

## Оглавление

Предисловие .....	5
Введение к части 2 .....	15
Глава 5 Основы конструктивной (алгоритмической) теории измерения.....	21
5.1. Эволюция понятия «измерение» в математике.....	21
5.2. Аксиомы Евдокса-Архимеда и Кантора.....	22
5.3. Проблема бесконечного в математике.....	23
5.4. Критика канторовской теории бесконечных множеств .....	25
5.5. Конструктивный подход к созданию математической теории измерения ....	28
5.6. «Индикаторная» модель измерения .....	30
5.7. Понятие оптимального алгоритма измерения .....	32
5.8. Классические алгоритмы измерения.....	35
5.9. Оптимальные $(n,k,0)$ -алгоритмы.....	40
5.10. Оптимальные $(n,k,1)$ -алгоритмы, приводящие к «арифметическому квадрату» .....	44
Глава 6 Принцип асимметрии измерения и фибоначчиевые алгоритмы измерения ...	53
6.1. Задача Баше-Менделеева .....	53
6.2. Принцип асимметрии измерения .....	57
6.3. Новая формулировка задачи Баше-Менделеева .....	59
6.4. Синтез оптимального фибоначчиевого алгоритма измерения.....	60
6.5. Пример «фибоначчиевого» алгоритма измерения .....	65
6.6. Основной результат алгоритмической теории измерения.....	68
6.7. Изоморфизм между алгоритмической теорией измерения и динамической теорией биологических популяций.....	72
6.8. Изоморфизм между алгоритмической теорией измерения и теорией позиционных систем счисления .....	79
6.9. Международное признание алгоритмической теории измерения .....	81
Глава 7 Эволюция систем счисления.....	85
7.1. Крупнейшее математическое открытие в истории математики.....	85
7.2. Основные вехи в истории систем счисления .....	88
7.3. Троичная симметричная система счисления и компьютер «Сетунь» .....	100
7.4. Система остаточных классов .....	109
Глава 8 Фибоначчиевые системы счисления .....	115
8.1. Тьюринг и числа Фибоначчи .....	115
8.2. $P$ -коды Фибоначчи.....	118

8.3. Особенности «фибоначчиевых» представлений.....	120
8.4. Диапазоны представления чисел в $p$ -коде Фибоначчи .....	123
8.5. Избыточность «фибоначчиевых» представлений.....	125
8.6. Удивительные аналогии между кодом Фибоначчи и генетическим кодом. ....	128
8.7. Счет по Фибоначчи.....	129
8.8. Элементы арифметики Фибоначчи .....	133
8.9. Выполнение логических операций с использованием базовых микроопераций.....	136
8.10. Фибоначчиево сложение и вычитание.....	137
8.11. Фибоначчиево умножение и деление .....	140
Глава 9 Коды золотой пропорции и «золотая» теория чисел.....	143
9.1. Система счисления Бергмана.....	143
9.2. Особенности представления чисел в системе Бергмана .....	145
9.3. Представление натуральных чисел .....	147
9.4. Новое конструктивное определение действительного числа.....	150
9.5. Элементы «золотой» теории чисел .....	154
9.6. $F$ - и $L$ -коды .....	158
Глава 10 Микропроцессоры Фибоначчи .....	167
10.1. Проблема контролеспособности и помехоустойчивости микропроцессоров, используемых в системах управления .....	167
10.2. О возможных причинах участвовавших катастроф при запуске российских и иностраных ракет.....	169
10.3. «Троянский конь» двоичной системы счисления и «фибоначчиевы» патенты.....	171
10.4. Особенности применения классических избыточных кодов в «компьютерных каналах» .....	178
10.5. Информационные основы микропроцессоров Фибоначчи.....	179
10.6. Преобразование чисел из традиционных систем счисления в код золотой пропорции.....	184
10.7. «Золотая» арифметика.....	190
10.8. Сравнение КФ и КЗП с классической двоичной системой.....	198
10.9. Контроль электронной памяти и регистров .....	201
10.10. Устройство для приведения КФ и КЗП к минимальной форме .....	205
10.11. Счетчик Фибоначчи для минимальной формы.....	209
10.12. Помехоустойчивое АЛУ Фибоначчи .....	221
10.13. Самоконтролирующийся «фибоначчиевый» модуль .....	224

10.14. «Золотые» аналого-цифровые и цифро-аналоговые преобразователи.....	227
Глава 11 «Золотая» троичная зеркально-симметричная арифметика.....	235
11.1. Троичное зеркально-симметричное представление .....	235
11.2. Троичные зеркально-симметричные $F$ - и $L$ -коды.....	241
11.3. Диапазон представления и избыточность троичного зеркально-симметричного представления .....	243
11.4. Троичное зеркально-симметричное сложение и вычитание .....	246
11.5. Троичное зеркально-симметричное умножение и деление .....	253
11.6. Матричный и конвейерный зеркально-симметричные сумматоры.....	258
11.7. Троичная логика и троичные элементы.....	262
11.8. Троичный зеркально-симметричный цифроаналоговый преобразователь	268
Глава 12 Перспективные идеи в развитии фибоначчьевого кодирования.....	273
12.1. Троичный код Фибоначчи.....	273
12.2. Троичная арифметика Фибоначчи .....	276
12.3. Новая теория кодирования, основанная на матрицах Фибоначчи.....	281
Послесловие ко второй части .....	297
Список литературы .....	307
Научная биография Алексея Стахова .....	315



## Предисловие

**Идея Гармонии и Золотого Сечения.** Каждый из нас наверняка задумывался над тем, почему Природа способна создавать такие удивительные гармоничные структуры, которые восхищают и радуют глаз. Почему художники, поэты, композиторы, архитекторы создают восхитительные произведения искусства из столетия в столетие. В чем же секрет и какие законы лежат в основе этих гармоничных созданий? Никто не сможет однозначно ответить на этот вопрос, но в этой книге сделана попытка приоткрыть завесу и рассказать об одной из тайн «гармонии мироздания» – «Золотом Сечении» или, как его еще называют, Золотой или Божественной Пропорции. Золотое сечение называется числом РНІ (Фи) в честь великого древнегреческого скульптора Фидия (Phidias), который использовал это число в своих скульптурах.

Научно-технический прогресс имеет длительную историю и прошел в своем историческом развитии несколько этапов (вавилонская и древнеегипетская культура, культура Древнего Китая и Древней Индии, древнегреческая культура, эпоха Средневековья, эпоха Возрождения, промышленная революция 18 в., великие научные открытия 19 в., научно-техническая революция 20 в.) и вошел в 21-й век, который открывает новую эпоху в истории человечества - эпоху Гармонии. Именно в античный период было сделано ряд выдающихся математических открытий, оказавших определяющее влияние на развитие материальной и духовной культуры и роль которых мы не всегда осознаем. К разряду таких открытий мы должны, прежде всего, отнести Вавилонскую 60-ричную систему счисления и открытый вавилонянами позиционный принцип представления чисел, лежащий в основе десятичной и двоичной систем счисления. В этот ряд мы должны поставить тригонометрию и геометрию Евклида, несоизмеримые отрезки, золотое сечение и Платоновы тела, начала теории чисел и теории измерения. И, хотя каждый из этих этапов имеет свою специфику, вместе с тем он обязательно включает содержание предшествующих этапов. В этом и состоит преемственность в развитии науки. Преемственность может осуществляться в различных формах. Одной из существенных форм ее выражения

являются фундаментальные научные идеи, которые пронизывают все этапы научно-технического прогресса и оказывают влияние на различные области науки, искусства, философии и техники. К разряду таких фундаментальных идей относится идея Гармонии, связанная с золотым сечением. По словам Б.Г. Кузнецова, исследователя творчества Альберта Эйнштейна, великий физик свято верил в то, что наука, физика в частности, всегда имела своей извечной фундаментальной целью "найти в лабиринте наблюдаемых фактов объективную гармонию". О глубокой вере выдающегося физика в существование универсальных законов гармонии мироздания свидетельствует и еще одно широко известное высказывание Эйнштейна: «Религиозность ученого состоит в восторженном преклонении перед законами гармонии».

**Высказывания Алексея Лосева и Иоганна Кеплера.** Какая главная идея лежала в основе древнегреческой науки? Подавляющее число исследователей дают следующий ответ: идея Гармонии, связанная с «золотым сечением». Как известно, в древнегреческой философии Гармония противостояла Хаосу и означала организованность Вселенной, Космоса. Выдающийся русский философ Алексей Лосев, исследователь эстетики античности и эпохи Возрождения, так оценивает основные достижения древних греков в этой области:

«Космос античным мыслителям периода зрелой классики представляется не просто некоей отвлеченной неопределенностью, (в таком случае он был бы только чистой мыслью), но совершенно живым и единым телом, содержащим в себе нерушимую цельность, несмотря на бесконечные различия всех его проявлений. С точки зрения Платона, да и вообще с точки зрения всей античной космологии мир представляет собой некое пропорциональное целое, подчиняющееся закону гармонического деления - золотого сечения (то есть, целое относится в нем к большей части, как большая часть к меньшей). Этому закону, кстати сказать, древние греки подчиняли и свои архитектурные сооружения. Их систему космических пропорций нередко в литературе изображают как курьезный результат безудержной и дикой фантазии. В такого рода объяснениях сквозит антинаучная беспомощность тех, кто это заявляет. Однако понять данный

историко-эстетический феномен можно только в связи с целостным пониманием истории, то есть, используя диалектико-материалистическое представление о культуре и ища ответа в особенностях античного общественного бытия».

В этом высказывании Алексей Лосев достаточно убедительно сформулировал «золотую» парадигму античной космологии. В ее основе лежат важнейшие идеи античной науки, которые в современной науке иногда трактуются как «курьезный результат безудержной и дикой фантазии». Прежде всего – это пифагорейская идея о числовой гармонии мироздания и космология Платона, основанная на Платоновых телах. Обратившись к геометрической структуре мироздания и арифметическим отношениям, выражающим гармонию, пифагорейцы предвосхитили возникновение математического естествознания, которое начало стремительно развиваться в 20-м веке. Идея Пифагора и Платона о всеобщей гармонии мироздания оказалась бессмертной.

Таким образом, в центре созданного древними греками математического учения о природе стояла «концепция гармонии», а сама математика древних греков и была «математикой гармонии» (“the mathematics of harmony”), которая непосредственно связана с золотым сечением - важнейшим математическим открытием античной науки в области гармонии.

А вот еще одно широко известное высказывание, касающееся золотого сечения. Оно принадлежит гениальному астроному Иоганну Кеплеру, автору трех знаменитых «Законов Кеплера». Свое восхищение золотым сечением Кеплер выразил в следующих словах:

«В геометрии существует два сокровища – теорема Пифагора и деление отрезка в крайнем и среднем отношении. Первое можно сравнить с ценностью золота, второе можно назвать драгоценным камнем».

Напомним, что старинная задача о делении отрезка в крайнем и среднем отношении, которая упоминается в этом высказывании, – это и есть золотое сечение!

Огромный интерес к этой проблеме в современной науке подтверждается достаточно внушительным перечнем книг по этой проблеме, опубликованных во второй половине 20 в. и начале 21 в. [1-50].

**Математическое учение о Природе древних греков.** Согласно мнению выдающегося американского историка математики Мориса Клайна [51], главный вклад древних греков, «оказавший решающее влияние на всю последующую культуру, состоял в том, что они взялись за изучение законов природы». Основной вывод, вытекающий из книги Мориса Клайна [51], состоит в том, что древние греки предложили новаторскую концепцию космоса, в котором все было подчинено математическим законам. Возникает вопрос: когда эта концепция была разработана? Ответ на этот вопрос также содержится в книге [51]:

«Греки преисполнились решимости доискаться до истин и, в частности, до истин о математических основах природы. Как следует приступить к поиску истин и как при этом гарантировать, что поиск действительно приводит к истинам? Греки предложили «план» такого поиска. Хотя он создавался постепенно на протяжении нескольких веков (VI-III вв. до н.э.), в истории науки расходятся во мнении относительно того, когда и кем этот план был впервые задуман, к III в. до н.э. план поиска истин был доведен до совершенства».

Таким образом, по мнению Клайна, новаторская концепция космоса, основанного на математических законах, была разработана древними греками в период с VI до III вв. до н.э. Но согласно утверждению А.Н. Колмогорова [52], в этот же период в Древней Греции «возникает математика как самостоятельная наука с ясным пониманием своеобразия ее метода и необходимости систематического развития ее основных понятий и предложений в достаточно общей форме».

Но тогда возникает вопрос: существовала ли какая-либо взаимосвязь между процессом создания математического учения о природе, что считается главным достижением древнегреческой науки, и процессом создания математики, которые протекали в Древней Греции в один и тот же период. Или это разные процессы? Оказывается, что такая связь, безусловно, существовала. Более того. Можно утверждать, что эти процессы фактически совпадали, то есть, математика, созданная древними греками, и их учение о природе, основанное на математических принципах, - это одно и то же. И наиболее ярким воплощением



процесса «Математизации Гармонии» являются «Начала» Евклида, написанные в III в. до н.э.

**Гипотеза Прокла.** Греческий философ Прокл Диадох (412-485) высказал необычную гипотезу, касающуюся «Начал» Евклида. Среди математических сочинений Прокла наиболее известным является его «Комментарий к первой книге «Начал» Евклида». В этом Комментарии он выдвигает следующую необычную гипотезу. Суть «гипотезы Прокла» состоит в следующем. Как известно, XIII-я, то есть, заключительная книга «Начал», посвящена изложению теории пяти правильных многогранников, которые играли главенствующую роль в «Космологии Платона» и в современной науке известны под названием Платоновых тел. Именно на это обстоятельство и обращает внимание Прокл. Как подчеркивает Эдуард Сороко [13], по мнению Прокла, Евклид «создавал «Начала» якобы не с целью изложения геометрии как таковой, а чтобы дать полную систематизированную теорию построения пяти «Платоновых тел», попутно осветив некоторые новейшие достижения математики».

Именно для решения этой задачи (в частности, для создания геометрической теории додекаэдра) Евклид уже в Книге II вводит задачу о делении отрезка в крайнем и среднем отношении («золотое сечение»), которая затем встречается и в других Книгах «Начал», в частности, в Книге XIII.

«Гипотеза Прокла» приводит к выводу, который может оказаться неожиданным для многих математиков. Оказывается, из «Начал» Евклида берут свое начало два направления математической науки - «Классическая Математика», позаимствовавшая в «Началах», аксиоматический подход, теорию чисел, теорию иррациональностей и геометрические аксиомы, и «Математика Гармонии», которая акцентирует свое внимание не на «аксиоматическом подходе», а на геометрической «задаче о делении отрезка в крайнем и среднем отношении» («золотом сечении») и на теории Платоновых тел, изложенной в Книге XIII «Начал» Евклида.

**Введение термина «математика гармонии».** В конце 20 в. для обозначения математического учения о природе, созданного древними греками,

был введен термин “the mathematics of harmony” (математика гармонии). Следует отметить, что этот термин выбран очень удачно, потому что он отражал главную идею античной науки – «Математизация Гармонии». Впервые этот термин был введен в небольшой статье “Harmony of spheres”, помещенной в The Oxford dictionary of philosophy [53]. В этой статье понятие “the mathematics of harmony” («математика гармонии») ассоциируется с «гармонией сфер», которая называлась также «гармонией мира» (harmonica mundi) или мировой музыкой (лат. musica mundana). Гармония сфер представляет собой античное и средневековое учение о музыкально-математическом устройстве космоса, восходящее к пифагорейской и платонической философской традиции.

Еще одно упоминание о «математике гармонии» применительно к древнегреческой математике мы встречаем в книге Vladimir Dimitrov. A new kind of social science. Study of self-organization of human dynamics, опубликованной в 2005 г. [54]. Важно подчеркнуть, что в книге [54] понятие “the mathematics of harmony” («математика гармонии») непосредственно ассоциируется с «золотым сечением» - важнейшим математическим открытием античной науки в области гармонии, которое в тот период называлось «делением отрезка в крайнем и среднем отношении».

Как вытекает из работ [53,54], в развитии «Математики Гармонии» в течение нескольких тысячелетий принимали участие выдающиеся мыслители, ученые и математики: Пифагор, Платон, Евклид, Фибоначчи, Пачоли, Кеплер, Кассини, Бине, Люка, Клейн, а в 20-м веке – известные математики Коксетер, Воробьев, Хоггатт и Вайда. И мы никак не можем игнорировать этот исторический факт.

**Числа Фибоначчи.** С золотым сечением тесно связаны числа Фибоначчи, открытые в 13 веке итальянским математиком Леонардо из Пизы, по прозвищу Фибоначчи. Они составляют числовой ряд, начинающийся с двух единиц, в котором каждое последующее число является суммой двух предыдущих. Отношение соседних чисел ряда Фибоначчи в пределе стремится к золотому сечению. Математическая теория чисел Фибоначчи получила дальнейшее развитие

в работах французских математиков 19-го века Бине («формулы Бине») и Люка («числа Люка»). Как упоминалось, во второй половине 20-го века эта теория получила развитие в работах канадского геометра Дональда Коксетера [1], советского математика Николая Воробьева [2], американского математика Вернера Хоггатта [3] и английского математика Стефана Вайды [4]. Развитие этого направления, в конечном итоге, привело к возникновению «Математики Гармонии» [47] - нового междисциплинарного направления современной науки, которое имеет отношение к современной математике, компьютерной науке, экономике, а также ко всему теоретическому естествознанию. Работы известных математиков Коксетера, Воробьева, Хоггатта и Вайды, а также исследования математиков-фибоначчистов, членов Американской Фибоначчи-Ассоциации, стали началом процесса «Гармонизации Математики», который активно продолжается и в 21-м веке. И этот процесс подтверждается огромным количеством книг в области «золотого сечения» и чисел Фибоначчи, опубликованных во второй половине 20 в. и начале 21 в. [1-50].

**Источники настоящей книги.** Настоящая книга является итогом более чем 40-летних исследований автора, связанных с «золотым сечением» и числами Фибоначчи, и основана на материале пяти книг, опубликованных автором в различные периоды своей научной деятельности. Первая книга «Введение в алгоритмическую теорию измерения» опубликована в 1977 г. [9] и посвящена изложению оригинальной математической теории измерения, касающейся теоретической метрологии и оснований математики. Фибоначчиевы алгоритмы измерения, синтезированные в рамках алгоритмической теории измерения [9], лежат в основе  $p$ -кодов Фибоначчи – новых способов позиционного представления натуральных чисел, которые являются обобщением классической двоичной системы.

Вторая публикация – это брошюра «Алгоритмическая теория измерения» [10], опубликованная издательством «Знание» в престижной серии «Математика и кибернетика» в 1979 г.

Третья книга «Коды золотой пропорции» опубликована в 1984 г. [12] и посвящена изложению нового направления в теории систем счисления – систем счисления с иррациональными основаниями, называемых также кодами золотой пропорции. Системы счисления с иррациональными основаниями являются новыми способами позиционного представления действительных чисел. Они переворачивают наши традиционные представления о системах счисления и могут быть положены в основу «золотой» теории чисел [55]. Коды Фибоначчи и коды золотой пропорции могут быть положены в основу нового направления в компьютерной науке – компьютеров Фибоначчи как нового направления в компьютерной технике, направленного на повышение информационной надежности компьютеров и защищенных 65 патентами США, Японии, Англии, Франции, ФРГ, Канады и др. стран.

Четвертая книга «Код да Винчи и ряды Фибоначчи», написанная в соавторстве с Анной Слученковой и Игорем Щербаковым, опубликована в 2006 г. [43] и представляет собой популярное изложение современной теории и приложений золотого сечения, чисел Фибоначчи и Платоновых тел.

Пятая книга “The Mathematics of Harmony. From Euclid to Contemporary Mathematics and Computer Science” опубликована в 2009 г. [47] и посвящена изложению «математики гармонии» – нового междисциплинарного направления современной науки, которое, по мнению академика Юрия Митропольского, представляет собой «большой теоретический вклад в развитие, прежде всего, «элементарной математики». С другой стороны, математика гармонии представляет собой «золотую парадигму» современной науки и отражает важнейшую тенденцию в современной науке – возрождение «гармонических идей» Пифагора, Платона и Евклида.

**Структура и цель книги.** Книга состоит из трех частей:

Часть 1. Золотое сечение, числа Фибоначчи, и Платоновы тела в истории науки и культуры.

Часть 2. Коды Фибоначчи и золотой пропорции как альтернатива классической двоичной системе счисления

### Часть 3. Математика гармонии как «золотая» парадигма современной науки

Главная цель книги – привлечь внимание широкой научной общественности и педагогических кругов к «математике гармонии» как новому типу «элементарной математики», представляющей интерес для современного математического образования, и как «золотой» парадигме современной науки, представляющей интерес для всей науки в целом, в том числе для информатики.

Книга написана популярно и рассчитана на широкий круг читателей, включая школьников, студентов, учителей школ, ученых различных специализаций, интересующихся историей математики, Платоновыми телами, золотым сечением, числами Фибоначчи и их приложениями в современной науке.

**Благодарности.** Заканчивая экскурс в предысторию возникновения этой книги, автор хотел бы поблагодарить всех, кто оказывал и оказывает огромную поддержку автору в развитии данного научного направления. Прежде всего, чувства глубокой благодарности автор испытывает к своему учителю профессору Александру Андреевичу Волкову (1924-2008), научному руководителю кандидатской (1966) и докторской (1972) диссертаций, а также выдающемуся украинскому математику академику Юрию Алексеевичу Митропольскому (1917-2008), который оказал огромную поддержку автору в развитии данного научного направления и способствовал публикации статей автора в академических изданиях Украины. И, конечно, огромная благодарность соратникам автора, членам Международного Клуба Золотого Сечения Эдуарду Сороко, Олегу Боднару, Сергею Петухову, Григорию Мартыненко, Самуилу Арансону, Борису Розину, Николаю Семенюте, Сергею Абачиеву, Валериану Владимирову, Виктору Цветкову, Анатолию Харитонову, Сергею Якушко, Олегу Когновицкому, Александру Южанникову, Анатолию Коновалову, Михаилу Быстрову, Александру Волошинову, Ирине Крючковой, Александру Иванусу, Леониду Тимошенко, Елене Терешиной, Ивану Райлян, Алексею Борисенко, Денису Клещеву, Юрию Черепяхину, Александру Чечуку, Юрию Цымбалисту, Татьяне Егоровой-Гудковой и многим другим. Автору было нелегко устанавливать новые научные контакты после переезда в Канаду в 2004 г. И автор благодарен зарубежным ученым –

аргентинскому математику Вере Шпинадель, американскому математику Джею Капраффу, чилийскому философу Дарио Саласу Соммеру, английскому физику Мохаммеду Ель-Нашию, немецкому ученому Волкмару Вейсу, канадским ученым Вадиму Геворкову и Льву Киришьяну, за поддержку научных исследований автора. Особая благодарность американскому философу проф. Скотту Олсену, одному из ведущих американских ученых в области золотого сечения, автору замечательной книги “The Golden Section. Nature’s Greatest Secret” (2006) [43]. Дружба с проф. Олсеном началась с 2005 г., когда Скотт Олсен приехал в Канаду специально для знакомства с автором настоящей книги. Проф. Скотт Олсен сыграл большую роль в публикации англоязычной книги автора “The Mathematics of Harmony. From Euclid to Contemporary Mathematics and Computer Science” (2010) [47]. В 2010 г. он принял участие в работе Международного Конгресса по Математике Гармонии (Одесса, 8-10 октября 2010 г.), а в августе 2011 г. он еще раз посетил Канаду, чтобы наметить перспективы совместной научной работы, обсудить с автором свою новую книгу и поддержать автора в связи с тяжелой болезнью супруги.

За постоянную поддержку и создание прекрасных условий для научного творчества особую благодарность и глубокие чувства любви и признательности автор выражает членам своей семьи, прежде всего, супруге Антонине, с которой прожито в любви и дружбе более 50 лет, и дочери Анне, без которой наша жизнь в Канаде была бы просто невозможной. И без их помощи, заботы и моральной поддержки эта книга никогда не была бы написана.

## Введение к части 2

**Математика в ее историческом развитии.** С чего начиналась математика? Каковы основные периоды в ее развитии? Ответ на этот далеко не простой вопрос мы находим в книге выдающегося советского математика Андрея Николаевича Колмогорова «Математика в ее историческом развитии» (1991) [52]. Колмогоров пишет:

«Ясное понимание самостоятельного положения математики как особой науки, имеющей собственный предмет и метод, стало возможным только после накопления достаточно большого фактического материала и возникло впервые в Древней Греции в 6-5 вв. до н.э. Развитие математики до этого времени естественно отнести к периоду зарождения математики, а к 6-5 вв. до н.э. приурочить начало периода элементарной математики. В течение этих двух первых периодов математического исследования имеют дело почти исключительно с весьма ограниченным запасом основных понятий, возникших еще на очень ранних ступенях исторического развития в связи с самыми простыми запросами хозяйственной жизни, сводившимися к счету предметов, измерению количества продуктов, площадей земельных участков, определению размеров отдельных частей архитектурных сооружений, измерению времени, коммерческим расчетам и т.п.».

В этой цитате, взятой из упомянутой книги, выделено два важнейших периода в развитии математики на этапе ее становления как самостоятельной науки: этап зарождения математики (догреческий период) и этап элементарной математики (от греческой математики 6-5 вв. до н.э. и до начала 17 в.), и обращено особое внимание на две главные практические задачи, которые стимулировали развитие математики на этапе ее зарождения. Это – задача счета и задача измерения.

**Поместные или позиционные системы счисления.** Именно на этапе зарождения математики было сделано несколько выдающихся открытий, которые оказали огромное влияние на развитие не только математики, но и всей

материальной культуры. Одним из них является открытие поместного или позиционного принципа представления чисел. Считается, что это открытие было сделано вавилонянами и воплощено ими в Вавилонской 60-ричной системе счисления. Системам счисления уделялось особое внимание на этапе зарождения математики, а их создание стало общим делом всего человечества. Кроме Вавилонской 60-ричной системы, на этапе зарождения математики была изобретена еще одна интересная система счисления. Речь идет о древнеегипетской системе счисления. Система счисления древних египтян была десятичной, но непозиционной. При этом главным достижением древних египтян в этой области стало открытие так называемого «метода удвоения». Этот метод лежит в основе оригинальных способов умножения и деления чисел, которые являются прообразом современных методов двоичного умножения и деления, используемых в современных компьютерах. Можно без преувеличения сказать, что именно эти два математических открытия в области систем счисления (позиционный принцип и метод удвоения) оказали затем огромное влияние на развитие не только математики, но и информатики. Открытый вавилонянами позиционный принцип лежит в основе всех наиболее известных систем счисления, в частности десятичной, 12-ричной, 20-ричной системы Майя, наконец, двоичной системы, лежащей в основе информатики, развитие которой, начиная со второй половины 20 в., стало едва ли не главным катализатором современного научно-технического прогресса.

**«Ренессанс» в теории систем счисления.** К сожалению, разработка теории систем счисления, по выражению Козьмы Пруткина, «не входит в круг понятий» современной математики. Поэтому в этой области современная математика не намного ушла вперед по сравнению с периодом своего зарождения. Однако, в связи с созданием компьютеров и информационных технологий во второй половине 20 в. неожиданно проявился огромный интерес к новым способам представления чисел и новым компьютерным арифметикам. При этом возник очень интересный аспект этой проблемы с исторической точки зрения. Спустя 4 тысячелетия после изобретения вавилонянами позиционного принципа, в области систем счисления



возник период своеобразного «ренессанса». Благодаря усилиям, прежде всего, специалистов в области компьютеров, математика как бы вновь возвратилась к периоду своего зарождения, когда именно системы счисления определяли содержание и сущность математики. Но ведь период зарождения математики, по мнению многих историков математики, считается чрезвычайно важным для развития математики. Именно в этот период были заложены основы такого важнейшего математического понятия как натуральное число и начала создаваться теория чисел, которая по праву называется «царицей математики». Но тогда вполне резонным является постановка следующего вопроса: а не могут ли современные системы счисления, созданные для удовлетворения утилитарных потребностей компьютерной техники, повлиять на развитие самого понятия числа и теории чисел и таким путем оказать влияние не только на развитие компьютерной науки, но и всей математики. Поиск ответа на этот вопрос и является одной из задач части 2 настоящей книги.

**Система счисления Бергмана.** В 1957 г. в математике произошло событие, которое, к сожалению, не было замечено математиками. Юный американский математик Джордж Бергман в виде увлекательной игры предложил новый способ позиционного представления чисел, названный им системой счисления с иррациональным основанием [55]. Суть нового метода представления чисел состояла в том, что основанием данной системы счисления является знаменитая «золотая пропорция» - уникальное иррациональное число  $\Phi = \frac{1+\sqrt{5}}{2}$ , которое было одним из главных выразителей гармонии Мироздания в древнегреческой науке и математике. Благодаря «системе Бергмана» античная идея Гармонии вошла в теорию чисел! Открытие американского вундеркинда, которому в тот момент было всего лишь 12 лет (случай беспрецедентный в математике!) по праву считается одним из наиболее крупных современных математических открытий в области систем счисления. Это открытие переворачивает наши представления не только о системах счисления, но и о соотношении между рациональными и иррациональными числами и таким образом затрагивает основания математики.

Идея систем счисления с иррациональными основаниями была подхвачена автором настоящей книги. В 1980 г. автором были предложены так называемые коды золотой пропорции [56], которые основаны на понятии золотых  $p$ -пропорций и являются широким обобщением системы Бергмана. Основы теории кодов золотой  $p$ -пропорции изложены в книге автора «Коды золотой пропорции» (1984) [12].

Но одновременно с новыми математическими результатами в области систем счисления [55,56] в современной науке были получены новые результаты в развитии математической теории измерения, которая восходит в своих началах к открытию пифагорейцами несоизмеримых отрезков и к «Началам» Евклида. Речь идет о разработке так называемой конструктивной (алгоритмической) теории измерения. Эта теория была создана в середине 60-х годов 20 в., как теория оптимальных алгоритмов аналого-цифрового преобразования [57,58]. Термин «алгоритмическая теория измерения» был введен автором настоящей книги в 1973 г. Основы этой теории изложены в книге автора «Введение в алгоритмическую теорию измерения» (1977) [9].

Основной методологической идеей алгоритмической теории измерения является так называемый принцип асимметрии измерения [58], который был сформулирован автором при изучении задачи Баше-Менделеева – исторически первой оптимизационной задачей в области теории измерения. Наиболее неожиданным результатом алгоритмической теории измерения, вытекающим из принципа асимметрии измерения, стали так называемые фибоначиевые алгоритмы измерения. Именно при решении этой задачи были введены  $p$ -числа Фибоначчи, которые, как оказалось (Джордж Пойа), могут быть также получены из треугольника Паскаля путем вычисления его «диагональные сумм».

Фибоначчиевые алгоритмы измерения привлекли к открытию так называемых  $p$ -кодов Фибоначчи [59-67] – новых способов позиционного представления натуральных чисел, которые являются обобщением классической двоичной системы счисления. Весами разрядов в  $p$ -кодах Фибоначчи являются  $p$ -числа Фибоначчи. Понятие  $p$ -кодов Фибоначчи является обобщением классической двоичной системы счисления. Количество  $p$ -кодов Фибоначчи

теоретически бесконечно, так как каждому  $p \in \{0,1,2,3,\dots\}$  соответствует свой  $p$ -код Фибоначчи.

Коды золотой  $p$ -пропорции [56] вместе с  $p$ -кодами Фибоначчи [59-67] и стали основой новой цифровой метрологии [63,64] и новых компьютерных арифметик – арифметики Фибоначчи и «золотой» арифметики, которые были положена в основу концепции компьютеров и микропроцессоров Фибоначчи [68] как нового направления в повышении информационной надежности компьютеров и микропроцессоров.

В 70-е и 80-е годы 20 в. это направление оказалось в центре внимания советской компьютерной науки и привлекло внимание не только советских научных деятелей, но и государственных деятелей. Было осуществлено широкое патентование советских изобретений по направлению «компьютеры Фибоначчи» за рубежом [69-82]. Более 60 патентов США, Японии, Англии, Франции, ФРГ, Канады и других стран являются официальными юридическими документами, которые подтверждают приоритет советской науки (и приоритет автора настоящей книги) в этом научном направлении.

Наконец, еще одним оригинальным результатом в этой области является так называемая троичная зеркально-симметричная арифметика. Такая арифметика была разработана автором в конце 20 в. и описана в статье A.P. Stakhov. Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic [83], опубликованной в 2002 г. в одном из наиболее престижных компьютерных журналов - «The Computer Journal». Статья вызвала большой резонанс в западной, в частности, американской науке. Первым, кто поздравил автора с этой публикацией, стал выдающийся американский ученый Дональд Кнут, автор книги «Искусство программирования», научного бестселлера 20 в.

Представляет также интерес разработанная автором новая теория кодирования, основанная на матрицах Фибоначчи [84]. Эта теория описана в статье Stakhov A. Fibonacci matrices, a generalization of the "Cassini formula", and a new coding theory, опубликованной в 2006 г. в журнале «Chaos, Solitons & Fractals» [85].

Часть 2 трехтомной монографии «Основы математики гармонии и ее приложения» посвящена изложению указанных выше результатов. Эта часть монографии состоит из 7 глав:

Глава 5. Основы конструктивной (алгоритмической) теории измерения

Глава 6. Принцип асимметрии измерения и фибоначчиевые алгоритмы измерения

Глава 7. Эволюция систем счисления

Глава 8. Фибоначчиевые системы счисления

Глава 9. Коды золотой пропорции и «золотая» теория чисел

Глава 10. Микропроцессоры Фибоначчи

Глава 11. «Золотая» троичная зеркально-симметричная арифметика

Глава 12. Перспективные идеи в развитии фибоначчиевого кодирования

Главная цель 2-й части книги «Основы математика гармонии и ее приложения» - дать ответ на вопрос: «Существует ли в современной информатике альтернатива классической двоичной системе счисления?». Эта часть посвящена изложению новых прикладных математических теорий: алгоритмической теории измерения, теории систем счисления с иррациональными основаниями (кодов Фибоначчи и золотой пропорции), троичной зеркально-симметричной арифметики и новой теории кодирования, основанной на матрицах Фибоначчи. Эти теории представляют интерес не только для информатики, но и для математического образования. Они могут быть использованы при проектировании компьютерных систем и микропроцессоров повышенной информационной надежности.

## Глава 5

# ОСНОВЫ КОНСТРУКТИВНОЙ (АЛГОРИТМИЧЕСКОЙ) ТЕОРИИ ИЗМЕРЕНИЯ

### 5.1. Эволюция понятия «измерение» в математике

Что такое «измерение»? В Большой Советской Энциклопедии мы находим следующее определение этого понятия:

«Измерение – операция, посредством которой определяется отношение одной (измеряемой) величины к другой однородной величине (принимаемой за единицу измерения); число, выражающее такое отношение, называется численным значением измеряемой величины».

Измерение является важным способом количественного познания объективного мира. Великому русскому ученому Д.И. Менделееву, создателю Периодической системы химических элементов, принадлежат следующие замечательные слова: «Наука начинается там, где начинают измерять. Точная наука немыслима без меры».

Проблема измерения играет такую же значительную роль в математике, как и в других областях науки, в частности в технике, физике и других «точных» науках.

Проследим теперь эволюцию понятия «измерения» в математике [9]. Как известно, первой "теорией измерений" был свод правил, которыми пользовались древнеегипетские землемеры. От этого свода правил, как свидетельствуют древние греки, берет начало геометрия, обязанная своим появлением (и названием) задаче об "измерении земли". Однако уже в Древней Греции происходит разбиение проблем измерения на прикладные, которые относятся к "логистике", и фундаментальные, относящиеся к геометрии; последние и становятся в центре античной математики. Наука об измерении развивается в этот период преимущественно как математическая теория. Главным математическим достижением этого периода является открытие несоизмеримых отрезков, сделанное

в научной школе Пифагора. Это открытие вызвало первый в истории математики кризис в ее основаниях и привело к введению иррациональных чисел – второго (после натуральных чисел) фундаментального понятия математики. Это открытие привело к формулировке «метода исчерпывания» Евдокса и «аксиомы измерения» (см. ниже), к которым в своих истоках восходит теория чисел, интегральное и дифференциальное исчисление.

## 5.2. Аксиомы Евдокса-Архимеда и Кантора

Для преодоления кризиса в основаниях античной математики, связанного с открытием «несоизмеримых отрезков», выдающийся геометр Евдокс предложил "метод исчерпывания", с помощью которого он построил остроумную теорию отношений, лежащую в основе античной теории континуума. "Метод исчерпывания" сыграл выдающуюся роль в развитии математики. Будучи прообразом интегрального исчисления, "метод исчерпывания" позволял античным математикам решать задачи вычисления объема пирамиды, конуса, шара. В современной математике метод исчерпывания находит свое отражение в аксиоме Евдокса-Архимеда, называемой также аксиомой измерения.

Теория измерения геометрических величин, восходящая к «несоизмеримым отрезкам», основывается на группе аксиом, называемых аксиомами непрерывности [86], которые включают в себя аксиомы Евдокса-Архимеда и Кантора или аксиому Дедекинда.

**Аксиома Евдокса-Архимеда («аксиома измерения»):** Для любых двух отрезков  $A$  и  $B$  (Рис.5.1) можно найти такое натуральное число  $n$ , чтобы

$$nB > A. \tag{5.1}$$

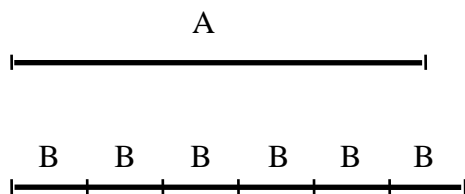


Рис.5.1. Аксиома Евдокса-Архимеда

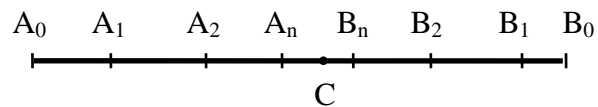


Рис.5.2. Аксиома Кантора

**Аксиома Кантора (о «стягивающихся отрезках»):** Если задана бесконечная последовательность отрезков  $A_0B_0, A_1B_1, A_2B_2, \dots, A_nB_n, \dots$  (Рис.5.2), «вложенных» друг в друга, то есть, каждый отрезок является частью предыдущего, тогда существует по крайней мере одна точка  $C$ , общая для всех отрезков.

Главным результатом теории геометрических величин является доказательство существования и единственности решения  $q$  «основного уравнения измерения»:

$$Q = qV, \quad (5.2)$$

где  $V$  - единица измерения;  $Q$  – измеряемая величина и  $q$  – результат измерения.

Несмотря на кажущуюся простоту сформулированных выше аксиом и всей математической теории измерения, она, тем не менее, является продуктом более чем двухтысячелетнего периода в развитии математики и содержит в себе ряд глубоких математических идей и понятий. Прежде всего, необходимо подчеркнуть, что "метод исчерпывания" Евдокса и вытекающая из него аксиома измерения (Рис.5.1) имеют практическое (эмпирическое) происхождение; они были позаимствованы древнегреческими математиками в практике измерений. В частности, "метод исчерпывания" является математической моделью процессов измерения объемов жидкостей и сыпучих тел путем "исчерпывания"; аксиома измерения, в свою очередь, концентрирует тысячелетний опыт человека, задолго до возникновения аксиоматического метода в математике миллиарды раз измерявшего расстояния, площади и временные интервалы, и представляет собой сжатую формулировку алгоритма измерения отрезка  $A$  с помощью отрезка  $B$ .

Суть этого алгоритма состоит в последовательном откладывании отрезка  $B$  на отрезке  $A$  и подсчете числа отрезков  $B$ , укладываемых на отрезке  $A$ . В современной практике измерений такой метод измерения называется «алгоритмом счета».

### **5.3. Проблема бесконечного в математике**

Аксиома Кантора (Рис.5.2) содержит в себе еще одно удивительное творение математической мысли - абстракцию актуальной бесконечности. Именно

такое представление о бесконечном лежит в основе канторовской теории бесконечных множеств. Представление об актуальной бесконечности в качестве образца канторовского (теоретико-множественного) стиля математического мышления было подвергнуто резкой критике со стороны представителей так называемого конструктивного подхода, который возник в математике 20-го века с целью преодоления кризиса в математике. С одной стороны, как подчеркивает Шанин Н.А. [87], такой подход обнаруживает глубокий разрыв с «данными экспериментального исследования природы»; с другой стороны, по меткому выражению А.А. Маркова [88], «мыслить себе бесконечный, т.е, никогда не завершаемый процесс как завершённый не удастся без грубого насилия над разумом, отвергающим такие противоречивые фантазии». Еще раньше эту же мысль другими словами выразил Д.Гильберт (известный своими «финитными» установками), который рассуждая о конечном и бесконечном, пришел к следующему заключению [89]:

«Мы хотим из всех наших рассуждений сделать некоторое резюме о бесконечности – общий вывод таков: бесконечное нигде не реализуется. Его нет в природе, и оно недопустимо как основа нашего разумного мышления, - здесь мы имеем замечательную гармонию между бытием и мышлением... Оперирование с бесконечным может стать надежным только через конечное».

Обнаруженные в начале 20-го века парадоксы, противоречия в канторовской теории бесконечных множеств значительно пошатнули устои математики. Были предприняты различные попытки укрепить их. Наиболее радикальной из них является конструктивное направление в обосновании математики [87,88], которое полностью исключает из рассмотрения абстракцию актуальной бесконечности и использует гораздо более «скромную» абстракцию бесконечного, называемую абстракцией потенциальной осуществимости.

Наиболее ярко противоречие между потенциальной и актуальной бесконечностями проявляют себя в математической теории измерения при анализе аксиомы Евдокса-Архимеда (Рис.5.1) и аксиомы Кантора (Рис.5.2). Для пояснения этого противоречия рассмотрим еще раз «основное уравнение измерения» (5.2). Идея его доказательства состоит в следующем. С помощью аксиомы Евдокса-



Архимеда из единицы измерения  $V$  по определенным правилам, называемым алгоритмом измерения, формируется некоторая последовательность "стягивающихся" отрезков, которые сравниваются с измеряемым отрезком  $Q$ ; при устремлении этого процесса в бесконечность на основании аксиомы Кантора для любого  $Q$  при заданном  $V$  всегда найдется такой измеряющий  $Q$  отрезок, сформированный из  $V$ , который "абсолютно точно" совпадет с  $Q$ . Наиболее существенным в этом доказательстве является вытекающее из аксиомы Кантора представление об измерении, как о процессе, завершающемся за бесконечное время. Таким образом, на начальном этапе доказательства уравнения (5.2) мы используем понятие потенциальной бесконечности (аксиома Евдокса-Архимеда), а на завершающем этапе мы «перепрыгиваем» через это понятие и используем понятие актуальной бесконечности (аксиома Кантора).

В этой связи уместно привести следующую цитату из книги [9]: «Уместно обратить внимание на внутреннюю противоречивость (в диалектическом смысле) теоретико-множественной теории измерения (и как следствие теории действительных чисел), допускающей в своих исходных положениях (аксиомы непрерывности) сосуществование диалектически противоречивых представлений о бесконечном (актуальной, "статической", завершённой бесконечности – в аксиомах Кантора, (и Дедекинда) и бесконечности потенциальной, "становящейся", незавершённой – в аксиоме Архимеда)».

То есть, существующая теория измерения и вытекающая из нее теория действительных чисел, основанные на аксиоме Кантора, являются внутренне противоречивыми и такие теории не могут быть положены в основание математики! Иначе и вся математика становится внутренне противоречивой теорией. Что на самом деле и случилось в математике в начале 20-го века, когда были обнаружены противоречия в канторовской теории бесконечных множеств. Удивительно, что такая простая идея до книги [9] не была замечена математиками.

## 5.4. Критика канторовской теории бесконечных множеств

**"Infinitum Actu Non Datur"**. Как известно, математика превратилась в дедуктивную науку в Древней Греции. Уже в VI в. до н.э. греческие философы

разрабатывали проблему бесконечного и связанную с ней проблему непрерывного и дискретного. Большое внимание развитию этого понятия уделял Аристотель. Он был первым, кто категорически начал возражать против использования актуальной бесконечности в науке, ссылаясь на то, что, зная способы счета конечного числа объектов, нельзя эти способы распространять на бесконечные множества. В своей «Физике» Аристотель утверждал:

«Остается альтернатива, согласно которой бесконечное имеет потенциальное существование... Актуально бесконечное не существует».

По мнению Аристотеля, актуальная бесконечность не нужна математике. Аристотелю принадлежит знаменитый тезис "Infinitum Actu Non Datur", что в переводе с латинского означает утверждение о невозможности существования логических или математических (т.е. всего лишь мыслимых, а не существующих в природе) актуально-бесконечных объектов.

