают геометрические образы контрольных кодовых комбинаций систематического кода.

Структурная схема одноразрядного сумматора с исправлением одиночных ошибок (рис. 4.3.1) может быть реализована путем установки декодирующих блоков 1, 2 на входных шинах операндов, которые представляются в систематическом коде, исправляющем одиночные ошибки операндов **AX, BY** с последующим выполнением операции суммирования либо вычитания в двух сумматорах раздельно для информационной и контрольной частей этого кода. На выходных шинах этих сумматоров устанавливается аналогичный декодирующий блок 3, исправляющий ошибки операций суммирования, выходной сигнал которого в систематическом коде с информационной $\mathbf{C''}(c_1 - c_4)$ и контрольной $\mathbf{Z''}(z_1 - z_3)$, если она необходима, частями поступают на выходные шины уст-

ройства. Отдельно на этой схеме представлен блок 4 для формирования сигнала переноса в следующий старший разряд по известному из первой главы логическому выражению:

$$P_n = a'_4\, b'_4 \vee c''_4\, \underline{b}'_4 \vee c''_4\, \underline{a}'_4,$$

где все составляющие сигналов разрядов в этой зависимости являются непрерывными множествами цифр **8** $\vee \ldots \vee$ **15**.

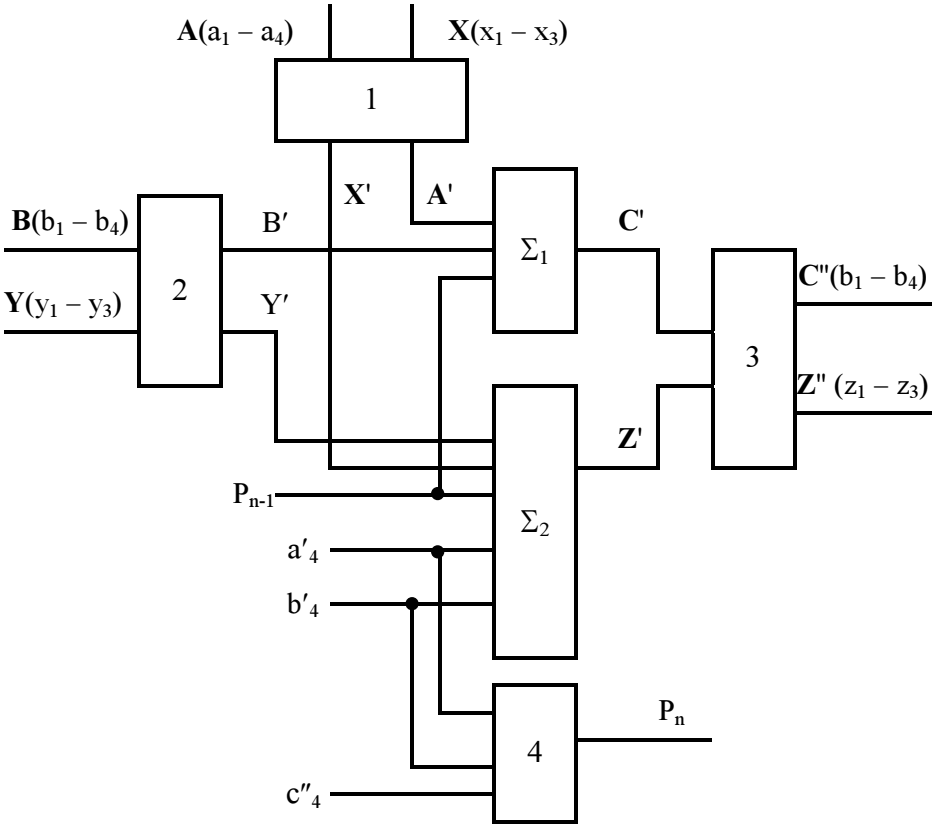


Рис. 4.3.1

Для систематического кода, где информационная часть представляется в основном двоичном коде, выполнение сумматора Σ_1 общеизвестно, а второй сумматор Σ_2 для контрольной части систематического кода может быть синтезирован на основании данных табл. 4.1.2. Пусть это будет совершенный код с координатами 15, 0. Тогда контрольные разряды операндов этого кода $x_1 - x_3$, $y_1 - y_3$, совместно с сигналами информационных разрядов операндов a'_4 и b'_4 составят полное основание системы счисления n = 16 (цифры **0** – **15**).

На примере кода первого операнда на рис. 4.3.2 приведены номера *0–15* весовых кодовых комбинаций, сигналы разрядов x'_1, x'_2, x'_3, a'_4 (для второго операнда это будут сигналы y'_1, y'_2, y'_3, b'_4) и соответствующие им цифровые сигналы **0**–**15** этого основания системы счисления.

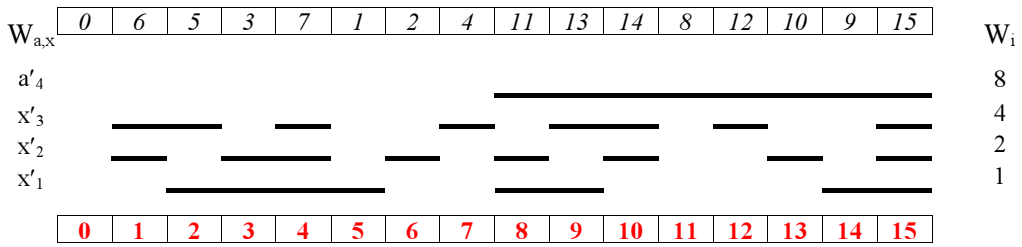


Рис. 4.3.2

Следующий этап создания устройства сводится к синтезу двухвходового сумматора, использующего этот код, что не представляет каких-либо трудностей.

Здесь необходимо отметить, что декодирующие блоки *1*, *2* в схеме рис. 4.3.1 неполные, поскольку предназначены только для исправления одиночных ошибок в старших разрядах операндов **A, B**, выходные сигналы которых a'_4, b'_4 применяются в блоке 4 формирования сигнала переноса P_n. Также в схеме рис. 4.3.1 можно осуществить дальнейшее уменьшение оборудования. Оно может заключаться в отказе от классического представления систематического кода, которое состоит в том, что его информационная часть всегда выполняется в основном двоичном коде. Такой вариант построения систематических кодов, предназначенных для устройств машинной арифметики, никем не рассматривался, поскольку ошибочно считалось, что другие типы двоичных кодов не арифметические.

Вместе с тем многие исследователи в течение десятилетий проводили поиск решения задачи исправления ошибок применением для устройств машинной арифметики остаточных либо циклических AN кодов, которые также являются систематическими в классическом понимании этого типа кода. Поскольку в таких типах кодов арифметические операции должны выполняться параллельно над исходными операндами и их наименьшими вычетами по некоторым выбранным модулям, то это равносильно, по нашему мнению, требованию применения совершенного основного двоичного кода не только в информационной части кода, но одновременно и в контрольной, что не выполнимо. Этим обстоятельством можно объяснить все неудачи таких поисков.

Необходимый синхронизм двух составных частей систематического кода, предназначенного для исправления ошибок в устройствах машинной арифметики, может быть реализован в двух вариантах:

а) использованием в его контрольной части основного совершенного двоичного кода, число разрядов которого всегда меньше информационной части систематического кода, где она будет иной двоичной структурой, в зависимости от выбранного алгоритма исправления определенного типа ошибок;

б) при равенстве числа разрядов информационной и контрольной частей систематического кода, когда как при классическом варианте выполнения систематического кода, так и его нетрадиционном варианте одна из этих частей

либо обе части одновременно будут в коде, отличном от основной двоичной структуры.

Все 192 классических варианта систематического кода табл. 4.2.2, исправляющих все одиночные ошибки, когда информационная часть такого кода выполняется в системе счисления основания n = 16, могут быть выбраны для синтеза нетрадиционных систематических кодов, в том числе и для кодов, контрольная часть которых может быть представлена в основном совершенном двоичном коде.

Совершенный код с координатами 15, 0 этой таблицы, как отмечалось ранее, является оптимальным кодом для операции исправления одиночных ошибок. Представим этот код в соотношениях его весовых комбинаций. Для этого в ячейках цифрового пространства от **0** до **15** (рис. 4.3.3, *а*) запишем в виде дроби эти соотношения: числитель дроби определяет весовой кодовый номер информационной части кода, а знаменатель – весовой кодовый номер его контрольной части.

0/0	*1/6*	*2/5*	*3/3*
4/7	*5/1*	*6/2*	*7/4*
8/3	*9/5*	*10/6*	*11/0*
12/4	*13/2*	*14/1*	*15/7*

а)

→

0/0	*5/1*	*6/2*	*3/3*
7/4	*2/5*	*1/6*	*4/7*
11/0	*14/1*	*13/2*	*8/3*
12/4	*9/5*	*10/6*	*15/7*

б)

Рис. 4.3.3

В классическом варианте систематического кода номера ячеек цифрового пространства (цифры от **0** до **15**) совпадают с весовыми номерами числителя дроби. Именно этот вариант представлен на рис. 4.3.3, *а*.

Для выполнения контрольной части систематического кода в основном двоичном варианте необходимо произвести перестановку значения соотношения весовых номеров таким образом, чтобы при последовательном прохождении ячеек цифрового пространства от **0** до **15** весовые значения знаменателя изменялись от *0* до *7,* и этот цикл изменения, очевидно, произойдет два раза. Один из вариантов такой перестановки представлен на рис. 4.3.3, *б*, который позволяет графически изобразить (рис. 4.3.4) зависимости сигналов разрядов информационной части кода a_1 – a_4 в соотношении цифр от **0** до **15** основания системы счисления и весовых значений его кодовых комбинаций от *0* до *15*.

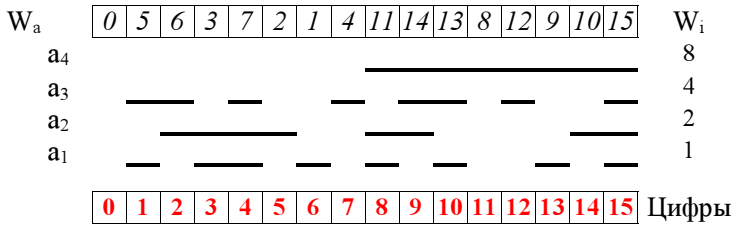


Рис. 4.3.4

Число таких кодов велико, но лучшими из них являются такие, где один из его разрядов информационной части является непрерывным множеством всех цифр второй половины основания системы счисления. Именно такой код изображен на рис. 4.3.4, где a_4 = **8** $\vee \ldots \vee$ **15**.

Аналогичная процедура синтеза может быть выполнена с любым из кодов табл. 4.2.2, но только суперсовершенные коды этой таблицы будут рассматриваться в дальнейшем. При этом необходимо отметить, что все декодирующие устройства остаются неизменными для всех синтезированных систематических кодов, в основу синтеза которых взяты данные этой таблицы.

Принимая сигналы разрядов кода с координатами 15, 0 в качестве базовых, представим все сигналы разрядов суперсовершенных кодов в обозначениях этого кода:

15, 0 (a_1, a_2, a_3, a_4)	15, 3 (a_1, a_2, $\underline{a}_3$, a_4)	22, 3 ($\underline{a}_2$, $\underline{a}_1$, a_4, a_3)	3, 4 (a_1, a_2, a_4, $\underline{a}_3$)
10, 4 ($\underline{a}_2$, $\underline{a}_1$, a_3, a_4)	15, 4 ($\underline{a}_1$, $\underline{a}_2$, a_3, a_4)	3, 0 (a_1, a_2, a_4, a_3)	10, 0 (a_2, a_1, a_3, a_4)
22, 0 (a_2, a_1, a_4, a_3)	3, 3 ($\underline{a}_1$, $\underline{a}_2$, a_4, a_3)	10, 3 (a_2, a_1, $\underline{a}_3$, a_4)	22, 4 (a_2, a_1, a_4, $\underline{a}_3$)
27, 3 ($\underline{a}_1$, $\underline{a}_2$, a_4, $\underline{a}_3$)	46, 3 ($\underline{a}_2$, $\underline{a}_1$, a_4, $\underline{a}_3$)	34, 4($\underline{a}_2$, $\underline{a}_1$, $\underline{a}_3$, a_4)	39, 5 ($\underline{a}_1$, $\underline{a}_2$, $\underline{a}_3$, a_4)

Поскольку все синтезированные здесь коды состоят из одинаковых составляющих, определяющих сигналы их разрядов, которые только меняются в них местами либо при этом дополнительно инвертируются, то вполне очевидно, что все они равноценны для синтеза суммирующих устройств.

Вопрос, какой из этих кодов либо все являются совершенными с точки зрения минимальных аппаратурных затрат при предельном быстродействии для других арифметических операций, остается открытым и может быть решен путем перебора вариантов.

Продолжим синтез суммирующего устройства при использовании кода, представленного на рис. 4.3.4. Опуская известные этапы построения геометрических образов выходных сигналов для четырех режимов его работы в цифровых координатах операндов, а также промежуточный последовательный переход к координатам операндов в кодовых комбинациях основного двоичного кода, представим эти результаты построения сразу непосредственно в координатах операндов кодовых комбинаций основного двоичного кода (рис. 4.3.5, *а – г*) соответственно для этих режимов работы.

Очевидно, принципиально можно реализовать покрытие всех 16 геометрических образов выходных сигналов и определить, например, их двухуровневые логические выражения $F(c_1 - c_4)$, $F'(c'_1 - c'_4)$, F'' $(c''_1 - c''_4)$, F''' $(c'''_1 - c'''_4)$ выходных сигналов сумматора для четырех режимов работы, которые пусть определяются сигналами α_1[**C** = **A** + **B**, P_{n-1}= 0*], α_2[**C** = **A** + **B**, P_{n-1}= 1*], α_3[**C** = **A** – **B**, Z_{n-1}= 0*], α_4[**C** = **A** – **B**, Z_{n-1}= 1*]. Тогда общее логическое выражение, определяющее построение сумматора, $L = F\,\alpha_1 \vee F'\alpha_2 \vee F''\alpha_3 \vee F'''\alpha_4$, что соответствует максимальному быстродействию сумматора, но при значительных аппаратурных затратах.

$P_{n-1}=0*$ $\qquad$ $P_{n-1}=1*$

c_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		*		*		*		*		*		*		*		*
1	*				*		*	*	*	*	*				*	
2			*	*		*					*	*	*	*		*
3	*		*	*	*	*					*		*	*		
4		*		*		*				*		*	*	*		*
5	*		*	*	*		*				*		*		*	
6		*				*	*	*	*	*		*				*
7	*	*					*	*	*	*					*	*
8		*					*	*	*	*		*			*	*
9	*	*			*		*	*	*	*					*	
10		*	*	*		*						*	*	*		*
11	*		*		*		*		*		*		*		*	
12			*	*	*	*					*	*	*	*		
13	*		*	*	*						*		*	*	*	
14		*				*		*	*	*		*		*		*
15	*		*		*		*	*	*		*				*	

c'_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*		*	*	*		*				*		*		*	
1		*		*		*				*		*	*	*		*
2	*				*		*	*	*	*	*				*	
3	*	*					*	*	*	*					*	*
4	*		*		*		*		*		*		*		*	
5		*				*	*	*	*	*		*				*
6	*		*	*	*	*					*		*	*		
7			*	*		*					*	*	*	*		*
8			*	*	*	*					*	*	*	*		
9		*	*	*		*						*	*	*		*
10	*		*		*		*	*	*		*				*	
11		*				*		*	*	*		*		*		*
12	*	*			*		*	*	*	*					*	
13		*					*	*	*	*		*			*	*
14	*		*	*	*						*		*	*	*	
15		*		*		*		*		*		*		*		*

а)

c_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			*	*			*	*			*	*			*	*
1				*			*			*	*	*	*		*	*
2	*			*				*		*	*			*	*	*
3	*	*	*			*	*		*				*			*
4					*	*	*		*	*	*	*	*			
5				*	*	*	*	*		*	*	*				
6	*	*		*	*	*	*			*			*			
7	*		*			*		*	*		*			*		*
8				*	*			*		*	*	*		*	*	
9		*	*		*	*	*		*			*	*			
10	*	*	*		*	*		*	*					*		
11	*	*			*	*			*	*			*	*		
12		*		*	*		*			*		*	*		*	
13			*					*	*		*	*		*	*	*
14	*	*	*						*				*	*	*	*
15	*	*	*	*				*						*	*	*

c'_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0				*	*	*	*	*		*	*	*				
1					*	*	*		*	*	*	*	*			
2				*			*			*	*	*	*		*	*
3	*		*			*		*	*		*			*		*
4	*	*			*	*			*	*			*	*		
5	*	*		*	*	*	*			*			*			
6	*	*	*			*	*		*				*			*
7	*			*				*		*	*			*	*	*
8		*		*	*		*			*		*	*		*	
9	*	*	*		*	*		*	*					*		
10	*	*	*	*				*						*	*	*
11	*	*	*						*				*	*	*	*
12		*	*		*	*	*		*			*	*			
13				*	*			*		*	*	*		*	*	
14			*					*	*		*	*		*	*	*
15			*	*			*	*			*	*			*	*

б)

c_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					*	*	*	*					*	*	*	*
1		*		*		*		*	*		*		*		*	
2			*		*		*	*		*			*	*		*
3		*			*	*		*			*		*		*	*
4	*		*	*			*		*	*		*		*		
5	*	*		*		*			*		*	*			*	
6	*		*		*		*			*		*		*		*
7	*	*	*	*					*	*	*	*				
8		*			*	*		*			*		*		*	*
9			*		*		*	*		*			*	*		*
10		*		*		*		*	*		*		*		*	
11					*	*	*	*					*	*	*	*
12	*	*	*	*					*	*	*	*				
13	*		*		*		*			*		*		*		*
14	*	*		*		*			*		*	*			*	
15	*		*	*			*		*	*		*		*		

c'_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*	*		*		*			*		*	*			*	
1	*		*	*			*		*	*		*		*		
2		*		*		*		*	*		*		*		*	
3	*	*	*	*					*	*	*	*				
4					*	*	*	*					*	*	*	*
5	*		*		*		*			*		*		*		*
6		*			*	*		*			*		*		*	*
7			*		*		*	*		*			*	*		*
8	*	*	*	*					*	*	*	*				
9		*		*		*		*	*		*		*		*	
10	*		*	*			*		*	*		*		*		
11	*	*		*		*			*		*	*			*	
12			*		*		*	*		*			*	*		*
13		*			*	*		*			*		*		*	*
14	*		*		*		*			*		*		*		*
15					*	*	*	*					*	*	*	*

в)

Рис. 4.3.5 (начало)

c_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0									*	*	*	*	*	*	*	*
1		*	*	*	*		*	*				*			*	
2		*	*	*	*			*				*		*	*	
3		*	*		*				*			*	*	*	*	
4		*	*	*	*	*	*	*				*				
5					*				*	*	*	*	*	*	*	
6		*			*				*	*		*	*	*	*	
7		*	*		*			*	*			*		*	*	
8	*			*		*	*	*		*	*					*
9	*					*	*		*	*	*		*			*
10	*					*			*	*	*		*	*		*
11	*	*	*	*	*	*	*	*								
12	*			*		*	*			*	*		*			*
13	*		*	*		*	*	*			*					*
14	*	*	*	*		*	*	*								*
15	*								*	*	*		*	*	*	*

c'_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					*				*	*	*	*	*	*	*	
1		*	*	*	*	*	*	*				*				
2		*	*	*	*		*	*				*			*	
3		*	*		*			*	*			*		*	*	
4	*	*	*	*	*	*	*	*								
5		*			*				*	*		*	*	*	*	
6		*	*		*				*			*	*	*	*	
7		*	*	*	*			*				*		*	*	
8	*			*		*	*				*		*			*
9	*					*			*	*	*		*	*		*
10	*								*	*	*		*	*	*	*
11	*	*	*	*		*	*	*								*
12	*					*	*		*	*	*		*			*
13	*			*		*	*	*		*	*					*
14	*		*	*		*	*	*			*					*
15									*	*	*	*	*	*	*	*

г)

Рис. 4.3.5 (продолжение)

c''_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		*		*		*		*		*		*		*		*
1	*		*	*	*						*		*	*	*	
2		*					*	*	*	*		*			*	*
3	*	*			*		*	*	*	*					*	
4		*				*		*	*	*		*		*		*
5	*		*		*		*	*	*		*				*	
6		*	*	*		*						*	*	*		*
7			*	*	*	*					*	*	*	*		
8			*	*		*					*	*	*	*		*
9	*		*	*	*	*					*		*	*		
10		*				*	*	*	*	*		*				*
11	*		*		*		*		*		*		*		*	
12	*	*					*	*	*	*					*	*
13	*				*		*	*	*	*	*				*	
14		*		*		*				*		*	*	*		*
15	*		*	*	*		*				*		*		*	

c'''_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*		*		*		*	*	*		*				*	
1		*				*		*	*	*		*		*		*
2	*		*	*	*						*		*	*	*	
3			*	*	*	*					*	*	*	*		
4	*		*		*		*		*		*		*		*	
5		*	*	*		*						*	*	*		*
6	*	*			*		*	*	*	*					*	
7		*					*	*	*	*		*			*	*
8	*	*					*	*	*	*					*	*
9		*				*	*	*	*	*		*				*
10	*		*	*	*		*				*		*		*	
11		*		*		*				*		*	*	*		*
12	*		*	*	*	*					*		*	*		
13			*	*		*					*	*	*	*		*
14	*				*		*	*	*	*	*				*	
15		*		*		*		*		*		*		*		*

а)

c''_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			*	*			*	*			*	*			*	*
1			*					*	*		*	*		*	*	*
2				*	*			*		*	*	*		*	*	
3		*	*		*	*	*		*			*	*			
4	*	*	*						*				*	*	*	*
5	*	*	*	*				*						*	*	*
6	*	*	*		*	*		*	*					*		
7		*		*	*		*			*		*	*		*	
8	*			*				*		*	*			*	*	*
9	*	*	*			*	*		*				*			*
10	*	*		*	*	*	*			*			*			
11	*	*			*	*			*	*			*	*		
12	*		*			*		*	*		*			*		*
13				*			*			*	*	*	*		*	*
14					*	*	*		*	*	*	*	*			
15				*	*	*	*	*		*	*	*				

c'''_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*	*	*	*				*						*	*	*
1	*	*	*						*				*	*	*	*
2			*					*	*		*	*		*	*	*
3		*		*	*		*			*		*	*		*	
4	*	*			*	*			*	*			*	*		
5	*	*	*		*	*		*	*					*		
6		*	*		*	*	*		*			*	*			
7				*	*			*		*	*	*		*	*	
8	*		*			*		*	*		*			*		*
9	*	*		*	*	*	*			*			*			
10				*	*	*	*	*		*	*	*				
11					*	*	*		*	*	*	*	*			
12	*	*	*			*	*		*				*			*
13	*			*				*		*	*	*			*	*
14				*			*			*	*	*	*			*
15			*	*			*	*		*	*	*			*	*

б)

Рис. 4.3.6 (начало)

c''_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0					*	*	*	*					*	*	*	*
1	*		*		*		*			*		*		*		*
2		*			*	*		*			*		*		*	*
3			*		*		*	*		*			*	*		*
4	*	*		*		*			*		*	*			*	
5	*		*	*			*		*	*		*		*		
6		*		*		*		*	*		*		*		*	
7	*	*	*	*					*	*	*	*				
8			*		*		*	*		*			*	*		*
9		*			*	*		*			*		*		*	*
10	*		*		*		*			*		*		*		*
11					*	*	*	*					*	*	*	*
12	*	*	*	*					*	*	*	*				
13		*		*		*		*	*		*		*		*	
14	*		*	*			*		*	*		*		*		
15	*	*		*		*			*		*	*			*	

c'''_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*		*	*			*		*	*		*		*		
1	*	*		*		*			*		*	*			*	
2	*		*		*		*			*		*		*		*
3	*	*	*	*					*	*	*	*				
4					*	*	*	*					*	*	*	*
5		*		*		*		*	*		*		*		*	
6			*		*		*	*		*			*	*		*
7		*			*	*		*			*		*		*	*
8	*	*	*	*					*	*	*	*				
9	*		*		*		*			*		*		*		*
10	*	*		*		*			*		*	*			*	
11	*		*	*			*		*	*		*		*		
12		*			*	*		*			*		*		*	*
13			*		*		*	*		*			*	*		*
14		*		*		*		*	*		*		*		*	
15					*	*	*	*					*	*	*	*

в)

c''_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0									*	*	*	*	*	*	*	*
1	*		*	*		*	*	*			*					*
2	*			*		*	*	*		*	*					*
3	*					*	*		*	*	*		*			*
4	*	*	*	*		*	*	*								*
5	*								*	*	*		*	*	*	*
6	*					*			*	*	*		*	*		*
7	*			*		*	*			*	*		*			*
8		*	*	*	*			*				*		*	*	
9		*	*		*				*			*	*	*	*	
10		*			*				*	*		*	*	*	*	
11	*	*	*	*	*	*	*	*								
12		*	*		*			*	*			*		*	*	
13		*	*	*	*		*	*				*			*	
14		*	*	*	*	*	*	*				*				
15					*				*	*	*	*	*	*	*	

c'''_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	*								*	*	*		*	*	*	*
1	*	*	*	*		*	*	*								*
2	*		*	*		*	*	*			*					*
3	*			*		*	*			*	*		*			*
4	*	*	*	*	*	*	*	*								
5	*					*			*	*	*		*	*		*
6	*					*	*		*	*	*		*			*
7	*			*		*	*	*		*	*					*
8		*	*		*			*	*			*		*	*	
9		*			*				*	*		*	*	*	*	
10					*				*	*	*	*	*	*	*	
11		*	*	*	*	*	*	*				*				
12		*	*		*				*			*	*	*	*	
13		*	*	*	*			*				*		*	*	
14		*	*	*	*		*	*				*			*	
15									*	*	*	*	*	*	*	*

г)

Рис. 4.3.6 (продолжение)

Рассмотрим другой менее затратный вариант синтеза, когда в качестве базовой принята схема, реализующая первый режим работы сумматора. Для этого необходимо определить логические выражения только для четырех сигналов разрядов c_1 – c_4. Выделяя в геометрических образах этих сигналов подмножества, которые определяются сигналами первых двух разрядов операндов a_1, a_2; b_1, b_2 и пронумерованы отдельно для каждого из этих разрядов соответственно номерами 1_1, 2_1, …; … ; 1_4, 2_4, …, представим в координатах основного двоичного кода сигналов a_3, a_4; b_3, b_4 геометрические образы выходных сигналов сумматора для всех четырех режимов его работы (рис. 4.3.6, *а – г*).

Из этого представления видно, что все выходные сигналы сумматора при переходе от одного режима к другому подчиняются одному и тому же закону изменения соответствующих сигналов операндов и выходных сигналов, а именно:

α_1	$c_1[a_3, a_4, b_3, b_4, c_1]$	$c_2[a_3, a_4, b_3, b_4, c_2]$	$c_3[a_3, a_4, b_3, b_4, c_3]$	$c_4[a_3, a_4, b_3, b_4, c_4]$
α_2	$c'_1[\underline{a}_3, a_4, \underline{b}_3, b_4, \underline{c}_1]$	$c'_2[\underline{a}_3, a_4, \underline{b}_3, b_4, \underline{c}_2]$	$c'_3[\underline{a}_3, a_4, \underline{b}_3, b_4, \underline{c}_3]$	$c'_4[\underline{a}_3, a_4, \underline{b}_3, b_4, \underline{c}_4]$
α_3	$c''_1[\underline{a}_3, a_4, b_3, b_4, c_1]$	$c''_2[\underline{a}_3, a_4, b_3, b_4, c_2]$	$c''_3[\underline{a}_3, a_4, b_3, b_4, \underline{c}_3]$	$c''_4[\underline{a}_3, a_4, b_3, b_4, c_4]$
α_4	$c'''_1[a_3, a_4, \underline{b}_3, b_4, \underline{c}_1]$	$c'''_2[a_3, a_4, \underline{b}_3, b_4, \underline{c}_2]$	$c'''_2[a_3, a_4, \underline{b}_3, b_4, c_3]$	$c'''_4[a_3, a_4, \underline{b}_3, b_4, \underline{c}_4]$

c_1

1_1	2_1	3_1	4_1
2_1	5_1	4_1	6_1
7_1	8_1	9_1	10_1
8_1	11_1	10_1	12_1

c'_1

$\underline{5}_1$	$\underline{2}_1$	$\underline{6}_1$	$\underline{4}_1$
$\underline{2}_1$	$\underline{1}_1$	$\underline{4}_1$	$\underline{3}_1$
$\underline{11}_1$	$\underline{8}_1$	$\underline{12}_1$	$\underline{10}_1$
$\underline{8}_1$	$\underline{7}_1$	$\underline{10}_1$	$\underline{9}_1$

c''_1

2_1	1_1	4_1	3_1
5_1	2_1	6_1	4_1
8_1	7_1	10_1	9_1
11_1	8_1	12_1	10_1

c'''_1

$\underline{2}_1$	$\underline{5}_1$	$\underline{4}_1$	$\underline{6}_1$
$\underline{1}_1$	$\underline{2}_1$	$\underline{3}_1$	$\underline{4}_1$
$\underline{8}_1$	$\underline{11}_1$	$\underline{10}_1$	$\underline{12}_1$
$\underline{7}_1$	$\underline{8}_1$	$\underline{9}_1$	$\underline{10}_1$

c_2

1_2	2_2	3_2	4_2
5_2	6_2	7_2	8_2
9_2	10_2	11_2	12_2
13_2	14_2	15_2	16_2

c'_2

$\underline{6}_2$	$\underline{5}_2$	$\underline{8}_2$	$\underline{7}_2$
$\underline{2}_2$	$\underline{1}_2$	$\underline{4}_2$	$\underline{3}_2$
$\underline{14}_2$	$\underline{13}_2$	$\underline{16}_2$	$\underline{15}_2$
$\underline{10}_2$	$\underline{13}_2$	$\underline{12}_2$	$\underline{11}_2$

c''_2

2_2	1_2	4_2	3_2
6_2	5_2	8_2	7_2
10_2	13_2	12_2	11_2
14_2	13_2	16_2	15_2

c'''_2

$\underline{5}_2$	$\underline{6}_2$	$\underline{7}_2$	$\underline{8}_2$
$\underline{1}_2$	$\underline{2}_2$	$\underline{3}_2$	$\underline{4}_2$
$\underline{13}_2$	$\underline{14}_2$	$\underline{15}_2$	$\underline{16}_2$
$\underline{9}_2$	$\underline{10}_2$	$\underline{11}_2$	$\underline{12}_2$

c_3

1_3	$\underline{2}_3$	2_3	$\underline{1}_3$
$\underline{4}_3$	3_3	$\underline{3}_3$	4_3
4_3	$\underline{3}_3$	3_3	$\underline{4}_3$
$\underline{1}_3$	2_3	$\underline{2}_3$	1_3

c'_3

$\underline{3}_3$	4_3	$\underline{4}_3$	3_3
2_3	$\underline{1}_3$	1_3	$\underline{2}_3$
$\underline{2}_3$	1_3	$\underline{1}_3$	2_3
3_3	$\underline{4}_3$	4_3	$\underline{3}_3$

c''_3

2_3	$\underline{1}_3$	1_3	$\underline{2}_3$
$\underline{3}_3$	4_3	$\underline{4}_3$	3_3
3_3	$\underline{4}_3$	4_3	$\underline{3}_3$
$\underline{2}_3$	1_3	$\underline{1}_3$	2_3

c'''_3

$\underline{4}_3$	3_3	$\underline{3}_3$	4_3
1_3	$\underline{2}_3$	2_3	$\underline{1}_3$
$\underline{1}_3$	2_3	$\underline{2}_3$	1_3
4_3	$\underline{3}_3$	3_3	$\underline{4}_3$

c_4

1_4	5_4	$\underline{2}_4$	6_4
7_4	$\underline{1}_4$	8_4	2_4
$\underline{4}_4$	9_4	3_4	10_4
11_4	4_4	12_4	$\underline{3}_4$

c'_4

1_4	$\underline{7}_4$	$\underline{2}_4$	8_4
$\underline{5}_4$	$\underline{1}_4$	$\underline{6}_4$	2_4
$\underline{4}_4$	$\underline{11}_4$	3_4	$\underline{12}_4$
$\underline{9}_4$	4_4	$\underline{10}_4$	$\underline{3}_4$

c''_4

5_4	1_4	6_4	$\underline{2}_4$
$\underline{1}_4$	7_4	2_4	$\underline{8}_4$
9_4	$\underline{4}_4$	10_4	3_4
4_4	11_4	$\underline{3}_4$	12_4

c'''_4

$\underline{7}_4$	1_4	$\underline{8}_4$	$\underline{2}_4$
$\underline{1}_4$	$\underline{5}_4$	2_4	$\underline{6}_4$
$\underline{11}_4$	$\underline{4}_4$	$\underline{12}_4$	3_4
4_4	$\underline{9}_4$	$\underline{3}_4$	$\underline{10}_4$

Рис. 4.3.7

Эти переключения просты и требуют минимального добавления оборудования к базовому блоку сумматора. Общие же аппаратурные затраты устройства суммирования с учетом того, что сумматор контрольной части систематического кода представляется в основном двоичном коде и поэтому выполняется только для трех разрядов, в четыре раза меньше, чем в первом варианте реализации этого устройства.

Пример 4. Синтез сумматора основания n = 16 предельного быстродействия с исправлением одиночных ошибок

Сумматор основания n = 16 предельного быстродействия, т.е. реализуемый двухуровневой логической схемой, даже без исправления одиночных ошибок из-за больших аппаратурных затрат пока практически не используется. Тем не менее представляет значительный интерес рассмотреть синтез подобного устройства с исправлением одиночных ошибок во входных шинах его операнд. На рис. 4.4.1 приведена блок-схема такого сумматора, где его операнды представлены в систематическом коде **AX**, **BY**, имеющем возможности по исправлению одиночных ошибок. Перед подачей на основной суммирующий блок их сигналы преобразуются в соответствующие цифры (0_a – 15_a), (0_x – 7_x), ($\underline{0}_x$ – $\underline{7}_x$), (0_b – 15_b), (0_y – 7_y), ($\underline{0}_y$ – $\underline{7}_y$).

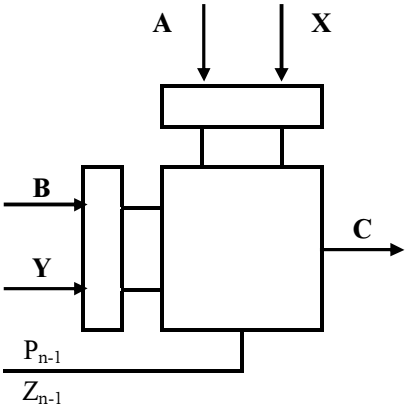


Рис. 4.4.1

В качестве систематического совершенного кода, в котором представляются входные и выходные сигналы сумматора **C = (A + B)**, а именно **AX, BY, CW**, выберем код табл. 4.2.2 с координатами 15, 0. Задание этого кода и логические зависимости, определяющие процедуру исправления одиночных ошибок в информационной части этого кода, могут быть представлены (рис. 4.4.2), как известно, пятью фигурами в многомерном цифровом пространстве, где в ячейках его информационной части размещаются прямые и инверсные значения цифр контрольной части кода.

0	6	3	5
7	1	4	2
5	3	6	0
2	4	1	7

15, 0

6	$\underline{0}$	3	$\underline{5}$
1	$\underline{7}$	4	$\underline{2}$
5	$\underline{3}$	0	$\underline{6}$
2	$\underline{4}$	7	$\underline{1}$

a'_1

5	3	$\underline{0}$	$\underline{6}$
2	4	$\underline{7}$	$\underline{1}$
6	0	$\underline{3}$	$\underline{5}$
1	7	$\underline{4}$	$\underline{2}$

a'_2

7	1	2	4
$\underline{0}$	$\underline{6}$	$\underline{5}$	$\underline{3}$
4	2	1	7
$\underline{3}$	$\underline{5}$	$\underline{6}$	$\underline{0}$

a'_3

3	5	6	0
4	2	1	7
$\underline{0}$	$\underline{6}$	$\underline{5}$	$\underline{3}$
$\underline{7}$	$\underline{1}$	$\underline{2}$	$\underline{4}$

a'_4

Рис. 4.4.2

Первая фигура на этом рисунке определяет только соотношения между номерами кодовых комбинаций информационной и контрольной частями кода, а другие фигуры представляют собой упомянутые логические выражения, соответственно для четырех разрядов этого кода, где каждая зависимость выражается логической суммой всех значений, расположенных в ячейках этого цифрового пространства. Подразумевается, что в каждой ячейке этого пространства размещается логическое произведение одной из цифр информационной части кода, которая совпадает с координатами этой ячейки и поэтому позволяет для краткости записи ее не отмечать, на прямое или инверсное значение цифры контрольной части кода.

Ниже приведены три эквивалентные формы представления этой зависимости для четырех разрядов систематического кода: первые две формы – геометрические, а третья форма – аналитическая, которая записана выражениями (4.4.1) – (4.4.4).

$$a'_1 = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 8 & 9 & 10 & 11 \\ \hline 12 & 13 & 14 & 15 \\ \hline \end{array}_{\,a} \;\&\; \begin{array}{|c|c|c|c|}\hline 6 & \underline{0} & 3 & \underline{5} \\ \hline 1 & \underline{7} & 4 & \underline{2} \\ \hline 5 & \underline{3} & 0 & \underline{6} \\ \hline 2 & \underline{4} & 7 & \underline{1} \\ \hline \end{array}_{\,x} = \begin{array}{llll} 0_a6_x \vee & 1_a\underline{0}_x \vee & 2_a3_x \vee & 3_a\underline{5}_x \vee \\ \vee\, 4_a1_x \vee & 5_a\underline{7}_x \vee & 6_a4_x \vee & 7_a\underline{2}_x \vee \\ \vee\, 8_a5_x \vee & 9_a\underline{3}_x \vee & 10_a0_x \vee & 11_a\underline{6}_x \vee \\ \vee\, 12_a2_x & 13_a\underline{4}_x \vee & 14_a7_x \vee & 15_a\underline{1}_x . \end{array} \qquad (4.4.1)$$

$$a'_2 = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 8 & 9 & 10 & 11 \\ \hline 12 & 13 & 14 & 15 \\ \hline \end{array}_{\,a} \;\&\; \begin{array}{|c|c|c|c|}\hline 5 & 3 & \underline{0} & \underline{6} \\ \hline 2 & 4 & \underline{7} & \underline{1} \\ \hline 6 & 0 & \underline{3} & \underline{5} \\ \hline 1 & 7 & \underline{4} & \underline{2} \\ \hline \end{array}_{\,x} = \begin{array}{llll} 0_a5_x \vee & 1_a3_x \vee & 2_a\underline{0}_x \vee & 3_a\underline{6}_x \vee \\ \vee\, 4_a2_x \vee & 5_a4_x \vee & 6_a\underline{7}_x \vee & 7_a\underline{7}_x \vee \\ \vee\, 8_a6_x \vee & 9_a0_x \vee & 10_a\underline{3}_x \vee & 11_a\underline{3}_x \vee \\ \vee\, 12_a1_x & 13_a7_x \vee & 14_a\underline{4}_x \vee & 15_a\underline{2}_x . \end{array} \qquad (4.4.2)$$

$$a'_3 = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 8 & 9 & 10 & 11 \\ \hline 12 & 13 & 14 & 15 \\ \hline \end{array}_{\,a} \;\&\; \begin{array}{|c|c|c|c|}\hline 7 & 1 & 2 & 4 \\ \hline \underline{0} & \underline{6} & \underline{5} & \underline{3} \\ \hline 4 & 2 & 1 & 7 \\ \hline \underline{3} & \underline{5} & \underline{6} & \underline{0} \\ \hline \end{array}_{\,x} = \begin{array}{llll} 0_a7_x \vee & 1_a1_x \vee & 2_a2_x \vee & 3_a4_x \vee \\ \vee\, 4_a\underline{0}_x \vee & 5_a\underline{6}_x \vee & 6_a\underline{5}_x \vee & 7_a\underline{3}_x \vee \\ \vee\, 8_a4_x \vee & 9_a2_x \vee & 10_a1_x \vee & 11_a7_x \vee \\ \vee\, 12_a\underline{3}_x & 13_a\underline{5}_x \vee & 14_a\underline{6}_x \vee & 15_a\underline{0}_x . \end{array} \qquad (4.4.3)$$

$$a'_4 = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 8 & 9 & 10 & 11 \\ \hline 12 & 13 & 14 & 15 \\ \hline \end{array}_{\,a} \;\&\; \begin{array}{|c|c|c|c|}\hline 3 & 5 & 6 & 0 \\ \hline 4 & 2 & 1 & 7 \\ \hline \underline{0} & \underline{6} & \underline{5} & \underline{3} \\ \hline \underline{7} & \underline{1} & \underline{2} & \underline{4} \\ \hline \end{array}_{\,x} = \begin{array}{llll} 0_a3_x \vee & 1_a5_x \vee & 2_a6_x \vee & 3_a0_x \vee \\ \vee\, 4_a4_x \vee & 5_a2_x \vee & 6_a1_x \vee & 7_a7_x \vee \\ \vee\, 8_a\underline{0}_x \vee & 9_a\underline{6}_x \vee & 10_a\underline{5}_x \vee & 11_a\underline{3}_x \vee \\ \vee\, 12_a\underline{1}_x & 13_a\underline{1}_x \vee & 14_a\underline{2}_x \vee & 15_a\underline{4}_x . \end{array} \qquad (4.4.4)$$

При этом для каждого разряда кода весь комплекс непрерывного множества цифр (0_a – 15_a) его информационной части находится в определенном жестком соотношении с комплексом прямых и инверсных значений кодовых комбинаций его контрольной части.

Следует ожидать, что такие же соотношения должны сохраняться не только для входных сигналов разрядов сумматора, но и для выходных сигналов. Очевидность этого утверждения подтверждается тем, что цифровые множества, определяющие сигналы разрядов информационной части кода, при выполнении операций суммирования и вычитания, которые задаются соответствующими таблицами истинности, передвигаются циклически и поэтому остаются неизменными. Тем самым всегда сохраняется соответствие между цифровыми сигналами информационной и контрольной частей систематического кода. Поэтому логические выражения (4.4.1) – (4.4.4) применимы для выходных сигналов разрядов сумматора, где вместо цифр (0_a – 15_a) необходимо подставить цифры (0_c – 15_c), а вместо контрольных цифр (0_x – 7_x) – контрольные цифры (0_w – 7_w).

Дальнейшая задача синтеза заключается в определении всех этих цифр сумматора. При этом громоздкость логических выражений может быть уменьшена их геометрическим представлением, как это показано выше.

Из выражений (4.4.1) – (4.4.4) следует, что достаточно определить логическое выражение для одного из разрядов кода, например первого, а остальные сигналы информационных разрядов получаются простой подстановкой иных значений контрольных цифр, что в равной степени относится к выходным разрядам сумматора.

Синтез устройства проведем на примере первого разряда сумматора для четырех режимов его работы. (**C = (A + B)**, $P_{n-1} = 0^*$; **C = (A + B)**, $P_{n-1} = 1^*$; **C = (A – B)**, $Z_{n-1} = 0^*$; **C = (A – B)**, $Z_{n-1} = 1^*$).

Общеизвестность таблиц сложения и вычитания, представленных в обычном цифровом коде, где координаты каждой выходной цифры определяют цифры операндов и, наоборот, координаты операндов определяют цифру результата сложения или вычитания, что позволяет непосредственно записать логические выражения для выходных цифр устройства сложения и вычитания. В геометрическом виде они представляются следующим образом:

а) для режима **C = (A + B)**, $P_{n-1} = 0^*$

$0_c / 6_w =$

$0_a / 6_x$	$15_a / \underline{1}_x$	$14_a / 7_x$	$13_a / \underline{4}_x$
$12_a\ 2_x$	$11_a / \underline{6}_x$	$10_a / 0_x$	$9_a / \underline{3}_x$
$8_a / 5_x$	$7_a / \underline{2}_x$	$6_a / 4_x$	$5_a / \underline{7}_x$
$4_a / 1_x$	$3_a / \underline{5}_x$	$2_a / 3_x$	$1_a / \underline{0}_x$

a / x

&

$0_b / 6_y$	$1_b / \underline{0}_y$	$2_b / 3_y$	$3_b / \underline{5}_y$
$4_b / 1_y$	$5_b / \underline{7}_y$	$6_b / 4_y$	$7_b / \underline{2}_y$
$8_b / 5_y$	$9_b / \underline{3}_y$	$10_b / 0_y$	$11_b / \underline{6}_y$
$12_b / 2_y$	$13_b / \underline{4}_y$	$14_b / 7_y$	$15_b / \underline{1}_y$

b / y ,

$1_c / \underline{0}_w =$

$1_a / \underline{0}_x$	$0_a / 6_x$	$15_a / \underline{1}_x$	$14_a / 7_x$
$13_a / \underline{4}_x$	$12_a / 2_x$	$11_a / \underline{6}_x$	$10_a / 0_x$
$9_a / \underline{3}_x$	$8_a / 5_x$	$7_a / \underline{2}_x$	$6_a / 4_x$
$5_a / \underline{7}_x$	$4_a / 1_x$	$3_a / \underline{5}_x$	$2_a / 3_x$

a / x

&

$0_b / 6_y$	$1_b / \underline{0}_y$	$2_b / 3_y$	$3_b / \underline{5}_y$
$4_b / 1_y$	$5_b / \underline{7}_y$	$6_b / 4_y$	$7_b / \underline{2}_y$
$8_b / 5_y$	$9_b / \underline{3}_y$	$10_b / 0_y$	$11_b / \underline{6}_y$
$12_b / 2_y$	$13_b / \underline{4}_y$	$14_b / 7_y$	$15_b / \underline{1}_y$

b / y ,

⋮

$15_c / \underline{1}_w =$

$15_a / \underline{1}_x$	$14_a / 7_x$	$13_a / \underline{4}_x$	$12_a / 2_x$
$11_a / \underline{6}_x$	$10_a / 0_x$	$9_a / \underline{3}_x$	$8_a / 5_x$
$7_a / \underline{2}_x$	$6_a / 4_x$	$5_a / \underline{7}_x$	$4_a / 1_x$
$3_a / \underline{5}_x$	$2_a / 3_x$	$1_a / \underline{0}_x$	$0_a / 6_x$

a / x

&

$0_b / 6_y$	$1_b / \underline{0}_y$	$2_b / 3_y$	$3_b / \underline{5}_y$
$4_b / 1_y$	$5_b / \underline{7}_y$	$6_b / 4_y$	$7_b / \underline{2}_y$
$8_b / 5_y$	$9_b / \underline{3}_y$	$10_b / 0_y$	$11_b / \underline{6}_y$
$12_b / 2_y$	$13_b / \underline{4}_y$	$14_b / 7_y$	$15_b / \underline{1}_y$

b / y . (4.4.5)

б) для режима **C = (A + B)**, $P_{n-1} = 1^*$

$0_c / 6_w =$

$15_a / \underline{1}_x$	$14_a / 7_x$	$13_a / \underline{4}_x$	$12_a / 2_x$
$11_a / \underline{6}_x$	$10_a / 0_x$	$9_a / \underline{3}_x$	$8_a / 5_x$
$7_a / \underline{2}_x$	$6_a / 4_x$	$5_a / \underline{7}_x$	$4_a / 1_x$
$3_a / \underline{5}_x$	$2_a / 3_x$	$1_a / \underline{0}_x$	$0_a / 6_x$

a / x

&

$0_b / 6_y$	$1_b / \underline{0}_y$	$2_b / 3_y$	$3_b / \underline{5}_y$
$4_b / 1_y$	$5_b / \underline{7}_y$	$6_b / 4_y$	$7_b / \underline{2}_y$
$8_b / 5_y$	$9_b / \underline{3}_y$	$10_b / 0_y$	$11_b / \underline{6}_y$
$12_b / 2_y$	$13_b / \underline{4}_y$	$14_b / 7_y$	$15_b / \underline{1}_y$

b / y ,

$1_c / \underline{0}_w =$

$0_a / 6_x$	$15_a / \underline{1}_x$	$14_a / 7_x$	$13_a / \underline{4}_x$
$12_a / 2_x$	$11_a / \underline{6}_x$	$10_a / 0_x$	$9_a / \underline{3}_x$
$8_a / 5_x$	$7_a / \underline{2}_x$	$6_a / 4_x$	$5_a / \underline{7}_x$
$4_a / 1_x$	$3_a / \underline{5}_x$	$2_a / 3_x$	$1_a / \underline{0}_x$

a / x

&

$0_b / 6_y$	$1_b / \underline{0}_y$	$2_b / 3_y$	$3_b / \underline{5}_y$
$4_b / 1_y$	$5_b / \underline{7}_y$	$6_b / 4_y$	$7_b / \underline{2}_y$
$8_b / 5_y$	$9_b / \underline{3}_y$	$10_b / 0_y$	$11_b / \underline{6}_y$
$12_b / 2_y$	$13_b / \underline{4}_y$	$14_b / 7_y$	$15_b / \underline{1}_y$

b / y ,

⋮

$15_c/\underline{1}_w =$

$14_a/7_x$	$13_a/\underline{4}_x$	$12_a/2_x$	$11_a/\underline{6}_x$
$10_a/0_x$	$9_a/\underline{3}_x$	$8_a/5_x$	$7_a/\underline{2}_x$
$6_a/4_x$	$5_a/\underline{7}_x$	$4_a/1_x$	$3_a/\underline{5}_x$
$2_a/3_x$	$1_a/\underline{0}_x$	$0_a/6_x$	$15_a/\underline{1}_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

, (4.4.6)

в) для режима **C = (A – B)**, $Z_{n-1} = 0$*

$0_c/6_w =$

$0_a/6_x$	$1_a/\underline{0}_x$	$2_a/3_x$	$3_a/\underline{5}_x$
$4_a/1_x$	$5_a/\underline{7}_x$	$6_a/4_x$	$7_a/\underline{2}_x$
$8_a/5_x$	$9_a/\underline{3}_x$	$10_a/0_x$	$11_a/\underline{6}_x$
$12_a/2_x$	$13_a/\underline{4}_x$	$14_a/7_x$	$15_a/\underline{1}_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

b / y ,

$1_c/\underline{0}_w =$

$1_a/\underline{0}_x$	$2_a/3_x$	$3_a/\underline{5}_x$	$4_a/1_x$
$5_a/\underline{7}_x$	$6_a/4_x$	$7_a/\underline{2}_x$	$8_a/5_x$
$9_a/\underline{3}_x$	$10_a/0_x$	$11_a/\underline{6}_x$	$12_a/2_x$
$13_a/\underline{4}_x$	$14_a/7_x$	$15_a/\underline{1}_x$	$0_a/6_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

b / y ,

⋮

$15_c\underline{1}_w =$

$15_a/\underline{1}_x$	$0_a/6_x$	$1_a/\underline{0}_x$	$2_a/3_x$
$3_a/\underline{5}_x$	$4_a/1_x$	$5_a/\underline{7}_x$	$6_a/4_x$
$7_a/\underline{2}_x$	$8_a/5_x$	$9_a/\underline{3}_x$	$10_a/0_x$
$11_a/\underline{6}_x$	$12_a/2_x$	$13_a/\underline{4}_x$	$14_a/7_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

. (4.4.7)

г) для режима **C = (A – B)**, $Z_{n-1} = 1$*

$0_c/6_w =$

$1_a/\underline{0}_x$	$2_a/3_x$	$3_a/\underline{5}_x$	$4_a/1_x$
$5_a/\underline{7}_x$	$6_a/4_x$	$7_a/\underline{2}_x$	$8_a/5_x$
$9_a/\underline{3}_x$	$10_a/0_x$	$11_a/\underline{6}_x$	$12_a/2_x$
$13_a/\underline{4}_x$	$14_a/7_x$	$15_a/\underline{1}_x$	$0_a/6_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

b / y ,

$1_c/\underline{0}_w =$

$2_a/3_x$	$3_a/\underline{5}_x$	$4_a/1_x$	$5_a/\underline{7}_x$
$6_a/4_x$	$7_a/\underline{2}_x$	$8_a/5_x$	$9_a/\underline{3}_x$
$10_a/0_x$	$11_a/\underline{6}_x$	$12_a/2_x$	$13_a/\underline{4}_x$
$14_a/7_x$	$15_a/\underline{1}_x$	$0_a/6_x$	$1_a/\underline{0}_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

b / y ,

⋮

$15_c/\underline{1}_w =$

$0_a/6_x$	$1_a/\underline{0}_x$	$2_a/3_x$	$3_a/\underline{5}_x$
$4_a/1_x$	$5_a/\underline{7}_x$	$6_a/4_x$	$7_a/\underline{2}_x$
$8_a/5_x$	$9_a/\underline{3}_x$	$10_a/0_x$	$11_a/\underline{6}_x$
$12_a/2_x$	$13_a/\underline{4}_x$	$14_a/7_x$	$15_a/\underline{1}_x$

a / x &

$0_b/6_y$	$1_b/\underline{0}_y$	$2_b/3_y$	$3_b/\underline{5}_y$
$4_b/1_y$	$5_b/\underline{7}_y$	$6_b/4_y$	$7_b/\underline{2}_y$
$8_b/5_y$	$9_b/\underline{3}_y$	$10_b/0_y$	$11_b/\underline{6}_y$
$12_b/2_y$	$13_b/\underline{4}_y$	$14_b/7_y$	$15_b/\underline{1}_y$

. (4.4.8)

Выражения (4.4.5) – (4.4.8) определяют не только выходные цифровые сигналы (0_c –15_c), общие для всех выходных сигналов c_1 – c_4 разрядов устройства, но и жестко связанные с ними цифры контрольных разрядов сигнала c_1. В этих выражениях это соответствие представлено записью в ячейках про-

странства дробью, где числитель относится к информационной части кода, а знаменатель – к контрольной части систематического кода. Следует отметить, что эти зависимости представляют собой сжатую запись логических выражений для выходного сигнала c_1.