**Критика канторовской теории множеств в 19 в. и начале 20 в.**  
Канторовская теория бесконечных множеств вызвала бурю протестов уже в 19 в. Детальный анализ критики этой теории проведен в главе «Изгнание из рая: новый кризис в основаниях математики» замечательной книги американского историка математики Мориса Клайна «Математика. Утрата определенности» [51]. Многие известные математики 19 в. высказались резко отрицательно по поводу этой теории. Леонид Кронекер (1823-1891), испытывавшей личную неприязнь к Кантору, назвал его шарлатаном. Анри Пуанкаре (1854-1912) называл теорию множеств «тяжелой болезнью» и считал ее своего рода «математической патологией». В 1908 г. он заявил: «Грядущие поколения будут рассматривать теорию множеств как болезнь, от которой они излечились».

К сожалению, у теории Кантора были не только противники, но и сторонники. Рассел назвал Кантора одним из великих мыслителей 19 в. В 1910 г. Рассел написал: «Решение проблем, издавна окутывавших тайной математическую бесконечность, является, вероятно, величайшим достижением, которым должен гордиться наш век». Рассела поддержал Гильберт: «Никто не изгонит нас из рая, созданного Кантором».

В своем выступлении на 1-м Международном Конгрессе Математиков в Цюрихе (1897) знаменитый математик Адамар подчеркнул, что главная привлекательная черта Канторовской теории множеств состоит в том, что впервые в математической истории дана классификация множеств на основе концепции «кардинального числа». По его мнению, удивительные математические результаты, которые вытекают из теории множеств Кантора, должны вдохновить математиков на новые открытия. Таким образом, в докладе Адамара Канторовская теория бесконечных множеств была вознесена на уровень главной математической теории, которая может быть положена в основания математики.

**Исследования Александра Зенкина.** В последние годы в работах выдающегося российского математика и философа Александра Зенкина [90], а также в работах других авторов [91-93] были предприняты радикальные попытки «очищения» математики от канторовской теории бесконечных множеств. Анализ канторовской теории бесконечных множеств, изложенной в статье [90], привел Александра Зенкина к заключению, что доказательства многих теорем Кантора о бесконечных множествах являются логически некорректными, а вся «теория Кантора» в некотором смысле является «математической мистификацией 19 в.». Математики 19-го века были очарованы Кантором и, приняв его необычную теорию без должного критического анализа, возвели ее в ранг величайших математических открытий 19-го века и положили ее в основания математики.

Обнаружение парадоксов в канторовской теории бесконечных множеств значительно остудило восторг математиков этой теорией, но окончательную точку в критическом анализе теории Кантора поставил Александр Зенкин [90]. Он показал, что главной ошибкой Кантора было принятие абстракции актуальной бесконечности, что, начиная с Аристотеля, недопустимо в математике. Но без абстракции актуальной бесконечности теория бесконечных множеств Кантора является несостоятельной! Как упоминалось, впервые на эту проблему обратил внимание Аристотель, который первым предупредил о невозможности использования понятия «актуальной бесконечности» в математике ("Infinitum Actus Non Datur").

В статье [93] поставлен следующий вопрос: «Не стоит ли современная математика на «лженаучном» фундаменте?». Пока математики не дали ответа на этот вопрос, касающийся оснований математики.

## **5.5. Конструктивный подход к созданию математической теории измерения**

В рамках конструктивного подхода к созданию математической теории измерения понятие «актуальной бесконечности» должно быть исключено из рассмотрения в силу его внутренней противоречивости («завершенная бесконечность»).

Идеи современных математиков-конструктивистов, рассматривающих абстракцию актуальной бесконечности, как внутренне противоречивое понятие («завершенная бесконечность»), приводят к мысли о возможности построения математической теории измерения на основе интуитивной, "практической" идеи о конечности измерительного процесса, согласно которой всякое измерение завершается за конечное число шагов, и конструктивной идеи потенциальной осуществимости, в соответствии с которой мы отвлекаемся от ограниченности наших возможностей в выборе средств и количества измерительных шагов (т.е. количество измерительных шагов может быть установлено как угодно большим и всегда существует потенциальная возможность для совершения следующего шага измерения).

Такое, на первый взгляд, незначительное изменение в подходе приводит к переосмысливанию самих задач математической теории измерения. При теоретико-множественном подходе измерение ведется "до точки", т.е. до абсолютно точного совпадения измеряемого и измеряющего отрезков (возможность такого абсолютно точного измерения вытекает из аксиомы Кантора). При конструктивном подходе измерение никогда не доходит "до точки", а результатом измерения всегда является некоторый отрезок, некоторый интервал неопределенности относительно истинного значения измеряемой величины. С увеличением числа шагов измерения

этот интервал сужается и может быть сделан как угодно малым, но никогда этот интервал не превращается в точку.

В своей известной работе "О философии математики" [94] Герман Вейль следующим образом выражает различие между классическим и конструктивным представлениями о континууме:

"Современному анализу континуум представляется в виде множества его точек, в континууме он видит лишь частный случай основного логического отношения элемента и множества. Но поразительно, что столь же фундаментальное отношение целого и части до сих пор не находило себе места в математике! Между тем обладание частями есть основное свойство континуума и брауэровская теория ... кладет это отношение в основание математического изучения континуума. В этом заключается собственно основание сделанной выше ... попытки исходить не из точек, а из интервалов, как из первичных элементов построения".

Одним из моментов в теоретико-множественной теории измерения, основанной на аксиомах Евдокса-Архимеда и Кантора, является выбор способа или алгоритма измерения, задающего систему счисления, в которой нумеруется число, являющееся результатом измерения. При бесконечном (в актуальном смысле) числе шагов измерения, т.е. при измерении "до точки", алгоритм не влияет на конечный результат измерения и поэтому проблема способов, алгоритмов измерения как серьезная математическая проблема здесь не возникает. Выбор алгоритма в значительной степени носит произвольный характер и, как правило, сводится к «десятичному» или «двоичному» алгоритму.

При конечном числе шагов измерения, т.е. при измерения "до интервала" [94], между алгоритмами измерения появляется различие в достигаемой с их помощью "точности" измерения, под которой в данном случае понимается отношение исходного интервала неопределенности к интервалу неопределенности на последнем, завершающем шаге измерения. При таких условиях вступает в действие вторая конструктивная идея об "эффективности" алгоритмов измерения, а задача синтеза "эффективных" или "оптимальных" алгоритмов измерения и выдвигается в качестве центральной задачи конструктивной (алгоритмической) теории измерения [9].

Таким образом, конструктивный подход к теории измерения приводит нас к формулировке задачи, которая, по существу, никогда не рассматривалась в математике, как серьезная математическая задача – задаче поиска «оптимальных» алгоритмов измерения. Решение такой задачи привело к развитию алгоритмической теории измерения [9], которую можно рассматривать как конструктивное направление в математической теории измерения.

## 5.6. «Индикаторная» модель измерения

**Понятие «индикаторного элемента» (ИЭ).** Ставя задачу создания алгоритмической теории измерения, необходимо еще раз уточнить, что же такое измерение, каковы его цели, что такое алгоритм измерения и какие средства используются для его реализации.

Прежде всего, заметим, что если мы хотим что-то измерять, мы должны знать объект измерения и диапазон измеряемых величин. Одно дело измерять космические расстояния, например, расстояние от Земли до Солнца, и совершенно другое - измерять атомные расстояния. Однако, когда мы переходим к «математическому измерению», мы отвлекаемся от физической природы измеряемых величин; при этом для всех случаев мы будем представлять диапазон измерения в виде геометрического отрезка  $AB$ . Ясно, что измеряемая величина есть одна из возможных величин, принадлежащих этому диапазону, то есть, до начала измерения существует некоторая «неопределенность» относительно измеряемой величины, иначе измерение было бы просто бессмысленным. Эту ситуацию «неопределенности» мы будем изображать с помощью «неизвестной» точки  $X$ , находящейся на отрезке  $AB$ .

И теперь мы можем сформулировать цель измерения. Цель измерения состоит в том, чтобы определить длину отрезка  $AX$ . На практике эта цель реализуется с помощью специальных устройств, например, рычажных весов или «компараторов». «Компараторы» осуществляют сравнение измеряемой величины с некоторой «эталонной величиной» или «мерой», сформированной из «единицы измерения», и в зависимости от результата сравнения выдают нам информацию об

измеряемой величине. Таким образом, суть измерения сводится к последовательным сравнениям измеряемой величины с некоторыми «мерами», которые мы формируем на каждом шаге измерения из единицы измерения  $V$ .

Для моделирования процесса сравнения измеряемого отрезка  $AX$  с «мерой»  $AC$ , в книге [9] введено важное понятие «индикаторного элемента» (ИЭ), который является своеобразной моделью «компаратора» или «рычажных весов», основных средств любого измерения. Будем считать, что каждый ИЭ может быть приложен к любой «известной» нам точке  $C$  отрезка  $AB$ , то есть, мы можем сформировать любую «меру» в процессе измерения. «Индикаторный элемент» дает нам информацию о взаимном расположении «неизвестной» точки  $X$  и «известной» точки  $C$ . Если ИЭ находится справа от точки  $X$ , он «индицирует» двоичный сигнал 0; в противном случае – двоичный сигнал 1.

**Конструктивная («индикаторная») модель измерения.** Для решения задач синтеза оптимальных  $n$ -шаговых алгоритмов измерения построим следующую математическую модель измерения (Рис.5.3). Пусть на отрезке  $AB$  находится «неизвестная» точка  $X$ . Длина отрезка  $AX$  определяется с помощью системы, состоящей из  $k$  «индикаторных элементов» (ИЭ).



Рис.5.3. Конструктивная («индикаторная») модель измерения

В результате приложения  $j$ -го ИЭ ( $j=1,2,\dots,k$ ) на  $l$ -м шаге алгоритма к некоторой точке  $C_j \in AB$  осуществляется "сравнение" отрезка  $AX$  и отрезка  $AC_j$ , т.е. определяется соотношение «меньше» ( $<$ ) или «больше или равно» ( $\geq$ ), в котором находятся сравниваемые отрезки  $AX$  и  $AC_j$ . Заметим, что указанное свойство ИЭ и является его основным определением.

Будем считать, что "выходной сигнал" или "показание"  $j$ -го ИЭ в точке  $C_j$  принимает значение 0, если имеет место соотношение  $AX < AC_j$ , и значение 1, если  $AX \geq AC_j$ . Будем также считать, что  $j$ -й ИЭ находится "справа" от точки  $X$ , если в результате его приложения к некоторой точке  $C_j$  его «показание» приняло значение 0, и "слева" – в противном случае.

Заметим, что задача измерения отрезка  $AX$  в «индикаторной» модели измерения (Рис.5.3) сводится к сужению интервала неопределенности относительно точки  $X$ . Процесс измерения состоит в том, что на первом шаге измерения индикаторные элементы прикладываются к некоторым точкам исходного интервала неопределенности  $AB$  и при этом на основании показаний ИЭ интервал неопределенности сужается до некоторого нового отрезка  $A_1B_1 \subset AB$ ; на всех последующих шагах индикаторные элементы прикладываются к точкам интервала неопределенности  $A_1B_1$ , выделенного на предыдущем шаге.

## 5.7. Понятие оптимального алгоритма измерения

На процесс измерения могут накладываться некоторые условия или ограничения  $S$ . Система формальных правил, определяющая для каждого отрезка  $AX$  на каждом из  $n$  шагов измерения совокупность точек приложения  $k$  индикаторных элементов на основании их показаний на предыдущих шагах при ограничениях на измерение  $S$ , называется  $(n, k, S)$ - алгоритмом измерения.

Таким образом, согласно построенной «индикаторной» модели измерения (Рис.5.3) процесс измерения состоит в последовательном прикладывании  $k$  ИЭ к точкам отрезка  $AB$  и сужению интервала неопределенности относительно точки  $X$ . Построенная модель измерения сводит задачу измерения отрезка  $AX$  к задаче детерминированного одномерного поиска координаты точки  $X$  на отрезке  $AB$  с помощью  $k$  ИЭ за  $n$  шагов. Такой методический прием позволяет эффективно использовать идеи и принципы теории одномерного поиска [95] для строгого определения понятия «оптимального»  $(n, k, S)$ -алгоритма измерения.



В результате действия  $(n, k, S)$ -алгоритма на отрезке  $AB$  при заданном  $X \in AB$  на последнем шаге алгоритма выделяется некоторый интервал неопределенности, содержащий точку  $X$ . Длина этого интервала  $\Delta$  и определяет «точность» измерения отрезка  $AX$ . Рассматривая действие алгоритма для всевозможных точек  $X$  отрезка  $AB$  и выделяя все интервалы неопределенности, содержащие соответствующие точки  $X$ , получаем множество интервалов неопределенности:

$$\Delta_1, \Delta_2, \dots, \Delta_N, \quad (5.3)$$

образующих некоторое разбиение отрезка  $AB$  на  $N$  отрезков, причем

$$AB = \Delta_1 + \Delta_2 + \dots + \Delta_N. \quad (5.4)$$

Ясно, что разбиение (5.3), удовлетворяющее соотношению (5.4), является некоторой «интегральной» характеристикой «качества»  $(n, k, S)$ -алгоритма, действующего на отрезке  $AB$ , и оно может быть использовано для введения строгого определения понятия «оптимального»  $(n, k, S)$ -алгоритма.

Рассмотрим отношение  $T_i = \frac{AB}{\Delta_i} (i = 1, 2, 3, \dots, N)$ , которое будем называть точностью определения точки  $X$ , принадлежащей интервалу  $\Delta_i$ . Выберем из разбиения (5.3) наибольший интервал неопределенности  $\Delta_{\max}$ . Отрезку  $\Delta_{\max}$  соответствует наименьшая (т.е. "наихудшая") точность определения точки  $X$

$$T_{\min} = \frac{AB}{\Delta_{\max}}, \quad (5.5)$$

с помощью которой мы и будем оценивать эффективность действия  $(n, k, S)$ -алгоритма на отрезке  $AB$ . Ясно, что каждое разбиение (5.3) характеризуется единственным значением величины (5.5). Назовем такое значение «точности»  $T_i$ , определяемое выражением (5.5),  $(n, k)$ -точностью  $(n, k, S)$ -алгоритма. Выбирая понятие  $(n, k)$ -точности в качестве критерия «эффективности» или «оптимальности», мы можем сравнивать различные  $(n, k, S)$ -алгоритмы по их  $(n, k)$ -точности, соответствующей заданным значениям  $n, k, S$ .

Проведенные рассуждения дают нам право ввести следующее строгое определение понятия оптимального  $(n, k, S)$ -алгоритма.

**Определение 5.1.** Оптимальным называется такой  $(n, k, S)$ -алгоритм, который при прочих равных условиях, то есть, при заданных значениях  $n, k, S$ , обеспечивает наибольшее значение  $(n, k)$ -точности, задаваемой (5.5).

Заметим, что в этом определении по существу использован так называемый «принцип минимакса», широко используемый в современной теории оптимальных систем [95]. Согласно этому принципу мы называем «оптимальной» такую стратегию действий, которая обеспечивает максимальное значение критерия эффективности для наихудшего случая.

Рассмотрим еще раз разбиение (5.3), которое является результатом действия  $(n, k, S)$ -алгоритма на отрезке  $AB$ , и оценим это разбиение с точки зрения введенного нами понятия  $(n, k)$ -точности. Нетрудно показать, что наибольшее значение  $(n, k)$ -точности при прочих равных условиях соответствует равномерному разбиению отрезка, когда все интервалы неопределенности (5.3) равны между собой, то есть,

$$\Delta_1 = \Delta_2 = \dots = \Delta_N = \Delta_{\max} = \frac{AB}{N}. \quad (5.6)$$

Подставляя в (5.5) вместо  $\Delta_{\max}$  его значение, задаваемое (5.6), получаем следующее выражение для  $(n, k)$ -точности алгоритма:

$$T = T_{\min} = N. \quad (5.7)$$

Это означает, что «оптимальные»  $(n, k, S)$ -алгоритмы всегда обеспечивают равномерное разбиение отрезка  $AB$ ; при этом «оптимальным» является такой  $(n, k, S)$ -алгоритм, который обеспечивает разбиение отрезка  $AB$  на наибольшее количество равных интервалов  $N$ . Ясно, что при заданном  $S$  критерий эффективности  $(n, k, S)$ -алгоритма, то есть, число уровней квантования  $N$ , зависит от числа шагов алгоритма  $n$  и числа индикаторных элементов  $k$ , участвующих в измерении, т.е.

$$N = F(n, k). \quad (5.8)$$

В дальнейшем функцию  $F(n, k)$  будем также называть «функцией эффективности» алгоритма измерения.

## 5.8. Классические алгоритмы измерения

В современной измерительной технике (в частности, в технике аналого-цифрового преобразования) широкое распространение получили следующие алгоритмы измерения:

**Алгоритм последовательного счета или просто алгоритм счета.** Этот алгоритм использует один ИЭ ( $k = 1$ ) и реализуется за  $n$  шагов; при этом отрезок  $AB$  с помощью одного ИЭ за  $n$  шагов разбивается на  $n + 1$  равных частей, т.е. в данном случае  $(n, 1)$ -точность алгоритма определяется следующей «функцией эффективности»:

$$N = F(n, 1) = n + 1. \quad (5.9)$$

Важно подчеркнуть, что этот алгоритм измерения имеет глубокие математические корни. Именно этот алгоритм лежит в основе «метода исчерпывания» Евдокса и «Евклидоваго определения» натурального числа:

$$N = \underbrace{1 + 1 + \dots + 1}_N, \quad (5.10)$$

которое задает не только натуральные числа, но и всю проблематику «элементарной теории чисел», основы которой изложены в «Началах» Евклида.

**Алгоритм поразрядного кодирования («двоичный» алгоритм измерения).** Этот алгоритм также использует один ИЭ ( $k = 1$ ) и реализуется за  $n$  шагов; при этом отрезок  $AB$  с помощью одного ИЭ разбивается на  $2^n$  равных частей, т.е. в данном случае  $(n, 1)$ -точность алгоритма определяется следующей «функцией эффективности»:

$$N = F(n, 1) = 2^n. \quad (5.11)$$

Заметим, что этот алгоритм «порождает» двоичное представление числа («двоичную систему»), лежащую в основе современной информационной технологии:

$$N = \sum_{i=0}^{n-1} a_i 2^i, \quad (5.12)$$

где  $a_i \in \{0,1\}$  - двоичная цифра, а  $2^i$  - вес  $i$ -го разряда.

Таким образом, рассмотренные выше простейшие алгоритмы измерения (алгоритм счета и «двоичный» алгоритм) порождают, с одной стороны, элементарную теорию чисел («алгоритм счета»), с другой стороны, являются основой современных компьютеров («двоичный» алгоритм).

**Алгоритм считывания.** Этот алгоритм реализуется за один шаг ( $n=1$ ) и использует  $k$  ИЭ; при этом с помощью  $k$  ИЭ отрезок  $AB$  разбивается на  $k+1$  равных частей, т.е. в данном случае  $(1,k)$ -точность алгоритма определяется следующей «функцией эффективности»:

$$N = F(1, n) = k + 1. \quad (5.13)$$

В качестве примера классических алгоритмов измерения рассмотрим действие алгоритм счета (Рис.5.4), «двоичного» алгоритма (Рис.5.5) и алгоритма считывания (Рис.5.6).

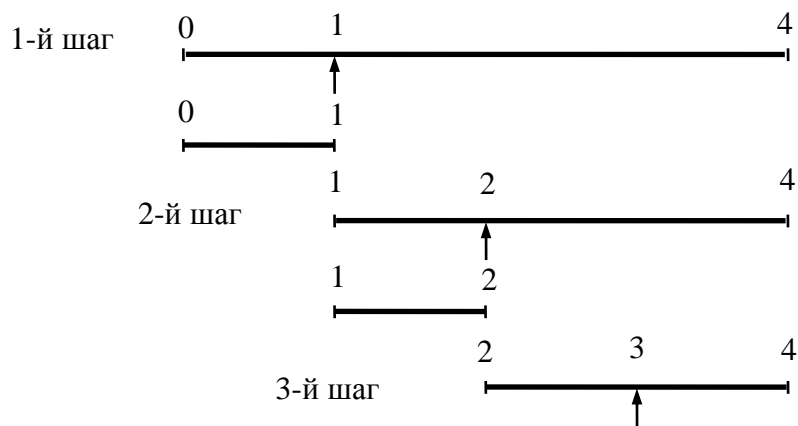


Рис.5.4. 3-шаговый «алгоритм счета»

Алгоритм счета, демонстрируемый на Рис.5.4, реализуется за 3 шага и использует только один ИЭ. Он разбивает отрезок  $[0,4]$  на 4 равные части.

Первый шаг состоит в приложении ИЭ к точке 1. При этом в зависимости от показания ИЭ могут возникнуть две ситуации: отрезок  $[0,1]$  и отрезок  $[0,4]$ .

Второй шаг.

(а) Если «показание» ИЭ в точке 1 равно 0 (ИЭ показал влево), это означает, что измеряемая точка  $X$  находится на отрезке  $[0,1]$ . В этой ситуации, процесс измерения заканчивается, так как координата точки  $X$  ( $X \in [0,1]$ ) определена с «точностью», определяемой единицей измерения.

(б) Если «показание» ИЭ в точке 1 равно двоичному сигналу 1 (ИЭ показал вправо), это означает, что измеряемая точка  $X$  находится на отрезке  $[1,4]$ . В этой ситуации процесс измерения продолжается и ИЭ на следующем шаге прикладывается к точке 2, в результате чего мы получаем две новые ситуации: отрезок  $[1,2]$  и отрезок  $[2,4]$ .

Третий шаг.

(в) Если «показание» ИЭ в точке 2 равно 0 (ИЭ показал влево), это означает, что измеряемая точка  $X$  находится на отрезке  $[1,2]$ . В этой ситуации процесс измерения заканчивается, так как координата точки  $X$  ( $X \in [1,2]$ ) определена с «точностью», соответствующей единице измерения.

(г) Если «показание» ИЭ в точке 2 равно 1 (ИЭ показал направо), это означает, что измеряемая точка  $X$  находится на отрезке  $[2,4]$ . В этой ситуации процесс измерения продолжается и ИЭ на следующем шаге прикладывается к точке 3. В результате мы получим две новые ситуации: отрезок  $[2,3]$  и отрезок  $[3,4]$  в зависимости от «показания» ИЭ на последнем шаге.

На Рис.5.5 показано действие 3-шагового «двоичного» алгоритма на отрезке  $[0,8]$ . Суть алгоритма ясна из Рис.5.5 и состоит в приложении ИЭ к середине интервала неопределенности относительно точки  $X$ , полученного на предыдущем шаге на основании «показания» ИЭ.

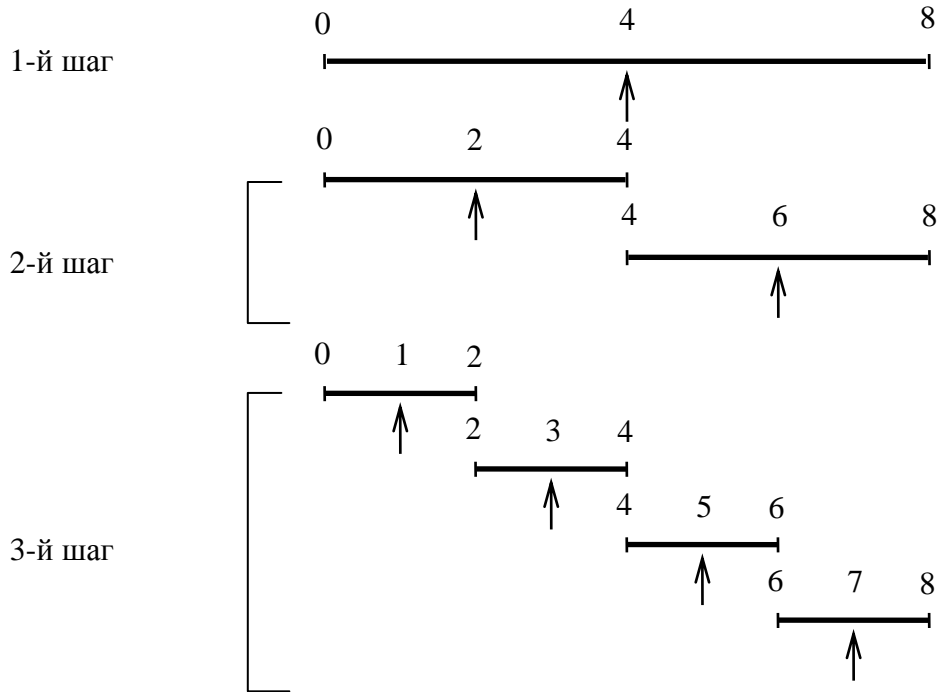


Рис.5.5. «Двоичный» алгоритм измерения

Наконец, на Рис.5.6 представлен пример алгоритма считывания, широко используемого на практике. Этот алгоритм лежит в основе традиционной измерительной линейки.

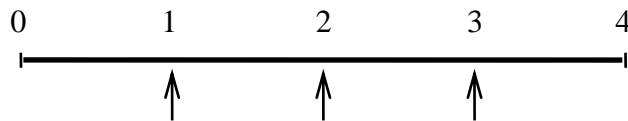


Рис.5.6. Алгоритм считывания

Алгоритм считывания, приведенный на Рис.5.6, реализуется за один шаг ( $n=1$ ) и имеет в своем распоряжении 3 ИЭ, которые прикладываются одновременно к точкам 1,2,3 отрезка  $[0,4]$ , как показано на Рис.5.6.

**Об ограничениях  $S$ .** В определении  $(n, k, S)$ -алгоритма присутствует понятие ограничения  $S$ , которое накладывается на алгоритм измерения. Ясно, что всякое «ограничение» снижает эффективность алгоритма, но, с другой стороны, каждое «ограничение», как будет показано ниже, приводит к весьма необычным алгоритмам измерения, которые могут представлять теоретический и практический интерес.

Начнем с простейших «ограничений», которые можно обнаружить из сравнительного анализа классических алгоритмов измерения, в частности, «двоичного» алгоритма и алгоритма счета. Сравним два 3-шаговые  $(n, k, S)$ -алгоритма, представленные на Рис.5.4 и 5.5. Оба алгоритма реализуются за одно и то же число шагов ( $n = 3$ ) и имеют в своем распоряжении по одному ИЭ ( $k = 1$ ). В чем же различие между этими  $(3, 1, S)$ -алгоритмами? Как показывает сравнительный анализ, различие между ними состоит в характере «движения» ИЭ на отрезке. Действительно, в примере на Рис.5.5 допускается «произвольное» движение ИЭ на отрезке  $[0, 8]$ , то есть, после любого шага на следующем шаге ИЭ может быть приложен, как справа, так и слева по отношению к точке его приложения на предыдущем шаге. Уместно для указанного выше «ограничения» использовать обозначение  $S \equiv 0$ . И тогда мы можем рассмотреть так называемые  $(n, k, 0)$ -алгоритмы, то есть, алгоритмы измерения, в которых «движение» ИЭ на отрезке подчиняется ограничению  $S \equiv 0$ .

Рассмотрим теперь алгоритм счета на Рис.5.4. В этом примере «движение» ИЭ подчиняется строгому ограничению: ИЭ в процессе действия алгоритма движется на отрезке только в одном направлении – от точки  $A$  к точке  $B$ . При этом ИЭ принимает участие в измерении до тех пор, пока он находится «слева» от точки  $X$ : как только ИЭ «перепрыгивает» через точку  $X$ , он выбывает из дальнейшего участия в измерении. Обозначим такое ограничение через  $S \equiv 1$ . И тогда мы вправе также рассмотреть так называемые  $(n, k, 1)$ -алгоритмы, то есть, алгоритмы измерения, в которых «движение» ИЭ на отрезке подчиняется ограничению  $S \equiv 1$ .

Заметим, что ограничения  $S \equiv 0$  и  $S \equiv 1$  не являются единственными возможными «ограничениями». Важно подчеркнуть, что каждое «ограничение» приводит к разработке того или иного нового класса «оптимальных» алгоритмов измерения, представляющих теоретический или практический интерес; поэтому поиск разумных «ограничений», накладываемых на алгоритм, является важной задачей алгоритмической теории измерения.

Существенно еще раз подчеркнуть, что рассмотренные выше классические алгоритмы измерения сыграли важную роль в развитии математики и информатики. В частности, алгоритм счета (Рис.5.4) лежит в основе Евклидова определения натурального числа и аксиомы Евдокса-Архимеда, а «двоичный» алгоритм измерения (Рис.5.5) лежит в основе двоичной системы, без которой невозможно представить себе современную информатику.

## 5.9. Оптимальные $(n,k,0)$ -алгоритмы

**Метод рекуррентных соотношений.** Как упоминалось выше, под  $(n,k,0)$ -алгоритмом мы будем понимать  $n$ -шаговый алгоритм измерения, который имеет в своем распоряжении  $k$  ИЭ, движение которых на отрезке  $AB$  подчиняется ограничению  $S \equiv 0$ . Напомним, что ограничение  $S \equiv 0$  по существу означает отсутствие каких-либо ограничений на «движение» ИЭ, то есть, каждый ИЭ на любом шаге алгоритма может быть приложен к любой точке отрезка  $AB$ .

Нас будет интересовать задача нахождения или синтеза «оптимального»  $n$ -шагового алгоритма, то есть, алгоритма, который при заданных  $n,k$  и заданном «ограничении»  $S \equiv 0$  разбивает отрезок  $AB$  на наибольшее число равных интервалов.

Для решения задач синтеза оптимальных  $(n,k,S)$ -алгоритмов воспользуемся так называемым «методом рекуррентных соотношений» [96], широко применяемым для решения комбинаторных задач. Суть метода состоит в том, что решение комбинаторной задачи, состоящей из  $n$  шагов, сводится к решению комбинаторной задачи для  $n-1$  шагов; при этом находится



«рекуррентное соотношение», связывающее оба решения. И так продолжается до тех пор, пока мы не придем к некоторой «простой» комбинаторной задаче, решение которой не вызывает затруднений.

**Синтез оптимального  $(n, k, 0)$  алгоритма.** Применительно к задаче синтеза оптимального  $(n, k, 0)$ -алгоритма суть метода рекуррентных соотношений состоит в следующем. Сделаем так называемое индуктивное предположение. Предположим, что для любых  $n$  и  $k$  существует оптимальный  $(n, k, 0)$ -алгоритм, который реализует на отрезке  $AB$   $(n, k)$ -точность, равную  $F(n, k)$ , то есть, «функция эффективности» алгоритма равна  $F(n, k)$ . Другими словами, наше индуктивное предположение состоит в том, что оптимальный  $(n, k, 0)$ -алгоритм разбивает отрезок  $AB$  на  $F(n, k)$  равных интервалов единичной длины  $\Delta=1$ , то есть,

$$AB = F(n, k). \quad (5.14)$$

Пусть теперь первый шаг оптимального  $(n, k, 0)$ -алгоритма на отрезке  $AB$  состоит в приложении  $k$  «индикаторных элементов» к точкам отрезка  $AB$ , как показано на Рис.5.7.

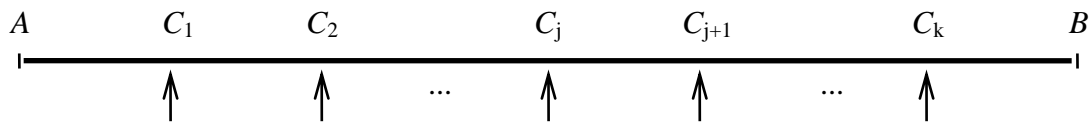


Рис.5.7. Первый шаг оптимального  $(n, k, 0)$ -алгоритма

После первого шага на основании «показаний»  $k$  ИЭ может возникнуть  $k + 1$  ситуаций:

$$(1): X \in AC_1; (2): X \in C_1C_2; \dots; (j+1): X \in C_jC_{j+1}; \dots; (k+1): X \in C_kB, \quad (5.15)$$

причем

$$AB = AC_1 + C_1C_2 + \dots + C_jC_{j+1} + \dots + C_kB. \quad (5.16)$$

Рассмотрим ситуацию  $X \in C_jC_{j+1}$ . В этой ситуации у нас осталось  $(n-1)$  шагов, поскольку один шаг уже сделан, и  $k$  ИЭ, которые, согласно ограничению  $S \equiv 0$ , могут быть приложены к любым точкам нового интервала неопределенности  $C_jC_{j+1}$ , содержащего точку  $X$ . Но тогда возникающая после 1-го шага ситуация ничем не отличается от исходной ситуации. Изменилось только число шагов, которые остались в нашем распоряжении для реализации алгоритма измерения, действующего на новом отрезке  $C_jC_{j+1}$ . И тогда в этой ситуации для поиска координаты точки  $X$  мы можем применить к новому интервалу неопределенности  $C_jC_{j+1}$  оптимальный  $(n-1, k, 0)$ -алгоритм, который, согласно нашему индуктивному предположению (5.14), разбивает отрезок  $C_jC_{j+1}$  на  $F(n-1, k)$  равных частей единичной длины  $\Delta=1$ , то есть,

$$C_jC_{j+1} = F(n-1, k). \quad (5.17)$$

Но проведенные рассуждения справедливы для любой из  $(k+1)$  ситуаций (5.15), то есть, для любого из отрезков (5.15) имеет место соотношение (5.17).

Используя теперь соотношение (5.16), связывающее исходный отрезок  $AB$  с отрезками (5.15), а также выражения (5.14), (5.16) и (5.17), мы получим следующую рекуррентную формулу для  $(n, k)$ -точности оптимального  $(n, k, 0)$ -алгоритма:

$$F(n, k) = (k+1)F(n-1, k). \quad (5.18)$$

А теперь попытаемся решить «рекуррентное соотношение» (5.18), то есть, получить представление для «функции эффективности»  $F(n, k)$  в «явном виде». Для этого применим для функции  $F(n-1, k)$  в выражении (5.18) ту же рекуррентную формулу (5.18), то есть, представим выражение (5.18) в виде:

$$F(n, k) = (k+1)(k+1)F(n-2, k). \quad (5.19)$$

Продолжая этот процесс, то есть, последовательно раскладывая в (5.19) все выражения для  $F(n-2, k)$ ,  $F(n-3, k)$ , ...,  $F(2, k)$  по рекуррентной формуле (5.18), мы получим следующее выражение для (5.18):

$$F(n, k) = (k+1)^{n-1} F(1, k). \quad (5.20)$$

В выражении (5.20) у нас имеется член  $F(1, k)$ , значение которого нам неизвестно. Чему равно  $F(1, k)$ ? Согласно нашим определениям, это  $(1, k)$ -точность оптимального  $(1, k, 0)$ -алгоритма, то есть, алгоритма, который реализуется за 1 шаг и имеет в своем распоряжении  $k$  ИЭ. Ясно, что в этом случае «оптимальная стратегия» состоит в том, чтобы с помощью  $k$  ИЭ исходный интервал неопределенности разделить на  $(k+1)$  равных частей;  $(1, k)$ -точность такого алгоритма равна

$$F(1, k) = k+1. \quad (5.21)$$

Подставляя выражение (5.21) в (5.20), получим выражение для  $(n, k)$ -точности оптимального  $(n, k, 0)$ -алгоритма в явном виде:

$$F(n, k) = (k+1)^n. \quad (5.22)$$

Легко уяснить, что стратегия оптимального  $(n, k, 0)$ -алгоритма на каждом шаге очень проста: необходимо на каждом шаге разбивать интервал неопределенности с помощью  $k$  ИЭ на  $(k+1)$  равных частей.

**Частные случаи оптимального  $(n, k, 0)$ -алгоритма.** Рассмотрим частные крайние случаи оптимального  $(n, k, 0)$ -алгоритма. Пусть  $n=1$ . В этом случае выражение (5.22) сводится к (5.21), а возникающий при этом оптимальный  $(1, k, 0)$ -алгоритм есть ни что иное, как рассмотренный нами выше алгоритм считывания (Рис.5.6).

Для случая  $k=1$  формула (5.22) сводится к выражению  $F(n, 1) = 2^n$ , которое задает значение  $(n, 1)$ -точности «двоичного» алгоритма, т.е., рассмотренный выше

«двоичный» алгоритм (Рис.5.5) является крайним частным случаем оптимального  $(n, k, 0)$ -алгоритма.

### **Оптимальные $(n, k, 0)$ -алгоритмы и позиционные системы счисления.**

Но мы уже знаем, что «двоичный» алгоритм измерения (Рис.5.4) «порождает» двоичную систему счисления (5.12). Но тогда ясно, что оптимальный  $(n, k, 0)$ -алгоритм в общем случае также «порождает» некоторую позиционную систему счисления с основанием  $R = k + 1$ , то есть, представление натурального числа в виде:

$$A = a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \dots + a_iR^i + \dots + a_1R^1 + a_0R^0, \quad (5.23)$$

где  $a_i \in \{0, 1, 2, \dots, k\}$  - цифра  $i$ -го разряда.

В частности, для случая  $k = 9$  формула (5.23) сводится к классической десятичной системе счисления:

$$A = a_{n-1}10^{n-1} + a_{n-2}10^{n-2} + \dots + a_i10^i + \dots + a_110^1 + a_010^0.$$

Для случая  $k = 59$  формула (5.23) сводится к Вавилонской 60-ричной системе счисления. Из этого рассуждения вытекает, что «оптимальные»  $(n, k, 0)$ -алгоритмы «порождают» все широко известные позиционные системы счисления (вавилонскую 60-ричную, десятичную, двоичную и др.).

Эти примеры полезны тем, что они указывают на тесную связь алгоритмической теории измерения [9] с теорией позиционных систем счисления, которая начала развиваться в математике еще с вавилонского периода.

## **5.10. Оптимальные $(n, k, 1)$ -алгоритмы, приводящие к «арифметическому квадрату»**

**Синтез оптимального  $(n, k, 1)$ -алгоритма.** А теперь синтезируем оптимальный  $(n, k, 1)$ -алгоритм [9]. Напомним, что ограничение  $S \equiv 1$  означает, что «индикаторные элементы» движутся вдоль отрезка  $AB$  только в одном

направлении - от точки  $A$  к точке  $B$ . Это означает, что если ИЭ на некотором шаге оказывается справа от искомой точки  $X$ , то этот ИЭ «выбывает из игры», то есть, не может быть использован в дальнейшем в процессе измерения. Ярким примером ограничения  $S \equiv 1$  является рассмотренный выше алгоритм счета (Рис.5.4), лежащий в основе «Евклидова определения» натурального числа (5.10).

Проведем те же рассуждения, что и при синтезе оптимального  $(n, k, 0)$ -алгоритма. Предположим, что для любых  $n$  и  $k$  существует оптимальный  $(n, k, 1)$ -алгоритм, который реализует на отрезке  $AB$   $(n, k)$ -точность, равную  $F(n, k)$ ; другими словами, оптимальный  $(n, k, 1)$ -алгоритм разбивает отрезок  $AB$  на  $F(n, k)$  равных интервалов единичной длины  $\Delta = 1$ , то есть,

$$AB = F(n, k). \quad (5.24)$$

Пусть первый шаг оптимального  $(n, k, 1)$ -алгоритма на отрезке  $AB$  состоит в приложении  $k$  «индикаторных элементов» к точкам отрезка  $AB$ , как показано на Рис.5.8.

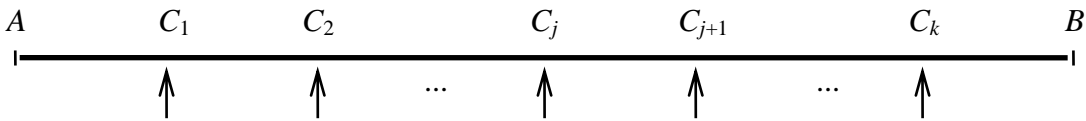


Рис.5.8. Первый шаг оптимального  $(n, k, 1)$ -алгоритма

После первого шага на основании «показаний»  $k$  ИЭ может возникнуть  $k + 1$  ситуаций:

$$(1): X \in AC_1; (2): X \in C_1C_2; \dots; (j+1): X \in C_jC_{j+1}; \dots; (k+1): X \in C_kB. \quad (5.24)$$

причем

$$AB = AC_1 + C_1C_2 + \dots + C_jC_{j+1} + \dots + C_kB. \quad (5.25)$$

А теперь проанализируем возникшие после первого шага ситуации (5.24).

Рассмотрим первую ситуацию  $X \in AC_1$ . В этой ситуации все ИЭ оказались справа от точки  $X$ . А это означает, что согласно ограничению  $S \equiv 1$  ни один из ИЭ

не может быть в дальнейшем приложен к точкам интервала  $AC_1$ , то есть, измерение фактически заканчивается после первого шага, так как мы «потеряли» все ИЭ. Но согласно нашему «индуктивному предположению» оптимальный  $(n, k, 1)$ -алгоритм «разбивает отрезок  $AB$  на  $F(n, k)$  равных частей единичной длины  $\Delta=1$ . Тогда отсюда вытекает, что первый интервал неопределенности  $AC_1$  и должен быть отрезком единичной длины, то есть,

$$AC_1 = 1. \quad (5.26)$$

Рассмотрим теперь вторую ситуацию  $X \in C_1C_2$ . В этой ситуации в нашем распоряжении осталось  $(n-1)$  шагов и 1 ИЭ, который находится слева от искомой точки  $X$  (все остальные ИЭ находятся справа от точки  $X$  и согласно ограничению  $S \equiv 1$  «выбывают из игры»). Тогда мы можем применить к отрезку  $C_1C_2$  оптимальный  $(n-1, 1, 1)$ -алгоритм, который, согласно индуктивному предположению (5.24), разбивает отрезок  $C_1C_2$  на  $F(n-1, 1)$  равных частей единичной длины  $\Delta=1$ , то есть,

$$C_1C_2 = F(n-1, 1). \quad (5.27)$$

Рассмотрим теперь ситуацию  $X \in C_jC_{j+1}$ . В этой ситуации в нашем распоряжении осталось  $(n-1)$  шагов и  $j$  ИЭ, которые находятся слева от искомой точки  $X$  (все остальные ИЭ находятся справа от точки  $X$  и согласно ограничению  $S \equiv 1$  «выбывают из игры»). Тогда мы можем применить к отрезку  $C_jC_{j+1}$  оптимальный  $(n-1, j, 1)$ -алгоритм, который, согласно индуктивному предположению (5.24), разбивает отрезок  $C_jC_{j+1}$  на  $F(n-1, j)$  равных частей единичной длины, то есть,

$$C_jC_{j+1} = F(n-1, j). \quad (5.28)$$

Наконец, в последней ситуации  $X \in C_kB$  все  $k$  ИЭ оказываются слева от точки  $X$ ; это означает, что мы можем применить к отрезку  $C_kB$  оптимальный

$(n-1, k, 1)$ -алгоритм, который, согласно индуктивному предположению (5.24), разбивает отрезок  $C_k B$  на  $F(n-1, k)$  равных частей единичной длины, то есть,

$$C_k B = F(n-1, k). \quad (5.29)$$

Учитывая соотношение (5.25), а также выражения (5.26)-(5.29), мы можем записать следующее рекуррентное соотношение для  $(n, k)$ -точности оптимального  $(n, k, 1)$ -алгоритма:

$$F(n, k) = 1 + F(n-1, 1) + \dots + F(n-1, j) + \dots + F(n-1, k-1) + F(n-1, k). \quad (5.30)$$

А теперь рассмотрим сумму

$$1 + F(n-1, 1) + \dots + F(n-1, j) + \dots + F(n-1, k-1), \quad (5.31)$$

взятую из выражении (5.30). Согласно рекуррентной формуле (5.30), сумма (5.31) равна  $F(n, k-1)$ , то есть,

$$F(n, k-1) = 1 + F(n-1, 1) + \dots + F(n-1, j) + \dots + F(n-1, k-1) \quad (5.32)$$

Используя (5.32), мы можем упростить рекуррентное соотношение (5.30) и записать его в виде:

$$F(n, k) = F(n, k-1) + F(n-1, k). \quad (5.33)$$

А теперь построим таблицу чисел  $F(n, k)$ , используя рекуррентную формулу (5.33). Для этого выясним, чему равны крайние значения функции  $F(n, k)$ , соответствующие значениям  $n=0$  и  $k=0$ , то есть, значения  $F(0, k)$  и  $F(n, 0)$ . Напомним, что  $F(0, k)$  есть  $(0, k)$ -точность оптимального  $(0, k, 1)$ -алгоритма, а  $F(n, 0)$  есть  $(n, 0)$ -точность оптимального  $(n, 0, 1)$ -алгоритма. Но согласно нашим определениям,  $(0, k, 1)$ -алгоритм есть  $(n, k, 1)$ -алгоритм, в котором число шагов равно нулю, то есть,  $n=0$ , а  $(n, 0, 1)$ -алгоритм есть  $(n, k, 1)$ -алгоритм, в котором число ИЭ равно нулю, то есть,  $k=0$ . Но из «физического» смысла решаемой задачи,  $(n, k, 1)$ -алгоритм, в котором либо  $n=0$ , либо  $k=0$  не может сузить исходный интервал неопределенности, и поэтому для такого  $(n, k, 1)$ -

алгоритма  $(n, k)$ -точность или функция эффективности всегда тождественно равна 1, то есть,

$$F(0, k) = F(n, 0) = 1. \quad (5.34)$$

**Арифметический квадрат.** Используя рекуррентное соотношение (5.33) и начальные условия (5.34), мы можем построить следующую таблицу для численных значений функции  $F(n, k)$  (см. Табл.5.1).

Таблица 5.1. Арифметический квадрат

	0	1	2	3	4	5	...	$n$
0	1	1	1	<b>1</b>	1	1	...	1
1	1	2	3	<b>4</b>	5	6	...	$F(n, 1)$
2	1	3	6	<b>10</b>	15	21	...	$F(n, 2)$
3	1	4	10	<b>20</b>	35	56	...	$F(n, 3)$
4	1	5	15	35	70	126	...	$F(n, 4)$
5	1	6	21	56	126	252	...	$F(n, 5)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$k$	1	$F(1, k)$	$F(2, k)$	$F(3, k)$	$F(4, k)$	$F(5, k)$	...	$F(n, k)$

Сравнивая эту таблицу с таблицей биномиальных коэффициентов, известной под названием «арифметический квадрат» или «прямоугольник Тарталья», мы приходим к неожиданному заключению, что «функция эффективности»  $F(n, k)$  выражается с помощью формулы для биномиальных коэффициентов, то есть,

$$F(n, k) = C_{n+k}^n = C_{n+k}^k. \quad (5.35)$$

**Оптимальный  $(n, k, 1)$ -алгоритм.** Действие оптимального  $(n, k, 1)$ -алгоритма может быть продемонстрировано с помощью арифметического квадрата (Табл.5.1). Действительно, для заданных  $n$  и  $k$  значение «функции эффективности»  $F(n, k)$  находится на пересечении  $n$ -го столбца и  $k$ -й строки



арифметического квадрата. Координаты точек приложения  $k$  ИЭ на отрезке  $AB$  (точки  $C_1, C_2, \dots, C_j, C_{j+1}, \dots, C_k$ ) относительно точки  $A$  (Рис.5.8) находятся в  $n$ -м столбце «арифметического квадрата», то есть:

$$AC_1 = 1, AC_2 = F(n, 1), \dots, AC_j = F(n, j-1), \dots, AC_k = F(n, k-1). \quad (5.36)$$

Если после первого шага алгоритма  $j$  ИЭ оказываются слева от точки  $X$ , а оставшиеся  $(k-j)$  ИЭ – справа от точки  $X$ , то «интервал неопределенности» относительно точки  $X$  уменьшается до отрезка  $C_j C_{j+1}$ . Длина этого отрезка равна  $F(n-1, j)$ . Этот биномиальный коэффициент находится на пересечении  $(n-1)$ -го столбца и  $j$ -й строки арифметического квадрата. Чтобы найти биномиальный коэффициент  $F(n-1, j)$ , мы должны сдвинуться от исходного коэффициента  $F(n, k)$  на один столбец влево (мы «потеряли» один шаг) и  $(k-j)$  строк вверх (мы «потеряли»  $(k-j)$  ИЭ).

На втором шаге мы принимаем точку  $C_j$  за новое начало координат. При этом 2-й шаг состоит в приложении  $j$  ИЭ к некоторым точкам  $D_1, D_2, \dots, D_j$  нового «интервала неопределенности»  $C_j C_{j+1}$ . Координаты точек  $D_1, D_2, \dots, D_j$  относительно нового начала координат (точка  $C_j$ ) находятся в  $(n-1)$ -м столбце арифметического квадрата выше биномиального коэффициента  $F(n-1, j)$ , то есть,

$$C_j D_1 = 1, C_j D_2 = F(n-1, 1), \dots, C_j D_3 = F(n-1, 2), C_j D_j = F(n-1, j-1) \quad (5.37)$$

Этот процесс заканчивается, когда будут «исчерпаны» либо все шаги алгоритма, либо все «индикаторные элементы».

**Пример оптимального  $(n, k, 1)$ -алгоритма.** В качестве примера рассмотрим действие оптимального  $(3, 3, 1)$ -алгоритма на отрезке  $[0, 20]$  (Рис. 5.9).

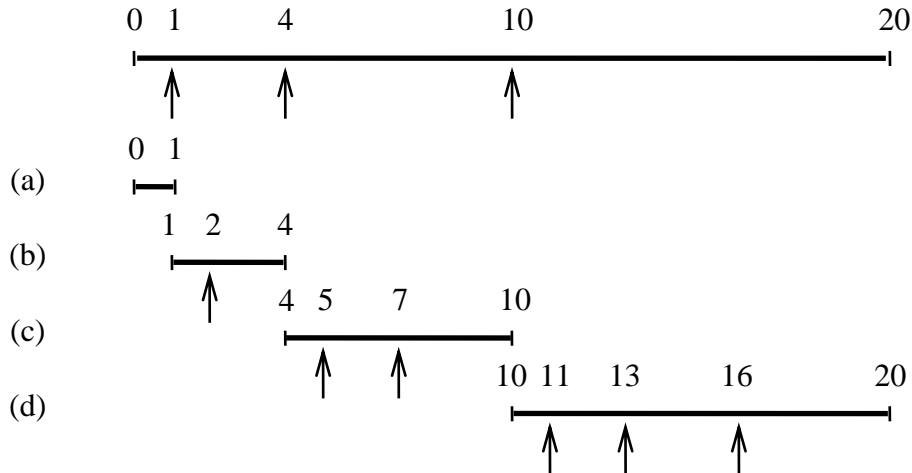


Рис.5.9. Оптимальный  $(3,3,1)$ -алгоритм

Оптимальный  $(3,3,1)$ -алгоритм реализуется за 3 шага и использует при этом 3 ИЭ. Заметим, что число  $F(3,3) = 20$ , которое находится на пересечении 3-го столбца и 3-й строки арифметического квадрата (Табл.5.1), равно длине исходного «интервала неопределенности» для данного алгоритма (выделено жирным шрифтом и заключено в квадрат в Табл.5.1). Значения координат приложений ИЭ на первом шаге  $(3,3,1)$ -алгоритма находятся в 3-м столбце арифметического квадрата (Табл.5.1) выше числа 20 (выделены жирным шрифтом).

Первый шаг алгоритма на отрезке  $[0, 20]$  состоит в приложении трех ИЭ к точкам 1, 4, 10 (Рис.5.9). После первого шага на основании «показаний» ИЭ может возникнуть 4 ситуации (Рис.5.9-а, -b, -с, -d).

Второй шаг.

Ситуация (а). Для этой ситуации процесс измерения заканчивается, так как все ИЭ находятся справа от точки  $X$ .

Ситуация (b). Для этой ситуации мы имеем только 1 ИЭ, который прикладывается к точке 2.

Ситуация (с). Для этой ситуации мы имеем 2 ИЭ, которые прикладываются к точкам 5 и 7.

Ситуация (d). Для этой ситуации мы имеем 3 ИЭ. В этом случае точка 10 является новым началом координат и имеющиеся в нашем распоряжении 3 ИЭ прикладываются к точкам 1,3,6 относительно нового начала координат. Просуммировав эти числа с числом 10, мы можем получить координаты точек приложения ИЭ на следующем шаге: 11, 13, 16.

После второго шага может возникнуть следующие ситуации: [1,2], [2,4], [4,5], [5,7], [7,10], [10,11], [11,13], [13,16], [16,20]. Заметим, что для ситуаций [1,2], [4,5], [10,11] процесс измерения заканчивается на втором шаге.

### Третий шаг.

Для ситуаций [2,4], [5,7] и [11,13] третий шаг состоит в приложении одного ИЭ к точкам 3, 6, 12, соответственно.

Для ситуаций [7,10] и [13,16] третий шаг состоит в приложении двух ИЭ к точкам 8, 9 или 14, 15 соответственно.

Для ситуации [16,20] третий шаг состоит в приложении трех ИЭ к точкам 17, 18, 19.

**Крайние частные случаи оптимального  $(n,k,1)$ -алгоритма.** А теперь рассмотрим крайние частные случаи найденного нами оптимального  $(n,k,1)$ -алгоритма, то есть, когда  $n=1$  или  $k=1$ . Для случая  $n=1$  выражение (5.35) принимает вид:

$$F(1,k) = C_{1+k}^1 = C_{1+k}^k = k+1. \quad (5.38)$$

Нетрудно убедиться, что в этом случае ( $n=1$ ) оптимальный  $(1,k,1)$ -алгоритм сводится к рассмотренному выше алгоритму считывания (Рис.5.6).

Для случая  $k=1$  выражение (5.35) принимает вид:

$$F(n,1) = C_{n+1}^n = C_{n+1}^1 = n+1. \quad (5.39)$$

Нетрудно убедиться, что в этом случае ( $k=1$ ) оптимальный  $(n,1,1)$ -алгоритм сводится к рассмотренному выше алгоритму счета (Рис.5.4).

### **Значение биномиальных алгоритмов для математики и информатики.**

Таким образом, основной математический результат, полученный выше при синтезе оптимального  $(n, k, 1)$ -алгоритма [9], состоит в том, что эти исследования привели нас к арифметическому квадрату и биномиальным коэффициентам. Подчеркнем, что этот результат является неожиданным, так как, приступая к синтезу этого алгоритма, мы не предполагали его связь с биномиальными коэффициентами. Это дает нам основание назвать  $(n, k, 1)$ -алгоритм биномиальным алгоритмом измерения.

Какое значение могут иметь биномиальные алгоритмы измерения для математической науки? Здесь важно подчеркнуть, что оптимальный  $(n, k, 1)$ -алгоритм представляет собой обобщение алгоритма счета (Рис.5.4), который, как упоминалось, исторически лежит в основе элементарной теории чисел, основы которой изложены в «Началах» Евклида. Поэтому мы можем высказать предположение, что рассмотренный выше биномиальный алгоритм измерения может быть использован для дальнейшего развития элементарной теории чисел.

Но существует еще одна идея, вытекающая из трактовки оптимальных  $(n, k, S)$ -алгоритмов, как новых позиционных представлений натуральных чисел. При таком подходе биномиальные алгоритмы измерения могут иметь практическое значение для современных компьютеров. Этот алгоритм «генерирует» весьма необычные позиционные системы счисления, в которых весами разрядов являются биномиальные коэффициенты. И кто знает, может быть, кто-либо из наших читателей, увлекшись такой необычной системой счисления, построит новый («биномиальный») компьютер?

И в этом предположении нет ничего необычного, если учесть, что украинский исследователь Алексей Борисенко в течение нескольких десятилетий успешно развивает теорию биномиальных систем счисления и на эту тему он написал интересную книгу «Биномиальный счет. Теория и практика» [97].

## Глава 6

# ПРИНЦИП АСИММЕТРИИ ИЗМЕРЕНИЯ И ФИБОНАЧЧИВЫЕ АЛГОРИТМЫ ИЗМЕРЕНИЯ

### 6.1. Задача Баше-Менделеева

**Задача о выборе наилучшей системы гирь.** В главе 2 мы рассказали о выдающемся итальянском математике Леонардо из Пизы (по прозвищу Фибоначчи). В современной науке его имя известно в связи с числовой последовательностью 1,1,2,3,5,8,13,21,34,..., к которой он пришел при решении «задачи о размножении кроликов». Эту задачу он описал в своей книге «Liber abaci», опубликованной в 1202 г.

Оказывается, что Фибоначчи внес существенный вклад и в развитие математической теории измерения, впервые сформулировав так называемую «задачу о выборе наилучшей системы гирь для взвешивания на рычажных весах» или проще «задачу о гирях». История задачи такова. Из сочинений Фибоначчи она перекочевала в сочинения еще одного знаменитого итальянского математика Луки Пачоли. Пачоли поместил ее в свою книгу «Summa de Arithmetica, Geometria, Proportioni et Proportionalita», опубликованную в 1494 г. Эта книга по праву считается математической энциклопедией эпохи Возрождения.

Затем задача о гирях появляется в «Сборнике приятных и занимательных задач» (1612 г.), написанном французским математиком Баше де Мизириаком. Из «Биографического словаря деятелей в области математики» [98] мы узнаем о Баше де Мизириаке следующее:

«Баше де Мизириак Гаспар Клод (9.10.1581 – 26.2.1638) – французский математик и поэт. Писал по-французски, по-итальянски и на латыни, знал греческий язык. В книге «Приятные и занимательные задачи» (Лион, 1612) Баше собрал старинные занимательные задачи. В задачах автор выделяет математически

интересные моменты и старается обобщить частные вопросы. Баше дал решение неопределенных уравнений 1-й степени в целых числах, где был близок к непрерывным дробям, в 1621 г. издал «Арифметику» Диофанта на греческом и латинском языках, снабдив ее дополнениями и примечаниями».

В русской историко-математической литературе «задача о гирях» известна также под названием «задачи Баше-Менделеева». О Баше де Мизириаке мы уже рассказали. Но кто такой Менделеев? Неужели это знаменитый русский химик Дмитрий Менделеев, автор Периодического закона? Да, это именно так. Но почему русский химик вдруг заинтересовался «задачей о гирях»? Ответ на этот вопрос дает ознакомление с некоторыми малоизвестными фактами из жизни гениального ученого. Во время происходивших в 1890 г. студенческих волнений в Петербургском университете Менделеев, который в тот период работал профессором этого университета, выступил в защиту студентов и в качестве протеста подал прошение об отставке с должности профессора университета. Его прошение было удовлетворено и поэтому в 1892 г. Менделеев был назначен ученым хранителем Депо образцовых гирь и весов, которое по инициативе Менделеева в 1893 г. было преобразовано в Главную палату мер и весов России. Ее директором Менделеев оставался до конца жизни. Таким образом, заключительный этап жизни великого ученого (с 1892 г. и до его кончины в 1907 г.) был связан с развитием измерительного дела и именно в этот период со свойственной ему активностью Менделеев интересуется различными задачами, связанными с измерениями; одной из них и была «задача о гирях». Под его непосредственным влиянием было выполнено математические исследования по этой проблеме [99,100]. Вклад Менделеева в развитие измерительного дела в России настолько велик, что «задача о гирях» была названа задачей Баше-Менделеева, а сам Менделеев назван отцом русской метрологии.

В чем же состоит суть этой задачи, впервые сформулированной Фибоначчи еще в 13 в.? Одним из наиболее древних измерительных устройств являются рычажные весы, которыми каждый из нас пользовался неоднократно. При взвешивании мы используем некоторую систему гирь; при этом «взвешивание» некоторого груза  $Q$  осуществляется путем его сравнения с гирями, имеющимися в

нашем распоряжении. Процедура взвешивания, выполняемая в соответствии с некоторыми правилами, называется алгоритмом взвешивания или алгоритмом измерения. При этом возникает задача о выборе оптимальной системы гирь, которая по существу сводится к задаче о нахождении оптимального алгоритма измерения. Но это именно та задача, которую мы сформулировали в качестве центральной задачи конструктивной (алгоритмической) теории измерения. Таким образом, заслуга Фибоначчи состоит в том, что он сформулировал первую в истории математики оптимизационную задачу в теории измерения.

**Варианты решения «задачи Баше-Менделеева».** Известны два варианта решения «задачи Баше-Менделеева» [9]. В первом случае взвешиваемый груз находится на левой чаше весов, а гири разрешается класть только на правую («свободную») чашу весов; во втором варианте гири разрешается класть на обе чаши весов.

Рассмотрим первый вариант задачи, приводящий к доказательству «оптимальности» двоичной системы гирь. Он формулируется следующим образом. Для упрощения задачи будем считать, что взвешиваемый груз принимает значения из множества неотрицательных целых чисел, то есть,  $Q = 0, 1, 2, 3, \dots$ . Рассмотрим некоторую систему, состоящую из  $n$  гирь:  $\{q_1, q_2, q_3, \dots, q_n\}$ , где  $q_1 = 1$  - это единица измерения, а все остальные гири находятся с единицей  $q_1 = 1$  в кратном отношении, причем:

$$q_n \geq q_{n-1} \geq q_{n-2} \dots \geq q_1 = 1. \quad (6.1)$$

Ясно, что максимальный груз, который может быть взвешен с помощью данной системы гирь, равен сумме всех имеющихся гирь, то есть

$$Q_{\max} = q_1 + q_2 + q_3 + \dots + q_n. \quad (6.2)$$

При этом гири необходимо подобрать так, чтобы из них можно было бы составить любой груз, кратный  $q_1 = 1$ , в диапазоне от 0 до наибольшего груза.

Ясно, что если задана единица измерения  $q_1 = 1$  и количество гирь  $n$ , то вес наибольшего груза является некоторой функцией от  $n$ , то есть,  $Q_{\max} = f(n)$ . Задача

состоит в том, чтобы при заданном  $n$  выбрать такую систему гирь, при которой вес наибольшего груза  $Q_{\max}$  был бы наибольшим для всех допустимых вариантов.

Поясним существо «задачи о гирях» на конкретном примере. Пусть требуется выбрать систему из четырех гирь среди следующих вариантов:  $\{1,3,5,10\}$ ,  $\{1,2,4,8\}$ ,  $\{1,2,3,5\}$ . Ясно, что первый вариант  $\{1,3,5,10\}$  нас не удовлетворяет, так как с его помощью невозможно, например, путем сложения составить веса 7, 12, 17. Второй и третий варианты удовлетворяют нашему условию, но второй вариант лучше, поскольку сумма гирь  $1+2+4+8=15$  для этого варианта больше суммы гирь  $1+2+3+5=11$  для третьего варианта.

Известно общее решение сформулированной выше задачи [9]. Оно состоит в выборе двоичной системы гирь  $\{1,2,4,\dots,2^{n-1}\}$ , с которыми последовательно (начиная со старшей гири) происходит сравнение измеряемого груза  $Q$ . Указанный способ измерения получил широкое распространение в измерительной практике и носит название алгоритма поразрядного кодирования или «двоичного» алгоритма измерения. Важно подчеркнуть, что применение этого алгоритма автоматически приводит к представлению результата измерения  $N$  в двоичной системе счисления:

$$N = \sum_{i=1}^n a_i 2^{i-1}, \quad (6.3)$$

где  $a_i \in \{0,1\}$  - двоичная цифра  $i$ -го разряда двоичного кода числа  $N$ . В задаче о гирях двоичные цифры имеют следующую физическую интерпретацию:  $a_i = 1$ , если в результате сравнения с гирей  $2^{i-1}$  рычажные весы остались в исходном положении «больше»;  $a_i = 0$ , если чаши весов перешли в противоположное положение «меньше» (мы рассматриваем случай, когда рычажные весы могут находиться только в двух крайних положениях «больше» и «меньше»).

Таким образом, в результате проведенных рассуждений мы неожиданно перешли от алгоритмов измерения к способам представления чисел и, как упоминалось, в этом и состоит одно из важных практических приложений алгоритмической теории измерения.



Во втором варианте задачи Баше-Менделеева гири разрешается класть на обе чаши весов. Оказалось [99,100], что для этого случая «оптимальным» решением является «троичная» система гирь  $\{1, 3, 9, 27, \dots, 3^{n-1}\}$ , которая «порождает» троичную симметричную систему счисления:

$$N = \sum_{i=1}^n b_i 3^{i-1}, \quad (6.4)$$

где  $b_i \in \{\bar{1} = -1, 0, 1\}$  - троичная цифра  $i$ -го разряда троичного представления числа  $N$ .

## 6.2. Принцип асимметрии измерения

Принципы конечности и потенциальной осуществимости измерительного акта, рассмотренные нами в главе 5, являются как бы «внешними» по отношению к измерению и носят столь общий характер, что есть опасность ее сведения к некоторому тривиальному результату (например, к доказательству «оптимальности» упомянутого выше «двоичного» алгоритма измерения, который сводит процесс измерения к последовательному сравнению измеряемой величины с «двоичными» гирями:  $2^{n-1}, 2^{n-2}, \dots, 2^0$ , начиная со старшей гири).

Для получения новых нетривиальных результатов методологический базис конструктивной (алгоритмической) теории измерения должен быть дополнен неким общим принципом, вытекающим из самой природы, сущности измерения. Такой принцип непосредственно вытекает из анализа «двоичного» алгоритма измерения, являющегося «оптимальным» решением рассмотренной выше «задачи о гирях» [58].

При внимательном анализе «двоичного» алгоритма измерения обнаруживается одна его особенность, которая имеет общий характер для любых мыслимых измерений, основанных на сравнении измеряемой величины с «эталонными гирями», и наглядно может быть продемонстрирована на модели

взвешивания на рычажных весах неизвестного груза  $Q < 2^n$  с помощью системы двоичных гирь:  $\{2^{n-1}, 2^{n-2}, \dots, 2^0\}$  (Рис.6.1).

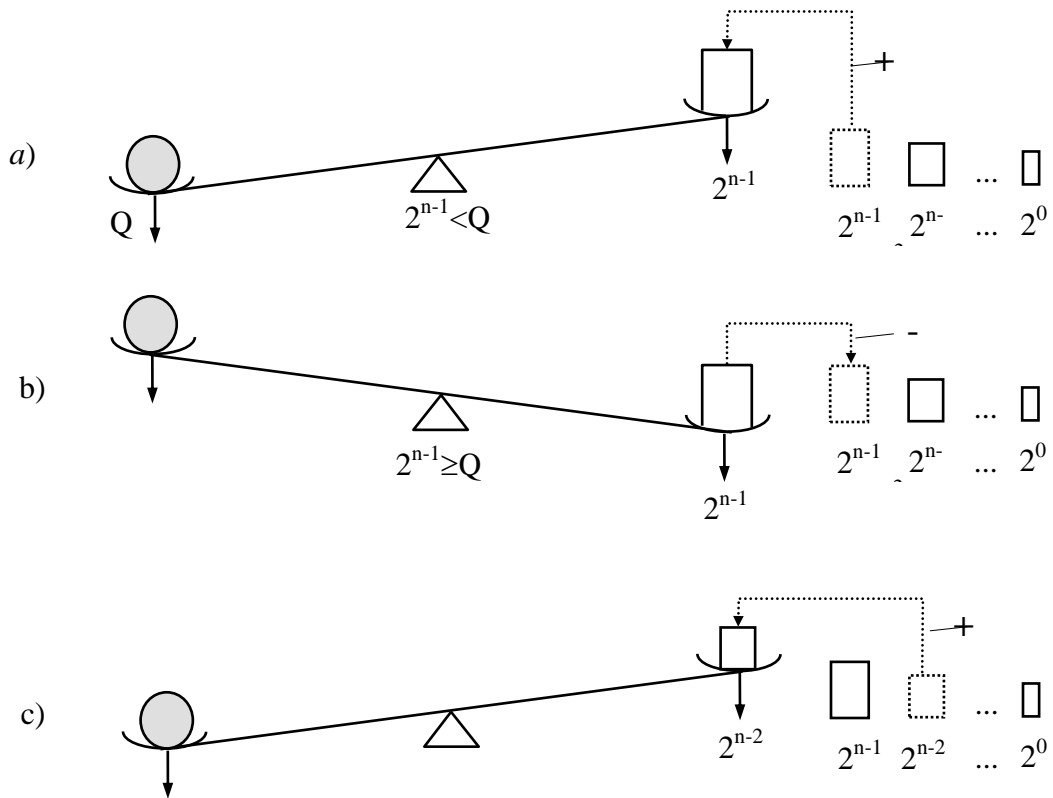


Рис.6.1. Принцип асимметрии измерения

На первом шаге взвешивания на правую чашу весов кладется «старшая» гиря весом  $2^{n-1}$  (Рис. 6.1-а), что обозначено символом «+» («добавить»). При этом могут возникнуть две ситуации, изображенные на Рис.6.1-а ( $2^{n-1} < Q$ ) и на Рис.6.1-б ( $2^{n-1} \geq Q$ ). В первом случае (Рис.6.1-а) следующий шаг состоит в том, чтобы добавить (+) на правую чашу весов очередную по старшинству гирю  $2^{n-2}$ . Во втором случае (Рис.6.1-б) «весовщик» должен выполнить две операции: снять (-) с правой чаши весов предыдущую гирю, после чего весы возвращаются в исходное положение (Рис.6.1-с); после возвращения рычажных весов в исходное положение

он должен добавить (+) на правую чашу весов следующую по старшинству гирию  $2^{n-2}$ .

Следовательно, логика любого сравнения с помощью рычажных весов асимметрична, так как предполагает различную степень сложности действий «весовщика» в зависимости от положений, в которых оказываются рычажные весы (устройство сравнения) после очередного шага сравнения; при этом действия «весовщика» после получения сигнала «больше» (правая чаша перевесила) оказываются «сложнее» по сравнению с его действиями после получения сигнала «меньше» (весы остались в исходном положении).

Обнаруженное свойство измерения и составляет содержание принципа асимметрии измерения [9,58]. Сложность действий «весовщика» после получения сигнала «больше» (ситуация на Рис.6.1-b) определяется двумя факторами. Во-первых, он должен снять гирию с правой чаши весов и, во-вторых, учесть время, затрачиваемое на «восстановление» весов в исходное положение. Введение восстановительного периода устройства сравнения (рычажных весов) и учет этого периода в математической модели измерения и его алгоритме и является центральной идеей алгоритмической теории измерения, вытекающей из принципа асимметрии измерения [9,58].

### 6.3. Новая формулировка задачи Баше-Менделеева

Введем теперь обнаруженное выше свойство измерения в «задачу о гириях», предложенную Фибоначчи. С этой целью будем рассматривать измерение как процесс, протекающий в дискретные моменты времени; и пусть операция «добавить гирию» выполняется за одну единицу дискретного времени, а операция «снять гирию» (которая сопровождается возвратом рычажных весов в исходное положение) выполняется за  $p$  единиц дискретного времени, причем  $p \in \{0, 1, 2, 3, \dots\}$ .

Ясно, что параметр  $p$  как бы моделирует инерционность рычажных весов. При этом случай  $p = 0$  соответствует той «идеальной ситуации», когда мы

пренебрегаем «инерционностью» рычажных весов. Именно этот случай и рассматривал Фибоначчи и все последующие ученые, которые изучали эту задачу [99,100]. Для остальных случаев  $p > 0$  мы имеем новые варианты «задачи о гирях», решение которых и составляет основное содержание рассматриваемой теории.

Дальнейшее обобщение задачи Баше-Менделеева состоит в увеличении рычажных весов от 1 до  $k$  ( $k$  – натуральное число), причем на левые чаши всех весов положен один и тот же груз  $Q$ ; такая ситуация соответствует случаю «параллельных измерений», когда одна и та же измеряемая величина сравнивается с «эталонными величинами» с помощью  $k$  «компараторов» (такой прием широко используется при измерении электрических величин). В этом случае обобщенная «задача о гирях» может быть сформулирована следующим образом: требуется найти «оптимальный»  $n$ -шаговый алгоритм взвешивания (измерения) с помощью системы из  $k$  рычажных весов («компараторов»), обладающих «инерционностью»  $p$ , при условии, что на каждом шаге измерения гири («эталонные величины») разрешается класть на свободные чаши тех и только тех рычажных весов («компараторов»), которые на этом шаге находятся в исходном положении «больше».

Ясно, что такая задача является несравненно значительно более сложной, чем классическая «задача о гирях» (сформулированная Фибоначчи еще в 13-м веке), которая является частным случаем сформулированной выше задачи для случая  $k=1$  и  $p=0$ .

## **6.4. Синтез оптимального фибоначчиевого алгоритма измерения**

**Фибоначчиевые алгоритмы измерения.** А теперь приступим к обсуждению наиболее неожиданного результата «алгоритмической теории измерения» [9,58], а именно к фибоначчиевым алгоритмам измерения.

Возвратимся снова к математической модели измерения, основанной на «индикаторных элементах», и попытаемся ввести в эту модель ограничение на

алгоритм, вытекающее из сформулированного выше принципа асимметрии измерения (Рис.6.1). Из этого принципа вытекает следующее «ограничение» на движение ИЭ вдоль отрезка  $AB$ . Пусть ИЭ приложен к точке  $C$  на первом шаге  $n$ -шагового алгоритма измерения (Рис.6.2). Тогда может возникнуть две ситуации (а) и (b), показанные на Рис.6.2. Ясно, что в ситуации на Рис.6.2-а нам разрешено прикладывать ИЭ к некоторой точке отрезка  $CB$  уже на следующем шаге. Однако для ситуации на Рис.6.2-б, исходя из принципа асимметрии измерения, нам запрещено прикладывать ИЭ к точкам отрезка  $AC$ , потому что рычажные весы, которым в нашей модели соответствует «индикаторный элемент», должны возвратиться в исходное положение в течение  $p$  единиц дискретного времени. Таким образом, ограничение  $S$ , вытекающие из принципа асимметрии измерения, состоит в том, что для ситуации на Рис.6.2-б запрещено прикладывать ИЭ к точкам отрезка  $AC$  в течение  $p$  последующих шагов алгоритма.

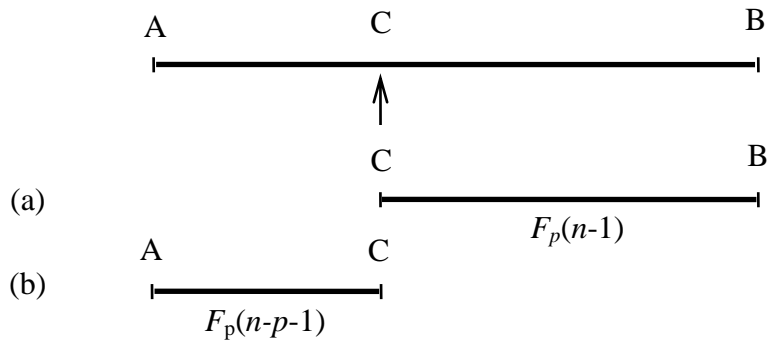


Рис.6.2. Первый шаг «фибоначчиевого» алгоритма измерения

А теперь используем тот же прием, который был нами успешно применен при синтезе оптимальных  $(n, k, 0)$ - и  $(n, k, 1)$ -алгоритмов. Предположим, что при заданном  $p$  для любого  $n$  существует оптимальный  $n$ -шаговый алгоритм, который реализует на отрезке  $AB$   $(n, 1)$ -точность, равную  $F_p(n)$ ; другими словами, наше индуктивное предположение состоит в том, что оптимальный  $n$ -

шаговый алгоритм разбивает отрезок  $AB$  на  $F_p(n)$  равных интервалов единичной длины  $\Delta=1$ , то есть,

$$AB = F_p(n). \quad (6.5)$$

Предположим теперь, что 1-й шаг оптимального  $n$ -шагового алгоритма, удовлетворяющего индуктивному предположению (6.5), состоит в приложении ИЭ к некоторой точке  $C$  (Рис.6.2). Тогда в результате этого шага может возникнуть две ситуации:  $(a) X \in AC$ ;  $(b) X \in CB$ .

Начнем со второй ситуации  $X \in CB$ . На языке «рычажных весов» эта ситуация означает, что весы остались в том же положении и тогда, согласно принципу асимметрии измерения, мы имеем право прикладывать на следующем шаге имеющийся в нашем распоряжении ИЭ к точкам отрезка  $CB$ . В этом случае, действуя на отрезке  $CB$  оптимальным  $(n-1)$ -шаговым алгоритмом, согласно индуктивному предположению (6.5), мы можем разделить интервал неопределенности  $CB$  на  $F_p(n-1)$  равных отрезков единичной длины, то есть,

$$CB = F_p(n-1). \quad (6.6)$$

Рассмотрим теперь ситуацию  $X \in AC$ . На языке «рычажных весов» эта ситуация означает, что весы перешли в противоположное состояние и тогда, согласно принципу асимметрии измерения, мы должны пропустить  $p$  шагов, пока «рычажные весы» не возвратятся в исходное положение. На языке модели «индикаторных элементов» это означает, что мы имеем право прикладывать имеющийся в нашем распоряжении ИЭ к точкам отрезка  $AC$  только спустя  $p$  шагов после совершения первого шага. Поскольку в нашем распоряжении осталось  $(n-1)$  шагов, то наши последующие рассуждения зависят от соотношения между числами  $p$  и  $(n-1)$ .

Предположим, что  $p \geq (n-1)$ . В этом случае все наши шаги будут исчерпаны до того, как «рычажные весы» перейдут в исходное положение, то есть, на этом процесс измерения фактически заканчивается. Тогда из «физических

соображений» вытекает, что отрезок  $AC$  должен быть отрезком единичной длины, то есть, для случая  $p \geq (n-1)$  мы будем иметь:

$$AC = 1. \quad (6.7)$$

Рассмотрим теперь случай  $p < (n-1)$ . Тогда через  $p$  шагов после первого шага, «рычажные весы» возвратятся в исходное положение, то есть, спустя  $p+1$  шагов с начала измерения, согласно принципу асимметрии измерения, мы имеем право продолжать измерение, прикладывая ИЭ к точкам отрезка  $AC$  (Рис.6.2-b). Но в этом случае в нашем распоряжении останется только  $(n-p-1)$  шагов. Действуя оптимальным  $(n-p-1)$ -шаговым алгоритмом, согласно все тому же индуктивному предположению (6.5), мы можем разделить отрезок  $AC$  на  $F(n-p-1)$  равных частей единичной длины, то есть,

$$AB = F(n-p-1). \quad (6.8)$$

Поскольку  $AB = AC + CB$ , то, используя результаты (6.5)-(6.8), получим следующее рекуррентное выражение для вычисления  $(n,1)$ -точности искомого оптимального  $n$ -шагового алгоритма:

$$F_p(n) = \begin{cases} n+1 & \text{для } p \geq n-1 \\ F_p(n-1) + F_p(n-p-1) & \text{для } p < n-1 \end{cases}. \quad (6.9)$$

**Частные случаи.** Рассмотрим некоторые специальные случаи формулы (6.9). Пусть  $p=0$ . Для этого случая формула (6.9) сводится к следующему рекуррентному соотношению:

$$F_0(n) = 2F_0(n-1) \quad (6.10)$$

$$F_0(1) = 2. \quad (6.11)$$

Ясно, что рекуррентная формула (6.10) с учетом начального условия (6.11) «порождает» следующую двоичную последовательность:

$$2, 4, 8, \dots, F_0(n) = 2^{n-1}.$$

Ясно также, что алгоритм измерения, соответствующий этому случаю, сводится к рассмотренному в главе 5 «двоичному» алгоритму (Рис.5.5).

Пусть  $p = \infty$ . Для сформулированной выше задачи это означает, что ИЭ «выбывает из игры», как только он оказывается справа от точки  $X$ . Для этого случая формула (6.9) принимает вид:  $F_p(n) = n + 1$ , которая соответствует рассмотренному в главе 5 алгоритму счета (Рис.5.4), широко используемому в измерительной практике.

Рассмотрим теперь случай  $p = 1$ . В этом случае рекуррентная формула (6.9) принимает следующий вид:

$$F_1(n) = F_1(n-1) + F_1(n-2) \text{ для } n > 2 \quad (6.12)$$

$$F_1(1) = 2; F_1(2) = 3. \quad (6.13)$$

Если теперь вычислить «функцию эффективности»  $F_1(n)$  согласно (6.12), (6.13), то получим следующую числовую последовательность: 2, 3, 5, 8, 13, 21, 34, ..., которая представляет собой ни что иное, как последовательность Фибоначчи, открытую Фибоначчи при решении задачи о размножении кроликов! Именно этот факт стал причиной, почему рассматриваемые алгоритмы измерения, задаваемые (6.12), (6.13), были названы «фибоначчиевыми» алгоритмами измерения [9].

В Табл.6.2 заданы значения функции эффективности  $F_p(n)$  для оптимальных «фибоначчиевых» алгоритмов для различных значений  $p$ .

Таблица 6.2. Функция эффективности  $F_p(n)$

$n$	1	2	3	4	5	6	7	8	9
$F_0(n)$	2	4	8	16	32	64	128	256	512
$F_1(n)$	2	3	5	8	13	21	34	55	89
$F_2(n)$	2	3	4	6	9	13	19	28	41
$F_3(n)$	2	3	4	5	7	10	14	19	26
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$F_{p=\infty}(n)$	2	3	4	5	6	7	8	9	10

Заметим, что числовые последовательности, задаваемые Табл.6.2, начиная с  $p \geq 2$  представляют собой новый класс числовых последовательностей, в которых



каждый член  $F_p(k)$ , начиная с  $k \geq p+1$  равен сумме двух предыдущих членов -  $F_p(k-1)$  и  $F_p(k-p-1)$ .

**Система гирь для фибоначчиевых алгоритмов измерения.** А теперь перейдем к системе гирь для «фибоначчиевых» алгоритмов измерения. Легко доказать [9], что для этого случая «оптимальная» система гирь  $W_p(n)$  задается следующей формулой:

$$W_p(n) = W_p(n-1) + W_p(n-p-1) \text{ для } n > p+1 \quad (6.14)$$

$$W_p(1) = W_p(2) = \dots = W_p(p+1) = 1. \quad (6.15)$$

Анализ выражений (6.14), (6.15) и Табл.6.3 приводит нас к неожиданному заключению. Оказывается, рекуррентное соотношение (6.14) при начальных условиях (6.15) задает числовые последовательности, полученные нами в главе 3 при исследовании так называемых «диагональных сумм» треугольника Паскаля и названных в [9]  $p$ -числами Фибоначчи!

Таблица 6.3 задает варианты «оптимальных» систем гирь, соответствующих различным значениям  $p$ .

Таблица 6.3. Оптимальные системы гирь для «фибоначчиевых» алгоритмов

$n$	1	2	3	4	5	6	7	8	9
$W_0(n)$	1	2	4	8	16	32	64	128	256
$W_1(n)$	1	1	2	3	5	21	34	55	89
$W_2(n)$	1	1	1	2	3	4	6	9	13
$W_3(n)$	1	1	1	1	2	3	4	5	7
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$W_{p=\infty}(n)$	1	1	1	1	1	1	1	1	1

## 6.5. Пример «фибоначчиевого» алгоритма измерения

Рассмотрим теперь пример «оптимального» «фибоначчиевого» алгоритма, для которого «оптимальная» система гирь задается выражениями (6.14), (6.15).

Пусть  $p = 1$  и  $n = 5$ . Рассмотрим 5-шаговый «фибоначчивый» алгоритм (Рис.6.3), соответствующий этому случаю.

Из Табл.6.2 вытекает, что рассмотренный выше 5-шаговый «фибоначчиевый» алгоритм разбивает исходный отрезок  $[0,13]$  на 13 равных частей. Чтобы реализовать такой алгоритм нам необходимо иметь 5 гирь, которые согласно Табл. 6.3 имеют следующие значения:  $\{1,1,2,3,5\}$ .

Рассмотрим первые три шага «фибоначчиевого» алгоритма, приведенного на Рис.6.3.

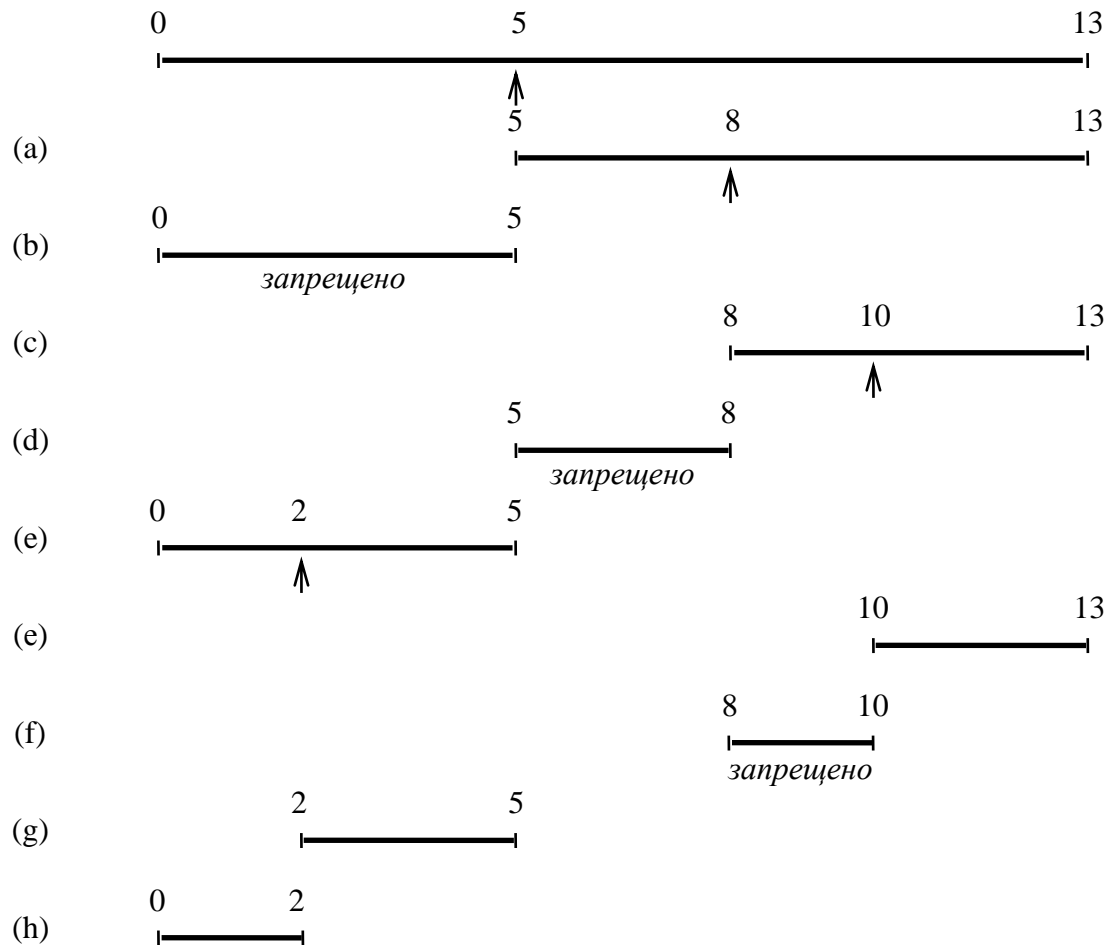


Рис.6.3. Пример «фибоначчиевого» алгоритма

Первый шаг алгоритма состоит в приложении ИЭ к точке 5 отрезка  $[0,13]$  (Рис.6.3). Можно видеть, что первый шаг состоит в разделении отрезка  $[0,13]$  в «фибоначчиевом» отношении:  $13 = 5 + 8$ . Тогда после первого шага возникают 2 ситуации (a) и (b).

Второй шаг.

(a) Для этой ситуации мы выбираем следующую по старшинству гирю 3 и с помощью ИЭ разделяем отрезок  $[5,13]$  в фибоначчиевом отношении:  $8 = 3 + 5$ . После второго шага возникают две новые ситуации (c) и (d).

(b) Для этой ситуации второй шаг является «пустым», так как согласно «ограничению»  $S$  запрещено прикладывать ИЭ к точкам отрезка  $[0,5]$  на втором шаге.

Третий шаг.

(c) В этой ситуации мы используем следующую гирю 2 и разделяем отрезок  $[8,13]$  с помощью ИЭ в «фибоначчиевом» отношении:  $5 = 2 + 3$ . Тогда после третьего шага возникают две новые ситуации (e) и (f).

(e) В ситуации (e) мы возвращаемся к ситуации (b), возникшей на 3-м шаге. В соответствии с «ограничением»  $S$  мы можем прикладывать ИЭ к точкам отрезка  $[0,5]$  на третьем шаге. При этом мы можем использовать гирю весом 2 и разделить отрезок  $[0,5]$  с помощью ИЭ в «фибоначчиевом» отношении:  $5 = 2 + 3$ . Тогда возникают две новые ситуации (g) и (h) после третьего шага.

Очень просто проследить действие «фибоначчиевого» алгоритма и на следующих двух шагах.

Таким образом, из этого примера можно усмотреть, что сущность «фибоначчиевого» алгоритма измерения состоит в последовательном разбиении «интервала неопределенности», полученного на предыдущем шаге, в «фибоначчиевом» отношении. Очень просто доказать, что это - общий принцип, который справедлив для любого значения  $p$ . При этом деление «интервала неопределенности» осуществляется согласно рекуррентному соотношению для  $p$ -чисел Фибоначчи.