Например, для первого режима работы сумматора (**A = (C + B)**, $P_{n-1} = 0$*) развернутое логическое выражение будет записано следующим образом:

$$
\begin{aligned}
c_1 = (&6_x6_y \vee \underline{1}_x\underline{0}_y \vee 7_x3_y \vee \underline{4}_x\underline{5}_y \vee 2_x1_y \vee \underline{6}_x\underline{7}_y \vee 0_x4_y \vee \underline{3}_x\underline{2}_y \vee 5_x5_y \vee \underline{2}_x\underline{3}_y \vee 4_x0_y \vee \underline{7}_x\underline{6}_y \vee 1_x2_y \vee \underline{5}_x\underline{4}_y \vee 3_x7_y \vee \underline{0}_x\underline{1}_y)\ 0_c \vee \\
\vee (&\underline{0}_x6_y \vee 6_x\underline{0}_y \vee \underline{1}_x3_y \vee 7_x\underline{5}_y \vee \underline{4}_x1_y \vee 2_x\underline{7}_y \vee \underline{6}_x4_y \vee 0_x\underline{2}_y \vee \underline{3}_x5_y \vee 5_x\underline{3}_y \vee \underline{2}_x0_y \vee 4_x\underline{6}_y \vee \underline{7}_x2_y \vee 1_x\underline{4}_y \vee \underline{5}_x7_y \vee 3_x\underline{1}_y)\ 1_c \vee \\
\vee (&3_x6_y \vee \underline{0}_x\underline{0}_y \vee 6_x3_y \vee \underline{1}_x\underline{5}_y \vee 7_x1_y \vee \underline{4}_x\underline{7}_y \vee 2_x4_y \vee \underline{6}_x\underline{2}_y \vee 0_x5_y \vee \underline{3}_x\underline{3}_y \vee 5_x0_y \vee \underline{2}_x\underline{6}_y \vee 4_x2_y \vee \underline{7}_x\underline{4}_y \vee 1_x7_y \vee \underline{5}_x\underline{1}_y)\ 2_c \vee \\
\vee (&\underline{5}_x6_y \vee 3_x\underline{0}_y \vee \underline{0}_x3_y \vee 6_x\underline{5}_y \vee \underline{1}_x1_y \vee 7_x\underline{7}_y \vee \underline{4}_x4_y \vee 2_x\underline{2}_y \vee \underline{6}_x5_y \vee 0_x\underline{3}_y \vee \underline{3}_x0_y \vee 5_x\underline{6}_y \vee \underline{2}_x2_y \vee 4_x\underline{4}_y \vee \underline{7}_x7_y \vee 1_x\underline{1}_y)\ 3_c \vee \\
\vee (&1_x6_y \vee \underline{5}_x\underline{0}_y \vee 3_x3_y \vee \underline{0}_x\underline{5}_y \vee 6_x1_y \vee \underline{1}_x\underline{7}_y \vee 7_x4_y \vee \underline{4}_x\underline{2}_y \vee 2_x5_y \vee \underline{6}_x\underline{3}_y \vee 0_x0_y \vee \underline{3}_x\underline{6}_y \vee 5_x2_y \vee \underline{2}_x\underline{4}_y \vee 4_x7_y \vee \underline{7}_x\underline{1}_y)\ 4_c \vee \\
\vee (&\underline{7}_x6_y \vee 1_x\underline{0}_y \vee \underline{5}_x3_y \vee 3_x\underline{5}_y \vee \underline{0}_x1_y \vee 6_x\underline{7}_y \vee \underline{1}_x4_y \vee 7_x\underline{2}_y \vee \underline{4}_x5_y \vee 2_x\underline{3}_y \vee \underline{6}_x0_y \vee 0_x\underline{6}_y \vee \underline{3}_x2_y \vee 5_x\underline{4}_y \vee \underline{2}_x7_y \vee 4_x\underline{1}_y)\ 5_c \vee \\
\vee (&4_x6_y \vee \underline{7}_x\underline{0}_y \vee 1_x3_y \vee \underline{5}_x\underline{5}_y \vee 3_x1_y \vee \underline{0}_x\underline{7}_y \vee 6_x4_y \vee \underline{1}_x\underline{2}_y \vee 7_x5_y \vee \underline{4}_x\underline{3}_y \vee 2_x0_y \vee \underline{6}_x\underline{6}_y \vee 0_x2_y \vee \underline{3}_x\underline{4}_y \vee 5_x7_y \vee \underline{2}_x\underline{1}_y)\ 6_c \vee \\
\vee (&\underline{2}_x6_y \vee 4_x\underline{0}_y \vee \underline{7}_x3_y \vee 1_x\underline{5}_y \vee \underline{5}_x1_y \vee 3_x\underline{7}_y \vee \underline{0}_x4_y \vee 6_x\underline{2}_y \vee \underline{1}_x5_y \vee 7_x\underline{3}_y \vee \underline{4}_x0_y \vee 2_x\underline{6}_y \vee \underline{6}_x2_y \vee 0_x\underline{4}_y \vee \underline{3}_x7_y \vee 5_x\underline{1}_y)\ 7_c \vee \\
\vee (&5_x6_y \vee \underline{2}_x\underline{0}_y \vee 4_x3_y \vee \underline{7}_x\underline{5}_y \vee 1_x1_y \vee \underline{5}_x\underline{7}_y \vee 3_x4_y \vee \underline{0}_x\underline{2}_y \vee 6_x5_y \vee \underline{1}_x\underline{3}_y \vee 7_x0_y \vee \underline{4}_x\underline{6}_y \vee 2_x2_y \vee \underline{6}_x\underline{4}_y \vee 0_x7_y \vee \underline{3}_x\underline{1}_y)\ 8_c \vee \\
\vee (&\underline{3}_x6_y \vee 5_x\underline{0}_y \vee \underline{2}_x3_y \vee 4_x\underline{5}_y \vee \underline{7}_x1_y \vee 1_x\underline{7}_y \vee \underline{5}_x4_y \vee 3_x\underline{2}_y \vee \underline{0}_x5_y \vee 6_x\underline{3}_y \vee \underline{1}_x0_y \vee 7_x\underline{6}_y \vee \underline{4}_x2_y \vee 2_x\underline{4}_y \vee \underline{6}_x7_y \vee 0_x\underline{1}_y)\ 9_c \vee \\
\vee (&0_x6_y \vee \underline{3}_x\underline{0}_y \vee 5_x3_y \vee \underline{2}_x\underline{5}_y \vee 4_x1_y \vee \underline{7}_x\underline{7}_y \vee 1_x4_y \vee \underline{5}_x\underline{2}_y \vee 3_x5_y \vee \underline{0}_x\underline{3}_y \vee 6_x0_y \vee \underline{1}_x\underline{6}_y \vee 7_x2_y \vee \underline{4}_x\underline{4}_y \vee 2_x7_y \vee \underline{6}_x\underline{1}_y)\ 10_c \vee \\
\vee (&\underline{6}_x6_y \vee 0_x\underline{0}_y \vee \underline{3}_x3_y \vee 5_x\underline{5}_y \vee \underline{2}_x1_y \vee 4_x\underline{7}_y \vee \underline{7}_x4_y \vee 1_x\underline{2}_y \vee \underline{5}_x5_y \vee 3_x\underline{3}_y \vee \underline{0}_x0_y \vee 6_x\underline{6}_y \vee \underline{1}_x2_y \vee 7_x\underline{4}_y \vee \underline{4}_x7_y \vee 2_x\underline{1}_y)\ 11_c \vee \\
\vee (&2_x6_y \vee \underline{6}_x\underline{0}_y \vee 0_x3_y \vee \underline{3}_x\underline{5}_y \vee 5_x1_y \vee \underline{2}_x\underline{7}_y \vee 4_x4_y \vee \underline{7}_x\underline{2}_y \vee 1_x5_y \vee \underline{5}_x\underline{3}_y \vee 3_x0_y \vee \underline{0}_x\underline{6}_y \vee 6_x2_y \vee \underline{1}_x\underline{4}_y \vee 7_x7_y \vee \underline{4}_x\underline{1}_y)\ 12_c \vee \\
\vee (&\underline{4}_x6_y \vee 2_x\underline{0}_y \vee \underline{6}_x3_y \vee 0_x\underline{5}_y \vee \underline{3}_x1_y \vee 5_x\underline{7}_y \vee \underline{2}_x4_y \vee 4_x\underline{2}_y \vee \underline{7}_x5_y \vee 1_x\underline{3}_y \vee \underline{5}_x0_y \vee 3_x\underline{6}_y \vee \underline{0}_x2_y \vee 6_x\underline{4}_y \vee \underline{1}_x7_y \vee 7_x\underline{1}_y)\ 13_c \vee \\
\vee (&7_x6_y \vee \underline{4}_x\underline{0}_y \vee 2_x3_y \vee \underline{6}_x\underline{5}_y \vee 0_x1_y \vee \underline{3}_x\underline{7}_y \vee 5_x4_y \vee \underline{2}_x\underline{2}_y \vee 4_x5_y \vee \underline{7}_x\underline{3}_y \vee 1_x0_y \vee \underline{5}_x\underline{6}_y \vee 3_x2_y \vee \underline{0}_x\underline{4}_y \vee 6_x7_y \vee \underline{1}_x\underline{1}_y)\ 14_c \vee \\
\vee (&\underline{1}_x6_y \vee 7_x\underline{0}_y \vee \underline{4}_x3_y \vee 2_x\underline{5}_y \vee \underline{6}_x1_y \vee 0_x\underline{7}_y \vee \underline{3}_x4_y \vee 5_x\underline{2}_y \vee \underline{2}_x5_y \vee 4_x\underline{3}_y \vee \underline{7}_x0_y \vee 1_x\underline{6}_y \vee \underline{5}_x2_y \vee 3_x\underline{4}_y \vee \underline{0}_x7_y \vee 6_x\underline{1}_y)\ 15_c\,,
\end{aligned}
\tag{4.4.9}
$$

где

$$0_c = 0_a0_b \vee 15_a1_b \vee 14_a2_b \vee 13_a3_b \vee 12_a4_b \vee 11_a5_b \vee 10_a6_b \vee 9_a7_b \vee 8_a8_b \vee 7_a9_b \vee 6_a10_b \vee 5_a11_b \vee 4_a12_b \vee \vee 3_a13_b \vee 2_a14_b \vee 1_a15_b,$$

$$1_c = 1_a0_b \vee 0_a1_b \vee 15_a2_b \vee 14_a3_b \vee 13_a4_b \vee 12_a5_b \vee 11_a6_b \vee 10_a7_b \vee 9_a8_b \vee 8_a9_b \vee 7_a10_b \vee 6_a11_b \vee \vee 5_a12_b \vee 4_a13_b \vee 3_a14_b \vee 2_a15_b,$$

$$2_c = 2_a0_b \vee 1_a1_b \vee 0_a2_b \vee 15_a3_b \vee 14_a4_b \vee 13_a5_b \vee 12_a6_b \vee 11_a7_b \vee 10_a8_b \vee 9_a9_b \vee 8_a10_b \vee \vee 7_a11_b \vee \vee 6_a12_b \vee 5_a13_b \vee 4_a14_b \vee 3_a15_b,$$

$$3_c = 3_a0_b \vee 2_a1_b \vee 1_a2_b \vee 0_a3_b \vee 15_a4_b \vee 14_a5_b \vee 13_a6_b \vee 12_a7_b \vee 11_a8_b \vee 10_a9_b \vee 9_a10_b \vee \vee 8_a11_b \vee \vee 7_a12_b \vee 6_a13_b \vee 5_a14_b \vee 4_a15_b,$$

$$4_c = 4_a0_b \vee 3_a1_b \vee 2_a2_b \vee 1_a3_b \vee 0_a4_b \vee 15_a5_b \vee 14_a6_b \vee 13_a7_b \vee 12_a8_b \vee 11_a9_b \vee 10_a10_b \vee 9_a11_b \vee \vee 8_a12_b \vee 7_a13_b \vee 6_a14_b \vee 5_a15_b,$$

$$5_c = 5_a0_b \vee 4_a1_b \vee 3_a2_b \vee 2_a3_b \vee 1_a4_b \vee 0_a5_b \vee 15_a6_b \vee 14_a7_b \vee 13_a8_b \vee 12_a9_b \vee 11_a10_b \vee 10_a11_b \vee \vee 9_a12_b \vee 8_a13_b \vee 7_a14_b \vee 6_a15_b,$$

$$6_c = 6_a0_b \vee 5_a1_b \vee 4_a2_b \vee 3_a3_b \vee 2_a4_b \vee 1_a5_b \vee 0_a6_b \vee 15_a7_b \vee 14_a8_b \vee 13_a9_b \vee 12_a10_b \vee 11_a11_b \vee \vee 10_a12_b \vee 9_a13_b \vee 8_a14_b \vee 7_a15_b,$$

$$7_c = 7_a0_b \vee 6_a1_b \vee 5_a2_b \vee 4_a3_b \vee 3_a4_b \vee 2_a5_b \vee 1_a6_b \vee 0_a7_b \vee 15_a8_b \vee 14_a9_b \vee 13_a10_b \vee 12_a11_b \vee \vee 11_a12_b \vee 10_a13_b \vee 9_a14_b \vee 8_a15_b,$$

$$8_c = 8_a0_b \vee 7_a1_b \vee 6_a2_b \vee 5_a3_b \vee 4_a4_b \vee 3_a5_b \vee 2_a6_b \vee 1_a7_b \vee 0_a8_b \vee 15_a9_b \vee 14_a10_b \vee 13_a11_b \vee \vee 12_a12_b \vee 11_a13_b \vee 10_a14_b \vee 9_a15_b,$$

$$9_c = 9_a0_b \vee 8_a1_b \vee 7_a2_b \vee 6_a3_b \vee 5_a4_b \vee 4_a5_b \vee 3_a6_b \vee 2_a7_b \vee 1_a8_b \vee 0_a9_b \vee 15_a10_b \vee 14_a11_b \vee \vee 13_a12_b \vee 12_a13_b \vee 11_a14_b \vee 10_a15_b,$$

$$10_c = 10_a0_b \vee 9_a1_b \vee 8_a2_b \vee 7_a3_b \vee 6_a4_b \vee 5_a5_b \vee 4_a6_b \vee 3_a7_b \vee 2_a8_b \vee 1_a9_b \vee 0_a0_b \vee 5_a11_b \vee 14_a12_b \vee \vee 13_a13_b \vee 12_a14_b \vee 11_a15_b,$$

$$11_c = 11_a0_b \vee 10_a1_b \vee 9_a2_b \vee 8_a3_b \vee 7_a4_b \vee 6_a5_b \vee 5_a6_b \vee 4_a7_b \vee 3_a8_b \vee 2_a9_b \vee 1_a10_b \vee 0_a11_b \vee 15_a12_b \vee$$
$$\vee 14_a13_b \vee 13_a14_b \vee 12_a15_b,$$
$$12_c = 12_a0_b \vee 11_a1_b \vee 10_a2_b \vee 9_a3_b \vee 8_a4_b \vee 7_a5_b \vee 6_a6_b \vee 5_a7_b \vee 4_a8_b \vee 3_a9_b \vee 2_a10_b \vee 1_a11_b \vee 0_a12_b \vee$$
$$\vee 15_a13_b \vee 14_a14_b \vee 13_a15_b,$$
$$13_c = 13_a0_b \vee 12_a1_b \vee 11_a2_b \vee 10_a3_b \vee 9_a4_b \vee 8_a5_b \vee 7_a6_b \vee 6_a7_b \vee 5_a8_b \vee 4_a9_b \vee 3_a10_b \vee 2_a11_b \vee 1_a12_b \vee$$
$$\vee 0_a13_b \vee 15a14_b \vee 14_a15_b,$$
$$14_c = 14_a0_b \vee 13_a1_b \vee 12_a2_b \vee 11_a3_b \vee 10_a4_b \vee 9_a5_b \vee 8_a6_b \vee 7_a7_b \vee 6_a8_b \vee 5_a9_b \vee 4_a10_b \vee 3_a11_b \vee 2_a12_b \vee$$
$$\vee 1_a13_b \vee 0_a14_b \vee 15a15_b,$$
$$15_c = 15_a0_b \vee 14_a1_b \vee 13_a2_b \vee 12_a3_b \vee 11_a4_b \vee 10_a5_b \vee 9_a6_b \vee 8_a7_b \vee 7_a8_b \vee 6_a9_b \vee 5_a10_b \vee 4_a11_b \vee 3_a12_b \vee$$
$$\vee 2_a13_b \vee 1_a14_b \vee 0_a15_b. \qquad (4.4.10)$$

Подставив (4.4.10) в (4.4.9), получим двухуровневую логическую схему основного блока сумматора рис. 4.4.1 для определения сигнала c_1, в котором одиночные ошибки в каждом из операндов не искажают этот сигнал. Реализация такой схемы не вызывает сомнений, но ее громоздкость очевидна. При последовательном выполнении операций формулами (4.4.9), (4.4.10) их реализация более реальна, причем формула (4.4.10) будет общей для остальных сигналов c_2, c_3, c_4, выражения которых аналогичны (4.4.5) – (4.4.8). В этих зависимостях необходимо в ячейках пространства записать только значение знаменателя, которое берется из представления систематического кода рис. 4.4.2.

Пример 5. Синтез устройства исправления одиночных ошибок двоичной системы счисления основания $n = 2^4$ без использования записи в ячейки пространства эквивалентных цифр его информационной части

Использование представленного в первом примере алгоритма синтеза устройства исправления ошибок в систематических кодах (с использованием промежуточной записи в ячейках пространства эквивалентных цифр его информационной части) и изложенного на примере основания $n = 2^4$ в случае больших оснований систем счисления их информационной части ($n = 2^{11}$, 2^{26}, 2^{57}, …) имеет единственное ограничение, которое заключается в необходимости использовать большие форматы листов для изложения промежуточных выкладок. Этот недостаток может быть уменьшен применением представленного ниже иного алгоритма синтеза. Этот вариант синтеза выполним, для большей его прозрачности, также для кода с основанием $n = 2^4$.

При этом синтезе будем стремиться максимально использовать симметрию соответствующих фигур геометрических образов многомерного цифрового пространства исправленных сигналов информационных ($a'_1 - a'_4$) и контрольных ($x'_1 - x'_3$) разрядов, что также будет служить оптимизации объема графического материала.

Связь между информационными разрядами ($a_1 - a_4$) и эквивалентными цифрами $0_x - 7_x$ контрольной части выбранного нами кода приведена на рис. 4.5.1 в многомерном цифровом пространстве координат двух частей этого систематического кода.

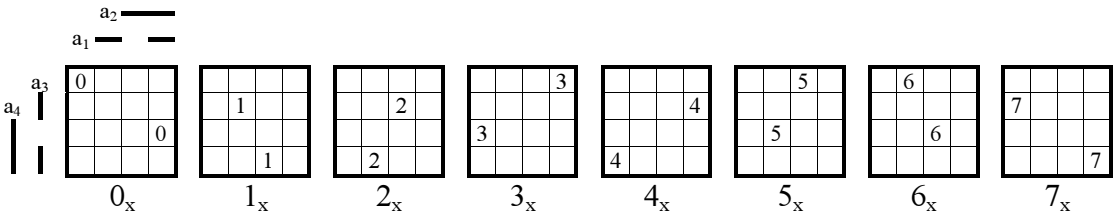


Рис. 4.5.1

Причем в ячейках цифрового пространства, где раньше записывались нами безошибочные штатные цифры информационной части кода, представлены цифры кодовых комбинаций его контрольной части. Все общее цифровое пространство систематического кода имеет при этом восемь слоев: первый слой соответствует номеру кодовой комбинации 0_x, второй – 1_x и т.д.

Следующий шаг предлагаемого алгоритма синтеза заключается в построении геометрических образов исправленных сигналов систематического кода. На рис. 4.5.2, 4.5.3 приведены геометрические образы исправленных разрядов соответственно информационной и контрольной частей этого кода. Процедура построения этих образов выглядит следующим образом.

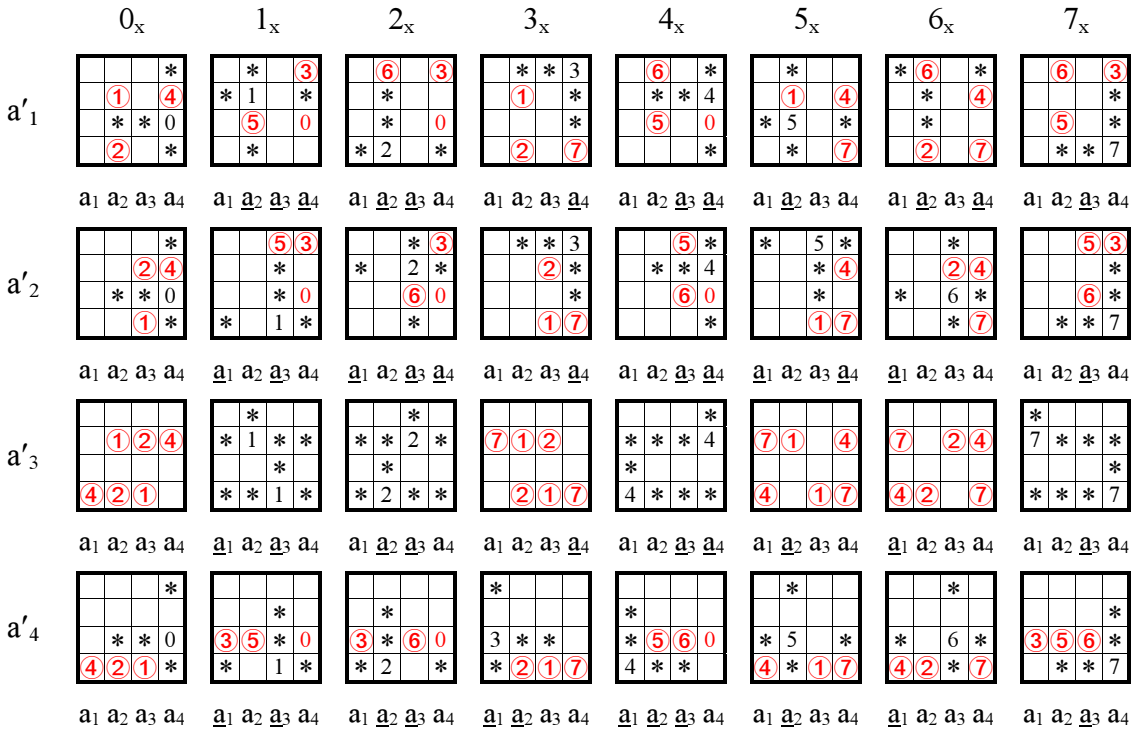


Рис. 4.5.2

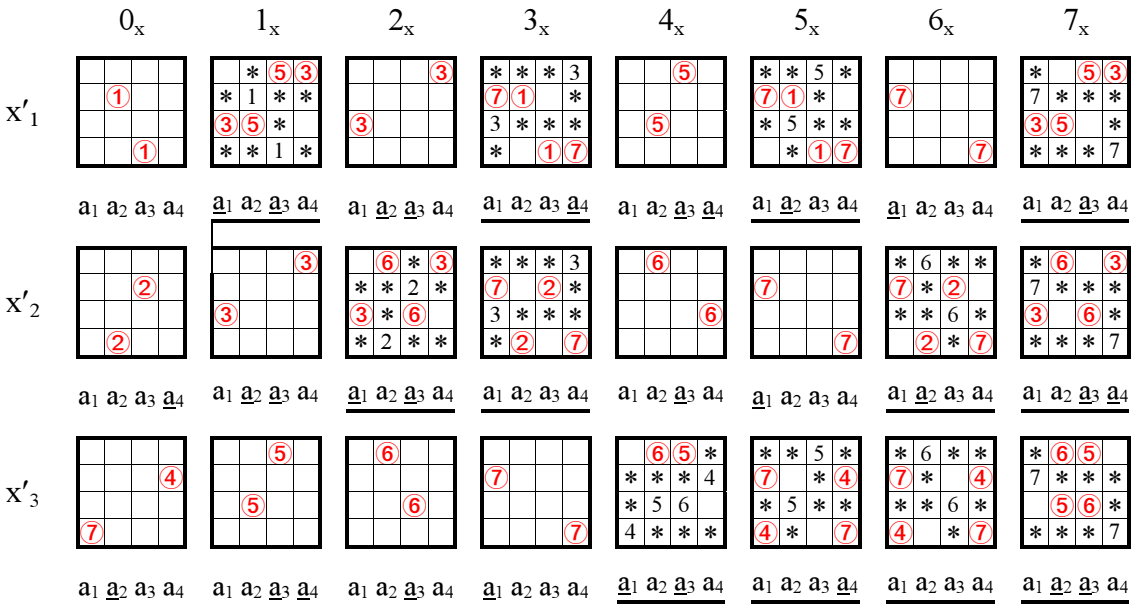


Рис. 4.5.3

Обратимся к разряду a'_1, где в ячейках каждого из слоев $0_x - 7_x$ пространства записан соответствующий им номер кодовой комбинации, который определяется из таких же номеров рис. 4.5.1, лежащих на вертикальных лучах, исходящих из сигналов разряда a_1. Для этого сигнала в каждом из слоев звездочками отмечаются ячейки с одиночными ошибками в информационных координатах $a_1 - a_4$ систематического кода, а номера, выделенные нами цветным шрифтом, определяют одиночные ошибки в контрольных координатах $x_1 - x_3$ систематического кода.

Аналогичным образом строятся геометрические образы исправленных сигналов $a'_2 - a'_4$ остальных разрядов информационной части кода.

Обозначим фигуры и соответствующие им логические выражения в каждом слое под номером 0_x для каждого из сигналов $a'_1 - a'_4$ следующим образом: $F^a_1(a_1, a_2, a_3, a_4)$, $F^a_2(a_1, a_2, a_3, a_4)$, $F^a_3(a_1, a_2, a_3, a_4)$, $F^a_4(a_1, a_2, a_3, a_4)$.

Причем фигуры F^a_1, F^a_2, F^a_4 соответствующими мысленными поворотами относительно осей симметрии цифрового пространства могут переходить одна в другую, а для покрытия каждой из них требуется одинаковая принципиальная схема. Например, если для фигуры $F^a_1(a_1, a_2, a_3, a_4)$ принять прямую последовательность аргументов, то для фигур F^a_2, F^a_4 необходимо в логическом выражении F^a_1 выполнить перестановки этих аргументов, что может быть записано следующим образом: $F^a_2 = F^a_1(a_1 \leftrightarrow a_2)$, $F^a_4 = F^a_1(a_1 \to a_4, a_2 \to a_1, a_4 \to a_2)$.

Однако из-за относительной сложности схемной реализации таких перестановок ограничимся в дальнейшем только такими поворотами относительно

осей симметрии пространства, которые реализуются простым инвертированием сигналов, а представленные выше функции будем определять раздельно:

$$F^a_1(a_1, a_2, a_3, a_4) = a_1\, a_3 \vee a_1\, a_2 \vee a_1\, a_4 \vee a_2\, \underline{a}_3\, a_4;$$
$$F^a_2(a_1, a_2, a_3, a_4) = a_2\, a_3 \vee a_1\, a_2 \vee a_2\, a_4 \vee a_1\, \underline{a}_3\, a_4;$$
$$F^a_3(a_1, a_2, a_3, a_4) = a_1\, a_3\, \underline{a}_4 \vee a_2\, a_3\, \underline{a}_4 \vee \underline{a}_1\, a_3\, a_4 \vee \underline{a}_2\, a_3\, a_4;$$
$$F^a_4(a_1, a_2, a_3, a_4) = a_4\, a_3 \vee a_1\, a_4 \vee a_2\, a_4 \vee a_1\, \underline{a}_3\, a_2. \qquad (4.5.1)$$

Геометрический образ функции в любом из слоёв многомерного пространства сигналов каждого разряда рис. 4.5.2 позволяет его мысленными поворотами получить все остальные. На этом рисунке под каждым из геометрических образов приведена прямая последовательность аргументов функций $F^a_1 - F^a_4$, где соответствующий поворот определяется инвертированием аргументов a_1, a_2, a_3, a_4, а для функции F^a_2 в слоях 1_x, 2_x, 4_x, 7_x дополнительно производится инвертирование самой этой функции, что условно будет отмечаться подчеркиванием всех аргументов функции.

Для построения геометрических образов исправленных сигналов x'_1, x'_2, x'_3 разрядов контрольной части кода рис. 4.5.3 необходимо снова обратиться к многомерному цифровому пространству рис. 4.5.1. Из него необходимо определить слои, содержащие безошибочные кодовые комбинации, для переноса содержимого их ячеек в одноименные слои пространства рис. 4.5.3. Содержимое слоёв первого рисунка переносится во второй рисунок по следующим правилам: для x'_1 это нечетные слои 1_x, 3_x, 5_x, 7_x; для x'_2 – 2_x, 3_x, 6_x, 7_x; для x'_2 – 4_x, 5_x, 6_x, 7_x.

При этом, как и прежде, на рис. 4.5.3 звездочками отмечаются ячейки с одиночными ошибками в информационных координатах систематического кода, а номера, выделенные цветом, определяют одиночные ошибки в контрольных координатах кода. Здесь для определения функций геометрических образов во всех слоях многомерного пространства достаточно определить только одну из них. Пусть это будет функция геометрического образа первого слоя 0_x сигнала x'_1

$$F^x(a_1, a_2, a_3, a_4) = a_1\, \underline{a}_2\, a_3\, \underline{a}_4 \vee \underline{a}_1\, a_2\, a_3\, a_4. \qquad (4.5.2)$$

Функции геометрических образов во всех остальных слоях сигнала x'_1 и слоях сигналов x'_2, x'_3 могут быть получены инвертированием аргументов этой функции либо с одновременным инвертированием самой функции, что соответственно представлено на этом рисунке под каждым геометрическим образом.

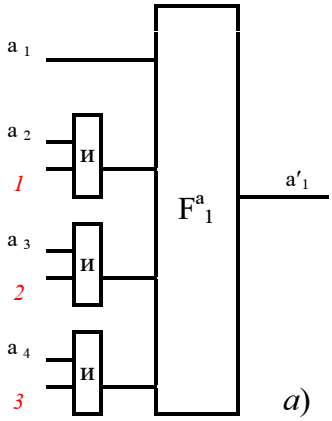
Следовательно, геометрические образы функций слоёв многомерного цифрового пространства рис. 4.5.2, 4.5.3 и соответствующие им функции определяют алгоритм инвертирования аргументов (a_1, a_2, a_3, a_4), (x_1, x_2, x_3) систематического кода, а также алгоритм инвертирования базовых функций F^a_3, F^x , что представлено в табл. 4.5.1. В этой таблице в координатах x_1, x_2, x_3 изображены геометрические образы функций, по сигналам которых выполняется соответствующее инвертирование. Геометрические образы и сами функции пронумерованы в таблице цветным курсивом.

Таблица 4.5.1

	$\underline{a}_1$	$\underline{a}_2$	$\underline{a}_3$	$\underline{a}_4$	$\underline{F}^a_3$, $\underline{F}^x$
	x_2 —— x_1 — — x_3				
a'_1		1	2	3	
a'_2	1		2	4	
a'_3	5	6	2	7	2
a'_4	3	4	2		
x'_1	5	6	2	7	9
x'_2	6	5	2	8	10
x'_3	7	8	2	5	11

Логические выражения (4.5.1) – (4.5.3) позволяют построить весьма простые принципиальные схемы, работающие в режиме реального времени, для исправления одиночных ошибок в информационных (рис. 4.5.4), контрольных (рис. 4.5.5) разрядах систематического кода, где во входных шинах логических блоков F^a_1 – F^a_4, F^x установлены двухвходовые управляемые внешними сигналами инверторы И. При нулевом значении сигналов на их управляющих входах, которые задаются логическими сигналами функций *1–8*, инверторы пропускают сигналы разрядов a_1 – a_4 с входа на выход без изменения, а при логической единице этих функций происходит инвертирование соответствующих сигналов a_1 – a_4. В качестве инверторов используются элементарные логические функции «исключающее ИЛИ» – $a_i\,\underline{n}_j \vee \underline{a}_i\,n_j$ где n_j = *1*, *2*, … , *8*.

К приведенным здесь принципиальным схемам необходимо дать некоторые пояснения относительно формирования функций *1–8*. Если в цифровой системе, где используется систематический код, поставлена только задача исправления ошибок, а её надежность удовлетворяет условиям эксплуатации, то блоки формирования функций *1–8* могут быть общими для всех информационных и контрольных разрядов кода. Для совмещения задач исправления ошибок и резервирования для каждого из этих разрядов необходимо иметь свои принципиальные схемы формирования функций *1–8*.



Таких достаточно простых функций восемь:

$1 = x_1\,\underline{x_2} \vee \underline{x_1}\,x_2$;

$2 = \underline{x_1}\,\underline{x_2}\,x_3 \vee x_1\,\underline{x_2}\,\underline{x_3} \vee \underline{x_1}\,x_2\,\underline{x_3} \vee x_1\,x_2\,x_3$;

$3 = x_1\,\underline{x_3} \vee \underline{x_1}\,x_3$;

$4 = x_2\,\underline{x_3} \vee \underline{x_2}\,x_3$;

$5 = x_1\,\underline{x_2}\,x_3 \vee \underline{x_1}\,x_2\,x_3$;

$6 = x_1\,\underline{x_2}\,x_3 \vee \underline{x_1}\,x_2\,\underline{x_3}$;

$7 = \underline{x_1}\,\underline{x_2}\,x_3 \vee x_1\,x_2\,\underline{x_3}$;

$8 = \underline{x_1}\,\underline{x_2}\,\underline{x_3} \vee x_1\,x_2\,x_3$; (4.5.3)

еще более просты функции $9 = x_1$, $10 = x_2$, $11 = x_3$.

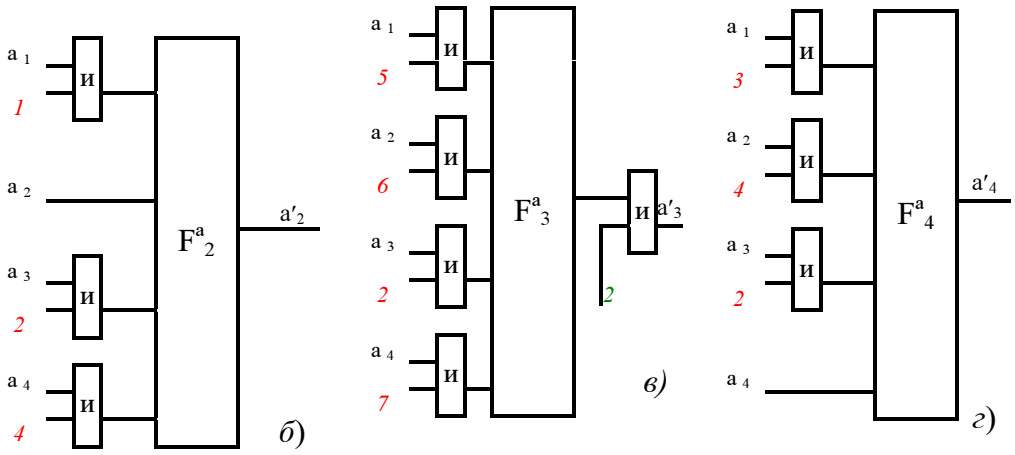


Рис. 4.5.4

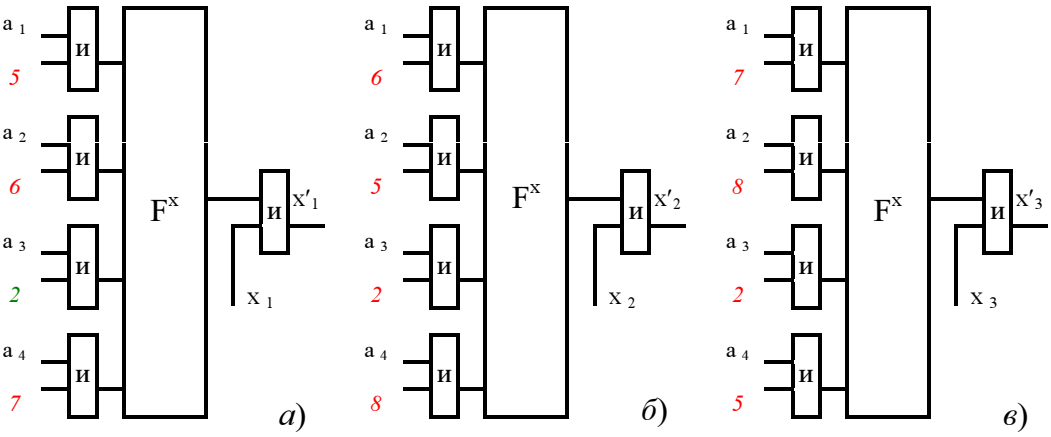


Рис. 4.5.5

Подведем некоторый промежуточный итог рассмотрения представленных выше примеров, которые реализуют алгоритмы синтеза логических и цифровых устройств, использующие понятия теории многомерных цифро-векторных

множеств, не делая разницы между ними. Примеры эти подбирались как можно шире, чтобы проиллюстрировать универсальность этой теории.

При этом нами используется широкое понятие алгоритма как строгой и конечной системы правил, которые определяют последовательность действий над некоторыми реальными объектами – цифрами расширенного натурального ряда чисел, помещаемыми в пронумерованные определенным образом материальные ячейки модели многомерного цифро-векторного пространства.

Это понятие алгоритма несколько отличается от принятого в «чистой» математике, которое представляется «как вычислительный процесс, начинающийся с произвольного исходного данного и направленный на получение полностью определяемого этим исходным данным результата». Однако это отличие не существенно и в конечном итоге построение геометрических образов логических функций и в особенности их покрытие также могут быть представлены «как определенный вычислительный процесс». Более того, единственной областью математики, которая работает с нечисловыми объектами, является геометрия. Поэтому она до настоящего времени не имела возможности опираться на вычислительную интуицию человека, которая исторически всегда использовала понятие алгоритма, и, следовательно, требовала более повышенной строгости рассуждений. Теория многомерных цифро-векторных множеств имеет непосредственную связь с геометрией, а также понятием алгоритма, что позволяет использовать его и в геометрии.

Подобные представления об этих вещах весьма полезны, поскольку ничто не является для нас более наглядным, чем фигура, ибо её можно осязать и видеть.

Декарт

Глава 5

СИСТЕМАТИЧЕСКИЙ КОД С ИСПРАВЛЕНИЕМ ОДИНОЧНЫХ ОШИБОК СИСТЕМЫ СЧИСЛЕНИЯ ОСНОВАНИЯ $n = 2^{11}$

Краткое и постановочное изложение варианта синтеза алгоритма систематических кодов больших оснований систем счисления ($n = 2^{11}, 2^{26}, 2^{57}, \ldots$), исправляющих все одиночные ошибки в информационной и контрольной его частях, которое было представлено в конце третьей главы, может быть здесь полностью завершено. Это стало возможным осуществить после успешного решения подобной задачи в последнем примере предыдущей главы. В нем при синтезе устройства исправления ошибок были учтены законы симметрии цифро-векторного пространства и симметрия промежуточных геометрических образов функций, что позволило получить достаточно простые и быстродействующие схемы для кода основания $n = 2^4$. Можно предвидеть получение таких же положительных результатов для систематических кодов еще больших оснований систем счисления.

5.1. Синтез кода и его кодирующего устройства

Как показано ранее, число таких систематических кодов огромно и даже для $n = 2^{11}$ их невозможно все рассмотреть, да в этом и нет необходимости: достаточно построить один из таких кодов, а все остальные могут быть получены соответствующими поворотами относительно осей симметрии координат информационной части кода.

На рис. 5.1, который аналогичен рис. 3.63, в ячейках пространства координат $a_5, \ldots, a_8$; x_4, $a_9, \ldots, a_{11}$ размещены фигуры $\sigma_1 - \sigma_8$, которые охватывают все варианты размещения цифр 0–15 основания $n = 2^4$ в семимерном пространстве координат $a_1, \ldots, a_4$; x_1, x_2, x_3.

На рис. 5.2, *а* изображены эти фигуры, где первая из них – σ_1 определяется выбранным нами образующим кодом основания $n = 2^4$ и функционально обозначается как $\sigma_1(a_1, \ldots, a_4; x_1, x_2, x_3)$. Все остальные геометрические фигуры определяются мысленными поворотами относительно осей координат x_1, x_2, x_3, что представляется следующим образом:

$\sigma_1(a_1, \ldots, a_4; x_1, x_2, x_3)$, $\sigma_2(a_1, \ldots, a_4; \underline{x}_1, x_2, x_3)$, $\sigma_3(a_1, \ldots, a_4; x_1, \underline{x}_2, x_3)$,
$\sigma_4(a_1, \ldots, a_4; \underline{x}_1, \underline{x}_2, x_3)$, $\sigma_5(a_1, \ldots, a_4; x_1, x_2, \underline{x}_3)$, $\sigma_6(a_1, \ldots, a_4; \underline{x}_1, x_2, \underline{x}_3)$,
$\sigma_7(a_1, \ldots, a_4; x_1, \underline{x}_2, \underline{x}_3)$, $\sigma_8(a_1, \ldots, a_4; \underline{x}_1, \underline{x}_2, \underline{x}_3)$.

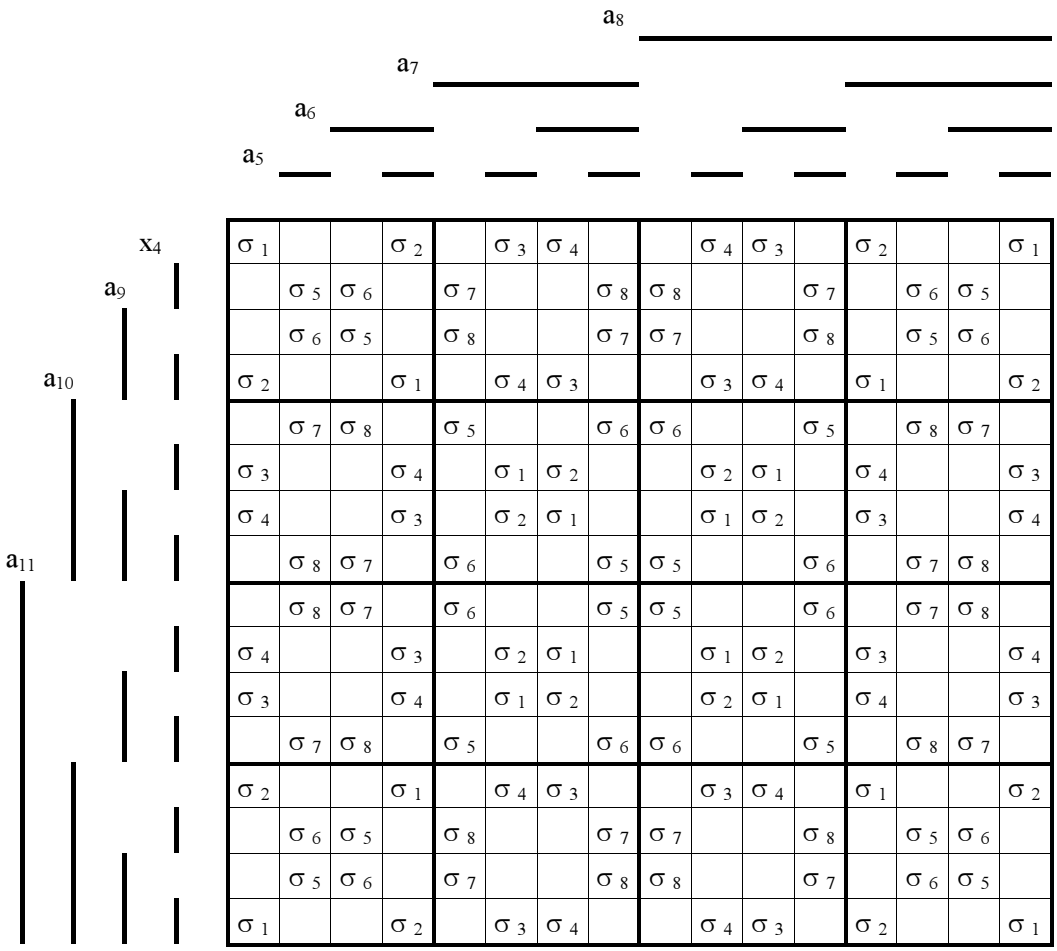
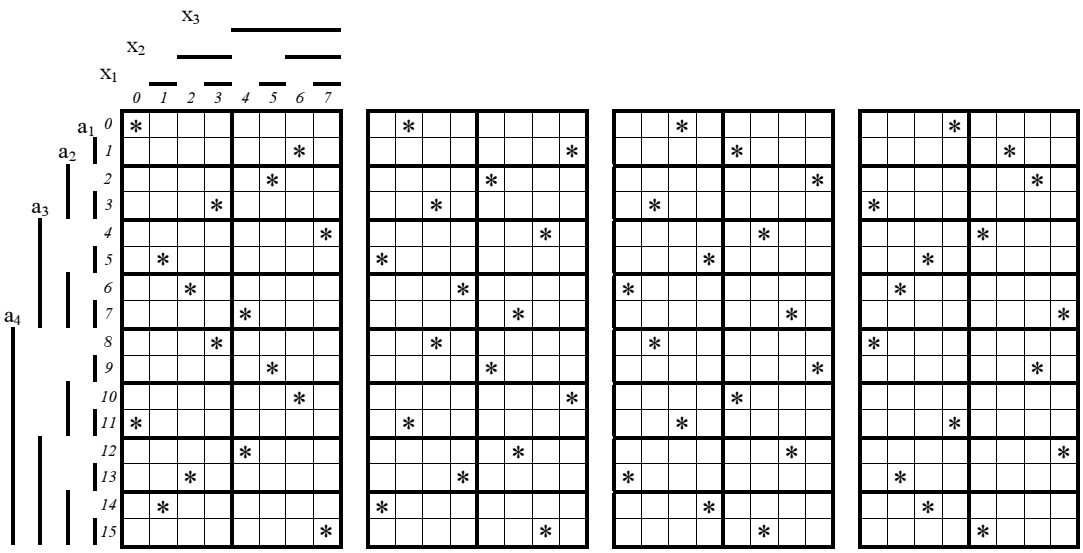


Рис. 5.1

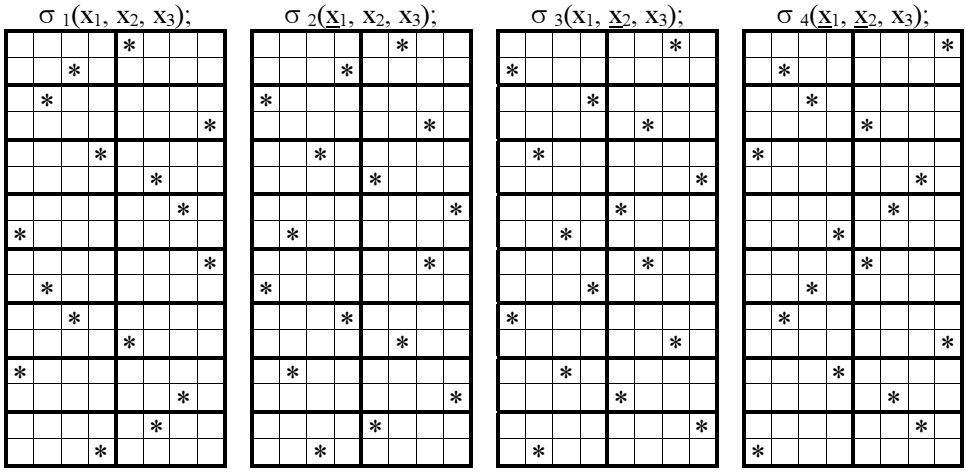
Нетрудно видеть, что все эти «геометрические» фигуры соответствуют кодам основания $n = 2^4$ (рис. 5.2, *б*), которые также являются кодами из табл. 4.2.2 соответственно с координатами 15, 0; 15, 6; 39, 2; 15, 4; 39, 5; 15, 2; 15, 1; 15, 3.

Теперь снова обратимся к рис. 5.1, из которого нам необходимо удалить координату аргумента x_4. Поскольку «весовое» значение этого сигнала равно 8, то выполнить это преобразование не представляет труда.

На рис. 5.3 представлено это весовое преобразование, где в определенных ячейках цифрового пространства координат $a_5, \ldots, a_{11}$ рядом с обозначениями фигур $\sigma_1 - \sigma_8$ установлен вверху знак «плюс» ($^+$). Установка этого знака означает, что фигуры $\sigma_1 - \sigma_8$, которые представляют определенные коды основания $n = 2^4$ цифрами контрольной части систематического кода этого основания, должны увеличиться на восемь единиц.



$\sigma_1(x_1, x_2, x_3)$; $\sigma_2(\underline{x}_1, x_2, x_3)$; $\sigma_3(x_1, \underline{x}_2, x_3)$; $\sigma_4(\underline{x}_1, \underline{x}_2, x_3)$;



$\sigma_5(x_1, x_2, \underline{x}_3)$; $\sigma_6(\underline{x}_1, x_2, \underline{x}_3)$; $\sigma_7(x_1, \underline{x}_2, \underline{x}_3)$; $\sigma_8(\underline{x}_1, \underline{x}_2, \underline{x}_3)$;

а)

0	6	5	3
7	1	2	4
3	5	6	0
4	2	1	7

σ_1; 15,0

1	7	4	2
6	0	3	5
2	4	7	1
5	3	0	6

σ_2; 15,6

2	4	7	1
5	3	0	6
1	7	4	2
6	0	3	5

σ_3; 39,2

3	5	6	0
4	2	1	7
0	6	5	3
7	1	2	4

σ_4; 15,4

4	2	1	7
3	5	6	0
7	1	2	4
0	6	5	3

σ_5; 39,5

5	3	0	6
2	4	7	1
6	0	3	5
1	7	4	2

σ_6; 15,2

6	0	3	5
1	7	4	2
5	3	0	6
2	4	7	1

σ_7; 15,1

7	1	2	4
0	6	5	3
4	2	1	7
3	5	6	0

σ_8; 15,3

б)

Рис. 5.2

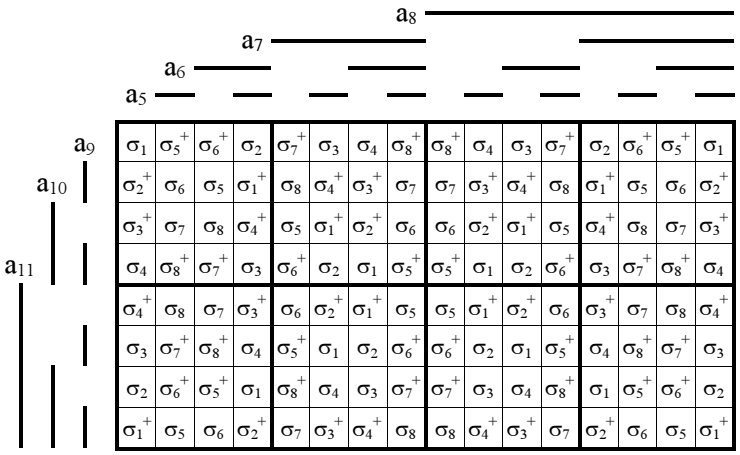


Рис. 5.3

В соответствии с рис. 5.3 полное развернутое представление синтезированного нами кода, исправляющего все одиночные ошибки систематического кода (i = 11, k = 4), изображено на рис. 5.4, *а, б*.

Для определения всех других подобных кодов, выполняющих такие же функции по исправлению одиночных ошибок, необходимо произвести все мысленные повороты относительно осей симметрии цифрового пространства координат $a_1 - a_{11}$.

Полученные таким образом геометрические образы синтезированных кодов весьма значительны: $S = 2^{11}(11!) = 81\ 749\ 606\ 400$. Из этих кодов необходимо будет убрать повторяющиеся коды. Пример такого алгоритма поиска кодов был выполнен нами для основания $n = 2^4$, но повторить его для данного случая практически невозможно из-за огромного числа таких кодов.