## 6.6. Основной результат алгоритмической теории измерения

Как упоминалось выше, дальнейшее обобщение задачи Баше-Менделеева, основанной на «принципе асимметрии измерения» (Рис.6.1), состоит в следующем. Увеличим число «рычажных весов», участвующих в измерении, от 1 до  $k$ . Пусть один и тот же груз  $Q$  находится на левой чаше всех рычажных весов. Это соответствует случаю «параллельного» измерения груза  $Q$  с использованием  $k$  рычажных весов за  $n$  шагов. Случай «параллельных» измерений используется очень часто в аналого-цифровых преобразователях электрических величин для повышения их быстродействия. В последнем случае измеряемая электрическая величина поступает параллельно на входы  $k$  «компараторов» (электрический аналог «рычажных весов»).

Пусть все «рычажные весы», участвующие в измерении, имеют «инерционность»  $p$  ( $p = 0, 1, 2, 3, \dots$ ). Тогда возникает задача синтеза оптимального  $(n, k, S)$ -алгоритма измерения, который использует  $k$  «рычажных весов», обладающих «инерционностью»  $p$ .

Для моделирования «инерционности» рычажных весов (которым соответствуют «индикаторные элементы» в нашей модели измерения) мы вводим понятие «состояния»  $j$ -го ИЭ на  $l$ -м шаге ( $j = 1, 2, 3, \dots, k; l = 1, 2, 3, \dots, n$ ). Обозначим это состояние через  $p_j(l)$ . Состояние  $p_j(l)$  имеет следующую «физическую» интерпретацию. Если на  $l$ -м шаге  $j$ -е «рычажные весы», соответствующие  $j$ -му ИЭ, находятся в исходном состоянии «больше» (это означает, что  $j$ -й ИЭ находится слева от точки  $X$ ), то считается, что  $p_j(l) = 0$ . Если же  $j$ -е «рычажные весы» перешли на  $l$ -м шаге в противоположное состояние «меньше» (то есть,  $j$ -й ИЭ оказывается справа от точки  $X$ ), то считается, что  $p_j(l) = p$ , где  $p$  – заданное целое число, характеризующее «инерционность» рычажных весов.

Если  $p_j(l) > 0$ , то это означает, что  $j$ -е «рычажные весы» находятся в стадии перехода в исходное состояние и, следовательно,  $j$ -й ИЭ не может быть приложен к точкам отрезка  $AB$ . Начиная с  $(l+1)$ -го шага, состояние  $j$ -го ИЭ с каждым шагом уменьшается на единицу.

В общем случае на первом шаге алгоритма «индикаторные элементы» находятся в некоторых «начальных состояниях»:

$$p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k. \quad (6.16)$$

Напомним, что каждое из «начальных» состояний (6.16) на каждом шаге может принимать значение из множества

$$p_t \in \{0, 1, 2, 3, \dots, p\}; \quad t = 1, 2, 3, \dots, k. \quad (6.17)$$

При этом перед каждым шагом происходит перенумерация ИЭ таким образом, чтобы их «начальные состояния» (6.16) находились в следующих соотношениях:

$$p_1 \leq p_2 \leq \dots \leq p_t \leq p_{t+1} \leq \dots \leq p_{k-1} \leq p_k. \quad (6.18)$$

В дальнейшем  $(n, k, S)$ -алгоритм, в котором «начальные условия» ИЭ (6.16) удовлетворяют условиям (6.18), будем называть  $(n; p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k)$ -алгоритмом.

Обозначим «функцию эффективности» оптимального  $(n; p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k)$ -алгоритма через

$$F_p(n, k) = F_p(n; p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k). \quad (6.19)$$

Заметим, что на каждом шаге к отрезку  $AB$  могут прикладываться только те ИЭ, которые находятся в состоянии 0, остальные ИЭ уменьшают свои «состояния» на 1.

Пусть на первом шаге оптимального  $(n, k, S)$ -алгоритма «начальные состояния» ИЭ таковы, что

$$p_1 = p_2 = \dots = p_t = 0; \quad p_{t+1} > 0. \quad (6.20)$$

Тогда мы можем обозначить «функцию эффективности» (6.19) такого алгоритма следующим образом:

$$F_p(n, k) = F_p \left( n; \underbrace{0, 0, \dots, 0}_r, p_{t+1}, \dots, p_{k-1}, p_k \right). \quad (6.21)$$

Из определения (6.21) вытекает, что «функция эффективности» зависит не только от числа шагов алгоритма  $n$  и количества ИЭ  $k$ , но и от начальных состояний ИЭ (6.20).

При синтезе оптимального  $(n, k, S)$ -алгоритма мы должны учитывать следующие обстоятельства. Во-первых, на первом шаге к отрезку  $AB$  могут быть приложены только те ИЭ из (6.20), которые находятся в состоянии 0 (то есть, ИЭ от 1-го до  $t$ -го). Далее, после выполнения первого шага «начальные состояния» (6.20) всех остальных ИЭ (от  $(t+1)$ -го до  $k$ -го) уменьшаются на 1. Кроме того, после выполнения первого шага все ИЭ, которые оказались слева от точки  $X$ , остаются в состоянии 0, а все ИЭ, оказавшиеся справа от точки  $X$ , переходят в состояние  $p$ .

Таким образом, после совершения 1-го шага «начальные состояния» ИЭ изменяются. Например, если  $r$  из  $t$  ИЭ оказались слева от точки  $X$ , то они остаются в состоянии 0, в то время как остальные  $(t-r)$  ИЭ, которые оказались справа от точки  $X$ , переходят в состояние  $p$ . ИЭ, которые перед 1-м шагом имели состояние  $p_{t+1} > 0$ , после 2-го шага уменьшают свои состояния на 1. Если теперь упорядочить ИЭ перед совершением 2-го шага согласно (6.18), то мы получим следующий набор «начальных состояний» ИЭ на 2-м шаге:

$$\underbrace{0, 0, \dots, 0}_r, p_{t+1} - 1, \dots, p_k - 1, \underbrace{p, p, \dots, p}_{t-r}.$$

Мы не будем углубляться в решение этой довольно сложной математической задачи. Отметим только, что впервые она была решена украинскими математиками Игорем Витенько и Алексеем Стаховым еще в 1970 г. [57]. Это решение изложено также в книге [9]. В работах [9,57] показано, что «функция эффективности» (6.21) оптимального  $(n, k, S)$ -алгоритма в общем случае выражается с помощью следующего рекуррентного соотношения:

$$F_p(n, k) = F_p \left( n; \underbrace{0, 0, \dots, 0}_t, p_{t+1}, \dots, p_{k-1}, p_k \right) = \sum_{j=0}^t F_p \left( n-1; \underbrace{0, 0, \dots, 0}_j, p_{t+1}-1, \dots, p_{k-1}-1, p_k-1, \underbrace{p, p, \dots, p}_{t-j} \right) \quad (6.22)$$

при следующем начальном условии:

$$F_p \left( 1; \underbrace{0, 0, \dots, 0}_t, p_{t+1}, \dots, p_{k-1}, p_k \right) = t + 1. \quad (6.23)$$

В книге [9] проанализированы крайние частные случаи оптимального  $(n; p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k)$ -алгоритма и его «функции эффективности», задаваемой (6.22), (6.23) для различных случаев.

Пусть  $p = 0$ . Напомним, что для этого случая все ИЭ могут прикладываться к точкам отрезка  $AB$  на каждом шаге и все ИЭ после 1-го шага остаются в состоянии 0. В книге [9] показано, что для случая  $p = 0$  оптимальный  $(n; p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_{k-1}, p_k)$ -алгоритм сводится к рассмотренному в главе 5 оптимальному  $(n, k, 0)$ -алгоритму, а «функции эффективности», задаваемая (6.22), (6.23), при этом задается выражением:

$$F_0(n, k) = (k + 1)^n, \quad (6.24)$$

в частности, выражением

$$F_0(n, 1) = 2^n \quad (6.25)$$

при  $k = 1$  (классический «двоичный» алгоритм).

Пусть теперь  $p = \infty$  и «начальные состояния» ИЭ (6.20) на первом шаге алгоритма равны:

$$p_1 = p_2 = \dots = p_{k-1} = p_k = 0.$$

Показано [9], что для этого случая оптимальный  $(n, k, S)$ -алгоритм сводится к рассмотренному в главе 5 оптимальному  $(n, k, 1)$ -алгоритму («биномиальному алгоритму»), а «функция эффективности» (6.22), (6.23) задается следующим выражением:

$$F(n, k) = C_{n+k}^n = C_{n+k}^k. \quad (6.26)$$

Наконец, рассмотрим случай  $k=1$ . Показано [9], что для этого случая оптимальный  $(n, k, S)$ -алгоритм сводится к рассмотренному выше фибоначчиевому алгоритму измерения, а рекуррентная формула для «функции эффективности» (6.22) сводится к выражению (6.9).

Основной результат алгоритмической теории измерения со всеми неожиданными соотношениями демонстрируется с помощью Табл.6.4.

Таблица 6.4. Основной результат алгоритмической теории измерения

	$p = 0$	$0 < p < \infty$	$p = \infty$
$k \geq 1$	$(k+1)^n$ ↓	$\leftarrow F_p(n, k) \rightarrow$ ↓	$C_{n+k}^k = C_{n+k}^n$ ↓
$k = 1$	$2^n$	$\leftarrow F_p(n) = F_p(n-1) + F_p(n-p-1) \rightarrow$	$n+1$
	Двоичные числа	$p$ – числа Фибоначчи	Натуральные числа

Таким образом, «неожиданность» полученного результата заключается в том, что решением обобщенного варианта задачи Баше-Менделеева оказалось весьма общее рекуррентное соотношение для «функции эффективности», задаваемое (6.22), (6.23). Это рекуррентное соотношение включает в качестве частных случаев ряд известных комбинаторных формул, в частности, формулы для числа размещений с повторениями  $(k+1)^n$ , для числа сочетаний  $C_{n+k}^k = C_{n+k}^n$ , для двоичных ( $2^n$ ) и натуральных ( $n+1$ ) чисел, а также рекуррентное соотношение для  $p$ -чисел Фибоначчи.

## 6.7. Изоморфизм между алгоритмической теорией измерения и динамической теорией биологических популяций

Что такое «динамическая теория биологических популяций»? Ответ на этот вопрос мы находим к книге «Динамическая теория биологических популяций»



[101]. Книга посвящена одному из интенсивно развивающихся разделов современной математической биологии - динамической теории биологических популяций, т. е. сообществ живых организмов одного вида, связанных участием в общих процессах самовоспроизведения и выживания. В ней рассмотрены вопросы динамики численности, возрастной и половой структуры популяции при различных типах ее взаимодействия с внешней средой; исследована динамика генетического состава популяций, находящихся под воздействием различных факторов эволюции; изложен ряд постановок задач управления численностью и структурой эксплуатируемых и подавляемых популяций. Книга начинается с «задачи о размножении кроликов», к которой исторически восходит динамическая теория биологических популяций.

**Что такое изоморфизм?** Концепция изоморфизма [от греческих слов *isos* (равный) и *morphe* (форма)] широко используется в математике. Неформально, изоморфизм представляет собой своего рода соответствие между объектами, которое показывают связь между свойствами или операциями. Если существует изоморфизм между двумя структурами, мы называем эти структуры изоморфными. Слово "изоморфизм" применяется, когда две сложные структуры могут быть отражены одна на другую таким образом, чтобы каждой части одной структуры соответствовала определенная часть другой структуры.

Мы можем привести следующие физические аналогии изоморфизма:

1. Деревянный и металлический кубы. Хотя их физическая природа различна, их геометрические структуры изоморфны.
2. Big Ben в Лондоне и наручные часы. Хотя эти часы отличаются размерами, их механизмы подсчета времени изоморфны.

Мы можем привести различные примеры изоморфизма в математике. В абстрактной алгебре, используются два основных изоморфизма: изоморфизм между группами, и изоморфизм между кольцами. Существует также изоморфизм между графами в теории графов. В линейной алгебре, изоморфизм может быть определен как линейное отображение двух векторных пространств.

**Изоморфизм между «рычажными весами» и «кроликами».** В работе [58] сформулирован «принцип асимметрии измерения», который, в коечном итоге, привел к фибоначиевым алгоритмам измерения. Следует отметить, что этот принцип имеет практическое происхождение и отражает некоторые технические свойства компараторов, используемых в аналого-цифровых преобразователях. Этот принцип основывается на концепции «инерционных рычажных весов», используемых при взвешивании. Если мы обозначим через  $I$  исходное (“initial”) положение рычажных весов (Рис.6.1-а) и через  $O$  – противоположное (“opposite”) положение (Рис.6.1-б), тогда функционирование рычажных весов может быть описано с помощью двух переходов:

$$I \rightarrow \begin{cases} I \\ O \end{cases} \quad (6.27)$$

$$O \rightarrow I. \quad (6.28)$$

Переход (6.27) означает, что рычажные весы могут перейти в одно из двух крайних положений,  $I$  или  $O$ , после того, как мы положим следующую эталонную гирию на свободную чашу рычажных весов. Переход (6.28) означает следующее. Если рычажные весы оказались в положении  $O$ , тогда в течение определенного времени рычажные весы возвращаются назад в исходное положение  $I$ .

А теперь рассмотрим еще раз «задачу о размножении кроликов», сформулированную Фибоначчи. Вспомним, что «закон размножения кроликов» сводится к следующему правилу. Каждая зрелая пара кроликов  $A$  производит новорожденную пару  $B$  в течение одного месяца. Новорожденная пара становится зрелой в течение месяца и затем начинает производить одну новорожденную пару ежемесячно. Мы можем промоделировать процесс «размножения кроликов» с помощью двух переходов:

$$A \rightarrow AB \quad (6.29)$$

$$B \rightarrow A. \quad (6.30)$$

Заметим, что переход (6.29) моделирует процесс «рождения» новой пары кроликов  $B$ , а переход (6.30) – «созревание» кроликов. При этом переход (6.29) отражает асимметрию размножения кроликов, поскольку согласно (6.29) зрелая

пара  $A$  в процессе «рождения» трансформируется в две неидентичные пары, зрелую пару  $A$  и новорожденную пару  $B$ .

Сравнивая переходы (6.27), (6.28) с переходами (6.29), (6.30), мы можем увидеть аналогию или изоморфизм между ними. Более того, этот изоморфизм подтверждается тем фактом, что решение этих задач приводит к одной и той же рекурсивной числовой последовательности – числам Фибоначчи!

**Обобщенный «принцип асимметрии размножения» в живой природе.** Используя модель «размножения кроликов», которая описывается переходами (6.29) и (6.30), мы можем обобщить «задачу о размножении кроликов» следующим способом. Зададимся некоторым неотрицательным числом  $p \geq 0$  и сформулируем следующую задачу:

«Пусть в огороженном месте имеется пара кроликов (самка и самец) в первый день января. Эта пара кроликов производит новую пару кроликов в первый день февраля и затем в первый день каждого следующего месяца. Каждая новорожденная пара кроликов становится зрелой уже через  $p$  ( $p = 0, 1, 2, 3, \dots$ ) месяцев и затем еще через месяц дает жизнь новой паре кроликов. Возникает вопрос: сколько пар кроликов будет в огороженном месте через год, то есть через 12 месяцев с начала размножения?»

Ясно, что для случая  $p = 1$  сформулированный выше обобщенный вариант «задачи о размножении кроликов» совпадает с классической задачей Фибоначчи, сформулированной в 13 веке (см. глава 2).

Заметим также, что случай  $p = 0$  соответствует такой идеализированной ситуации, когда кролики становятся зрелыми сразу же после рождения. Этот случай можно промоделировать с помощью перехода:

$$A \rightarrow AA. \quad (6.31)$$

Ясно, что переход (6.31) отражает «симметрию» процесса «размножения кроликов», когда зрелая пара  $A$  превращается в две идентичные зрелые пары  $AA$ . Нетрудно показать, что для этого случая «кролики» размножаются согласно

«принципу дихотомии», то есть, количество «кроликов» ежемесячно удваивается: 1, 2, 4, 8, 16, 32, ... .

А теперь рассмотрим случай  $p > 0$ . Проанализируем новый вариант «задачи о размножении кроликов» с учетом новых правил их «размножения», когда процесс «размножения» удовлетворяет более сложной системе «переходов». Действительно, пусть  $A$  и  $B$  – пары зрелых и новорожденных кроликов, соответственно. Тогда переход (6.29), как и раньше, моделирует процесс ежемесячного появления новорожденной пары кроликов  $B$  от каждой зрелой пары  $A$ .

Проанализируем теперь процесс превращения новорожденной пары  $B$  в зрелую пару  $A$ . Поскольку процесс «созревания» новорожденной пары  $B$  протекает теперь в течение  $p$  ( $p = 0, 1, 2, 3, \dots$ ) месяцев, то этот процесс включает следующие промежуточные стадии, соответствующие каждому месяцу:

$$\begin{aligned}
 B &\rightarrow B_1 \\
 B_1 &\rightarrow B_2 \\
 B_2 &\rightarrow B_3 \quad . \\
 &\dots\dots\dots \\
 B_{p-1} &\rightarrow A
 \end{aligned}
 \tag{6.32}$$

Например, для случая  $p = 2$  процесс превращения новорожденной пары  $B$  в зрелую пару  $A$  состоит из двух промежуточных стадий:

$$B \rightarrow B_1 \tag{6.33}$$

$$B_1 \rightarrow A \tag{6.34}$$

Тогда, принимая во внимание новые «правила игры», задаваемые переходами (6.29), (6.33) и (6.34), процесс «размножения кроликов» для случая  $p = 2$  может быть представлен Табл.6.5.

Таблица 6.5. Размножение «кроликов» для случая  $p = 2$

Дата	Пары кроликов	$A$	$B$	$B_1$	$A + B + B_1$
1 января	$A$	1	0	0	1
1 февраля	$AB$	1	1	0	2
1 марта	$ABB_1$	1	1	1	3
1 апреля	$ABB_1A$	2	1	1	4
1 мая	$ABB_1AAB$	3	2	1	6
1 июня	$ABB_1AABAB_1$	4	3	2	9
1 июля	$ABB_1AABAB_1AB_1A$	6	4	3	13

Заметим, что столбец  $A$  задает число зрелых пар для каждой стадии размножения, столбец  $B$  задает число новорожденных пар, столбец  $B_1$  задает число новорожденных пар, находящихся на стадии  $B_1$ , столбец  $A + B + B_1$  задает общее число кроликов на каждой стадии размножения.

Анализ числовых последовательностей в каждом столбце

$$A : 1, 1, 1, 2, 3, 4, 6, 9, 13, 19, \dots$$

$$B : 0, 1, 1, 1, 2, 3, 4, 6, 9, 13, 19, \dots$$

$$B_1 : 0, 0, 1, 1, 1, 2, 3, 4, 6, 9, 13, 19, \dots$$

$$A + B + B_1 : 1, 2, 3, 4, 6, 9, 13, 19, \dots$$

показывает, что они подчиняются одной и той же закономерности: каждое число в последовательности равно сумме предыдущего числа и числа, отстоящего от него на 2 позиции. Однако, как мы знаем, этой закономерности подчиняются 2-числа Фибоначчи (см. главу 4)!

Исследуя эту задачу для общего случая  $p(p = 0, 1, 2, 3, \dots)$ , нетрудно прийти к заключению, что  $p$ -числа Фибоначчи, полученные нами при синтезе фибоначчиевых алгоритмов измерения, являются решением обобщенного варианта «задачи о размножении кроликов». Таким образом, они отражают «обобщенный принцип асимметрии», лежащий в основе размножения объектов живой природы!»

На первый взгляд, кажется, что приведенная выше «обобщенная задача о размножении кроликов» не имеет никакого физического смысла. Однако, мы не должны делать этот поспешный вывод! Статья [102] посвящена приложениям  $p$ -

чисел Фибоначчи для моделирования роста биологических клеток. В статье утверждается:

"При кинетическом анализе клеточного роста, обычно делается предположение, что деление клеток является симметричным, что приводит к двум идентичным дочерним клеткам. Суть полуконсервативной репликации хромосомной ДНК означает полное тождество между дочерними клетками. Однако, в бактериях, насекомых, нематодах и растениях, деление клеток является регулярно асимметричным, с пространственными и функциональными различиями между этими двумя продуктами деления»

В частности, в статье [102] анализируются модели клеточного роста, основанного на использовании 2- и 3-чисел Фибоначчи. При этом авторы статьи [102] пришли к следующему неожиданному результату:

«Бинарное деление клетки является регулярно асимметричным для большинства видов. Рост клеток при асимметричном бинарном делении может быть представлен обобщенными числами Фибоначчи... Наши модели впервые на уровне единичной клетки обеспечивают рациональную основу для возникновения чисел Фибоначчи и других рекурсивных числовых последовательностей в филлотаксисе и при структурировании в биологии, основанном на появлении регулярной асимметрии при бинарном делении ".

Таким образом, выводы статьи [102] являются убедительным подтверждением того факта, что процессы «размножения» в природе подчиняются «обобщенному принципу асимметрии», основанному на  $p$ -числах Фибоначчи.

А теперь возвратимся снова к сформулированному выше «принципу асимметрии измерения», который привел нас к фибоначчиевым алгоритмам измерения. Мы можем видеть, что переход (6.32) в «обобщенной задаче о размножении кроликов» подобен возврату рычажных весов в исходное положение в задаче Баше-Менделеева. Это означает, что «обобщенная задача о размножении кроликов» изоморфна фибоначчиевому алгоритму измерения!

Такой изоморфизм приводит нас еще к одной идее. Так как «принцип асимметрии измерения» лежит в основе основного результата алгоритмической теории измерения, мы вправе выдвинуть следующую гипотезу. Алгоритмическая

теория измерения [9], основной результат которой описывается весьма общим рекуррентным соотношением (6.22) при начальном условии (6.23), изоморфна динамической теории биологических популяций [101]. Это означает, что благодаря концепции изоморфизма мы можем использовать математические результаты алгоритмической теории измерения [9] для развития математической теории динамической теории биологических популяций [101], которая, на первый взгляд, кажется весьма далекой от алгоритмической теории измерения [9].

## **6.8. Изоморфизм между алгоритмической теорией измерения и теорией позиционных систем счисления**

Полученный выше основной результат алгоритмической теории измерения (Табл.6.4) сам по себе представляет интерес, как для комбинаторики, так и для теории чисел, однако он может привести к более глубоким выводам методологического характера, если учесть, что «при своем зарождении понятие числа, ставшее затем основой арифметики, не только имело конкретный характер, но и было неотделимо от понятия измерения, легшего позднее в основу геометрии. В процессе дальнейшего развития математики эти понятия все больше дифференцируются и вместе с тем каждый раз на новом, высшем этапе происходит их объединение» ([103], с. 16).

В этом высказывании подчеркнута связь понятия числа с понятием измерения. Это дает нам основание утверждать следующее. Алгоритмическая теория измерения затрагивает основания математики, в частности, такой слабо развитый раздел современной теории чисел, как системы счисления. В книге [9], опубликованной в 1977 г., автор написал следующее:

«Восходящее к Платону пренебрежительное отношение к «школьной» арифметике и ее проблемам, а также отсутствие какой-либо достаточно серьезной потребности к созданию систем счисления в практике вычислений, в течение ряда столетий всецело удовлетворявшейся десятичной системой, а в последние десятилетия - двоичной системой (в цифровой вычислительной технике), и могут служить объяснением того факта, что в теории чисел не уделялось должного

внимания системам счисления и в этой части она не намного ушла вперед по сравнению с периодом своего зарождения».

И с этой точки зрения алгоритмическая теория измерения [9] представляет собой необычную математическую теорию. Ее необычность состоит в том, что в ней впервые в истории математики была поставлена задача исследования алгоритмов измерения. При этом каждый «оптимальный» алгоритм измерения «порождает» свой позиционный способ представления натуральных чисел. То есть, существует изоморфизм между оптимальными  $(n, k, S)$ -алгоритмами и позиционными системами счисления. Например, оптимальные  $(n, k, 0)$ -алгоритмы изоморфны всем известным позиционным системам счисления, включая Вавилонскую 60-ричную, десятичную и двоичную системы счисления.

Оптимальные  $(n, k, 1)$ -алгоритмы изоморфны биномиальным системам счисления, которые уже переходят в область практических применений [97].

Наконец, фибоначиевые алгоритмы измерения изоморфны так называемым  $p$ -кодам Фибоначчи, получившим применение в современной информатике и измерительной технике. Эти коды мы рассмотрим в последующих главах.

Из проведенных рассуждений вытекает роль алгоритмической теории измерения [9] для развития математики и информатики.

Во-первых, мы должны признать, что в алгоритмической теории измерения [9] впервые в истории математики была предпринята попытка синтеза не только новых, неизвестных ранее алгоритмов измерения, но и изоморфных им новых, неизвестных ранее позиционных систем счисления. Благодаря новым позиционным системам счисления теория систем счисления начинает приобретать характер строгой математической теории.

Это означает, что алгоритмическая теория измерения касается одного из самых слабо развитых в теоретическом отношении раздела теории чисел, как системы счисления. Благодаря концепции изоморфизма, алгоритмическая теория измерения становится мощным источником для развития теории систем счисления как важного раздела «элементарной теории чисел».



С другой стороны, новые системы счисления, синтезированные в алгоритмической теории измерения, порождают новые компьютерные арифметики, что представляет фундаментальный интерес для теории компьютеров.

## **6.9. Международное признание алгоритмической теории измерения**

**Всесоюзная конференция ИИС-73.** В развитии алгоритмической теории измерения» можно выделить ряд событий, которые можно считать некоторыми «вехами» в ее становлении. Само понятие «алгоритмической теории измерения» впервые было введено автором в 1973 г. в докладе «Алгоритмическая теория измерения», прочитанном на Всесоюзной конференции по информационно-измерительным системам ИИС-73 (Украина, г. Ивано-Франковск). Высокая оценка этого доклада участниками конференции, в частности, выдающимся советским специалистом, одним из основателей «информационной техники» проф. Темниковым Федором Евгеньевичем, воодушевили автора на дальнейшие исследования в этом направлении и написание специальной научной книги по алгоритмической теории измерения [9].

**Доклад в Австрии.** Впервые широкая международная апробация «алгоритмической теории измерения» состоялось в 1976 г. в Австрии – сначала (23-го февраля 1976 г.) на заседании Математического института двух Грацких университетов, где по инициативе известного австрийского математика проф. Александра Айгнера автор выступил с докладом «Алгоритмическая теория измерения и основания компьютерной арифметики» и затем (3-го марта 1976 г.) в Вене, где по инициативе проф. Рихарда Эйера, зав. кафедрой обработки информации Венского технического университета, автор выступил с таким же докладом на объединенном заседании кибернетического и компьютерного обществ Австрии.

**Книга «Введение в алгоритмическую теорию измерения» (1977).** Знаковым событием в широком признании «алгоритмической теории изменения» стала публикация в 1977 г. книги Алексея Стахова «Введение в алгоритмическую теорию измерения». Книгу опубликовало в 1977 г. издательство «Советское радио» [9]. По прошествии 25 лет с момента публикации этой книги автор не только не отказался от основных идей, изложенных в ней, а еще раз убедился в правоте этих идей, которые привели автора в последние годы к концепции новой математики, "математики гармонии" [47], основанной на числах Фибоначчи, золотом сечении и алгоритмической теории измерения.

**8-й Конгресс ИМЕКО.** В мае 1979 г. в Москве состоялся 8-й Конгресс ИМЕКО (так называется Международная конфедерация по измерительной технике и приборостроению), в рамках которого состоялась Международная конференция по измерительной технике и приборостроению. Конгресс ИМЕКО стал крупным событием в международной научной жизни, а в его работе приняли участие выдающиеся ученые всего мира.

Пленарное заседание Конгресса состоялось 21 мая в знаменитом Киноконцертном зале "Россия". Центральным событием пленарного заседания стал доклад "Теоретические и физико-метрологические проблемы дальнейшего развития измерительной техники в промышленности и науке", авторами которого были проф. Ю.В. Тарбеев (НПО "ВНИИ им. Д.И. Менделеева") и проф. Д. Хофман (Университет им. Ф. Шиллера, Йена, ГДР). Доклад делал проф. Хофман. Полной неожиданностью для автора настоящей книги было многократное упоминание в докладе "алгоритмической теории измерения" как одного из новых направлений в теоретической метрологии. Анализируя состояние современной теории измерения, авторы упомянутого доклада подчеркивали, что "за последние годы предложены разные варианты общей теории измерений: и "информационная теория измерений", и "квантово-механическая (физическая) теория измерений", и "теоретико-множественная теория измерений", и "алгоритмическая теория измерений" и др."

Для автора этот случай особенно запомнился потому, что это было по существу первое признание «алгоритмической теории измерения» на таком высоком международном уровне.

**Доклад на Международном симпозиуме «Интеллектуальные измерения».** В 1986 г. автор выступил с докладом «Алгоритмическая теория измерения» на пленарном заседании Международного симпозиума «Интеллектуальные измерения», который состоялся в немецком городе Йена. На этом докладе присутствовал известный советский ученый проф. Владимир Кнеллер, который в то время был главным редактором журнала «Измерения, контроль, автоматизация». Он сразу же предложил автору написать статью по «алгоритмической теории измерения» для его журнала и такая статья вскоре после Симпозиума была опубликована в этом престижном журнале [65].

**Статья в Международном журнале “Computers & Mathematics with Applications”.** В 1989 г. Международный журнал “Computers & Mathematics with Applications” опубликовал большую статью А.Р. Stakhov. “The Golden Section in the Measurement Theory” [66]. Это была первая публикация автора на английском языке. Необходимо отметить, что этот журнал всегда пользовался и пользуется высоким авторитетом в научном мире. Рецензентами статей являются ученые с мировым именем, и поэтому публикация статьи в этом журнале также является международным признанием научного направления.

**Статья в академическом журнале «Управляющие системы и машины».** В 1994 г. Международный журнал «Управляющие системы и машины» (Академия наук Украины) опубликовал статью автора «Алгоритмическая теория измерения: новый взгляд на теорию позиционных систем счисления и компьютерную арифметику» [67]. Чтобы подчеркнуть значимость этой публикации редакция журнала вынесла название статьи «Алгоритмическая теория измерения» на переднюю обложку журнала в качестве одной из основных тем выпуска наряду с

такими темами, как «Информатизация Украины» и «Новые методы в информатике».

Таким образом, без всяких сомнений понятие «алгоритмической теории измерения», введенное автором в 1973 г., широко вошло в современную научно-техническую литературу, а «алгоритмическая теория измерения» [9] в настоящее время является признанным оригинальным научным направлением современной математики и теоретической метрологии.

## Глава 7

# ЭВОЛЮЦИЯ СИСТЕМ СЧИСЛЕНИЯ

### 7.1. Крупнейшее математическое открытие в истории математики

Каждый человек на земном шаре, окончивший хотя бы четыре класса начальной или «церковно-приходской» школы, знает, по меньшей мере, две полезные вещи: он умеет писать и читать и использовать десятичную систему счисления для выполнения простейших арифметических операций. И эта система кажется нам настолько простой и элементарной, что многие из нас с большим недоверием отнесутся к утверждению, что десятичная система является одним из крупнейших математических открытий за всю историю математики. И чтобы убедить читателя в этом, обратимся к мнению «авторитетов».

Пьер Симон Лаплас (1749-1827), французский математик, член Парижской академии наук, почетный иностранный член Петербургской академии наук:

«Мысль выражать все числа 9 знаками, придавая им, кроме значения по форме, еще значение по месту, настолько проста, что именно из-за этой простоты трудно понять, насколько она удивительна. Как нелегко было прийти к этой методе, мы видим на примере величайших гениев греческой учености Архимеда и Аполлония, от которых эта мысль осталась скрытой».

М.В. Остроградский (1801-1862), русский математик, член Петербургской академии наук и многих иностранных академий:

«Нам кажется, что после изобретения письменности самым большим открытием было использование так называемой десятичной системы счисления. Мы хотим сказать, что соглашение, с помощью которого мы можем выразить все полезные числа двенадцатью словами и их окончаниями, является одним из самых замечательных созданий человеческого гения.»

Жюль Таннери (1848-1910), французский математик, член Парижской академии наук:

«Что касается до нынешней системы письменной нумерации, в которой употребляется девять значащих цифр и ноль, и относительное значение цифр определяется особым правилом, то эта система была введена в Индии в эпоху, которая не определена точно, но, по-видимому, после христианской эры. Изобретение этой системы есть одно из самых важных событий в истории науки, и несмотря на привычку пользоваться десятичной нумерацией, мы не можем не изумляться чудной простоте ее механизма».

Создание первых систем счисления относится к периоду зарождения математики, когда потребности счета предметов, измерения времени, земельных участков и количества продуктов привели к созданию простейших понятий арифметики натуральных чисел. Как подчеркивает А.Н. Колмогоров [52], «только на основе разработанной системы устного счисления возникают письменные системы счисления и постепенно вырабатываются приемы выполнения над натуральными числами четырех арифметических действий».

В истории систем счисления выделяют несколько этапов: начальная стадия счета, непозиционные системы счисления, алфавитные системы нумерации, поместные или позиционные системы счисления. Начальная стадия счета характеризуется изображением сосчитываемых множеств при помощи частей тела, особенно пальцев рук и ног, палочек, узлов веревки и т.д. Как подчеркивается в статье [104], «несмотря на крайнюю примитивность этого способа изображения, он сыграл исключительную роль в развитии понятия числа». Таким образом, именно системы счета сыграли определяющую роль в формировании понятия натурального числа, одного из фундаментальных понятий математической науки.

К сожалению, историки компьютерной науки, владея развитой компьютерной теорией, иногда забывают о той роли, которую сыграли системы счисления в истории компьютеров. Ведь первые проекты счетных приборов (абакон и арифмометров), прообразов современных компьютеров, появились задолго до возникновения алгебры логики, теории алгоритмов – и главную роль при создании таких счетных приборов сыграли именно системы счисления и правила выполнения в них простейших арифметических операций. Об этом не следует забывать, прогнозируя дальнейшее развитие компьютерной науки.

Почему же в теории чисел и теоретической арифметике не уделялось системам счисления того внимания, которое они несомненно заслуживали. Все дело – в «традиции». В греческой математике, достигшей высокого уровня развития, впервые произошло сохранившееся до наших дней разделение математики на «высшую» математику, к которой относилась геометрия и теория чисел, и «логистику» - прикладную науку о технике арифметических вычислений («школьная» арифметика), геометрических измерениях и построениях. Уже со времени Платона «логистика» третировалась как низшая, прикладная дисциплина, не входящая в круг образования философа и ученого. Восходящее к Платону пренебрежительное отношение к «школьной» арифметике и ее проблемам, а также отсутствие какой-либо достаточно серьезной потребности в создании новых систем счисления в практике вычислений, которая в течение последних столетий всецело удовлетворялась десятичной системой, а в последние десятилетия – двоичной системой (в информатике), и может служить объяснением того факта, что в теории чисел не уделялось должного внимания системам счисления, и в этой части она не намного ушла вперед по сравнению с периодом своего зарождения.

Интерес к системам счисления сохранился только в истории математики. Хотя историки математики и утверждают, что именно проблема счета была одним из главных стимулирующих факторов в развитии математики на этапе ее зарождения, однако после изобретения десятичной системы эта проблема в математике как бы была решена «полностью и окончательно» и никаких серьезных исследований в этом направлении (кроме обсуждения проблем методики преподавания десятичной системы в средней школе) в математико-педагогической литературе не проводилось.

Ситуация резко изменилась во второй половине 20-го века после появления современных компьютеров. Именно в этой области опять проявился огромный интерес к способам представления чисел и новым компьютерным арифметикам. Кроме использования двоичной системы счисления (неймановские принципы), уже на начальном этапе «компьютерной эры» были предприняты попытки использования в компьютерах и других, отличных от двоичной, систем счисления (в этом отношении наиболее ярким примером является первый в истории

компьютеров троичный компьютер «Сетунь», спроектированный Н.П. Брусенцовым с использованием троичной симметричной системы счисления). В этот период появились системы счисления с «экзотическими» названиями и свойствами: система остаточных классов, с комплексным основанием, негепозиционная, факториальная, биномиальная системы счисления и т.д. [105]. Все они имели те или иные преимущества по сравнению с двоичной системой и были направлены на улучшение тех или иных технических характеристик компьютеров; некоторые из них стали основой для создания новых компьютерных проектов (троичный компьютер «Сетунь», компьютер, основанный на системе остаточных классов и т.д.)

Но есть и другой интересный аспект этой проблемы. Спустя 4 тысячелетия после изобретения вавилонянами позиционного принципа, в области систем счисления возник период своеобразного «ренессанса». Благодаря усилиям, прежде всего, специалистов в области компьютеров, математика как бы вновь возвратилась к периоду своего зарождения, когда именно системы счисления определяли содержание и сущность математики. Но ведь этот период, по мнению многих историков математики [52,104], считается чрезвычайно важным для развития математики. Именно в этот период были заложены основы такого важнейшего математического понятия как натуральное число и истоки самой теории чисел, которая по праву называется «царицей математики». Но тогда вполне резонным является постановка следующего вопроса: а не могут ли современные системы счисления, созданные для удовлетворения утилитарных потребностей компьютерной техники, повлиять на развитие самого понятия числа и теории чисел и таким путем оказать влияние не только на развитие компьютерной науки, но и всей математики. Поиск ответа на этот вопрос и является одной из задач настоящей книги.

## **7.2. Основные вехи в истории систем счисления**

**Вавилонская шестидесятеричная система счисления.** Уже в период зарождения математики было сделано одно из крупнейших математических



открытий в области систем счисления. Речь идет о позиционном принципе представления чисел. Как подчеркивается в статье [104], «первой известной нам системой счисления, основанной на поместном или позиционном принципе, является шестидесятеричная система древних вавилонян, возникшая примерно за 2000 лет до н.э.».

Для объяснения вопроса о ее происхождении в истории математики возникло несколько конкурирующих гипотез. По мнению одних исследователей, эта система связана с двенадцатеричной системой счисления ( $60 = 5 \times 12$ , где 5 — число пальцев на руке). Существует также гипотеза О. Нейгебауэра (1927) [106] о том, что после аккадского завоевания шумерского государства там долгое время одновременно существовали две денежно-весовые единицы: шекель (сикль) и мина, причём было установлено их соотношение 1 мина = 60 шекелей. Позднее это деление стало привычным и породило соответствующую систему записи любых чисел. Известный историк математики М. Кантор первоначально предположил, что шумерийцы (первичное население долины Евфрата) считали год равным 360 суткам и что шестидесятеричная система имеет астрономическое происхождение. По гипотезе другого историка математика Г. Кевича, в долине Евфрата встретились два народа, из которых у одного была десятичная система счисления, а у другого основанием было число 6 (возникновение такого основания Кевич объясняет особым счетом на пальцах, в котором сжатая в кулак рука означала 6). Благодаря слиянию обеих систем возникло «компромиссное» основание 60.

Вавилонское государство также унаследовало шестидесятеричную систему и передало её, вместе с таблицами наблюдений за небом, греческим астрономам. В более позднее время шестидесятеричная система использовалась арабами, а также древними и средневековыми астрономами, в первую очередь, для представления дробей. Поэтому средневековые учёные часто называли шестидесятеричные дроби «астрономическими».

В XIII веке влиятельный ректор Парижского университета Пётр Филомен выступил за повсеместное внедрение шестидесятеричной системы в Европе. В XV веке с аналогичным призывом выступил Иоганн Гмунден, профессор математики Венского университета. Обе инициативы остались без последствий.

Начиная с XVI века, десятичные дроби в Европе полностью вытесняют шестидесятеричные. Сейчас остатки шестидесятеричной системы используются в измерении углов и времени

Заметим, что гипотезы Нейгебауэра, Кантора и Кевича касаются, прежде всего, вопроса о происхождении основания 60, но не самого позиционного принципа представления чисел. На последний вопрос отвечает еще одна гипотеза Нейгебауэра об измерительном происхождении позиционного принципа. Согласно этой гипотезе [106], «основные этапы образования позиционной системы в Вавилоне были таковы: (1) установление количественного соотношения между двумя самостоятельными существовавшими системами мер и (2) опускание названий разрядовых единиц при письме». Эти этапы возникновения позиционных систем Нейгебауэр считает совершенно общими, подчеркивая при этом, что «позиционная шестидесятеричная система ... оказалась вполне естественным конечным результатом долгого развития, ничем принципиально не отличающегося от аналогичных процессов в других культурах». Кстати, гипотеза Нейгебауэра хорошо согласуется с идеей изоморфизма между алгоритмической теорией измерения и позиционными системами счисления. Согласно этой идее, алгоритмическая теория измерения [9] не только порождает все известные позиционные системы счисления (десятичную, двоичную, 60-ричную и др.), но и является источником новых систем счисления (фибоначчиевой и биномиальной).

Что касается основания 60, которое, по мнению Нейгебауэра, возникло как синтез вавилонских систем мер (1 мина = 60 шекелей), то более убедительной все же является гипотеза М. Кантора о его «астрономическом» происхождении. Как показано в главе 4 настоящей книги при обсуждении новой гипотезы происхождения древнеегипетского календаря, выбор числа 60 в качестве основания вавилонской системы счисления, а также чисел 12, 30 и 360, как «узловых чисел» всех календарных систем, а также систем измерения времени и угловых величин, можно объяснить с позиций астрологических и астрономических знаний и основанных на них представлениях о гармонии Вселенной. В главе 4 мы уже упоминали о роли числовых характеристик додекаэдра 12, 30, 60 и  $360 = 12 \times 30$  в происхождении Египетского календаря и систем измерения времени и угловых

величин. Возможно, что именно додекаэдр, который в античной науке символизировал гармонию Вселенной, и послужил причиной выбора числа 60 (число плоских углов на поверхности додекаэдра) в качестве основания Вавилонской 60-ричной системы счисления.

**Алфавитная и римская системы нумерации.** У многих народов широкое распространение получили так называемые алфавитные системы нумерации, когда буквам алфавита присваивались некоторые числовые значения. Алфавитными системами счисления пользовались древние армяне, грузины, греки (ионическая система счисления), арабы, евреи и другие народы Ближнего Востока. В славянских богослужебных книгах греческая алфавитная система была переведена на буквы кириллицы. Например, с алфавитной нумерацией связано возникновение так называемого «звериного» числа 666. Если записать на древнееврейском языке два слова «император Нерон», а затем подсчитать сумму числовых значений входящих в него букв, то она окажется равной 666. Таким образом, число 666 является скрытым обозначением имени Нерона, гонителя христианства, человека-зверя, который с крайней жестокостью расправлялся со своими политическими противниками и с ведома которого были убиты его мать, обе жены, философ Сенека и многие другие.

Славяне также использовали алфавитную нумерацию. Но чтобы отличить буквы от чисел, они ставили над буквами, изображающими числа, особый знак, названный «титло».

Всем известна римская система нумерации, которая предшествовала появлению нашей позиционной системы счисления. Она является непозиционной (точнее, частично позиционной). В ней в качестве цифр используются латинские буквы: I обозначает 1, V — 5, X — 10, L — 50, C — 100, D — 500, M — 1000. Например, II = 1 + 1 = 2; здесь символ I обозначает 1 независимо от места в числе. Элементы «позиционности» данной системы счисления состоят в том, что при некоторых записях чисел, значение цифр зависит от их позиции в этой записи. Например, IV=5-1=4, в то время как VI=5+1=6. Существенным недостатком

римской нумерации является то, что она не приспособлена для производства арифметических действий в письменном виде.

**Древнеегипетская десятичная непозиционная система счисления.**

Древнеегипетская десятичная система счисления возникла во второй половине третьего тысячелетия до н. э. Она носила иероглифический характер. Египетская система нумерации была десятичной, но непозиционной. Числа изображались в виде суммы некоторых «узловых чисел»  $1, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7$ . Для обозначения «узловых чисел» использовались специальные иероглифические знаки, которые играли роль «цифр». Например, в записи числа 325, использовалось 5 иероглифов, изображающих «узловое число» 1, 2 иероглифа, изображающего число 10, и 3 иероглифа, изображающего число 100.

Основным достижением египетской арифметики считается изобретение метода удвоения, с помощью которого они выполняли операции умножения и деления чисел. Для того, чтобы умножить число 35 на 12 египетский математик поступал следующим образом. Он составлял таблицу. В первом столбце таблицы помещались двоичные числа  $2^k$  ( $k = 0, 1, 2, 3, \dots$ ), во втором столбце первое число являлось одним из сомножителей (в данном случае 35), а каждое последующее число равнялось удвоенному предыдущему.

1	35		
2	70		
/4	140	→	140
/8	280	→	280
			$35 \times 12 = 420$

Затем наклонной чертой отмечались те двоичные числа первого столбца, сумма которых равна второму сомножителю ( $12 = 8 + 4$ ). Результат умножения получался путем сложения чисел второго столбца, соответствующих числам левого столбца.

Анализ египетского метода умножения, основанного на «принципе удвоения», приводит нас к весьма неожиданному заключению. Используемое в

первом столбце разложение целого числа по степеням двойки есть ни что иное, как его представление в двоичной системе счисления ( $12 = 1100$ ). С другой стороны, если сомножитель 35 представить в двоичной системе счисления ( $35 = 100011$ ), то используемое во втором столбце «удвоение» числа 35, осуществляемое на каждом шаге умножения, может быть осуществлено путем сдвига кода числа 35 на один разряд влево ( $70 = 1000110, 140 = 10001100$  и т. д.). Другими словами, рассмотренный выше египетский способ умножения путем «удвоения» по существу совпадает с основным алгоритмом умножения чисел в современных компьютерах!

Операция «удвоения» является основной и при делении целых чисел. Если, например, требовалось разделить число 30 на 6, то египетский вычислитель действовал следующим образом:

/1	$6 \leq 30$	$6 \leq 6$
2	$12 \leq 30$	$12 > 6$
/4	$24 \leq 30$	
8	$48 > 30$	

В первый столбец записывались двоичные числа  $2^k$  ( $k = 0, 1, 2, 3, \dots$ ), в первой строке второго столбца записывался делитель 6, а каждое последующее число второго столбца удваивалось. На каждом шаге «удвоения» числа второго столбца сравнивались с делимым 30 до тех пор, пока очередное «удвоенное» число (в данном случае число 48) не оказывалось строго больше делимого:  $48 > 30$ . После этого из делимого вычиталось «удвоенное» число предыдущего столбца ( $30 - 24 = 6$ ), а в первом столбце отмечалось наклонной чертой «двоичное» число предыдущего столбца, то есть, число 4. Затем с разностью ( $30 - 24 = 6$ ) проделывалась та же процедура, что и во втором столбце, то есть, делитель 6 и все «удвоенные» числа сравнивались с разностью 6 до тех пор, пока «удвоенное» число не оказывалось строго больше разности 6 ( $12 > 6$ ). После отметки наклонной чертой числа 1 первого столбца и вычитания делителя 6 из разности 6 получалось число 0 ( $6 - 6 = 0$ ) или число, меньшее делителя. На этом деление заканчивалось.

Сумма «двоичных» чисел первого столбца, отмеченных наклонной чертой ( $4+1=5$ ) и представляла собой результат деления.

Таким образом, нельзя не восхищаться гениальностью египетских математиков, которые много тысячелетий тому назад изобрели способы умножения и деления чисел, которые используются в современных компьютерах!

**Система счисления майя.** Появление позиционной системы обозначения чисел считается одной из основных вех в истории материальной культуры. В ее создании принимали участие целые народы. В 6 в. н.э. подобная система возникла у племени майя. Наиболее распространено мнение, что основанием системы счисления майя является число 20, имеющее «пальцевое» происхождение. Однако известно, что в системе майя есть одно отступление от двадцатеричного основания. Вес «узлового числа, следующего за «узловым» числом 20, индейцы майя выбрали равным 360 (а не  $20^2 = 400$ ). Все последующие веса разрядов являются производными от чисел 20 и 360, которые и выступают в роли «узловых» чисел, образующих систему майя. Как подчеркивается в статье [104], это «объясняется тем, что год майя делили на 18 месяцев, по 20 дней в каждом, плюс еще пять дней». Таким образом, как и основание вавилонской системы, «узловые» числа системы майя имеют астрономическое происхождение. Существенно подчеркнуть, что годовой календарь майя по своей структуре ( $360+5$ ) совпадал с египетским календарем. Учитывая высокий уровень развития культуры майя, можно высказать предположение, что майя были знакомы с Платоновыми телами и что их годовой календарь был связан с икосаэдром – правильным многогранником, двойственным додекаэдру. Как упоминалось (см. глава 4), икосаэдр представляет собой правильный 20-гранник, гранями которого были правильные треугольники (отсюда деление месяца на 20 дней в календаре майя и выбор числа 20 в качестве первого узлового числа их системы счисления). Икосаэдр имеет 30 ребер (как и у додекаэдра) и 12 вершин ( $30 \times 12 = 360$ ). В каждой вершине сходится 5 углов, то есть, общее число углов на поверхности икосаэдра равно  $5 \times 12 = 60$ . Таким образом, числовые характеристики икосаэдра также связаны с 12-, 30- и 60-летними циклами Солнечной системы.

**Десятичная система.** Мы используем для повседневных вычислений десятичную систему счисления. Древнейшая известная запись позиционной десятичной системы обнаружена в Индии в 6 в. н.э. Величайшим изобретением индийцев является введение нуля. Недавно американский математик Чарлз Сейф опубликовал книгу "Биография цифры ноль». В этой книге он следующим образом оценивает роль нуля в математике: «В цифре ноль таится намек на неопишное и невыразимое, в ней заключено беспредельное и бесконечное. Вот почему ее издавна боялись, ненавидели, а то и запрещали". На протяжении тысячелетий люди обходились без нуля: эта цифра была неведома ни египтянам, ни римлянам, ни грекам, ни древним евреям. Лишь в Индии ноль окончательно занял своё место, эта запись распространилась затем по всему миру. Индийская нумерация пришла сначала в арабские страны, затем и в Западную Европу. О ней рассказал среднеазиатский математик аль-Хорезми. Простые и удобные правила сложения и вычитания чисел, записанных в позиционной системе, сделали её особенно популярной. А поскольку труд аль-Хорезми был написан на арабском, то за индийской нумерацией в Европе закрепилось неправильное (с исторической точки зрения) название — «арабская» (арабские цифры).

В Европу десятичная нумерация проникла из Исламского Востока. Наиболее ранние рукописи на арабском языке, содержащие индийскую позиционную запись чисел, относятся к 9-му столетию нашей эры. Одним из первых в Европе понял преимущества новой нумерации французский церковнослужитель и математик Герберт, который в 999 году стал римским папою под именем Сильвестра II. Новоиспеченный папа попытался провести реформу в преподавании математики и ввести новую систему нумерации. Однако нововведение встретило яростный гнев со стороны инквизиции. Папу обвинили в том, что он «продал душу сарацинским дьяволам». Реформу постарались провалить, и папа-математик вскоре умер. Но и после смерти его не оставили в покое. Несколько столетий ходили слухи, что из мраморного саркофага папы непрерывно сочится серный дым и слышится шорох чертей.

Хотя первые записи арабско-индийскими цифрами встречаются в испанских рукописях еще в 10-м веке, десятичная система начинает закрепляться в Европе,

только начиная с 12-го века. Новая нумерация в Европе встретила сопротивление как со стороны официальной схоластической науки того времени, так и со стороны отдельных правительств. Так, например, в 1299 г. во Флоренции купцам было запрещено пользоваться новыми цифрами, в бухгалтерии приказано было либо пользоваться римскими цифрами, либо писать числа словами.

Убежденным сторонником использования арабско-индийской системы счисления в торговой практике был известный итальянский математик Леонардо из Пизы (Фибоначчи), получивший математическое образование в арабских странах. В своем сочинении “Liber abaci” (1202) он писал:

«Девять индусских знаков – суть следующие: 9, 8, 7, 6, 5, 4, 3, 2, 1. С помощью этих знаков и знака 0, который называется по-арабски “zephirum”, можно написать какое угодно число».

Здесь словом “zephirum” Фибоначчи передал арабское слово “as-sifr”, являющееся дословным переводом индусского слова “sunya”, то есть пустое, служившее названием нуля. Слово “zephirum” дало начало французскому и итальянскому слову “zero” (нуль). С другой стороны, то же арабское слово “as-sifr” было передано через “ziffer”, откуда произошли французское слово “chiffre”, немецкое “ziffer”, английское “cipher” и русское «цифра».

В начале 17-го века новая нумерация проникает в Россию, но православная церковь встречает ее в штыки и объявляет новую нумерацию колдовской и безбожной. Закрепилась десятичная нумерация в России только после издания в 1703 году знаменитой «Арифметики» Магницкого, в которой все вычисления в тексте производились исключительно с использованием десятичной системы счисления.

Существует много мнений по поводу выбора числа 10 в качестве основания десятичной системы счисления; наиболее общепринятым является мнение, что основание десятичной системы имеет «пальцевое» происхождение. Однако не следует забывать, что число 10 всегда несло в себе особую смысловую нагрузку в древней науке. Пифагорейцы называли это число четверицей или тетраксисом. В нем, говоря словами Эмпедокла, «вечно текущей природы ... корень источник». Четверица  $10=1+2+3+4$  считалась у пифагорейцев одной из высших ценностей и



являлась «символом всей Вселенной», так как она содержала в себе четыре «основных элемента»: единицу или «монаду», обозначающую по Пифагору, дух, из которого проистекает весь видимый мир; двойку или «диаду» ( $2=1+1$ ), символизирующую материальный атом; тройку или «триаду» ( $3=2+1$ ), то есть, символ живого мира. Четверка или «тетрада» ( $4=3+1$ ) соединяла живой мир с монадой и поэтому символизировала целое, то есть, «видимое и невидимое». А поскольку тетраксис  $10=1+2+3+4$ , то он выражал собой «Все». Таким образом, гипотеза о «гармоничном» происхождении числа 10 имеет не меньшее право на существование, как и «пальцевая».

**Двоичная система.** История двоичной системы довольно любопытна. В ее создании принимали участие многие народы и выдающиеся ученые. Начнем со знаменитой китайской «Книги Перемен», наиболее значительного произведения древнекитайской философии. Существующая система «Книги» сложилась в основном при Чжоужской династии. Она состоит из 64 символов — гексаграмм, каждый из которых выражает ту или иную жизненную ситуацию во времени с точки зрения её постепенного развития. Полный набор из 8 триграмм и 64 гексаграмм, который по существу является аналогом 3-битных и 6-битных цифр, был известен в древнем Китае в классических текстах «Книги Перемен». Порядок гексаграмм в «Книге Перемен», расположенных в соответствии со значениями соответствующих двоичных цифр (от 0 до 63), и метод их получения был разработан китайским учёным и философом Шао Юн в XI веке. Однако нет доказательств, свидетельствующих о том, что Шао Юн разработал правила двоичной арифметики.

Следующее упоминание о двоичной системе имеется в работах индийского поэта и математика Пингала (200 год до н.э.), который разработал математические основы для описания поэзии с использованием двоичной системы счисления.

В 1605 году Френсис Бэкон описал систему, буквы алфавита которой могут быть сведены к последовательностям двоичных цифр. Френсис Бэкон сделал важный шаг в становлении общей теории двоичного кодирования, обратив

внимание на тот факт, что указанный метод кодирования может быть использован применительно к любым объектам.

Создателем двоичной арифметики считается великий европеец, немецкий ученый Вильгельм Готфрид Лейбниц (1646-1716). Начиная со студенческих лет и до конца жизни, он занимался исследованием свойств двоичной системы счисления, ставшей в дальнейшем основной при создании компьютеров. Двоичная система была полностью описана Лейбницем в XVII веке в работе *Explication de l'Arithmétique Binaire*. Он придавал ей некий мистический смысл и считал, что на ее базе можно создать универсальный язык для объяснения явлений мира и использования во всех науках, в том числе в философии. Сохранилось изображение медали, нарисованное Лейбницем в 1697 г., поясняющее соотношение между двоичной и десятичной системами исчисления. На ней изображена табличка из двух столбцов, в одном числа от 0 до 17 в десятичной системе, а в другом – те же числа в двоичной системе счисления. Как человек, увлекающийся китайской культурой, Лейбниц знал о «Книге Перемен» и одним из первых заметил, что гексаграммы соответствуют двоичным числам от 0 до 111111. Он считал, что это является свидетельством крупных китайских достижений в философской математике того времени. Лейбниц не рекомендовал двоичную систему взамен десятичной для практического счета, но подчеркивал, что «вычисление с помощью двоек, т.е. 0 и 1, в вознаграждение его длиннот является для науки основным и даже в практике чисел, а особенно в геометрии: причиной того служит то обстоятельство, что при сведении чисел к простейшим началам, каковы 0 и 1, всюду выявляется чудесный порядок» [104]. Этим высказыванием он предвосхитил современную «компьютерную революцию», основанную на двоичной системе счисления.

В 1854 году английский математик Джордж Буль опубликовал знаковую работу, описывающую алгебраические системы применительно к логике, которая в настоящее время известна как Булева алгебра или алгебра логики. Его логическому исчислению было суждено сыграть важную роль в разработке современных цифровых электронных схем.

В 1937 году Клод Шеннон представил к защите докторскую диссертацию «Символический анализ релейных и переключательных схем». В этой диссертации булева алгебра и двоичная арифметика были использованы применительно к электронным реле и переключателям. На диссертации Шеннона по существу основана вся современная цифровая техника.

В ноябре 1937 года Джордж Штибиц, впоследствии работавший в Bell Labs, создал на базе реле компьютер «Model K» (от англ. «Kitchen», кухня, где производилась сборка), который выполнял двоичное сложение. В конце 1938 года Bell Labs развернула исследовательскую программу во главе со Штибицом. Созданный под его руководством компьютер, завершённый 8 января 1940 года, умел выполнять операции с комплексными числами. Во время демонстрации на конференции American Mathematical Society в Дартмутском колледже 11 сентября 1940 года Штибиц продемонстрировал возможность посылки команд удалённому калькулятору комплексных чисел по телефонной линии с использованием телетайпа. Это была первая попытка использования удалённой вычислительной машины посредством телефонной линии. Среди участников конференции, бывших свидетелями демонстрации, были Джон фон Нейман, Джон Мокли и Норберт Винер, впоследствии писавшие об этом в своих мемуарах.

Блестящие предсказания Лейбница относительно роли двоичной системы счисления в современной науке сбылись только через 2,5 столетия, когда выдающийся американский ученый, физик и математик Джон фон Нейман (1903-1957) после тщательного анализа достоинств и недостатков первого в компьютерной истории электронного компьютера ЭНИАК отдал решительное предпочтение двоичной системе счисления в качестве универсального способа кодирования информации в электронных компьютерах. Независимо от Неймана к этим же принципам пришел выдающийся советский ученый С.А. Лебедев, генеральный конструктор первых советских компьютеров. Поэтому в российской историко-математической литературе фундаментальные принципы построения современных электронных компьютеров называются также «принципами Неймана-Лебедева».

Идея Неймана использовать двоичную систему счисления в электронных компьютерах была обусловлена как присущими ей арифметическими преимуществами, так и спецификой электронных элементов. «Наш основной блок памяти, - писал Нейман, - по своей природе приспособлен к двоичной системе ... Триггер в сущности – опять-таки двоичное устройство ... Основное же преимущество двоичной системы по сравнению с десятичной состоит в большей простоте и быстродействии, с которыми могут выполняться элементарные операции.... Дополнительное замечание состоит в том, что основная часть машины по своему характеру является не арифметической, а логической. Новая логика, будучи системой типа «да-нет», в основном двоичная. Поэтому двоичное построение арифметических устройств существенно содействует построению более однородной машины, которая может быть лучше скомпонована и более эффективна».

Таким образом, как подчеркивают многие выдающиеся математики, открытие вавилонянами позиционного принципа, а затем индусами десятичной системы счисления, основанной на позиционном принципе, а также создание двоичной системы по праву можно отнести к разряду действительно эпохальных математических открытий за всю историю математики, существенно повлиявших на развитие материальной культуры, в частности, на развитие компьютерной техники.

### **7.3. Трои́чная симметричная система счисления и компьютер «Сетунь»**

**Понятие канонической позиционной системы счисления.** Рассмотренные выше Вавилонская 60-ричная, десятичная и двоичная системы счисления принадлежат к классу так называемых канонических позиционных систем счисления [105], которые задаются следующим общим выражением:

$$x = \sum_{i=-k}^n b_i R^i, \quad (7.1)$$

где  $x$  – некоторое действительное число (целое, дробное или смешанное),  $R$  – основание системы счисления (7.1),  $b_i$  – цифра, а  $R^i$  – вес  $i$ -го разряда. Цифры системы счисления (7.1) принимают значения из некоторого алфавита  $A = \{a_1, a_2, \dots, a_m\}$ . Для десятичной системы алфавит  $A$  состоит из 10 цифр, то есть,  $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

Сокращенная запись системы счисления (7.1) имеет вид:

$$x = b_n b_{n-1} \dots b_0, b_{-1} b_{-2} \dots b_{-k}. \quad (7.2)$$

Общепринятым в канонической системе (7.1) является использование в качестве основания  $R$  некоторого натурального числа, большего или равного 2. В этом случае особенность «канонической» системы (7.1) состоит в том, что основание  $R$  и число цифр  $m$  алфавита  $A = \{a_1, a_2, \dots, a_m\}$  совпадают, то есть  $R = m$ .

Кроме десятичной, широко употребляются в настоящее время также другие позиционные системы счисления типа (7.1), имеющие основания 2, 3, 8, 12, 16, 60:

- 2 — двоичная (в дискретной математике, информатике, программировании);
- 3 — троичная;
- 8 — восьмеричная;
- 12 — двенадцатеричная (счёт дюжинами);
- 16 — шестнадцатеричная (используется в программировании, информатике);
- 60 — шестидесятеричная (единицы измерения времени, измерение углов и, в частности, координат, долготы и широты).

**Понятие симметричной системы счисления.** Симметричные системы счисления являются обобщением понятия позиционной системы счисления (7.1); при этом это обобщение осуществляется за счет расширения понятия «цифра» в позиционном представлении. Как известно, числовые эквиваленты цифр в канонической системе счисления (7.1), как правило, являются целыми числами, принимающими значения из множества  $\{0, 1, 2, \dots, R-1\}$ , где  $R$  – основание системы счисления. Основным недостатком такой системы счисления является

невозможность представления отрицательных чисел без введения специального «знакового разряда». Как известно, для представления отрицательных чисел в компьютерах широко используются понятия «обратного» и «дополнительного» кодов. Таким образом, отрицательные числа в рамках традиционных позиционных систем счисления играют роль некоторых «изгоев», которые не могут быть представлены в традиционной системе в «прямом коде».

Для устранения указанного недостатка и были введены так называемые «симметричные» системы счисления [105]. Рассмотрим следующую позиционную систему счисления:

$$N = \sum_{i=0}^{n-1} b_i R^i, \quad (7.3)$$

где  $R = 2S + 1$ ,  $b_i \in \{\bar{S}, \overline{S-1}, \dots, \bar{1}, 0, 1, 2, \dots, S\}$ ,  $\bar{S} = -S$ .

Таким образом, «симметричная» система (7.3) имеет две основные особенности:

1. Основанием системы всегда является нечетное число  $R = 2S + 1$ , то есть  $R = 3, 5, 7, \dots$

2. Цифры системы принимают значения из множества  $b_i \in \{\bar{S}, \overline{S-1}, \dots, \bar{1}, 0, 1, 2, \dots, S\}$ ,

где  $\bar{S} = -S$ . Это означает, что симметричная система (7.3) использует  $2S + 1$  цифр; одна из этих цифр имеет числовой эквивалент 0, остальные  $2S$  цифр разбиваются на две группы: первая из них имеет положительные числовые эквиваленты  $1, 2, 3, \dots, S$ ; вторая группа цифр имеет отрицательные эквиваленты  $\bar{1}, \bar{2}, \bar{3}, \dots, \bar{S}$ .

Такой выбор множества цифр имеет существенные преимущества по сравнению с канонической позиционной системой счисления, задаваемой формулой (7.1): в симметричной системе (7.3) можно представлять отрицательные числа без введения понятий обратного и дополнительного кодов.

Нетрудно доказать [105], что в симметричной системе (7.3) можно представить с помощью  $n$  разрядов  $R^n$  различных целых чисел (положительных и отрицательных) в диапазоне от

$$N_{\min} = -\frac{R^n - 1}{2} \text{ до } N_{\max} = \frac{R^n - 1}{2}. \quad (7.4)$$

**Троичная симметричная система счисления.** Подобно тому, как двоичная система счисления, использующая цифры  $\{0,1\}$ , является наиболее простой среди канонических позиционных систем (7.1), среди симметричных систем счисления типа (7.3) наиболее простой является троичная симметричная система, обладающая следующими числовыми параметрами: основание системы равно 3, в качестве цифр используется три цифровых знака  $\{\bar{1}, 0, 1\}$ .

Для рассматриваемого случая ( $R = 3$ ) представление целых чисел в симметричной системе (7.3) принимает вид:

$$N = \sum_{i=0}^{n-1} b_i 3^i, \quad (7.5)$$

где  $b_i \in \{\bar{1}, 0, 1\}$  - троичная цифра, а  $3^i$  - вес  $i$ -го разряда.

Ясно, что сокращенные записи максимального и минимального числа в системе (7.5) имеют следующий вид, соответственно:

$$N_{\max} = \underbrace{111\dots 1}_n \quad (7.6)$$

$$N_{\min} = \underbrace{\bar{1}\bar{1}\bar{1}\dots\bar{1}}_n. \quad (7.7)$$

Цифровые записи (7.6), (7.7) имеют следующую алгебраическую интерпретацию:

$$N_{\max} = 3^{n-1} + 3^{n-2} + \dots + 3^1 + 3^0 = \frac{3^n - 1}{2} \quad (7.8)$$

$$N_{\min} = -3^{n-1} - 3^{n-2} - \dots - 3^1 - 3^0 = -\frac{3^n - 1}{2}. \quad (7.9)$$

Тогда из проведенных рассуждений вытекает, что в троичной симметричной системы счисления (7.5) с помощью  $n$  разрядов можно представить  $3^n$  целых чисел (включая, положительные, отрицательные числа и число 0) в диапазоне от

$$N_{\min} = -\frac{3^n - 1}{2} \text{ до } N_{\max} = \frac{3^n - 1}{2}.$$

**Представление отрицательных чисел.** Рассмотрим два  $n$ -разрядные числа, представленные в троичной симметричной системе (7.5):

$$A = 1b_{n-2}b_{n-3}\dots b_1b_0 \quad (7.10)$$

$$B = \bar{1}b_{n-2}b_{n-3}\dots b_1b_0. \quad (7.11)$$

Заметим, что число  $A$  содержит положительную единицу, а число  $B$  – отрицательную единицу  $\bar{1}$  в старшем разряде цифровых записей (7.10), (7.11), а цифры  $b_{n-1}, b_{n-3}, \dots, b_1, b_0$  могут принимать любые значения из множества  $\{\bar{1}, 0, 1\}$ .

Нетрудно показать, что, какими бы не были значения цифр  $b_{n-1}, b_{n-3}, \dots, b_1, b_0$ , цифровая запись числа (7.10) всегда представляет только положительное число независимо от значений остальных цифр  $b_{n-1}, b_{n-3}, \dots, b_1, b_0$  в цифровой записи (7.10). Этот вывод непосредственно вытекает из неравенства:

$$3^{n-1} > \frac{3^{n-1} - 1}{2}, \quad (7.12)$$

где  $3^{n-1}$  – вес старшего, то есть,  $(n-1)$ -го разряда в цифровой записи (7.10), а  $\frac{3^{n-1} - 1}{2}$  – сумма весов остальных разрядов в цифровой записи (7.10).

Для случая (7.11) неравенство (7.12) превращается в другое неравенство:

$$-3^{n-1} < -\frac{3^{n-1} - 1}{2}. \quad (7.13)$$

Отсюда вытекает, что цифровая запись числа (7.11) всегда представляет только отрицательное число независимо от значений остальных цифр  $b_{n-1}, b_{n-3}, \dots, b_1, b_0$  в цифровой записи (7.11).

Из этих рассуждений вытекает очень важный вывод о том, что информация о знаке числа, представленного в троичной симметричной системе (7.5), содержится в старшей значащей цифре цифровой записи числа; при этом, если старший значащий разряд числа является положительной единицей, то число является положительным, если же старший значащий разряд является отрицательной единицей ( $\bar{1}$ ), то число является отрицательным.



Таким образом, мы нашли очень важное свойство троичной симметричной системы счисления (7.5). В этой системе не требуется специального символа для представления знака числа. Положительные и отрицательные числа в этой системе представляются в «прямом» коде. При этом это свойство инвариантно относительно арифметических операций. Это означает, что все арифметические операции в системе (7.5) могут выполняться в «прямом» коде без использования понятий «обратного» и «дополнительного» кодов.

Следует отметить, что важным преимуществом троичной симметричной системы (7.5) является существование очень простого правила получения числа противоположного знака из цифрового представления исходного числа. Для этого к исходному троичному симметричному представлению применяется правило троичной инверсии, суть которого состоит в следующем:

$$\bar{1} \rightarrow 1; 0 \rightarrow 0; 1 \rightarrow \bar{1} \quad (7.14)$$

**Пример 7.1.** Пусть исходная цифровая запись десятичного числа 56 в троичной симметричной системе (7.5) имеет вид:

$$56_{10} = 1\bar{1}01\bar{1}_3.$$

Тогда из этой записи, применяя правило (7.14), легко получить число противоположного знака, то есть,

$$-56_{10} = \bar{1}101\bar{1}_3.$$

**Троичная симметричная арифметика.** В основе троичного симметричного сложения лежит следующее элементарное тождество, связывающее степени числа 3:

$$2 \times 3^i = 3^{i+1} - 3^i. \quad (7.15)$$

Из тождества (7.15) вытекает Табл.7.1, задающая правило сложения троичных симметричных одноразрядных чисел.

Таблица 7.1. Правило сложения одноразрядных чисел  $a_k + b_k$

$a_k + b_k$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}1$	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$1\bar{1}$

Из Табл.7.1 вытекает ряд особенностей троичного симметричного сложения. Эти особенности возникают при сложении единиц одного знака:

$$1+1=1\bar{1} \text{ и } \bar{1}+\bar{1}=\bar{1}1. \quad (7.16)$$

Из соотношений (7.16) вытекает, что при сложении единиц одного знака возникает перенос в старший разряд и промежуточная сумма. При этом перенос имеет тот же знак, что и знак слагаемых единиц, а промежуточная сумма имеет противоположный знак.

Троичное симметричное сложение является основной арифметической операцией в рассматриваемой системе счисления. Действительно, легко показать, что вычитание в данной системе легко сводится к сложению. Пусть требуется выполнить вычитание:  $A - B$ . Запишем разность  $A - B$  в виде:

$$A - B = A + (-B). \quad (7.17)$$

Как следует из (7.17), вычитание  $A - B$  сводится к сложению, если к вычитаемому  $B$  применить правило троичной инверсии (7.14).

Троичное симметричное умножение основывается на следующем тривиальном тождестве, связывающем степени числа 3:

$$3^m \times 3^n = 3^{m+n}. \quad (7.18)$$

Из (7.18) вытекает Табл.7.2, задающая правило умножения троичных разрядов  $a_k \times b_k$ .

Таблица 7.2. Правило умножения одноразрядных чисел  $a_k \times b_k$

$a_k \times b_k$	$\bar{1}$	0	1
$\bar{1}$	1	0	$\bar{1}$
0	0	0	0
1	$\bar{1}$	0	1

Троичное симметричное деление подобно делению в классической двоичной системе счисления и сводится к сдвигу делителя и его вычитанию из делимого.

Следующие примеры демонстрируют правила троичного симметричного сложения и умножения:

Сложение	Умножение
0 1 1 1 0 $\bar{1}$	$\bar{1}$ 0 $\bar{1}$
0 0 1 $\bar{1}$ $\bar{1}$ 1	×
<u>1 <math>\bar{1}</math> <math>\bar{1}</math> 0 <math>\bar{1}</math> 0</u>	<u>1 <math>\bar{1}</math></u>
	1 0 1
	$\bar{1}$ 0 $\bar{1}$
	<u><math>\bar{1}</math> 1 <math>\bar{1}</math> 1</u>

**Троичный компьютер «Сетунь».** Троичная симметричная система счисления (7.1) стала ключевой идеей нового принципа конструирования компьютеров [107], в основу которого положены три идеи: 1) использование троичной логики; 2) использование троичного элемента памяти; 3) использование троичной симметричной системы (7.1). С использованием этого принципа, названного троичным принципом Брусенцова в честь его создателя [83], уже на заре компьютерной эры был создан компьютер «Сетунь», первый в истории компьютерной техники троичный компьютер. Компьютер «Сетунь» был разработан на магнитных элементах, и поэтому не получил широкого практического применения, но его архитектура, основанная на «троичном принципе Брусенцова», оказалась настолько совершенной, что в настоящее время проект этого компьютера привлекает внимание многих компьютерных специалистов.

Первый образец троичного компьютера был изготовлен в Московском университете в 1958 г. Наладка компьютера была проведена в рекордно короткие сроки. Уже на десятый день комплексной наладки компьютер заработал! Такого в практике наладчиков разрабатываемых в те годы вычислительных машин еще не

было. Машину назвали "Сетунь" - по имени речки неподалеку от Московского университета.



**Николай Брусенцов**

Серийное освоение компьютера "Сетунь" было осуществлено Казанским заводом математических машин еще в 60-е годы. Было выпущено 50 экземпляров этой машины, которая надежно и продуктивно работала во всех климатических зонах от Калининграда до Магадана, от Одессы и Ашхабада до Новосибирска и Якутска. В 1961-1968 гг. на основе опыта "Сетуни" Н.П. Брусенцов вместе с известным программистом Жоголевым разработали архитектуру новой машины "Сетунь-70".

Известный советский компьютерный специалист профессор Д.А. Поспелов в свое время написал [105]:

«Барьеры, стоящие на пути приложения троичной симметричной системы счисления в компьютерах, являются препятствиями технического порядка. До сих пор не разработаны экономичные и эффективные элементы с тремя устойчивыми состояниями. Как только такие элементы будут разработаны, большая часть компьютеров универсального типа и многие специальные компьютеры по всей вероятности будут разработаны таким образом, чтобы они функционировали в троичной симметричной системе счисления».

Также известный американский ученый Дональд Кнут выразил мнение, что замена двоичного триггера ("flip-flop") на троичный триггер ("flip-flap-flop") в один прекрасный день обязательно произойдет.

Необходимо отметить, что разработки процессоров, основанных на троичной симметричной системе счисления, активно продолжаются в современной информатике. Например, в работе [108] представлена разработка цифрового

троичного процессора для выполнения быстрого преобразования Фурье. Весьма обнадеживающей для троичных компьютеров является информация о том, что группа ученых Пенсильванского университета создала из нанопроволоки память высокой емкости, которая способна сохранять троичные значения (триты) вместо повсеместно используемых двоичных значений (биты) [109]. Если исходить из оценки, что 32 бита соответствуют 20 тритам, а одно нановолокно может кодировать один трит, то при использовании троичной логики с использованием одинакового количества волокон можно записать больший объем информации. При этом, количество записанной информации увеличивается, как минимум, на 60 процентов.

Двоичная и троичная симметричные системы счисления являются наиболее яркими примерами того, как системы счисления могут повлиять на развитие информационной технологии. Двоичная система уже привела к возникновению современной информационной технологии. Есть все основания полагать, что троичная симметричная система (7.1) может привести к новой «троичной информационной технологии» или даже «троичной информационной революции» - так стремительно развивается это научное направление.

К сожалению, как и многие оригинальные советские разработки, машину "Сетунь-70" ждала трагическая судьба. Руководство Московского университета не смогло оценить революционное значение разработок Николая Брусенцова для компьютерной науки. В своей замечательной книге "История вычислительной техники в лицах" член-корреспондент Академии наук Украины Б.Н. Малиновский описывает эту трагедию так: "Создание ЭВМ - не дело университетской науки, так полагало начальство. Первое детище Брусенцова - машина "Сетунь" (экспериментальный образец, проработавший безотказно 17 лет) была варварски уничтожена, - ее разрезали на куски и выбросили на свалку".

## **7.4. Система остаточных классов**

**Элементы теории сравнений.** Основным преимуществом рассмотренных выше позиционных систем счисления является компактность цифровой записи

числа и относительная простота арифметических операций, которые сводятся к операциям над цифрами числа. Однако существенным недостатком классических позиционных систем счисления является наличие длинных цепей переноса единицы от младшего к старшему разряду, которые могут возникнуть при сложении цифровых записей чисел. Это обстоятельство необходимо учитывать при проектировании компьютеров, так как наличие «длинных переносов» снижает быстродействие процессора. Именно попытка устранить этот принципиальный недостаток позиционных систем счисления и привела к разработке оригинальной системы счисления, названной системой остаточных классов [111].

Толчком к исследованиям в этой области послужили опубликованные в 1955-1957 гг. работы чешских ученых М. Валаха и А. Свободы, посвященные представлению чисел в виде совокупности неотрицательных вычетов по группе взаимно простых оснований. Идея была подхвачена советскими учеными И.Я. Акушским и Д.И. Юдицким, опубликовавшими книгу [111], фундаментальный труд по новой машинной арифметике.

В основе «системы остаточных классов» лежит теоретико-арифметический метод, созданный еще Гауссом и называемый методом сравнений. Рассмотрим суть этого метода на конкретном примере. Для этого рассмотрим множество натуральных чисел. Выберем теперь некоторое натуральное число, например 5, которое будем называть модулем.

Разделим теперь некоторое натуральное число  $a$  на «модуль» 5 с «остатком». Ясно, что «остаток»  $r$  принимает значение из множества  $\{0, 1, 2, 3, 4\}$ . Теперь разделим все натуральные числа на 5 классов в зависимости от значения остатка  $r$ , возникающего при делении натуральных чисел на модуль 5:

$$\begin{aligned}
 r = 1: & 1, 6, 11, 21, \dots \\
 r = 2: & 2, 7, 12, 17, 22, \dots \\
 r = 3: & 3, 8, 13, 18, 23, \dots \\
 r = 4: & 4, 9, 14, 19, 24, \dots \\
 r = 0: & 0, 5, 10, 15, 20, \dots
 \end{aligned}
 \tag{7.19}$$

Числа в каждом классе образуют арифметическую прогрессию с формулами общего члена соответственно:

$$a_n = 5n - 4$$

$$a_n = 5n - 3$$

$$a_n = 5n - 2$$

$$a_n = 5n - 1$$

$$a_n = 5n$$

Эти прогрессии разбивают все натуральные числа на 5 непересекающихся классов, то есть, каждое натуральное число входит только в один класс. Будем называть такие прогрессии классами по модулю 5. В общем случае можно ввести понятие о классах по любому другому натуральному модулю  $m$ . Числа, входящие в класс, называются вычетами по модулю  $m$ . Если из каждого класса выбрать по одному вычету, то их совокупность будет называться полной системой вычетов по модулю  $m$ . Обычно в полную систему вычетов выбирают из каждого класса наименьшие числа. Поэтому для заданного модуля  $m$  полная система вычетов содержит следующие числа:  $\{0, 1, 2, \dots, m-1\}$ .

Над вычетами можно выполнять арифметические операции по заданному модулю  $m$ . Например, сложение двух вычетов сводится к обычному десятичному сложению, но сумма заменяется вычетом по модулю  $m$ . То же самое касается операции умножения вычетов по модулю  $m$ : над вычетами осуществляется традиционное десятичное умножение, а произведение заменяется вычетом по модулю  $m$ . Ниже (Табл.7.3 и 7.4) задают правила сложения и умножения вычетов по модулю 5 с учетом введенного выше определения этих операций.

Таблица 7.3. Сложение вычетов по модулю 5

$\oplus$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Примеры сложения вычетов по модулю 5:

$$3 \oplus 4 \equiv 2(\text{mod } 5); \quad 4 \oplus 11 \equiv 0(\text{mod } 5).$$

Таблица 7.4. Умножение вычетов по модулю 5

⊗	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	2	4	2
4	0	4	3	2	1

Примеры умножения вычетов по модулю 5:

$$4 \otimes 3 \equiv 2 \pmod{5}; \quad 3 \otimes 3 \equiv 4 \pmod{5}$$

Таблицы 7.3 и 7.4 по существу задают арифметику вычетов по модулю 5. Легко убедиться, что оба действия обладают переместительным и сочетательным свойствами, а операция умножения вычетов обладает и распределительным свойством относительно сложения.

**Представление чисел в системе остаточных классов.** Суть «системы остаточных классов» (СОК) состоит в представлении натурального числа в виде совокупности вычетов по некоторому множеству взаимно простых модулей:

$$p_1, p_2, \dots, p_i, \dots, p_k. \quad (7.20)$$

Тогда представлением натурального числа  $A$  в СОК является следующая совокупность вычетов

$$A = (a_1, a_2, a_3, \dots, a_k), \quad (7.21)$$

где  $a_i = A - \left[ \frac{A}{p_i} \right] \times p_i; i = 1, 2, 3, \dots, k; \left[ \frac{A}{p_i} \right]$  - целая часть частного  $\frac{A}{p_i}$ .

Если, например,  $p_1 = 2, p_2 = 3, p_3 = 5$ , тогда число 11 может быть представлено следующими вычетами:  $11 = (1, 2, 1)$ .

Ясно, что для другого варианта «модулей» (7.20), например  $(p_1 = 2, p_2 = 3, p_3 = 7)$  число 11 имеет другое представление в СОК, а именно:  $11 = (1, 1, 3)$ .



Заметим, что множество модулей (7.20) определяет диапазон  $N$  однозначного представления чисел в СОК. Этот диапазон равен произведению всех модулей, то есть,

$$N = p_1 \times p_2 \times p_3 \times \dots \times p_k. \quad (7.22)$$

Следовательно, имея модули  $p_1=2$ ,  $p_2=3$ ,  $p_3=5$ , мы можем представить 30 натуральных чисел в диапазоне от 0 до  $N-1=29$ , а используя модули  $p_1=2$ ,  $p_2=3$ ,  $p_3=7$ , мы можем представить 42 натуральных числа в диапазоне от 0 до 41.

**Использование системы остаточных классов.** Мы не будем более подробно освещать правила выполнения арифметических операций в СОК. Попытаемся ответить на вопрос; каково же главное предназначение СОК применительно к процессорам и компьютерам? В 60-е годы И.Я. Акушский и Д.И. Юдицкий в своей книге «Машинная арифметика в остаточных классах» [111] отвечают на этот вопрос следующим образом:

«Следует отметить, что позиционные системы счисления, в которых представляется и обрабатывается информация в современных вычислительных машинах, обладает существенным недостатком – наличием межразрядных связей, которые накладывают свой отпечаток на способы реализации арифметических операций, усложняют аппаратуру и ограничивают быстродействие. Поэтому естественно изыскание возможностей такой арифметики, в которой бы межразрядные связи отсутствовали. Оказалось, что такая арифметика может быть построена на базе непозиционной системы счисления, в частности системы остаточных классов».

Таким образом, главной побудительной причиной разработки машинной арифметики в СОК явилось преодоление серьезного недостатка позиционных систем счисления, в частности, двоичной системы – наличие межразрядных связей при выполнении арифметических операций, что принципиально снижало быстродействие компьютеров.

Важно подчеркнуть, что Акушский и Юдицкий были не только теоретиками СОК, но и принимали активное участие в реализации компьютера на основе СОК. Такой компьютер начал создаваться в Советском Союзе еще в 1957 г.

В компьютере, основанном на СОК, была достигнута рекордная производительность – 1.25 млн. операций в секунду, что почти на порядок превышало производительность компьютеров того времени, которые проектировались на основе классической двоичной системы счисления.

Будучи весьма перспективной в отношении межразрядных связей, система остаточных классов оказалась далеко не безупречной в других отношениях, на что обращают внимание Акушский и Юдицкий в своей книге [111]:

«Однако столь удобной в одном отношении системе остаточных классов присущ ряд недостатков в других отношениях: ограниченность действия этой системы полем целых положительных чисел, трудность определения соотношений чисел по величине, определения выхода результата операции из диапазона и т.д.»

И хотя авторам книги [111] удалось существенно продвинуться в преодолении этих трудностей, но полностью разрешить их не удалось. «Компьютерной революции» на основе СОК не получилось, то есть, проектировщики компьютеров отдавали все же предпочтение классической позиционной двоичной системе счисления, то есть, позиционный принцип представления чисел, одно из величайших изобретений математической мысли, остался непоколебимым. Преодоление недостатков, присущих двоичной системе счисления, надо было искать в рамках позиционного представления. И появление систем счисления с иррациональными основаниями, названных кодами Фибоначчи и кодами золотой пропорции [9,12,55,59-68] стало дальнейшим шагом в развитии машинной арифметики.

## Глава 8

# ФИБОНАЧЧИЕВЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

### 8.1. Тьюринг и числа Фибоначчи

Алан Матисон Тьюринг (1912—1954) — английский математик, логик, криптограф, изобретатель машины Тьюринга. Алан Тьюринг имеет широкую и по-прежнему возрастающую репутацию одного из наиболее творческих мыслителей 20-го века. Его научные интересы - от теоретической информатики до искусственного интеллекта и биологии - охватывают многие новые исследовательские темы 21-го века. В ознаменование выдающихся достижений Алана Тьюринга в области теоретической информатики учреждена премия Тьюринга (Turing Award) - самая престижная премия в информатике, вручаемая Ассоциацией вычислительной техники за выдающийся научно-технический вклад в этой области.



**Алан Тьюринг (1912—1954)**

Тьюринг учился в Шерборнской школе, где проявил незаурядные способности к математике и химии, затем в Королевском колледже Кембриджского университета, который окончил в 1934. Непосредственным его учителем, а впоследствии коллегой был математик Ньюмен (1897-1984); Тьюринг слушал его курс по основаниям математики в 1935. В том же году Тьюринг получил стипендию Королевского колледжа для работы над диссертацией. Именно в этот период он опубликовал ключевую статью с изложением того научного открытия, которое мы теперь называем машиной Тьюринга - фундаментальной идеи в

области информатики и математической логики. В 1936-1938 он стажировался в Принстонском университете в США, где его научным руководителем был американский логик А.Чёрч (1903-1995). После получения докторской степени Тьюринг отклонил предложение Джона фон Неймана остаться в США и вернулся в Кембридж, где получил стипендию Королевского колледжа для занятий логикой и теорией чисел, посещая одновременно семинары Л.Витгенштейна по философии математики.

Во время Второй Мировой войны Тьюринг работал в Блечли Парке — британском криптографическом центре, где возглавлял одну из исследовательских групп, занимавшихся расшифровкой закодированных немецкой шифровальной машиной «Энигма» сообщений Кригсмарине и Люфтваффе. Читая закодированные немецкие сообщения, можно сделать заключение, что в марте 1943 года Великобритания стояла на грани поражения в битве за Атлантику и во всей Второй мировой войне. Вполне вероятно, что без расшифровки кода «Энигмы» ход этой войны был бы иным.

После того как Джон фон Нейман в США предложил план создания компьютера EDVAC, аналогичные работы были развернуты в Великобритании в Национальной физической лаборатории, где Тьюринг проработал с 1945 по 1948. Ученый предложил весьма амбициозный проект ACE (Automatic Computing Engine - Автоматическая Вычислительная Машина), который, однако, так и не был реализован. 1947-1948 академический год Тьюринг провел в Кембридже, а в мае 1948 М.Ньюмен предложил ему пост преподавателя и заместителя директора вычислительной лаборатории Манчестерского университета, занявшего к этому времени лидирующие позиции в разработке вычислительной техники в Великобритании. В 1951 Тьюринг был избран членом Лондонского королевского общества.

Менее широко известно, что кроме исследований в области теоретической информатики Тьюринг увлекался решением еще одной научной проблемы, не имеющей, на первый взгляд, прямого отношения к информатике. Речь идет о филлотаксисе — широко известном ботаническом явлении, лежащем в основе процессов формообразования многих ботанических объектов. Когда и где Тьюринг

проявил интерес к проблеме филлотаксиса? Ответ на этот вопрос можно найти в статье [112]. В статье утверждается, что еще в школе Тьюринг ознакомился с классической книгой «On Growth and Form», опубликованной в 1917 г. английским биологом и математиком D'Arcy Wentworth Thompson (1860-1948). В этой книге детально рассматривается явление филлотаксиса. Когда Тьюринг вернулся из США в Кембридж (1947-1948), он посещал лекции по физиологии и именно тогда сделал первые попытки дать логическое описание нервной системы и продолжил свои исследования по филлотаксису. Первая статья Тьюринга на эту тему [113] опубликована в 1952 г. Позже, уже после его смерти в 1954 г., его последователи обработали и опубликовали в 1992 г. его вторую статью на эту тему [114].

Таким образом, мы можем сформулировать основные научные задачи, решение которых принесли Тьюрингу славу создателя теоретической информатики и крупного ученого-мыслителя 20-го века:

1. Машина Тьюринга, которая является расширением конечного автомата и, согласно тезису Чёрча–Тьюринга, способна имитировать любую абстрактную вычислительную машину, реализующую процесс пошагового вычисления.
2. Расшифровка криптографического кода немецкой шифровальной машины «Энигма», что существенно повлияло на ход Второй Мировой войны.
3. Проект Автоматической Вычислительной Машины ACE (Automatic Computing Engine)
4. Исследования по филлотаксису

Исследования Тьюринга по филлотаксису, конечно, можно рассматривать как «хобби» Тьюринга, не имеющее прямого отношения к его главным результатам в области информатики и вычислительной техники. Однако, не следует забывать, что Алан Тьюринг являлся одним из выдающихся ученых-мыслителей 20-го века. И его исследования по филлотаксису, связанные с исследованиями по созданию логической модели мозга – уникальной естественной вычислительной машины, нельзя рассматривать иначе, как гениальное предчувствие использования естественной «математики природы» для создания вычислительных машин будущего.