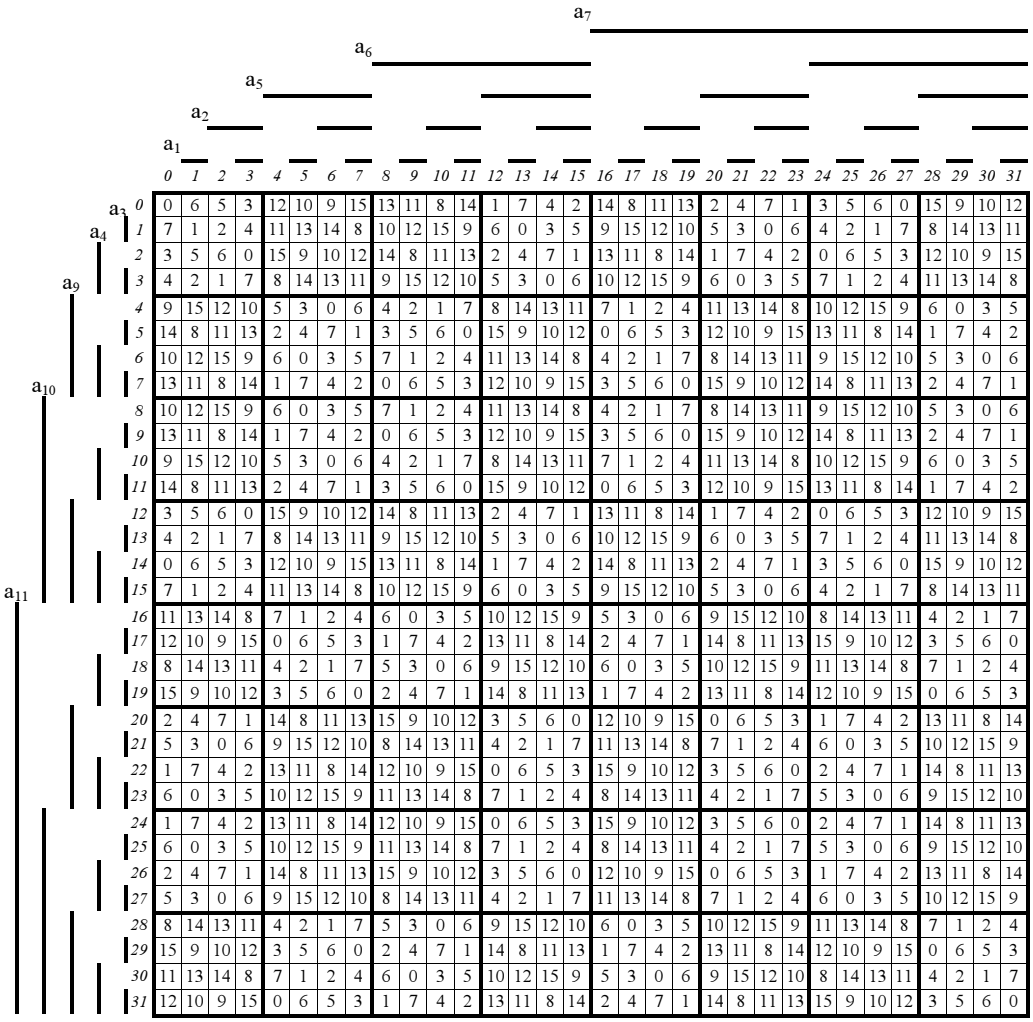
Теперь наступила очередь выполнить операцию кодирования контрольной части кода, когда из сигналов информационных разрядов $a_1, \ldots, a_{11}$ формируются контрольные разряды x_1, x_2, x_3 конкретного систематического кода. Это самая простая задача любого подобного синтеза.

Учитывая, что сигналы контрольных разрядов определяются цифровыми множествами $x_1 = 1 \vee 3 \vee \ldots \vee 15$, $x_2 = (2 \vee 3) \vee (6 \vee 7) \vee (10 \vee 11) \vee (14 \vee 15)$, $x_3 = (4 \vee \ldots \vee 7) \vee (12 \vee \ldots \vee 15)$, $x_4 = (8 \vee \ldots \vee 15)$, на основании рис. 5.4 *а, б* не представляет труда определить геометрические образы этих сигналов в системе координат информационной части кода.

Выполнив эти простые операции по построению геометрических образов сигналов контрольных разрядов в многомерном цифровом пространстве координат $a_1, \ldots, a_{11}$, получим возможность построить структурные схемы формирования соответственно сигналов x_1, x_2, x_3.

В структурных схемах рис. 5.5, *а – г,* которые предназначены для формирования соответственно сигналов x_1, x_2, x_3, в каждом из составляющих их блоков изображены геометрические образы промежуточных логических функций.

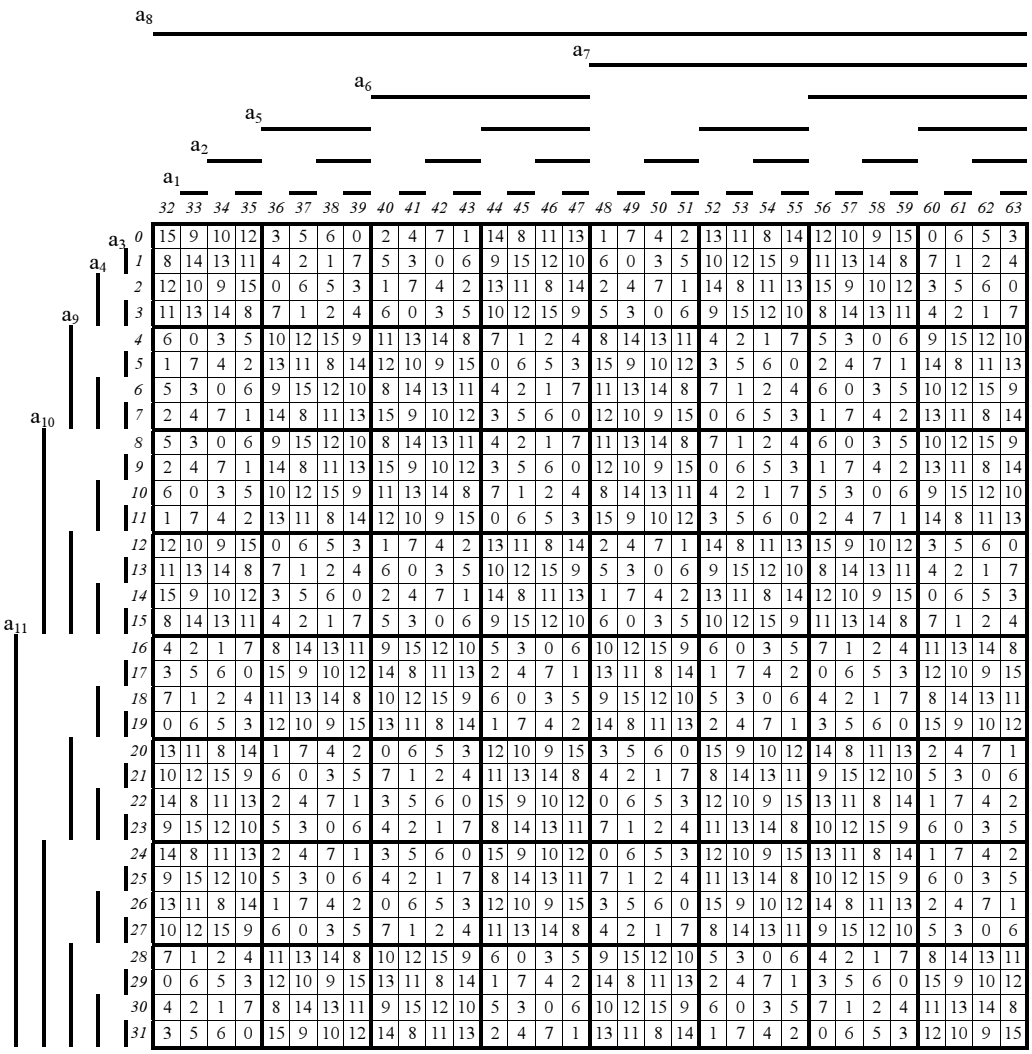
Выбранные геометрические образы и соответствующие им логические функции в этих блоках являются базовыми. Выполняя соответствующий поворот относительно осей симметрии пространства либо выполняя инвертирования выходного сигнала в каждом из этих блоков (выбор того или иного варианта преобразования диктуется оптимизацией аппаратурных затрат), можно реализовать покрытие информационного пространства в координатах a_1, … ,a_{11}, образующих геометрический образ функций x_1, x_2, x_3, что последовательно выполнено на рис. 5.5, *а – г*.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12
1	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11
2	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15
3	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8
4	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5
5	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2
6	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6
7	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1
8	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6
9	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1
10	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5
11	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2
12	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15
13	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8
14	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12
15	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11
16	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7
17	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0
18	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4
19	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3
20	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14
21	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9
22	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13
23	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10
24	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13
25	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10
26	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14
27	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9
28	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4
29	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3
30	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7
31	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0

а)

Рис. 5.4 (начало)



	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3
1	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4
2	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0
3	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7
4	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10
5	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13
6	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9
7	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14
8	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9
9	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14
10	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10
11	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13
12	12	10	9	15	0	6	5	3	1	7	4	2	13	11	8	14	2	4	7	1	14	8	11	13	15	9	10	12	3	5	6	0
13	11	13	14	8	7	1	2	4	6	0	3	5	10	12	15	9	5	3	0	6	9	15	12	10	8	14	13	11	4	2	1	7
14	15	9	10	12	3	5	6	0	2	4	7	1	14	8	11	13	1	7	4	2	13	11	8	14	12	10	9	15	0	6	5	3
15	8	14	13	11	4	2	1	7	5	3	0	6	9	15	12	10	6	0	3	5	10	12	15	9	11	13	14	8	7	1	2	4
16	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8
17	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15
18	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11
19	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12
20	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1
21	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6
22	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2
23	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5
24	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2
25	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5
26	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1
27	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6
28	7	1	2	4	11	13	14	8	10	12	15	9	6	0	3	5	9	15	12	10	5	3	0	6	4	2	1	7	8	14	13	11
29	0	6	5	3	12	10	9	15	13	11	8	14	1	7	4	2	14	8	11	13	2	4	7	1	3	5	6	0	15	9	10	12
30	4	2	1	7	8	14	13	11	9	15	12	10	5	3	0	6	10	12	15	9	6	0	3	5	7	1	2	4	11	13	14	8
31	3	5	6	0	15	9	10	12	14	8	11	13	2	4	7	1	13	11	8	14	1	7	4	2	0	6	5	3	12	10	9	15

б)

Рис. 5.4 (продолжение)

Все промежуточные функции геометрических образов функциональных блоков рис. 5.5, *а – г* просты, и поэтому нет необходимости приводить их аналитическую запись.

Можно потребовать, например, чтобы все принципиальные схемы формирования контрольных разрядов были выполнены двухуровневыми, тогда схемы рис. 5.5, *а – г* будут несколько усложнены (рис. 5.6, *а*).

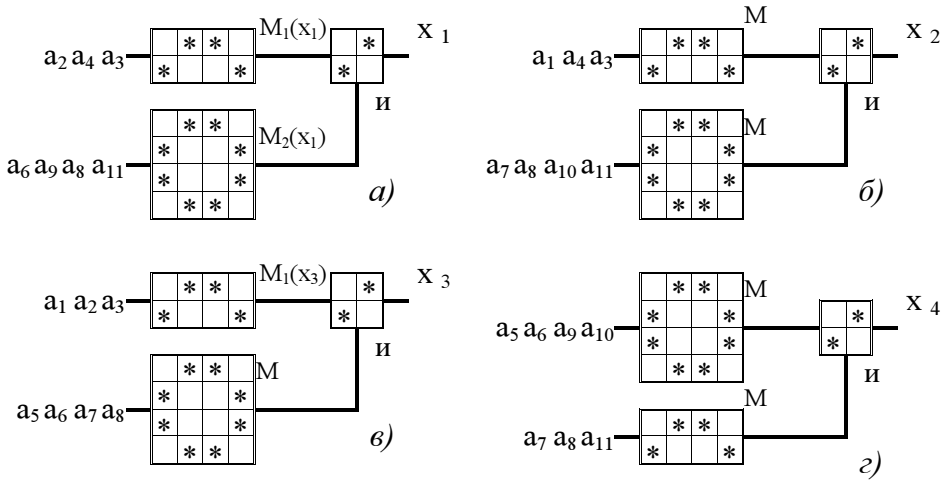
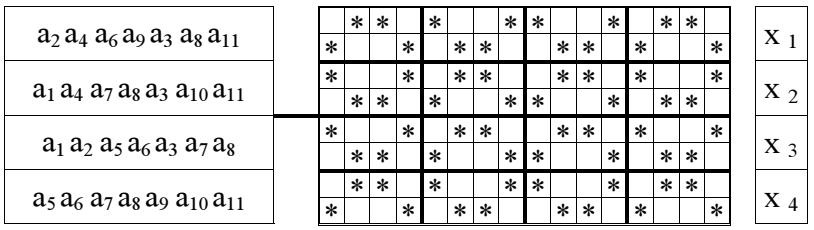
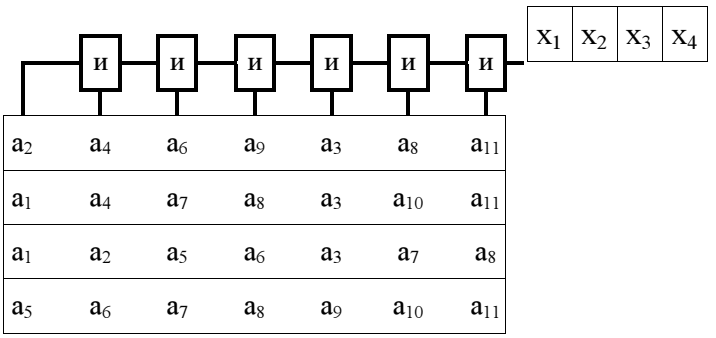


Рис. 5.5



а)



б)

Рис. 5.6

При этом здесь используется функциональная одинаковая схема во всех разрядах шифратора, а на входные шины каждого из них подаются разные ансамбли входных сигналов информационных разрядов кода, что приведено на групповой схеме рис. 5.6, *а*.

Минимальные аппаратурные затраты достигаются в многоуровневом варианте (см. рис. 5.6, *б*), где приведена групповая комбинационная схема шифратора.

Дальнейшее желание уменьшить аппаратурные затраты приведет к необходимости применить здесь конечный синхронный автомат, где используется последовательный пошаговый режим формирования сигналов x_1, x_2, x_3, x_4 с передачей их данных в регистр памяти. Реализация такого автомата значительно упрощается тем, что определенные группы множеств содержат подмножества, которые определенными поворотами относительно осей симметрии пространства переходят в этих множествах одно в другое.

Вполне очевидно, что такой синтез устройства будет сопровождаться определенным снижением быстродействия выполнения операции кодирования, а на современном уровне технологии изготовления интегральных схем вряд ли целесообразно выполнять такой синтез для рассматриваемой нами системы счисления основания $n = 2^{11}$.

5.2. Исправление одиночных ошибок в информационной части кода

Для пояснения алгоритма декодирования сигналов информационной части кода необходимо предварительно остановиться на принципе построения геометрических образов исправленных сигналов разрядов информационной части кода в объединенном цифровом пространстве координат систематического кода $a_1, \ldots, a_{11}, x_1 - x_4$.

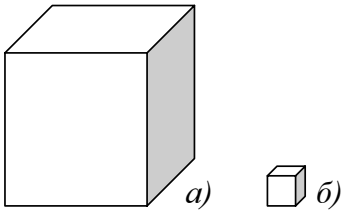


Рис. 5.7

Раздельное изображение многомерного цифрового пространства информационной и контрольной частей кода (рис. 5.7, *а*, *б*) представляет собой объединение для информационной части ячеек, которые последовательно нумеруются определенным жестким образом цифрами от 0_a до $(2^i - 1)_a$, а для контрольной части кода число таких ячеек будет 2^k с нумерацией ячеек подобным образом от 0_x до $(2^k - 1)_x$. Для рассматриваемого основания системы счисления число ячеек информационной части кода – 2048, а контрольной части – 16.

В таком систематическом коде каждому номеру $0_x, \ldots, 15_x$ контрольной части кода ставится в соответствие $2^{(i-k)} = 128$ номеров информационной части этого кода.

Объединенное цифровое пространство координат систематического кода (рис. 5.8) состоит из шестнадцати ($0_x, \ldots, 15_x$) пространств, каждое из которых равно по числу ячеек информационному пространству рис. 5.7, *а*, где в каждом из них содержится 128 ячеек, предназначенных

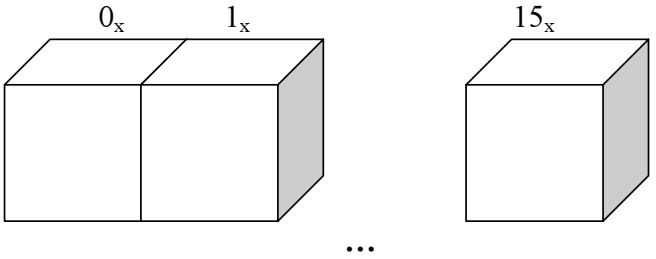


Рис. 5.8

для размещения номеров ячеек информационной части пространства. Таким образом, 2048 номеров ячеек информационной части кода разносятся по шестнадцати цифровым пространствам емкостью 2048 ячеек каждое.

Сжатым изображением этого объединенного цифрового пространства является информационное пространство, где в каждой его ячейке записан номер ячейки пространства контрольной части кода. В нашем случае это пространство изображено на рис. 5.4, *а*, *б*, которые представляют соответственно два слоя ячеек в трехмерных координатах a_1, a_2, a_5, a_6, a_7; a_3, a_4, a_9, a_{10}, a_{11}; a_8, где в каждой ячейке записана одна из цифр 0_x, …, 15_x контрольной части кода.

Для универсального цифрового пространства информационной части кода, когда все его 2048 ячеек заполнены логическими единицами, они будут перенесены в эквивалентное объединенное пространство рис. 5.8. Оставшиеся незаполненные ячейки пространства этого рисунка предназначены для размещения ошибок в информационных и контрольных разрядах систематического кода. В случае исправления всех одиночных ошибок все ячейки пространства также будут заполнены номерами кодовых комбинаций одиночных ошибок.

Известно, что каждый сигнал разрядов информационной части кода $a_1 – a_{11}$ имеет свой геометрический образ в пространстве этих координат, а эти геометрические образы весьма просты и представляют собой объединение ячеек, лежащих на вертикальных или горизонтальных линиях, исходящих из сигналов этих разрядов. При этом номера контрольных цифр, лежащие внутри этих геометрических образов сигналов $a_1 – a_{11}$, переносятся в ячейки с одинаковыми информационными номерами, которые расположены в цифровых пространствах под номерами 0_x, …, 15_x.

Подобным образом в объединенном цифровом пространстве координат систематического кода формируется безошибочная часть геометрического образа сигналов разрядов $a_1 – a_{11}$, а после выполнения этой операции наступает этап определения ячеек, в которые перемещаются кодовые комбинации при возникновении одиночных ошибок. Эта операция достаточно подробно изложена в гл. 3, широко использовалась в примерах гл. 4 и поэтому не требует дополнительных пояснений. Этим этапом завершается формирование геометрических образов исправления сигналов в информационных разрядах кода. Причем подробное представление этих простых и понятных геометрических преобразований требует большого объема материала (352 с.), и поэтому здесь не приводится.

Ниже будут приведены конечные результаты представленных шагов такого алгоритма раздельно для сигналов разрядов $a_1 – a_4$; a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, а также представлен синтез принципиальных схем исправления одиночных ошибок в этих разрядах.

1. Исправление одиночных ошибок в информационных разрядах $a_1 – a_4$

Геометрический образ исправленных сигналов $a'_1 – a'_4$ информационной части систематического кода представлен в объединенном цифровом пространстве координат этого кода на рис. 5.9, *а – г*.

На каждом из этих рисунков изображены в пространстве координат $a_5 – a_{11}$ соответственно под номерами $0_x – 3_x$; $4_x – 7_x$; $8_x – 11_x$; $12_x – 15_x$ части этого геометрического образа. На 1/16 всего цифрового пространства, которая представлена под номером 0_x, выделен цветом геометрический образ базовой функции. Многократными мысленными поворотами геометрического образа этой базовой функции относительно осей симметрии пространства координат a_5, a_6, a_9, a_{10} осуществляется покрытие геометрических образов сигналов.

Под каждыми геометрическими образами $0_x – 3_x$; $4_x – 7_x$; $8_x – 11_x$; $12_x – 15_x$ приведены соответствующие таблицы, где записаны эти повороты, осуществляемые путем инвертирования сигналов разрядов a_5, a_6, a_9, a_{10}.

В ячейках пространств рис. 5.9, *а – г* для каждого из сигналов $a'_1 – a'_4$ записаны соответствующие логические функции – сигналов a_i (i = 1, 2, 3, 4); $m_1 – m_8$, последние из которых различны для каждого из сигналов $a_1 – a_4$.

На рис. 5.10 изображены геометрические образы функций $m_1 – m_8$ для каждого из разрядов $a_1 – a_4$, где все функции m_1 выбраны нами базовыми, а остальные в каждом из разрядов могут быть получены мысленными поворотами относительно осей симметрии пространства координат $a_1 – a_4$. Здесь под каждым геометрическим образом функций $m_1 – m_8$ дается запись этих поворотов, которые реализуются посредством инвертирования сигналов $a_1 – a_4$.

Из рис. 5.9, *а – г* очевидно, что необходимые повороты относительно осей пространства координат a_5, a_6, a_9, a_{10} являются общими для всех сигналов разрядов $a_1 – a_4$. Геометрические образы функций $M(\underline{a}_5)$, $M(\underline{a}_6)$, $M(\underline{a}_9)$, $M(\underline{a}_{10})$, по которым осуществляется инвертирование сигналов a_5, a_6, a_9, a_{10} с целью покрытия геометрических образов исправленных сигналов $a'_1 – a'_4$, приведены в соответствующих координатах пространств на рис. 5.11, *а – г*.

Принципиальные схемы, реализующие функции $M(\underline{a}_5)$, $M(\underline{a}_6)$, $M(\underline{a}_9)$, $M(\underline{a}_{10})$, могут быть выполнены как двухуровневыми, так и многоуровневыми. Поэтому на этих рисунках дополнительно к двухуровневому варианту приведен один из возможных вариантов их реализации в многоуровневом исполнении, который уменьшает затраты оборудования с одновременным снижением быстродействия. В многоуровневом варианте нами используются управляемые инверторы, которые по управляющему сигналу осуществляют инверсию выходных сигналов функций соответствующих блоков. Именно такой вариант представления принципиальных схем блоков для исправления ошибок будет использоваться нами в дальнейшем. Это связано с необходимостью уменьшения представляемого графического материала, а переход к быстродействующим двухуровневым схемам, если возникает такая необходимость, не представляет каких-либо сложностей.

$a'_1 - a'_4$

a_5 a_6 a_7 a_8 a_9 a_{10} a_{11}

m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1

0_x, 1_x, 2_x, 3_x

$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$
$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$
$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$
$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$
$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$

0_x, 1_x, 2_x, 3_x

а)

Рис. 5.9 (начало)

$a'_1 - a'_4$; a_5; a_6; a_7; a_8; a_9; a_{10}; a_{11}

4_x	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
5_x	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
6_x	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
7_x	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1

4_x	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
5_x	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
6_x	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$
7_x	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$

б)

Рис. 5.9 (продолжение)

$a'_1 - a'_4$

a_5 a_6 a_7 a_8

8_x	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
9_x	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
10_x	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7
	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
11_x	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1
	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8
	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2
	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1
	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1
	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1
	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1
	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7

8_x	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$
	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ a_9\ a_{10}$	$\underline{a}_5\ a_6\ a_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$
9_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$
	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$
10_x	$a_5\ a_6\ a_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$
	$\underline{a}_5\ \underline{a}_6\ a_9\ a_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ a_6\ a_9\ a_{10}$
11_x	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$	$a_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a}_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ a_{10}$

в)

Рис. 5.9 (продолжение)

$a'_1 - a'_4$; a_5; a_6; a_7; a_8; a_9; a_{10}; a_{11}

12_x	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
13_x	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
14_x	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
15_x	a_1	m_7	m_8	a_1	m_2	a_1	a_1	m_1	m_1	a_1	a_1	m_2	a_1	m_8	m_7	a_1
	m_5	a_1	a_1	m_6	a_1	m_4	m_3	a_1	a_1	m_3	m_4	a_1	m_6	a_1	a_1	m_5
	m_3	a_1	a_1	m_4	a_1	m_6	m_5	a_1	a_1	m_5	m_6	a_1	m_4	a_1	a_1	m_3
	a_1	m_1	m_2	a_1	m_8	a_1	a_1	m_7	m_7	a_1	a_1	m_8	a_1	m_2	m_1	a_1
	m_4	a_1	a_1	m_3	a_1	m_5	m_6	a_1	a_1	m_6	m_5	a_1	m_3	a_1	a_1	m_4
	a_1	m_2	m_1	a_1	m_7	a_1	a_1	m_8	m_8	a_1	a_1	m_7	a_1	m_1	m_2	a_1
	a_1	m_8	m_7	a_1	m_1	a_1	a_1	m_2	m_2	a_1	a_1	m_1	a_1	m_7	m_8	a_1
	m_6	a_1	a_1	m_5	a_1	m_3	m_4	a_1	a_1	m_4	m_3	a_1	m_5	a_1	a_1	m_6

12_x	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ a_9\ a_{10}$
	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$
13_x	$a_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ a_9\ a_{10}$
	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$
14_x	$a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ a_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$
	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ a_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$
15_x	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$
	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$

г)

Рис. 5.9 (окончание)

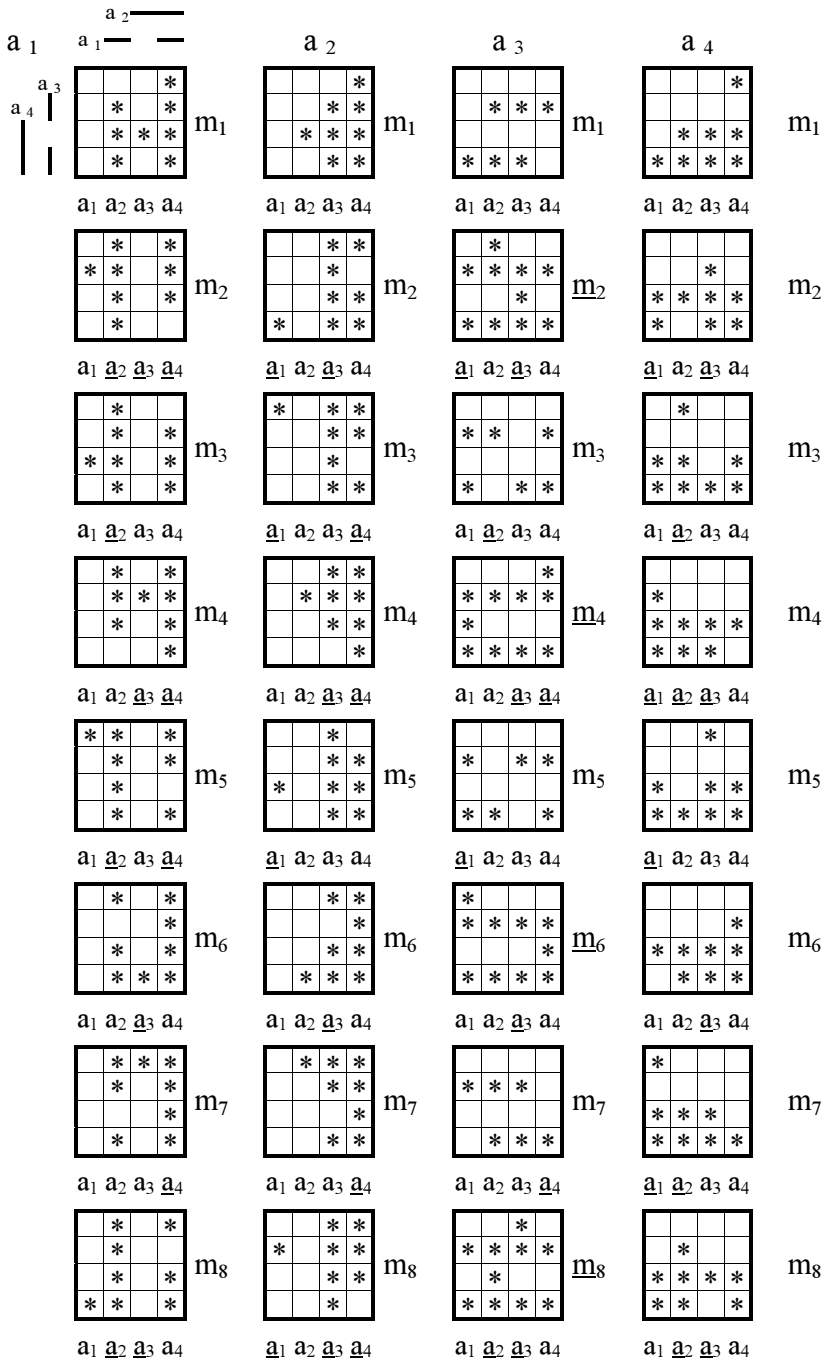


Рис. 5.10

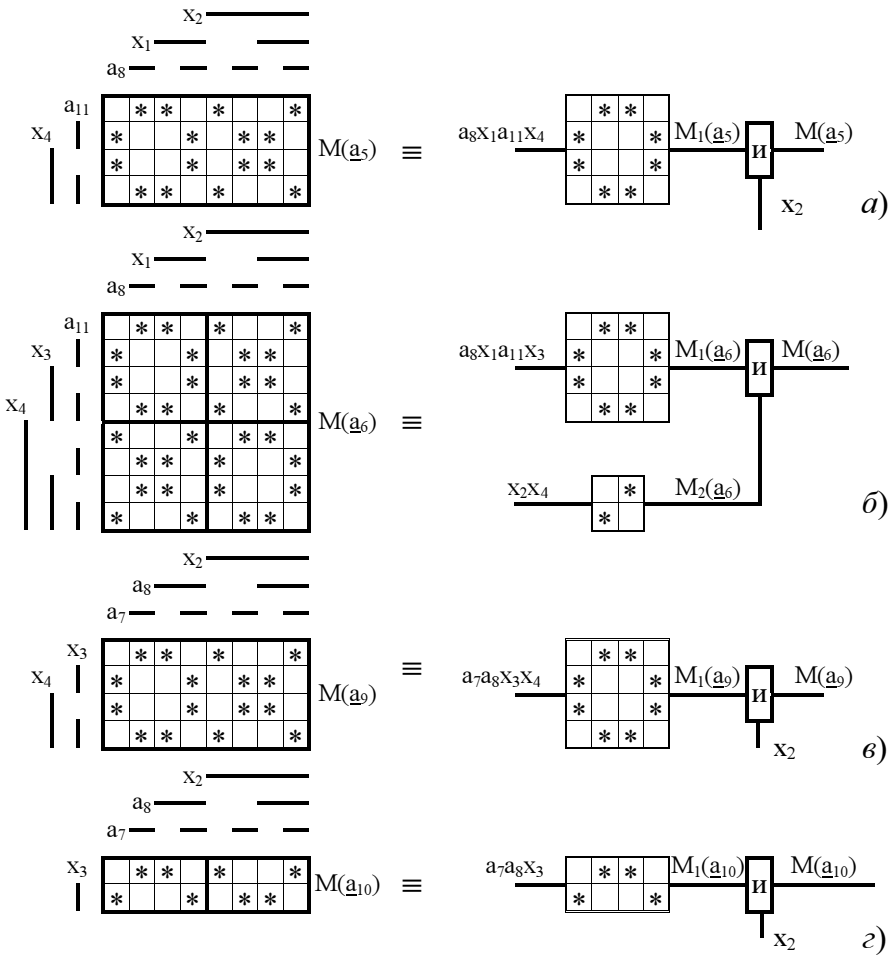


Рис. 5.11

Геометрические фигуры рис. 5.9, *а – г*, 5.11, которые представляют собой определенные логические выражения, позволяют выполнить покрытие геометрических образов исправленных сигналов разрядов $a'_1 – a'_4$. Эти геометрические образы располагаются в координатах $a_1 – a_4$; $x_1 – x_4$ общего цифрового пространства систематического кода, а их покрытие производится многократными поворотами относительно осей симметрии цифрового пространства с координатами a_5, a_6, a_9, a_{10} базового геометрического образа (БГО) в этих же координатах.

Функциональная схема, осуществляющая требуемое покрытие, приведена на рис. 5.12. В этой схеме все необходимые для реализации этого покрытия повороты базового геометрического образа сигналов разрядов $a_1 – a_4$ выполняются соответствующим инвертированием сигналов a_5, a_6, a_9, a_{10}, в координатах которых расположен этот базовый геометрический образ.

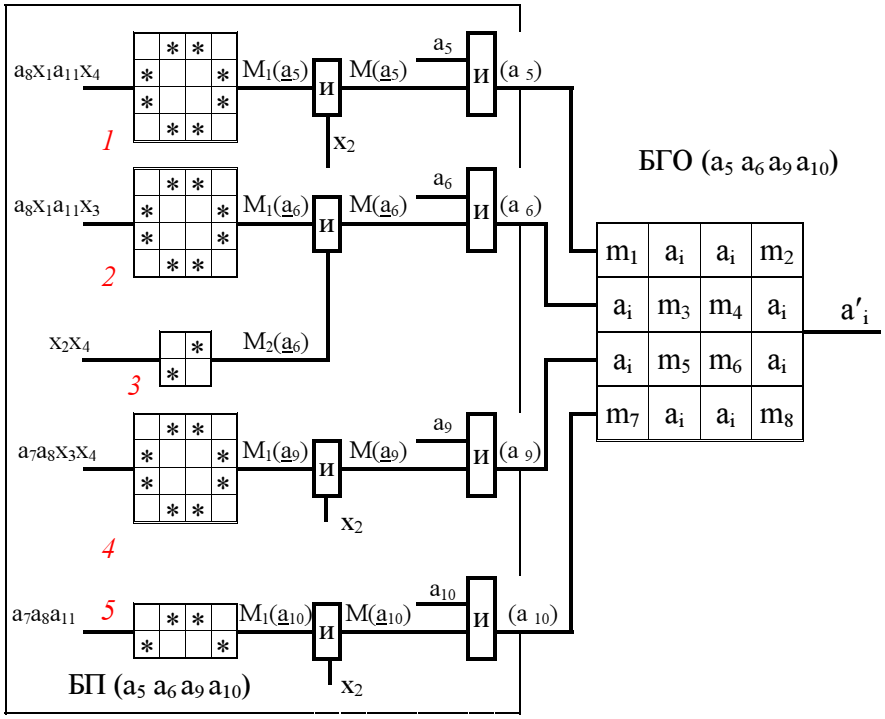


Рис. 5.12

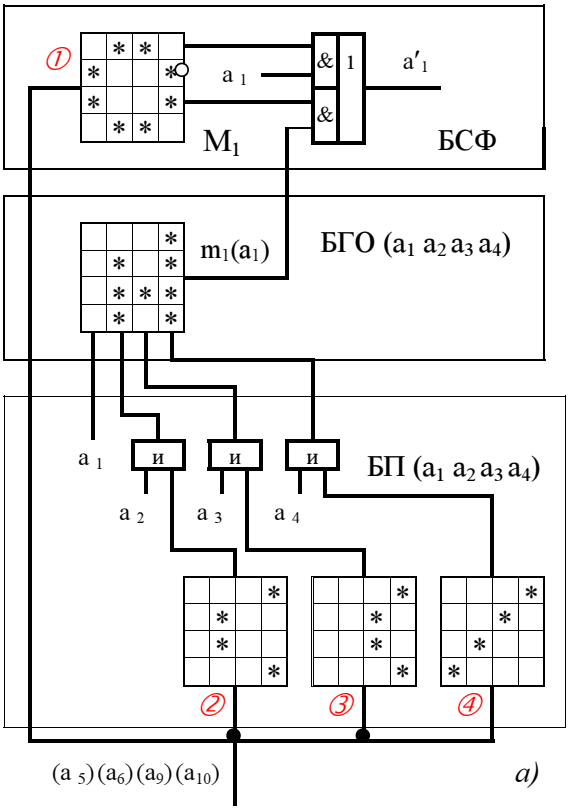


Рис. 5.13 (начало)

Схема рис. 5.12 является общей для всех сигналов разрядов a_1 – a_4, где функциональная базовая схема для каждого сигнала a'_1 – a'_4 будет различной. Эти функциональные базовые схемы для каждого из сигналов a'_1 – a'_4 приведены соответственно на рис. 5.13, *а, б, в, г*.

В свою очередь каждая функциональная базовая схема этого рисунка содержит свой базовый элемент, геометрический образ которого $m_1(a_i)$ в координатах a_1 – a_4 используется для получения других геометрических образов аналогичных функций $m_2(a_i)$ – $m_8(a_i)$ в этих же координатах (рис. 5.10).

Это покрытие выполняется посредством соответствующего поворота относительно осей цифрового пространства a_5, a_6, a_9, a_{10}.

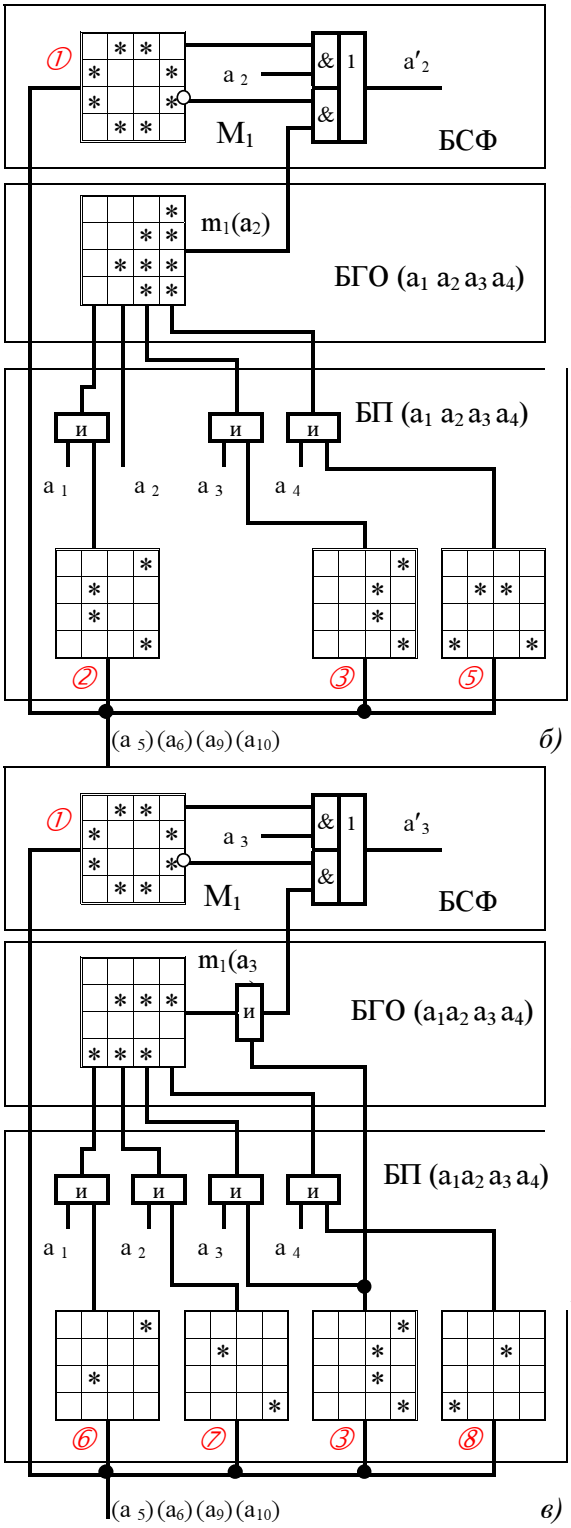


Рис. 5.13(продолжение)

В каждой функциональной схеме рис. 5.13, *а, б, в, г* имеется одинаковый функциональный элемент $M(a_i)$, который определяет размещение функций $m_1(a_i)$ – $m_8(a_i)$, а также сигналов a_i (i = 1, 2, 3, 4) в координатах a_5, a_6, a_9, a_{10} пространства. Совместное использование функции $M(a_i)$ и функций $m_1(a_i)$ – $m_8(a_i)$, представленных в координатах a_1 – a_4, позволяет посредством подачи их выходных сигналов на входные шины элементарной логической схемы 2И-ИЛИ получить исправление всех одиночных ошибок сигналов a'_1 – a'_4 первых четырех разрядов систематического кода.

Таким образом, используя базовые геометрические образы соответствующих функций в координатах двух цифровых пространств (a_1, a_2, a_3, a_4; a_5, a_6, a_9, a_{10}) и выполняя повороты относительно осей симметрии этих пространств, осуществляется покрытие геометрических образов исправления всех одиночных ошибок сигналов первых четырех разрядов информационной части систематического кода, расположенных в общей системе координат этого кода ёмкостью 2^{15}.

В этих функциональных многоуровневых схемах весьма просто реализуется, как увеличение числа уровней, так и их уменьшение теоретически вплоть до двухуровневой схемы, которая в данном случае будет весьма громоздкой.

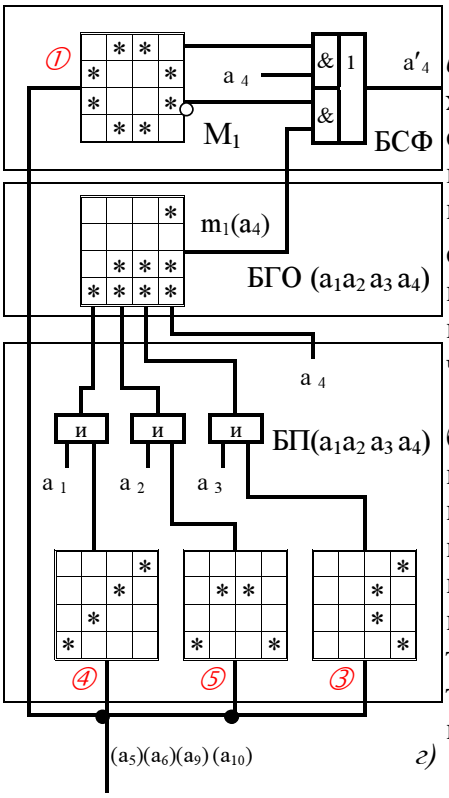


Рис. 5.13(окончание)

Функциональные блоки на рис. 5.13, *а, б, в, г*, входами которых являются синхронно изменяющиеся инвертированием сигналы разрядов a_5, a_6, a_9, a_{10}, могут быть выполнены общими для первых четырех информационных разрядов систематического кода. Сквозная нумерация ① – ⑧ курсивом таких функциональных блоков на этих рисунках, позволяет определить их число в этой части устройства.

Очевидно, что другие функциональные блоки, например, рис. 5.12 могут также иметь одинаковые геометрические образы при одинаковых координатах с аналогичными блоками в других информационных и контрольных разрядах систематического кода. Ответ на этот вопрос можно дать только в конце решения нашей задачи синтеза всего устройства исправления одиночных ошибок.

Кроме отмеченного выше функционального базового геометрического образа БГО($a_5a_6a_9a_{10}$) в координатах цифрового пространства $a_5a_6a_9a_{10}$ в устройстве исправления одиночных ошибок разрядов a_1 – a_4 имеются следующие функциональные блоки: 1. на рис. 5.12 блок поворотов БП ($a_5a_6a_9a_{10}$) относительно осей симметрии цифрового пространства координат a_7 a_8 a_{11}; x_1 –x_4 2. на каждом из рис. 5.13 *а, б, в, г* по три блока – блок поворота БП (a_1 –a_4) относительно осей симметрии пространства координат a_1 –a_4; блок базового геометрического образа БГО в координатах пространства a_1 –a_4; блок совмещения функций БСФ.

Синтез всех этих блоков, кроме последнего блока, очевиден из изложенного выше. В задачу блока БСФ входит передача на выходные шины разряда a'_i (i = 1, 2, 3, 4) устройства сигналов a_i либо сигналов функций $m_1(a_i)$ – $m_8(a_i)$, которые определяются из геометрических образов сигналов a'_1 – a'_4 рис. 5.9, *а, б, в, г* функциями M_1, $m_1(a_i)$ по следующему логическому выражению

$$a'_i = M_1\, a_i \vee \underline{M}_1\, m_1(a_i).$$

2. Исправление одиночных ошибок в пятом и последующих информационных разрядах

Синтез алгоритма устройства исправления всех одиночных ошибок в пятом разряде и во всех последующих информационных разрядах кода начина-

ется с построения геометрического образа соответствующего исправленного сигнала $a'_5 - a'_{11}$ в объединенном цифровом пространстве координат $a_1 - a_{11}$; $x_1 - x_4$. Пример такого построения для сигнала a'_5 представлен на рис. 5.15 *а, б, в, г*. Построение подобных геометрических образов было выполнено нами ранее для сигналов $a'_1 - a'_4$ и поэтому детали этого построения будут опущены.

Также как и прежде, на каждом рис. 5.15, *а, б, в, г* изображены в пространстве координат $a_5 - a_{11}$ соответственно под номерами $0_x - 3_x$, $4_x - 7_x$, $8_x - 11_x$, $12_x - 15_x$, части этого геометрического образа. На 1/16 всего цифрового пространства, которая представлена под номером 0_x, выделен цветом геометрический образ базовой функции. Многократными мысленными поворотами геометрического образа этой базовой функции относительно осей симметрии пространства координат a_5, a_6, a_9, a_{10} осуществляется покрытие геометрического образа сигнала a'_5.

Под геометрическими образами всех этих частей цифрового пространства ($0_x - 3_x$; $4_x - 7_x$; $8_x - 11_x$; $12_x - 15_x$) приведены такие же части пространства, в ячейках которых зафиксированы эти повороты, осуществляемые посредством инвертирования сигналов разрядов a_5, a_6, a_9, a_{10}.

В пространстве координат a_5, a_6, a_9, a_{10} геометрического образа базовой функции имеются пустые, незаполненные ячейки (логические нули), ячейки заполненные логическими единицами (*), а также прямые и инверсные значения функций в системе координат пространства $a_1 - a_4$.

Последние представлены здесь в сокращенной форме записи и являют собой геометрические образы функций, которые изображены на рис. 5.14.

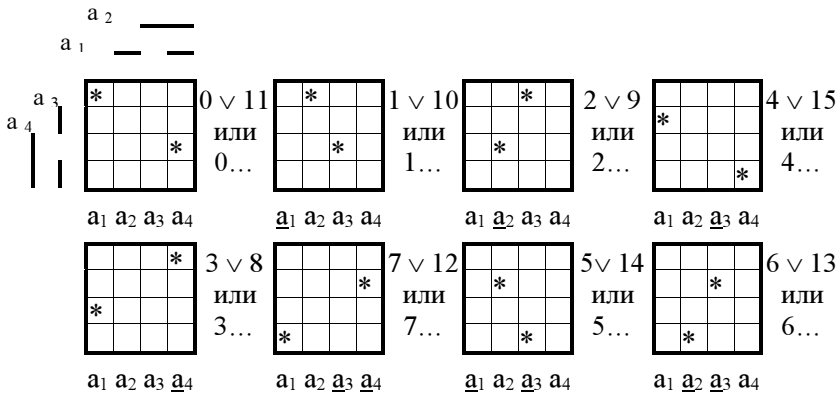


Рис. 5.14

В свою очередь каждая из этих функций может быть получена из другой мысленными поворотами относительно осей симметрии координат $a_1 - a_4$ этого пространства. На рис. 5.15 в качестве такой базовой исходной функции выбрана первая функция

$$0 \ldots = 0 \vee 11 = \underline{a}_1\, \underline{a}_2\, \underline{a}_3\, \underline{a}_4 \vee a_1\, a_2\, \underline{a}_3\, a_4,$$

а все остальные функции рис. 5.15 будут формироваться нами соответствующими поворотами относительно осей симметрии этого цифрового пространства.

a'_5 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11}

0_x		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
1_x		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
2_x		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
3_x		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…

a_7 a_8 a_{11}

0_x	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$
	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$
1_x	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
2_x	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$
	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$
3_x	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$
	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$

а)

Рис. 5.15 (начало)

a'_5 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11}

4_x		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
5_x		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
6_x		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
7_x		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…

a_7 a_8 a_{11}

4_x	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$
	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$
5_x	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
6_x	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
7_x	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$
	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$

б)

Рис. 5.15 (продолжение)

a'_5 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11}

8_x	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
9_x	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
10_x	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
11_x	4…	*		1…		2…	7…	*		7…	2…	*	1…	*		4…
		6…	3…	*	0…	*		5…	5…	*		0…		3…	6…	*
		5…	0…	*	3…	*		6…	6…	*		3…		0…	5…	*
	7…	*		2…		1…	4…	*		4…	1…	*	2…	*		7…
		0…	5…	*	6…	*		3…	3…	*		6…		5…	0…	*
	2…	*		7…		4…	1…	*		1…	4…	*	7…	*		2…
	1…	*		4…		7…	2…	*		2…	7…	*	4…	*		1…
		3…	6…	*	5…	*		0…	0…	*		5…		6…	3…	*

a_7 a_8 a_{11}

8_x	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
11_x	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$

в)

Рис. 5.15 (продолжение)

a'_5 a_8 a_7 a_6 a_5

	a_9															
12_x	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
a_{10}		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
a_{11}	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
13_x	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
14_x	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
	3…	*		6…	7…	*		2…	2…	*		7…	6…	*		3…
15_x	3…	*		6…		5…	0…	*		0…	5…	*	6…	*		3…
		1…	4…	*	7…	*		2…	2…	*		7…		4…	1…	*
		2…	7…	*	4…	*		1…	1…	*		4…		7…	2…	*
	0…	*		5…		6…	3…	*		3…	6…	*	5…	*		0…
		7…	2…	*	1…	*		4…	4…	*		1…		2…	7…	*
	5…	*		0…		3…	6…	*		6…	3…	*	0…	*		5…
	6…	*		3…		0…	5…	*		5…	0…	*	3…	*		6…
		4…	1…	*	2…	*		7…	7…	*		2…		1…	4…	*

a_8 a_7

	a_{11}			
12_x	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
13_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
14_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
15_x	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$

г)

Рис. 5.15 (окончание)

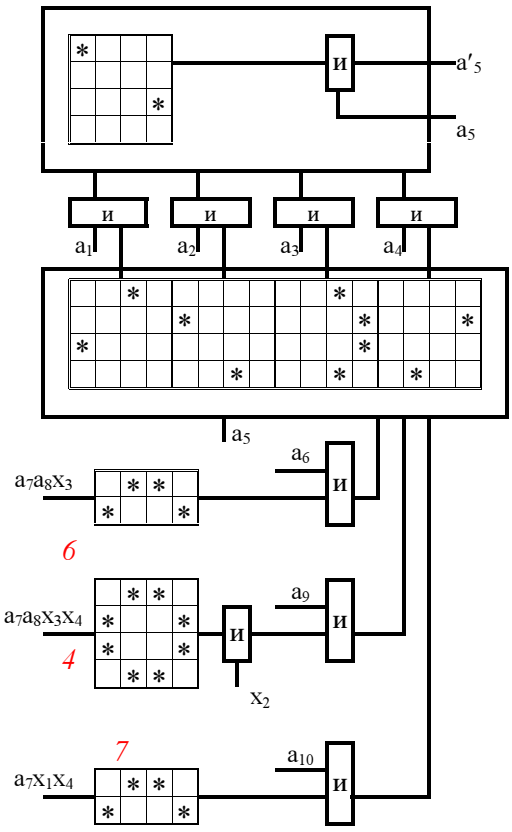


Рис. 5.16

Всё сказанное в последнем абзаце относительно сокращенной формы записи функций в системе координат пространства $a_1 - a_4$, а также сами эти функции и их геометрические образы (см. рис. 5.14) с возможностью соответствующим поворотом относительно осей симметрии этого пространства переходить одна в другую, а также выбором исходной базовой функции (0 …), будет нами полностью распространено и использовано в дальнейшем рассмотрении устройств исправления одиночных ошибок во всех остальных информационных разрядах.

Функциональная схема исправления одиночных ошибок в пятом информационном разряде (рис. 5.16) состоит из последовательно соединенных двух блоков поворота БП (a_5 a_6 a_9 a_{10}), БП ($a_1 - a_4$), которые посредством инвертирования соответствующих им разрядов осуществляют необходимые покрытия геометрического образа функции a'_5 (рис. 5.15, *а – г*) в объединенном цифровом пространстве координат $a_1 - a_{11}$; $x_1 - x_4$ базовым геометрическим образом функции $0 \vee 11$ (см. рис. 5.14) в блоке БГО ($a_1 - a_4$).

В отличие от функциональной схемы исправления одиночных ошибок в разрядах $a_1 - a_4$ (см. рис. 5.13, *а – г*) здесь отсутствует блок совмещения функций БСФ, а роль этого блока выполняет элементарный инвертор И, на управляющий вход которого поступает сигнал a_5. Инвертирование выходного сигнала блока БГО ($a_1 - a_4$) происходит, когда $a_5 = 1^*$, что полностью соответствует необходимости заполнения ячеек пространства координат a_5, a_6, a_9, a_{10} логическими единицами и инверсными значениями функций от ($0 \vee 11$) до ($7 \vee 12$) в системе координат $a_1 - a_4$ (см. рис. 5.13, *а – г*).