## 8.2. $P$ -коды Фибоначчи

Подобно тому, как «двоичный» алгоритм измерения порождает двоичную систему счисления, «фибоначчиевые» алгоритмы измерения [9], порождают новые способы позиционного представления натуральных чисел:

$$N = a_n F_p(n) + a_{n-1} F_p(n-1) + \dots + a_i F_p(i) + \dots + a_1 F_p(1), \quad (8.1)$$

где  $a_i \in \{0,1\}$  – двоичная цифра  $i$ -го разряда позиционного представления (8.1);  $n$  – разрядность кода (8.1);  $F_p(i) (i=1,2,3,\dots,n)$  – вес  $i$ -го разряда, равный  $i$ -му  $p$ -числу Фибоначчи.

В этом позиционном представлении веса разрядов  $F_p(i)$  связаны следующим рекуррентным соотношением:

$$F_p(i) = F_p(i-1) + F_p(i-p-1); \quad F_p(1) = F_p(2) = \dots = F_p(p+1) = 1. \quad (8.2)$$

Сокращенная запись суммы (8.1) имеет следующий вид:

$$N = a_n a_{n-1} \dots a_i \dots a_1. \quad (8.3)$$

Запись суммы (8.1) в виде (8.3) мы будем называть фибоначчиевой цифровой записью (ФЦЗ) числа  $N$ .

Заметим, что сумма (8.1) включает в себя бесконечное число различных позиционных представлений, потому что каждое  $p (p=0,1,2,3,\dots)$  «порождает» свое собственное позиционное представление типа (8.1).

Пусть  $p=0$ . Для этого случая 0-числа Фибоначчи  $F_0(i)$  совпадают с «двоичными» числами, то есть,  $F_0(i) = 2^{i-1}$  и поэтому сумма (8.1) принимает вид классического двоичного кода:

$$N = a_n 2^{n-1} + a_{n-1} 2^{n-2} + \dots + a_i 2^{i-1} + \dots + a_1 2^0. \quad (8.4)$$

Пусть  $p=1$ . Для этого случая 1-числа Фибоначчи  $F_1(i)$  совпадают с классическими числами Фибоначчи, то есть,  $F_1(i) = F_i$  и для этого случая сумма (8.1) принимает следующий вид:

$$N = a_n F_n + a_{n-1} F_{n-1} + \dots + a_i F_i + \dots + a_1 F_1. \quad (8.5)$$

Напомним, что веса разрядов  $F_i$  в сумме (8.5) связаны рекуррентным соотношением Фибоначчи:

$$F_i = F_{i-1} + F_{i-2}; \quad F_1 = F_2 = 1. \quad (8.6)$$

Именно на этом основании позиционное представление (8.5) называется кодом Фибоначчи, а более общее представление (8.1) –  $p$ -кодом Фибоначчи [9].

Важно, однако, подчеркнуть, что представление (8.5) в теории чисел Фибоначчи [3] связывают с так называемыми суммами Цекендорфа, названными так в честь бельгийского любителя математики Эдуарда Цекендорфа (1901-1983).

Цекендорф первым начал исследовать представление (8.5) и при этом еще в 1939 г. доказал теорему Цекендорфа. Эта теорема утверждает, что каждое натуральное число может быть представлено единственным образом в виде суммы (8.5), состоящей из одного или нескольких различных чисел Фибоначчи таким образом, чтобы сумма (8.5) не включала два соседних числа Фибоначчи. Например,  $100 = 89 + 8 + 3$ . При этом существуют другие представления числа 100 в виде (8.5), которые не удовлетворяют этому правилу, например,  $100 = 89 + 8 + 2 + 1$ ,  $100 = 55 + 34 + 83$ . Однако, эти «фибонассиевые» представления не являются суммами Цекендорфа.

Пусть теперь  $p = \infty$ . В этом случае все  $p$ -числа Фибоначчи (8.2) тождественно равны 1, то есть, для любого  $i$  имеем:  $F_p(i) = 1$ . В этом случае сумма (8.1) принимает следующий вид:

$$N = \underbrace{1 + 1 + \dots + 1}_N. \quad (8.7)$$

Таким образом,  $p$ -коды Фибоначчи (8.1) являются широким обобщением двоичного кода (8.4), соответствующего случаю  $p = 0$ . Частными случаями  $p$ -кодов Фибоначчи являются код Фибоначчи (8.5) ( $p = 1$ ) и «унитарный» код (8.7) ( $p = \infty$ ), выражение для которого совпадает с «Евклидовым» определением натурального числа, которое лежит в основе элементарной теории чисел.

Существенно подчеркнуть важную роль алгоритмической теории измерения [9] в создании  $p$ -кодов Фибоначчи. Ясно одно. Не будь принципа асимметрии измерения, сформулированного автором при анализе задачи Баше-Менделеева [58],

и вытекающих из этого принципа фибоначчиевых алгоритмов измерения, вполне возможно, что  $p$ -коды Фибоначчи вообще никогда не появились бы в современной математике и информатике. Но ведь  $p$ -код Фибоначчи, задаваемый формулой (8.1), является обобщением двоичной системы (8.4)! Благодаря  $p$ -коду Фибоначчи (8.1) мы теперь знаем, что существует бесконечное число «двоичных» позиционных представлений натуральных чисел, которые задаются некоторой общей формулой (8.1)! И классическая двоичная система (8.4) является лишь частным случаем  $p$ -кода Фибоначчи (8.1).

Но не следует забывать, что двоичная система (8.4) является основой современных компьютеров! Но тогда возникает вопрос: если мы будем использовать  $p$ -код Фибоначчи (8.1) для представления чисел, то, возможно, мы придем к новым компьютерам - компьютерам Фибоначчи, как новому направлению в развитии компьютерной техники! Об этом более детально мы расскажем ниже.

### 8.3. Особенности «фибоначчиевых» представлений

**Многозначность представления чисел.** Представление чисел в  $p$ -коде Фибоначчи (8.1) и классическом коде Фибоначчи (8.5) является многозначным, то есть, каждая двоичная комбинация (8.3), которая является цифровой записью  $p$ -кода Фибоначчи (8.1), представляет некоторое натуральное число  $N$ , в то время как одному и тому же натуральному числу  $N$  соответствует некоторое множество двоичных комбинаций. Для случая классического кода Фибоначчи (8.5) это свойство демонстрируется с помощью Таблиц 8.1 и 8.2.

Табл.8.1 задает отображение множества 5-разрядных ФЦЗ  $A_i (i = 0, 1, 2, 3, \dots, 31)$  на множество целых чисел: 0, 1, 2, 3, ..., 12. Табл.8.2 задает отображение множества целых чисел 0, 1, 2, 3, ..., 12 на множество 5-разрядных ФЦЗ  $A_i (i = 0, 1, 2, 3, \dots, 31)$ .



Таблица 8.1

$A_i$	5	3	2	1	1	$N$	$A_i$	5	3	2	1	1	$N$
$A_0$	0	0	0	0	0	0	$A_{16}$	1	0	0	0	0	5
$A_1$	0	0	0	0	1	1	$A_{17}$	1	0	0	0	1	6
$A_2$	0	0	0	1	0	1	$A_{18}$	1	0	0	1	0	6
$A_3$	0	0	0	1	1	2	$A_{19}$	1	0	0	1	1	7
$A_4$	0	0	1	0	0	2	$A_{20}$	1	0	1	0	0	7
$A_5$	0	0	1	0	1	3	$A_{21}$	1	0	1	0	1	8
$A_6$	0	0	1	1	0	3	$A_{22}$	1	0	1	1	0	8
$A_7$	0	0	1	1	1	4	$A_{23}$	1	0	1	1	1	9
$A_8$	0	1	0	0	0	3	$A_{24}$	1	1	0	0	0	8
$A_9$	0	1	0	0	1	4	$A_{25}$	1	1	0	0	1	9
$A_{10}$	0	1	0	1	0	4	$A_{26}$	1	1	0	1	0	9
$A_{11}$	0	1	0	1	1	5	$A_{27}$	1	1	0	1	1	10
$A_{12}$	0	1	1	0	0	5	$A_{28}$	1	1	1	0	0	10
$A_{13}$	0	1	1	0	1	6	$A_{29}$	1	1	1	0	1	11
$A_{14}$	0	1	1	1	0	6	$A_{30}$	1	1	1	1	0	11
$A_{15}$	0	1	1	1	1	7	$A_{31}$	1	1	1	1	1	12

Таблица 8.2

$0 = \{A_0\}$
$1 = \{A_1, A_2\}$
$2 = \{A_3, A_4\}$
$3 = \{A_3, A_4, A_8\}$
$4 = \{A_7, A_9, A_{10}\}$
$5 = \{A_{11}, A_{12}, A_{16}\}$
$6 = \{A_{13}, A_{14}, A_{17}, A_{18}\}$
$7 = \{A_{15}, A_{19}, A_{20}\}$
$8 = \{A_{21}, A_{22}, A_{24}\}$
$9 = \{A_{23}, A_{25}, A_{26}\}$
$10 = \{A_{27}, A_{28}\}$
$11 = \{A_{29}, A_{30}\}$
$12 = \{A_{31}\}$

Как вытекает Табл.8.1, с помощью 5-разрядных ФЦЗ  $A_i$  ( $i = 0, 1, 2, 3, \dots, 31$ ) в коде Фибоначчи (8.5) можно представить 13 целых чисел от минимального числа  $N_{\min} = 0$  до максимального числа  $N_{\max} = 12$ . При этом, как следует из Табл.8.2, все числа (за исключением минимального числа  $0=00000$  и максимального числа  $12=11111$ ) имеют многозначное кодовое представление.

**Свертка и развертка «фибоначчиевых» разрядов.** Таким образом, основной особенностью «фибоначчиевых» сумм (8.1) и (8.5) является многозначность представления одного и того же числа. Различные ФЦЗ одного и того же числа  $N$  в кодах (8.1) и (8.5) для случая  $p > 0$  могут быть получены одно из другого путем специфических преобразований над двоичными «фибоначчиевыми» разрядами, называемыми сверткой и разверткой «фибоначчиевых» разрядов. Эти кодовые преобразования выполняются в рамках одной и той же ФЦЗ (8.3), представляющей суммы (8.1) и (8.5), и основаны на рекуррентных соотношениях (8.2) и (8.6), связывающих веса разрядов в суммах (8.1) и (8.5).

В дальнейшем мы ограничимся рассмотрением кода Фибоначчи (8.5), весами разрядов которого являются классические числа Фибоначчи (8.6). Для этого случая операция свертки выполняется над группой из трех «фибоначчиевых» разрядов  $a_i a_{i-1} a_{i-2} = 011$ . Свертка состоит в замене тройки «фибоначчиевых» разрядов своими отрицаниями, то есть,

$$[011 \rightarrow 100]. \quad (8.8)$$

Операция развертки выполняется над группой «фибоначчиевых» разрядов  $a_i a_{i-1} a_{i-2} = 100$  и состоит в замене тройки «фибоначчиевых» разрядов своими отрицаниями, то есть,

$$[100 \rightarrow 011]. \quad (8.9)$$

Основное свойство этих операций состоит в том, что их выполнение в «фибоначчиевой» цифровой записи (8.3) не приводит к изменению числа, представляемого этой цифровой записью. Ниже приведены примеры применения операций свертки и развертки для получения различных ФЦЗ чисел 7 и 5:

$$\text{Свертка : } 7 = \begin{cases} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{cases} \quad \text{Развертка : } 5 = \begin{cases} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{cases} \quad (8.10)$$

Если в ФЦЗ (8.3) выполнить все возможные свертки, то мы придем к характерной ФЦЗ, в которой две единицы рядом не встречаются (см. нижнюю ФЦЗ числа 7). Такую ФЦЗ будем называть минимальной формой ФЦЗ числа. В такой ФЦЗ содержится наименьшее число единиц по сравнению с другими ФЦЗ одного и того же числа. Заметим, что младший разряд минимальной формы всегда равен 0.

Если в ФЦЗ (8.3) выполнить все возможные развертки, то мы придем еще к одной характерной ФЦЗ, в которой два нуля рядом справа от старшего значащего разряда не встречаются (см. нижнюю ФЦЗ числа 5). Такую ФЦЗ будем называть максимальной формой ФЦЗ числа. В такой ФЦЗ содержится наибольшее число единиц по сравнению с другими ФЦЗ одного и того же числа. Заметим, что младший разряд максимальной формы всегда равен 1.

## 8.4. Диапазоны представления чисел в $p$ -коде Фибоначчи

С помощью  $n$ -разрядного  $p$ -кода Фибоначчи (8.1) можно представить определенное количество целых чисел в некотором диапазоне. Наименьшее число этого диапазона равно

$$N_{\min} = 0 = \underbrace{00\dots 0}_n, \quad (8.11)$$

а максимальное число равно

$$N_{\max} = \underbrace{11\dots 1}_n. \quad (8.12)$$

ФЦЗ (8.12) является сокращенным представлением следующей суммы:

$$N_{\max} = F_p(n) + F_p(n-1) + \dots + F_p(i) + \dots + F_p(1). \quad (8.13)$$

В главе 3 при изучении свойств  $p$ -чисел Фибоначчи мы установили, что сумма  $p$ -чисел Фибоначчи в правой части соотношения (8.13) равна  $F_p(n+p+1)-1$ , откуда вытекает:

$$N_{\max} = F_p(n+p+1)-1. \quad (8.14)$$

Из проведенных рассуждений вытекает следующая теорема.

**Теорема 8.1.** С помощью  $n$ -разрядного  $p$ -кода Фибоначчи (8.1) можно представить  $F_p(n+p+1)$  целых чисел в диапазоне  $[0 \div F_p(n+p+1)-1]$ .

Напомним, что при  $p=0$ ,  $F_0(i) = 2^{i-1}$  и  $p$ -код Фибоначчи (8.1) сводится к двоичному коду (8.4). Как известно, максимальное  $n$ -разрядное двоичное число равно  $2^n - 1$ , что является частным случаем выражения (8.14), соответствующего случаю  $p=0$ .

При  $p=1$   $p$ -код Фибоначчи (8.1) сводится к коду Фибоначчи (8.5), весами разрядов которого являются классические числа Фибоначчи. Для этого случая максимальное  $n$ -разрядное число, которое может быть представлено в коде (8.5), равно

$$N_{\max} = F_1(n+2)-1 = F_{n+2}-1. \quad (8.15)$$

При этом Теорема 8.1 выглядит следующим образом.

**Теорема 8.2.** С помощью  $n$ -разрядного кода Фибоначчи (8.5) можно представить  $F_{n+2}$  целых чисел в диапазоне  $[0 \div F_{n+2} - 1]$ .

**Диапазон представления чисел в минимальной форме.** Важную роль в теории  $p$ -кодов Фибоначчи играет следующее утверждение, доказанное в [9].

**Теорема 8.3.** Для заданного целого  $p \geq 0$  любое натуральное число  $N$  может быть единственным способом представлено в виде:

$$N = F_p(n) + r, \quad (8.16)$$

где  $F_p(n)$  – некоторое  $p$ -число Фибоначчи, а  $r$  – остаток, удовлетворяющий следующему неравенству:

$$0 \leq r < F_p(n-p). \quad (8.17)$$

Заметим, что при  $p = 0$   $F_0(n) = 2^{n-1}$  и выражения (8.16) и (8.17) сводятся к следующим соотношениям, которые лежат в основе двоичной системы счисления (8.4):

$$N = 2^{n-1} + r; \quad 0 \leq r < 2^{n-1}. \quad (8.18)$$

При  $p = 1$   $F_1(n) = F_n$  (число Фибоначчи) и выражения (8.16) и (8.17) сводятся к следующим выражениям:

$$N = F_n + r; \quad 0 \leq r < F_{n-1}. \quad (8.19)$$

Раскладывая натуральное число  $N$  в соответствии с выражениями (8.16) и (8.19) и затем все полученные при этом остатки  $r$  в соответствии с теми же выражениями (8.16) и (8.19), мы получим характерную ФЦЗ (8.3), в которой после каждой 1 будет следовать не менее  $p$  нулей. Такое представление мы будем называть минимальной формой  $p$ -кода Фибоначчи. Доказано [9], что минимальная форма любого натурального числа  $N$  единственна.

Применительно к коду Фибоначчи (8.5) это означает, что минимальная форма – это такая ФЦЗ (8.3), в котором двух единиц рядом не встречается.

Возникает вопрос: сколько чисел может быть представлено в  $p$ -коде Фибоначчи (8.1) и коде Фибоначчи (8.5), если числа представлять только в

минимальной форме? Ответ на этот вопрос дают следующее утверждение, доказанное в [9].

**Теорема 8.4.** Каждое натуральное число  $N$  имеет единственное представление в минимальной форме  $p$ -кода Фибоначчи (8.1); при этом в минимальной форме  $n$ -разрядного  $p$ -кода Фибоначчи можно представить  $F_p(n+1)$  целых чисел в диапазоне от 0 до  $F_p(n+1)-1$ , где  $F_p(n+1)$  –  $(n+1)$ -е  $p$ -число Фибоначчи.

Частным случаем теоремы 8.4 является следующее утверждение, справедливое для кода Фибоначчи (8.5).

**Теорема 8.5.** Каждое натуральное число  $N$  имеет единственное представление в минимальной форме кода Фибоначчи (8.5); при этом в минимальной форме  $n$ -разрядного кода Фибоначчи (8.5) можно представить  $F_{n+1}$  целых чисел в диапазоне от 0 до  $F_{n+1}-1$ , где  $F_{n+1}$  –  $(n+1)$ -е число Фибоначчи.

## 8.5. Избыточность «фибоначчиевых» представлений

Ясно, что представления чисел в коде Фибоначчи (8.5) и  $p$ -коде Фибоначчи (8.1) являются «избыточными» в сравнении с классическим двоичным кодом (8.4). Теоремы 8.1 - 8.5 дают возможность количественно определить значение избыточности кода Фибоначчи (8.5) и  $p$ -кода Фибоначчи (8.1).

Что такое «избыточность» кода? Это понятие введено в теории корректирующих кодов [115]. Различают абсолютную и относительную избыточность. В качестве эталона для сравнения, как правило, используется классическая двоичная система (8.4), которая обладает «нулевой избыточностью». Обычно для обеспечения возможности обнаружения ошибок, к информационным разрядам, число которых равно  $m$ , добавляют определенное число корректирующих или избыточных разрядов  $k$ , значения которых формируются из исходных (информационных) разрядов по определенным правилам. Число  $k$  определяет «абсолютную избыточность». Если обозначить через  $n = m + k$

количество разрядов корректирующего кода, то можно вычислить «относительную избыточность» корректирующего кода согласно следующему выражению [115]:

$$R = \frac{k}{m} = \frac{n-m}{m} = \frac{n}{m} - 1. \quad (8.20)$$

При таком определении избыточности она характеризует относительное удлинение разрядной сетки «избыточного» кода по сравнению с «неизбыточным» кодом.

В качестве примера вычислим «относительную избыточность» кода Фибоначчи (8.5). Согласно Теореме 8.2 с помощью  $n$ -разрядного кода Фибоначчи (8.5) можно представить  $F_{n+2}$  целых чисел, если не накладывать ограничений на форму представления, а согласно Теореме 8.4 -  $F_{n+1}$  целых чисел, если использовать минимальную форму. Ясно, что для представления этого же диапазона чисел с помощью «неизбыточного» двоичного кода (8.4) нам потребуется либо  $m_1 \approx \log_2 F_{n+2}$  либо  $m_1 \approx \log_2 F_{n+1}$  двоичных разрядов. Подставляя эти выражения в формулу (8.20), мы получим следующие выражения для «относительной избыточности» кода Фибоначчи (8.5):

$$R_1 = \frac{n}{\log_2 F_{n+2}} - 1 \quad (8.21)$$

и

$$R_2 = \frac{n}{\log_2 F_{n+1}} - 1. \quad (8.22)$$

Для упрощения выражений (8.21), (8.22) обратимся к так называемой формуле Бине для чисел Фибоначчи, выведенной в главе 2:

$$F_n = \begin{cases} \frac{\Phi^n + \Phi^{-n}}{\sqrt{5}} & \text{для } n = 2k + 1; \\ \frac{\Phi^n - \Phi^{-n}}{\sqrt{5}} & \text{для } n = 2k \end{cases}. \quad (8.23)$$

При достаточно большом  $n$  выражение (8.23) можно заменить следующим приближенным выражением:

$$F_n \approx \frac{\Phi^n}{\sqrt{5}}, \quad (8.24)$$

где  $\Phi = \frac{1+\sqrt{5}}{2} \approx 1.618$  - золотая пропорция.

Используя (8.24) и выражая числа Фибоначчи  $F_{n+2}$  и  $F_{n+1}$  в формулах (8.21), (8.22) через золотую пропорцию  $\Phi$ , мы получим следующие выражения для относительной избыточности кода Фибоначчи (8.5):

$$R_1 = \frac{n}{\log_2 \frac{\Phi^{n+2}}{\sqrt{5}}} - 1 = \frac{n}{n \log_2 \Phi + 2 \log_2 \Phi - \frac{1}{2} \log_2 5} - 1 \quad (8.25)$$

$$R_2 = \frac{n}{\log_2 \frac{\Phi^{n+1}}{\sqrt{5}}} - 1 = \frac{n}{n \log_2 \Phi + \log_2 \Phi - \frac{1}{2} \log_2 5} - 1. \quad (8.26)$$

Легко показать, что при устремлении  $n$  к бесконечности выражения (8.25) и (8.26) стремятся к одному и тому же пределу:

$$R = \frac{1}{\log_2 \Phi} - 1 \approx 0.44. \quad (8.27)$$

Заметим, что дробь 0.44 (44%) и задает предельное значение величины относительной избыточности кода Фибоначчи (8.5).

**Основание кода Фибоначчи.** Что же является основанием кода Фибоначчи (8.5)? Как известно, в канонических позиционных системах счисления основание иногда определяется как отношение соседних весов разрядов [105]. Применяя такое определение, под основанием кода Фибоначчи (8.5) можно понимать предел отношения двух соседних весов разрядов, то есть, предел отношения  $F_n / F_{n-1}$  при  $n \rightarrow \infty$ . Но в главе 2 мы установили, что пределом этого отношения является золотая пропорция, откуда следует, что основанием кода Фибоначчи (8.5) является иррациональное число  $\Phi = \frac{1+\sqrt{5}}{2}$ . Точно так же можно доказать, что основанием  $p$ -кода Фибоначчи (8.1) является золотая  $p$ -пропорция  $\Phi_p$  - положительный корень алгебраического уравнения  $x^{p+1} - x^p - 1 = 0$ .

## 8.6. Удивительные аналогии между кодом Фибоначчи и генетическим кодом

Открытие генетического кода считается одним из наиболее важных научных открытий современной науки [44]. Как известно, наследственная информация любых живых организмов кодируется с помощью текстов, составленных из трехбуквенных слов – триплетов. Алфавит генетического кода состоит из четырех нитрогенных базисов: *A* (аденин), *C* (цитозин), *G* (гуанин), *T* (тиамин). Используя трехбуквенные триплеты, мы можем закодировать 21 объект, включая 20 аминокислот и один дополнительный объект, называемый стоп-кодоном (знак пунктуации). Ясно, что число различных трехбуквенных комбинаций из четырех нитрогенных базисов равно  $4^3 = 64$ . Эти комбинации используются для кодирования 21-го объекта. В этой ситуации некоторые из 21 объектов кодируются несколькими триплетами. Это называется вырожденностью генетического кода. Нахождение соответствия между триплетами и аминокислотами (включая стоп-кодон) интерпретируется как расшифровка генетического кода.

Теперь рассмотрим 6-разрядный код Фибоначчи (8.5), использующий 6 первых чисел Фибоначчи 1,1,2,3,5,8 в качестве весов разрядов:

$$N = a_6 \times 8 + a_5 \times 5 + a_4 \times 3 + a_3 \times 2 + a_2 \times 1 + a_1 \times 1. \quad (8.28)$$

Анализ позиционного представления (8.28) позволяет установить следующие неожиданные аналогии между 6-разрядным кодом Фибоначчи (8.28) и генетическим кодом:

**Первая аналогия.** Число 6-разрядных двоичных комбинаций равно  $2^6 = 64$ . Это означает, что для представления чисел 6-разрядный код Фибоначчи (8.28) использует  $2^6 = 64$  двоичных комбинаций от 000000 до 111111, что совпадает с количеством триплетов генетического кода ( $4^3 = 64$ ).

**Вторая аналогия.** Используя 6-разрядный код Фибоначчи (8.5), мы можем представить 21 число от минимального числа 0=000000 до максимального числа 20=111111. Заметим, что с использованием триплетного кодирования мы можем представить 21 объект, включая 20 аминокислот и стоп-кодон, который индицирует завершение протеинового синтеза.



**Третья аналогия.** Основным свойством кода Фибоначчи (8.5) является множественность представления чисел. За исключением минимального числа 0 и максимального числа 20, которые имеют единственные представления - 000000 и 111111, соответственно, все остальные числа от 1 до 19 имеют множественное представление в коде Фибоначчи (8.5). Необходимо отметить, что генетический код обладает подобным же свойством, которое называется вырожденностью генетического кода.

Таким образом, между кодом Фибоначчи (8.5) и генетическим кодом существуют очень интересные аналогии, которые дают основание выделить код Фибоначчи (8.5) в качестве особого кода для избыточного кодирования информации. Избыточность кода Фибоначчи (8.5), в пределе равная 44%, совпадает с избыточностью генетического кода. Это позволяет выдвинуть гипотезу, что, возможно, избыточность в 44% и является «оптимальной» избыточностью для любых систем, включая информационные системы.

## 8.7. Счет по Фибоначчи

Как известно, счётчики являются важными функциональными устройствами компьютеров и микропроцессоров. Они используются для построения таймеров или для выборки инструкций из ПЗУ в микропроцессорах. Они могут использоваться как делители частоты в управляемых генераторах частоты (синтезаторах). Во многих случаях применения счетчиков к ним предъявляется требование высокого быстродействия. Как известно, быстродействие счетчика характеризуется временем установления в нем нового состояния. «Критической» ситуацией для счетчиков, построенных на основе классической двоичной системы счисления (8.4), является установление нового состояния счетчика в следующей ситуации:

$$011111111+1=100000000.$$

В этом случае осуществляется перенос единицы из младшего в старший разряд счетчика. Если счетчик строится на так называемых «счетных триггерах», то такой счетчик обладает очень низким быстродействием. Для ускорения переноса

используются схемы группового переноса, что приводит к усложнению схемы счетчика.

Покажем теперь, что фибоначиевые счетчики, основанные на микрооперациях свертки (суммирующий счетчик) и развертки (вычитающий счетчик) обладают определенными преимуществами по сравнению с двоичными счетчиками.

**Суммирующий счетчик Фибоначчи.** Счет единиц в коде Фибоначчи (суммирующий счетчик) осуществляется следующим образом. Перед добавлением единицы в младший разряд исходная ФЦЗ, соответствующая числу  $N$ , приводится в такую форму, чтобы значение младшего разряда было равным 0. Затем в младший разряд такой ФЦЗ добавляется 1, что приводит к тому, что числовое содержание счетчика становится равным  $N + 1$ . После этого с помощью свертки ФЦЗ числа  $N + 1$  приводится в такую форму, чтобы значение младшего разряда стало равным 0.

Продемонстрируем «счет по Фибоначчи» на следующем примере:

$$\begin{aligned}
 000000 + 1 &= 0000\mathbf{01} = 000010 = 1 \\
 000010 + 1 &= 000\mathbf{011} = 000100 = 2 \\
 000100 + 1 &= 0001\mathbf{01} = 000110 = 3 \\
 00\mathbf{011}0 + 1 &= 0010\mathbf{01} = 001010 = 4 \\
 001010 + 1 &= 001\mathbf{011} = 001100 = 5 \\
 0\mathbf{011}00 + 1 &= 0100\mathbf{01} = 010010 = 6 \\
 010010 + 1 &= 010\mathbf{011} = 010100 = 7 \\
 010100 + 1 &= 0101\mathbf{01} = 010110 = 8 \\
 01\mathbf{011}0 + 1 &= \mathbf{011}0\mathbf{01} = 100010 = 9 \\
 100010 + 1 &= 100\mathbf{011} = 100100 = 10 \\
 100100 + 1 &= 1001\mathbf{01} = 100110 = 11 \\
 10\mathbf{011}0 + 1 &= 1010\mathbf{01} = 101010 = 12 \\
 101010 + 1 &= 101\mathbf{011} = 1\mathbf{011}00 = \mathbf{110}000 = 000000
 \end{aligned} \tag{8.29}$$

Здесь жирным шрифтом в скобках выделены те ситуации, когда в ФЦЗ осуществляются свертки. Рассмотрим, например, ситуацию перехода от ФЦЗ числа  $8 = 010110$  к ФЦЗ числа  $9$ . В этом случае одновременно с записью 1 в младший

(первый) разряд ФЦЗ осуществляется свертка единиц 2-го и 3-го разрядов в 4-й разряд ( $011 \rightarrow 100$ ), что приводит к получению ФЦЗ числа  $9 = 011001$ . На следующем этапе в этом же ФЦЗ осуществляются две свертки одновременно, что приводит к появлению новой ФЦЗ:  $\boxed{011}00\boxed{01} \rightarrow 1000010 = 9$ . Таким образом, особенность «фибоначчиевого» счета состоит в том, что в любой ситуации переход от ФЦЗ числа  $N$  к ФЦЗ числа  $N+1$  осуществляется за время последовательного выполнения не более двух сверток. Заметим, что нижний ряд таблицы (8.29) соответствует переполнению счетчика Фибоначчи.

**Вычитающий счетчик Фибоначчи.** Вычитание единиц в коде Фибоначчи (8.5) (вычитающий счетчик) осуществляется путем вычитания единицы из младшего разряда ФЦЗ числа  $N$ , в котором значение младшего разряда равно 1, с последующей разверткой в младших разрядах. Рассмотрим пример функционирования вычитающего счетчика Фибоначчи:

$$\begin{aligned}
 1111-1 &= 11\boxed{10} = 1101 = 6 \\
 1101-1 &= 1\boxed{100} = 1011 = 5 \\
 1011-1 &= 10\boxed{10} = 1001 = 4 \\
 \boxed{100}1-1 &= 0110 = 0101 = 3 \\
 0101-1 &= 0\boxed{100} = 0011 = 2 \\
 0011-1 &= 00\boxed{10} = 0001 = 1 \\
 0001-1 &= 0000 = 0
 \end{aligned} \tag{8.30}$$

Здесь жирным шрифтом в скобках выделены те ситуации, когда в счетчике Фибоначчи осуществляются развертки. Таким образом, особенность вычитающего счетчика Фибоначчи состоит в том, что в любой ситуации переход от ФЦЗ числа  $N$  к ФЦЗ числа  $N-1$  осуществляется за время последовательного выполнения не более двух разверток.

Рассмотренные выше алгоритмы функционирования суммирующего и вычитающего счетчиков Фибоначчи показывают, что в таких счетчиках заложены предпосылки для конструирования быстрых счетчиков (без использования сложных схем группового переноса). То есть, уже этот простейший пример

показывает определенные преимущества кода Фибоначчи (8.5) перед классическим двоичным кодом (8.4).

**Первая диссертация по счетчикам Фибоначчи.** В 1971 г. автор был избран на должность зав. кафедрой информационно-измерительной техники Таганрогского радиотехнического университета. Именно там возникла идея создания новой компьютерной арифметики – арифметики Фибоначчи. Этот научный результат впервые был изложен автором в статье «Избыточные позиционные системы счисления», опубликованной в 1974 г. в одном из научных сборников Таганрогского радиотехнического института [59]. Там же, в Таганроге, были написаны статьи на эту тему, опубликованные в известных компьютерных журналах и сборниках [60-62]. В дальнейшем к этой научной тематике были привлечены талантливые студенты, а затем аспиранты института Юрий Вишняков, Владимир Лужецкий, Алексей Азаров, Николай Соляниченко и другие, которые позже защитили кандидатские и даже докторские диссертации (Вишняков, Азаров, Лужецкий). Однако первой из них была кандидатская диссертация Юрия Вишнякова «Разработка принципов построения и исследование пересчетных устройств в  $p$ -кодах Фибоначчи». Диссертация была защищена 29.сентября 1977 г. на специализированном совете Таганрогского радиотехнического института.

Как неисповедимы судьбы научных направлений? Значительно позже автор узнал, что одновременно с этими исследованиями подобные исследования по «компьютерам Фибоначчи» проводились в США (Университет Мэриленд) под руководством доктора Ньюкомба (Newcomb) [120-125]. И в этом университете в 1979 г. была защищена докторская диссертация V.D. Hong “A Class of Arithmetic Burst-Error-Correcting Codes for the Fibonacci Computer” [125]. Уже из названия диссертации следует, что цели исследований, проводившихся в Университете Мэриленд под руководством доктора Ньюкомба, и исследований автора в этой области, начавшиеся в Таганрогском радиотехническом институте, были одинаковы – создание самоконтролирующихся и самокорректирующихся вычислительных и измерительных систем, основанных на использовании кодов Фибоначчи. Но автору приятно отметить, что диссертация Юрия Вишнякова,

защищенная в 1977 г., была все же первой диссертацией по «компьютерам Фибоначчи» в мировой компьютерной науке.

## 8.8. Элементы арифметики Фибоначчи

**Сравнение чисел в коде Фибоначчи.** Простота сравнения чисел и вытекающая отсюда «наглядность» цифровой записи чисел является одним из важнейших преимуществ канонических позиционных систем счисления, включая двоичную систему (8.4). Известно, что сравнение чисел, представленных в двоичном коде (8.3), осуществляется поразрядно, начиная со старшего разряда кода (8.3). Доказано [9], что  $p$ -коды Фибоначчи (8.1) обладают таким же свойством. Единственное отличие состоит в том, что перед сравнением ФЦЗ (8.3) они приводятся к минимальной форме, после чего минимальные формы сравниваются поразрядно, начиная со старшего разряда, до момента появления первой пары несовпадающих разрядов.

Например, для сравнения двух чисел  $A = 00111101101$  и  $B = 00111110110$ , представленных в коде Фибоначчи (8.5), необходимо выполнить следующее:

1. Приведение сравниваемых ФЦЗ к минимальной форме:

$$A = 0\boxed{011}11\boxed{011}\boxed{01} = 010\boxed{011}1010 = 0101001010;$$

$$B = 0\boxed{011}111\boxed{011}0 = 010\boxed{011}11000 = 01010\boxed{011}000 = 01010100000.$$

Здесь все кодовые комбинации, подлежащие свертке на каждом этапе приведения к минимальной форме, выделены.

2. Поразрядное сравнение минимальных форм чисел  $A$  и  $B$ , начиная со старшего разряда, слева направо до появления первой пары несовпадающих разрядов:

$$A = 01010[\mathbf{0}]10010$$

$$B = 01010[\mathbf{1}]00000.$$

Мы видим, что первая пара несовпадающих разрядов содержит двоичную цифру 0 в минимальной форме первого числа  $A$  и двоичную цифру 1 в минимальной форме второго числа  $B$ . Это означает, что  $B > A$ .

**Преобразование классического двоичного кода в код Фибоначчи и наоборот.** Примеры такого преобразования представлены в Табл.8.3.

Таблица 8.3

<u>ДК → КФ</u>							<u>КФ → ДК</u>										
ДК	$2^i$		8	5	3	2	1	1	КФ	$F_i$	$2^3$	$2^2$	$2^1$	$2^0$			
1	$2^3$	→	8	=	1	0	0	0	1	8	→	8	=	1	0	0	0
0	$2^2$	→	0	=	0	0	0	0	0	5	→	0	=	0	0	0	0
1	$2^1$	→	2	=	0	0	0	1	1	3	→	3	=	0	0	1	1
1	$2^0$	→	1	=	0	0	0	0	0	2	→	0	=	0	0	0	0
			11	=	1	0	0	1	0	1	→	0	=	0	0	0	0
			МФ	=	1	0	1	0	0	1	→	0	=	0	0	0	0
												11	=	1	0	1	1

Слева представлено преобразование двоичного кода в код Фибоначчи (ДК→КФ), справа – обратное преобразование (КФ→ДК). Суть такого преобразования очень проста. В таблице слева веса разрядов двоичного кода (ДК) представляются в коде Фибоначчи (КФ), после чего все КФ, соответствующие единичным весам разрядов исходного ДК, суммируются и сумма приводится к минимальной форме (МФ). В таблице справа эта же идея используется для преобразования кода Фибоначчи в двоичный код: КФ→ДК.

**Базовые микрооперации.** Как известно, в классической двоичной арифметике основной арифметической операцией является сложение. Вычитание сводится к сложению путем использования понятий инверсного и дополнительного кодов. Умножение и деление выполняются с использованием операций сложения, вычитания и сравнения. В основе двоичного сложения лежит тривиальное тождество, связывающее двоичные числа:

$$2^m + 2^m = 2^{m+1}, \tag{8.31}$$

откуда вытекает следующая широко известная таблица сложения двоичных чисел:

0	+	0	=	0
0	+	1	=	1
1	+	1	=	1
1	+	1	=	10

(8.32)

На первом этапе именно такой подход был применен для создания правил фибоначчиевого сложения [9,59]. Представляя сумму двух чисел Фибоначчи в виде

$$F_n + F_n = F_n + F_{n-1} + F_{n-2} = F_{n+1} + F_{n-2}, \quad (8.33)$$

мы можем сконструировать следующую таблицу для фибоначчиевого сложения, подобную таблице (8.32):

$$\begin{array}{|l} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 1 = 1 \\ 1 + 1 = 1001 \end{array} . \quad (8.34)$$

Из таблицы (8.34) вытекает следующая особенность фибоначчиевого сложения. При сложении двух единиц (1+1) в  $k$ -х разрядах слагаемых возникает два переноса из  $k$ -го разряда в  $(k+1)$ -й и  $(k-2)$ -й разряды. При этом вычитание сводится к сложению путем введения понятий инверсного и дополнительного кодов. Умножение и деление сводится к сложению, вычитанию и сравнению. Подобная арифметика разработана в статье [59] и книге [9].

Однако, позже была разработана новая арифметика Фибоначчи, которая оказалась более перспективной с точки зрения создания самоконтролирующихся компьютерных структур. Эта арифметика основана использовании на четырех базовых микрооперациях [18]:

Свертка :  $\boxed{011 \uparrow 100}$  Развертка :  $\boxed{100 \rightarrow 011}$

Перемещение :  $\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \rightarrow & = \\ \hline 0 & 1 \\ \hline \end{array}$  Поглощение :  $\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \updownarrow & = \\ \hline 1 & 0 \\ \hline \end{array}$

Операции «свертка» и «развертка» рассмотрены выше: они основаны на основном рекуррентном соотношении (8.6), связывающем веса «фибоначчиевых» разрядов. Эти операции являются одноместными, то есть, выполняются в рамках ФЦЗ, хранящейся в одном регистре. Напомним, что выполнение этих операций не приводит к изменению значения числа, представляемого кодом Фибоначчи.

Микрооперации «перемещение» и «поглощение» являются двухместными, то есть, выполняются в рамках ФЦЗ, расположенных в двух регистрах. Например, микрооперация перемещение выполняется над одноименными ( $k$ -ми) разрядами двух ФЦЗ при условии, что значение этого разряда в верхнем регистре равно 1, а в нижнем регистре равно 0. Операция состоит в «перемещении» ( $\downarrow$ ) 1 из верхнего регистра в нижний. Микрооперация «поглощение» также является двухместной операцией. Она состоит во взаимном «поглощении» ( $\updownarrow$ ) единиц, находящихся в  $k$ -х разрядах верхнего и нижнего регистров.

## 8.9. Выполнение логических операций с использованием базовых микроопераций

Мы можем продемонстрировать возможность выполнения логических операций над кодовыми комбинациями с использованием базовых микроопераций. Выполним, например, все возможные перемещения из верхнего регистра  $A$  в нижний регистр  $B$ :

$$\begin{array}{cccccccc}
 A & = & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
 & & \downarrow & & & \downarrow & & & \\
 B & = & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
 \hline
 A' & = & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 B' & = & 1 & 1 & 0 & 1 & 1 & 0 & 1
 \end{array}$$

Мы получили две новые кодовые комбинации  $A'$  и  $B'$  как результат выполнения перемещений. Важно подчеркнуть, что двоичная комбинация  $A'$  является логической конъюнкцией ( $\wedge$ ) исходных двоичных комбинаций  $A$  и  $B$ , то есть,

$$A' = A \wedge B.$$

В то же время двоичная комбинация  $B'$  является логической дизъюнкцией ( $\vee$ ) исходных двоичных комбинаций  $A$  и  $B$ , то есть,

$$B' = A \vee B.$$

Логическая операция «сложение по модулю 2» выполняется путем одновременного выполнения всех возможных операций перемещения и поглощения. Например,



$$\begin{array}{r}
 A = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\
 \quad \uparrow \quad \downarrow \quad \quad \downarrow \downarrow \quad \uparrow \\
 B = 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1 \\
 \hline
 A' = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = \text{const } 0 \\
 B' = 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0 = A \oplus B
 \end{array}$$

Мы видим, что две новые кодовые комбинации  $A' = \text{const } 0$  и  $B' = A \oplus B$  являются результатом этого преобразования.

Логическая операция «инверсия кода  $A$ » сводится к выполнению операции поглощения над исходной кодовой комбинацией  $A$  и «единичной» кодовой комбинацией  $B = \text{const } 1$ :

$$\begin{array}{r}
 A = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\
 \quad \downarrow \quad \downarrow \quad \quad \downarrow \downarrow \quad \downarrow \\
 B = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
 \hline
 A' = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = \text{const } 0 \\
 B' = 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0 = \bar{A}
 \end{array}$$

Мы видим, что две новые кодовые комбинации  $A' = \text{const } 0$  и  $B' = \bar{A}$  являются результатом этого преобразования.

## 8.10. Фибоначчиево сложение и вычитание

**Фибоначчиево сложение.** В качестве примера рассмотрим сложение чисел  $A_0 = 010100100$  и  $B_0 = 001010100$ , которые размещаются в верхнем регистре  $A$  и нижнем регистре  $B$ , соответственно.

При фибоначчиевом сложении мы используем микрооперации «перемещение», «свертка» и «развертка». Сумма формируется в нижнем регистре  $B$ .

Первый шаг фибоначчиевого сложения состоит в выполнении всех возможных перемещений двоичных 1 из регистра  $A$  в регистр  $B$ :

$$\begin{array}{r}
 A_0 = 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0 \\
 \quad \quad \downarrow \quad \downarrow \\
 B_0 = 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0 \\
 \hline
 A_1 = 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0 \\
 B_1 = 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0
 \end{array}$$

В результате в регистрах  $A$  и  $B$  формируются две новые ФЦЗ  $A_1$  и  $B_1$ .

Второй шаг состоит в выполнении всех возможных разверток в ФЦЗ числа  $A_1$  и всех возможных сверток в ФЦЗ числа  $B_1$ , то есть,

$$\begin{aligned} A_1 &= 000000\boxed{100} \rightarrow A_2 = 000000011 \\ B_1 &= \boxed{011}110100 \rightarrow B_2 = 100110100 \end{aligned}$$

Третий шаг состоит в выполнении всех возможных перемещений двоичных 1 из регистра  $A$  в регистр  $B$ :

$$\begin{array}{r} A_2 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \mathbf{1 \ 1} \\ \phantom{A_2} \phantom{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0} \phantom{0 \ 0} \phantom{\downarrow \downarrow} \\ B_2 = 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ \mathbf{0 \ 0} \\ \hline A_3 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\ B_3 = 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \end{array}$$

Сложение закончено, поскольку все двоичные 1 перемещены из регистра  $A$  в регистр  $B$ . После приведения ФЦЗ  $B_3$  к минимальной форме, мы получим сумму  $B_3 = A + B$ , представленную в минимальной форме:

$$B_3 = 10\boxed{011}\boxed{011}1 = 1010010\boxed{01} = 101001010.$$

**Фибоначчиево вычитание.** Идея фибоначчиевого вычитания числа  $B$  из числа  $A$ , основанного на базовых микрооперациях, состоит во взаимном поглощении всех двоичных единиц в ФЦЗ, хранящихся в регистрах  $A$  и  $B$ , до тех пор, пока одна из них не станет равной 0. Для этого используются микрооперации «поглощение» и «развертка». Результат вычитания всегда формируется в регистре, который содержит большее число. Если результат вычитания формируется в верхнем регистре  $A$ , это означает, что результат вычитания имеет знак «+»; в противном случае результат вычитания отрицательный.