Исходными данными для этого синтеза во всех остальных устройствах исправления ошибок в последующих информационных разрядах служат геометрические образы исправленных сигналов $a'_6 - a'_{11}$ (рис. 5.17, *а – г* – 5.22, *а – г*). На основании этих геометрических образов на рис. 5.23 – 5.28 синтезированы функциональные схемы исправления одиночных ошибок в этих разрядах.

a'_6								a_8								
				a_7												
		a_6														
	a_5															
0_x a_{10} a_{11}	a_9	5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
1_x		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
2_x		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
3_x		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…

		a_8			
	a_7				
0_x a_{11}	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
1_x	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
2_x	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
3_x	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$

а)

Рис. 5.17 (начало)

a_8

a'_6 a_7

a_6

a_5

a_9

4_x a_{10}

a_{11}

	2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
	1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
	4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
	7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…

5_x

	7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
	4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
	1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
	2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…

6_x

	4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
	7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
	2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
	1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…

7_x

	1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
	2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
	7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
	4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…

a_8

a_7

a_{11}

$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$

4_x

$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$

5_x

$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$

6_x

$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$

7_x

б)

Рис. 5.17 (продолжение)

a'_6 a_8 a_7 a_6 a_5

	a_9															
8_x	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
9_x	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
	2…		*	7…		4…	1…	*		1…	4…	*	7…		*	2…
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
10_x	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
11_x	1…		*	4…		7…	2…	*		2…	7…	*	4…		*	1…
		3…	6…	*	5…		*	0…	0…		*	5…		6…	3…	*
		0…	5…	*	6…		*	3…	3…		*	6…		5…	0…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
		5…	0…	*	3…		*	6…	6…		*	3…		0…	5…	*
	7…		*	2…		1…	4…	*		4…	1…	*	2…		*	7…
	4…		*	1…		2…	7…	*		7…	2…	*	1…		*	4…
		6…	3…	*	0…		*	5…	5…		*	0…		3…	6…	*

a_8 a_7 a_{11}

8_x	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
11_x	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$

в)

Рис. 5.17 (продолжение)

a_8

a_7

a'_6

a_6

a_5

12_x	5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
		7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
		4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
	6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
		1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
	3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
	0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
		2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
13_x	0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
		2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
		1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
	3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
		4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
	6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
	5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
		7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
14_x	3…		*	6…		5…	0…	*		0…	5…	*	6…		*	3…
		1…	4…	*	7…		*	2…	2…		*	7…		4…	1…	*
		2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
	0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
		7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
	5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
	6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
		4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
15_x	6…		*	3…		0…	5…	*		5…	0…	*	3…		*	6…
		4…	1…	*	2…		*	7…	7…		*	2…		1…	4…	*
		7…	2…	*	1…		*	4…	4…		*	1…		2…	7…	*
	5…		*	0…		3…	6…	*		6…	3…	*	0…		*	5…
		2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
	0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…
		2…	7…	*	4…		*	1…	1…		*	4…		7…	2…	*
	0…		*	5…		6…	3…	*		3…	6…	*	5…		*	0…

a_8

a_7

12_x	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
13_x	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
14_x	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
15_x	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$

г)

Рис. 5.17 (окончание)

								a_8								
a'_7						a_7										
				a_6												
			a_5													
0_x	a_9		6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
	a_{10}	4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
		7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
	a_{11}		5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
		2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
			0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
			3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
		1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
			3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
1_x		1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
		2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
			0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
		7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
			5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
			6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
		4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
			0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
2_x		2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
		1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
			3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
		4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
			6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
			5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
		7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
			5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
3_x		7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
		4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
			6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
		1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
			3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
			0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
		2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…

			a_8		
		a_7			
0_x	a_{11}	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
		$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
1_x		$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
		$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
2_x		$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
3_x		$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
		$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$

а)

Рис. 5.18 (начало)

4_x		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
5_x	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
6_x	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
7_x	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…

4_x	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
5_x	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
6_x	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
7_x	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$

б)

Рис. 5.18 (продолжение)

a'_7 a_8 a_7 a_6 a_5

	a_9															
8_x	1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
a_{10}		3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
		0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
a_{11}	2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
		5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
	7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
	4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
		6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
9_x	4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
		6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
		5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
	7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
		0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
	2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
	1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
		3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
10_x	7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
		5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*
		6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
	4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
		3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
	1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
	2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
		0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
11_x	2…			7…	*	4…	1…	*		1…	4…		7…	*	*	2…
		0…	5…		6…	*	*	3…	3…			6…	*	5…	0…	*
		3…	6…		5…	*	*	0…	0…			5…	*	6…	3…	*
	1…			4…	*	7…	2…	*		2…	7…		4…	*	*	1…
		6…	3…		0…	*	*	5…	5…			0…	*	3…	6…	*
	4…			1…	*	2…	7…	*		7…	2…		1…	*	*	4…
	7…			2…	*	1…	4…	*		4…	1…		2…	*	*	7…
		5…	0…		3…	*	*	6…	6…			3…	*	0…	5…	*

a_8 a_7

	a_{11}			
8_x	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
11_x	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$

в)

Рис. 5.18 (продолжение)

a_8

a'_7 a_7

a_6

a_5

12_x a_9	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
a_{10}		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
a_{11}	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
13_x	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
14_x	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
15_x	5…			0…	*	3…	6…	*		6…	3…		0…	*	*	5…
		7…	2…		1…	*	*	4…	4…			1…	*	2…	7…	*
		4…	1…		2…	*	*	7…	7…			2…	*	1…	4…	*
	6…			3…	*	0…	5…	*		5…	0…		3…	*	*	6…
		1…	4…		7…	*	*	2…	2…			7…	*	4…	1…	*
	3…			6…	*	5…	0…	*		0…	5…		6…	*	*	3…
	0…			5…	*	6…	3…	*		3…	6…		5…	*	*	0…
		2…	7…		4…	*	*	1…	1…			4…	*	7…	2…	*

a_8

a_7

12_x a_{11}	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
13_x	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
14_x	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
15_x	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$

г)

Рис. 5.18 (окончание)

a'_8	a_5	a_6	a_7	a_8												
0_x	a_9		3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
	a_{10}	1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
		2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
	a_{11}		0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
		7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
			5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
			6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
		4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
1_x			6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
		4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
		7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
			5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
		2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
			0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
			3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
		1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
2_x			5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
		7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
		4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
			6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
		1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
			3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
			0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
		2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
3_x			0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
		2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
		1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
			3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
		4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
			6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
			5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
		7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…

	a_7	a_8		
0_x	a_{11} $F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$
	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$
1_x	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$
	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
2_x	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ a_9\ a_{10})$
	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
3_x	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$
	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$

а)

Рис. 5.19 (начало)

a_8

a'_8 a_7

a_6

a_5

	a_9															
4_x		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
a_{10}	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
a_{11}		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
5_x	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
6_x	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
7_x	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…

a_8

a_7

	a_{11}			
4_x	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$
	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$
5_x	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$
	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
6_x	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ a_9\ a_{10})$
	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
7_x	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$
	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$

б)

Рис. 5.19 (продолжение)

a_8

a'_8 a_7

a_6

a_5

	a_9															
8_x	4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
a_{10}		6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
		5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
a_{11}	7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
		0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
	2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
	1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
		3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
	1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
9_x		3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
		0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
	2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
		5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
	7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
	4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
		6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
	2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
10_x		0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*
		3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
	1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
		6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
	4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
	7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
		5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
	7…			2…		1…	4…		*	4…	1…	*	2…	*	*	7…
11_x		5…	0…		3…			6…	6…	*	*	3…	*	0…	5…	*
		6…	3…		0…			5…	5…	*	*	0…	*	3…	6…	*
	4…			1…		2…	7…		*	7…	2…	*	1…	*	*	4…
		3…	6…		5…			0…	0…	*	*	5…	*	6…	3…	*
	1…			4…		7…	2…		*	2…	7…	*	4…	*	*	1…
	2…			7…		4…	1…		*	1…	4…	*	7…	*	*	2…
		0…	5…		6…			3…	3…	*	*	6…	*	5…	0…	*

a_8

a_7

	a_{11}			
8_x	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$
	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$
11_x	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$

в)

Рис. 5.19 (продолжение)

a_8

a'_8 a_7

a_6

a_5

12_x a_9	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
a_{10}		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
a_{11}	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
13_x	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
14_x	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
15_x	0…			5…		6…	3…		*	3…	6…	*	5…	*	*	0…
		2…	7…		4…			1…	1…	*	*	4…	*	7…	2…	*
		1…	4…		7…			2…	2…	*	*	7…	*	4…	1…	*
	3…			6…		5…	0…		*	0…	5…	*	6…	*	*	3…
		4…	1…		2…			7…	7…	*	*	2…	*	1…	4…	*
	6…			3…		0…	5…		*	5…	0…	*	3…	*	*	6…
	5…			0…		3…	6…		*	6…	3…	*	0…	*	*	5…
		7…	2…		1…			4…	4…	*	*	1…	*	2…	7…	*

a_8

a_7

12_x a_{11}	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
13_x	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
14_x	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
15_x	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$

г)

Рис. 5.19 (окончание)

									a_8							
a'_9							a_7									
						a_6										
					a_5											
0_x	a_9		2…	7…		4…			1…	1…			4…		7…	2…	
	a_{10}	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
		3…			6…		5…	0…			0…	5…		6…			3…
	a_{11}	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
		6…			3…		0…	5…			5…	0…		3…			6…
		*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
			7…	2…		1…			4…	4…			1…		2…	7…	
		5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
			7…	2…		1…			4…	4…			1…		2…	7…	
1_x		5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
		6…			3…		0…	5…			5…	0…		3…			6…
		*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
		3…			6…		5…	0…			0…	5…		6…			3…
		*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
			2…	7…		4…			1…	1…			4…		7…	2…	
		0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
			4…	1…		2…			7…	7…			2…		1…	4…	
2_x		6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
		5…			0…		3…	6…			6…	3…		0…			5…
		*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
		0…			5…		6…	3…			3…	6…		5…			0…
		*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
			1…	4…		7…			2…	2…			7…		4…	1…	
		3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
			1…	4…		7…			2…	2…			7…		4…	1…	
3_x		3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
		0…			5…		6…	3…			3…	6…		5…			0…
		*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
		5…			0…		3…	6…			6…	3…		0…			5…
		*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
			4…	1…		2…			7…	7…			2…		1…	4…	
		6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…

			a_8		
		a_7			
0_x	a_{11}	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
1_x		$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
2_x		$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
3_x		$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
		$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$

а)

Рис. 5.20 (начало)

a'_9																
4_x		5…	0…		3…			6…	6…			3…		0…	5…	
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	4…			1…		2…	7…			7…	2…		1…			4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	1…			4…		7…	2…			2…	7…		4…			1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
5_x		0…	5…		6…			3…	3…			6…		5…	0…	
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	1…			4…		7…	2…			2…	7…		4…			1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	4…			1…		2…	7…			7…	2…		1…			4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
6_x		3…	6…		5…			0…	0…			5…		6…	3…	
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	2…			7…		4…	1…			1…	4…		7…			2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	7…			2…		1…	4…			4…	1…		2…			7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
7_x		6…	3…		0…			5…	5…			0…		3…	6…	
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	7…			2…		1…	4…			4…	1…		2…			7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	2…			7…		4…	1…			1…	4…		7…			2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…

4_x	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
5_x	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
6_x	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
7_x	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$

б)

Рис. 5.20 (продолжение)

a'_9 a_8 a_7 a_6 a_5

8_x a_9	5…			0…		3…	6…			6…	3…		0…			5…
a_{10}	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
		4…	1…		2…			7…	7…			2…		1…	4…	
a_{11}	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…			5…		6…	3…			3…	6…		5…			0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
9_x	0…			5…		6…	3…			3…	6…		5…			0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
		4…	1…		2…			7…	7…			2…		1…	4…	
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…			0…		3…	6…			6…	3…		0…			5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
10_x	3…			6…		5…	0…			0…	5…		6…			3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…			3…		0…	5…			5…	0…		3…			6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
11_x	6…			3…		0…	5…			5…	0…		3…			6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…			6…		5…	0…			0…	5…		6…			3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*

a_8 a_7 a_{11}

8_x	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
9_x	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
11_x	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$

в)

Рис. 5.20 (продолжение)

a'_9								a_8								
				a_7												
		a_6														
	a_5															
12_x	a_9	2…			7…		4…	1…			1…	4…		7…			2…
	a_{10}	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
			3…	6…		5…			0…	0…			5…		6…	3…	
	a_{11}	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
			6…	3…		0…			5…	5…			0…		3…	6…	
		4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
		7…			2…		1…	4…			4…	1…		2…			7…
		*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
13_x		7…			2…		1…	4…			4…	1…		2…			7…
		*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
			6…	3…		0…			5…	5…			0…		3…	6…	
		4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
			3…	6…		5…			0…	0…			5…		6…	3…	
		1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
		2…			7…		4…	1…			1…	4…		7…			2…
		*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
14_x		4…			1…		2…	7…			7…	2…		1…			4…
		*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
			5…	0…		3…			6…	6…			3…		0…	5…	
		7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
			0…	5…		6…			3…	3…			6…		5…	0…	
		2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
		1…			4…		7…	2…			2…	7…		4…			1…
		*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
15_x		1…			4…		7…	2…			2…	7…		4…			1…
		*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
			0…	5…		6…			3…	3…			6…		5…	0…	
		2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
			5…	0…		3…			6…	6…			3…		0…	5…	
		7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
		4…			1…		2…	7…			7…	2…		1…			4…
		*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*

		a_8			
		a_7			
12_x	a_{11}	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2\ (a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
13_x		$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
		$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2\ (a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
14_x		$F_2\ (a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
		$F_2\ (\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$
15_x		$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2\ (a_5\ a_6\ a_9\ a_{10})$
		$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2\ (\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$

г)

Рис. 5.20 (окончание)

a_8

a_7

a'_{10}

a_6

a_5

	a_9															
0_x		1…	4…		7…			2…	2…			7…		4…	1…	
a_{10}	3…			6…		5…	0…			0…	5…		6…			3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
a_{11}	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
		4…	1…		2…			7…	7…			2…		1…	4…	
1_x	6…			3…		0…	5…			5…	0…		3…			6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
		7…	2…		1…			4…	4…			1…		2…	7…	
2_x	5…			0…		3…	6…			6…	3…		0…			5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	3…			6…		5…	0…			0…	5…		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
		2…	7…		4…			1…	1…			4…		7…	2…	
3_x	0…			5…		6…	3…			3…	6…		5…			0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…

a_8

a_7

	a_{11}			
0_x	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
1_x	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
2_x	$F_2\ (a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_1\ (a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
3_x	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2\ (a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1\ (a_5\ a_6\ \underline{a}_9\ a_{10})$

а)

Рис. 5.21 (начало)

a_8

a_7

a'_{10}

a_6

a_5

4_x a_9		6…	3…		0…			5…	5…			0…		3…	6…	
a_{10}	4…			1…		2…	7…			7…	2…		1…			4…
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
a_{11}	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
		3…	6…		5…			0…	0…			5…		6…	3…	
5_x	1…			4…		7…	2…			2…	7…		4…			1…
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
		0…	5…		6…			3…	3…			6…		5…	0…	
6_x	2…			7…		4…	1…			1…	4…		7…			2…
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
		5…	0…		3…			6…	6…			3…		0…	5…	
7_x	7…			2…		1…	4…			4…	1…		2…			7…
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…

a_8

a_7

4_x a_{11}	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$
5_x	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
6_x	$F_2\ (a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
	$F_1\ (a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
7_x	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2\ (a_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (a_5\ \underline{a}_6\ a_9\ a_{10})$

б)

Рис. 5.21 (продолжение)

a_8

a_7

a'_{10}

a_6

a_5

8_x a_9	6…			3…		0…	5…			5…	0…		3…			6…
a_{10}		4…	1…		2…			7…	7…			2…		1…	4…	
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
a_{11}	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…			5…		6…	3…			3…	6…		5…			0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
9_x	3…			6…		5…	0…			0…	5…		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
10_x	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
		4…	1…		2…			7…	7…			2…		1…	4…	
	6…			3…		0…	5…			5…	0…		3…			6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
11_x	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…			6…		5…	0…			0…	5…		6…			3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*

a_8

a_7

8_x a_{11}	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1\ (a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
10_x	$F_2\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
	$F_1\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
11_x	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$

в)

Рис. 5.21 (продолжение)

a_8

a'_{10} a_7

a_6

a_5

12_x a_9	1…			4…		7…	2…			2…	7…		4…			1…
a_{10}		3…	6…		5…			0…	0…			5…		6…	3…	
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
a_{11}	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…			1…		2…	7…			7…	2…		1…			4…
13_x		6…	3…		0…			5…	5…			0…		3…	6…	
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	7…			2…		1…	4…			4…	1…		2…			7…
14_x		5…	0…		3…			6…	6…			3…		0…	5…	
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…			7…		4…	1…			1…	4…		7…			2…
15_x		0…	5…		6…			3…	3…			6…		5…	0…	
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*

a_8

a_7

12_x a_{11}	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$
	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1\ (a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$
13_x	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1\ (a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$
14_x	$F_2\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_1\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
15_x	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_2\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_1\ (\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$

г)

Рис. 5.21 (окончание)

								a_8								
	a'_{11}					a_7										
				a_6												
			a_5													
0_x	a_9		4…	1…		2…			7…	7…			2…		1…	4…	
	a_{10}	6…			3…		0…	5…			5…	0…		3…			6…
		5…			0…		3…	6…			6…	3…		0…			5…
	a_{11}		7…	2…		1…			4…	4…			1…		2…	7…	
		0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
		*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
		*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
		3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
1_x			1…	4…		7…			2…	2…			7…		4…	1…	
		3…			6…		5…	0…			0…	5…		6…			3…
		0…			5…		6…	3…			3…	6…		5…			0…
			2…	7…		4…			1…	1…			4…		7…	2…	
		5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
		*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
		*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
		6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
2_x			2…	7…		4…			1…	1…			4…		7…	2…	
		0…			5…		6…	3…			3…	6…		5…			0…
		3…			6…		5…	0…			0…	5…		6…			3…
			1…	4…		7…			2…	2…			7…		4…	1…	
		6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
		*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
		*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
		5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
3_x			7…	2…		1…			4…	4…			1…		2…	7…	
		5…			0…		3…	6…			6…	3…		0…			5…
		6…			3…		0…	5…			5…	0…		3…			6…
			4…	1…		2…			7…	7…			2…		1…	4…	
		3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
		*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
		*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
		0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…

			a_8		
		a_7			
0_x	a_{11}	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$
		$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$
1_x		$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ a_6\ a_9\ a_{10})$
		$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ a_{10})$
2_x		$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$
		$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$
3_x		$F_1(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$
		$F_2(a_5\ a_6\ \underline{a_9}\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ a_9\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}})$

а)

Рис. 5.22 (начало)

a_8

a_7

a'_{11}

a_6

a_5

4_x a_9		3…	6…		5…			0…	0…			5…		6…	3…	
a_{10}	1…			4…		7…	2…			2…	7…		4…			1…
	2…			7…		4…	1…			1…	4…		7…			2…
a_{11}		0…	5…		6…			3…	3…			6…		5…	0…	
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
		6…	3…		0…			5…	5…			0…		3…	6…	
5_x	4…			1…		2…	7…			7…	2…		1…			4…
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
		5…	0…		3…			6…	6…			3…		0…	5…	
6_x	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
		0…	5…		6…			3…	3…			6…		5…	0…	
7_x	2…			7…		4…	1…			1…	4…		7…			2…
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…

a_8

a_7

4_x a_{11}	$F_1(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$
	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$
5_x	$F_1(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
	$F_2(\underline{a_5}\ a_6\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ a_9\ \underline{a_{10}})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
6_x	$F_1(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$
	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$	$F_2(\underline{a_5}\ \underline{a_6}\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(a_5\ \underline{a_6}\ \underline{a_9}\ a_{10})$
7_x	$F_1(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_1(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_1(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$
	$F_2(a_5\ \underline{a_6}\ a_9\ \underline{a_{10}})$	$F_2(a_5\ a_6\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10})$	$F_2(\underline{a_5}\ a_6\ a_9\ \underline{a_{10}})$

б)

Рис. 5.22 (продолжение)

a'_{11} a_8 a_7 a_6 a_5

8_x a_9	3…			6…		5…	0…			0…	5…		6…			3…
a_{10}		1…	4…		7…			2…	2…			7…		4…	1…	
		2…	7…		4…			1…	1…			4…		7…	2…	
a_{11}	0…			5…		6…	3…			3…	6…		5…			0…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
9_x	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
10_x	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
		4…	1…		2…			7…	7…			2…		1…	4…	
	6…			3…		0…	5…			5…	0…		3…			6…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
11_x	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…			6…		5…	0…			0…	5…		6…			3…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*

a_8 a_7 a_{11}

8_x	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$
	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$
9_x	$F_1(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
	$F_2(a_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$
10_x	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$
11_x	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ a_6\ a_9\ \underline{a}_{10})$
	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ a_{10})$	$F_2(a_5\ a_6\ a_9\ \underline{a}_{10})$

в)

Рис. 5.22 (продолжение)

a_8

a_7

a'_{11}

a_6

a_5

12_x

a_9

a_{10}

a_{11}

4…			1…		2…	7…			7…	2…		1…			4…
	6…	3…		0…			5…	5…			0…		3…	6…	
	5…	0…		3…			6…	6…			3…		0…	5…	
7…			2…		1…	4…			4…	1…		2…			7…
*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*

13_x

1…			4…		7…	2…			2…	7…		4…			1…
	3…	6…		5…			0…	0…			5…		6…	3…	
	0…	5…		6…			3…	3…			6…		5…	0…	
2…			7…		4…	1…			1…	4…		7…			2…
*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*

14_x

2…			7…		4…	1…			1…	4…		7…			2…
	0…	5…		6…			3…	3…			6…		5…	0…	
	3…	6…		5…			0…	0…			5…		6…	3…	
1…			4…		7…	2…			2…	7…		4…			1…
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*

15_x

7…			2…		1…	4…			4…	1…		2…			7…
	5…	0…		3…			6…	6…			3…		0…	5…	
	6…	3…		0…			5…	5…			0…		3…	6…	
4…			1…		2…	7…			7…	2…		1…			4…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*

a_8

a_7

a_{11}

$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$
$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$

12_x

$F_1(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(\underline{a}_5\ a_6\ a_9\ a_{10})$
$F_2(a_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(\underline{a}_5\ a_6\ a_9\ a_{10})$

13_x

$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(\underline{a}_5\ \underline{a}_6\ a_9\ a_{10})$	$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$

14_x

$F_1(\underline{a}_5\ a_6\ \underline{a}_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$	$F_1(a_5\ a_6\ a_9\ a_{10})$	$F_1(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$
$F_2(a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ a_6\ a_9\ a_{10})$	$F_2(a_5\ a_6\ \underline{a}_9\ \underline{a}_{10})$	$F_2(a_5\ \underline{a}_6\ a_9\ \underline{a}_{10})$

15_x

г)

Рис. 5.22 (окончание)

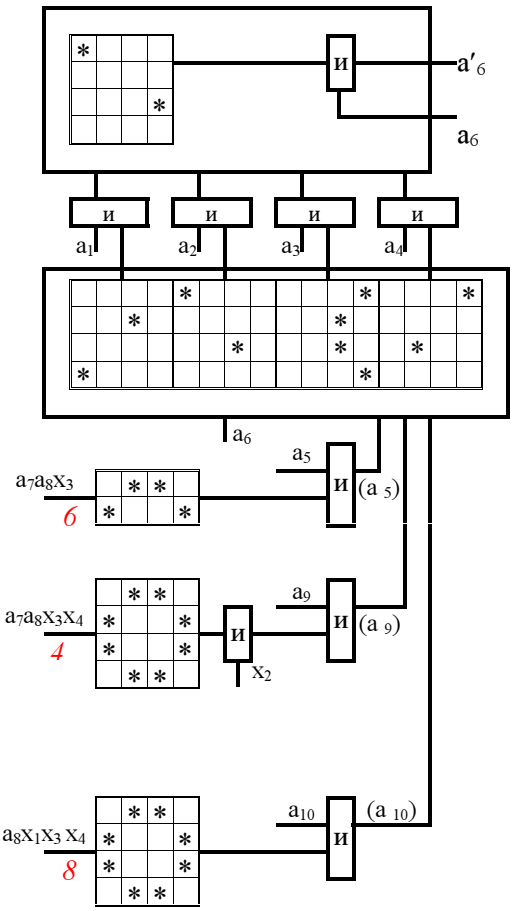


Рис. 5.23

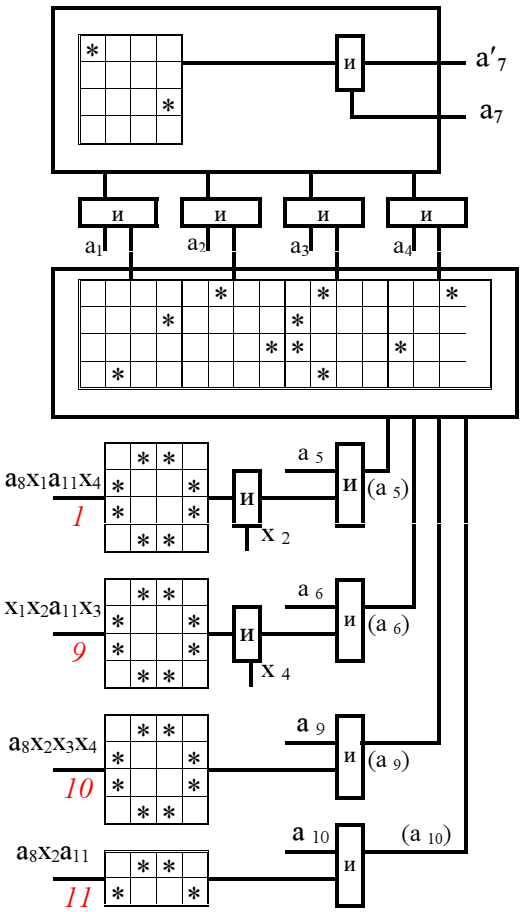


Рис. 5.24

Процедура составления этих структурных схем весьма проста.

Функциональная схема БГО (a_1 – a_4) представляет собой покрытие геометрического образа функции 0…, изображенной на рис. 5.15, *а* каждая функциональная схема БП (a_1 – a_4) состоит из четырех частей, предназначенных соответственно для выполнения операции инвертирования сигналов a_1 – a_4 на входе БГО (a_1 – a_4) отдельно в каждой из схем рис. 5.16, рис. 5.23 – 5.28.

Представим построение этих функциональных схем на примере пятого разряда для сигнала a'_5 рис. 5.16.

Основой для такого построения служит базовый геометрический образ функций в промежуточных координатах a_5 a_6 a_9 a_{10} базового геометрического образа (рис. 5.15, *а*), где во второй ячейке расположена базовая функция 0…, в третьей ячейке – функция 5…, в пятой ячейке – функция 2…, в восьмой ячейке – функция 7… и т.д. Определяем на основании этих данных и их геометрических образов функций 0… – 7… (см. рис. 5.14) необходимые координаты пространства для инвертирования сигналов на входе блока БГО (a_1 – a_4) функциональной схемы (см. рис. 5.16). Эта процедура заполнения ячеек пространства координат a_5 a_6 a_9 a_{10} совершается следующим образом:

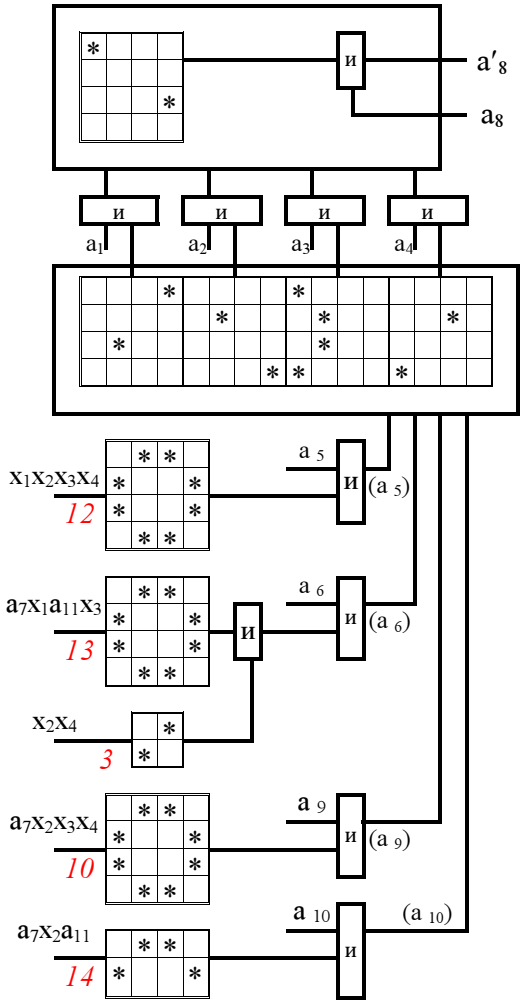


Рис. 5.25

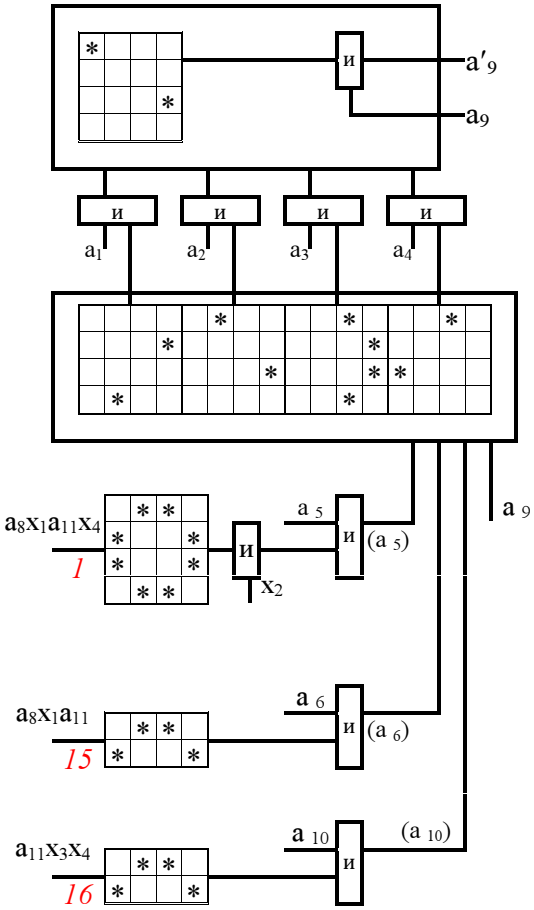


Рис. 5.26

в ячейке, где расположена базовая функция 0…, инвертирование не производится, и эта ячейка остается пустой (логический нуль) для всех сигналов $a_1 – a_4$; в следующей ячейке, где расположена функция 5…, инвертируются сигналы на входах a_1, a_3, и эта ячейка заполняется логической единицей для этих функций сигналов, и т.д.

Так в системе промежуточных координат (a_5) (a_6) (a_9) (a_{10}) определяются геометрические образы функций инвертирования сигналов на входе блока БГО ($a_1 – a_4$).

Аналогично выполняется построение функциональных схем блоков БГО ($a_1 – a_4$) для всех последующих разрядов $a_6 – a_{11}$ устройства исправления одиночных ошибок.

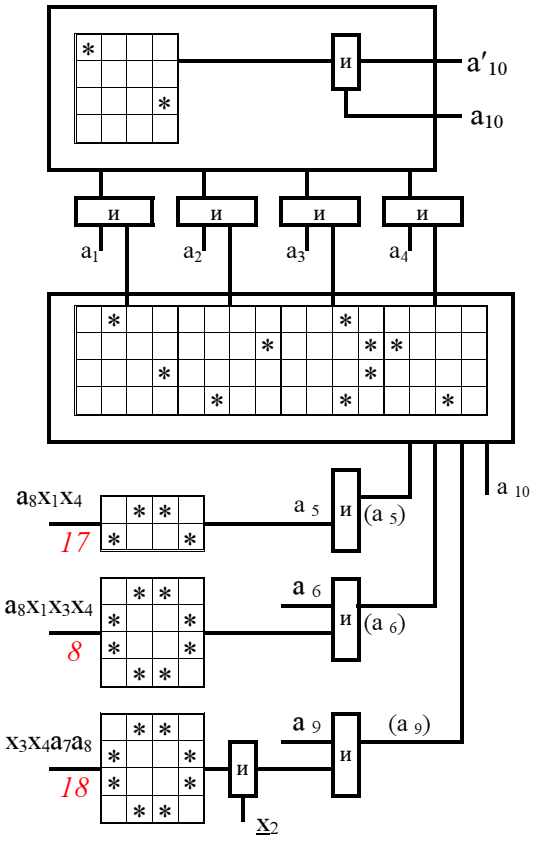


Рис. 5.27

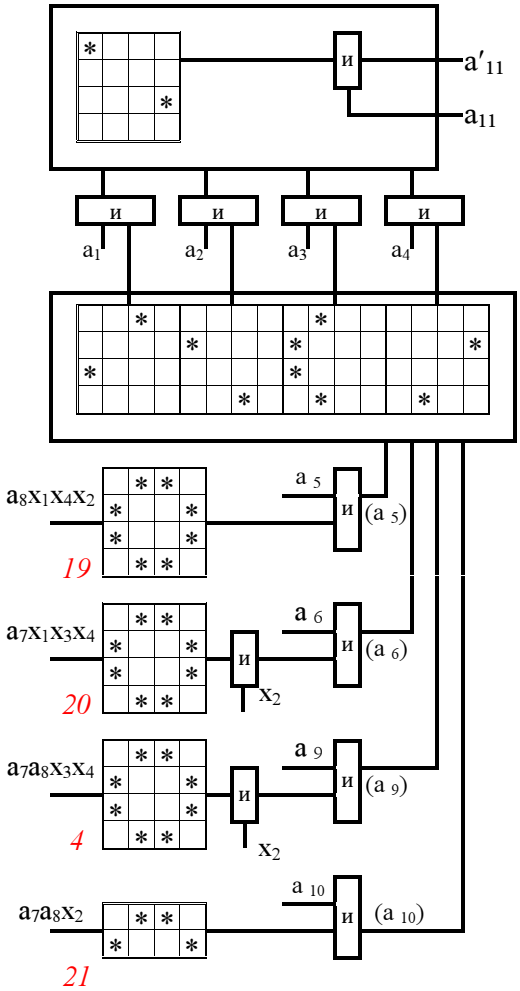


Рис. 5.28

Построение блоков БП (a_5 a_6 a_9 a_{10}) для всех функциональных схем рис. 5.16, 5.23 – 5.28 выполняется таким же образом, как было описано выше для схем информационных разрядов a_1 – a_4.

5.3. Исправление одиночных ошибок в контрольной части кода

Определение геометрических образов исправленных сигналов x'_1 – x'_4 контрольных разрядов начинается с этапа представления универсального цифрового пространства его информационной части в общих координатах a_1 – a_{11}; x_1 – x_4 систематического кода.

Этот этап синтеза условно показан на рис. 5.29, *а*, где шестнадцать цифровых пространств в координатах a_1 – a_{11}, каждое из которых обозначено соответственно 0_x – 15_x, размещают в себе разнесенные по этим пространствам ячейки универсального пространства его информационной части. Например, в пространстве под номером 0_x размещаются ячейки с одноименными номерами, представленными на рис. 5.4, *а, б* и т.д.

0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	10_x	11_x	12_x	13_x	14_x	15_x	*а*)
	$1'_x$		$3'_x$		$5'_x$		$7'_x$		$9'_x$		$11'_x$		$13'_x$		$15'_x$	x_1
		$2'_x$	$3'_x$			$6'_x$	$7'_x$			$10'_x$	$11'_x$			$14'_x$	$15'_x$	x_2
				$4'_x$	$5'_x$	$6'_x$	$7'_x$					$12'_x$	$13'_x$	$14'_x$	$15'_x$	x_3
								$8'_x$	$9'_x$	$10'_x$	$11'_x$	$12'_x$	$13'_x$	$14'_x$	$15'_x$	x_4

б)

	$1'_x$		$3'_x$		$5'_x$		$7'_x$		$9'_x$		$11'_x$		$13'_x$		$15'_x$	x_1
		$2'_x$	$3'_x$			$6'_x$	$7'_x$			$10'_x$	$11'_x$			$14'_x$	$15'_x$	x_2
				$4'_x$	$5'_x$	$6'_x$	$7'_x$					$12'_x$	$13'_x$	$14'_x$	$15'_x$	x_3
								$8'_x$	$9'_x$	$10'_x$	$11'_x$	$12'_x$	$13'_x$	$14'_x$	$15'_x$	x_4

в)

1_x	$\underline{0}_x$	3_x	$\underline{2}_x$	5_x	$\underline{4}_x$	7_x	$\underline{6}_x$	9_x	$\underline{8}_x$	11_x	$\underline{10}_x$	13_x	$\underline{12}_x$	15_x	$\underline{14}_x$	x'_1
2_x	3_x	$\underline{0}_x$	$\underline{1}_x$	6_x	7_x	$\underline{4}_x$	$\underline{5}_x$	10_x	11_x	$\underline{8}_x$	$\underline{9}_x$	14_x	15_x	$\underline{12}_x$	$\underline{13}_x$	x'_2
4_1	5_x	6_x	7_x	$\underline{0}_x$	$\underline{1}_x$	$\underline{2}_x$	$\underline{3}_x$	12_x	13_x	14_x	15_x	$\underline{8}_x$	$\underline{9}_x$	$\underline{10}_x$	$\underline{11}_x$	x'_3
8_x	9_x	10_x	11_x	12_x	13_x	14_x	15_x	$\underline{0}_x$	$\underline{1}_x$	$\underline{2}_x$	$\underline{3}_x$	$\underline{4}_x$	$\underline{5}_x$	$\underline{6}_x$	$\underline{7}_x$	x'_4

г)

Рис. 5.29

На втором этапе (см. рис. 5.29, *б*) производится выделение и объединение из этих пространств таких, которые составляют только штатные ячейки разрядов контрольной части кода по известным выражениям:

$x_1 = 1_x \vee 3_x \vee \ \ldots \vee 15_x$; $x_2 = (2_x \vee 3_x) \vee (6_x \vee 7_x) \vee (10_x \vee 11_x) \vee (14_x \vee 15_x)$;

$x_3 = (4_x \vee \ldots \vee 7_x) \vee (12_x \vee \ldots \vee 15_x)$; $x_4 = (8_x \vee \ldots \vee 15_x)$.

На следующем этапе осуществляется определение ячеек в общих координатах $a_1 - a_{11}$; $x_1 - x_4$ пространства с одиночными ошибками. При этом выделение таких ячеек с одиночными ошибками производится с использованием табл. 3.1.1. Этот этап разбивается на два условных подэтапа. На первом из них определяются ячейки пространства в координатах $a_1 - a_{11}$, которые полностью составляют контрольные разряды $x_1 - x_4$ с одиночными ошибками только в информационных разрядах. Такие пространства обозначены на рис. 5.29, *в* под номерами $0'_x - 15'_x$. На втором подэтапе определяются ячейки с одиночными ошибками в контрольных разрядах систематического кода общего пространства координат $a_1 - a_{11}$; $x_1 - x_4$, которые совместно с одиночными ошибками информационной части образуют четыре пространства (см. рис. 5.29, *г*). В этих пространствах располагаются соответственно геометрические образы сигналов $x'_1 - x'_4$. Здесь под номерами $0_x - 15_x$, $\underline{0}_x - \underline{15}_x$ условно представлены геометрические образы, а также инверсии геометрических образов рис. 5.29, *а*.

Результатом такого поэтапного синтеза являются геометрические образы (рис. 5.30, *а – г* – 5.33, *а – г*) исправленных сигналов контрольной части систематического кода в объединенном цифровом пространстве его координат. Как и для информационной части кода, на каждом из этих рисунков в пространстве координат соответственно под номерами $0_x - 3_x$; $4_x - 7_x$; $8_x - 11_x$; $12_x - 15_x$ приведены части соответствующего геометрического образа исправленной функции.

x'_1								a_8								
				a_7												
		a_6														
	a_5															
0_x																
a_9	5…			0…		3…	6…			6…	3…		0…			5…
a_{10}		7…	2…		1…			4…	4…			1…		2…	7…	
		4…	1…		2…			7…	7…			2…		1…	4…	
a_{11}	6…			3…		0…	5…			5…	0…		3…			6…
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…			6…		5…	0…			0…	5…		6…			3…
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
1_x	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5..	*	*	5..	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
2_x	3…			6…		5…	0…			0…	5…		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
		2…	7…		4…			1..	1..			4…		7…	2…	
	0…			5…		6…	3…			3…	6…		5…			0…
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
3_x	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*

	a_8			
	a_7			
a_{11}				
0_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
$\underline{1}_x$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
2_x	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$
$\underline{3}_x$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$

а)

Рис. 5.30 (начало)

a_8

a_7

x'_1

a_6

a_5

4_x

a_9

a_{10}

a_{11}

2…			7…		4…	1…			1…	4…		7…			2…
	0…	5…		6…			3…	3…			6…		5…	0…	
	3…	6…		5…			0…	0…			5…		6…	3…	
1…			4…		7…	2…			2…	7…		4…			1…
	6…	3…		0…			5…	5…			0…		3…	6…	
4…			1…		2…	7…			7…	2…		1…			4…
7…			2…		1…	4…			4…	1…		2…			7…
	5…	0…		3…			6…	6…			3…		0…	5…	

5_x

7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*

6_x

4…			1…		2…	7…			7…	2…		1…			4…
	6…	3…		0…			5…	5…			0…		3…	6…	
	5…	0…		3…			6…	6…			3…		0…	5…	
7…			2…		1…	4…			4…	1…		2…			7…
	0…	5…		6…			3…	3…			6…		5…	0…	
2…			7…		4…	1…			1…	4…		7…			2…
1…			4…		7…	2…			2…	7…		4…			1…
	3…	6…		5…			0…	0…			5…		6…	3…	

7_x

1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*

a_8

a_7

a_{11}

4_x	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
$\underline{5_x}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
6_x	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
$\underline{7_x}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$

б)

Рис. 5.30 (продолжение)

a_8

a_7

x'_1 a_6

a_5

8_x a_9		2…	7…		4…			1…	1…			4…		7…	2…	
a_{10}	0…			5…		6…	3…			3…	6…		5…			0…
	3…			6…		5…	0…			0…	5…		6…			3…
a_{11}		1…	4…		7…			2…	2…			7…		4…	1…	
	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
9_x	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
10_x		4…	1…		2…			7…	7…			2…		1…	4…	
	6…			3…		0…	5…			5…	0…		3…			6…
	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…			6…		5…	0…			0…	5…		6…			3…
11_x	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…

a_8

a_7

8_x a_{11}	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$
	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
$\underline{9}_x$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$
10_x	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
$\underline{11}_x$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$

в)

Рис. 5.30 (продолжение)

a_8

a_7

x'_1 a_6

a_5

12_x		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
13_x	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
14_x		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…
15_x	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…

a_8

a_7

12_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
13_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
14_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
15_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$

г)

Рис. 5.30 (окончание)

a_8

a_7

x'_2 a_6

a_5

0_x	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…			5…		6…	3…			3…	6…		5…			0…
	3…			6…		5…	0…			0…	5…		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
1_x	3…			6…		5…	0…			0…	5…		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…			5…		6…	3…			3…	6…		5…			0…
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
	6…			3…		0…	5…			5…	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
2_x	0..	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0..
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
3_x	5…	*	*	0..	*	3…	6…	*	*	6…	3…	*	0..	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*

a_8

a_7

0_x	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
1_x	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
2_x	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
3_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$

а)

Рис. 5.31 (начало)

a_8

a_7

x'_2 a_6

a_5

4_x a_9	1…			4…		7…	2…			2…	7…		4…			1…
a_{10}		3…	6…		5…			0…	0…			5…		6…	3…	
		0…	5…		6…			3…	3…			6…		5…	0…	
a_{11}	2…			7…		4…	1…			1…	4…		7…			2…
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
5_x	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
6_x	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
7_x	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*

a_8

a_7

4_x a_{11}	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$
	$\underline{a}_5\ a_6\ a_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$
5_x	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ a_{10}$	$a_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$
	$a_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ a_9\ \underline{a}_{10}$
$\underline{6}_x$	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$a_5\ a_6\ a_9\ \underline{a}_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$
	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ \underline{a}_{10}$
$\underline{7}_x$	$a_5\ \underline{a}_6\ \underline{a}_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ \underline{a}_{10}$	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$
	$\underline{a}_5\ a_6\ \underline{a}_9\ a_{10}$	$\underline{a}_5\ \underline{a}_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a}_6\ \underline{a}_9\ \underline{a}_{10}$

б)

Рис. 5.31 (продолжение)

a_8

a_7

x'_2 a_6

a_5

	1…	4…		7…			2…	2…			7…		4…	1…	
3…			6…		5…	0…			0…	5…		6…			3…
0…			5…		6…	3…			3…	6…		5…			0…
	2…	7…		4…			1…	1…			4…		7…	2…	
5…			0…		3…	6…			6…	3…		0…			5…
	7…	2…		1…			4…	4…			1…		2…	7…	
	4…	1…		2…			7…	7…			2…		1…	4…	
6…			3…		0…	5…			5…	0…		3…			6…

8_x a_9 a_{10} a_{11}

	4…	1…		2…			7…	7…			2…		1…	4…	
6…			3…		0…	5…			5…	0…		3…			6…
5…			0…		3…	6…			6…	3…		0…			5…
	7…	2…		1…			4…	4…			1…		2…	7…	
0…			5…		6…	3…			3…	6…		5…			0…
	2…	7…		4…			1…	1…			4…		7…	2…	
	1…	4…		7…			2…	2…			7…		4…	1…	
3…			6…		5…	0…			0…	5…		6…			3…

9_x

*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…

10_x

*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…

11_x

a_8

a_7

a_{11}	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
8_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
9_x	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
$\underline{10_x}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$
$\underline{11_x}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$

в)

Рис. 5.31 (продолжение)

a_8

a_7

x'_2

a_6

a_5

12_x

a_9

a_{10}

a_{11}

	6…	3…		0…			5…	5…			0…		3…	6…	
4…			1…		2…	7…			7…	2…		1…			4…
7…			2…		1…	4…			4…	1…		2…			7…
	5…	0…		3…			6…	6…			3…		0…	5…	
2…			7…		4…	1…			1…	4…		7…			2…
	0…	5…		6…			3…	3…			6…		5…	0…	
	3…	6…		5…			0…	0…			5…		6…	3…	
1…			4…		7…	2…			2…	7…		4…			1…

13_x

	3…	6…		5…			0…	0…			5…		6…	3…	
1…			4…		7…	2…			2…	7…		4…			1…
2…			7…		4…	1…			1…	4…		7…			2…
	0…	5…		6…			3…	3…			6…		5…	0…	
7…			2…		1…	4…			4…	1…		2…			7…
	5…	0…		3…			6…	6…			3…		0…	5…	
	6…	3…		0…			5…	5…			0…		3…	6…	
4…			1…		2…	7…			7…	2…		1…			4…

14_x

*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…

15_x

*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…

a_8

a_7

a_{11}

12_x	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
13_x	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
$\underline{14}_x$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$
	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
$\underline{15}_x$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$
	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$

г)

Рис. 5.31 (окончание)

a_8

a_7

x'_3 a_6

a_5

0_x a_9	7…			2…		1…	4…			4…	1…		2…			7…
a_{10}		5…	0…		3…			6…	6…			3…		0…	5…	
		6…	3…		0…			5…	5…			0…		3…	6…	
a_{11}	4…			1…		2…	7…			7…	2…		1…			4…
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
1_x	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
2_x	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
3_x	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	

a_8

a_7

0_x a_{11}	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$
1_x	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
2_x	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
3_x	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$

а)

Рис. 5.32 (начало)

a_8

a_7

x'_3

a_6

a_5

4_x	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
5_x	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
6_x	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
7_x	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*

a_8

a_7

a_{11}

4_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
5_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
6_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
7_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$

б)

Рис. 5.32 (продолжение)

a_8

a_7

x'_3 a_6

a_5

8_x		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
9_x		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
10_x		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
11_x		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…

a_8

a_7

a_{11}

8_x	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$
	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
9_x	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$
	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
10_x	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
11_x	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$

в)

Рис. 5.32 (продолжение)

a_8

a_7

x'_3 a_6

a_5

12_x	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
13_x	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
14_x	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
15_x	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…

a_8

a_7

12_x	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$
13_x	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
14_x	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
15_x	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$

г)

Рис. 5.32 (окончание)

a_8

a_7

x'_4 a_6

a_5

0_x		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
	6…			3…		0…	5..			5..	0…		3…			6…
		4…	1…		2…			7…	7…			2…		1…	4…	
	3…			6..		5…	0…			0…	5…		6..			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
		2…	7…		4…			1…	1…			4…		7…	2…	
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
1_x	0…			5…		6…	3…			3…	6…		5…			0…
	3…			6…		5..	0…			0…	5..		6…			3…
		1…	4…		7…			2…	2…			7…		4…	1…	
	6..			3…		0…	5…			5…	0…		3…			6..
		4…	1…		2…			7…	7…			2…		1…	4…	
		7…	2…		1…			4…	4…			1…		2…	7…	
	5…			0…		3…	6…			6…	3…		0…			5…
		1…	4…		7…			2…	2…			7…		4…	1…	
2_x	3…			6…		5..	0…			0…	5..		6…			3…
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
		4…	1…		2…			7…	7…			2…		1…	4…	
	6..			3…		0…	5…			5…	0…		3…			6..
		4…	1…		2…			7…	7…			2…		1…	4…	
3_x	6…			3…		0…	5..			5..	0…		3…			6…
	5…			0…		3…	6…			6…	3…		0…			5…
		7…	2…		1…			4…	4…			1…		2…	7…	
	0…			5…		6…	3…			3…	6…		5…			0…
		2…	7…		4…			1…	1…			4…		7…	2…	
		1…	4…		7…			2…	2…			7…		4…	1…	
	3…			6..		5…	0…			0…	5…		6..			3…

a_8

a_7

0_x	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
1_x	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
2_x	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
3_x	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$

а)

Рис. 5.33 (начало)

a_8

a_7

x'_4

a_6

a_5

4_x		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
		5…	0…		3…			6…	6…			3…		0…	5…	
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
5_x	7…			2…		1…	4…			4…	1…		2…			7…
	4…			1…		2…	7…			7…	2…		1…			4…
		6…	3…		0…			5…	5…			0…		3…	6…	
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
		0…	5…		6…			3…	3…			6…		5…	0…	
	2…			7…		4…	1…			1…	4…		7…			2…
		6…	3…		0…			5…	5…			0…		3…	6…	
6_x	4…			1…		2…	7…			7…	2…		1…			4…
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
		3…	6…		5…			0…	0…			5…		6…	3…	
	1…			4…		7…	2…			2…	7…		4…			1…
		3…	6…		5…			0…	0…			5…		6…	3…	
7_x	1…			4…		7…	2…			2…	7…		4…			1…
	2…			7…		4…	1…			1…	4…		7…			2…
		0…	5…		6…			3…	3…			6…		5…	0…	
	7…			2…		1…	4…			4…	1…		2…			7…
		5…	0…		3…			6…	6…			3…		0…	5…	
		6…	3…		0…			5…	5…			0…		3…	6…	
	4…			1…		2…	7…			7…	2…		1…			4…

a_8

a_7

a_{11}

4_x	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$
	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$
5_x	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$
	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$
6_x	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$
	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
7_x	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$
	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$

б)

Рис. 5.33 (продолжение)

x'_4 a_5 a_6 a_7 a_8

8_x a_9	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
a_{10}	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
a_{11}	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
9_x	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
10_x	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
11_x	3…	*	*	6…	*	5…	0…	*	*	0…	5…	*	6…	*	*	3…
	*	1…	4…	*	7…	*	*	2…	2…	*	*	7…	*	4…	1…	*
	*	2…	7…	*	4…	*	*	1…	1…	*	*	4…	*	7…	2…	*
	0…	*	*	5…	*	6…	3…	*	*	3…	6…	*	5…	*	*	0…
	*	7…	2…	*	1…	*	*	4…	4…	*	*	1…	*	2…	7…	*
	5…	*	*	0…	*	3…	6…	*	*	6…	3…	*	0…	*	*	5…
	6…	*	*	3…	*	0…	5…	*	*	5…	0…	*	3…	*	*	6…
	*	4…	1…	*	2…	*	*	7…	7…	*	*	2…	*	1…	4…	*

a_7 a_8

8_x a_{11}	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$
$\underline{9_x}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ a_9\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ \underline{a_9}\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$
$\underline{10_x}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$
	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ \underline{a_{10}}$	$a_5\ a_6\ a_9\ a_{10}$
$\underline{11_x}$	$\underline{a_5}\ a_6\ \underline{a_9}\ a_{10}$	$\underline{a_5}\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ \underline{a_6}\ \underline{a_9}\ a_{10}$
	$a_5\ a_6\ \underline{a_9}\ a_{10}$	$a_5\ \underline{a_6}\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ a_6\ a_9\ \underline{a_{10}}$	$\underline{a_5}\ \underline{a_6}\ \underline{a_9}\ a_{10}$

в)

Рис. 5.33 (продолжение)

a_8

a_7

x'_4 a_6

a_5

12_x	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
13_x	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
14_x	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
15_x	4…	*	*	1…	*	2…	7…	*	*	7…	2…	*	1…	*	*	4…
	*	6…	3…	*	0…	*	*	5…	5…	*	*	0…	*	3…	6…	*
	*	5…	0…	*	3…	*	*	6…	6…	*	*	3…	*	0…	5…	*
	7…	*	*	2…	*	1…	4…	*	*	4…	1…	*	2…	*	*	7…
	*	0…	5…	*	6…	*	*	3…	3…	*	*	6…	*	5…	0…	*
	2…	*	*	7…	*	4…	1…	*	*	1…	4…	*	7…	*	*	2…
	1…	*	*	4…	*	7…	2…	*	*	2…	7…	*	4…	*	*	1…
	*	3…	6…	*	5…	*	*	0…	0…	*	*	5…	*	6…	3…	*

a_8

a_7

12_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
13_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
14_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
15_x	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$
	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$	$a_5\ a_6\ a_9\ a_{10}$

г)

Рис. 5.33 (окончание)

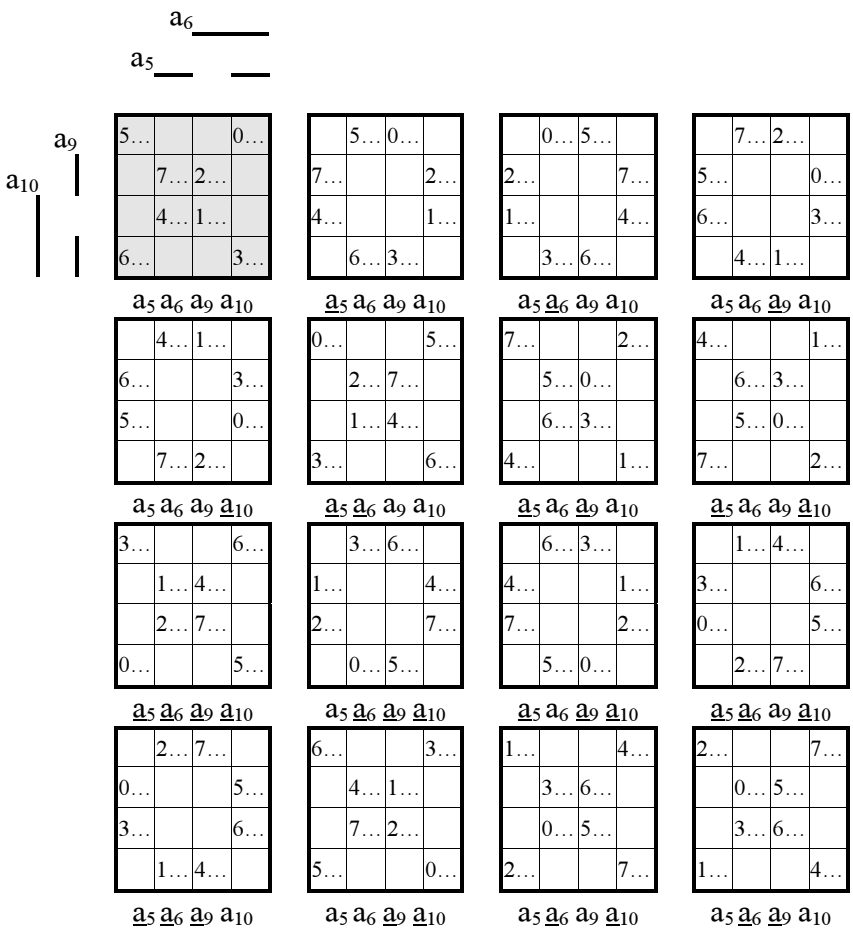


Рис. 5.34

На рис. 5.30 в пространстве под номером 0_x выделена цветом 1/16 часть этого пространства, которая выбрана нами базовой фигурой для покрытия всех геометрических образов рис. 5.30, *а – г* – 5.33, *а – г* .