Рассмотрим следующий пример. Пусть требуется произвести фибоначчиевое вычитание числа  $B_0 = 101010010$  из числа  $A_0 = 101001000$ .

Первый шаг состоит в поглощении всех возможных 1 в числах  $A_0$  и  $B_0$ :

$$\begin{array}{r}
A_0 = 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \\
\quad \quad \uparrow \quad \uparrow \\
B_0 = 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0. \\
\hline
A_1 = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \\
B_1 = 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0
\end{array}$$

Второй шаг состоит в выполнении микроопераций «развертка» в ФЦЗ чисел  $A_1$  и  $B_1$ :

$$\begin{array}{l}
A_1 = 00000\boxed{100}0 \rightarrow A_2 = 000000110 \\
B_1 = 0000\boxed{100}\boxed{10} = B_2 = 000001101
\end{array}$$

Третий шаг состоит в выполнении микроопераций «поглощение» над ФЦЗ чисел  $A_2$  и  $B_2$ :

$$\begin{array}{r}
A_2 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \\
\quad \quad \quad \quad \quad \quad \quad \quad \uparrow \\
B_2 = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1. \\
\hline
A_3 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \\
B_3 = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1
\end{array}$$

Четвертый шаг состоит в развертке ФЦЗ чисел  $A_3$  и  $B_3$ :

$$\begin{array}{l}
A_3 = 0000000\boxed{10} \rightarrow A_4 = 000000001 \\
B_3 = 00000\boxed{100}\boxed{1} \rightarrow A_4 = 000000111
\end{array}$$

Пятый шаг состоит в выполнении поглощений над ФЦЗ чисел  $A_4$  и  $B_4$ :

$$\begin{array}{r}
A_4 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \\
\quad \quad \quad \quad \quad \quad \quad \quad \uparrow \\
B_4 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1. \\
\hline
A_5 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
B_5 = 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0
\end{array}$$

Вычитание закончено. После приведения ФЦЗ числа  $B_5$  к минимальной форме получаем:

$$B_5 = 00000\boxed{011}0 = 000001000.$$

Результат вычитания находится в регистре  $B$ . Это означает, что результат вычитания отрицательный, то есть разность чисел  $A - B$  равна:

$$R = A - B = -000001000.$$

## 8.11. Фибоначчиево умножение и деление

**Фибоначчиево умножение.** Предлагаемый ниже фибоначчиевый метод умножения восходит к египетскому методу удвоению [9], который лежал в основе их десятичной арифметики. Суть фибоначчиевого умножения, подобного египетскому методу умножения, состоит в следующем. Рассмотрим произведение  $P = A \times B$ , где числа  $A$  и  $B$  представлены в  $p$ -коде Фибоначчи (8.1). Используя представление множителя  $B$  в  $p$ -коде Фибоначчи (8.1), мы можем представить произведение  $P = A \times B$  в следующем виде:

$$P = A \times b_n F_p(n) + A \times b_{n-1} F_p(n-1) + \dots + A \times b_i F_p(i) + \dots + A \times b_1 F_p(1), \quad (8.35)$$

где  $F_p(i)$  – число Фибоначчи,  $b_i (i = 1, 2, 3, \dots, n)$  – двоичная цифра ФЦЗ числа  $B$ .

Алгоритм фибоначчиевого умножения непосредственно вытекает из выражения (8.35). Умножение сводится к сложению частных произведений вида  $A \times b_i F_p(i)$ . Они формируются из множителя  $A$  в соответствии со специальной процедурой, напоминающей рассмотренный выше египетский метод удвоения.

Продemonстрируем фибоначчиево умножение на конкретном примере для случая кода Фибоначчи (8.5). Предположим, что необходимо умножить числа  $A = 41$  и  $B = 305$ . Для этого сделаем следующее:

1. Сконструируем таблицу из трех столбцов, обозначенных буквами  $F, G, P$ .
2. Сформируем последовательность Фибоначчи 1, 1, 2, 3, 5, 8, 13, 21, 34 в столбце  $F$ .
3. Сформируем в столбце  $G$  последовательность чисел  $G(n)$ , вычисляемых в соответствии с рекуррентным соотношением  $G(n) = G(n-1) + G(n-2)$  при начальных условиях  $G(1) = G(2) = 305$  (число  $B = 305$ ).
4. Отметим жирным шрифтом и наклонной линией (/) все числа столбца  $F$ , которые в сумме образуют второй множитель  $41(41 = 34 + 5 + 2)$ .
5. Отметим жирным шрифтом все числа столбца  $G$ , соответствующие отмеченным числам столбца  $F$  и перепишем их в столбец  $P$ .
6. Суммируя все числа столбца  $P$ , получим произведение:  $41 \times 305 = 12505$ .

<i>F</i>	<i>G</i>	<i>P</i>
1	305	
1	305	
<b>/2</b>	<b>610</b>	<b>→ 610</b>
3	915	
<b>/5</b>	<b>1525</b>	<b>→ 1525</b>
8	2440	
13	3965	
21	6505	
<b>/34</b>	<b>10370</b>	<b>→ 10370</b>
<b>41 = 34 + 5 + 2</b>	<b>41 × 305</b>	<b>= 12505</b>

**Фибоначчиево деление.** Фибоначчиево деление подобно египетскому методу деления, основанному на методе удвоения [9]. Рассмотрим этот метод на конкретном примере. Пусть требуется разделить число  $A = 481$  (делимое) на число  $B = 13$  (делитель) в коде Фибоначчи (11.5). Деление осуществляется в несколько этапов.

Первый этап:

1. Построим таблицу, состоящую из трех столбцов  $F, G, D$ .
2. Сформируем последовательность Фибоначчи 1, 1, 2, 3, 5, 8, 13, 21, 34 в столбце  $F$ .
3. Сформируем в столбце  $G$  последовательность чисел  $G(n)$ , вычисляемых в соответствии с рекуррентным соотношением  $G(n) = G(n-1) + G(n-2)$  при начальных условиях  $G(1) = G(2) = 13$  (делитель  $B = 13$ ).
4. На каждом шаге формирования чисел  $G(n)$  сравниваем их с делимым  $A = 481$ , вписанным в столбец  $D$ , и фиксируем результаты сравнения ( $\leq$  или  $>$ ) до получения первого результата сравнения «больше»:  $615 > 481$ .
5. Отмечаем жирным шрифтом и наклонной линией (/) число Фибоначчи 34 столбца  $F$ , которое соответствует предыдущему числу 442 столбца  $G$ .
6. Вычислим разность  $R_1 = 481 - 442 = 39$ :

<i>F</i>	<i>G</i>	<i>D</i>
1	13	≤ 481
1	13	≤ 481
2	26	≤ 481
3	39	≤ 481
5	65	≤ 481
8	104	≤ 481
13	169	≤ 481
21	273	≤ 481
<b>/34</b>	<b>442</b>	≤ 481
55	615	> 481

→  $R_1 = 481 - 442 = 39$ .

Второй этап:

Второй этап фибоначчиевского деления состоит в повторении первого этапа, но при этом вместо делимого  $A = 481$  используется разность  $R_1 = 39$ :

<i>F</i>	<i>G</i>	<i>D</i>
1	13	≤ 39
1	13	≤ 39
2	26	≤ 39
<b>/3</b>	<b>39</b>	≤ 39
5	65	> 39

→  $R_2 = 39 - 39 = 0$ .

Поскольку разность  $R_2 = 39 - 39 = 0$ , это означает, что фибоначчиевое деление закончено. Результат деления равен сумме всех отмеченных чисел Фибоначчи в столбце *F* первой и второй таблицы, то есть,  $34 + 3 = 37$ .

## Глава 9

# КОДЫ ЗОЛОТОЙ ПРОПОРЦИИ И «ЗОЛОТАЯ» ТЕОРИЯ ЧИСЕЛ

### 9.1. Система счисления Бергмана

Наиболее революционным предложением в современной теории систем счисления по праву можно считать систему счисления, предложенную в 1957 г. американским математиком Джорджем Бергманом [55]. Математическое выражение для системы счисления Бергмана имеет вид:

$$A = \sum_i a_i \Phi^i, \quad (9.1)$$

где  $A$  – некоторое действительное число и  $a_i \in \{0,1\}$  – двоичная цифра  $i$ -го разряда ( $i = 0, \pm 1, \pm 2, \pm 3, \dots$ ),  $\Phi^i$  – вес  $i$ -го разряда в системе счисления (9.1),  $\Phi = \frac{1+\sqrt{5}}{2}$  – основание системы счисления (9.1).

На первый взгляд, может показаться, что не существует никакой особенности в выражении (9.1) по сравнению с известными позиционными системами счисления, в частности, с двоичной системой

$$A = \sum_i a_i 2^i, \quad (9.2)$$

но это только на первый взгляд. Главная особенность состоит в том, что Бергман использовал «золотую пропорцию»  $\Phi = \frac{1+\sqrt{5}}{2}$  в качестве основания своей системы счисления (9.1).

Рассмотрим систему Бергмана (9.1) с теоретико-числовой точки зрения. Именно ее основание  $\Phi = \frac{1+\sqrt{5}}{2}$  определяет все необычные свойства системы (9.1) с точки зрения представления чисел.

В главе 1 мы установили, что «золотая пропорция» обладает следующим замечательным свойством:

$$\Phi^i = \Phi^{i-1} + \Phi^{i-2} = \Phi \times \Phi^{i-1}, \quad (9.3)$$

где  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ .

Рассмотрим представления чисел в системе Бергмана (9.1). Ясно, что сокращенная цифровая запись числа  $A$  в системе (9.1) имеет следующий вид:

$$A = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}. \quad (9.4)$$

Сокращенную цифровую запись числа  $A$  в системе (9.1) в дальнейшем будем называть золотой цифровой записью (ЗЦЗ) числа  $A$ .

Из (9.4) вытекает, что ЗЦЗ числа  $A$  по своей форме полностью совпадает с представлением числа в двоичной системе счисления (9.2). Она представляет собой двоичную кодовую комбинацию, разделенную запятой на две части, левую часть  $a_n a_{n-1} \dots a_1 a_0$ , соответствующую «весам»:  $\Phi^n, \Phi^{n-1}, \dots, \Phi^1, \Phi^0 = 1$ , и правую часть  $a_{-1} a_{-2} \dots a_{-m}$ , соответствующую «весам»:  $\Phi^{-1}, \Phi^{-2}, \dots, \Phi^{-m}$ . Напомним, что «веса»  $\Phi^i$  ( $i = 0, \pm 1, \pm 2, \pm 3, \dots$ ) связаны между собой математическими соотношениями (9.3).

В качестве примера рассмотрим двоичную кодовую комбинацию 100101. Ясно, что в двоичной системе (9.2) она представляет собой следующее натуральное число:

$$A = 100101 = 2^5 + 2^2 + 2^0 = 37.$$

В то же время в системе Бергмана (9.1) двоичная комбинация 100101 представляет следующее действительное число:

$$A = 100101 = \Phi^5 + \Phi^2 + \Phi^0. \quad (9.5)$$

В теории чисел Фибоначчи [4] доказано следующее тождество, которое связывает  $i$ -ю степень «золотой пропорции»  $\Phi^i$  с числами Фибоначчи и Люка:

$$\Phi^i = \frac{L_i + F_i \sqrt{5}}{2} \quad (i = 0, \pm 1, \pm 2, \pm 3, \dots). \quad (9.6)$$

Используя формулу (9.6), мы можем установить, что число  $A$ , задаваемое выражением (9.5), равно:



$$A = 100101 = \frac{11+5\sqrt{5}}{2} + \frac{3+\sqrt{5}}{2} + 1 = \frac{16+6\sqrt{5}}{2} = 8+3\sqrt{5}. \quad (9.7)$$

Существенно подчеркнуть, что число (9.7) является иррациональным числом. Это означает, что мы представили иррациональное число (9.7) в системе Бергмана (9.1), в виде кодовой комбинации 100101, состоящей из конечного числа битов!

В частности, основание системы (9.1) («золотая пропорция») представляется в системе Бергмана (9.1) традиционным образом, то есть:

$$\Phi = \frac{1+\sqrt{5}}{2} = 10.$$

Но из нашего предыдущего опыта мы знаем, что в канонических позиционных системах счисления [105] невозможно представить иррациональное число в виде цифровой записи, использующей конечное числа цифр. Именно поэтому возможность представления некоторых иррациональных чисел (степеней «золотой пропорции» и их сумм) с использованием конечной совокупности двоичных цифр есть первый неожиданный результат системы Бергмана (9.1), который вступает в противоречие с нашими традиционными представлениями о системах счисления.

## 9.2. Особенности представления чисел в системе Бергмана

**«Свертка» и «развертка» двоичных разрядов.** Выше при изучении кода Фибоначчи (8.5) мы ввели две важные операции над кодовыми словами, называемые «сверткой» и «разверткой». Оказывается, что эти же операции применимы к кодовым словам, представляющим числа в системе (9.1). Рассмотрим еще раз тождество (9.3), связывающее веса разрядов в системе (9.1). Заметим, что тождество  $\Phi^i = \Phi^{i-1} + \Phi^{i-2}$  по форме аналогично рекуррентному соотношению Фибоначчи (8.6). Переходя на уровень ЗЦЗ (9.4), представляющих числа в системе Бергмана (9.1), мы можем дать следующую кодовую интерпретацию тождества  $\Phi^i = \Phi^{i-1} + \Phi^{i-2}$ :

$$100 \square 011. \quad (9.8)$$

Ясно, что подобно коду Фибоначчи (8.5) замена тройки соседних разрядов 100 в ЗЦЗ (9.4) на тройку 011 (100→011) соответствует выполнению в кодовом слове (9.4) микрооперации «развертка», описанной выше, в то время как обратное преобразование (011→100) соответствует микрооперации «свертка». Ясно, что на основании (9.3) выполнение в ЗЦЗ любого числа микроопераций свертки и развертки не изменяет числа, представляемого этой ЗЦЗ.

Рассмотрим еще раз ЗЦЗ 100101, представляющую число (9.7) в системе Бергмана (9.1). Выполняя в ЗЦЗ 100101 микрооперацию развертки над старшим разрядом, мы получим новую ЗЦЗ 011101, представляющую то же самое число (9.7) в системе Бергмана:

$$A = 011101 = \Phi^4 + \Phi^3 + \Phi^2 + \Phi^0. \quad (9.9)$$

Мы можем также применить микрооперацию «развертка» к младшему разряду ЗЦЗ (9.9). При этом мы получим новую ЗЦЗ этого же числа, в которой появились двоичные цифры в дробной части числа:

$$A = 01110 \boxed{1.00} = 011100.11 = \Phi^4 + \Phi^3 + \Phi^2 + \Phi^{-1} + \Phi^{-2}.$$

Продолжая процесс развертки младшего разряда новой ЗЦЗ этого же числа, мы можем получить теоретически бесконечное количество ЗЦЗ одного и того же числа в системе Бергмана (9.1).

Таким образом, в системе Бергмана (9.1) числа имеют многозначное представление, то есть, каждому числу  $A$  соответствует некоторое множество ЗЦЗ типа (9.4). Это – второй неожиданный результат, вытекающий из рассмотрения системы Бергмана (9.1).

**Формы представления чисел в системе Бергмана.** Используя введенные выше микрооперации «свертка» и «развертка», мы можем получить две крайние ЗЦЗ одного и того же числа в системе Бергмана (9.1). Рассмотрим, например, ЗЦЗ 0111111. Если мы выполним в ней все возможные «свертки», то получим при этом ЗЦЗ, называемую минимальной формой:

$$\boxed{011}1111 = 10\boxed{011}11 = 1010\boxed{011} = 1010100 \text{ (минимальная форма).}$$

Заметим, что в минимальной форме двух единиц рядом не встречается.

Рассмотрим теперь ЗЦЗ 100000, которая представляет иррациональное число  $\Phi^5 = \frac{11+5\sqrt{5}}{2}$ . Выполняя в ЗЦЗ 100000 все возможные «развертки», мы можем получить еще одну характерную ЗЦЗ, называемую максимальной формой:

$$\boxed{100}0000 = 01\boxed{100}00 = 0101\boxed{100} = 0101011 \text{ (максимальная форма).}$$

Заметим, что в максимальной форме двух нулей рядом не встречается.

### 9.3. Представление натуральных чисел

Покажем теперь, как можно получить все ЗЦЗ натуральных чисел, используя микрооперации «свертка» и «развертка». Начнем с числа 1. Оно может быть выражено через «золотую пропорцию» следующим способом:

$$1 = \Phi^0. \quad (9.10)$$

Используя систему Бергмана (9.1), мы можем представить число (9.10) следующим образом:

$$1 = \Phi^0 = 1.00. \quad (9.11)$$

Заметим, что в ЗЦЗ (9.11) точка отделяет 0-й разряд от разрядов с отрицательными индексами.

Затем, используя «развертку», мы можем представить число (9.11) следующим образом:

$$1 = \Phi^0 = 0.11 = \Phi^{-1} + \Phi^{-2}. \quad (9.12)$$

А теперь добавим бит 1 в 0-й разряд ЗЦЗ (9.12). В результате получим ЗЦЗ числа 2:

$$2 = 1.11. \quad (9.13)$$

Применяя микрооперацию «свертка» к старшим разрядам ЗЦЗ (9.13), мы получаем новую ЗЦЗ («минимальную форму») числа 2:

$$2 = 10.01 = \Phi^1 + \Phi^{-2}. \quad (9.14)$$

Добавляя бит 1 в 0-й разряд ЗЦЗ числа (9.14), мы получаем ЗЦЗ числа 3:

$$3 = 11.01. \quad (9.15)$$

Применяя микрооперацию «свертка» к старшим разрядам ЗЦЗ (9.15), мы получаем новую ЗЦЗ («минимальную форму») числа 3:

$$3 = 100,01 = \Phi^2 + \Phi^{-2}.$$

Используя «свертки» и «развертки» и продолжая эти преобразования, нетрудно получить ЗЦЗ натуральных чисел 4 и 5:

$$4 = 101,01 = \Phi^2 + \Phi^0 + \Phi^{-2}$$
$$5 = 1000,1001 = \Phi^3 + \Phi^{-1} + \Phi^{-4}.$$

Продолжая этот процесс до бесконечности, можно получить ЗЦЗ всех натуральных чисел в системе Бергмана (9.1). А это означает, что мы доказали следующее утверждение.

**Теорема 9.1.** Любое натуральное число в системе Бергмана представляется в виде конечной суммы степеней золотой пропорции.

Это утверждение, возможно, представляет собой наиболее неожиданный результат, вытекающий из рассмотрения системы Бергмана (9.1).

А теперь возвратимся на 2,5 тысячелетия назад и представим себе реакцию пифагорейцев на Теорему 9.1. Согласно главной доктрине пифагорейцев «Все есть число», в основе мироздания лежат натуральные числа и их отношения, так как любую вещь в природе можно выразить как отношение двух натуральных чисел (хотя после открытия «несоизмеримых отрезков» эта доктрина была подвергнута сомнению, в результате чего и появилось понятие иррационального числа). Но мы только что показали, что любое натуральное число может быть выражено через «золотую пропорцию». Из этого рассуждения с необходимостью вытекает новая доктрина, которую пифагорейцы немедленно сформулировали бы, если бы знали о Теореме 9.1: «Все есть золотая пропорция»!

В начале главы 7 мы сделали весьма смелое утверждение о том, что позиционный принцип представления чисел, открытый Вавилонскими математиками и который лежит в основе всех позиционных систем счисления (десятичной, двоичной и т.д.), возможно, являются крупнейшим математическим открытием античной математики, которое вместе с наиболее распространенными системами счисления (десятичной и двоичной) оказало огромное влияние, как на развитие материальной культуры, так и современной технологии. Можно ли

представить современное общество без компьютеров и Интернета? Каждый человек в мире однозначно ответит, что компьютеры и Интернет являются эпохальными изобретениями, которые произвели революционные преобразования в современном обществе. Но не следует забывать, что в основе современных компьютеров лежит двоичная система счисления, которая является результатом коллективной творческой деятельности многих народов (Вавилон – позиционный принцип представления чисел, Древний Египет – «метод удвоения», Древний Китай – «Книга Перемен», Готфрид Лейбниц – создатель двоичной арифметики, Джон фон Нейман – создатель новых принципов конструирования электронных компьютеров).

Возникает вопрос: а как с этих позиций оценить открытие юного американского математика Джорджа Бергмана, который в 1957 г. описал весьма необычную систему счисления, названную им системой счисления с иррациональным основанием [55]? Отвечая на этот вопрос, мы можем сделать еще одно смелое заявление: система Бергмана является крупнейшим современным открытием в области систем счисления, сравнимым разве что с открытием позиционного принципа представления чисел, а также десятичной и двоичной систем счисления. Она переворачивает наши представления о системах счисления; более того - соотношение между рациональными и иррациональными числами. В этой системе на первый план выдвигается иррациональное число «золотая пропорция», которое становится основанием всех чисел, так как с его помощью может быть представлено любое действительное число. И очевидно, что этот математический результат представляет фундаментальный интерес для «элементарной теории чисел»!

Хотя система Бергмана представляла собой результат принципиальной важности не только для систем счисления, но и для теории чисел, но в тот период (50-е годы прошлого столетия) она просто не была замечена ни математиками, ни инженерами. И свое математическое открытие юный математик Джордж Бергман оценил весьма скромно [55]:

"Я не знаю никакого полезного приложения для систем счисления, подобных этой, кроме как для умственного упражнения и приятного

времяпровождения, хотя эта система счисления может быть полезной для алгебраической теории чисел".

Однако развитие компьютерной и измерительной техники опровергло пессимистическое мнение Бергмана относительно практического приложения его системы счисления. В отличие от классической двоичной системы, система счисления Бергмана обладает "естественной" избыточностью, которая может быть эффективно использована для контроля компьютеров. В 70-е и 80-е годы 20-го столетия в бывшем Советском Союзе под руководством автора настоящей книги были выполнены научные и инженерные разработки, основанные на системе счисления Бергмана [12,18,56,63,64]. Эти разработки, с одной стороны, показали исключительную эффективность системы Бергмана для проектирования самокорректирующихся аналого-цифровых и цифроаналоговых преобразователей и помехоустойчивых процессоров [18], а с другой стороны, привели к новому геометрическому определению действительного числа [117], что может стать началом так называемой «золотой» теории чисел.

#### 9.4. Новое конструктивное определение действительного числа

**Евклидово определение натурального числа.** Если возвратиться к истокам элементарной теории чисел, изложенной в «Началах» Евклида, то мы увидим, что она начинается со следующего определения натурального числа, известного под названием «Евклидоваго определения»:

Пусть

$$S = \{1,1,1,\dots\} \quad (9.16)$$

представляет собой бесконечное множество геометрических отрезков, называемых «монадами» или единицами. Тогда, согласно Евклиду, натуральное число  $N$  может быть определено следующим образом:

$$N = \underbrace{1+1+\dots+1}_N. \quad (9.17)$$

Несмотря на кажущуюся простоту такого определения, оно сыграло большую роль в развитии теории чисел и лежит в основе многих полезных математических понятий, в частности, понятий простого и составного числа, умножения, деления, а также понятий делимости и сравнения, которые являются одними из основных понятий элементарной теории чисел, то есть, определение (9.17) «порождает» как натуральные числа, так и всю проблематику их теории.

**Конструктивный подход к определению числа.** Известен также «конструктивный подход» к определению числа, согласно которому всякое «конструктивное» действительное число  $A$  является некоторым математическим объектом, задаваемым с помощью математической формулы (9.2).

Определение числа, задаваемое двоичной системой (9.2), имеет следующую геометрическую интерпретацию. Пусть

$$B = \{2^i, i = 0, \pm 1, \pm 2, \pm 3, \dots\} \quad (9.18)$$

- множество геометрических отрезков длины  $2^i$ . Тогда «конструктивными» действительными числами называются все математические объекты, которые могут быть представлены в виде конечной суммы геометрических отрезков из (9.18) в виде (9.2).

Ясно, что определение (9.2) выделяет из множества действительных чисел только некоторую часть чисел, которые могут быть представлены в виде конечной суммы (9.2). Такие числа мы будем называть конструктивными. Все остальные действительные числа, которые не могут быть представлены в виде суммы (9.2), являются неконструктивными. Ясно, что к «неконструктивным» числам относятся, прежде всего, все иррациональные числа, в частности, главные математические константы  $\pi$  и  $e$ , число  $\sqrt{2}$ , «золотая пропорция» и т.д. Но в рамках определения (9.2) к разряду «неконструктивных» мы должны отнести и некоторые рациональные числа (например,  $2/3$ ,  $3/7$  и т.д.), называемые «периодическими дробями», которые не могут быть представлены в виде конечной суммы (9.2). Заметим, что такой подход к числам, то есть, их деление на «конструктивные» и «неконструктивные», коренным образом отличается от принятого в математике

деления чисел на «рациональные» и «иррациональные», так как к разряду «неконструктивных» относятся все «периодические дроби», а к классу «конструктивных», как будет показано ниже, могут быть отнесены некоторые иррациональные числа (степени «золотой пропорции» и их суммы).

Заметим, что хотя определение (9.2) значительно ограничивает множество действительных чисел, это никак не умаляет его значение с «практической», вычислительной точки зрения. Ясно, что любое «неконструктивное» действительное число может быть представлено в виде (9.2) приближенно, причем ошибка приближения  $\Delta$  будет неограниченно уменьшаться по мере увеличения количества членов в (9.2), однако  $\Delta \neq 0$  для «неконструктивных» действительных чисел. По существу, в современных компьютерах мы пользуемся только «конструктивными» числами, задаваемыми (9.2), и это нас вполне устраивает, потому что любое «неконструктивное» число может быть представлено в виде (9.2) с погрешностью, потенциально стремящейся к 0.

**Определение Ньютона.** В течение многих тысячелетий математики развивали и уточняли понятие действительного числа. В 17-м веке в период зарождения современной науки и математики разрабатывается ряд методов изучения непрерывных процессов, и понятие действительного числа вновь выходит на передний план. Наиболее отчетливо новое определение этого понятия дается одним из основоположников математического анализа И. Ньютоном в его «Всеобщей Арифметике»:

«Под числами мы понимаем не столько множество единиц, сколько отвлеченное отношение какой-нибудь величины к другой величине того же рода, принятой нами за единицу».

Эта формулировка дает нам единое определение действительного числа, рационального или иррационального. Если теперь рассмотреть Евклидово определение (9.17) с точки зрения определения Ньютона, то в качестве «другой величины того же рода, принятой за единицу», выступает «монада». В двоичной системе счисления (9.2) роль «единицы», играет число 2, то есть, основание системы счисления.



**Новое конструктивное определение действительного числа.** Рассмотрим теперь бесконечное множество геометрических отрезков, являющихся степенями золотой  $p$ -пропорции  $\Phi_p$  ( $p = 0, 1, 2, 3, \dots$ ):

$$B = \{\Phi_p^i, i = 0, \pm 1, \pm 2, \pm 3, \dots\}, \quad (9.19)$$

где все степени  $\Phi_p^i$  связаны между собой математическим тождеством:

$$\Phi_p^i = \Phi_p^{i-1} + \Phi_p^{i-p-1} = \Phi \times \Phi_p^{i-1}. \quad (9.20)$$

Используя множество (9.19), можно «сконструировать» следующий метод позиционного представления чисел [12,56]:

$$A = \sum_i a_i \Phi_p^i, \quad (9.21)$$

где,  $\Phi_p$  - основание системы счисления (9.21),  $\Phi_p^i$  - вес  $i$ -го разряда,  $a_i \in \{0, 1\}$  - двоичная цифра  $i$ -го разряда,  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ ,  $p = 0, 1, 2, 3, \dots$ .

Заметим, что выражение (9.21) включает в себя бесконечное количество позиционных способов представления чисел (систем счисления), так как каждому  $p$  ( $p = 0, 1, 2, 3, \dots$ ) соответствует своя система счисления типа (9.21). Заметим, что при  $p = 0$  основание  $\Phi_p = \Phi_0 = 2$  и система счисления (9.21) сводится к классической двоичной системе (9.2).

Рассмотрим случай  $p = 1$ . Для этого случая основанием системы счисления (9.21) является классическая «золотая пропорция»  $\Phi = \frac{1+\sqrt{5}}{2}$  и система (9.21) сводится к системе Бергмана (9.1).

Рассмотрим теперь случай  $p = \infty$ . Для этого случая основание  $\Phi_p$  стремится к 1, а это означает, что в пределе выражение (9.21) сводится к классическому Евклидовому определению, задаваемому (9.17), в котором под «единицей» («монадой») понимается бесконечно малый отрезок  $\Delta$ .

Таким образом, мы можем рассматривать позиционный способ представления чисел, задаваемый (9.21), как весьма широкое обобщение и развитие «Евклидоваго определения» (9.17) и конструктивного определения действительного числа, основанного на двоичной системе (9.2).

Подчеркнем еще раз, что определение (9.21) разделяет все действительные числа на две группы: «конструктивные числа», которые могут быть представлены в виде конечной суммы степеней золотой  $p$ -пропорции в виде (9.21), и «неконструктивные» числа, которые не могут быть представлены в виде конечной суммы (9.21).

Ясно, что все степени золотой  $p$ -пропорции типа  $\Phi_p^i, i = 0, \pm 1, \pm 2, \pm 3, \dots$  могут быть представлены в виде (9.21). Например,

$$\Phi_p^3 = 1000; \quad \Phi_p^2 = 100; \quad \Phi_p^1 = 10; \quad \Phi_p^{-1} = 0,1; \quad \Phi_p^{-2} = 0,01; \quad \Phi_p^{-3} = 0,001.$$

Это означает, что все иррациональные числа типа  $\Phi_p^i, i = 0, \pm 1, \pm 2, \pm 3, \dots$  (степени золотой  $p$ -пропорции) являются «конструктивными числами» в рамках определения (9.21). Из определения (9.21) также вытекает, что все действительные числа, являющиеся суммами степеней золотой  $p$ -пропорции, также являются «конструктивными числами» в смысле (9.21). Например, действительное число  $A = \Phi_p^2 + \Phi_p^{-1} + \Phi_p^{-3}$  в соответствии с (9.21) может быть представлено в виде следующей «золотой» цифровой записи:

$$A = 100.101.$$

Заметим, что возможность представления некоторых иррациональных чисел (степеней золотой  $p$ -пропорции и их сумм) в виде конечной совокупности «битов» является еще одним необычным свойством введенных выше позиционных представлений (9.21).

## 9.5. Элементы «золотой» теории чисел

**Z-свойство натуральных чисел.** Как упоминалось выше, «Евклидово определение» (9.17), «порождает» не только натуральные числа, но и всю проблематику их теории. Но тогда можно высказать предположение, что новое определение действительного числа, задаваемое (9.21), может стать источником новых теоретико-числовых результатов. Рассмотрим один из таких результатов, названных в [117] Z-свойством натуральных чисел.

Для доказательства этого свойства представим натуральное число  $N$  в системе Бергмана (9.1):

$$N = \sum_i a_i \Phi^i, i = 0, \pm 1, \pm 2, \pm 3, \dots \quad (9.22)$$

Представление натурального числа  $N$  в форме (9.22) будем называть  $\Phi$ -кодом натурального числа  $N$  [117].

Выше мы доказали, что сумма (9.22) для любого натурального  $N$  является конечной, то есть, любое натуральное число может быть представлено в виде конечной суммы степеней «золотой пропорции», и этот далеко не тривиальный математический результат сам по себе представляет определенный интерес и приоткрывает еще одну «тайну» натуральных чисел.

Рассмотрим еще раз выражение (9.6), связывающее числа Фибоначчи  $F_n$  и числа Люка  $L_n$  с «золотой пропорцией»  $\Phi$ .

Напомним, что числа Фибоначчи и Люка представляют собой две бесконечные числовые последовательности, простирающиеся от  $-\infty$  до  $+\infty$ , то есть, задаваемые как для положительных, так и для отрицательных значений индексов  $n$  (см. Табл. 9.1).

Таблица 9.1. Расширенные числа Фибоначчи и Люка

$n$	0	1	2	3	4	5	6	7	8	9	10
$F_n$	0	1	1	2	3	5	8	13	21	34	55
$F_{-n}$	0	1	-1	2	-3	5	-8	13	-21	34	-55
$L_n$	2	1	3	4	7	11	18	29	47	76	123
$L_{-n}$	2	-1	3	-4	7	-11	18	-29	47	-76	123

В дальнейшем мы будем использовать следующие известные тождества для чисел Фибоначчи и чисел Люка [4]:

$$F_{i+1} = \frac{L_i + F_i}{2} \quad (9.23)$$

$$L_{i+1} = F_{i+1} + 2F_i. \quad (9.24)$$

В справедливости этих тождеств можно убедиться, используя Табл.9.1.

А теперь подставим выражение (9.6) вместо  $\Phi^i$  в выражение (9.22). В результате получим:

$$N = \frac{1}{2}(A + B\sqrt{5}), \quad (9.25)$$

где

$$A = \sum_i a_i L_i \quad (9.26)$$

$$B = \sum_i a_i F_i. \quad (9.27)$$

Выражение (9.25) может быть также записано в виде:

$$2N = A + B\sqrt{5}. \quad (9.28)$$

Заметим, что двоичные цифры  $a_i$  в выражениях (9.26), (9.27) совпадают с соответствующими двоичными цифрами в выражении (9.22), задающем  $\Phi$ -код натурального числа  $N$ .

Рассмотрим теперь выражение (9.25). Если принять во внимание, что суммы (9.26), (9.27) всегда являются целыми числами, то возникает вопрос: при каких условиях выражение (9.25) может быть справедливым в общем случае, то есть, для любого натурального  $N$ ? Ответ очень простой: это возможно только в том случае, если член  $A$ , задаваемый (9.26), является четным числом, равным  $2N$ , а член  $B$ , задаваемый (9.27), тождественно равен 0, то есть:

$$A = \sum_i a_i L_i = 2N \quad (9.29)$$

$$B = \sum_i a_i F_i = 0. \quad (9.30)$$

А теперь сравним выражения (9.22) и (9.30). Так как двоичные цифры  $a_i$  в рассмотренных выражениях совпадают, то это означает, что выражение (9.30) может быть получено из выражения (9.22) путем простой замены всех степеней золотой пропорции  $\Phi^i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$  в формуле (9.22) соответствующими числами Фибоначчи  $F_i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$ .

Результаты (9.29) и (9.30) сформулируем в виде следующих теорем.

**Теорема 9.2 (Z-свойство натуральных чисел).** Если в выражении для  $\Phi$ -кода любого натурального числа  $N$ , задаваемого (9.22), заменить все степени золотой пропорции  $\Phi^i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$  соответствующими числами Фибоначчи

$F_i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$ , то возникающая при этом сумма  $\sum_i a_i F_i$  тождественно равна нулю независимо от исходного натурального числа  $N$ , то есть,

$$\sum_i a_i F_i = 0. \quad (9.31)$$

**Теорема 9.3 (D-свойство натуральных чисел).** Если в выражении для  $\Phi$ -кода любого натурального числа  $N$ , задаваемого (9.22), заменить все степени золотой пропорции  $\Phi^i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$  соответствующими числами Люка  $L_i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$ , то возникающая при этом сумма  $\sum_i a_i L_i$  равна удвоенному натуральному числу  $N$ , то есть,

$$\sum_i a_i L_i = 2N. \quad (9.32)$$

Примем без доказательства еще одно утверждение [47], которое является обобщением Теоремы 9.2.

**Теорема 9.4 ( $Z_p$ -свойство натуральных чисел).** Если представить в виде (9.21) некоторое натуральное число  $N$ , то есть, записать

$$N = \sum_i a_i \Phi_p^i, \quad (9.33)$$

где  $\Phi_p$  - основание системы счисления (9.33),  $\Phi_p^i$  - вес  $i$ -го разряда,  $a_i \in \{0, 1\}$  - двоичная цифра  $i$ -го разряда,  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ ,  $p = 0, 1, 2, 3, \dots$ , а затем в выражении (9.33) заменить все степени золотой  $p$ -пропорции  $\Phi_p^i$  соответствующими  $p$ -числами Фибоначчи  $F_p(i) (i = 0, \pm 1, \pm 2, \pm 3, \dots)$ , то возникающая при этом сумма  $\sum_i a_i F_p(i)$  всегда тождественно равна нулю независимо от исходного натурального числа  $N$ , то есть,

$$\sum_i a_i F_p(i) = 0. \quad (9.34)$$

Существенно подчеркнуть, что свойства (9.30), (9.32) и (9.34) справедливы только для натуральных чисел! Это означает, что спустя 2.5 тысячелетия после начала изучения натуральных чисел нам удалось открыть новые свойства

натуральных чисел, названных  $Z$ - или  $Z_p$ -свойствами (от слова «Zero» - ноль) [117]. И это стало возможным только благодаря введению новых способов представления натуральных чисел, задаваемых (9.22) и (9.33). Это означает, что сформулированный выше новый подход к геометрическому определению действительных чисел, задаваемый (9.1) и (9.21), может стать началом нового направления в теории чисел.

**Контроль арифметических операций над целыми числами.** Мы можем использовать  $Z$ -свойство натуральных чисел для контроля арифметических операций. Например, рассмотрим операцию алгебраического суммирования двух целых чисел  $N_1 \pm N_2$ . В результате суммирования мы всегда получим новое целое число. Если суммирование осуществляется над числами, представленными в  $\Phi$ -коде, то слагаемые и результат суммирования могут быть проверены на основе  $Z$ -свойства. Подобный же вывод справедлив для «золотого» умножения. Результатом «золотого» деления являются два целых числа, частное  $Q$  и остаток  $R$ . Это означает, что результат деления также может быть проверен с использованием  $Z$ -свойства.

## 9.6. $F$ - и $L$ -коды

**Понятия  $F$ - и  $L$ -кодов.** Принимая во внимание тождества (9.29) и (9.30), выражение (9.25) может быть также представлено в следующем виде:

$$N = \frac{1}{2}(A + B), \quad (9.35)$$

где  $A$  определяется выражением (9.29), а  $B$  - выражением (9.30).

Принимая во внимание выражения (9.29) и (9.30), а также тот факт, что все двоичные коэффициенты в этих выражениях совпадают, мы можем представить выражение (9.35) в следующем виде:

$$N = \sum_i a_i \frac{L_i + F_i}{2}. \quad (9.36)$$

Используя тождество (9.23), мы можем записать выражение (9.36) в виде:

$$N = \sum_i a_i F_{i+1}. \quad (9.37)$$

Выражение (9.37) будем называть  $F$ -кодом числа  $N$  [117]. Таким образом,  $F$ -код представляет собой двоичное позиционное представление натурального числа  $N$ , в котором весами разрядов являются расширенные числа Фибоначчи  $F_{i+1}$ , выбираемые из Табл. 9.1.

А теперь сравним выражения (9.22) и (9.37). Так как двоичные цифры  $a_i$  в этих выражениях совпадают для любого  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ , то отсюда вытекает, что  $F$ -код числа  $N$  может быть получен из  $\Phi$ -кода (9.22) того же самого натурального числа  $N$  путем простой замены степеней золотой пропорции  $\Phi^i$  в формуле (9.22) на соответствующие числа Фибоначчи  $F_{i+1}$  для всех  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ .

Представим теперь  $F$ -код числа  $N$  в следующей форме:

$$N = \sum_i a_i F_{i+1} + 2B, \quad (9.38)$$

где член  $B$  определяется выражением (9.30). Тогда выражение (9.38) может быть представлено в следующем виде:

$$N = \sum_i a_i (F_{i+1} + 2F_i). \quad (9.39)$$

Принимая во внимание тождество (9.24), выражение (9.39) может быть представлено в следующем виде:

$$N = \sum_i a_i L_{i+1}. \quad (9.40)$$

Будем называть выражение (9.40)  $L$ -кодом числа  $N$  [117]. Таким образом,  $L$ -код представляет собой двоичное позиционное представление натурального числа  $N$ , в котором весами разрядов являются расширенные числа Люка  $L_{i+1}$ .

Так как двоичные цифры в выражениях (9.22) и (9.40) совпадают, отсюда вытекает, что  $L$ -код числа  $N$  может быть получен из  $\Phi$ -кода (9.22) того же самого числа  $N$  путем замены степеней золотой пропорции  $\Phi^i$  в формуле (9.22) на соответствующие числа Люка  $L_{i+1}$  для всех  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ .

Ясно, что  $L$ -код числа  $N$  может быть получен также из  $F$ -кода того же самого числа  $N$  путем замены чисел Фибоначчи  $F_{i+1}$  в формуле (9.37) на числа Люка  $L_{i+1}$ .

Таким образом, существуют три различных способа представления одного и того же натурального числа  $N$ :  $\Phi$ -код (9.22), задающий натуральное число  $N$  в виде конечной суммы степеней золотой пропорции,  $F$ -код (9.37), задающий натуральное число  $N$  в виде суммы чисел Фибоначчи, и  $L$ -код (9.40), задающий натуральное число  $N$  в виде суммы чисел Люка; при этом по форме двоичной записи все эти коды совпадают.

В качестве примера рассмотрим представление десятичного числа 10 (основание десятичной системы) в  $\Phi$  коде (9.22):

$$10 = 10100.0101. \quad (9.41)$$

Для  $\Phi$ -кода (9.22) ЗЦЗ (9.41) имеет следующую алгебраическую интерпретацию:

$$10 = \Phi^4 + \Phi^2 + \Phi^{-2} + \Phi^{-4}. \quad (9.42)$$

Используя формулу (9.6), мы можем представить сумму (9.42) следующим образом:

$$10 = \frac{L_4 + F_4\sqrt{5}}{2} + \frac{L_2 + F_2\sqrt{5}}{2} + \frac{L_{-2} + F_{-2}\sqrt{5}}{2} + \frac{L_{-4} + F_{-4}\sqrt{5}}{2}. \quad (9.43)$$

Используя Табл.9.1, мы можем записать следующее:

$$L_{-2} = L_2; L_{-4} = L_4; F_{-2} = -F_2; F_{-4} = -F_4. \quad (9.44)$$

Тогда с учетом (9.44) выражение (9.43) может быть записано в следующем виде:

$$10 = \frac{2(L_4 + L_2)}{2} = L_4 + L_2 = 7 + 3.$$

Теперь рассмотрим интерпретацию ЗЦЗ (9.41) в виде  $F$ - и  $L$ -кодов:

$$F\text{-код: } 10 = F_5 + F_3 + F_{-1} + F_{-3} = 5 + 2 + 1 + 2$$

$$L\text{-код: } 10 = L_5 + L_3 + L_{-1} + L_{-3} = 11 + 4 - 1 - 4$$

Также мы можем проверить справедливость  $Z$ -свойства для ЗЦЗ (9.42). Если мы заменим степени  $\Phi^i$  в (9.42) на соответствующие числа Фибоначчи  $F_i$  для всех  $i = 0, \pm 1, \pm 2, \pm 3, \dots$ , мы получим следующий результат:



$$Z\text{-свойство: } F_4 + F_2 + F_{-2} + F_{-4} = 3 + 1 + (-1) + (-3) = 0.$$

**Сдвиги  $\Phi$ -,  $F$ - и  $L$ -кодов.** Еще одно необычное теоретико-числовое свойство системы Бергмана (9.1) возникает при сдвиге  $\Phi$ -,  $F$ - и  $L$ -кодов.

Еще раз подчеркнем, что «золотые» цифровые записи одного и того же натурального числа  $N$  в  $\Phi$ -,  $F$ - и  $L$ -кодах совпадают и по форме их ЗЦЗ невозможно определить, о каком из кодов идет речь. Однако, различие между  $\Phi$ -,  $F$ - и  $L$ -кодами сразу же возникает, когда мы начинаем сдвигать их ЗЦЗ влево или вправо.

Обозначим через  $N_{(k)}$  и  $N_{(-k)}$  - числовые результаты сдвига некоторой ЗЦЗ, задающей число  $N$ , на  $k$  разрядов влево или вправо, соответственно.

Оказывается, что числовые результаты  $N_{(k)}$  и  $N_{(-k)}$  будут различаться, в зависимости от того, как мы будем интерпретировать ЗЦЗ ( $\Phi$ -,  $F$ - или  $L$ -код).