Следовательно, мысленными поворотами относительно осей симметрии пространства с координатами a_5 a_6 a_9 a_{10} можно осуществить покрытие всех геометрических образов и тем самым определить логические функциональные зависимости исправления одиночных ошибок в контрольной части кода.

Под каждым из этих геометрических образов приведены в соответствующих ячейках пространств на рис. 5.30, *а – г* – 5.33, *а – г* эти мысленные повороты, а соответствующие этим поворотам преобразования базового геометрического образа представлены на рис. 5. 34.

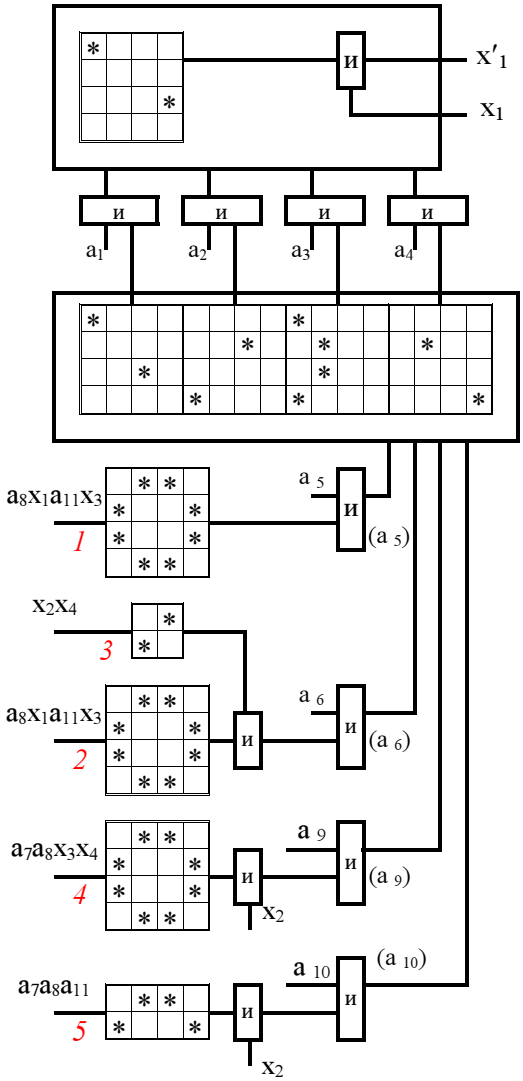


Рис. 5.35

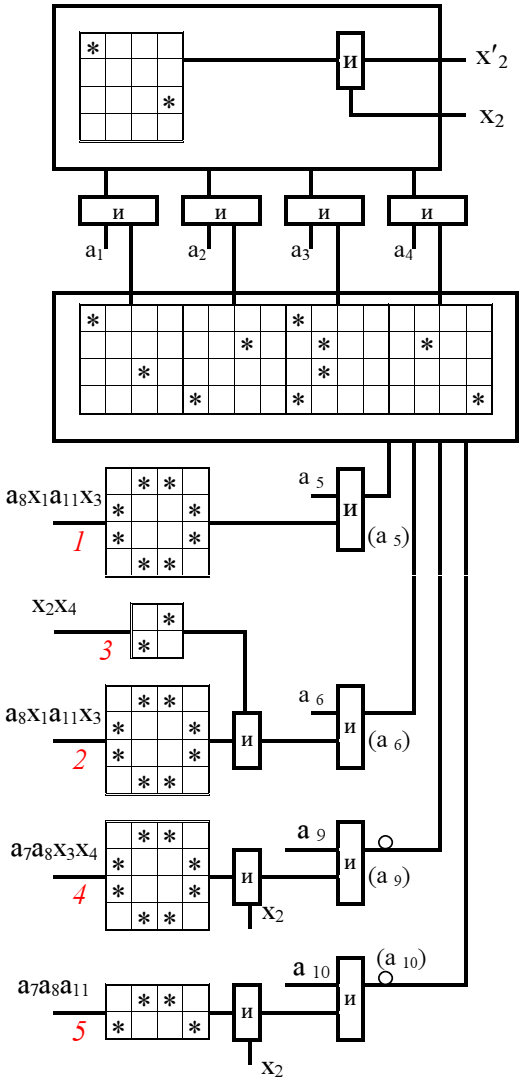


Рис. 5.36

Выберем в качестве базового геометрического образа в координатах пространства a_1 – a_4 такой же образ, какой был принят нами для информационных разрядов a_5 – a_{11}. Этот образ соответствует функции 0… (см. рис. 5.14). Тогда, аналогично описанной выше процедуре построения функциональных схем исправления ошибок информационной части систематического кода, могут быть синтезированы функциональные схемы его контрольной части.

Эти функциональные схемы для разрядов x_1 – x_4 приведены соответственно на рис. 5.35 – 5.38. Все схемы этих рисунков содержат одинаковые блоки базового геометрического образа БГО (a_1 – a_4), которые управляются последовательно соединенными блоками поворота координат БП ($a_7a_8a_{11}$; x_1 – x_4) и БП (a_1 – a_4).

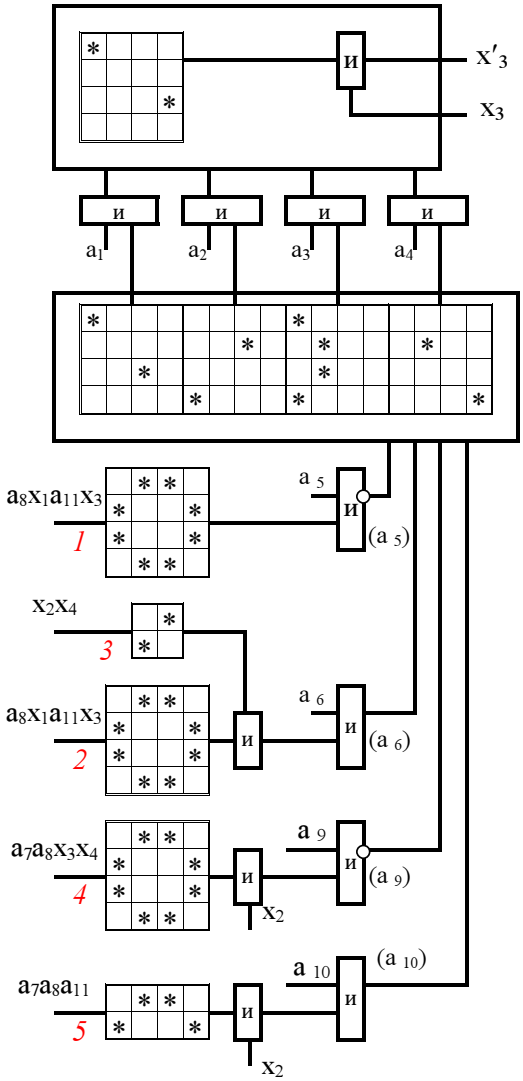


Рис. 5.37

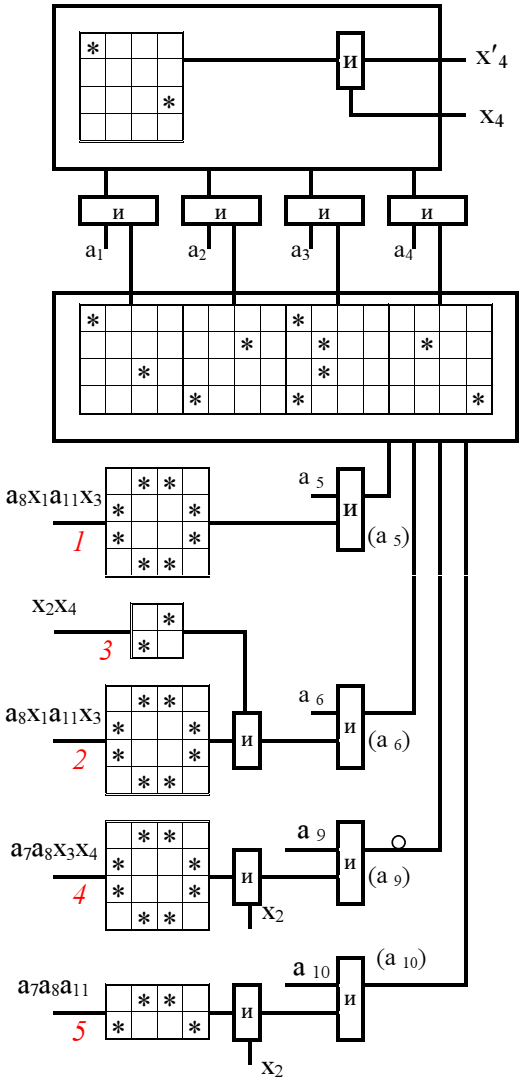


Рис. 5.38

Функциональные схемы БП (a_1 – a_4) в каждом из контрольных разрядов x_1 – x_4 одинаковые, а схемы БП (a_5 a_6 a_9 a_{10}) отличаются незначительно только инверсиями выходных сигналов (a_5), (a_6), (a_9), (a_{10}) и содержат одинаковые функциональные схемы, нумерация которых совпадает с аналогичным блоком информационной части кода сигналов a_1 – a_4.

Общее число функциональных схем в устройстве, исправляющем одиночные ошибки всего систематического кода, равно *21*.

Функционально все схемы этих блоков однотипны и отличаются одна от другой только числом входов и порядком подаваемых на эти входы сигналов разрядов систематического кода a_7, a_8, a_{11}; x_1 – x_4.

Представляет определенный практический интерес рассмотреть изменение функциональных схем исправления одиночных ошибок во всех разрядах систематического кода, когда число его информационных разрядов уменьшается от 11 до 5. Синтезируемые таким образом новые систематические коды, когда число контрольных разрядов остается неизменным, сохраняют возможность исправления всех одиночных ошибок в информационной и контрольной его части, но использование цифрового пространства координат при этом уменьшается, т.е. в этом цифровом пространстве будут появляться ячейки, не заполненные одиночными ошибками.

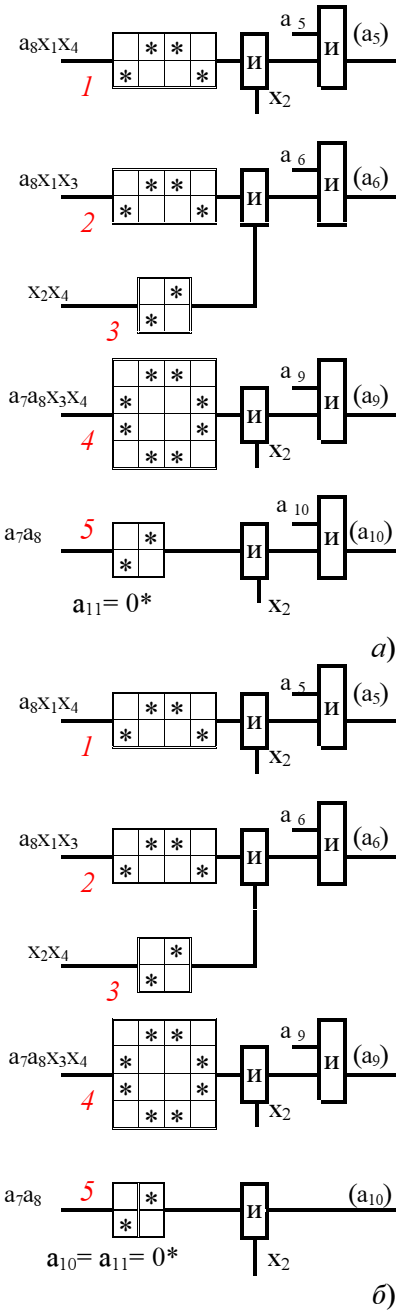


Рис. 5.39 (начало)

Это изменение функциональных схем касается только одного блока поворота координат БП (a_5 a_6 a_9 a_{10}), а все остальные блоки остаются неизменными (см. рис. 5.12, 5.16 – 5.28, 5.35 – 5.38). Если цифровое пространство, где все ячейки заполняются штатными цифрами и их одиночными ошибками, является совершенным, то новое видоизмененное контролируемое пространство будет квазисовершенным.

В качестве примера синтеза таких квазисовершенных кодов рассмотрим последовательное уменьшение информационных разрядов первичного совершенного кода и проследим изменение блоков БП (a_5 a_6 a_9 a_{10}) только для первых четырех его информационных разрядов (см. рис. 5.12), для которых этот блок общий. При этом, исходя из наглядности преобразований, не будем стремиться уменьшать уровень схем этого блока, т.е. стремиться повысить их быстродействие, хотя при таких упрощениях это целесообразно выполнять.

Изменение этих блоков в других информационных и контрольных разрядах кода при таком геометрическом синтезе не представляет какой-либо сложности и может быть выполнено читателем самостоятельно.

Все последовательные преобразования блока БП (a_5 a_6 a_9 a_{10}) при этих синтезах приведены на рис. 5.39, *а* – *е*, что сопровождается соответствующим изменением функциональных схем *1* – *5*.

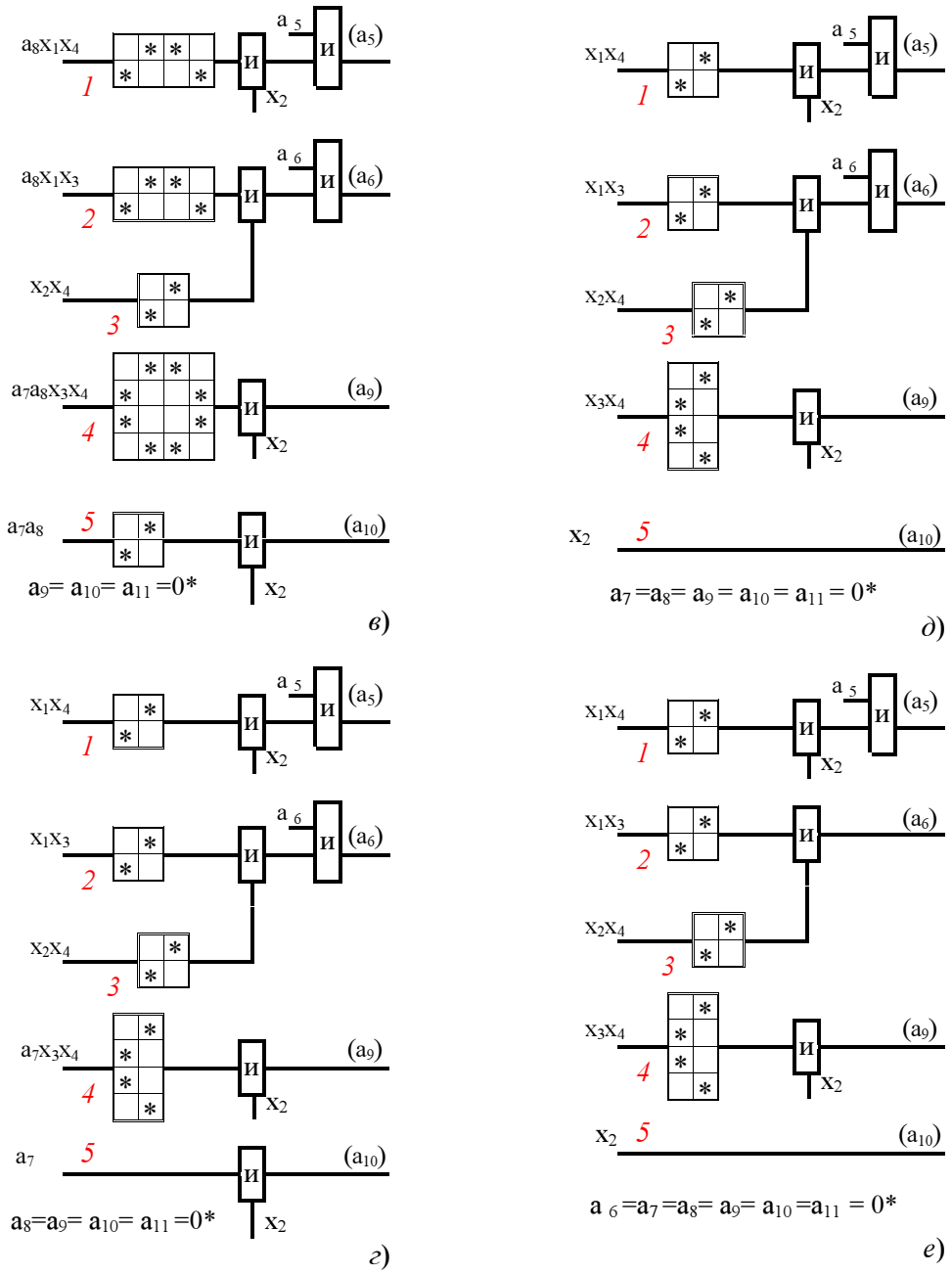


Рис. 5.39 (продолжение)

В заключение синтеза необходимо отметить следующее: все функциональные схемы блоков БП (a_5 a_6 a_9 a_{10}), а также схемы шифраторов обладают определенным видом симметрии. Это свидетельствует о том, что при выполнении мысленных поворотов относительно осей симметрии пространства,

совпадающих с осями симметрии геометрических образов, последние остаются неизменными. Следовательно, существует некоторое определенное число эквивалентных схем, которые отличаются одна от другой только ансамблями их входных сигналов.

Во всех этих схемах используются три геометрических образа, на которых необходимо остановиться.

Это прежде всего схема инвертора m_1m_2 f, которая общеизвестна (исключающее ИЛИ) и не требует дополнительных пояснений.

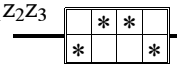
Вторым является образ трехвходовой функции $z_1z_2z_3$  f , где число таких ансамблей входа определяется табл. 5.3.1.

Таблица 5.3.1

$z_1z_2z_3$	$z_1z_3z_2$	$\underline{z_1}\underline{z_2}z_3$	$\underline{z_1}\underline{z_3}z_2$
$z_2z_1z_3$	$z_2z_3z_1$	$\underline{z_2}\underline{z_1}z_3$	$\underline{z_2}\underline{z_3}z_1$
$z_3z_2z_1$	$z_3z_1z_2$	$\underline{z_3}\underline{z_2}z_1$	$\underline{z_3}\underline{z_1}z_2$

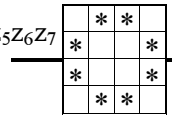
Третьим будет образ четырехвходовой функции $z_4z_5z_6z_7$  f , где табл. 5.3.2 определяет для нее аналогичные значения входных сигналов.

Таблица 5.3.2

$z_4z_5z_6z_7$	$z_4z_6z_5z_7$	$z_4z_7z_6z_5$	$z_4z_5z_7z_6$	$z_4z_6z_7z_5$	$z_4z_7z_5z_6$
$z_5\ z_4z_6z_7$	$z_5z_6z_4z_7$	$z_5z_7z_6z_4$	$z_5z_4z_7z_6$	$z_5z_6z_7z_4$	$z_5z_7z_4z_6$
$z_6z_5z_4z_7$	$z_6z_4z_5z_7$	$z_6z_7z_4z_5$	$z_5z_6z_7z_4$	$z_6z_4z_7z_5$	$z_6z_7z_5z_4$
$z_7z_6z_5z_4$	$z_7z_5z_6z_4$	$z_7z_4z_5z_6$	$z_7z_6z_4z_5$	$z_7z_5z_4z_6$	$z_7z_4z_6z_5$
$\underline{z_4}z_5z_6z_7$	$\underline{z_4}\underline{z_6}z_5z_7$	$\underline{z_4}\underline{z_7}z_6z_5$	$\underline{z_4}\underline{z_5}z_7z_6$	$\underline{z_4}\underline{z_6}z_7z_5$	$\underline{z_4}\underline{z_7}z_5z_6$
$\underline{z_5}\underline{z_4}z_6z_7$	$\underline{z_5}\underline{z_6}z_4z_7$	$\underline{z_5}\underline{z_7}z_6z_4$	$\underline{z_5}\underline{z_4}z_7z_6$	$\underline{z_5}\underline{z_6}z_7z_4$	$\underline{z_5}\underline{z_7}z_4z_6$
$\underline{z_6}\underline{z_5}z_4z_7$	$\underline{z_6}\underline{z_4}z_5z_7$	$\underline{z_6}\underline{z_7}z_4z_5$	$\underline{z_5}\underline{z_6}z_7z_4$	$\underline{z_6}\underline{z_4}z_7z_5$	$\underline{z_6}\underline{z_7}z_5z_4$
$\underline{z_7}\underline{z_6}z_5z_4$	$\underline{z_7}\underline{z_5}z_6z_4$	$\underline{z_7}\underline{z_4}z_5z_6$	$\underline{z_7}\underline{z_6}z_4z_5$	$\underline{z_7}\underline{z_5}z_4z_6$	$\underline{z_7}\underline{z_4}z_6z_5$
$z_4z_5\underline{z_6}\underline{z_7}$	$z_4z_6\underline{z_5}\underline{z_7}$	$z_4z_7\underline{z_6}\underline{z_5}$	$z_4z_5\underline{z_7}\underline{z_6}$	$z_4z_6\underline{z_7}\underline{z_5}$	$z_4z_7\underline{z_5}\underline{z_6}$
$z_5z_4\underline{z_6}\underline{z_7}$	$z_5z_6\underline{z_4}\underline{z_7}$	$z_5z_7\underline{z_6}\underline{z_4}$	$z_5z_4\underline{z_7}\underline{z_6}$	$z_5z_6\underline{z_7}\underline{z_4}$	$z_5z_7\underline{z_4}\underline{z_6}$
$z_6z_5\underline{z_4}\underline{z_7}$	$z_6z_4\underline{z_5}\underline{z_7}$	$z_6z_7\underline{z_4}\underline{z_5}$	$z_5z_6\underline{z_7}\underline{z_4}$	$z_6z_4\underline{z_7}\underline{z_5}$	$z_6z_7\underline{z_5}\underline{z_4}$
$z_7z_6\underline{z_5}\underline{z_4}$	$z_7z_5\underline{z_6}\underline{z_4}$	$z_7z_4\underline{z_5}\underline{z_6}$	$z_7z_6\underline{z_4}\underline{z_5}$	$z_7z_5\underline{z_4}\underline{z_6}$	$z_7z_4\underline{z_6}\underline{z_5}$
$\underline{z_4}\underline{z_5}\underline{z_6}\underline{z_7}$	$\underline{z_4}\underline{z_6}\underline{z_5}\underline{z_7}$	$\underline{z_4}\underline{z_7}\underline{z_6}\underline{z_5}$	$\underline{z_4}\underline{z_5}\underline{z_7}\underline{z_6}$	$\underline{z_4}\underline{z_6}\underline{z_7}\underline{z_5}$	$\underline{z_4}\underline{z_7}\underline{z_5}\underline{z_6}$
$\underline{z_5}\ \underline{z_4}\underline{z_6}\underline{z_7}$	$\underline{z_5}\underline{z_6}\underline{z_4}\underline{z_7}$	$\underline{z_5}\underline{z_7}\underline{z_6}\underline{z_4}$	$\underline{z_5}\underline{z_4}\underline{z_7}\underline{z_6}$	$\underline{z_5}\underline{z_6}\underline{z_7}\underline{z_4}$	$\underline{z_5}\underline{z_7}\underline{z_4}\underline{z_6}$
$\underline{z_6}\underline{z_5}\underline{z_4}\underline{z_7}$	$\underline{z_6}\underline{z_4}\underline{z_5}\underline{z_7}$	$\underline{z_6}\underline{z_7}\underline{z_4}\underline{z_5}$	$\underline{z_5}\underline{z_6}\underline{z_7}\underline{z_4}$	$\underline{z_6}\underline{z_4}\underline{z_7}\underline{z_5}$	$\underline{z_6}\underline{z_7}\underline{z_5}\underline{z_4}$
$\underline{z_7}\underline{z_6}\underline{z_5}\underline{z_4}$	$\underline{z_7}\underline{z_5}\underline{z_6}\underline{z_4}$	$\underline{z_7}\underline{z_4}\underline{z_5}\underline{z_6}$	$\underline{z_7}\underline{z_6}\underline{z_4}\underline{z_5}$	$\underline{z_7}\underline{z_5}\underline{z_4}\underline{z_6}$	$\underline{z_7}\underline{z_4}\underline{z_6}\underline{z_5}$

Здесь нумерация сигналов на этих двух функциональных схемах принята условно сквозной $z_1 - z_7$.

В каждой функциональной схеме блоков БП ($a_5\ a_6\ a_9\ a_{10}$) на входе могут быть поданы ансамбли сигналов в соответствии с этими таблицами, что определяет огромное количество эквивалентных схем исправления ошибок систематического кода.

5.4. Структурная схема исправления ошибок во всех разрядах систематического кода

Простота и наглядность представленного выше геометрического синтеза очевидны и не требуют дополнительных пояснений.

Однако следует провести некоторый критический разбор представленного варианта синтеза, где определенным достижением является то, что одна и та же схема блока БП ($a_5\ a_6\ a_9\ a_{10}$) используется в устройствах исправления ошибок в информационных ($a_1 - a_4$) и контрольных ($x_1 - x_4$) разрядах, где для каждого из разрядов меняются только сочетание прямых и инверсных сигналов на выходе этого блока. Причем в контрольных разрядах для реализации покрытия функций исправления одиночных ошибок применяется один общий базовый геометрический образ функции в координатах пространства $a_5\ a_6\ a_9\ a_{10}$.

Отличие же устройств исправления ошибок в информационных разрядах $a_5 - a_{11}$ заключается в том, что в каждом из них использовался свой базовый геометрический образ в координатах пространства $a_5\ a_6\ a_9\ a_{10}$, который располагался в начале координат общего цифрового пространства систематического кода. Это обстоятельство привело к значительному усложнению всей схемы устройства исправления ошибок, которая связана с необходимостью иметь различные блоки БП ($a_5\ a_6\ a_9\ a_{10}$) в каждом информационном разряде $a_5 - a_{11}$.

Несмотря на то, что базовые геометрические образы функций в координатах $a_5\ a_6\ a_9\ a_{10}$ для информационных ($a_1 - a_4$) и контрольных ($x_1 - x_4$) разрядов различны, они имеют одинаковую структуру построения, которая определяется одинаковым расположением составляющих их функций ($m_1 \leftrightarrow 5\ldots$, $m_2 \leftrightarrow 0\ldots$, $m_3 \leftrightarrow 7\ldots$, $m_4 \leftrightarrow 2\ldots$, $m_5 \leftrightarrow 4\ldots$, $m_6 \leftrightarrow 1\ldots$, $m_7 \leftrightarrow 6\ldots$, $m_8 \leftrightarrow 3\ldots$,) в этих координатах базового геометрического образа. Именно это позволяет иметь один блок БП ($a_5\ a_6\ a_9\ a_{10}$) для информационных ($a_1 - a_4$) и контрольных ($x_1 - x_4$) разрядов.

Учитывая это обстоятельство, используем одинаковый базовый геометрический образ, который был выбран для контрольных разрядов (см. рис. 5.34), для информационных разрядов ($a_1 - a_4$) систематического кода. Тогда, выполняя все перечисленные выше правила построения геометрических образов исправленных сигналов $a'_1 - a'_{11}$, получим один блок БП ($a_5\ a_6\ a_9\ a_{10}$) для всех информационных и контрольных разрядов этого кода, где для каждого из разрядов меняются только сочетание прямых и инверсных сигналов на входных шинах этого блока.

Выходные сигналы этого блока, которые поступают на входные шины соответствующих блоков БП($a_1 - a_4$), задаются данными табл. 5.4.1.

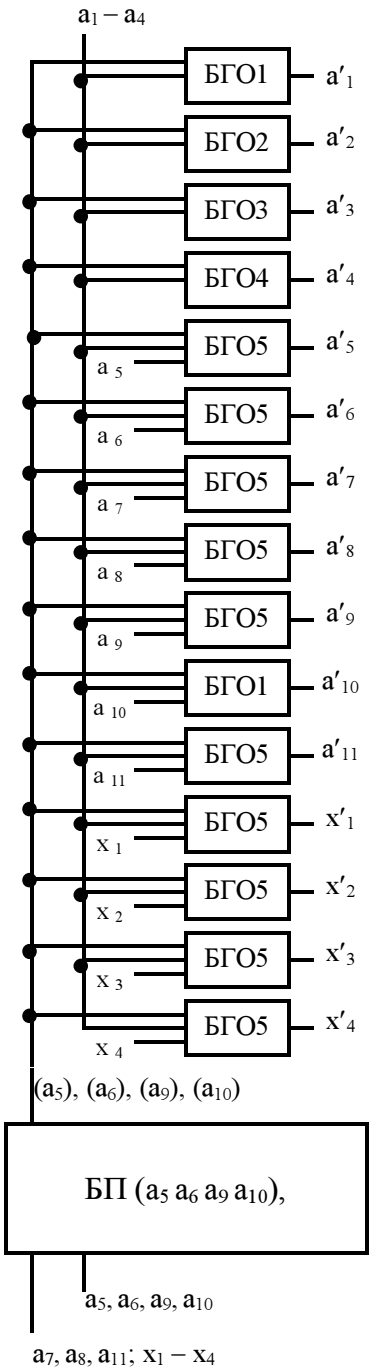


Рис. 5.40

Таблица 5.4.1

Разряды	Сигналы блока БП($a_5\, a_6\, a_9\, a_{10}$)							
	(a_5)	($\underline{a}_5$)	(a_6)	($\underline{a}_6$)	(a_9)	($\underline{a}_9$)	(a_{10})	($\underline{a}_{10}$)
$a_1 - a_4$	*		*		*		*	
a_5	*			*	*		*	
a_6		*	*		*		*	
a_7		*	*			*		*
a_8	*			*		*		*
a_9		*		*		*	*	
a_{10}		*		*	*			*
a_{11}	*		*		*			*
x_1	*		*		*		*	
x_2	*		*			*		*
x_3		*	*			*	*	
x_4	*		*			*	*	

Синтезированная таким образом структурная схема устройства исправления одиночных ошибок (рис. 5.40) содержит общий функциональный блок поворота координат БП ($a_5\, a_6\, a_9\, a_{10}$), прямые и инверсные сигналы (a_5), (a_6), (a_9), (a_{10}) которого поступают на первые входные шины блоков БГО1 – БГО5. На вторые входные шины этих блоков подаются сигналы $a_1 - a_4$ первых четырех информационных разрядов. Блоки БГО1 – БГО4 формируют соответственно исправленные сигналы $a'_1 - a'_4$, а одиннадцать блоков БГО5 формируют аналогичные сигналы $a'_5 - a'_{11}$; $x'_1 - x'_4$, где в каждом из этих блоков имеется третий вход, на который поступают соответствующие этому блоку сигналы $a_5 - a_{11}$; $x_1 - x_4$.

При уменьшении числа информационных разрядов кода происходит исчезновение соответствующих им блоков БГО5, а также изменяется общий блок БП (a_5 a_6 a_9 a_{10}), как это представлено выше на функциональных схемах рис. 5.39, *а – е*.

Функциональная схема БП (a_5 a_6 a_9 a_{10}) состоит из трех четырехвходовых функций, геометрический образ которых обладает полной симметрией относительно осей двухмерного пространства, и одной трехвходовой функцией, имеющей ограниченную симметрию в этой мерности пространства. Учитывая это обстоятельство, в соответствии с данными табл. 5.3.1, 5.3.2 общее число эквивалентных схем этого блока равно 10616832, в каждой из которых меняются только местами либо одновременно инвертируются сигналы на входных шинах составляющих их схем.

Математиков не должен смущать практический подход, ибо решение практических задач часто приводит к глубокому исследованию «абстрактных» математических объектов.

Э. Берлекэмп

Есть одна опасность в обращении к классическим работам, кажется, что на них все ссылаются, но редко кто их читает.

Г. Глинский

Глава 6

ПРОДОЛЖЕНИЕ ГЕОМЕТРИЧЕСКОГО СИНТЕЗА КОДОВ, ИСПРАВЛЯЮЩИХ ОШИБКИ

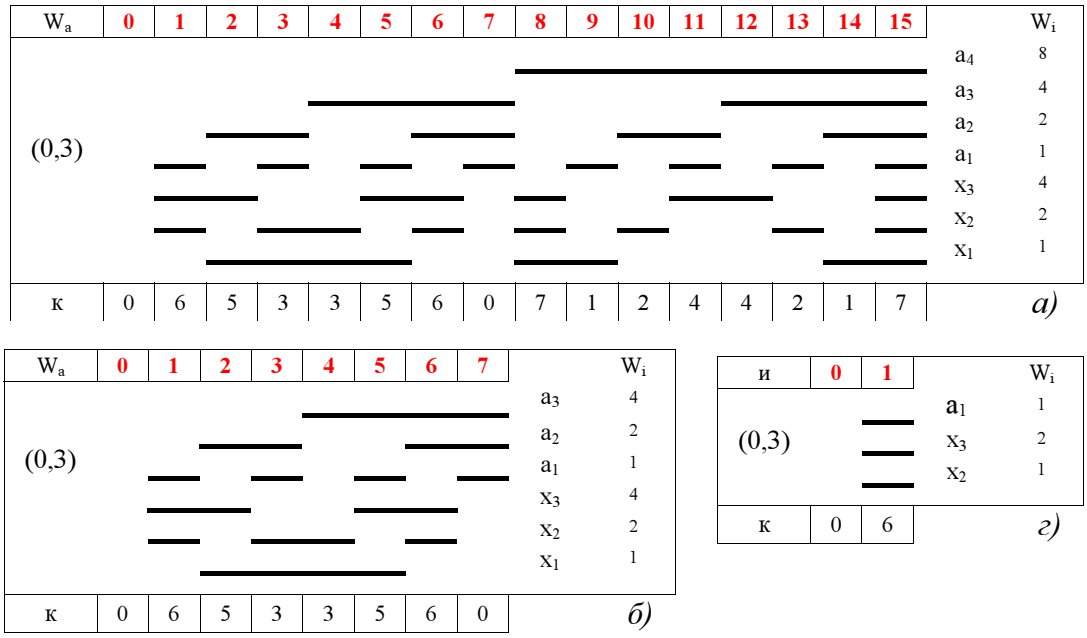
История исследования систематических совершенных кодов с информационной частью основания $n = 2^4$ начинается с работы математика Р. Фишера (1942) [1], где представлен один из этих кодов. Аналогичный код этого основания был в качестве примера приведен в работе К. Шеннона (1948) [3] и в дальнейшем был обобщен М. Голеем (1949) [4]. Однако в литературе эти коды обычно связывают с именем Р. Хемминга (1950), который в [5] представил некоторое число этих кодов, позволяющих как обнаружить, так и исправить некоторые типы ошибок. Историческая неточность такого представления совершенных кодов однозначно отмечается в книге Э. Берлекэмпа [6], внесшего существенный вклад в теорию кодов, исправляющих ошибки.

К совершенным систематическим кодам, исправляющим определенные группы ошибок (одиночные; одиночные и двойные; одиночные, двойные и тройные и т.д.), необходимо отнести также мажоритарный код для основания $n = 2$, который представлен в работе Дж. фон Неймана (1952) [2]. Совершенные коды этого основания содержат один информационный разряд и определенное количество контрольных разрядов, совпадающих с информационным. Число этих контрольных разрядов равно 2, 4, 6, …, что гарантирует кодовое расстояние между цифрами **0** и **1**, – соответственно 3, 5, 7, … . Эти числа и определяют минимальное кодовое расстояние, которое должно быть между штатными цифрами любой позиционной системы счисления, обеспечивающими исправление соответствующих им групп ошибок.

Вместе с тем необходимо отметить, что уже на начальных этапах исследований совершенных кодов [9] высказывалось мнение, что «имеются некоторые результаты, показывающие, что совершенных кодов мало, и кажется вполне правдоподобным, что не существует других совершенных кодов», кроме известных в это время. В дальнейшем это предположение получило «подтвер-

ждение» в работах финских [8] и советских [7] авторов, которые в 1972 г. «строго доказали» отсутствие каких-либо совершенных двоичных кодов, отличных от тех, которые были им известны. Позднее эти же авторы и американский ученый Ван Линт «получили полное решение вопроса о нахождении всех совершенных кодов, использующих p^k элементарных сигналов, где p – произвольное число, а k – любое целое положительное число; таких кодов оказалось крайне мало» [10].

Ошибочность этих научных выводов показана в предыдущих главах книги: число совершенных кодов, исправляющих все одиночные ошибки, как в информационной, так и контрольной части систематического кода, где информационная часть кода представляется в основном двоичном варианте, определяется выражением $S = 2^k (i\,!)$, в котором число контрольных разрядов k = 3, 4, 5, … , а соответствующее им число информационных разрядов определяется выражением $i = 2^k - 1 - k$ (i = 4, 11, 26, …), неограниченно велико. Причем если в качестве информационной части кода использовать любые двоичные коды



оснований систем счисления $n = 2^i$, тогда число совершенных кодов будет увеличено: $S = 2^k (i!)\,(2^i!)$.

Число квазисовершенных кодов, которые являются образующими из совершенных кодов оснований систем счисления $n = 2^4, 2^{11}, 2^{26}\ldots$, также необозримо велико.

На рис. 6.1, *а* представлен пример совершенного кода (k = 3, i = 4) соответственно основанию $n = 2^4$ в его информационной части (координаты кода 0,3

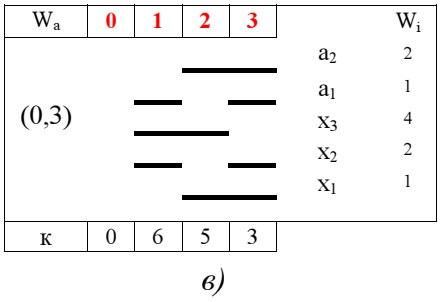


Рис. 6.1

табл. 4.9.2, а на рис. 6.1, *б* – квазисовершенный код (k = 3, i = 3) основания n = 7, который получен из совершенного кода основания n = 16. Аналогичным образом формируются квазисовершенный код (k = 3, i = 2) основания n = 4 (см. рис. 6.1, *в*), а также совершенный код (k = 2, i = 1) основания n = 2 (см. рис. 6.1, *г*), который является мажоритарным кодом Дж. фон Неймана.

Подобным образом из следующего совершенного кода (k = 4, i = 11) можно получить все квазисовершенные коды, которые при четырех контрольных разрядах исправляют все одиночные ошибки в обеих частях систематического кода оснований от 2^{10} до 2^5 и т.д.

Поскольку совершенные коды играют значительную роль в системах передачи информации и еще большую в цифровых и логических системах управления, где кроме исправления ошибок при передаче и хранении информации они могут успешно использоваться в блоках безошибочной машинной арифметики, необходимо продолжить анализ и синтез таких совершенных и квазисовершенных кодов.

При этом, кроме применения двоичных кодов в информационной части систематического кода, необходимо рассмотреть использование здесь недвоичных кодов, например многофазных, интегральных, обычных цифровых и т.д. и не только для исправления всех одиночных ошибок, но и расширения спектра исправляемых ошибок.

С этой целью первоначально исследуем кодовые расстояния между кодовыми комбинациями основного двоичного кода. Для этого в цифровом пространстве этих кодовых комбинаций, например для основания n =16, кодовые расстояния могут представляться как результат операции сложения или вычитания по «модулю 2» операндов **A** и **B** и будут изображены соответствующей таблицей. Поскольку операция сложения и вычитания по «модулю 2» одна и та же, то такая таблица, если прямой и обратный код преобразуются один в другой простым инвертированием сигналов всех его разрядов, а именно это характерно для основного двоичного кода, будет симметрична относительно главной и побочной диагонали.

На рис. 6.2, *а* приведена часть такой таблицы, включающая значения кодовых расстояний на главной и побочной диагонали, а также значения над этими диагоналями. Первая строка этой таблицы содержит кодовые расстояния между первой кодовой комбинацией, которая определяется цифрой **0**, и всеми остальными. Вторая строка – между второй кодовой комбинацией (цифра **1**) и последующими кодовыми комбинациями и т.д. Последняя строка таблицы определяет кодовое расстояние между кодовыми комбинациями, соответствующими цифрам **14** и **15**.

Вполне очевидно, что данные этой таблицы являются суммой четырех аналогичных таблиц (рис. 6.2, *б – д*), которые задают кодовые расстояния между сигналами каждого отдельного разряда соответственно $a_1 – a_4$. Все эти таблицы имеют одинаковую структуру, которая определяется свойствами основного двоичного кода.

Остановимся на данных кодовых расстояний рис. 6.2, *а*. Для основания n = 16 (цифры **0**–**15**) на побочной диагонали таблицы располагаются значения кодовых расстояний d = 4; для основания n = 8 (цифры **0**–**7**) – значения d = 3; для основания n = 4 (цифры **0**–**7**) – значения d = 2.

На побочных диагоналях таблиц кодовых расстояний отдельных разрядов (см. рис. 6.2, *б – д*) располагаются значения d = 1 и d = 0.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							a_4								
			a_3												
	a_2														
a_1															
0	1	1	2	1	2	2	3	1	2	2	3	2	3	3	4
	0	2	1	2	1	3	2	2	1	3	2	3	2	4	
		0	1	2	3	1	2	2	3	1	2	3	4		
			0	3	2	2	1	3	2	2	1	4			
				0	1	1	2	2	3	3	4				
					0	2	1	3	2	4					
						0	1	3	4						
							0	4							

а)

a_1															
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
		0	1	0	1	0	1	0	1	0	1	0	1		
			0	1	0	1	0	1	0	1	0	1			
				0	1	0	1	0	1	0	1				
					0	1	0	1	0	1					
						0	1	0	1						
							0	1							

б)

a_2															
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
		0	0	1	1	0	0	1	1	0	0	1	1	0	0
			0	1	1	0	0	1	1	0	0	1	1	0	0
					0	1	1	0	0	1	1	0	0	1	1
					0	1	1	0	0	1	1	0	0	1	1
						0	0	1	1	0	0	1	1	0	0
							0	1	1	0	0	1	1	0	0

в)

a_3															
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	0	1	1	1	1	0	0	0	0	1	1	1	
		0	0	1	1	1	1	0	0	0	0	1	1		
			0	1	1	1	1	0	0	0	0	1			
				0	0	0	0	1	1	1	1				
					0	0	0	1	1	1					
						0	0	1	1						
							0	1							

г)

Рис. 6.2 (начало)

a_4															
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
		0	0	0	0	0	0	1	1	1	1	1	1		
			0	0	0	0	0	1	1	1	1	1			
				0	0	0	0	1	1	1	1				
					0	0	0	1	1	1					
						0	0	1	1						
							0	1							

д)

Рис. 6.2 (продолжение)

Сумма кодовых расстояний двух кодовых комбинаций, сумма весовых значений которых равна (n – 1), определяется значением кодового расстояния побочной диагонали этих таблиц. Для рис. 6.2, *а* при n = 16 это d = 4, а для всех остальных таблиц (рис. 6.2, *б* – *д*) кодовое расстояние d = 1.

Все 192 совершенных кода основания $n = 2^4$, представленные в табл. 4.2.2, соответствуют четырем типам таблиц кодовых расстояний между кодовыми комбинациями цифр **0**–**15**.

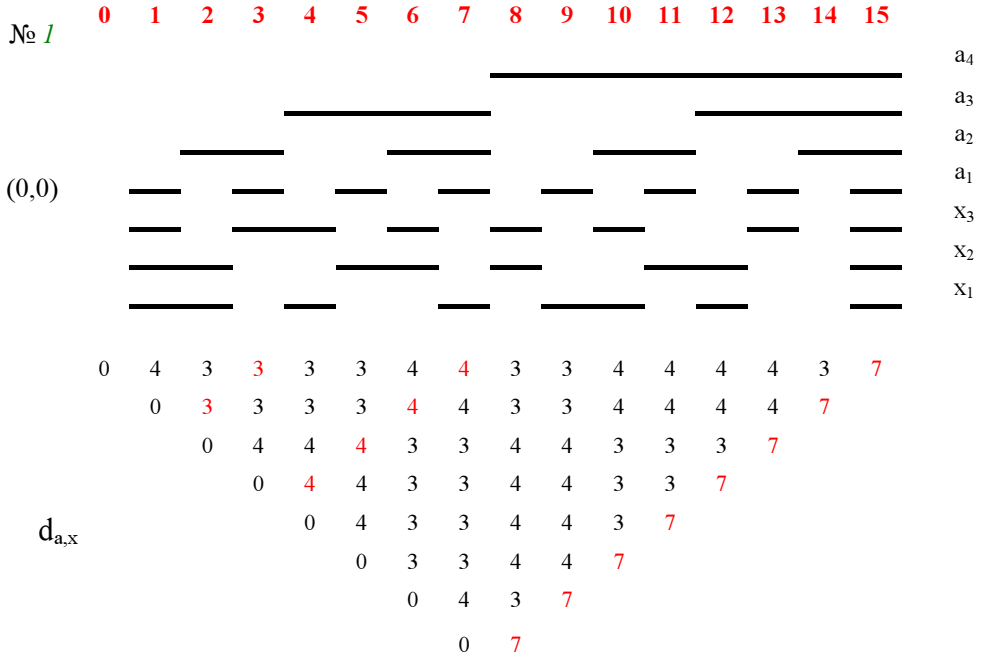
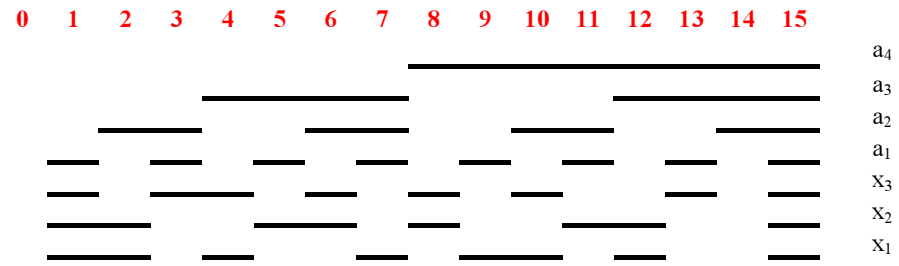


Рис. 6.3

№ *2*

(1,0)



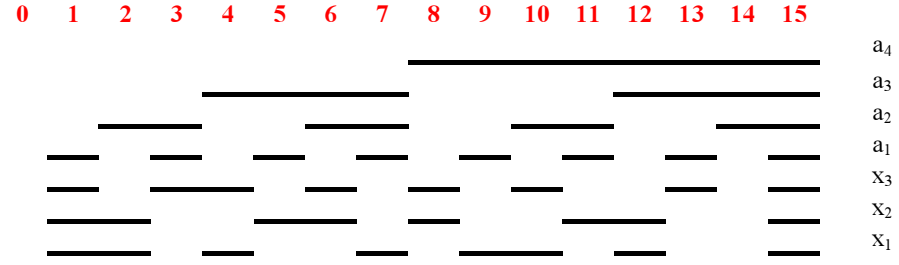
$d_{a,x}$

0	3	4	3	3	4	3	4	3	4	3	4	4	3	4	7
	0	3	4	4	3	4	3	4	3	4	3	3	4	7	
		0	3	3	4	3	4	3	4	3	4	4	7		
			0	4	3	4	3	4	3	4	3	7			
				0	3	4	3	4	3	4	7				
					0	3	4	3	4	7					
						0	3	4	7						
							0	7							

Рис. 6.4

№ *3*

(2,0)



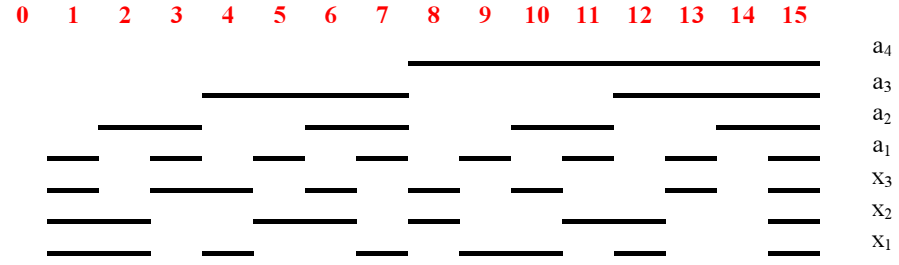
$d_{a,x}$

0	3	3	4	4	3	3	4	3	4	4	3	3	4	4	7
	0	4	3	3	4	4	3	4	3	3	4	4	3	7	
		0	3	3	4	4	3	4	3	3	4	4	7		
			0	4	3	3	4	3	4	4	3	7			
				0	3	3	4	3	4	4	7				
					0	4	3	4	3	7					
						0	3	4	7						
							0	7							

Рис. 6.5

№ *4*

(3,0)



$d_{a,x}$

0	3	3	4	3	4	4	3	4	3	3	4	3	4	4	7
	0	4	3	4	3	3	4	3	4	4	3	4	3	7	
		0	3	4	3	4	3	4	3	4	3	4	7		
			0	3	4	3	4	3	4	3	4	7			
				0	3	4	3	4	3	4	7				
					0	4	3	4	3	7					
						0	4	3	7						
							0	7							

Рис. 6.6

На рис. 6.3–6.6 приведены эти данные соответственно для кодов табл. 4.2.2 с координатами (0, 0), (1, 0), (2, 0), (3, 0).

Обозначим эти кодовые комбинации курсивом под номерами *№ 1*, *№ 2*, *№ 3*, *№ 4* и в координатах табл. 4.2.2 разместим под этими номерами соответствующие им системы кодовых расстояний.

Результаты этого отражены в табл. 6.1.

Поскольку сигналы контрольных разрядов здесь, также как сигналы информационных разрядов кодов, при переводе из прямого кода в обратный преобразуются один в другой простым инвертированием сигналов, то сложение кодовых расстояний двух кодовых комбинаций, сумма весовых значений информационной части кода которых (n – 1) = 15, будет равно кодовому расстоянию d = 7, находящемуся на побочной диагонали таблиц рис. 6.3–6.6.

Таблица 6.1

	0	1	2	3	4	5	6	7
	№							
0	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
1	*2*	*2*	*2*	*2*	*2*	*2*		
2	*3*	*3*	*3*	*3*	*3*		*3*	
3	*4*	*4*	*4*	*4*	*4*			*4*
4	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
5	*3*	*3*	*3*	*3*	*3*		*3*	
6	*2*	*2*	*2*	*2*	*2*	*2*		
7	*4*	*4*	*4*	*4*	*4*			*4*
8	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
9	*4*	*4*	*4*	*4*	*4*			*4*
10	*3*	*3*	*3*	*3*	*3*		*3*	
11	*2*	*2*	*2*	*2*	*2*	*2*		
12	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
13	*2*	*2*	*2*	*2*	*2*	*2*		
14	*4*	*4*	*4*	*4*	*4*			*4*
15	*3*	*3*	*3*	*3*	*3*		*3*	
16	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
17	*4*	*4*	*4*	*4*	*4*			*4*
18	*2*	*2*	*2*	*2*	*2*	*2*		
19	*3*	*3*	*3*	*3*	*3*		*3*	
20	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
21	*3*	*3*	*3*	*3*	*3*		*3*	
22	*4*	*4*	*4*	*4*	*4*			*4*
23	*2*	*2*	*2*	*2*	*2*	*2*		

	0	1	2	3	4	5	6	7
	№							
24								
25				*2*	*2*			
26			*3*		*3*			
27			*4*	*4*				
28								
29			*3*			*3*		
30				*2*	*2*			
31			*4*	*4*				
32								
33			*4*	*4*				
34			*3*		*3*			
35				*2*	*2*			
36								
37				*2*	*2*			
38			*4*	*4*				
39			*3*			*3*		
40								
41			*4*	*4*				
42				*2*	*2*			
43			*3*		*3*			
44								
45			*3*		*3*			
46			*4*	*4*				
47				*2*	*2*			

6.1. Синтез кодов с информационной частью в основном двоичном коде основания $n = 2^4$, исправляющих одиночные и двойные ошибки

Кодовые комбинации, уже имеющие одиночные ошибки, при повторном возникновении в них одиночных ошибок будут содержать двойные ошибки либо могут вернуться в исходное безошибочное состояние. Очевидно, что исправление в этих кодах групп одиночных и двойных ошибок можно выполнить, если в таблице их кодовых расстояний будет этих значений не менее пяти.

Применение в контрольной части такого систематического кода, например совершенного кода, исправляющего все одиночные ошибки основания $n = 2^4$, которое должно совпадать с основанием информационной части синтезируемого кода, не позволит обеспечить необходимое минимальное кодовое расстояние. В качестве иллюстрации этого утверждения выберем такой синтезированный код, где в его контрольной части выбран код из табл. 4.2.2 с координатами 0, 0, и построим таблицу кодовых расстояний этого систематического кода (рис. 6. 7).

Синтезированный подобным образом код будет иметь информационную (a_1 – a_4) и контрольную (x_2 – x_8) части, а также соответствующие им кодовые расстояния. Сумма этих кодовых расстояний, определяющая общие кодовые расстояния систематического кода, содержит значения меньше пяти (d = 4) и, следовательно, не дает возможности исправлять одновременно все одиночные и двойные ошибки.

Для того чтобы этот код позволял исправлять все одиночные и двойные ошибки, необходимо в его контрольную часть добавить еще один разряд – x_1, который имеет кодовые расстояния 0–1 (см. рис. 6.7).

Сумма кодовых расстояний (a_1 – a_4) и (x_1 – x_8) представляет кодовые расстояния синтезированного совершенного систематического кода, позволяющего исправлять все одиночные и двойные ошибки. Очевидно, что число таких совершенных кодов основания $n = 2^4$ представляется табл. 4.2.2, которая определяет контрольную часть этого кода.

Квазисовершенные коды, позволяющие исправлять все одиночные и двойные ошибки меньших оснований систем счисления, просто определяются из рис. 6.6, где для основания $n = 2^3$ информационная и контрольная части кода содержат соответственно разряды – (a_1 – a_3), (x_1 – x_7); для основания $n = 2^2$ это – (a_1 – a_2), (x_1 – x_6); а для основания n =2 происходит превращение кода в совершенный код Дж. Неймана, где общее число разрядов кода равно пяти.

Соотношения весовых значений кодовых комбинаций информационной (**0**–**15**) и контрольной (0, 30, 39, 57, 75, 85, 108, 114, 141, 147, 170, 180, 192, 216, 225, 255) частей совершенного систематического кода позволяют представить в соответствующем многомерном цифровом пространстве штатные и ошибочные значения цифр всех этих оснований систем счисления и на их основе построить геометрические образы исправленных разрядов систематического кода.

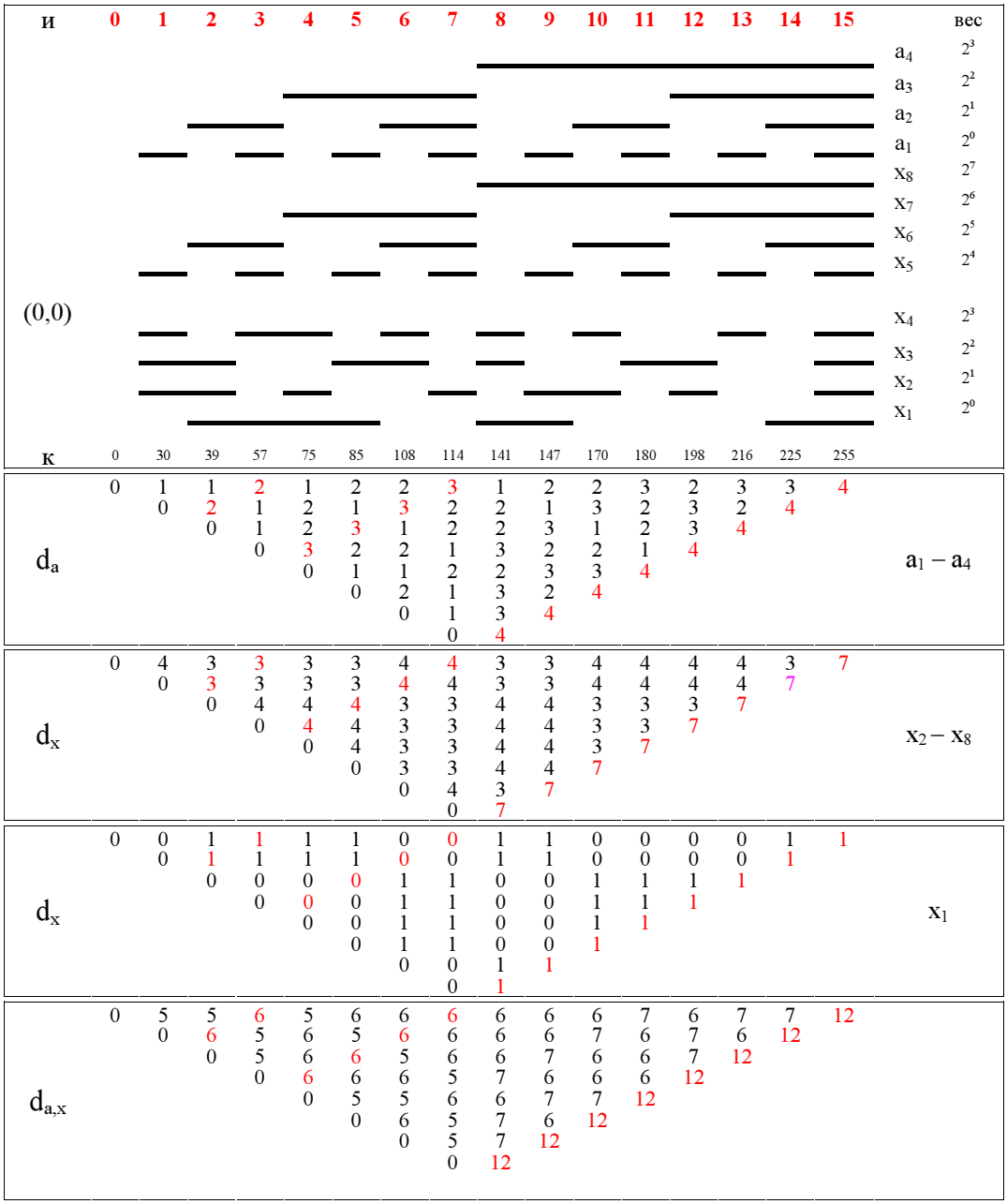


Рис. 6.7

Результаты таких построений для основания системы счисления $n = 2^4$ приведены в табл. 6.1.1, где красным цветом отмечены штатные цифры, черным – цифры при одиночной ошибке в разрядах систематического кода, а с одиночными и двойными штрихами – цифры при двойных ошибках в этих разрядах кода.

Таблица 6.1.1

Контр. часть кода	Информационная часть кода															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**0**	0	0	0'	0	0'	0'		0	0'	0'		0'			
1	0	0'	0'		0'				0'							
2	0	0'	0'		0'				0'							
3	0"		2"		4"					9"						
4	0	0'	0'		0'				0'							
5	0"		2"			5"			8				12"			
6	0"	1"	2"													
7	2'		2	2'			2'				2'					
8	0	0'	0'		0'				0'							
9	0"			3"	4"				8"							
10	0"	1"			4"						10"					
11	4'				4	4'	4'						4'			
12	0"	1"					6"		8"							
13	8'								8	8'	8'		8'			
14	1'	1		1'		1'				1'						
15		1"	2"		4"				8							
16	0	0'	0'		0'				0'							
17	0"			3"		5"				9"						
18	0"	1"						7"		9"						
19		9'							9'	9		9'		9'		
20	0"	1"										11"				
21		5'			5'	5		5'						5'		
22	1'	1		1'		1'				1'						
23		1"	2"			5"				9"						
24	0"	1"		3"										13"		
25		3'	3'	3				3'				3'				
26	1'	1		1'		1'				1'						
27		1"		3"	4"					9"						
28	1'	1		1'		1'				1'						
29		1"		3"		5"			8"							
30	1	**1**	1'	1	1'	1		1'	1'	1		1'		1'		
31	1'	1		1'		1'				1'						
32	0	0'	0'		0'				0'							
33	0"		2"	3"											14"	
34	0"		2"					7"			10"					
35	2'		2	2'			2'				2'					
36	0"		2				6"					11"				
37	2'		2	2'			2'				2'					
38	2'		2	2'			2'				2'					
39	2	2'	**2**	2	2'		2	2'	2'		2	2'			2'	
40	0"			3"			6"				10"					
41		3'	3'	3				3'				3'				
42			10'						10'		10	10'			10'	
43			2"	3"	4"						10"					
44			6'		6'		6	6'							6'	
45			2"	3"			6"		8"							
46		1"	2"				6"				10"					
47	2'		2	2'			2'				2'					
48	0"			3"				7"				11"				
49		3'	3'	3				3'				3'				
50				7'		7'	7'	7								7'
51			2"	3"				7"		9"						

Продолжение табл. 6.1.1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
52				11'						11'	11'	11				11'
53			2"	3"								11"				
54		1"	2"					7"				11"				
55	2'		2	2'			2'				2'					
56		3'	3'	3				3'				3'				
57	3'	3	3	3		3'	3'	3		3'	3'	3				3'
58		1"		3"				7"			10"					
59		3'	3'	3				3'				3'				
60		1"		3"			6"					11"				
61		3'	3'	3				3'				3'				
62	1'	1		1'		1'				1'						
63		1"	2"	3"												15"
64	0	0'	0'		0'				0'							
65	0"				4"	5"									14"	
66	0"				4"			7"					12"			
67	4'				4	4'	4'						4'			
68	0"					5"	6"						12"			
69		5'			5'	5		5'						5'		
70					12'				12'				12	12'	12'	
71			2"		4"	5"							12"			
72	0"				4"		6"							13"		
73	4'				4	4'	4'						4'			
74	4'				4	4'	4'						4'			
75	4	4'	4'		4	4	4	4'	4'				4	4'	4'	
76			6'		6'		6	6'							6'	
77					4"	5"	6"		8"							
78		1"			4"		6"						12"			
79	4'				4	4'	4'						4'			
80	0"					5"		7"						13"		
81		5'			5'	5		5'						5'		
82				7'		7'	7'	7								7'
83					4"	5"		7"		9"						
84		5'			5'	5		5'						5'		
85	5'	5		5'	5	5	5'	5		5'			5'	5		5'
86		1"						7"					12"			
87		5'			5'	5		5'						5'		
88						13'				13'			13'	13		13'
89				3"	4"	5"								13"		
90		1"			4"			7"						13"		
91	4'				4	4'	4'						4'			
92		1"				5"	6"							13"		
93		5'			5'	5		5'					5'	13'		
94	1	1		1		1				1						
95		1"			4"	5"										15"
96	0"						6"	7"							14"	
97							14'				14'		14'		14	14'
98				7'		7'	7'	7								7'
99			2"		4"			7"							14"	
100			6'		6'		6	6'							6'	
101			2"			5"	6"								14"	
102			2"				6"	7"					12"			
103	2'		2	2'			2'				2'					

Продолжение табл. 6.1.1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
104			6'		6'		6	6'							6'	
105				3"	4"		6"								14"	
106					4"		6"	7"			10"					
107	4'				4	4'	4'						4'			
108	6'		6	6'	6	6'	6	6			6'		6'		6	6'
109			6'		6'		6	6'							6'	
110			6'		6'		6	6'							6'	
111			2"		4"		6"									15"
112				7'		7'	7'	7								7'
113				3"		5"		7"							14"	
114		7'	7'	7	7'	7	7	7				7'		7'	7'	7
115				7'		7'	7'	7								7'
116							6"	7"				11"				
117		5'			5'	5		5'						5'		
118				7'		7'	7'	7								7'
119			2"					7"								15"
120				3"			6"	7"						13"		
121		3'	3'	3				3'				3'				
122				7'		7'	7'	7								7'
123				3"	4"			7"								15"
124			6'		6'		6	6'							6'	
125				3"		5"	6"									
126		1"					6"	7								15"
127								15'				15'		15'	15'	15
128	0	0'	0'		0'				0'							
129	0"								8"	9"					14"	
130	0"									9"	10"		12"			
131		9'							9'	9		9'		9'		
132	0"								8"			11"	12"			
133	8'								8	8'	8'		8'			
134					12'				12'				12	12'	12'	
135			2"						8"	9"			12"			
136	0"								8"		10"			13"		
137	8'								8	8'	8'		8'			
138			10'						10'		10	10'			10'	
139					4"				8"	9"	10"					
140	8'								8	8'	8'		8'			
141	8	8'	8'		8'				8	8	8	8'	8	8'	8'	
142		1"							8"		10"		12"			
143	8'								8	8'	8'		8'			
144	0"									9"		11"		13"		
145		9'							9'	9		9'		9'		
146		9'							9'	9		9'		9'		
147	9'	9		9'		9'			9	9	9'	9	9'	9		9'
148				11'						11'	11'	11				11'
149									8"	9"		11"				
150		1"								9"		11"	12"			
151		9'							9'	9		9'		9'		
152						13'				13'			13'	13		13'
153				3"					8"	9"				13"		
154		1"								9"	10"			13"		
155		9'							9'	9		9'		9'		

Продолжение табл. 6.1.1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
156		1"							8"			11"		13"		
157	8'								8	8'	8'		8'			
158	1'	1		1'		1'				1'						
159		1"							8"	9"						15"
160	0"										10"	11"			14"	
161							14'				14'		14'		14	15'
162			10'						10'		10	10'			10'	
163			2"							9"	10"				14"	
164				11'						11'	11'	11				11'
165			2"						8"			11"			14"	
166			2"								10"	11"	12"			
167	2'		2	2'		2'				2'						
168			10'						10'		10	10'			10'	
169				3"					8"		10"				14"	
170	10'		10	10'			10'		10	10'	**10**	10	10'		10	10'
171			10'						10'		10	10'			10'	
172							6"		8"		10"	11"				
173	8'								8	8'	8'		8'			
174			10'						10'		10	10'			10'	
175			2"						8"		10"					15"
176				11'						11'	11'	11				11'
177				3"						9"		11"			14"	
178										9"	10"	11"				
179		9'							9'	9		9'		9'		
180		11'	11'	11				11'	11'	11	11	**11**		11'	11'	11
181				11'						11'	11'	11				11'
182				11'						11'	11'	11				11'
183			2"							9"		11"				15"
184				3"							10"	11"		13"		
185		3'	3'	3				3'				3'				
186			10'						10'		10	10'			10'	
187				3"						9"	10"					15"
188				11'						11'	11'	11				11'
189				3"					8"			11"				15"
190		1"									10"	11"				15"
191								15'				15'		15'	15'	15
192	0"												12"	13"	14"	
193							14'				14'		14'		14	14'
194					12'				12'				12	12'	12'	
195					4"					9"			12"		14"	
196					12'				12'				12	12'	12'	
197						5"			8"				12"		14"	
198	12'				12	12'	12'		12	12'	12'		**12**	12	12	12'
199					12'				12'				12	12'	12'	
200						13'				13'			13'	13		13'
201					4"				8"					13"	14"	
202					4"						10"		12"	13"		
203	4'				4	4'	4'						4'			
204							6"		8"				12"	13"		
205	8'								8	8'	8'		8'			
206					12'				12'				12	12'	12'	
207					4"				8"				12"			15"
208						13'				13'			13'	13		13'
209						5"				9"				13"	14"	
210								7"		9"			12"	13"		
211		9'							9'	9		9'		9'		

Окончание табл. 6.1.1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
212						5"						11"	12"	13"		
213		5'			5'	5		5'						5'		
214					12'				12'				12	12'	12'	
215						5"				9"			12"	12"		15"
216		13'			13'	13		13'	13'	13		13'	13	13	13'	13
217						13'				13'			13'	13		13'
218						13'				13'			13'	13		13'
219					4"					9"				13"		15"
220						13'				13'			13'	13		13'
221						5"			8"					13"		15"
222		1"											12"	13"		15"
223								15'				15'		15'	15'	15
224							14'				14'		14'		14	14'
225			14'		14'		14	14'	14'		14	14'	14	14'	14	14
226								7"			10"		12"		14"	
227							14'				14'		14'		14	14'
228							6"					11"	12"		14"	
229							14'				14'		14'		14	14'
230					12'				12'				12	12'	12'	
231			2"										12"		14"	15"
232							6"				10"			13"	14"	
233							14'				14'		14'		14	14'
234			10'						10'		10	10'			10'	
235					4"						10"				14"	15"
236			6'		6'		6	6'							6'	
237							6"		8"						14"	15"
238							6"				10"		12"			15"
239								15'				15'		15'	15'	15
240								7"				11"		13"	14"	
241							14'				14'		14'		14	14'
242				7'		7'	7'	7								7'
243								7"		9"					14"	15"
244				11'						11'	11'	11				11'
245						5"						11"			14"	15"
246												11"	12"			15"
247								15'				15'		15'	15'	15
248						13'				13'			13'	13		13'
249				3"										13"	14"	15"
250								7"			10"			13"		15"
251								15'				15'		15'	15'	15
252						6"						11"		13"		15"
253								15'				15'		15'	15'	15
254								15'				15'		15'	15'	15
255				15'		15'	15'	15		15'	15'	15	15'	15	15	15

Дальнейшие действия по определению геометрических образов исправленных сигналов информационных и контрольных разрядов систематического кода, а также реализация покрытия этих геометрических образов и нахождения соответствующих им логических функций не представляют какой-либо сложности.

При этом число кодов контрольных разрядов, позволяющих исправлять все одиночные и двойные ошибки, будет определяться данными табл. 6.1.2, аналогичной табл. 4.2.2.

Таблица 6.1.2

	0	1	2	3	4	5	6	7
0	0 14 7 9 11 5 12 2 13 3 10 4 6 8 1 15	14 0 9 7 5 11 2 12 3 13 4 10 8 6 15 1	7 9 0 14 12 2 11 5 10 4 13 3 1 15 6 8	11 5 12 2 0 14 7 9 6 8 1 15 13 3 10 4	13 3 10 4 6 8 1 15 0 14 7 9 11 5 12 2	9 7 14 0 2 12 5 11 4 10 3 13 15 1 8 6	5 11 2 12 14 0 9 7 8 6 15 1 3 13 4 10	3 13 4 10 8 6 15 1 14 0 9 7 5 11 2 12
1	0 7 14 9 11 12 5 2 13 10 3 4 6 1 8 15	7 0 9 14 12 11 2 5 10 13 4 3 1 6 15 8	14 6 0 7 5 2 11 12 3 4 13 10 8 15 6 1	11 12 5 2 0 7 14 9 6 1 8 15 13 10 3 4	13 10 3 4 6 1 8 15 0 7 14 9 11 12 5 2	9 14 7 0 2 5 12 11 4 3 10 13 15 8 1 6	12 11 2 5 7 0 9 14 1 6 15 8 10 13 4 3	10 13 4 3 1 6 15 8 7 0 9 14 12 11 2 5
2	0 11 7 12 14 5 9 2 13 6 10 1 3 8 4 15	11 0 12 7 5 14 2 9 6 13 1 10 8 3 15 4	7 12 0 11 9 2 14 5 10 1 13 6 4 15 3 8	14 5 9 2 0 11 7 12 3 8 4 15 13 6 10 1	13 6 10 1 3 8 4 15 0 11 7 12 14 5 9 2	12 7 11 0 2 9 5 14 1 10 6 13 15 4 8 3	5 14 2 9 11 0 12 7 8 3 15 4 6 13 1 10	6 13 1 10 8 3 15 4 11 0 12 7 5 14 2 9
3	0 13 11 6 7 10 12 1 14 3 5 8 9 4 2 15	13 0 6 11 10 7 1 12 3 14 8 5 4 9 15 2	11 6 0 13 12 1 7 10 5 8 14 3 2 15 9 4	7 10 12 1 0 13 11 6 9 4 2 15 14 3 5 8	14 3 5 8 9 4 2 15 0 13 11 6 7 10 12 1	6 11 13 0 1 12 10 7 8 5 3 14 15 2 4 9	10 7 1 12 13 0 6 11 4 9 15 2 3 14 8 5	3 14 8 5 4 9 15 2 13 0 6 11 10 7 1 12
4	0 14 11 5 7 9 12 2 13 3 6 8 10 4 1 15	14 0 5 11 9 7 2 12 3 13 8 6 4 10 15 1	11 5 0 14 12 2 7 9 6 8 13 3 1 15 10 4	7 9 12 2 0 14 11 5 10 4 1 15 13 3 6 8	13 3 6 8 10 4 1 15 0 14 11 5 7 9 12 2	5 11 14 0 2 12 9 7 8 6 3 13 15 1 4 10	9 7 2 12 14 0 5 11 4 10 15 1 3 13 8 6	3 13 8 6 4 10 15 1 14 0 5 11 9 7 2 12
5	0 7 11 12 14 9 5 2 13 10 6 1 3 4 8 15	7 0 12 11 9 14 2 5 10 13 1 6 4 3 15 8	11 12 0 7 5 2 14 9 6 1 13 10 8 15 3 4	14 9 5 2 0 7 11 12 3 4 8 15 13 10 6 1	13 10 6 1 3 4 8 15 0 7 11 12 14 9 5 2	12 11 7 0 2 5 9 14 1 6 10 13 15 8 4 3	9 14 2 5 7 0 12 11 4 3 15 8 10 13 1 6	10 13 1 6 4 3 15 8 7 0 12 11 9 14 2 5
6	0 11 14 5 7 12 9 2 13 6 3 8 10 1 4 15	11 0 5 14 12 7 2 9 6 13 8 3 1 10 15 4	14 5 0 11 9 2 7 12 3 8 13 6 4 15 10 1	7 12 9 2 0 11 14 5 10 1 4 15 13 6 3 8	13 6 3 8 10 1 4 15 0 11 14 5 7 12 9 2	5 14 11 0 2 9 12 7 8 3 6 13 15 4 1 10	12 7 2 9 11 0 5 14 1 10 15 4 6 13 8 3	6 13 8 3 1 10 15 4 11 0 5 14 12 7 2 9
7	0 13 7 10 11 6 12 1 14 3 9 4 5 8 2 15	13 0 10 7 6 11 1 12 3 14 4 9 8 5 15 2	7 10 0 13 12 1 11 6 9 4 14 3 2 15 5 8	11 6 12 1 0 13 7 10 5 8 2 15 14 3 9 4	14 3 9 4 5 8 2 15 0 13 7 10 11 6 12 1	10 7 13 0 1 12 6 11 4 9 3 14 15 2 8 5	6 11 1 12 13 0 10 7 8 5 15 2 3 14 4 9	3 14 4 9 8 5 15 2 13 0 10 7 6 11 1 12
8	0 14 13 3 11 5 6 8 7 9 10 4 12 2 1 15	14 0 3 13 5 11 8 6 9 7 4 10 2 12 15 1	13 3 0 14 6 8 11 5 10 4 7 9 1 15 12 2	11 5 6 8 0 14 13 3 12 2 1 15 7 9 10 4	7 9 10 4 12 2 1 15 0 14 13 3 11 5 6 8	3 13 14 0 8 6 5 11 4 10 9 7 15 1 2 12	5 11 8 6 14 0 3 13 2 12 15 1 9 7 4 10	9 7 4 10 2 12 15 1 14 0 3 13 5 11 8 6
9	0 7 13 14 11 12 6 1 14 9 3 4 5 2 8 15	7 0 14 13 12 11 1 6 9 14 4 3 2 5 15 8	13 14 0 7 6 1 11 12 3 4 14 9 8 15 5 2	11 12 6 1 0 7 13 14 5 2 8 15 14 9 3 4	14 9 3 4 5 2 8 15 0 7 13 14 11 12 6 1	14 13 7 0 1 6 12 11 4 3 9 14 15 8 2 5	12 11 1 6 7 0 14 13 2 5 15 8 9 14 4 3	9 14 4 3 2 5 15 8 7 0 14 13 12 11 1 6
10	0 11 13 6 14 5 3 8 7 12 10 1 9 2 14 15	11 0 6 13 5 14 8 3 12 7 1 10 2 9 15 14	13 6 0 11 3 8 14 5 10 1 7 12 14 15 9 2	14 5 3 8 0 11 13 6 9 2 14 15 7 12 10 1	7 12 10 1 9 2 14 15 0 11 13 6 14 5 3 8	6 13 11 0 8 3 5 14 1 10 12 7 15 14 2 9	5 14 8 3 11 0 6 13 2 9 15 14 12 7 1 10	12 7 1 10 2 9 15 14 11 0 6 13 5 14 8 3
11	0 13 14 3 7 10 9 4 11 6 5 8 12 1 2 15	13 0 3 14 10 7 4 9 6 11 8 5 1 12 15 2	14 3 0 13 9 4 7 10 5 8 11 6 2 15 12 1	7 10 9 4 0 13 14 3 12 1 2 15 11 6 5 8	11 6 5 8 12 1 2 15 0 13 14 3 7 9 9 4	3 14 13 0 4 9 10 7 8 5 6 11 15 2 1 12	10 7 4 9 13 0 3 14 1 12 15 2 6 11 8 5	6 11 8 5 1 12 15 2 13 0 3 14 10 7 4 9

Продолжение табл. 6.1.2

	0				1				2				3				4				5				6				7			
12	0	14	7	9	14	0	9	7	7	9	0	14	13	3	10	4	11	5	12	2	9	7	14	0	3	13	4	10	5	11	2	12
	13	3	10	4	3	13	4	10	10	4	13	3	0	14	7	9	6	8	1	15	4	10	3	13	14	0	9	7	8	6	15	1
	11	5	12	2	5	11	2	12	12	2	11	5	6	8	1	15	0	14	7	9	2	12	5	11	8	6	15	1	14	0	9	7
	6	8	1	15	8	6	15	1	1	15	6	8	11	5	12	2	13	3	10	4	15	1	8	6	5	11	2	12	3	13	4	10
13	0	7	14	9	7	0	9	14	14	9	0	7	13	10	3	4	11	12	5	2	9	14	7	0	10	13	4	3	12	11	2	5
	13	10	3	4	10	13	4	3	3	4	13	10	0	7	14	9	6	1	8	15	4	3	10	13	7	0	9	14	1	6	15	8
	11	12	5	2	12	11	2	5	5	2	11	12	6	1	8	15	0	7	14	9	2	5	12	11	1	6	15	8	7	0	9	14
	6	1	8	15	1	6	15	8	8	15	6	1	11	12	5	2	13	10	3	4	15	8	1	6	12	11	2	5	10	13	4	3
14	0	11	7	12	11	0	12	7	7	12	0	11	13	6	10	1	14	5	9	2	12	7	11	0	6	13	1	10	5	14	2	9
	13	6	10	1	6	13	1	10	10	1	13	6	0	11	7	12	3	8	4	15	1	10	6	13	11	0	12	7	8	3	15	4
	14	5	9	2	5	14	2	9	9	2	14	5	3	8	4	15	0	11	7	12	2	9	5	14	8	3	15	4	11	0	12	7
	3	8	4	15	8	3	15	4	4	15	3	8	14	5	9	2	13	6	10	1	15	4	8	3	5	14	2	9	6	13	1	10
15	0	13	11	6	13	0	6	11	11	6	0	13	14	3	5	8	7	10	12	1	6	11	13	0	3	14	8	5	10	7	1	12
	14	3	5	8	3	14	8	5	5	8	14	3	0	13	11	6	9	4	2	15	8	5	3	14	13	0	6	11	4	9	15	2
	7	10	12	1	10	7	1	12	12	1	7	10	9	4	2	15	0	13	11	6	1	12	10	7	4	9	15	2	13	0	6	11
	9	4	2	15	4	9	15	2	2	15	9	4	7	10	12	1	14	3	5	8	15	2	4	9	10	7	1	12	3	14	8	5
16	0	14	11	5	14	0	5	11	11	5	0	14	13	3	6	8	7	9	12	2	5	11	14	0	3	13	8	6	9	7	2	12
	13	3	6	8	3	13	8	6	6	8	13	3	0	14	11	5	10	4	1	15	8	6	3	13	14	0	5	11	4	10	15	1
	7	9	12	2	9	7	2	12	12	2	7	9	10	4	1	15	0	14	11	5	2	12	9	7	4	10	15	1	14	0	5	11
	10	4	1	15	4	10	15	1	1	15	10	4	7	9	12	2	13	3	6	8	15	1	4	10	9	7	2	12	3	13	8	6
17	0	7	11	12	7	0	12	11	11	12	0	7	13	10	6	1	14	9	5	2	12	11	7	0	10	13	1	6	9	14	2	5
	13	10	6	1	10	13	1	6	6	1	13	10	0	7	11	12	3	4	8	15	1	6	10	13	7	0	12	11	4	3	15	8
	14	9	5	2	9	14	2	5	5	2	14	9	3	4	8	15	0	7	11	12	2	5	9	14	4	3	15	8	7	0	12	11
	3	4	8	15	4	3	15	8	8	15	3	4	14	9	5	2	13	10	6	1	15	8	4	3	9	14	2	5	10	13	1	6
18	0	11	14	5	11	0	5	14	14	5	0	11	13	6	3	8	7	12	9	2	5	14	11	0	6	13	8	3	12	7	2	9
	13	6	3	8	6	13	8	3	3	8	13	6	0	11	14	5	10	1	4	15	8	3	6	13	11	0	5	14	1	10	15	4
	7	12	9	2	12	7	2	9	9	2	7	12	10	1	4	15	0	11	14	5	2	9	12	7	1	10	15	4	11	0	5	14
	10	1	4	15	1	10	15	4	4	15	10	1	7	12	9	2	13	6	3	8	15	4	1	10	12	7	2	9	6	13	8	3
19	0	13	7	10	13	0	10	7	7	10	0	13	14	3	9	4	11	6	12	1	10	7	13	0	3	14	4	9	6	11	1	12
	14	3	9	4	3	14	4	9	9	4	14	3	0	13	7	10	5	8	2	15	4	9	3	14	13	0	10	7	8	5	15	2
	11	6	12	1	6	11	1	12	12	1	11	6	5	8	2	15	0	13	7	10	1	12	6	11	8	5	15	2	13	0	10	7
	8	8	2	15	8	5	15	2	2	15	5	8	11	6	12	1	14	3	9	4	15	2	8	5	6	11	1	12	3	14	4	9
20	0	14	13	3	14	0	3	13	13	3	0	14	7	9	10	4	11	5	6	8	3	13	14	0	9	7	4	10	5	11	8	6
	7	9	10	4	9	7	4	10	10	4	7	9	0	14	13	6	12	2	1	15	4	10	9	7	14	0	3	13	2	12	15	1
	11	5	6	8	5	11	8	6	6	8	11	5	12	2	1	15	0	14	13	3	8	6	5	11	2	12	15	1	14	0	3	13
	12	2	1	15	2	12	15	1	1	15	12	2	11	5	6	8	7	9	10	4	15	1	2	12	5	11	8	6	9	7	4	10
21	0	7	13	10	7	0	10	13	13	10	0	7	14	9	3	4	11	12	6	1	10	13	7	0	9	14	4	3	12	11	1	6
	14	9	3	4	9	14	4	3	3	4	14	9	0	7	13	10	5	2	8	15	4	3	9	14	7	0	10	13	2	5	15	8
	11	12	6	1	12	11	1	6	6	1	11	12	5	2	8	15	0	7	13	10	1	6	12	11	2	5	15	8	7	0	10	13
	5	2	8	15	2	5	15	8	8	15	5	2	11	12	6	1	14	9	3	4	15	8	2	5	12	11	1	6	9	14	4	3
22	0	11	13	6	11	0	6	13	13	6	0	11	7	12	10	1	14	5	3	8	6	13	11	0	12	7	1	10	5	14	8	3
	7	12	10	1	12	7	1	10	10	1	7	12	0	11	13	6	9	2	4	15	1	10	12	7	11	0	6	13	2	9	15	4
	14	5	3	8	5	14	8	3	3	8	14	5	9	2	4	15	0	11	13	6	8	3	5	14	2	9	15	4	11	0	6	13
	9	2	4	15	2	9	15	4	4	15	9	2	14	5	3	8	7	12	10	1	15	4	2	9	5	14	8	3	12	7	1	10
23	0	13	14	3	13	0	3	14	14	3	0	13	11	6	5	8	7	10	9	4	3	14	13	0	6	11	8	5	10	7	4	9
	11	6	5	8	6	11	8	5	5	8	11	6	0	13	14	3	12	1	2	15	8	5	6	11	13	0	3	14	1	12	15	2
	7	10	9	4	10	7	4	9	9	4	7	10	12	1	2	15	0	13	14	3	4	9	10	7	1	12	15	2	13	0	3	14
	12	1	2	15	1	12	15	2	2	15	12	1	7	10	9	4	11	6	5	8	15	2	1	12	10	7	4	9	6	11	8	5

Продолжение табл. 6.1.2

	0	1	2	3	4	5	6	7
24	15 1 8 6 4 10 3 13 2 12 5 11 8 7 14 0	1 15 6 8 10 4 13 3 12 2 11 5 7 8 0 14	8 6 15 1 3 13 4 10 5 11 2 12 14 0 8 7	4 10 3 13 15 1 8 6 8 7 14 0 2 12 5 11	2 12 5 11 8 7 14 0 15 1 8 6 4 10 3 13	6 8 1 15 13 3 10 4 11 5 12 2 0 14 7 8	10 4 13 3 1 15 6 8 7 8 0 14 12 2 11 5	12 2 11 5 7 8 0 14 1 15 6 8 10 4 13 3
25	15 8 1 6 4 3 10 13 2 5 12 11 9 14 7 0	8 15 6 1 3 4 13 10 5 2 11 12 14 9 0 7	1 6 15 8 10 13 4 3 12 11 2 5 7 0 9 14	4 3 10 13 15 8 1 6 9 14 7 0 2 5 12 11	2 5 12 11 9 14 7 0 15 8 1 6 4 3 10 13	6 1 8 15 13 10 3 4 11 12 5 2 0 7 14 9	3 4 13 10 8 15 6 1 14 9 0 7 5 2 11 12	5 2 11 12 14 9 0 7 8 15 6 1 3 4 13 10
26	15 4 8 3 1 10 6 13 2 9 5 14 12 7 11 0	4 15 3 8 10 1 13 6 9 2 14 5 7 12 0 11	8 3 15 4 6 13 1 10 5 14 2 9 11 0 12 7	1 10 6 13 15 4 8 3 12 7 11 0 2 9 5 14	2 9 5 14 12 7 11 0 15 4 8 3 1 10 6 13	3 8 4 15 13 6 10 1 14 5 9 2 0 11 7 12	10 1 13 6 4 15 3 8 7 12 0 11 9 2 14 5	9 2 14 5 7 12 0 11 4 15 3 8 10 1 13 6
27	15 2 4 9 8 5 3 14 1 12 10 7 6 11 13 0	2 15 9 4 5 8 14 3 12 1 7 10 11 6 0 13	4 9 15 2 3 14 8 5 10 7 1 12 13 0 6 11	8 5 3 14 15 2 4 9 6 11 13 0 1 12 10 7	1 12 10 7 6 11 13 0 15 2 4 9 8 5 3 14	9 4 2 15 14 3 5 8 7 10 12 1 0 13 11 6	5 8 14 3 2 15 9 4 11 6 0 13 12 1 7 10	12 1 7 10 11 6 0 13 2 15 9 4 5 8 14 3
28	15 1 4 10 8 6 3 13 2 12 9 7 5 11 14 0	1 15 10 4 6 8 13 3 12 2 7 9 11 5 0 14	4 10 15 1 3 13 8 6 9 7 2 12 14 0 5 11	8 6 3 13 15 1 4 10 5 11 14 0 2 12 9 7	2 12 9 7 5 11 14 0 15 1 4 10 8 6 3 13	10 4 1 15 13 3 6 8 7 9 12 2 0 14 11 5	6 8 13 3 1 15 10 4 11 5 0 14 12 2 7 9	12 2 7 9 11 5 2 14 1 15 10 4 6 8 13 3
29	15 8 4 3 1 6 10 13 2 5 6 14 12 11 7 0	8 15 3 4 6 1 13 10 5 2 14 9 11 12 0 7	4 3 15 8 10 13 1 6 9 14 2 5 7 0 12 11	1 6 10 13 15 8 4 3 12 11 7 0 2 5 9 14	2 5 9 14 12 11 7 0 15 8 4 3 1 6 10 13	3 4 8 15 13 10 6 1 14 9 5 2 0 7 11 12	6 1 13 10 8 15 3 4 11 12 0 7 5 2 14 9	2 2 14 9 11 12 0 7 8 15 3 4 6 1 13 10
30	15 4 1 10 8 3 6 13 2 9 12 7 5 14 11 0	4 15 10 1 3 8 13 6 9 2 7 12 14 5 0 11	1 10 15 4 6 13 8 3 12 7 2 9 11 0 5 14	8 3 6 13 15 4 1 10 5 14 11 0 2 9 12 7	2 9 12 7 5 14 11 0 15 4 1 10 8 3 6 13	10 1 4 15 13 6 3 8 7 12 9 2 0 11 14 5	3 8 13 6 4 15 10 1 14 5 0 11 9 2 7 12	9 2 7 12 14 5 0 11 4 15 10 1 3 8 13 4
31	15 2 8 5 4 9 3 14 1 12 6 11 10 7 13 0	2 15 5 8 9 4 14 3 12 1 11 6 7 10 0 13	8 5 15 2 3 14 4 9 6 11 1 12 13 0 10 7	4 9 3 14 15 2 8 5 10 7 13 0 1 12 6 11	1 12 6 11 10 7 13 0 15 2 8 5 4 9 3 14	5 8 2 15 14 3 9 4 11 6 12 1 0 13 7 10	9 4 14 3 2 15 5 8 7 10 0 13 12 1 11 6	12 1 11 6 7 10 0 13 2 15 5 8 9 4 14 3
32	15 1 2 12 4 10 9 7 8 6 5 11 3 13 14 0	1 15 12 2 10 4 7 9 6 8 11 5 13 3 0 14	2 12 15 1 9 7 4 10 5 11 8 6 14 0 3 13	4 10 9 7 15 1 2 12 3 13 14 0 8 6 5 11	8 6 5 11 3 13 14 0 15 1 2 12 4 10 9 7	12 2 1 15 7 9 10 4 11 5 6 8 0 14 13 3	10 4 7 9 1 15 12 2 13 3 0 14 6 8 11 5	6 8 11 5 13 3 0 14 1 15 12 2 10 4 7 9+
33	15 8 2 5 4 3 9 14 1 6 12 11 10 13 7 0	8 15 5 2 3 4 14 9 6 1 11 12 13 10 0 7	2 5 15 8 9 14 4 3 12 11 1 6 7 0 10 13	4 3 9 14 15 8 2 5 10 13 7 0 1 6 12 11	1 6 12 11 10 13 7 0 15 8 2 5 4 3 9 14	5 2 8 15 14 9 3 4 11 12 6 1 0 7 13 10	3 4 14 9 8 15 5 2 13 10 0 7 6 1 11 12	6 1 11 12 13 10 0 7 8 15 5 2 3 4 14 9
34	15 4 2 9 1 10 12 7 8 3 5 14 6 13 11 0	4 15 9 2 10 1 7 12 3 8 14 5 13 6 0 11	2 9 15 4 12 7 1 10 5 14 8 3 11 0 6 13	1 10 12 7 15 4 2 9 6 13 11 0 8 3 5 14	8 3 5 14 6 13 11 0 15 4 2 9 1 10 12 7	9 2 4 15 7 12 10 1 14 5 3 8 0 11 13 6	10 1 7 12 4 15 9 2 13 6 0 11 3 8 14 5	3 8 14 5 13 6 0 11 4 15 9 2 10 1 7 12
35	15 2 1 12 8 5 6 11 4 9 10 7 3 14 13 0	2 15 12 1 5 8 11 6 9 4 7 10 14 3 0 13	1 12 15 2 6 11 8 5 10 7 4 9 13 0 3 14	8 5 6 11 15 2 1 12 3 14 13 0 4 9 10 7	4 9 10 7 3 14 13 0 15 2 1 12 8 5 6 11	12 1 2 15 11 6 5 8 7 10 9 4 0 13 14 3	5 8 11 6 2 15 12 1 14 3 0 13 9 4 7 10	9 4 7 10 14 3 0 13 2 15 12 1 5 8 11 6

Окончание табл. 6.1.2

	0	1	2	3	4	5	6	7
36	15 1 8 6 2 12 5 11 4 10 3 13 9 7 14 0	1 15 6 8 12 2 11 5 10 4 13 3 7 9 0 14	8 6 15 1 5 11 2 12 3 13 4 10 14 0 9 7	2 12 5 11 15 1 8 6 9 7 14 0 4 10 3 13	4 10 3 13 9 7 14 0 15 1 8 6 2 12 5 11	6 8 1 15 11 5 12 2 13 3 10 4 0 14 7 9	12 2 11 5 1 15 6 8 7 9 0 14 10 4 13 3	10 4 13 3 7 9 0 14 1 12 6 8 12 2 11 5
37	15 8 1 6 2 5 12 11 4 3 10 13 9 14 7 0	8 15 6 1 5 2 11 12 3 4 13 10 14 9 0 7	1 6 15 8 12 11 2 5 10 13 4 3 7 0 9 14	2 5 12 11 15 8 1 6 9 14 7 0 4 3 10 13	4 3 10 13 9 14 7 0 15 8 1 6 2 5 12 11	6 1 8 15 11 12 5 2 13 10 3 4 0 7 14 9	5 2 11 12 8 15 6 1 14 9 0 7 3 4 13 10	3 4 13 10 14 9 0 7 8 15 6 1 5 2 11 12
38	15 4 8 3 2 9 5 14 1 10 6 13 12 7 11 0	4 15 3 8 9 2 14 5 10 1 13 6 7 12 0 11	8 3 15 4 5 14 2 9 6 13 1 10 11 0 12 7	2 9 5 14 15 4 8 3 12 7 11 0 1 10 6 13	1 10 6 13 12 7 11 0 15 4 8 3 2 9 5 14	3 8 4 15 14 5 9 2 13 6 10 1 0 11 7 12	9 2 14 5 4 15 3 8 7 12 0 11 10 1 13 6	10 1 13 6 7 12 0 11 4 15 3 8 9 2 14 5
39	15 2 4 9 1 12 10 7 8 5 3 14 6 11 13 0	2 15 9 4 12 1 7 10 5 8 14 3 11 6 0 13	4 9 15 2 10 7 1 12 3 14 8 5 13 0 6 11	1 12 10 7 15 2 4 9 6 11 13 0 8 5 3 14	8 5 3 14 6 11 13 0 15 2 4 9 1 12 10 7	9 4 2 15 7 10 12 1 14 3 5 8 0 13 11 6	12 1 7 10 2 15 9 4 11 6 0 13 5 8 14 3	5 8 14 3 11 6 0 13 2 15 9 4 12 1 7 10
40	15 1 4 10 2 12 9 7 8 6 3 13 5 11 14 0	1 15 10 4 12 2 7 9 6 8 13 3 11 5 0 14	4 10 15 1 9 7 2 12 3 13 8 6 14 0 5 11	2 12 9 7 15 1 4 10 5 11 14 0 8 6 3 13	8 6 3 13 5 11 14 0 15 1 4 10 2 12 9 7	10 4 1 15 7 9 12 2 13 3 6 8 0 14 11 5	12 2 7 9 1 15 10 4 11 5 0 14 6 8 13 3	6 8 13 3 11 5 0 14 1 15 10 4 12 2 7 9
41	15 8 4 3 2 5 9 14 1 6 10 13 12 11 7 0	8 15 3 4 5 2 14 9 6 1 13 10 11 12 0 7	4 3 15 8 9 14 2 5 10 13 1 6 7 0 12 11	2 5 9 14 15 8 4 3 12 11 7 0 1 6 10 13	1 6 10 13 12 11 7 0 15 8 4 3 2 5 9 14	3 4 8 15 14 9 5 2 13 10 6 1 0 7 11 12	5 2 14 9 8 15 3 4 11 12 0 7 6 1 13 10	6 1 13 10 11 12 0 7 8 15 3 4 5 2 14 9
42	15 4 1 10 2 9 12 7 8 3 6 13 5 14 11 0	4 15 10 1 9 2 7 12 3 8 13 6 14 5 0 11	1 10 15 4 12 7 2 9 6 13 8 3 11 0 5 14	2 9 12 7 15 4 1 10 5 14 11 0 8 3 6 13	8 3 6 13 5 14 11 0 15 4 1 10 2 9 12 7	10 1 4 15 7 12 9 2 13 6 3 8 0 11 14 5	9 2 7 12 4 15 10 1 14 5 0 11 3 8 13 6	3 8 13 6 14 5 0 11 4 15 10 1 9 2 7 12
43	15 2 8 5 1 12 6 11 4 9 3 14 10 7 13 0	2 15 5 8 12 1 11 6 9 4 14 3 7 10 0 13	8 5 15 2 6 11 1 12 3 14 4 9 13 0 10 7	1 12 6 11 15 2 8 5 10 7 13 0 4 9 3 14	4 9 3 14 10 7 13 0 15 2 8 5 1 12 6 11	5 8 2 15 11 6 12 1 14 3 9 4 0 13 7 10	12 1 11 6 2 15 5 8 7 10 0 13 9 4 14 3	9 4 14 3 7 10 0 13 2 15 5 8 12 1 11 6
44	15 1 2 12 8 6 5 11 4 10 9 7 3 13 14 0	1 15 12 2 6 8 11 5 10 4 7 9 13 3 0 14	2 12 15 1 5 11 8 6 9 7 4 10 14 0 3 13	8 6 5 11 15 1 2 12 3 13 14 0 4 10 9 7	4 10 9 7 3 13 14 0 15 1 2 12 8 6 5 11	12 2 1 15 11 5 6 8 7 9 10 4 0 14 13 3	6 8 11 5 1 15 12 2 13 3 0 14 10 4 7 9	10 4 7 9 13 3 0 14 1 15 12 2 6 8 11 5
45	15 8 2 5 1 6 12 11 4 3 9 14 10 13 7 0	8 15 5 2 6 1 11 12 3 4 14 9 13 10 0 7	2 5 15 8 12 11 1 6 9 14 4 3 7 0 10 13	1 6 12 11 15 8 2 5 10 13 7 0 4 3 9 14	4 3 9 14 10 13 7 0 15 8 2 5 1 6 12 11	5 2 8 15 11 12 6 1 14 9 3 4 0 7 13 10	6 1 11 12 8 15 5 2 13 10 0 7 3 4 14 9	3 4 14 9 13 10 0 7 8 15 5 2 6 1 11 12
46	15 4 2 9 8 3 5 14 1 10 12 7 6 13 11 0	4 15 9 2 3 8 14 5 10 1 7 12 13 6 0 11	2 9 15 4 5 14 8 3 12 7 1 10 11 0 6 13	8 3 5 14 15 4 2 9 6 13 11 0 1 10 12 7	1 10 12 7 6 13 11 0 15 4 2 9 8 3 5 14	9 2 4 15 14 5 3 8 7 12 10 1 0 11 13 6	3 8 14 5 4 15 9 2 13 6 0 11 10 1 7 12	10 1 7 12 13 6 0 11 4 15 9 2 3 8 14 5
47	15 2 1 12 4 9 10 7 8 5 6 11 3 14 13 0	2 15 12 1 9 4 7 10 5 8 11 6 14 3 0 13	1 12 15 2 10 7 4 9 6 11 8 5 13 0 3 14	4 9 10 7 15 2 1 12 3 14 13 0 8 5 6 11	8 5 6 11 3 14 13 0 15 2 1 12 4 9 10 7	12 1 2 15 7 10 9 4 11 6 5 8 0 13 14 3	9 4 7 10 2 15 12 1 14 3 0 13 5 8 11 6	5 8 11 6 14 3 0 13 2 15 12 1 9 4 7 10

Кодовые расстояния между комбинациями сигналов разрядов контрольной части систематического кода, исправляющего одиночные и двойные ошибки для всех типов кодов основания $n = 2^4$, одинаковы и на примере четырех кодов с координатами (0, 0), (1, 0), (2, 0), (3, 0) таблицы приведены на рис. 6.8.

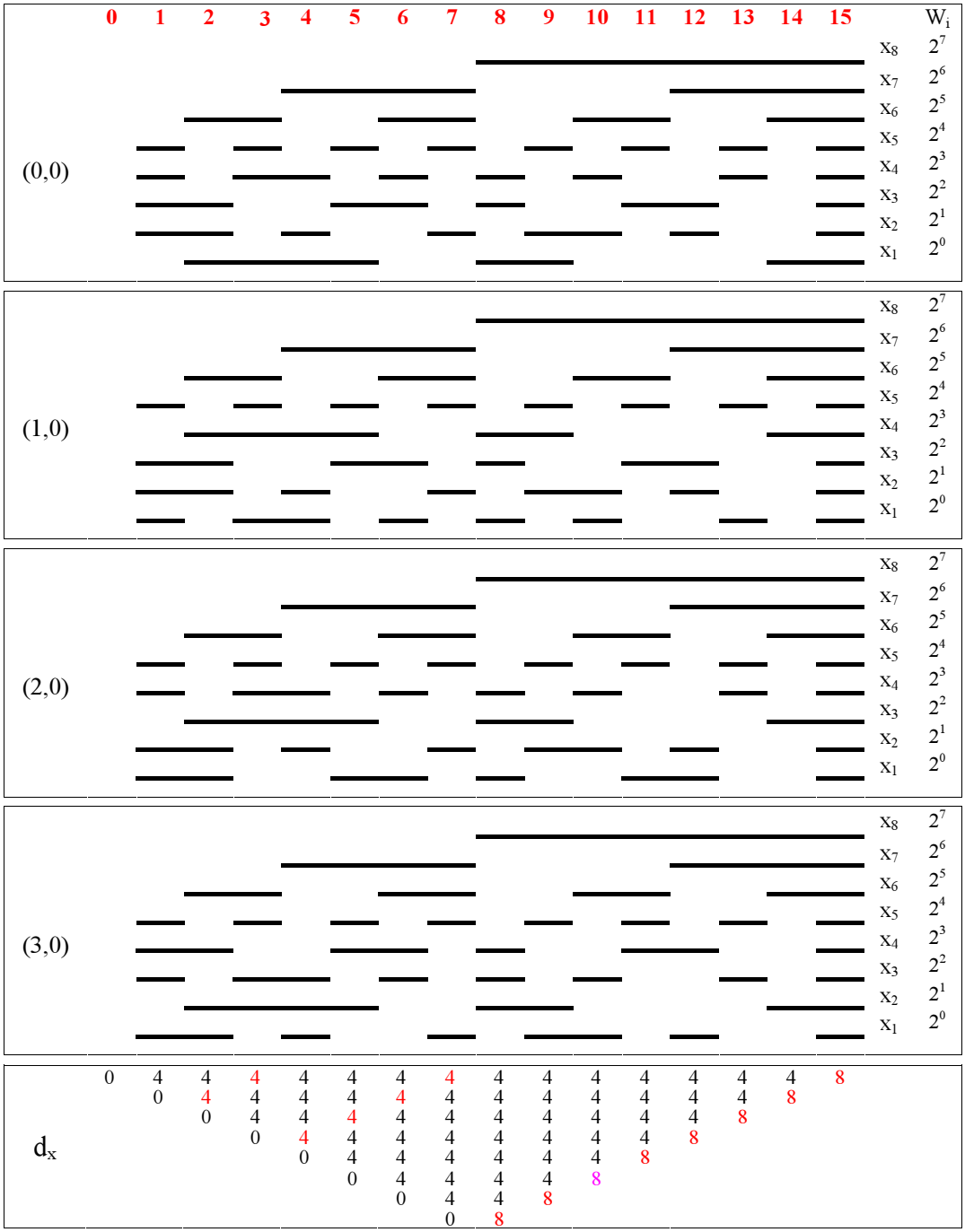


Рис. 6.8

Сложение этих кодовых расстояний с кодовыми расстояниями комбинаций сигналов разрядов основного двоичного кода, где минимальное кодовое расстояние равно $d_{мин} = 1$, позволило нам синтезировать код с минимальным кодовым расстоянием $d_{мин} = 5$, что гарантирует возможность исправления всех одиночных и двойных ошибок разрядов систематического кода.

Коды контрольных разрядов, подобные рис. 6.8 и имеющие минимальные кодовые расстояния $d_{мин} = 4$ во всех ячейках, кроме расположенных на диагоналях этих таблиц, будем называть «образующими» контрольных разрядов систематического кода определенного основания системы счисления. Для основания $n = 2^4$ число таких «образующих» кодов – 384.

Для того чтобы иметь возможность исправлять все одиночные, двойные и тройные ошибки, необходимо иметь в систематическом коде минимальное кодовое расстояние $d_{мин} = 7$. Сложение кодовых расстояний совершенного систематического кода, исправляющего все одиночные ошибки ($d_{мин} = 3$), с «образующим» кодом ($d_{мин} = 4$) позволяет выполнить эту задачу синтеза. Тогда число систематических кодов, исправляющих все одиночные, двойные и тройные ошибки, будет равно произведению этих кодов $S_{1,2,3} = S_{dмин\ [7]} = 192 \times 384 = 73728$. Пример такого кода, в основу которого взяты коды из табл. 4.2.2 и 6.1.2 с одноименными координатами (0,0), изображен на рис. 6.9.

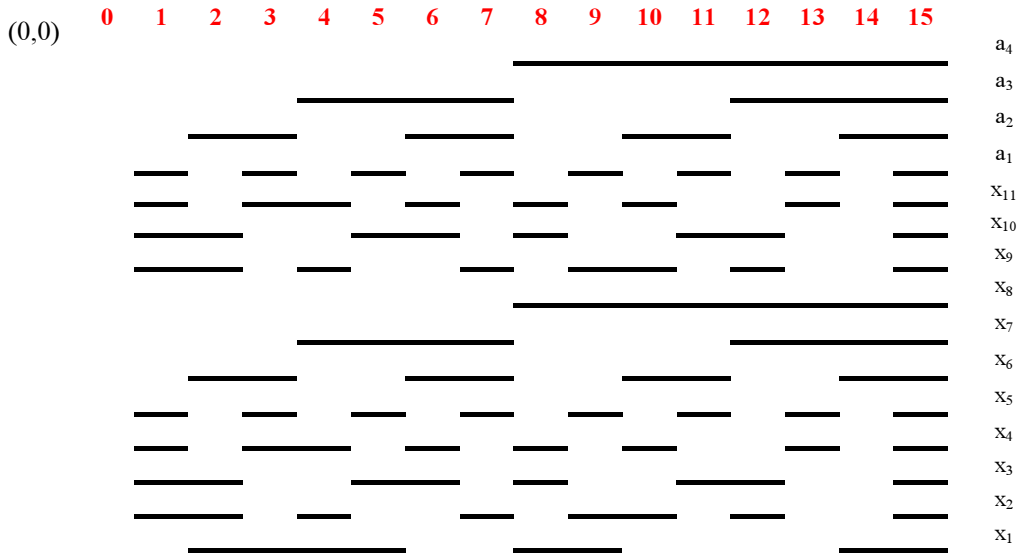


Рис. 6.9

Здесь между сигналами разрядов систематического кода существуют следующие соотношения: $a_1 = x_5$, $a_2 = x_6$, $a_3 = x_7$, $a_4 = x_8$.

Для исправления всех одиночных, двойных, тройных и четверных ошибок необходимо иметь в систематическом коде минимальное кодовое расстояние $d_{мин} = 9$. Следовательно, сложение кодовых расстояний совершенного систематического кода, исправляющего все одиночные и двойные ошибки ($d_{мин} = 5$), с «образующим» кодом ($d_{мин} = 4$) позволяет это выполнить. Так же, как и выше, число систематических кодов, исправляющих все одиночные, двойные и тройные ошибки, будет равно произведению этих кодов $S_{1,2,3,4} = S_{dмин\ [9]} = 384 \times 384 = 147456$, а пример такого кода при таких же условиях его формирования приведен на рис. 6.10.

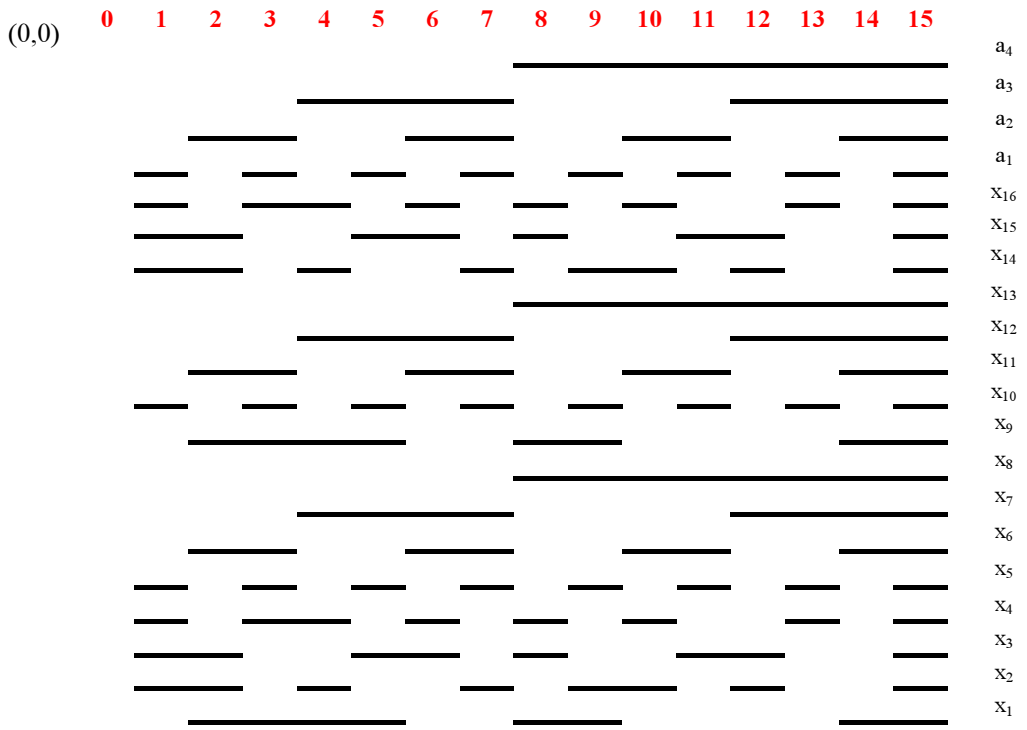


Рис. 6.10

Соотношения между разрядами восьми групп сигналов этого систематического кода следующие: $a_1 = x_{13} = x_5$; $a_2 = x_{14} = x_6$; $a_3 = x_{15} = x_7$; $a_4 = x_{16} = x_8$; $x_4 = x_{12}$; $x_3 = x_{11}$; $x_2 = x_{10}$; $x_1 = x_9$.

В первых четырех группах сигналов, в каждой из которых содержится один информационный и два контрольных разряда, что характеризует совершенный мажоритарный код Дж. Неймана, имеется возможность автономно от других разрядов весьма просто исправлять все одиночные ошибки и определенное число двойных, тройных и четверных ошибок в сигналах этих групп. Данное обстоятельство может быть использовано при реализации принципиальных схем исправления ошибок в этом типе кодов.

Определим общее число всех исправляемых ошибок, когда каждый информационный разряд систематического кода основания $n = 2^4$ представлен в мажоритарном коде Дж. Неймана. Для этого обозначим контрольные разряды для каждого информационного разряда буквами x, y:

a_1	a_2	a_3	a_4
x_1	x_2	x_3	x_4
y_1	y_2	y_3	y_4

.

Общее число всех типов возможных ошибок определяется числом сочетаний из 12 по 1, 2, 3 и 4. Это соответственно значения: 12, 66, 220, 495. Очевидно, что число всех исправляемых одиночных ошибок – 12, все остальные типы

ошибок исправляются не полностью, поскольку в каждой из групп может быть исправлена только одна ошибка.

Число исправляемых двойных, тройных и четверных ошибок следует из их графического представления на рис. 6. 11–6.13, где цветом выделен сигнал соответствующего разряда в каждой из групп, который участвует в соответствующем сочетании за исключением сигналов этой группы.

Тогда число исправляемых двойных ошибок – 54, тройных – 76, четверных – 81.

a_1	a_2	a_3	a_4
	x_2	x_3	x_4
	y_2	y_3	y_4

	a_2	a_3	a_4
x_1		x_3	x_4
y_1		y_3	y_4

		a_3	a_4
x_1	x_2		x_4
y_1	y_2		y_4

			a_4
x_1	x_2	x_3	
y_1	y_2	y_3	

x_1	x_2	x_3	x_4
	y_2	y_3	y_4

	x_2	x_3	x_4
y_1		y_3	y_4

		x_3	x_4
y_1	y_2		y_4

			x_4
y_1	y_2	y_3	

y_1	y_2	y_3	y_4

	y_2	y_3	y_4

		y_3	y_4

Рис. 6.11

a_1	a_2	a_3	a_4
		x_3	x_4
		y_3	y_4

	a_2	a_3	a_4
x_1			x_4
y_1			y_4

		a_3	a_4
x_1	x_2		
y_1	y_2		

			a_4
x_1	x_2	x_3	
	y_2	y_3	

a_1		a_3	a_4
	x_2	x_3	x_4
		y_3	y_4

	a_2		a_4
x_1		x_3	x_4
y_1			y_4

		a_3	
x_1	x_2		x_4
y_1	y_2		

			a_4
	x_2	x_3	
y_1	y_2	y_3	

a_1		a_3	a_4
		x_3	x_4
	y_2	y_3	y_4

	a_2		a_4
x_1			x_4
y_1		y_3	y_4

		a_3	
x_1	x_2		
y_1	y_2		y_4

x_1	x_2	x_3	x_4
		y_3	y_4

	x_2	x_3	x_4
y_1			y_4

		x_3	x_4
y_1	y_2		

			x_4
y_1	y_2	y_3	

x_1		x_3	x_4
	y_2	y_3	y_4

	x_2		x_4
y_1		x_3	y_4

		x_3	
y_1	y_2		y_4

y_1	y_2	y_3	y_4

	y_2	y_3	y_4

Рис. 6.12

a_1	a_2	a_3	a_4
			x_4
			y_4

a_1	a_2		a_4
		x_3	x_4
			y_4

a_1	a_2		a_4
			x_4
		y_3	y_4

	a_2	a_3	a_4
x_1			
y_1			

	a_2	a_3	
x_1			x_4
y_1			

	a_2	a_3	
x_1			
y_1			y_4

		a_3	a_4
x_1	x_2		
	y_2		

		a_3	a_4
	x_2		
y_1	y_2		

			a_4
x_1	x_2	x_3	
		y_3	

			a_4
x_1		x_3	
	x_2	y_3	

a_1		a_3	a_4
	x_2		x_4
		y_4	

a_1			a_4
	x_2	x_3	x_4
			y_4

a_1			a_4
	x_2		x_4
		y_3	y_4

	a_2		a_4
x_1		x_3	
	y_2		y_4

	a_2		
x_1		x_3	x_4
			y_4

	a_2		
x_1		x_3	
y_1			y_4

		a_3	
x_1	x_2		x_4
	y_2		y_4

		a_3	
x_1	x_2		
	y_2		y_4

			a_4
	x_2	x_3	
			y_4

			a_4
	x_2		
			y_4

a_1		a_3	a_4
			x_4
	y_2		y_4

a_1			a_4
		x_3	x_4
	y_2		y_4

a_1			a_4
			x_4
	y_2	y_3	y_4

	a_2		a_4
			x_4
x_1			x_4
		y_3	y_4

	a_2		a_4
		a_3	
y_1		y_3	y_4

		a_3	
x_1	x_2		
	y_2		y_4

		a_3	
	x_2		
y_1	y_2		y_4

x_1	x_2	x_3	x_4
			y_4

x_1	x_2		x_4
		y_3	y_4

	x_2	x_3	x_4
y_1			y_4

		x_3	x_4
y_1	y_2		

x_1	x_2		x_4
y_1	y_2	y_3	

x_1		x_3	x_4
	y_2		y_4

x_1			x_4
	y_2	y_3	y_4

	x_2		x_4
y_1		y_3	y_4

		x_3	
y_1	y_2		y_4

y_1	y_2	y_3	y_4

Рис. 6.13

Дальнейший синтез систематических кодов, исправляющих еще большее количество подобных ошибок, не вызывает каких-либо трудностей. Он сводится к простой операции сложения соответствующих кодовых расстояний: $d_{мин\,[11]} = d_{мин\,[7]} + d_{мин\,[4]}$, $d_{мин\,[13]} = d_{мин\,[9]} + d_{мин\,[4]}$, $d_{мин\,[15]} = d_{мин\,[11]} + d_{мин\,[4]}$ и т.д.

Так же просто определяется число этих кодов: $S_{dмин\,[11]} = S_{dмин\,[7]}\, S_{dмин\,[4]}$, $S_{dмин\,[13]} = S_{dмин\,[9]}\, S_{dмин\,[4]}$, $S_{dмин\,[15]} = S_{dмин\,[1]}\, S_{dмин\,[4]}$ и т.д.

6.2. Синтез кодов с информационной частью в основном двоичном коде основания 2^{11}, исправляющих все группы ошибок

Начнем рассмотрение синтеза систематического кода с информационной частью основания $n = 2^{11}$ с более простого примера синтеза квазисовершенного кода основания $n = 2^5$, исправляющего одиночные и двойные ошибки.

Для основания $n = 2^5$ «образующий код», где минимальное расстояние между кодовыми комбинациями сигналов разрядов $d_{мин} = 4$, формируется из контрольной части совершенного кода ($k = 4$, $n = 2^{11}$), исправляющего все одиночные ошибки.

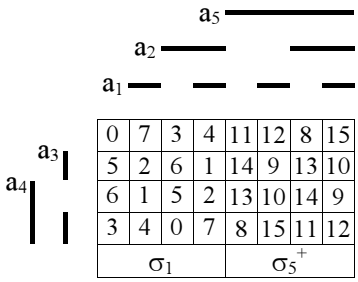


Рис. 6.14

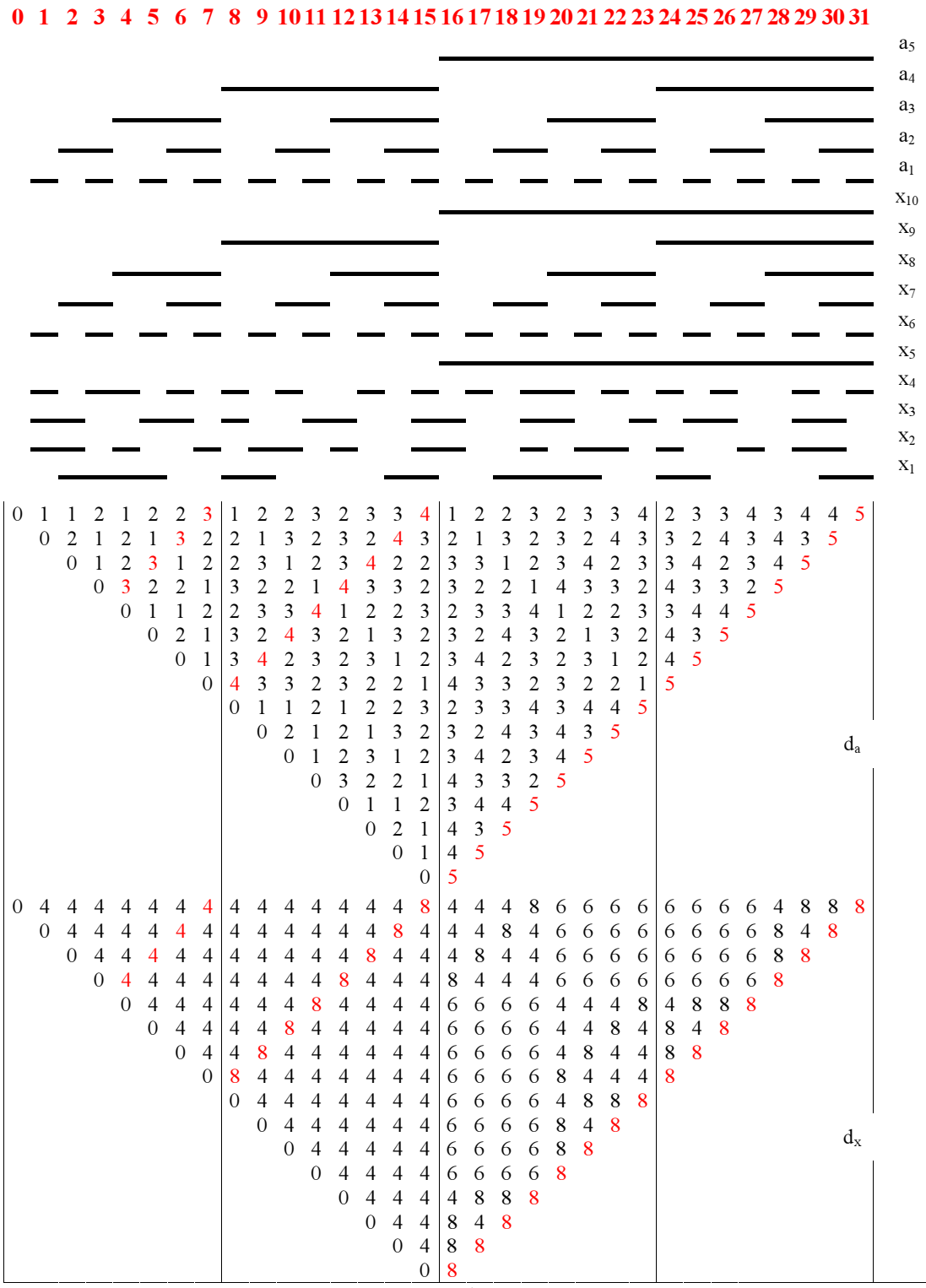
Поскольку число таких совершенных кодов большое, то первоначально ограничимся одним из них. Пусть соотношения между информационными и контрольными разрядами этого совершенного кода определяются кодом табл. 4.2.2 с координатами 0,0. При этом связь между информационными и контрольными цифрами кода будет соответствовать рис. 6.14, где в ячейках пространства информационной части квазисовершенного кода ($k = 4$, $i = 5$), исправляющего все одиночные ошибки, записаны цифры его контрольной части.

Тогда «образующий код» основания $n = 2^5$ будет включать контрольные разряды $x_1 \div x_{10}$ (рис. 6.15, *а*) систематического кода, исправляющего все одиночные и двойные ошибки. На этом рисунке также приведены кодовые расстояния в информационной $D(a_1 \div a_5)$ и контрольной $D(x_1 \div x_{10})$ частях кода, где их арифметическая сумма определяет кодовое расстояние $D(a_1 \div a_5, x_1 \div x_{10})$ всего кода (рис. 6.15, *б*).

Как показано выше, дальнейший синтез совершенных систематических кодов, исправляющих еще большее количество подобных ошибок, если известен «образующий», не вызывает каких-либо трудностей и сводится к простой операции сложения соответствующих кодовых расстояний: $d_{мин\,[7]} = d_{мин\,[3]} + d_{мин\,[4]}$, $d_{мин\,[9]} = d_{мин\,[5]} + d_{мин\,[4]}$, $d_{мин\,[11]} = d_{мин\,[7]} + d_{мин\,[4]}$ и т.д.

Это в равной степени относится как к совершенным, так и к квазисовершенным кодам. Поэтому перед нами стоит задача перейти от синтеза квазисовершенного кода основания $n = 2^5$ к синтезу совершенных кодов основания $n = 2^{11}$, исправляющих все группы ошибок.

Для этого необходимо синтезировать «образующий код» основания $n = 2^{11}$ при известности подобного кода основания $n = 2^5$.



а)

Рис. 6.15 (начало)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	5	5	6	5	6	6	7	5	6	6	7	6	7	7	12	5	6	6	11	8	9	9	10	8	9	9	10	7	12	12	13	
	0	6	5	6	5	7	6	6	5	7	6	7	6	12	7	6	5	11	6	9	8	10	9	9	8	10	9	12	7	13		
		0	5	6	6	7	5	7	6	6	6	7	7	6	6	6	11	6	6	9	10	8	9	9	10	8	9	12	13			
			0	7	6	6	6	6	5	6	5	12	7	7	6	7	6	9	9	10	9	9	8	10	9	9	8	13				
				0	5	5	5	7	6	7	12	5	6	6	7	8	9	10	10	5	6	6	11	7	12	12	13					
					0	6	6	6	7	12	7	6	5	7	6	9	8	8	9	6	5	11	6	12	7	13						
						0	5	7	12	6	7	6	7	5	6	9	10	9	9	6	11	5	6	12	13							
							0	12	7	7	6	7	6	6	5	10	9	8	8	11	6	6	5	13								
								0	5	5	6	5	6	6	7	8	9	9	10	7	12	12	13									
									0	6	5	6	5	7	6	9	8	10	9	9	7	13										
										0	5	6	7	5	6	9	10	8	9	9	12											
											0	7	6	6	5	10	9	9	8	13												
												0	5	5	6	7	12	12	13													
													0	6	5	12	7	13														
														0	5	12	12															
															0	13																

$d_{a,x}$

б)

Рис. 6.15 (продолжение)

«Образующий» код основания $n = 2^5$ является частью аналогичного кода основания $n = 2^{11}$ и может быть представлен в пространстве координат его информационной части разрядов $a_1 - a_5$ (рис. 6.16а) либо в координатах только одного разряда a_5. В последнем случае это – две ячейки пространства, где содержание первой ячейки обозначено σ_1, второй – σ_5^+.

Это подпространство является частью пространства координат $a_5 - a_{11}$ (рис. 6.16б), которое было синтезировано в гл. 3.

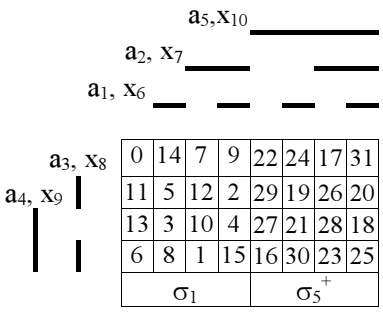


Рис. 6.16а

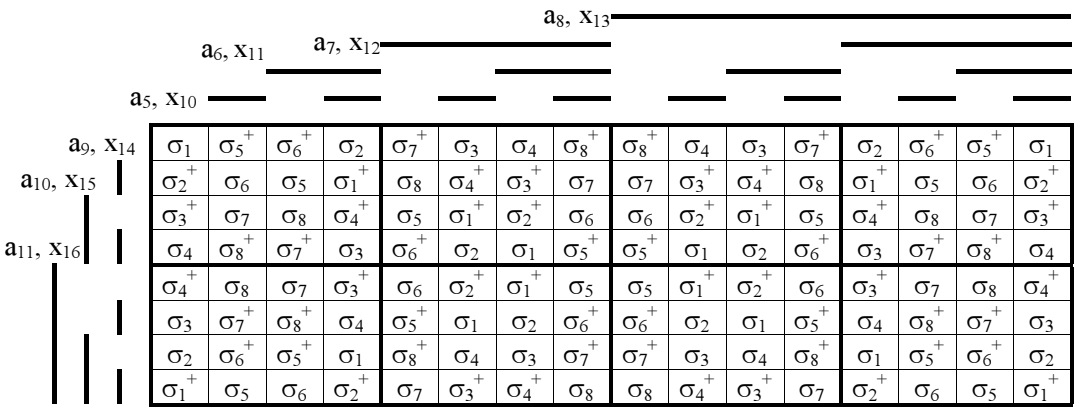


Рис. 6.16б

Содержимое ячеек подпространств $\sigma_1 - \sigma_8$ это – цифровые (весовые) значения контрольных разрядов $x_1 - x_5$ «образующего кода», как и аналогичное содержимое подпространств $\sigma_1^+ - \sigma_8^+$, переходят одно в другое соответствующими мысленными поворотами относительно осей симметрии координат $a_1 - a_5$.

Если положение координат для σ_1 и σ_1^+ принять исходными, что обозначается как ,

$$\sigma_1(a_1, a_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 0 & 14 & 7 & 9 \\ \hline 11 & 5 & 12 & 2 \\ \hline 13 & 3 & 10 & 4 \\ \hline 6 & 8 & 1 & 15 \\ \hline \end{array}, \quad \sigma_1^+(a_1, a_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 16 & 30 & 23 & 25 \\ \hline 27 & 21 & 28 & 18 \\ \hline 29 & 19 & 26 & 20 \\ \hline 22 & 24 & 17 & 31 \\ \hline \end{array},$$

то мысленные повороты относительно осей симметрии пространства координат $a_1 - a_4$ определяются для $\sigma_2 - \sigma_5$ и $\sigma_2^+ - \sigma_5^+$ следующими зависимостями:

$$\sigma_2(\underline{a}_1, a_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 5 & 11 & 2 & 12 \\ \hline 14 & 0 & 9 & 7 \\ \hline 8 & 6 & 15 & 1 \\ \hline 3 & 13 & 4 & 10 \\ \hline \end{array} \quad \sigma_2^+(\underline{a}_1, a_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 21 & 27 & 18 & 28 \\ \hline 30 & 16 & 25 & 23 \\ \hline 24 & 22 & 31 & 17 \\ \hline 19 & 29 & 20 & 26 \\ \hline \end{array}$$

$$\sigma_3(a_1, \underline{a}_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 12 & 2 & 11 & 5 \\ \hline 7 & 9 & 0 & 14 \\ \hline 1 & 15 & 6 & 8 \\ \hline 10 & 4 & 13 & 3 \\ \hline \end{array} \quad \sigma_3^+(a_1, \underline{a}_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 28 & 18 & 27 & 21 \\ \hline 23 & 25 & 16 & 30 \\ \hline 17 & 31 & 22 & 24 \\ \hline 26 & 20 & 29 & 19 \\ \hline \end{array}$$

$$\sigma_4(\underline{a}_1, \underline{a}_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 9 & 7 & 14 & 0 \\ \hline 2 & 12 & 5 & 11 \\ \hline 4 & 10 & 3 & 13 \\ \hline 15 & 1 & 8 & 6 \\ \hline \end{array} \quad \sigma_4^+(\underline{a}_1, \underline{a}_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 25 & 23 & 30 & 16 \\ \hline 18 & 28 & 21 & 27 \\ \hline 20 & 26 & 19 & 29 \\ \hline 31 & 17 & 24 & 22 \\ \hline \end{array},$$

$$\sigma_5(a_1, a_2, \underline{a}_3, \underline{a}_4) = \begin{array}{|c|c|c|c|}\hline 6 & 8 & 1 & 15 \\ \hline 13 & 3 & 10 & 4 \\ \hline 11 & 5 & 12 & 2 \\ \hline 0 & 14 & 7 & 9 \\ \hline \end{array} \quad \sigma_5^+(a_1, a_2, \underline{a}_3, \underline{a}_4) = \begin{array}{|c|c|c|c|}\hline 22 & 24 & 17 & 31 \\ \hline 29 & 19 & 26 & 20 \\ \hline 27 & 21 & 28 & 18 \\ \hline 16 & 30 & 23 & 25 \\ \hline \end{array},$$

$$\sigma_6(a_1, \underline{a}_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 7 & 9 & 0 & 14 \\ \hline 12 & 2 & 11 & 5 \\ \hline 10 & 4 & 13 & 3 \\ \hline 1 & 15 & 6 & 8 \\ \hline \end{array}, \quad \sigma_6^+(a_1, \underline{a}_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 23 & 25 & 16 & 30 \\ \hline 28 & 18 & 27 & 21 \\ \hline 26 & 20 & 29 & 19 \\ \hline 17 & 31 & 22 & 24 \\ \hline \end{array},$$

$$\sigma_7(\underline{a}_1, a_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 14 & 0 & 9 & 7 \\ \hline 5 & 11 & 2 & 12 \\ \hline 3 & 13 & 4 & 10 \\ \hline 8 & 6 & 15 & 1 \\ \hline \end{array}, \quad \sigma_7^+(\underline{a}_1, a_2, a_3, a_4) = \begin{array}{|c|c|c|c|}\hline 30 & 16 & 25 & 23 \\ \hline 21 & 27 & 18 & 28 \\ \hline 19 & 29 & 20 & 26 \\ \hline 24 & 22 & 31 & 17 \\ \hline \end{array},$$

$$\sigma_8(a_1, a_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 11 & 5 & 12 & 2 \\ \hline 0 & 14 & 7 & 9 \\ \hline 6 & 8 & 1 & 15 \\ \hline 13 & 3 & 10 & 4 \\ \hline \end{array}, \quad \sigma_8^+(a_1, a_2, \underline{a}_3, a_4) = \begin{array}{|c|c|c|c|}\hline 27 & 21 & 28 & 18 \\ \hline 16 & 30 & 23 & 25 \\ \hline 22 & 24 & 17 & 31 \\ \hline 29 & 19 & 26 & 20 \\ \hline \end{array}.$$

Эти зависимости определяют только первую часть разрядов ($x_1 - x_5$) «образующего кода» основания $n = 2^{11}$, а остальные его разряды ($x_6 - x_{16}$) совпадают соответственно с информационными разрядами ($a_1 - a_{11}$) систематического кода.

Поскольку «образующий код» полностью совпадает с контрольной частью систематического кода, исправляющего все одиночные и двойные ошибки, то задача синтеза такого кода основания $n = 2^{11}$ нами выполнена и нет каких-либо

препятствий для построения кодов этого основания, исправляющих любые группы из последовательных ошибок.

Аналогичные этапы синтеза могут быть выполнены для любых совершенных систематических кодов последующих еще больших оснований систем счисления ($n = 2^{26}$, 2^{57}, 2^{120}…). Число контрольных разрядов таких кодов соответствует данным табл. 6.2.1. Каждая строка этой таблицы отвечает определенному минимальному кодовому расстоянию $d_{мин\ [1]}$, $d_{мин\ [3]}$, $d_{мин\ [5]}$, … кода, где $d_{мин\ [1]}$ определяет основной двоичный код (k = 0 для всех оснований систем счисления) без исправления ошибок.

Таблица 6.2.1

	$n = 2^4$	$n = 2^{11}$	$n = 2^{26}$	$n = 2^{57}$	$n = 2^{120}$	Исправляемые ошибки
$d_{мин\ [1]}$	k = 0	k = 0	k = 0	k = 0	k = 0	0
$d_{мин\ [3]}$	k = 3	k = 4	k = 5	k = 6	k = 7	1
$d_{мин\ [5]}$	k = 8	k = 16	k = 32	k = 64	k = 128	1,2
$d_{мин\ [7]}$	k = 11	k = 20	k = 37	k = 70	k = 135	1,2,3
$d_{мин\ [9]}$	k = 16	k = 32	k = 64	k = 128	k = 256	1,2,3,4
$d_{мин\ [11]}$	k = 19	k = 36	k = 69	k = 134	k = 263	1,2,3,4,5
$d_{мин\ [13]}$	k = 24	k = 48	k = 96	k = 192	k = 384	1,2,3,4,5,6
$d_{мин\ [15]}$	k = 27	k = 52	k = 101	k = 198	k = 391	1,2,3,4,5,6,7

Вторая строка этой таблицы с минимальным кодовым расстоянием $d_{мин\ [3]}$ определяет число контрольных разрядов всех совершенных кодов, исправляющих одиночные ошибки. Остальные строки таблицы относятся к квазисовершенным кодам, которые по определению используют не все многомерное цифровое пространство систематического кода, но исправляют все ошибки, определяемые минимальным кодовым расстоянием строки.

Как ранее установлено, число совершенных кодов этой таблицы, исправляющих все одиночные ошибки, огромно и определяется простой зависимостью $S = 2^k(i!)$, а число квазисовершенных кодов, исправляющих большее число ошибок, где $k > i$, не зависит от числа контрольных разрядов k и значительно больше. Оно совпадает с числом поворотов относительно осей симметрии информационной части систематического код $S = 2^i(i!)$. Число этих квазисовершенных кодов относится только к основаниям систем счисления $n = 2^4$, 2^{11}, 2^{26}, 2^{57}, … этой таблицы.

В общее число квазисовершенных кодов необходимо также добавить коды промежуточных оснований систем счисления, которые известным образом формируются из кодов этой таблицы. Это квазисовершенные коды, исправляющие все одиночные ошибки, оснований систем счисления, отличных от $n = 2^4$, 2^{11}, 2^{26}, 2^{57}, … . Образующими таких квазисовершенных кодов являются совершенные коды табл. 6.2.1.

6.3. Синтез систематических многофазных и интегральных кодов

Необходимость распространить понятие совершенного кода на систематический код, где информационная часть представлена не основным двоичным кодом, а иными типами кодов, определяется неоднородностью существующих более или менее сложных автоматов. Ярким примером таких автоматов являются цифровые устройства электроприводов и систем энергоснабжения.

Многофазные и интегральные коды являются неотъемлемой частью таких устройств. Эти коды при числе фаз больше двух обладают избыточностью и позволяют, как показано ранее, исправлять определенные типы ошибок. Причем возможности исправления большего количества ошибок возрастают с увеличением числа фаз, но исправить, например, все одиночные ошибки в многофазных и интегральных кодах невозможно. Следовательно, решить одновременно задачи исправления ошибок и резервирования без каких-либо дополнений и изменений здесь не получится.

Учитывая, что кодовые комбинации любого многофазного кода могут рассматриваться как совокупность единичных множеств определенных систематических совершенных кодов, синтез совершенных многофазных кодов неразрывно связан с совершенными и квазисовершенными кодами определенных оснований систем счисления.

Причем информационная часть многочисленных совершенных многофазных кодов, подобно двоичному принципу кодирования, может иметь основной многофазный код. Представляется целесообразным в качестве такого основного многофазного кода принять код в записи Либау – Крейга, структура которого одинакова для четного и нечетного числа фаз. Систематический многофазный код, где число фаз равно числу разрядов совершенных двоичных кодов (4, 11, 26, 57,…), из которого он формируется, может также называться совершенным. Все остальные подобные многофазные коды, где число фаз не совпадает с разрядами совершенных двоичных кодов, следует относить к числу квазисовершенных кодов.

При этом очевидно, что систематические многофазные коды с числом фаз включительно до четырех могут использовать все соотношения между информационными и контрольными кодовыми комбинациями систематических совершенных кодов, представленных табл. 4.2.2.

6.3.1. Двухфазный код

Синтез совершенных многофазных кодов начнем с квазисовершенного двухфазного кода, который также является не основным двоичным кодом.

Информационная часть этого систематического кода (рис. 6.17, *а*) имеет минимальное кодовое расстояние $d_{мин} = 1$, а кодовые расстояния контрольных частей кодов $d_x(3)$, определяемые, например, соответственно четырьмя значениями табл. 4.2.2 с координатами 0, 0; 1, 0; 2, 0; 3, 0 (см. рис. 6.17, *б – д*), позволяют получить для этих систематических кодов минимальное кодовое

расстояние $d_{мин}$ = 3 (см. рис. 6.17, *е – з*). Следовательно, эти систематические квазисовершенные двухфазные коды позволяют исправлять все одиночные ошибки. Аналогичные операции могут быть выполнены для всех остальных кодов табл. 4.2.2, но мы ограничимся и в дальнейшем только четырьмя подобными кодами.

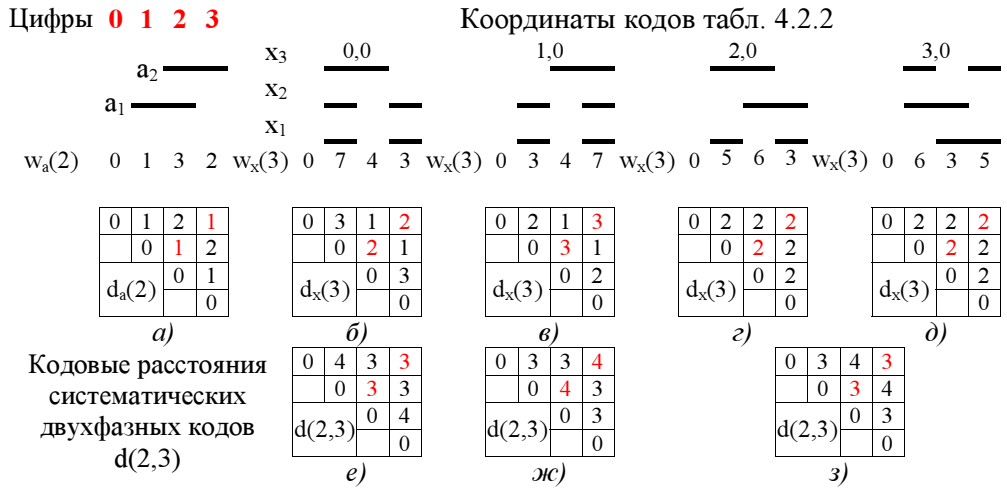


Рис. 6.17

Все эти систематические коды равноценны и содержат по три контрольных разряда, но третий и четвертый коды здесь наиболее предпочтительны для использования, поскольку имеют одинаковые кодовые расстояния контрольной части d = 2.

Для синтеза квазисовершенных систематических двухфазных кодов, исправляющих все одиночные и двойные ошибки, необходимо первоначально составить таблицу совершенных двоичных кодов, исправляющих эти типы ошибок. Эта новая таблица, которая из-за ее громоздкости здесь не приводится, должна быть аналогична табл. 4.2.2. Из этой новой таблицы нами был синтезирован в предыдущем параграфе один совершенный код (см. рис. 6.7), который можно расположить в ней под координатой 0,0. Тогда по аналогии с табл. 4.2.1 несложно представить все остальные 192 кода. Остановимся только на четырех из них, например, соответственно с координатами таблицы 0, 0; 1, 0; 2, 0; 3, 0. Эти коды из новой таблицы приведены на рис. 6.18.

0,0			
0	30	39	57
75	85	108	114
141	147	170	180
198	216	225	255

1,0			
0	39	30	57
75	108	85	114
141	170	147	180
198	225	216	255

2,0			
0	75	39	108
30	85	57	114
141	198	170	225
147	216	180	255

3,0			
0	141	75	198
39	170	108	225
30	147	85	216
57	180	114	255

Рис. 6.18

Исходя из весовых значений кодовых комбинаций информационной части кода (рис. 6.17, *а*), а также соответствующих им весовых значений кодовых комбинаций контрольной части четырех систематических кодов рис. 6.18, несложно представить все кодовые расстояния их контрольных кодовых комбинаций (рис. 6. 19).

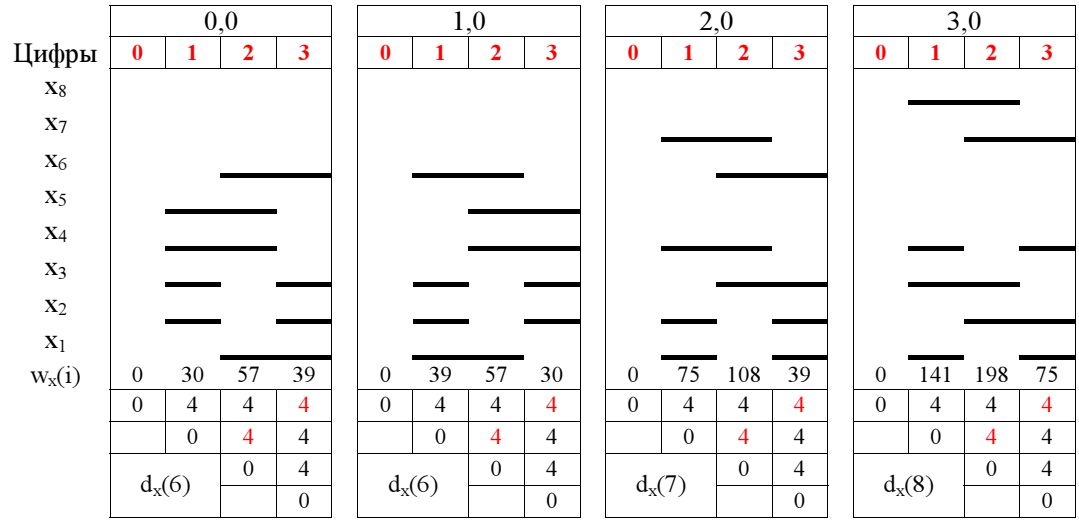


Рис. 6.19

Цифры	0	1	2	3
	0	5	6	5
		0	5	6
	d(2,6)		0	5
				0

Рис. 6.20

Кодовые расстояния контрольных разрядов во всех четырех систематических кодах (см. рис. 6.19) равны d = 4. Следовательно, все кодовые расстояния этих систематических кодов также одинаковы (рис. 6.20). Причем в первых двух систематических кодах имеется по шесть контрольных разрядов, в третьем коде – семь разрядов, а четвертом коде – восемь разрядов, а минимальное кодовое расстояние всех кодов, как это необходимо, $d_{мин} = 5$. Поэтому, учитывая затраты оборудования, только первые два кода можно считать квазисовершенными.

6.3.2. Трехфазный код

Информационная часть систематического кода, представленная в трехфазном варианте (рис. 6.21, *а*), так же, как и для любых многофазных кодов, имеет минимальное кодовое расстояние $d_{мин} = 1$. Кодовые расстояния контрольных частей кодов, определяемые, например, соответственно четырьмя значениями табл. 4.2.2 с координатами 0, 0; 1, 0; 2, 0; 3, 0 (см. рис. 6.21, *б – д*), дают возможность получить для этих систематических кодов минимальное кодовое расстояние $d_{мин} = 3$ (см. рис. 6.21, *е – и*). При этом очевидно, что эти систематические квазисовершенные трехфазные коды исправляют все одиночные ошибки. Аналогичные операции могут быть выполнены для всех остальных кодов табл. 4.2.2.

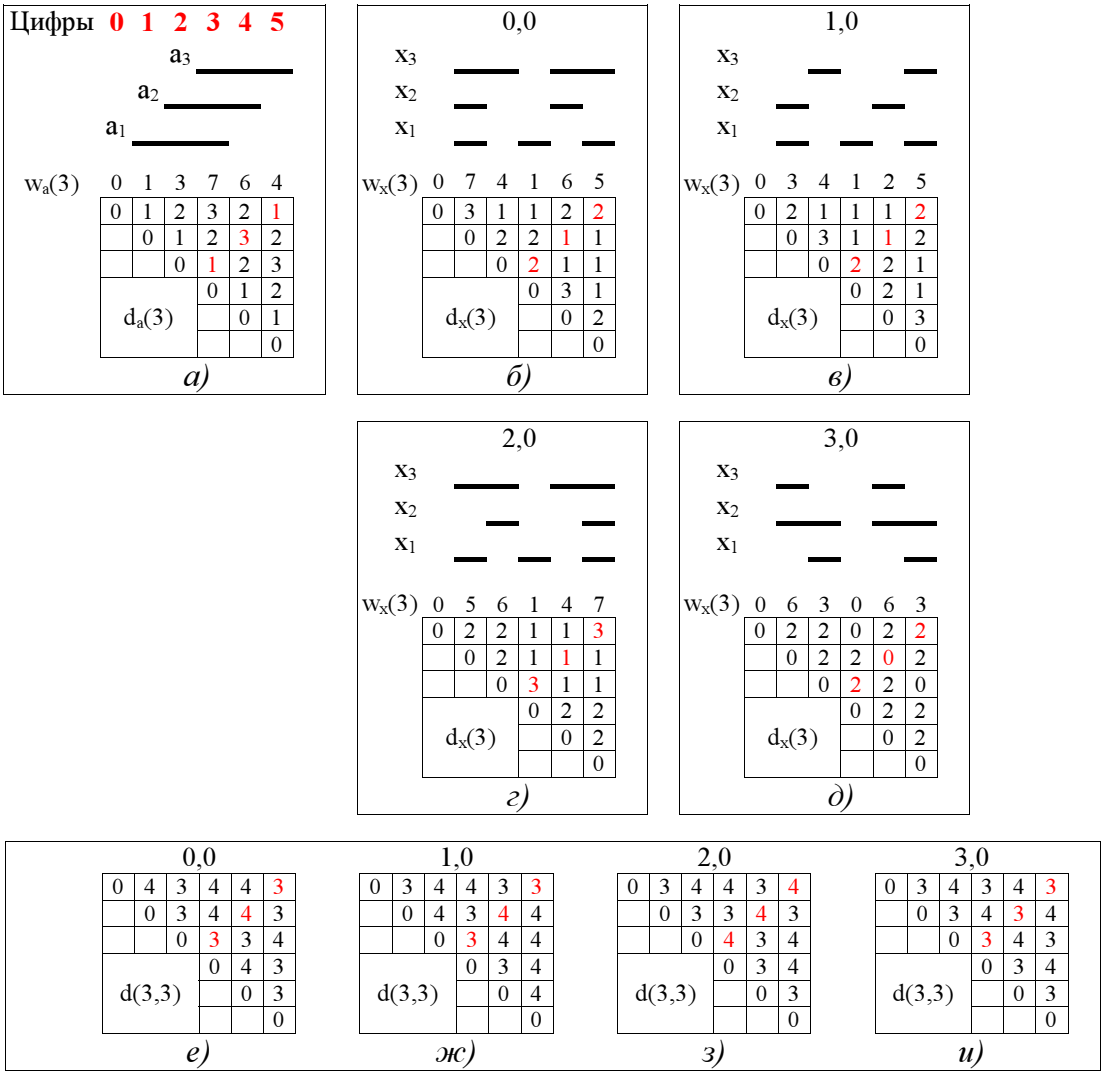


Рис. 6.21

Для синтеза систематических квазисовершенных трехфазных кодов, исправляющих все одиночные и двойные ошибки, необходимо обратиться к совершенным кодам, исправляющим эти же типы ошибок. Четыре типа таких совершенных кодов использовались нами при синтезе квазисовершенных двухфазных кодов, где на рис. 6.18 были выделены цветом кодовые комбинации контрольных разрядов, соответствующие информационным весовым значениям двухфазного кода. При выполнении аналогичных действий для трехфазного информационного кода контрольные разряды этих четырех систематических квазисовершенных кодов (рис. 6.22, *а – г*) будут иметь одинаковые кодовые расстояния d = 4.

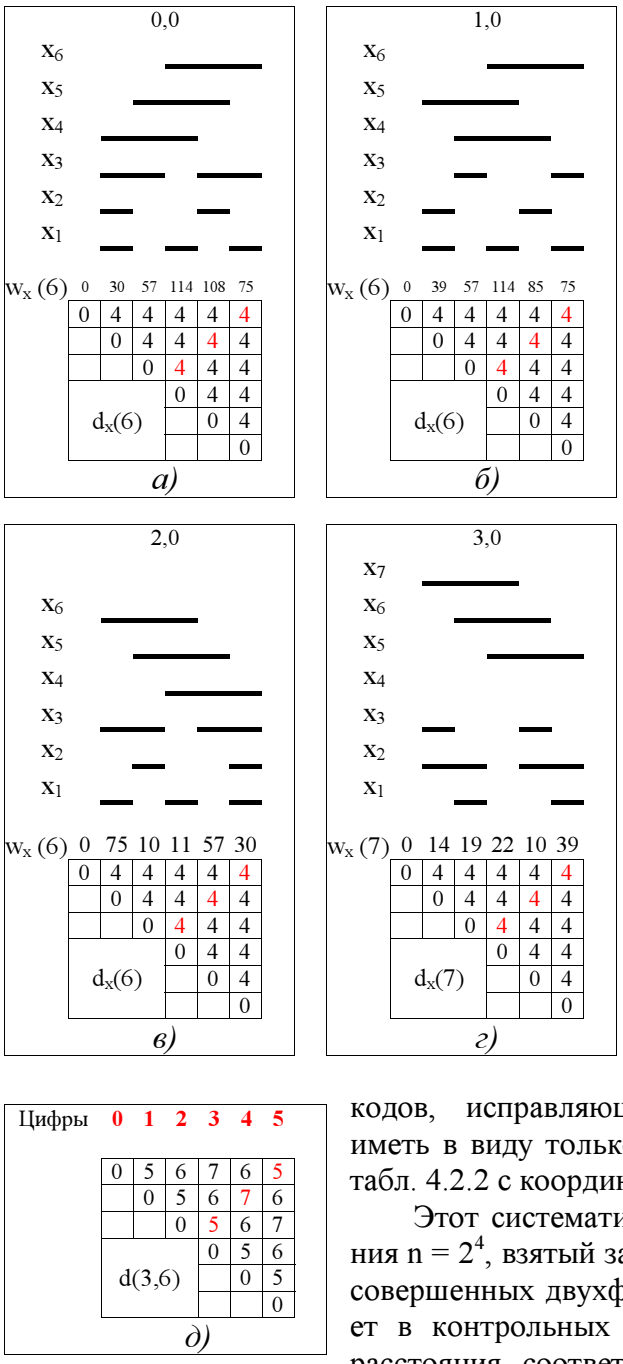


Рис. 6.22

Кодовые расстояния d(3,6) всех этих систематических кодов, которые равны сумме кодовых расстояний информационной и контрольной их частей (см. рис. 6.22, *д*), имеют минимальное значение $d_{мин}$ = 5, что гарантирует исправление всех одиночных и двойных ошибок. Однако для первых двух кодов требуется шесть контрольных разрядов, а для третьего кода – семь контрольных разрядов.

Поэтому только первые три кода из четырех, представленных выше, могут считаться квазисовершенными. Для определения всех квазисовершенных кодов необходимо рассмотреть все 192 совершенных кода, исправляющих одиночные и двойные ошибки, что выходит за рамки настоящей работы.

В дальнейшем изложении будем рассматривать только коды, исправляющие все одиночные ошибки. Поэтому из рассмотренных двухфазных и трехфазных кодов информационных частей систематических кодов, исправляющих эти типы ошибок, будем иметь в виду только образующий совершенный код табл. 4.2.2 с координатами 0,3.

Этот систематический совершенный код основания $n = 2^4$, взятый за основу при формировании квазисовершенных двухфазного и трехфазного кодов, имеет в контрольных частях все одинаковые кодовые расстояния, соответственно равные d = 2, и поэтому таблица кодовых расстояний контрольной части этих кодов может представляться при необходимости только первой строкой, из которой очевидны все другие строки таблицы (см. рис. 6.17, *д*; 6.21, *д*).

6.3.3. Многофазные коды с числом фаз m = 4, 8, 16, …

Рассмотренные в предыдущих двух параграфах квазисовершенные двухфазные и трехфазные систематические коды также практически решают задачу синтеза всех совершенных и квазисовершенных многофазных кодов с числом фаз, кратным двум и трем.

Геометрическая симметрия многофазных кодов ведет к симметрии весовых значений их кодовых комбинаций, а также к симметрии кодовых расстояний информационных частей. Все это позволяет использовать для всех этих системных кодов одинаковую контрольную часть при минимальном числе разрядов, равном трем.

Обратимся к многофазному коду с числом фаз m, где весовые значения их кодовых комбинаций $w_a(m)$ и кодовые расстояния $d_a(m)$ находятся в определенных соотношениях между собой и цифрами основания системы счисления n = 2m.

Эти соотношения выражаются следующим образом:

Цифры	**0**	**1**	**…**	**m**	**m +1**	**m +2**	**…**	**2m – 1**
$w_a(m)$	$2^0 - 1$	$2^1 - 1$	…	$2^m - 1$	$2^m - 2^1$	$2^m - 2^2$	…	$2^m - 2^{m-1}$
$d_a(m)$	0	1	…	m	m –1	m –2	…	1

Для совершенного четырехфазного кода эти соотношения приведены на рис. 6.23, *а*, где также показаны весовые значения кодовых комбинаций контрольных разрядов $w_x(2)$ и значения кодовых расстояний этих разрядов $d_x(3,2)$, где в скобках первым представлено число контрольных разрядов – 3, а вторым – число фаз квазисовершенного двухфазного кода, контрольные разряды которого используются в данном случае.

Кодовое расстояние систематического совершенного четырехфазного кода d(4;3,2), равное сумме кодовых расстояний информационной и контрольной частей, имеет минимальное кодовое расстояние $d_{мин}(4;3,2) = 3$, что гарантирует исправление всех одиночных ошибок.

Более сжатое представление совершенного четырехфазного кода приведено на рис. 6.23, *б*, которое показано только соотношениями между цифрами основания системы счисления и значениями $w_a(4)$, $w_x(3,2)$, $d_a(4)$, $d_x(3,2)$, d(4;3,2).

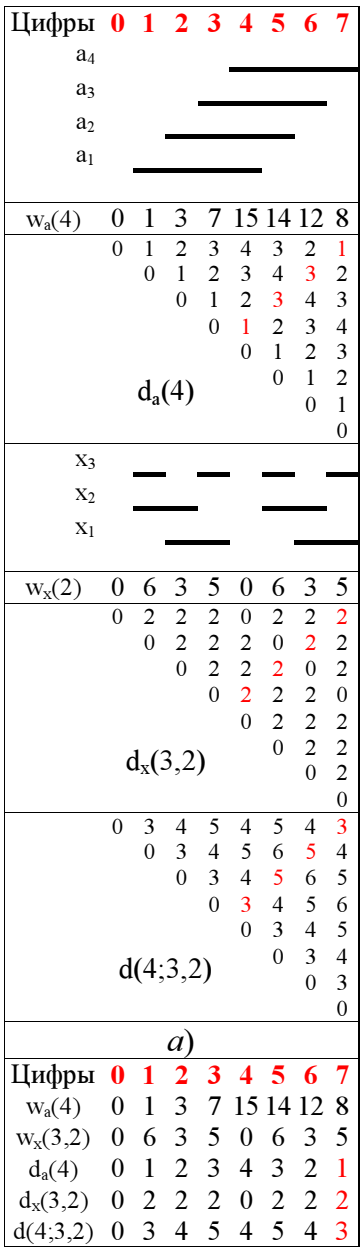


Рис. 6.23

Тогда для восьмифазного кода аналогичные соотношения запишутся следующим образом:

Цифры	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$w_a(8)$	0	1	3	7	15	31	63	127	255	254	252	248	240	224	182	128
$w_x(3,2)$	0	6	3	5	0	6	3	5	0	6	3	5	0	6	3	5
$d_a(8)$	0	1	2	3	4	5	6	7	8	7	6	5	4	3	2	1
$d_x(3,2)$	0	2	2	2	0	2	2	2	0	2	2	2	0	2	2	2
d(8;3,2)	0	3	4	5	4	7	8	9	8	9	8	7	4	5	4	3

и т.д.

6.3.4. Многофазные коды с числом фаз m = 6, 9, 12, 15, 18 …

При синтезе системных квазисовершенных многофазных кодов с числом фаз, одновременно кратных двум и трем (m = 6, 12, 18,…), можно использовать контрольные разряды как квазисовершенных двухфазных кодов, так и трехфазных.

Покажем это на примере шестифазного кода. Использование контрольных разрядов квазисовершенного двухфазного кода позволяет синтезировать квазисовершенный шестифазный код со следующими соотношениями:

Цифры	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$w_a(6)$	0	1	3	7	15	31	63	62	60	56	48	32
$w_x(3,2)$	0	6	3	5	0	6	3	5	0	6	3	5
$d_a(6)$	0	1	2	3	4	5	6	5	4	3	2	1
$d_x(3,2)$	0	2	2	2	0	2	2	2	0	2	2	2
d(6;3,2)	0	3	4	5	4	7	8	7	4	5	4	3

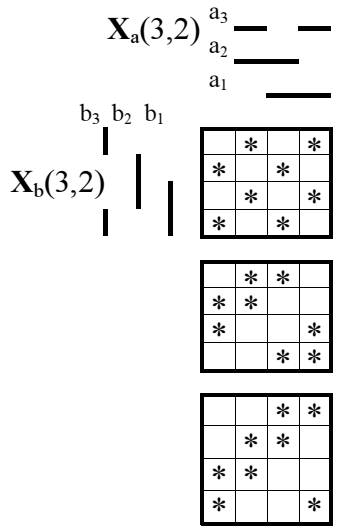
Аналогичные зависимости, когда контрольные разряды совпадают с такими же разрядами квазисовершенного трехфазного кода, представляются следующим образом:

Цифры	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$w_a(6)$	0	1	3	7	15	31	63	62	60	56	48	32
$w_x(3,3)$	0	6	3	0	6	3	0	6	3	0	6	3
$d_a(6)$	0	1	2	3	4	5	6	5	4	3	2	1
$d_x(3,3)$	0	2	2	0	2	2	0	2	2	0	2	2
d(6;3,3)	0	3	4	5	6	7	8	7	6	5	4	3

Эти два типа системных квазисовершенных шестифазных кодов, исправляющих все одиночные ошибки, имеют одинаковую длину кода и равноценны по затратам оборудования для операций кодирования и декодирования. Однако второй системный код, где контрольные разряды равны контрольным разрядам квазисовершенного трехфазного кода, имеют определенные преимущества по

затратам оборудования для ряда устройств машинной арифметики и логических схем.

Например, двухвходовой сумматор контрольных разрядов двухфазного квазисовершенного кода (рис. 6.24, *а*) содержит больше элементов, чем сумматор контрольных разрядов трехфазного квазисовершенного кода (см. рис. 6.24, *б*).



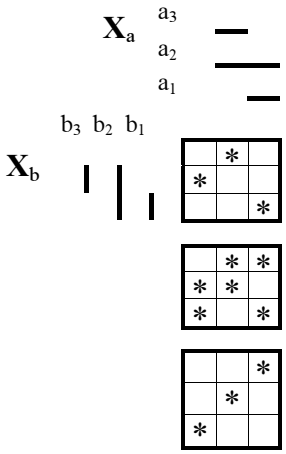
Сумматор $\mathbf{X}_c(3,2) = \mathbf{X}_a(3,2) + \mathbf{X}_b(3,2)$

$c_1(3,2) = a_3\,\underline{b_3} \vee \underline{a_3}\,b_3$

$c_2(3,2) = a_2\,\underline{b_1}\,\underline{b_2} \vee \underline{a_1}\,\underline{b_1}\,b_2 \vee \underline{a_2}\,b_1\,b_2 \vee a_1\,b_1\,b_3$

$c_3(3,2) = a_1\,\underline{b_1}\,\underline{b_2} \vee a_2\,\underline{b_1}\,b_2 \vee \underline{a_1}\,b_1\,b_2 \vee \underline{a_2}\,b_1\,b_3$

а)



Сумматор $X_c(3,3) = X_a(3,3) + X_b(3,3)$

$c_1(3,3) = a_3\,\underline{b_2} \vee \underline{a_2}\,b_3 \vee a_1\,b_1$

$c_2(3,3) = c_1(3,3) + c_3(3,3) = a_2\,\underline{b_2} \vee \underline{a_1}\,b_3 \vee \underline{a_3}\,b_1$

$c_3(3,3) = a_1\,\underline{b_2} \vee \underline{a_2}\,b_1 \vee a_3\,b_3$

б)

Рис. 6.24

Для систематических квазисовершенных многофазных кодов с числом фаз, кратным трем (m = 9, 15, 21,…), нет альтернативы выбора – число их контрольных разрядов равно контрольным разрядам квазисовершенного трехфазного кода.

6.3.5. Интегральные коды

Интегральные коды обладают высокой избыточностью и, аналогично многофазным кодам, позволяют исправлять ошибки определенной кратности. С увеличением основания системы счисления, которая соответствует этим кодам, возможности исправления ошибок увеличиваются.

Цифры	0	1	2	3	4	5
$d_a(5)$	0	1	2	3	4	5
$d_x(3,3)$	0	2	2	0	2	2
$d_a(5;3,3)$	0	3	4	3	6	7

а)

Цифры	0	1	2	3	4	5	6	7
$d_a(7)$	0	1	2	3	4	5	6	7
$d_x(3,2)$	0	2	2	2	0	2	2	2
$d_a(7;3,2)$	0	3	4	5	4	7	8	9

б)

$d_a(11)$	0	1	2	3	4	5	6	7	8	9	10	11
$d_x(3,3)$	0	2	2	0	2	2	0	2	2	0	2	2
$d_a(11;3,3)$	0	3	4	3	6	7	6	9	10	9	12	13

в)

$d_a(11)$	0	1	2	3	4	5	6	7	8	9	10	11
$d_x(3,2)$	0	2	2	2	0	2	2	2	0	2	2	2
$d_a(11;3,2)$	0	3	4	5	4	7	8	9	8	11	12	13

г)

Рис. 6.25

Однако исправить все ошибки определенной кратности в этих кодах также нельзя. Подобно многофазным кодам здесь возможно осуществить синтез квазисовершенных кодов, где информационная часть системного кода – это интегральный код, а его контрольная часть совпадает с контрольной частью квазисовершенного двухфазного или трехфазного кодов.

Для интегрального кода основания n = 5 (рис. 6.25, *а*), как и для любых интегральных кодов, кодовые расстояния всегда совпадают с цифрами основания системы счисления. Сумма кодовых расстояний этого интегрального кода $d_a(5)$ и кодовых расстояний контрольных разрядов квазисовершенного трехфазного кода $d_x(3,3)$ образуют кодовые расстояния систематического кода $d_a(5;3,3)$, исправляющего все одиночные ошибки ($d_{мин} = 3$).

Интегральный код основания n = 8 (см. рис. 6.25, *б*) совместно с контрольными разрядами квазисовершенного двухфазного кода $d_x(3,2)$ образует систематический квазисовершенный интегральный код $d_a(7;3,2)$, где также $d_{мин} = 3$.

Интегральный код $d_a(11)$ основания n = 12 может образовывать два системных квазисовершенных кода: первый код (см. рис. 6.25, *в*) включает контрольные разряды квазисовершенного трехфазного кода $d_x(3,3)$, второй (см. рис. 6.25, *г*) – контрольные разряды квазисовершенного двухфазного кода $d_x(3,2)$. На этих рисунках из-за их очевидности не показаны сигналы информационных разрядов интегрального кода. Кодовые расстояния этих системных кодов $d_a(11;3,3)$, $d_a(11;3,2)$ также имеют минимальные кодовые $d_{мин} = 3$.

Дальнейшее представление интегральных квазисовершенных кодов больших оснований систем счисления очевидно из вышеизложенного.

6.3.6. Синтез декодирующих блоков для системных многофазных и интегральных кодов

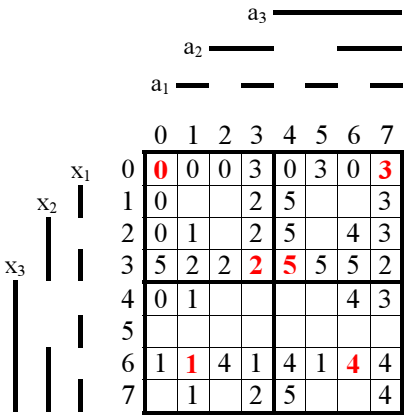
Общеизвестно, что кодирование любых типов кодов, используемых как для передачи цифровых данных, так и для выполнения логических и арифметических операций, «не составляет проблемы» [10]. При этом использование систематических кодов для любых вычислительных машин является предпочтительным. Не составляют здесь исключения и многофазные коды, которые используются в информационной части представленных выше системных квазисовершенных кодов.

Не существует принципиального отличия в подходе к синтезу декодирующих устройств таких квазисовершенных кодов от того, который был изложен нами для двоичного принципа кодирования. Поэтому рассмотрим только два систематических кода, предназначенных для исправления одиночных ошибок, где информационная часть кода выполнена в трехфазном и четырехфазном варианте. Из этих двух примеров будет ясно построение декодирующих устройств для кодов любой фазности, кратных трем и двум.

Соотношения между кодовыми комбинациями информационной a_1, a_2, a_3 и контрольной x_1, x_2, x_3 частей систематического трехфазного кода (см. рис. 6.21, *д*) позволяют составить геометрический образ рабочей и нерабочей части шестимерного пространства координат (рис. 6.26).

Рабочая часть этого пространства содержит штатные цифры **0–5** и цифры 0–5 с одиночной ошибкой в контрольной либо информационной частях систематического кода.

Указанные два рисунка позволяют сформировать геометрические образы исправленных информационных a'_1, a'_2, a'_3 (рис. 6.27) и контрольных x'_1, x'_2, x'_3 (рис. 6.28) разрядов систематического трехфазного кода.



	0	1	2	3	4	5	6	7
0	**0**	0	0	3	0	3	0	**3**
1	0			2	5			3
2	0	1		2	5		4	3
3	5	2	2	**2**	**5**	5	5	2
4	0	1					4	3
5								
6	1	**1**	4	1	4	1	**4**	4
7		1		2	5			4

Рис. 6.26

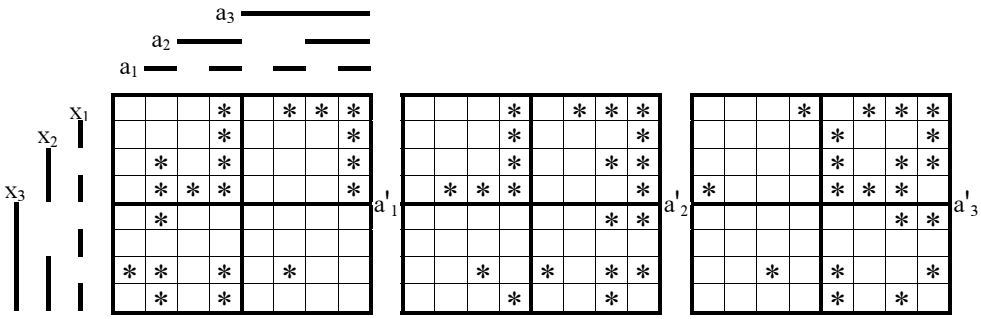


Рис. 6.27

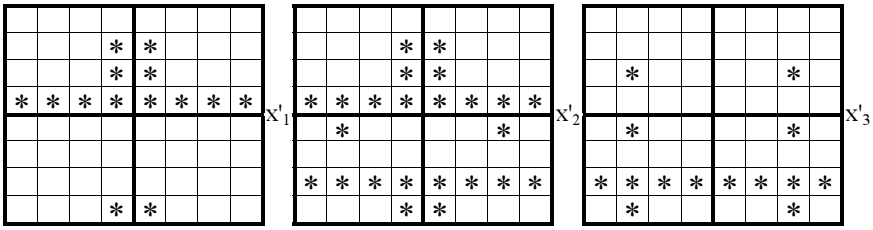


Рис. 6.28

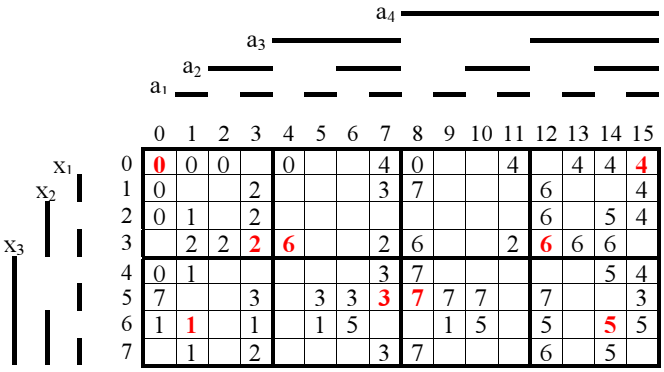


Рис. 6.29

Поскольку размещение в цифровом пространстве с информационными и контрольными координатами штатных цифр соответствующих им оснований систем счисления, а также этих же цифр с одиночной ошибкой полностью определяет геометрические образы исправленных сигналов разрядов систематического кода, то для четырехфазного кода ограничимся только представлением этого промежуточного геометрического образа (рис. 6.29).

Очевидно, что всякий интегральный код (см. рис. 6.25, *а, б*) можно считать как начальную часть многофазного кода неограниченно большой любой фазности. Поэтому для построения систематического интегрального кода, исправляющего все одиночные ошибки, в качестве контрольных разрядов допустимо использовать контрольные разряды как квазисовершенных двухфазных кодов, так и трехфазных. В дальнейшем синтез декодирующих блоков систематических интегральных кодов будет полностью совпадать с многофазным вариантом этих кодов.

6.3.7. Синтез кода, управляющего переключением транзисторов трехфазного инвертора напряжения и исправляющего все одиночные ошибки

В большинстве инверторов напряжения для переключения силовых транзисторов используются управляющие многофазные коды, которые ранее нами были достаточно полно исследованы. Вместе с тем имеются такие трехфазные инверторы напряжения, где применяются специальные коды основания системы счисления $n = 6$.

Этот тип кода (рис. 6.30) содержит шесть информационных разрядов $a_1 - a_6$, где кодовые расстояния между кодовыми комбинациями $d_a(6)$ имеют только два значения: 2 и 4. Для того чтобы иметь возможность исправлять все одиночные ошибки, необходимо сформировать систематический код с одним контрольным разрядом x.

Созданный таким образом систематический код имеет кодовые расстояния $d_{a,x}(6, 1)$ с минимальным значением три, а геометрический образ цифрового пространства, в ячейках которого размещены (рис. 6.31) штатные цифры **0**–**5** и цифры с одиночными ошибками 0 – 5, позволяет определить геометрические образы всех исправленных сигналов разрядов $a'_1 - a'_6$, x' и соответственно синтезировать принципиальные схемы их декодирующих блоков.

Цифры	**0**	**1**	**2**	**3**	**4**	**5**
a_6	—	—				
a_5			—	—		
a_4					—	—
a_3				—	—	
a_2	—					—
a_1		—	—			
$w_a(6)$	34	33	17	20	12	10
$d_a(6)$	0	2	4	4	4	2
		0	4	4	4	4
			0	2	4	4
				0	2	4
					0	2
x		—		—		—
$d_x(1)$	0	1	0	1	0	1
		0	1	0	1	0
			0	1	0	1
				0	1	0
					0	1
$d_{a,x}(6, 1)$	0	3	4	5	4	3
		0	3	4	5	4
			0	3	4	5
				0	3	4
					0	3

Рис. 6.30

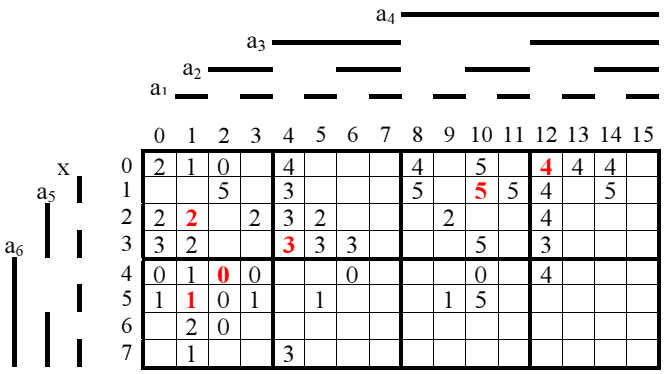


Рис. 6.31

Заключение

Теория многомерных цифро-векторных множеств находится в стадии становления, и ее базис может быть окончательно сформулирован только на завершающем этапе создания.

Тем не менее применение этой теории даже на данном промежуточном этапе позволило решить многие практические задачи синтеза цифровых и логических устройств, работающих в режиме реального времени. В основу теории положена цифровая версия геометрического пространства известного ученого академика Е.С. Федорова, которая исследована нами без применения аксиоматического метода. Это дает повод «вспомнить о том, что аксиоматический метод в его уточненном виде отнюдь не является исконным методом математики». И далее: «…в области элементарной арифметики и алгебры, ориентировка на прямые содержательные рассуждения, осуществляемые без предложений аксиоматического характера, разработана в наиболее чистом виде» [*Гилберт Д., Бернайс* П. Основания математики (Сер. Математическая логика и основания математики). М., 1979].

Использование теории многомерных цифро-векторных множеств для решения конкретных практических задач электропривода и систем энергоснабжения приведено в книге: *Кочергин В.И.* Теория многомерных цифровых множеств в приложениях к электроприводам и системам электропитания. Томск: Изд-во Том. ун-та, 2002.

В теоретическом плане в теории решены следующие проблемы:

1. Сняты любые ограничения (тип кода, основание системы счисления, число разрядов и операндов, число входов и выходов и т.д.) по синтезу оптимальных по быстродействию и затратам оборудования цифровых и комбинационных логических устройств, работающих в режиме реального времени.

2. Доказано, что любые коды позиционных систем счисления являются арифметическими, в которых исправление ошибок любой кратности может решаться комбинационными логическими схемами в режиме реального времени.

3. Доказано, что число совершенных и квазисовершенных кодов позиционных систем счисления неограниченно велико.

Ценность любой теории, в том числе и математической, всегда определяется потенциальными возможностями её развития в соответствии с внутренней логикой этой теории и в значительной степени практическими приложениями её результатов, что было доказано в данной и упомянутой выше книгах автора.

Литература

К введению

1. *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. М.: Сов. радио, 1968.
2. *Гильберт Д., Кон-Фонссен С.* Наглядная геометрия / Пер. с нем. 3-е изд. М.: Наука, 1981.
3. *Бибило П.Н.* Синтез комбинационных ПЛМ-структур для СБИС. Минск: Наука и техника, 1992.
4. *Бибило П.Н., Енин С.И.* Синтез комбинационных схем методами функциональной декомпозиции. Минск: Наука и техника, 1987.
5. *Бибило П.Н.* Применение дизъюнктивных разложений при синтезе структур ПЛИС // Автоматизация проектирования дискретных систем: Матер. II. Междунар. конф. (12–14 нояб. 1997 г., Минск) / Ин-т техн. кибернетики АН Беларуси. Минск, 1997. Т. 3.
6. *Бояринов И.М., Кабатянский Г.А.* Арифметические итеративные коды, исправляющие независимые ошибки // Проблемы передачи информации. 1979. Т. 15, вып. 1.
7. *Бояринов И.М., Кабатянский Г.А.* Совершенные арифметические AN-коды, исправляющие одиночные ошибки // Проблемы передачи информации. 1976. Т. 12, вып. 1.
8. *Бояринов И.М., Кабатянский Г.А.* Один класс арифметических итеративных кодов // Вопросы кибернетики / АН СССР. Научный совет по комплексной проблеме «Кибернетика». М., 1977. Вып. 29.
9. *Бояринов И.М.* О декодировании AN-кодов, исправляющих пакеты ошибок // Проблемы передачи информации. 1972. Т. 8, вып. 2.
10. *Буль Е.С., Чапенко В.П.* Декомпозиция булевых функций посредством решения логического уравнения // Автоматика и вычислительная техника. 1996. № 4.
11. *Брейтон Р.К., Хэтчел Г.Д., Сонджованни-Винчентелли А.Л.* Синтез многоуровневых комбинационных схем // ТИИЭР. 1990. Т. 78, № 2. Февр.
12. *Брюхович Е.И.* О проблемах автоматического контроля в ЭВМ и контролеспособности позиционных счислений // Управляющие системы и машины. 1977. № 4.
13. *Брюхович Е.И.* Экстремальная эффективность аппаратного контроля ЭВМ и принципиальная возможность ее достижения на основе естественной избыточности позиционных счислений // Управляющие системы и машины. 1979. № 6.
14. *Брюхович Е.И.* Автоматический контроль и производительность ЭВМ // Управляющие системы и машины. 1979. № 4.
15. *Василенко В.С.* К вопросу обнаружения и исправления ошибок в представлении чисел в остаточных классах // Управляющие системы и машины. 1977. № 4.
16. *Гриценко И.М.* Недвоичные арифметические корректирующие коды // Проблемы передачи информации. 1969. Т. 5, вып. 4.
17. *Дынькин В.Н., Тененгольц Г.М., Хабелашвили Г.И.* Об одном классе циклических арифметических кодов // Сообщения АН ГССР. 1969. Т. 55, № 3.
18. *Дискретная* математика и математические вопросы кибернетики / Под ред. С.В. Яблонского и О.Б. Лупанова. М.: Наука, 1974.
19. *Дадаев Ю.Г.* Арифметические разделимые коды с исправлением независимых ошибок // Изв. АН СССР. Техническая кибернетика. 1965. № 6.
20. *Дадаев Ю.Г.* Об использовании корректирующей способности арифметических кодов с исправлением ошибок // Изв. АН СССР. Техническая кибернетика. 1968. № 2.

21. *Дадаев Ю.Г.* Арифметические композиционные коды с исправлением ошибок // Проблемы передачи информации.1968. Т. 4, вып. 2.

22. *Дадаев Ю.Г.* Арифметические коды, исправляющие ошибки. М.: Сов. радио, 1969.

23. *Дадаев Ю.Г.* К теории циклических арифметических кодов // Проблемы передачи информации. 1970. Т. 6, вып. 1.

24. *Дадаев Ю.Г.* Циклическая структура AN-кодов // Проблемы передачи информации. 1970. Т. 6, вып. 4.

25. *Дадаев Ю.Г.* К задаче определения минимальных расстояний циклических AN-кодов // Проблемы передачи информации. 1974. Т. 10, вып. 1.

26. *Дадаев Ю.Г.* О применении арифметических кодов, исправляющих ошибки в ЦВМ // Кодирование в сложных системах. М.: Наука, 1974.

27. *Дадаев Ю.Г.* Помехоустойчивое кодирование в вычислительных машинах // Вопросы кибернетики. М.: АН СССР. Научный совет по комплексной проблеме «Кибернетика». 1977. Вып. 28.

28. *Дадаев Ю.Г.* Теория арифметических кодов. М.: Радио и связь, 1981.

29. *Димитриев Ю.К., Хорошевский В.Г.* Вычислительные системы из мини-ЭВМ. М.: Радио и связь, 1982.

30. *Закревский А.Д.* Синтез асинхронных автоматов на ЭВМ. Минск: Наука и техника, 1975.

31. *Закревский А.Д.* Логический синтез каскадных схем. М.: Наука, 1981.

32. *Закревский А.Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971.

33. *Кондратьев В.Н., Трофимов Н.И.* Корректирующие коды с расстоянием, не менее пяти по Питерсону // Изв. АН СССР. Техническая кибернетика. 1969. № 3.

34. *Кочергин В.И.* Теория многомерных цифро-векторных множеств в технических системах управления: Дис. д-ра техн. наук. Томск: ТПУ, 2003.

35. *Кочергин В.И.* Теория многомерных цифровых множеств в приложениях к электроприводам и системам электропитания. Томск: Изд-во Том. ун-та, 2002.

36. *Кладов Г.К., Шпильберг А.Я.* Об одном классе избыточных арифметических кодов // Кибернетика. 1966. № 4.

37. *Новиков С.В.* Теория регулярных структур. Минск: Университетское изд-во, 1987.

38. *Новиков С.В.* Метод реализации системы частичных булевых функций схемой на программируемых логических матрицах // Автоматика и вычислительная техника. 1980. № 6.

39. *Обнаружение* и исправление ошибок в дискретных устройствах / В.С. Толстяков, В.Н. Номоконов, И.Л. Ерош и др. М.: Сов. радио, 1972.

40. *Питерсон У., Уэлсон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.

41. *Погарцев А.Г.* Новые алгоритмы совместной минимизации булевых функций // Автоматика и вычислительная техника. 1980. № 1.

42. *Путинцев Н.Д.* Аппаратный контроль управляющих цифровых вычислительных машин. М.: Сов. радио, 1966.

43. *Селлерс Ф.* Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972.

44. *Ситниченко С.И.* О декодировании арифметических корректирующих кодов // Изв. АН СССР. Техническая кибернетика. 1970. № 4.

45. *Стахов А.П.* «Золотая» пропорция в цифровой технике // Автоматика и вычислительная техника. 1980. № 1.

46. *Стахов А.П.* Перспективы применения систем счисления с иррациональными основаниями в технике аналого-цифрового и цифроаналогового преобразования // Измерения, контроль, автоматизация. 1981. № 6 (40).

47. *Стахов А.П.* Использование естественной избыточности «фибоначчиевых» систем счисления для контроля вычислительных систем // Автоматика и вычислительная техника. 1975. № 6.

48. *Стахов А.П.* Фибоначчиевы двоичные позиционные системы счисления // Кодирование и передача дискретных сообщений в системах связи. М.: Наука, 1976.

49. *Стахов А.П., Лужецкий В.А.* Машинная арифметика ЦВМ в кодах Фибоначчи и золотой пропорции. М.: Научный совет АН СССР по комплексной проблеме «Кибернетика», 1981.

50. *Стахов А.П.* Коды золотой пропорции. М.: Радио и связь, 1984.

51. *Соколов О.Б., Еникеев И.И.* Класс арифметических кодов с исправлением нескольких ошибок // Проблемы передачи информации. 1976. Т. 3, вып. 4.

52. *Супрун В.П.* Метод реализации на ПЛМ булевых функций, заданных в интервальной форме // Автоматика и вычислительная техника. 1982. № 5.

53. *Соловьев В.В.* Проектирование цифровых систем на основе программируемых логических интегральных схем. М.: Горячая линия – Телеком, 2001.

54. *Тауглих Г.Л.* Циклические коды, исправляющие двойные пакеты арифметических ошибок // Проблемы передачи информации. 1976. Т. 12, вып. 4.

55. *Трофимов Н.Н.* К определению границы избыточных кодов с коррекцией арифметических ошибок // Изв. АН СССР. Техническая кибернетика. 1968. № 1.

56. *Ушакова Г.Н.* Аппаратный контроль и надежность специализированных ЭВМ. М.: Сов. радио, 1969.

57. *Уткин А.А.* Анализ логических сетей и техника булевых вычислений. Минск: Наука и техника, 1979.

58. *Федоров Е.С.* Начала учения о фигурах. М., 1953 (перепечатка издания 1905 г.).

59. *Хетагуров Я.А., Руднев Ю.П.* Повышение надежности цифровых устройств методами избыточного кодирования. М.: Энергия, 1974.

60. *Шалыто А.А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000.

61. *Шоломов Л.А.* Основы теории дискретных логических и вычислительных устройств. М.: Наука, 1980.

62. *Шестаков Е.А.* О декомпозиции систем полностью определенных булевых функций методом тождественных отображений // Автоматизация проектирования дискретных систем: Матер. II Междунар. конф. (12–14 нояб. 1997 г., Минск) / Ин-т техн. кибернетики АН Беларуси. Минск, 1997. Т. 2.

63. *Шестаков Е.А.* Декомпозиции системы полностью определенных булевых функций по покрытию аргументов // Автоматика и вычислительная техника. 1994. № 1.

64. *Шестаков Е.А.* Декомпозиции системы частичных булевых функций по покрытию аргументов // Автоматика и вычислительная техника. 1994. № 6.

65. *Яблонский С.В.* Введение в дискретную математику. М.: Наука, 1986.

66. *Ashenhurst R.L.* The decomposition of switching functions // Proc. of the Int. Symposium on Theory of switching functions. 1957.

67. *Barrows J.T.* A new method for coustracting multiple error correcting linear residul codes. Coord. Science Lab/Univ. Illinois, Urbana. 1966.

68. *Bergman G.* A number system with an irrational base // Mathematics Magazine. 1957. № 31.

69. *Brand D.* PLA-based synthesis without PLA's // The Proc. Int. Workshop on Logic Synthesis. 1989. May.

70. *Brayton R.K.* Factoring logic functions // IBM Journal of Research and Development. 1987. Vol. 31, № 2.

71. *Brayton R.K., Rudell R., Sangiovanni-Vincentelli A., Wang A.* A multi-level logic optimization and the rectangular covering problem // The Proc. of the Int. Conf. On Computer-Aided Design (ICCAD). 1987. Nov.

72. *Brayton R.K., Rudell R., Sangiovanni-Vincentelli A., Wang A.* MIS. A multiple-level logic optimization system // IEEE Trans. On CAD. 1987. Vol. CAD-6, № 6.

73. *Brayton R.K., McMullen C.* Decomposition and factorization of Boolean expressions // The Proc. of the Int. Symposium on Circuits and Systems (ISCAS-82). 1982. Apr.

74. *Brzozowski J.A., Luba T.* Decomposition of Boolean function specified by cubes. Part 1: Theory of serial decompositions using blankets. Research Report CS-97-01. Canada: University of Waterloo, 1997.

75. *Bryant R.E.* Graph-based algorithms for Booleam function manipulation // IEEE Trans. on Computers. 1986. Vol. 35, № 8.

76. *Brown D.T.* Error detecting and correcting binary codes for arithmetic operations // IRE Trans. 1960, Vol. EC-9, № 3.

77. *Chang S.-H., Tsao-Wu N.T.* Discussion on «Arithmetic codes with large distance» // IEEE Trans. 1968, Vol. IT-14, № 1.

78. *Chang S.C., Marek-Sadowska M., Hwang T.T.* Technology mapping for LUT FPGAs based on decomposition of binary dicision diagrams // IEEE Trans. on CAD. 1996. Vol. 15, № 10. Oct.

79. *Chen C.-L., Chien R.-T., Liu C.-K.* On majority-logic-decodable arithmetic codes // IEEE Trans. 1973. Vol. IT-19, № 5.

80. *Ciesielski M.J., Yang S.* PLADE: A two-stage PLA decomposition // IEEE Trans. on CAD. 1992. Vol. 11, № 8. Aug.

81. *Coudert O., Madre J., Fraisse H.A.* New viepoint on two-level logic minimization // The Proc. of the 30th Design Automation Conference (DAC). 1993.

82. *Coudert O., Madre J.* Implicit and incremental computation of primes and essential primes of Booolean functions // The Proc. of the 30th Design Automation Conference (DAC). 1992.

83. *Curtis H.A.* A new approach to design of switching circuits. Princeton; N.J.: Van Nostrand, 1962.

84. *Davidson E.* An algoristhm for NAND decomposition under network constraints // IEEE Trans on Computers. 1969. Vol. C-18, № 12.

85. *Diamond J.M.* Cheking codes for digital computers // Proc. IRE. 1955. Vol. 43.

86. *Goto M., Facuma T.* Perfect nonbinary AN codes with distance three // Information and Control. 1975. Vol. 27, № 4.

87. *Goto M.* A note on perfect decimal AN codes // Information and Control. 1975. Vol. 29, № 4.

88. *Goto M., Facuma T.* The distance of arithmetic codes // Memories Faculty Eng. 1969. Vol. 20, № 2.

89. *Goto M., Facuma T.* Nonbinary AN codes with distance not less than five // IEEE Trans. 1973. Vol. IT-19, № 1.

90. *Gregory D., Bartlett K., DeCeus A., Hachtel G.* / SOCRATES: A system for automatically synthesizing and optimizing combinational logic // Proc. of the 23th Design Automation Conference (DAC). 1986.

91. *Hong S.J.* On bounds and implementation of arithmetic codes // Coord. Science Lab. / Univ. 1969. Rep. R-437.

92. *Hong S., Muroga S.* Absolute Minimization of Completely Specified Swwitching Function // IEEE Trans. On Computers. 1991. Vol. 40, № 1.

93. *Hwang T.-Y., Hartmann C.R.P.* Some results on arithmetic codes of composite length // IEEE Trans. 1978. Vol. IT-24, № 1.

94. *Jozwiak L.* General decomposision and its use in digital circuit synthesis, VLSI design // An International Journal of Custom-Chip Design Simulation, and Testing, Special Issue on Decomposition in VLSI Design. 1995. Vol. 3, № 3–4.

95. *Jozwiak L.* Information relationships and measures: An analysis apparatus for efficient information system synthesis // Proc. of the 25rd EUROMICRO Conference (Milan, Italy, September 8–10). 1999. Vol. 1.

96. *Lavagno L., Malik S., Brayton R., Sangiovanni-Vincentelli A.* MIS-MV: Optimization of multi-level logic with multiple valued inputs // Proc. of the 27th Design Automation Conference (DAC). 1990.

97. *Luba T., Kalinowski J., Jasinski K.* PLATO: A CAD toolfor logic synthesis based on decomposition // Proc. of the European Conference on Design Automation. 1991.

98. *Luba T.* Multi-level logic synthesis based on decomposition // Microprocessors and Microsystems. 1994. Vol. 18, № 8.

99. *Kania D.* Two-level logic synthesis on PAL-based CPLDand FPGA using decomposition // Proc. of the 25rd EUROMICRO Conference (Mlan, Italy, September 8–10). 1999. Vol. 1.

100. *Kautz W.H.* Fibonacci codes for synchronization control // IEEE Trans. Inform Theory. 1965. Vol. 11, № 8.

101. *Mc. Cluskey E.J.* Minimization of boolen function // Bell System Techn. J. 1956. Vol. 35.

102. *Mandelbaum D.* Multivalued arithmetic burst error codes // IEEE Int. Conv. Rec. 1966. Vol. 14.

103. *Mandelbaum D.* A comparison of linear sequential circuits and arithmetic sequences // IEEE Trans. 1967. Vol. EC-16, № 2.

104. *Malic S., Brayton R., Newton A., Sangiovanni-Vincentelli A.* Two-levelminimization of multivaluad functions with large offsets // IEEE Trans. on Computers. 1993. Vol. 42, № 11.

105. *Mathony H.-J.* Universal logic design algorithm and its application to the synthesis of two-level switching circuits // IEE Proceedings. 1989. Vol. 136, part E, № 3.

106. *Massey J.L., Garcia O.N.* Error correcting codes in computers arithmetic. Advances in information sciences / Ed. by // J.T. Tov. N.Y.: Plenum Press, 1971. Vol. 4.

107. *Murgai R., Brayton R., Sangiovanni-Vincentelli A.* Optimum functional decomposition using encoding // Proc. of the 31th Dising Automation Conference (DAC). 1987.

108. *Murgai R., Shenoy N., Brayton R., Sangiovanni-Vincentelli A.* Improved logic synthesis algorithm for table look up architectures // Proc. of the Int. Conf. On Computer-Aided Design (ICCAD). 1991.

109. *Patel D., Luba T.* Dependence set and functional decomposition of Boolean functions // International Journal of Electronics. 1993. Vol. 75, № 2.

110. *Peterson W. W.* On checking an adder. IBM J. Res. Dev. 1958. Vol. 2, № 2.

111. *Prihozhy A.* If-diagrams: theory and application // Proc. of the Int. Conference PATMOS (UCL, Begium). 1997.

112. *Rao T.R.N., Trehan A.* Single-error-correcting nonbinary arithmetic codes // IEEE Trans. 1970. Vol. IT-16, № 5.

113. *Rawski M., Jozwiak L., Nowicka M., Luba T.* Non-disjoint decomposition of Boolean function and its application in FPGA-oriented technology mapping // Proc. of the 23rd EUROMICRO Conference (Budapest, Hungry. Sept. 1–4). 1997.

114. *Rawski M., Jozwiak L., Luba T.* The inflence of the number of values in sub-functions on the effectiveness and efficiency of the functional decomposition // Proc. of the 25rd EUROMICRO Conference (Milan, Italy. Sept. 8–10). 1999. Vol. 1.

115. *Roth P.J., Karp R.M.* Minimization over boolean graphs // IBM Journal of research and development. 1962. Vol. 6, № 2.

116. *Sangiovanni-Vincentelli A., Gamal A.E., Rose J.* Synthesis methods for field Programmable Gate arrays // Proceedings of the IEEE. 1993. Vol. 81, № 7. July.

117. *Tsao-Wu N.T., Chang S.-H.* On the evaluation of minimum distance of binary arithmetic cyclic codes // IEEE Trans. 1969. Vol. IT-15, № 5.

118. *Venkateswaran R., Mazumder P.* A survey of DA techniques for PLD and FPGA based systems // INTEGRATION the VLSI Journal. 1994. № 17.

119. *Villa T., Kam T., Brayton R.K., Sangiovanni-Vincentelli A.* Synthesis of finite state machines Boston: Logic Optimization. Kluwer Academic Publishers, 1998.

120. *Villa T., Sangiovanni-Vincentelli A.* NOVA: State assignment for finite state machines for optional two-level logic inplementation // IEEE Trans. on CAD.1990. Vol. C-9, № 9.

121. *Wan W., Perkowski M.A.* A new approach to the decomposition of incompletely specified multioutput function based on graph coloring and local transformation and its.

122. *Wang T.L., Liu C.-K.* Weight-preserved single-error-correcting scheme for binary adders // IEEE Trans. 1974. Vol. C-23, № 10.

К главе 1

1. *Ачасова С.М., Бадман О.А.* Матричный метод синтеза комбинационных схем и логических преобразователей конечных автоматов // Изв. АН СССР. Техническая кибернетика. 1975. № 6.

2. *Ачасова С.М.* Алгоритмы синтеза автоматов на программируемых матрицах. М.: Радио и связь, 1987.

3. А. с. 1080134 (СССР). Устройство для сравнения кодов / В.И. Кочергин, С.В. Кульбицкий, А.М. Кривенцов // Открытия. Изобретения. 1981. № 10.

4. А. с. 868750 (СССР). Устройство для суммирования / А.Ф. Лекарев, В.И. Кочергин // Открытия. Изобретения. 1981. № 36.

5. А. с. 922730 (СССР). Устройство для сложения и вычитания / В.И. Кочергин, С.В. Кульбицкий, Л.В. Селиванова // Открытия. Изобретения. 1982. № 15.

6. А. с. 734681 (СССР). Одноразрядный сумматор / В.И. Кочергин // Открытия. Изобретения. 1980. № 18.

7. А. с. 739530 (СССР). Одноразрядный сумматор / В.И. Кочергин // Открытия. Изобретения. 1980. № 21.

8. А. с. 822183 (СССР). Устройство для суммирования / В.И. Кочергин // Открытия. Изобретения. 1981. № 14.

9. А. с. 826341 (СССР). Устройство для умножения / В.И. Кочергин, А.Ф. Лекарев // Открытия. Изобретения. 1981. № 16.

10. А. с. 922728 (СССР). Устройство для формирования сигнала переноса при суммировании многофазных кодов / В.И. Кочергин, А.Ф. Лекарев // Открытия. Изобретения. 1982. № 15.

11. А. с. 993264 (СССР). Многоразрядное устройство для сложения и вычитания / В.И. Кочергин, С.В. Кульбицкий // Открытия. Изобретения. 1983. № 4.

12. А. с. 543116 (СССР). Устройство управления реверсивным преобразователем / В.И. Кочергин, А.М. Кривенцов, Г.М. Данков // Открытия. Изобретения. 1977. №2.

13. *Байцер Б.* Архитектура вычислительных комплексов. М.: Мир, 1974.

14. *Глушков В.М., Цейтлин Г.М., Ющенко Е.Л.* Алгебра. Языки. Программирование. Киев: Наук. думка, 1978.

15. *Закревский А.Д.* Логический синтез каскадных схем. М.: Наука, 1981.

16. *Клейн Ф.* Элементарная математика с точки зрения высшей: В 2 т. Т. 1. Арифметика. Алгебра. Анализ / Пер. с нем.; под ред. В.Г. Болтянского. 4-е изд. М.: Наука, 1987.

17. *Кочергин В.И., Гоголин В.А.* Теория многомерных цифровых множеств в приложениях к системам энергоснабжения и электропривода // Матер. междунар. науч.-техн. конф. «Электромеханические преобразователи энергии». Томск: ТПУ, 2001.

18. *Математическая* энциклопедия / Гл. ред. И.М. Виноградов. М.: Советская энциклопедия, 1985. Т.5.

19. *Питерсон У., Уэлсон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.

20. *Савельев А.Я.* Арифметические и логические основы цифровых автоматов: Учебник. М.: Высш. школа, 1980.

21. *Самофалов К.Г., Корнейчук В.И., Тарасенко В.П.* Электронные цифровые вычислительные машины. Киев: Вища школа, 1976.

22. *Энциклопедия* кибернетики. Киев, 1975.

23. *Яблонский С.В.* О суперпозициях функций алгебры логики // Матем. сб. 1952. № 2.

24. *Яблонский С.В.* Функциональные построения в к-значной логике // Труды МИАН СССР. М.: Изд-во АН СССР, 1958. Т. 51.

25. *Яблонский С.В.* Введение в дискретную математику. М.: Наука, 1986.

26. *Mc.Cluskey E.J.* Minimization of boolen function // Bell System Techn. J. 1956. Vol. 35.

27. *Quine W.V.* The problem if simplifying of truth function // Amer. Math. Monthly. 1952.

К главе 2

1. А. с. 911514 (СССР). Устройство умножения / В.И. Кочергин // Открытия. Изобретения. 1982. № 9.

2. А. с. 868750 (СССР). Устройство для суммирования / А.Ф. Лекарев, В.И. Кочергин // Открытия. Изобретения. 1986. № 31.

3. А. с. 1170451 (СССР). Устройство для умножения числа на ряд констант / В.И. Кочергин, С.В. Кульбицкий, А.М. Кривенцов // Открытия. Изобретения. 1985. № 28.

4. А. с. 1252772 (СССР). Устройство для деления / В.И. Кочергин // Открытия. Изобретения. 1986. № 31.

5. А. с. 1291974 (СССР). Устройство для деления / В.И. Кочергин // Открытия. Изобретения. 1987. № 7.

6. А. с. 1080134 (СССР). Устройство для сравнения кодов / В.И. Кочергин, С.В. Кульбицкий, А.М. Кривенцов // Открытия. Изобретения. 1984. № 10.

7. А. с. 351326 (СССР). Управляемый делитель частоты / В.И. Кочергин, И.А. Подоплелов // Открытия. Изобретения. 1972. № 27.

8. А. с. 402157 (СССР). Реверсивный декадный счетчик импульсов / В.И. Кочергин, Н.С. Баранов // Открытия. Изобретения. 1973. № 41.

9. А. с. 421135 (СССР). Реверсивный десятичный счетчик / В.И. Кочергин, С.Д. Морозов, А.С. Кулешов // Открытия. Изобретения. 1974. № 11.

10. А. с. 441636 (СССР). Устройство управления реверсивным многофазным инвертором / В.И. Кочергин, С.Д. Морозов, А.И. Новоселов // Открытия. Изобретения. 1974. № 32

11. А. с. 514443 (СССР). Реверсивный делитель частоты / В.И.Кочергин // Открытия. Изобретения. 1976. № 18.

12. А. с. 758525 (СССР). Кольцевой счетчик / В.И. Кочергин, Н.С. Баранов, С.В. Кульбицкий // Открытия. Изобретения. 1980. № 31.

13. А. с. 834935 (СССР). Пересчетное устройство / В.И. Кочергин, Н.С. Баранов // Открытия. Изобретения. 1981. № 20.

14. А. с. 987681 (СССР). Регистр / В.И. Кочергин // Открытия. Изобретения. 1983. № 1.

15. А. с. 1398091 (СССР). Реверсивное счетное устройство / В.И. Кочергин // Открытия. Изобретения. 1988. № 19.

16. А. с. 1064276 (СССР). Преобразователь позиционного кода в двоичный код / В.И. Кочергин, С.В. Кульбицкий // Открытия. Изобретения. 1983. № 48.

17. *Беркс А., Гольдстейн Г., Нейман Дж.* Предварительное рассмотрение логической конструкции электронного вычислительного устройства // Киберн. сб. 1964. № 9.

18. *Букреев И.Н., Мансуров Б.М., Горячев В.И.* Микроэлектронные схемы цифровых устройств. М.: Сов. радио, 1975.

19. *Микроэлектроника* и однородные структуры для построения логических и вычислительных устройств / Прангишвили И.В., Абрамова Н.А., Бабичева Е.В., Игнатушенко В.В. М.: Наука, 1967.

20. *Проектирование* микроэлектронных цифровых устройств / Под ред. С.А. Майорова. М.: Сов. радио, 1977.

21. *Рабинер Л., Гоулд Б.* Теория и применение цифровой обработки сигналов. М.: Мир, 1978.

22. *Ричардс Р.К.* Арифметические операции на ЦВМ. М.: ИЛ, 1957.

23. *Храпченко В.М.* Методы ускорения арифметических операций, основанные на преобразовании многоразрядного кода // Вопросы радиоэлектроники. Сер. УП ЭВТ. 1965. Вып.8.

24. *Цифроаналоговый* электропривод на электродвигателях мощностью от 0,25 до 1,0 кВт, управляемый от микропроцессора. Теоретические и экспериментальные исследования / Науч. рук. НИР В.И. Кочергин. Отчет о НИР (промежуточный), № ГР 01.83.0. 038666. Томск: НПО «Полюс», 1983.

25. *Цифроаналоговый* электропривод на электродвигателях мощностью от 0,25 до 1,0 кВт, управляемый от микропроцессора. Теоретические и экспериментальные

исследования / Науч. рук. НИР В.И. Кочергин. Отчет о НИР (заключительный), № ГР 01.83.0. 038666. Томск : НПО «Полюс», 1985.

26. *Чирков М.К., Шауман А.М.* Основы функциональной структуры вычислительных машин. Л.: Изд-во ЛГУ, 1974.

27. *Шауман А.М.* Основы машинной арифметики. Л.: Изд-во Ленингр. ун-та, 1979.

28. *A simple* motion estimator for variable-reluctance motors / Harris WALTER d., Lan. Jeffrey H. // Conf. Rec. IEEE Ind. Appl. Soc. 23 rd Annu. Meet. Pittsburgh, Pa, Oct. 2–7, 1988. Pt 1. N.Y., 1988.

29. *Cappa M., Hamacher V.C.* An augment interactive array for high-speed binary division IEEE Trans. comput. 1973. Vol. 22, № 2.

30. *Jonson* counters even-and-cycle length // Electronic Engineer. 1971. № 9.

К главе 3

1. А. с. 543116 (СССР). Устройство управления реверсивным преобразователем / В.И. Кочергин, А.М. Кривенцов, Г.М. Данков // Открытия. Изобретения. 1977. №2.

2. А. с. 987681 (СССР). Регистр / В.И. Кочергин // Открытия. Изобретения. 1983. № 1.

3. А. с. 1228269 (СССР). Делитель-счетчик многофазного кода (его варианты) / В.И. Кочергин, С.В. Кульбицкий // Открытия. Изобретения. 1986. № 16.

4. А. с. 514443 (СССР). Реверсивный делитель частоты / В.И. Кочергин // Открытия. Изобретения. 1976. № 18.

5. *Брюхович Е.И.* О проблемах автоматического контроля в ЭВМ и контролеспособности позиционных счислений // Управляющие системы и машины. 1977. № 4.

6. *Кларк Дж. мл., Кейн Дж.* Кодирование с исправлением ошибок в системах цифровой связи. М.: Радио и связь, 1987.

7. *Нейман Дж., фон.* Теория самовоспроизводящихся автоматов. М.: Мир, 1971.

8. *Микроэлектроника* и однородные структуры для построения логических и вычислительных устройств / Прангишвили И.В., Абрамова Н.А., Бабичева Е.В., Игнатушенко В.В. М.: Наука, 1967.

9. С*агомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989.

10. *Самофалов К.Г, Корнейчук В.И., Тарасенко В.П.* Электронные цифровые вычислительные машины. Киев: Вища школа, 1976.

11. *Селлерс Ф., Сяо М.-Ю., Бирнсон Л.* Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972.

12. *Garter W.C., Schneider P.R.* Design of Dynamically checked Computers // JEID Congress. Edinburgh, Scjtland. 1968.

13. *Short R.A.* The attainment of reliable digital systems through the use of redundancy // A Survey, Computer Group News, 2, 2–27 (1968. March).

14. *Нейман Дж., фон.* Вероятностная логика и синтез надежных организмов из ненадежных элементов // Автоматы: Сб. М.: Мир, 1956. С. 100–108.

15. *Диллон Б., Синх Ч.* Инженерные методы обеспечения надежности систем. М.: Мир, 1984.

К главе 4

1. *Кочергин В.И., Белицкая Л. А.* Синтез суммирующих устройств в нетрадиционных двоичных кодах // Современная техника и технологии СТТ 2004: Труды X Междунар. науч.-практ. конф. студентов, аспирантов и молодых ученых. Томск: ТПУ, 2004.

2. *Кочергин В.И., Белицкая Л. А.* Синтез многовходовых устройств суммирования и вычитания, в которых сигнал переноса или займа формируется по входным и выходным составляющим его сигналов // Современная техника и технологии СТТ 2005: Труды XI Междунар. науч.-практ. конф. студентов, аспирантов и молодых ученых. Томск: ТПУ, 2005.

3. *Kochergin V. I., Belitskaya L.A.* Synthesis of the Multi-Input Adder with the Maximal Operating Speed. A Publication of the International Association of Science and Technology for Development – IASTED «Automation, Control and Information Technology - 2005». Novosibirsk, 2005.

4. *Кочергин В.И., Белицкая Л. А.* Синтез одноразрядного сумматора, не реагирующего на одиночные ошибки во входных сигналах операндов // Решетневские чтения: Труды IX науч. конф. с междунар. участием. Красноярск: СибГАУ, 2005 (10–12 нояб.).

5. *Кочергин В.И., Белицкая Л. А., Гоголин В. А.* Синтез контролеспособных устройств суммирования и вычитания систем управления электроприводов // Электромеханические преобразователи энергии: Труды XI Междунар. науч.-практ. конф. Томск: ТПУ, 2005.

6. *Кочергин В.И., Белицкая Л. А.* Синтез одноразрядного сумматора с исправлением одиночных ошибок // Электронные средства и системы управления: Труды III Междунар. науч.-практ. конф. Томск: ТУСУР, 2005 (12–14 окт.).

7. *Питерсон У.* Коды, исправляющие ошибки. М.: Мир, 1964.

К главе 6

1. *Fisher R.A.* The theory of confounding in factoial experiments in relation to the theory of group // Ann. Eugenics. 1942. Vol. 11. P. 341–353; Math. Rev. Vol. 4. P. 127.

2. *Нейман Дж., фон.* Вероятностная логика и синтез надежных организмов из ненадежных компонент // Автоматы. М.: Мир, 1956.

3. *Shannon C.E.* A mathematical theory of communication // The Bell System Theh. Journ. 1948. Vol. 27. P. 379–423.

4. *Golay M.E.* Notes on digital coding // Proc. IRE. 1949. Vol. 37. P. 657.

5. *Hamming R.W.* Error detecting and error correcting codes // Bell System Tech. J. 1950. Vol. 29. P. 147–160. [Рус. пер.: *Хемминг Р.В.* Коды с обнаружением и исправлением ошибок // Коды с обнаружением и исправлением ошибок. М.: ИЛ, 1956. С. 7–22].

6. *Berlekamp E.R.* Algebraic coding theory. N.Y., 1968. [Рус. пер.: *Берлекэмп Э.* Алгебраическая теория кодирования. М.: Мир, 1971].

7. *Зиновьев В.А., Леонтьев В.К.* О совершенных кодах // Проблемы передачи информации. Vol. 8. № 1. 1972.

8. *Tietäväinen A., Perko A.T.* The are unknow perect binary codes // Ann. Univ. Turku. Ser. A, I. 1971. № 1. P. 3 – 10.

9. *Peterson W.W.* Error – correcting codes // Cambridge: The M.I.T. Press. Mass. Math. Rev. 1961. Vol. 22. P. 12003. [Рус. пер.: *Питерсон У.* Коды, исправляющие ошибки. М.: Мир, 1964].

10. *Дадаев Ю.Г.* Арифметические коды, исправляющие ошибки. М.: Сов. радио, 1969.

Научное издание

КОЧЕРГИН Валерий Иванович

Теория многомерных цифро-векторных множеств

Редактор В.Г. Лихачева
Верстка С.И. Кочин

Лицензия ИД № 04617 от 24.04.01 Подписано в печать 17.03.06.
Формат 70x100 $^{1}/_{16}$. Бумага офсетная № 1. Гарнитура «Таймс». Печать офсетная.
Печ. л. 23, 75; уч-изд. л. 22; усл. печ. л. 21, 88. Тираж 100 экз. Заказ №

ОАО «Издательство ТГУ», 634029, Томск, ул. Никитина, 4
ОАО «Издательство Асиновское», г. Асино, ул. Проектная, 24.