**Сдвиги  $\Phi$ -кода.** Если мы будем интерпретировать ЗЦЗ, как представление числа  $N$  в  $\Phi$ -коде, тогда сдвиг влево (то есть, в сторону старших разрядов) на один разряд будет соответствовать умножению числа  $N$  на основание  $\Phi$ , а сдвиг вправо (то есть, в сторону младших разрядов) – делению числа  $N$  на основание  $\Phi$  (или умножению на  $\Phi^{-1}$ ), то есть,

$$N_{(k=1)} = N \times \Phi = \sum_i a_i \Phi^{i+1} \quad (9.45)$$

$$N_{(k=-1)} = N \times \Phi^{-1} = \sum_i a_i \Phi^{i-1}. \quad (9.46)$$

Ясно, что при сдвиге ЗЦЗ, интерпретируемой как  $\Phi$ -код числа  $N$ , на  $k$  разрядов влево или вправо будет соответствовать умножению числа  $N$  на  $\Phi^k$  или  $\Phi^{-k}$ , соответственно, то есть сдвиг приведет к получению следующих чисел:

$$N_{(k)} = N \times \Phi^k = \sum_i a_i \Phi^{i+k} \quad (9.47)$$

$$N_{(-k)} = N \times \Phi^{-k} = \sum_i a_i \Phi^{i-k}. \quad (9.48)$$

**Сдвиги  $F$ -кода.** Если мы будем интерпретировать ЗЦЗ числа  $N$  в  $F$ -коде (9.37), тогда в результате сдвига ЗЦЗ на  $k$  разрядов влево мы получим следующую сумму:

$$N_{(k)} = \sum_i a_i F_{i+k+1}. \quad (9.49)$$

В «теории чисел Фибоначчи» [4] известны обобщенные числа Фибоначчи, которые задаются рекуррентным соотношением  $G_n = G_{n-1} + G_{n-2}$  при некоторых начальных условиях  $G_0$  и  $G_1$ . Это рекуррентное соотношение «порождает» бесконечное количество рекуррентных числовых последовательностей, которые зависят от начальных условий  $G_0$  и  $G_1$ . В частности, при  $G_0 = 0$  и  $G_1 = 1$  мы получим числа Фибоначчи  $F_n$ , а при  $G_0 = 2$  и  $G_1 = 1$  - числа Люка  $L_n$ .

В книге [4] доказано следующее тождество, связывающее числа  $G_n$  и  $F_n$ :

$$G_{n+m} = F_{m-1} G_n + F_m G_{n+1}. \quad (9.50)$$

Для случая  $G_k = F_k$  тождество (9.50) принимает вид:

$$F_{n+m} = F_{m-1} F_n + F_m F_{n+1}. \quad (9.51)$$

При  $n = i$  и  $m = k + 1$  тождество (9.51) принимает вид:

$$F_{i+1+k} = F_k F_i + F_{k+1} F_{i+1}. \quad (9.52)$$

Подставляя (9.52) в выражение (9.49), мы получим:

$$N_{(k)} = \sum_i a_i F_{i+k+1} = \sum_i a_i (F_k F_i + F_{k+1} F_{i+1}) = F_k \sum_i a_i F_i + F_{k+1} \sum_i a_i F_{i+1}. \quad (9.53)$$

Учитывая (9.30) и (9.37), мы можем записать выражение (9.53) в следующем виде:

$$N_{(k)} = F_{k+1} N. \quad (9.54)$$

Рассмотрим теперь, к чему приведет сдвиг вправо на  $k$  разрядов ЗЦЗ, интерпретируемой как  $F$ -код числа  $N$ . В результате такого сдвига мы получим следующую сумму:

$$N_{(-k)} = \sum_i a_i F_{i-k+1}. \quad (9.55)$$

Если принять  $n = i$  и  $m = -k + 1$ , мы можем записать тождество (9.51) в следующем виде:

$$F_{i-k+1} = F_{-k}F_i + F_{-k+1}F_{i+1}. \quad (9.56)$$

Подставляя (9.56) в выражение (9.55), после несложных преобразований с учетом (9.30) и (9.37), мы получим:

$$N_{(k)} = \sum_i a_i F_{i-k+1} = F_{-k+1} N. \quad (9.57)$$

Сформулируем полученные результаты в виде следующей теоремы.

**Теорема 9.5.** Сдвиг «золотой» цифровой записи, интерпретируемой как  $F$  – код натурального числа  $N$ , на  $k$  разрядов влево (то есть, в сторону старших разрядов) соответствует умножению числа  $N$  на число Фибоначчи  $F_{k+1}$ , а ее сдвиг на  $k$  разрядов вправо (то есть, в сторону младших разрядов) соответствует умножению числа  $N$  на число Фибоначчи  $F_{-k+1}$ .

Исследуем теперь формулу (9.57). В случае сдвига на один разряд вправо ( $k=1$ ) формула (9.57) принимает следующий вид:

$$\sum_i a_i F_i = F_0 N. \quad (9.58)$$

Поскольку число Фибоначчи  $F_0=0$ , то формула (9.58) превращается в формулу (9.30), которая задает  $Z$  – свойство.

Заметим, что формула (9.57) может быть использована в ряде приложений. При сдвиге вправо на три разряда, формула (9.57) принимает вид:

$$N_{(k=-3)} = F_{-2} N. \quad (9.59)$$

Учитывая, что  $F_{-2} = -1$ , формула (9.59) принимает вид:

$$N_{(k=-3)} = -N. \quad (9.60)$$

Формула (9.60) вместе с формулой (9.30), задающей  $Z$  – свойство  $\Phi$  – кода, может быть использована для контроля измерительных систем и числовых преобразований в компьютере.

**Сдвиги  $L$ -кода.** А теперь исследуем сдвиги ЗЦЗ, интерпретируемых как  $L$  – код числа  $N$ , задаваемый (9.40). Сдвигая такую ЗЦЗ вправо и влево на  $k$  разрядов, мы получим следующие суммы:

$$N_{(k)} = \sum_i a_i L_{i+k+1} \quad (9.61)$$

$$N_{(-k)} = \sum_i a_i L_{i-k+1}. \quad (9.62)$$

Принимая  $G_k = L_k$  и используя тождество (9.50), мы можем представить числа Люка  $L_{i+k+1}$  и  $L_{i-k+1}$  следующим образом:

$$L_{i+1+k} = L_k F_i + L_{k+1} F_{i+1} \quad (9.63)$$

$$L_{i-k+1} = L_{-k} F_i + L_{-k+1} F_{i+1}. \quad (9.64)$$

Тогда выражения (9.61) и (9.62) могут быть представлены в следующем виде:

$$N_{(k)} = \sum_i a_i L_{i+k+1} = L_k \sum_i a_i F_i + L_{k+1} \sum_i a_i F_{i+1} \quad (9.65)$$

$$N_{(-k)} = \sum_i a_i L_{i-k+1} = L_{-k} \sum_i a_i F_i + L_{-k+1} \sum_i a_i F_{i+1}. \quad (9.66)$$

Учитывая (9.30) и (9.37), мы можем представить выражения (9.65) и (9.66) в следующем виде:

$$N_{(k)} = L_{k+1} N \quad (9.67)$$

$$N_{(-k)} = L_{-k+1} N. \quad (9.68)$$

Полученные результаты мы можем сформулировать в виде следующей теоремы.

**Теорема 9.6.** Сдвиг «золотой» цифровой записи, интерпретируемой как  $L$ -код натурального числа  $N$ , на  $k$  разрядов влево (то есть, в сторону старших разрядов) соответствует умножению числа  $N$  на число Люка  $L_{k+1}$ , а ее сдвиг на  $k$  разрядов вправо (то есть, в сторону младших разрядов) соответствует умножению числа  $N$  на число Люка  $L_{-k+1}$ .

Рассмотрим формулу (9.68). Для случая  $k=1$  (сдвиг вправо на один разряд) формула (9.68) принимает следующий вид:

$$\sum_i a_i L_i = L_0 N. \quad (9.69)$$

Так как число Люка  $L_0 = 2$ , то отсюда вытекает, что формула (9.69) сводится к формуле (9.32), задающей  $D$ -свойство.

Заметим, что  $F$ -и  $L$ -коды, задаваемые (9.37) and (9.40), и их удивительные свойства, задаваемые Теоремами 9.5, 9.6, являются дополнительными свидетельствами продуктивности конструктивного подхода к расширению теории чисел, который вытекает из системы Бергмана (9.1).



## Глава 10

### МИКРОПРОЦЕССОРЫ ФИБОНАЧЧИ

#### 10.1. Проблема контролеспособности и помехоустойчивости микропроцессоров, используемых в системах управления

**Основные направления развития микропроцессоров.** В современном мире трудно найти область техники, где не применялись бы микропроцессоры. Они применяются при вычислениях, для выполнения функций управления, а также при обработке звука и изображения. В зависимости от области применения микропроцессора изменяются требования к нему. Это накладывает отпечаток на внутреннюю структуру микропроцессора. В настоящее время определилось три направления развития микропроцессоров:

- универсальные микропроцессоры;
- микроконтроллеры;
- сигнальные микропроцессоры.

Универсальные микропроцессоры используются для построения вычислительных машин. В них используются самые передовые решения по повышению быстродействия, при этом меньшее внимание обращается на габариты, стоимость и потребляемую энергию.

Микроконтроллеры используются для управления малогабаритными и дешёвыми устройствами связи; они раньше назывались однокристалльными микроЭВМ. В микроконтроллерах, в отличие от универсальных микропроцессоров, особое внимание уделяется именно габаритам, стоимости и потребляемой энергии.

Сигнальные процессоры используются для решения задач, которые традиционно решала аналоговая схемотехника. К сигнальным процессорам предъявляются специфические требования. От них требуются максимальное быстродействие, малые габариты, легкая стыковка с аналого-цифровыми и цифро-аналоговыми преобразователями, большая разрядность обрабатываемых данных и

небольшой набор математических операций, обязательно включающий операцию умножения-накопления и аппаратную организацию циклов. В этих микропроцессорах тоже важны такие параметры как стоимость, габариты и потребляемая мощность, но здесь приходится мириться с большими значениями этих характеристик по сравнению с микроконтроллерами.

**Академик Хетагуров о низкой информационной надежности микропроцессоров.** Традиционно основное внимание при создании микропроцессоров и микроконтроллеров уделялось повышению быстродействия, уменьшению стоимости и потребляемой мощности. К сожалению, меньшее внимание уделялось проблеме повышения информационной надежности, помехоустойчивости, контролеспособности, достоверности данных на выходе микропроцессоров и микроконтроллеров. Это обстоятельство в последние годы вызывает тревогу многих известных специалистов в области микропроцессорной техники. В статье академика Я.А. Хетагурова «Обеспечение национальной безопасности систем реального времени» (ВС/NW 2009; №2 (15):11.1) высказаны интересные соображения, касающиеся использования микропроцессоров и микроконтроллеров иностранного производства:

«Применение микропроцессоров, контроллеров и программного обеспечения вычислительных средств (ВС) иностранного производства для решения задач в системах реального времени (СРВ) военного, административного и финансового назначения таит в себе большие проблемы. Это своего рода «тroyанский конь», роль которого только стала проявляться. Потери и вред от их использования могут существенно повлиять на национальную безопасность России... Отсутствие в иностранных вычислительных средствах широкого профиля контроля, необходимого для обеспечения требуемой достоверности выдаваемых данных в СРВ, приводит либо к использованию программных методов контроля, которые увеличивают быстродействие в 1,5-2,5 раза и потребление электроэнергии, либо применению мажоритарного метода контроля, использующего 3 вычислительных устройства ШП, что повышает требования к быстродействию на



10-15%, однако увеличивает объём аппаратуры ВС в среднем в 3,3 раза и потребление электроэнергии в 3,4 раза».

Таким образом, микропроцессоры иностранного производства, широко используемые в современных российских разработках, по мнению академика Хетагурова, являются «троянским конем», который угрожает национальной безопасности систем реального времени военного, административного и финансового назначения.

## **10.2. О возможных причинах участившихся катастроф при запуске российских и иностранных ракет**

К сожалению, при запуске ракет иногда возникают катастрофы. Особенно часто эти катастрофы происходили в 2010, 2011 годах. Вот перечень наиболее известных из них:

1 февраля 2011 г. ракета - носитель "Рокот" с разгонным блоком "Бриз-КМ" вывела с космодрома "Плесецк" геодезический спутник "Гео-ИК-2" на нерасчётную орбиту.

18 августа 2011 г. с космодрома "Байконур" не выведен на орбиту телекоммуникационный геостационарный спутник "Экспресс-АМ4" Госпредприятия "Космическая связь". Причиной аварии была названа небрежность, допущенная специалистами РКК "Энергия" и недостаточный контроль со стороны "Роскосмоса".

24 августа 2011 г. надёжнейшая ракета - носитель "Союз-У" с космодрома "Байконур" не вывела на орбиту для стыковки с МКС грузовой корабль "Прогресс М-12-м", который долго искали затем на Алтае. Авария случилась на 325-й секунде полета.

9 ноября 2011 г. неудачно стартовала межпланетная станция "Фобос - Грунт", связь с которой долго не могли установить из-за нерасчётной орбиты. До 15 января станция бороздила околоземное пространство и упала в Тихий океан.

23 декабря 2011 г. с космодрома "Плесецк" спутник связи "Меридиан" не был выведен ракетой - носителем "Союз - 2.1б" - и упал на жилые дома в Новосибирской области.

За рассматриваемый период в США неудачно стартовали и были потеряны 5 космических аппаратов, в Китае - один.

Статья «О возможной причине участвовавших аварий при выводе российских спутников» [118] написана автором как реакция на сообщения СМИ об аварийном запуске космического аппарата связи «Меридиан» (23 декабря 2011 г.). В СМИ сообщалось, что причиной падения спутника "Меридиан-5" стали неполадки в двигательной установке третьей ступени ракеты "Союз 2.1б", которая выводила аппарат в космос. В статье [118] были высказаны следующие соображения по поводу возможной причины этой аварии:

«Одной из возможных причин ... могут быть «сбои» в цифровой системе управления двигателями. По странному совпадению, аварии начались после усовершенствования системы управления и ее переводе на цифровую технику. И вот в этом, возможно, и «зарыта собака». Дело в том, что цифровые системы управления, основанные на современных микропроцессорах, обладают очень низкой информационной надежностью по сравнению с аналоговыми системами. Иногда достаточно сбоя одного электронного элемента (триггера) в микропроцессоре системы управления для того, чтобы система начала выполнять ложную команду, что может стать причиной аварии. Сбой цифровой системы управления вызывается как внутренними, так и внешними факторами. Сбой может возникнуть, например, в результате мощного внешнего электромагнитного воздействия на ракету-носитель в период запуска (электромагнитный терроризм)».

Следующая статья автора на эту тему [119] была реакцией на обнародованные в СМИ выводы комиссии Роскосмоса о причинах аварии при запуске межпланетной станции «Фобос-Грунт» (9 ноября 2011 г.). По этому поводу в СМИ было опубликовано следующее сообщение:

«При создании автоматической межпланетной станции "Фобос-Грунт" использовались некондиционные импортные микросхемы, это могло привести к нештатной ситуации с аппаратом», - заявил глава Роскосмоса Владимир Поповкин,

пишет РИА Новости. - Там микросхемы, которые были применены, в отличие от тех микросхем, которые раньше применялись на "Фрегате", они меньше по типономиналу. Там вместо 200 нанометров где-то 90 нанометров. Но это импортная компонентная база, и, конечно, вот причина, наверное, в этом", - сказал Поповкин на совещании по развитию отечественной космонавтики.

«Применение импортных микросхем - это болезнь не только наша, но и целые разборки идут в последнее время и в НАСА, и Минобороны США по контрафакту этой продукции", - заявил глава Роскосмоса.

Таким образом, впервые Роскомос официально признал, что причиной аварии является применение импортных микросхем, что является «головной болью» не только Роскомоса, но и НАСА и Минобороны США. И тем самым была подтверждена гипотеза, высказанная в статье [118].

### **10.3. «Троянский конь» двоичной системы счисления и «фибоначчиевы» патенты**

**Главный недостаток двоичной системы счисления.** Как известно, двоичная система была введена в компьютерную технику в 1946 г. Джоном фон Нейманом совместно с его коллегами по Принстонскому институту перспективных исследований. Одним из «неймановских принципов» было обоснование использования в электронных компьютерах двоичной системы счисления. На тот период это было абсолютно правильное и взвешенное решение, так как двоичная система в наибольшей степени отвечала двоичному характеру электронных элементов и требованиям булевой логики. Кроме того, следует учитывать то обстоятельство, что в тот период других, альтернативных систем счисления в науке просто не существовало. Выбор был очень небольшой: десятичная система или двоичная система. Предпочтение было отдано двоичной системе. Однако вместе с двоичной системой в компьютерную технику был введен «троянский конь» в виде нулевой избыточности двоичной системы. Отсутствие избыточности означает, что все двоичные кодовые комбинации в рамках двоичной системы являются «разрешенными», что делает невозможным обнаружение каких-либо ошибок,

которые неизбежно (с большей или меньшей вероятностью) могут возникнуть в элементах электронных систем под влиянием различных внешних и внутренних факторов (радиация, электромагнитные воздействия, помехи в шинах питания и т.д.).

Таким образом, из вышеизложенного вытекает далеко не оптимистичный вывод. Человечество становится заложником классической двоичной системы счисления, которая лежит в основе современных микропроцессоров и информационных технологий. Поэтому дальнейшее развитие микропроцессорной техники и основанной на ней информационной технологии, использующей двоичную систему счисления, следует признать неприемлемым направлением для определенных сфер приложений. Двоичная система не может служить информационной и арифметической основой специализированных компьютерных и измерительных систем (космос, управление транспортом и сложными технологическими объектами), а также наноэлектронных систем, где проблемы надежности, помехоустойчивости, контролеспособности, стабильности, живучести систем выходят на передний план.

К сожалению, микроэлектроника вынуждена была взять на вооружение все технические решения классической компьютерной техники вместе с двоичной системой. Вместе с ней ее «троянский конь» переселился в микропроцессоры и микроконтроллеры. В настоящее время двоичная система начинает постепенно завоевывать свои позиции и в наноэлектронике. Это означает, что «троянский конь» двоичной системы («нулевая избыточность») беспрепятственно может переселиться в наноэлектроннику, что может привести к непредсказуемым последствиям для дальнейшего развития информационных технологий (статья академика Хетагурова является первым серьезным предупреждением для современного информационного сообщества).

**Первые работы по избыточным системам счисления.** Впервые исследования по избыточным способам позиционного представления чисел были предприняты автором настоящей книги в начале 70-х годов 20 в. в Таганрогском радиотехническом институте (ТРТИ) (1971-1976). В докторской диссертации

автора (1972) были синтезированы так называемые фибоначчиевые алгоритмы измерения, которые привели к введению  $p$ -кодов Фибоначчи как новых способов позиционного представления натуральных чисел. В этот же период были опубликованы первые работы автора по избыточным позиционным системам счисления [59-61]. Позже (1980) автором были введены коды золотой  $p$ -пропорции [56]. Основы теории  $p$ -кодов Фибоначчи и кодов золотой  $p$ -пропорции изложены в книгах автора [9,12].

**Фибоначчиевые патенты.** В 1976 г. автор настоящей книги в течение двух месяцев стажировался в Венском техническом университете (Австрия). На заключительной стадии пребывания в Австрии автор выступил с докладом «Алгоритмическая теория измерения и основания компьютерной арифметики» на объединенном заседании Компьютерного и Кибернетического обществ Австрии. Доклад вызвал большой интерес австрийских ученых. В связи с этим посол СССР в Австрии Ефремов написал письмо в Государственный комитет СССР по науке и технике письмо, которое содержало следующее предложение:

«С учетом выраженного интереса у австрийских ученых к изобретению проф. Стахова А.П. по вопросу создания новой системы исчисления на основе "фибоначчиевых" чисел (создание самоконтролирующихся ЦВМ) считали бы целесообразным ускорить процесс оформления его заявок на изобретение, что позволит также сохранить приоритет советской науки и, возможно, получить экономический эффект».

Предложение Посла СССР в Австрии было вынесено на государственный уровень и, начиная с 1976 г., во всех ведущих странах-производителях компьютерной техники (США, Япония, Англия, Франция, Германия, Канада и др. страны) было начато широкое патентование советских изобретений по направлению «компьютеры Фибоначчи».

Как известно, изобретательская деятельность является одной из форм творческой деятельности. Имена выдающихся изобретателей широко известны (Эдисон, Стив Джобс и др.). Каждое изобретение является определенным творческим шагом по сравнению с известными решениями в этой области,

называемыми прототипами или аналогами. Как и во всякой области творческой деятельности, изобретение изобретению - рознь. Особую ценность представляют изобретения пионерного характера, то есть, изобретения, которые не имеют прототипа или аналога. Их появление можно считать своеобразным открытием в технической области.

Документом, защищающим авторские права на изобретение, в бывшем СССР являлось авторское свидетельство, а в странах Запада - патент. Необходимо также отметить, что защита изобретения с помощью авторского свидетельства или патента имеет локальный характер, то есть, они защищают права автора или страны только на территории того государства, в котором выдан патент. В частности, за пределами СССР изобретение, на которое было выдано авторское свидетельство в СССР, могло быть использовано любой фирмой без каких-либо юридических или экономических санкций со стороны СССР. Однако в тех случаях, когда государство признавало особую ценность того или иного изобретения, а также тогда, когда речь шла о защите приоритета или экономических интересов страны, ставился вопрос о патентной защите такого изобретения в других странах. В качестве таких стран, как правило, выбирались потенциальные производители продукции, которая может быть создана на основе этого изобретения.

Для получения зарубежного патента оформляется соответствующая заявка с учетом требований патентного ведомства той или иной страны, и эта заявка подается в это ведомство с ходатайством выдать патент на изобретение. Надо отметить, что патентное ведомство какой-либо страны не заинтересовано в выдаче патента, потому что это ущемляет права собственных производителей, и с этой целью проводится весьма тщательная патентная экспертиза, которая во многих случаях заканчивается отказом в выдаче патента. Необходимо также отметить, что все услуги по патентованию (даже в случае отрицательного решения) оплачивает страна, которая ходатайствует о выдаче патента. Вот почему решение о патентовании советских изобретений за рубежом (особенно в области электроники и вычислительной техники) принимались только в исключительных случаях. Таким исключительным случаем оказались изобретения в области «компьютеров Фибоначчи»

Главная цель патентования состояла в том, чтобы защитить приоритет советской науки. Предметом патентования была новая компьютерная арифметика. Однако, в соответствии с патентными законами большинства стран невозможно получить патент на математическое изобретение, каким, в сущности, и была "фибоначчиева" арифметика. Поэтому возникла мысль о косвенной защите этой арифметики через компьютерные устройства, ее реализующие. Такими устройствами являются регистры, счетчики, сумматоры, множительные и делительные устройства и т.д., то есть, основные операционные устройства «компьютера Фибоначчи». При этом желательно было придумать такой оригинальный операционный элемент, который мог бы претендовать на пионерное изобретение, с использованием которого можно было бы создать все остальные операционные элементы. В результате таких рассуждений родилась идея многозвенной формулы изобретения, первый пункт которой и защищал пионерное "фибоначчиевое" изобретение.

Что же должно было стать пионерным "фибоначчиевым" изобретением? Таким изобретением стало устройство приведения кода Фибоначчи к минимальной форме [69]. Изобретение не имело аналогов и прототипа, и было признано в СССР, а затем и в других странах изобретением пионерного характера. Затем были разработаны другие операционные элементы компьютера Фибоначчи (в частности, счетчики и сумматоры). Это также были новые элементы компьютерной техники, но они не были пионерными изобретениями, поскольку все они основывались на "устройстве приведения кода Фибоначчи к минимальной форме" [69].

С экономической точки зрения (стоимость патентования) выгоднее патентовать заявку на изобретение с так называемой многозвенной формулой. По существу речь шла о такой крупномасштабной заявке, которая включала бы в себя несколько десятков технических решений, объединенных общей идеей и общей формулой изобретения. Вот такой и была первая заявка на изобретение «Устройство приведения  $p$ -кодов Фибоначчи к минимальной форме». Ее центральным (пионерным) техническим решением был способ приведения кода Фибоначчи к минимальной форме и устройство для его реализации; на основе

этого устройства строились все остальные операционные узлы компьютера Фибоначчи.

Первая заявка, принятая к патентованию, по своему объему представляла собой книгу средних размеров, так как содержала свыше 200 страниц текстового материала, около 100 рисунков (операционные устройства и их элементы), а многозвенная формула изобретения состояла из 85 пунктов. Это означало, что на патентование выдвигалось 85 технических решений, то есть, 85 изобретений. Всего же к патентованию было принято 12 заявок.

Патентование осуществлялось в 8 странах (США, Япония, Англия, Франция, ФРГ, Канада, Польша и ГДР). Для этого каждая из 12 заявок составлялась с учетом патентного закона каждой страны и затем переводилась на соответствующий язык. Таким образом, по каждой заявке авторам приходилось работать с 8 экспертами-патентоведами (специалистами по патентным законам соответствующих стран) и с 5 техническими переводчиками (английский, французский, немецкий, польский и японский языки).

Каковы же итоги этого беспрецедентного патентования? На советские изобретения по «компьютерам Фибоначчи» выдано свыше 60 зарубежных патентов США, Японии, Англии, Франции, ФРГ, Канады, Польши и ГДР. Это свидетельствует о том, что идея компьютера Фибоначчи является абсолютно новой и оригинальной, а указанные патенты являются официальными юридическими документами, подтверждающими приоритет советской науки (и автора настоящей книги) в новом направлении в области компьютерной техники. Упомянутое выше «Устройство приведения р-кодов Фибоначчи к минимальной форме» [69] являлось основным техническим решением из перечня решений, представленных к патентованию, и на него было выдано 13 патентов [70-82].

В 1977 г. автор был избран на должность зав. кафедрой вычислительной техники Винницкого политехнического института (Украина), а в 1986 г. был назначен директором Специального конструкторско-технологического бюро «Модуль» при Винницком политехническом институте. Именно в этом институте были выполнены инженерные разработки, основанные на кодах Фибоначчи и «золотой пропорции», которые в тот период превышали по своим техническим



параметрам мировой уровень [18]. Разработки велись в интересах Министерства общего машиностроения (советское ракетное министерство), которое и финансировало проект «компьютеры Фибоначчи».

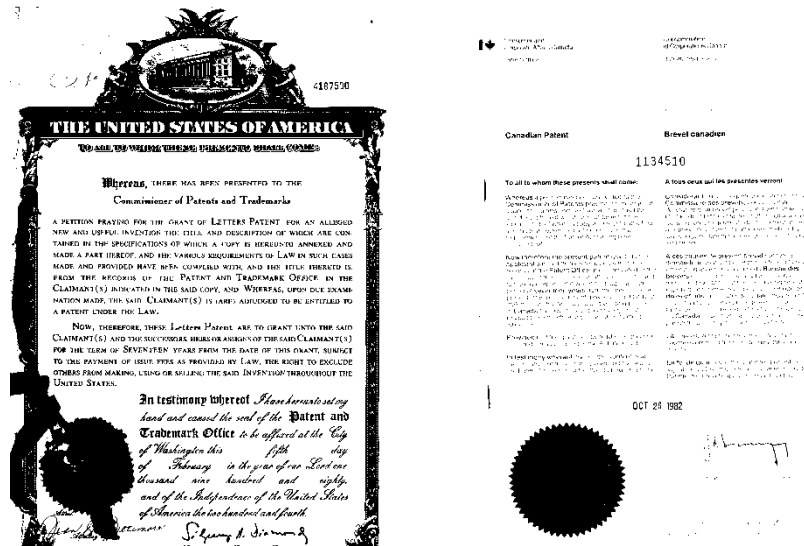


Рис. 10.1. Патенты США и Канады на изобретение "Устройство приведения р-кода Фибоначчи к минимальной форме"

В июне 1989 г. направление было обсуждено на специальном заседании Президиума Академии наук Украины. Инициатором этого обсуждения стал Президент АНУ Борис Евгеньевич Патон, который оказал огромную помощь в развитии этого направления.

К сожалению, так называемая «горбачевская перестройка» и последовавший за этим развал Советского Союза нанесли непоправимый удар по развитию многих научных направлений, который выполнялись для оборонных приложений, в том числе, и научному направлению по «компьютерам Фибоначчи». Начиная с 1989 г., финансирование этих разработок резко сократилось, а затем и полностью прекратилось. Научный коллектив, созданный в СКТБ «Модуль», распался, а само КБ было расформировано.

Но это не означает, что концепция «компьютеров Фибоначчи» устарела. Наоборот, на современном этапе развития компьютерной технологии эта

концепция стала еще более актуальной в новейшей области компьютерной техники – микропроцессорной технике. Именно это обстоятельство, а также участвовавшие аварии при запуске ракет, связанные со сбоями в бортовых микропроцессорах [118,119], стали побудительной причиной вновь возвратиться к этим идеям [68], впервые высказанным автором в 70-е годы 20 в.

В заключение необходимо отметить, что наряду с советскими разработками в области «арифметики Фибоначчи» и «компьютеров Фибоначчи» в этот же период подобные работы проводились в США (Университет штата Мериленд) [120-125]. Эти работы являются подтверждением того факта, что в 70-е годы 20 в. понятие «компьютеры Фибоначчи» широко вошло в американскую научно-техническую литературу.

#### **10.4. Особенности применения классических избыточных кодов в «компьютерных каналах»**

Как известно, классические избыточные коды [115] создавались с целью повышения надежности передачи информации в системах связи. Главной особенностью их применения в системах связи является ПОСЛЕДОВАТЕЛЬНЫЙ характер передачи сообщений. Поэтому центральной теоретической идеей в каналах связи стала модель «симметричного канала» [115], согласно которой наиболее вероятными являются ошибки малой кратности (одиночные или двойные). В теории корректирующих кодов [115] эта идея привела к введению понятия кодового расстояния, которое и стало на многие годы главным критерием для оценки эффективности корректирующих кодов.

Однако, предложение использовать классические избыточные коды (коды Хэмминга, циклические, турбо-коды и др.) для повышения надежности компьютеров и процессоров, в частности, нанокomпьютеров и нанопроцессоров, вызывает большие сомнения в силу следующих особенностей «вычислительных каналов» [62].

Основной особенностью «компьютерных каналов» по сравнению с «каналами связи» является ПАРАЛЛЕЛЬНЫЙ характер хранения, передачи и

обработки информации. Это приводит к следующим ограничениям на применение классических избыточных кодов в компьютерных структурах [62]:

1. Для компьютерных структур (счетчиков, регистров, сумматоров, памяти) модель ошибок, основанная на концепции «симметричного канала», принятой в теории связи [115], не соответствует реальным экспериментальным данным.

2. Экспериментальные данные показывают, что типичный вероятностный закон распределения ошибок в «компьютерных каналах» отличается от модели «симметричного канала» и приближается к равномерному распределению, то есть, все ошибки (одиночные, двойные, тройные и т.д.), возникающие в компьютерных структурах, обладают примерно одинаковой вероятностью. Это означает, что понятие «минимального кодового расстояния» применительно к кодам, используемым в «компьютерных каналах», теряет практический смысл в качестве критерия оценки эффективности кода. В «компьютерных каналах» на передний план выдвигается только один критерий – ошибкообнаруживающая способность кода, зависящая от количества избыточных разрядов.

3. При использовании классических избыточных кодов в «компьютерных каналах» мы вынуждены использовать два кода: исходный неизбыточный код (представление информации в двоичной системе) для выполнения арифметических операций и избыточный код для обеспечения контроля преобразований информации. Это создает так называемую проблему кодирования-декодирования. Ее суть состоит в том, что кодеры и декодеры для параллельных компьютерных структур являются довольно сложными с аппаратурной точки зрения. В большинстве случаев их сложность превышает сложность компьютерных структур, которые они контролируют. В качестве примера вспомним кодер и декодер для простейшего избыточного кода – кода с проверкой на четность.

## **10.5. Информационные основы микропроцессоров Фибоначчи**

**Основная идея.** На современном этапе развития информационной технологии человечество становится заложником классической двоичной системы

счисления, которая лежит в основе современных микропроцессоров и информационных технологий. Поэтому дальнейшее развитие микропроцессорной техники и основанной на ней информационной технологии на основе классической двоичной системы счисления применительно к специальным областям приложений следует признать неэффективным направлением. Двоичная система не может служить информационной и арифметической основой специализированных компьютерных и измерительных систем (космос, управление транспортом и сложными технологическими объектами), а также наноэлектронных систем, где проблемы надежности, помехоустойчивости, контролеспособности, стабильности, живучести систем выходят на передний план.

В таких системах целесообразно отказаться от классической двоичной системы счисления как информационной и арифметической основы специализированных компьютерных систем и наноэлектронных систем и перейти при их проектировании на новые избыточные системы счисления (коды Фибоначчи и золотой пропорции), сохраняющие все известные преимущества классической двоичной системы счисления (позиционность представления чисел, простота арифметических правил, использование двух цифр  $\{0,1\}$  для представления чисел, простые правила сравнения и округления чисел и др.) и позволяющие, за счет их «естественной» избыточности, совпадающей с избыточностью генетического кода, улучшить надежность, контролеспособность, помехоустойчивость компьютерных систем и тем самым повысить информационную надежность компьютеров.

Мы настолько свыклись со сбоями в нашем персональном компьютере, что на это давно перестали обращать внимание. В этом случае самый эффективный путь в восстановлении работоспособности компьютера состоит в его «перезагрузке». В подавляющем числе случаев компьютер после «перезагрузки» начинает работать нормально, что подтверждает тот факт, что причиной неисправной работы компьютера был именно «сбой», а не «отказ». Но ведь бортовой компьютер по существу состоит из тех же электронных компонентов и микропроцессоров, что и наш персональный компьютер. Но при этом работа бортового компьютера протекает в режиме реального времени и в условиях активного влияния агрессивной окружающей среды, что повышает вероятность

«сбоев». При этом «перезагрузка» бортового компьютера недопустима, так он должен обеспечивать функции управления в режиме реального времени.

Космическая техника – эта первая область, где человечество впервые реально столкнулось с этой проблемой [118,119]. Но если в случае «сбоя» персонального компьютера, цена от последствий этого «сбоя» весьма незначительна, то в случае «сбоя» бортового компьютера цена этого «сбоя» возрастает до уровня национальной катастрофы. Нельзя играть в «страусиные игры». Пора признать чрезвычайную актуальность этой проблемы и вынести ее на национальный уровень. Об этом прямо говорит академик Хетагуров, статья которого называется «Обеспечение национальной безопасности систем реального времени». И нужны новые и смелые оригинальные решения, позволяющие повысить информационную надежность микропроцессоров. Одним из таких решений является использование кодов Фибоначчи и кодов золотой пропорции при проектировании микропроцессоров.

**Коды, используемые в микропроцессоре Фибоначчи.** Арифметической и информационной основой микропроцессоров Фибоначчи являются два новых позиционных представления, рассмотренные выше:

1. Код Фибоначчи (КФ):

$$N = a_n F_n + a_{n-1} F_{n-1} + \dots + a_i F_i + \dots + a_1 F_1. \quad (10.1)$$

2. Код золотой пропорции (КЗП):

$$A = \sum_i a_i \Phi^i. \quad (10.2)$$

Существенно подчеркнуть, что коэффициенты  $a_i$  в КФ (10.1) и КЗП (10.2) принимают только два значения 0 и 1 и в этом отношении КФ и КЗП подобны классической двоичной системе.

**Сходство между КФ и КЗП.** Позиционные представления (10.1) и (10.2) близки по своей структуре и своим свойствам. Подчеркнем еще раз, что коды (10.1) и (10.2), подобно двоичной системе счисления являются двоичными кодами, то есть, используют для представления чисел двоичный алфавит  $\{0,1\}$ . То есть,

существенное преимущество двоичной системы (ее согласованность с двоичным характером электронных элементов и булевой логикой) в этих кодах сохраняется.

Общим между кодами (10.1) и (10.2) является наличие одной и той же математической связи между весами разрядов. В КФ (10.1) она задается рекуррентным соотношением Фибоначчи:

$$F_i = F_{i-1} + F_{i-2}; \quad F_1 = F_2 = 1, \quad (10.3)$$

в КЗП (10.2) – следующим «аддитивным» соотношением:

$$\Phi^i = \Phi^{i-1} + \Phi^{i-2}. \quad (10.4)$$

Так как, согласно (10.3) и (10.4), вес каждого разряда в кодах (10.1) и (10.2) равен сумме двух предыдущих, это означает, что к этим кодам применимы одни и те же микрооперации – «свертка» и «развертка» двоичных разрядов:

$$100 \square 011. \quad (10.5).$$

Это также означает, что цифровые записи чисел в этих кодах обладают двумя характерными крайними представлениями – минимальной формой, в которой двух единиц рядом не встречается, и максимальной (развернутой) формой, в которой двух нулей рядом не встречается, начиная со старшего значащего разряда (в сторону младших разрядов). Минимальная форма получается из представлений (10.1) и (10.2) путем применения микрооперации «свертка», в то время как максимальная форма – путем применения микрооперации «развертка».

Важным преимуществом кодов (10.1) и (10.2) является то, что, подобно двоичному коду, они являются позиционными системами счисления. При этом операция сравнения чисел в кодах (10.1) и (10.2) выполняется по тем же правилам, что и в двоичной системе, если коды (10.1) и (10.2) предварительно представить в минимальной форме.

Наконец,  $n$ -разрядные коды Фибоначчи и золотой пропорции обладают избыточностью, которая проявляет себя в свойстве многозначности представления одного и того же числа. И в этом отношении коды (10.1) и (10.2) подобны генетическому коду, в котором реализуется подобный же принцип кодирования кодонов (свойство вырожденности генетического кода). При этом численное значение относительной кодовой избыточности КФ (10.1) и КЗП (10.2) в пределе равно 0,44 (44%) и совпадает с избыточностью генетического кода.

Таким образом, будущие компьютеры, основанные на КФ (10.1) и КЗП (10.2), будут использовать те же принципы информационной избыточности, которые заложены природой в генетическом коде и проявляют себя в многочисленных созданиях живой природы (сосновые шишки, ананасы, кактусы, головки подсолнечников и др.) в виде «фибоначчиевых» спиралей на поверхности этих филлотаксисных объектов.

Следует отметить, что переход компьютерной техники на новые избыточные двоичные системы счисления означает новый этап развития тех видов специализированных компьютерных систем, для которых главным является обеспечение высокой информационной надежности выполняемых в компьютере или микропроцессоре операций.

**Различие между КФ и КЗП.** В то же время между КФ (10.1) и КЗП (10.2) имеются существенные различия.

Первое различие состоит в том, что КФ (10.1) предназначен для кодирования натуральных чисел, в то время как КЗП (10.2) – для кодирования действительных чисел.

Для КЗП (10.2) существует следующее «мультипликативное» соотношение между весами разрядов:

$$\Phi^i = \Phi \times \Phi^{i-1}. \quad (10.6)$$

В КФ (10.1) такое соотношение отсутствует.

Что касается «мультипликативного» соотношения (10.6), то КЗП (10.2) подобен классической двоичной системе счисления, в которой веса двоичных разрядов также связаны подобным «мультипликативным» соотношением:

$$2^i = 2 \times 2^{i-1}. \quad (10.7)$$

Важно подчеркнуть, что соотношение (10.6) лежит в основе ряда важных арифметических свойств КЗП, которые сближают КЗП с классической двоичной системой.

Первое из этих свойств – это возможность сдвига «золотой» цифровой записи (ЗЦЗ)

$$A = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-(m-1)} a_m \quad (10.8)$$

влево и вправо. При этом сдвиг ЗЦЗ на один разряд влево соответствует умножению числа  $A$ , представленного ЗЦЗ (10.8), на основание системы  $\Phi$ , то есть,

$$A \times \Phi = a_n a_{n-1} \dots a_1 a_0 a_{-1} \cdot a_{-2} \dots a_{-(m-1)} a_m, \quad (10.9)$$

в то время как сдвиг вправо соответствует делению на основание системы  $\Phi$ , то есть,

$$A \times \Phi^{-1} = a_n a_{n-1} \dots a_1 \cdot a_0 a_{-1} a_{-2} \dots a_{-(m-1)} a_m. \quad (10.10)$$

## 10.6. Преобразование чисел из традиционных систем счисления в код золотой пропорции

**Табличный метод.** Известны два метода преобразования чисел из одной системы счисления в другую [12]. Первый из них называется табличным методом, второй – аналитическим методом. Табличный метод основан на предварительном создании специальной таблицы ЗЦЗ натуральных и действительных чисел. Такая таблица хранится в постоянной памяти компьютера. Например, ЗЦЗ натуральных чисел  $N$  хранятся в памяти по адресам, которые являются двоичными представлениями этих натуральных чисел (Табл.10.1).

Table 10.1. Таблица «золотых» цифровых записей натуральных чисел

Адрес $N$	$\Phi^4$	$\Phi^3$	$\Phi^2$	$\Phi^1$	$\Phi^0$	$\Phi^{-1}$	$\Phi^{-2}$	$\Phi^{-3}$	$\Phi^{-4}$
0=0000	0	0	0	0	0.	0	0	0	0
1=0001	0	0	0	0	1.	0	0	0	0
2=0010	0	0	0	1	0.	0	1	0	0
3=0011	0	0	1	0	0.	0	1	0	0
4=0100	0	0	1	0	1.	0	1	0	0
5=0101	0	1	0	0	0.	1	0	0	1
6=0110	0	1	0	1	0.	0	0	0	1
7=0111	1	0	0	0	0.	0	0	0	1
8=1000	1	0	0	0	1.	0	0	0	1
9=1001	1	0	0	1	0.	0	1	0	1
10=1010	1	0	1	0	0.	0	1	0	1



**Аналитический метод преобразования правильных дробей.** Существо метода состоит в выполнении определенных арифметических операций над правильной дробью в исходной системе счисления для получения ЗЦЗ.

Предположим, что правильная дробь  $A$  в КЗП (10.2) представляется следующим образом:

$$A = a_{-1}\Phi^{-1} + a_{-2}\Phi^{-2} + a_{-3}\Phi^{-3} + \dots + a_{-n}\Phi^{-n} = 0.a_{-1}a_{-2}a_{-3}\dots a_{-n}. \quad (10.11)$$

Предположим также, что правильная дробь  $A$  представлена в минимальной форме. Тогда, умножая число (10.11) на основание  $\Phi$ , мы получим следующий результат:

$$A \times \Phi = a_{-1} + a_{-2}\Phi^{-1} + a_{-3}\Phi^{-2} + \dots + a_{-n}\Phi^{-n+1} = a_{-1}.a_{-2}a_{-3}\dots a_{-n}, \quad (10.12)$$

где  $a_{-1}$  – целая часть произведения (10.12) и сумма

$$A_1 = a_{-2}\Phi^{-1} + a_{-3}\Phi^{-2} + \dots + a_{-n}\Phi^{-n+1} \quad (10.13)$$

является дробной частью произведения (10.12).

Таким образом, из этих рассуждений вытекает, что после первого умножения исходной дроби (10.11) на основание  $\Phi$  целая часть произведения  $A \times \Phi$  является первой двоичной цифрой  $a_{-1}$  ЗЦЗ дробного числа (10.11).

После умножения дробного числа (10.13) на основание  $\Phi$ , мы получим следующий результат:

$$A_1 \times \Phi = a_{-2} + a_{-3}\Phi^{-1} + \dots + a_{-n}\Phi^{-n+2} = a_{-2}.a_{-3}a_{-4}\dots a_{-n}. \quad (10.14)$$

Анализ (10.14) показывает, что второе умножение приводит к получению второй двоичной цифры  $a_{-2}$  ЗЦЗ дробного числа (10.11).

Продолжая процесс умножения  $n$  раз, мы получим ЗЦЗ исходной дроби (10.11).

**Пример 10.1.** Преобразовать правильную десятичную дробь  $\frac{1}{2}$  в КЗП (10.2).

Решение:

Первое умножение:

$$\frac{1}{2} \times \Phi = \frac{1}{2} \times \frac{1 + \sqrt{5}}{2} = \frac{1 + \sqrt{5}}{4} = 0.809. \quad (10.14)$$

Так как целая часть произведения (10.14) равна 0, то из этого вытекает, что первая двоичная цифра ЗЦЗ десятичной дроби  $\frac{1}{2}$  равна:  $a_{-1} = 0$ .

Второе умножение:

$$\left(\frac{1}{2} \times \Phi\right) \times \Phi = \frac{1}{2} \times \Phi^2 = \frac{1}{2} \times \frac{3+\sqrt{5}}{2} = \frac{3+\sqrt{5}}{4} = 1.309 \quad (10.15)$$

Так как целая часть произведения (10.15) равна 1, то это означает, что  $a_{-2} = 1$ . Для выполнения следующего умножения необходимо вычесть 1 из числа (10.15):

$$\frac{3+\sqrt{5}}{4} - 1 = \frac{\sqrt{5}-1}{4} = \frac{1}{2} \times \Phi^{-1}.$$

Третье умножение:

$$\left(\frac{1}{2} \times \Phi^{-1}\right) \times \Phi = \frac{1}{2} = 0.5.$$

Мы получили правильную дробь  $\frac{1}{2}$  в результате третьего умножения. Это означает, что третья двоичная цифра ЗЦЗ исходной десятичной дроби  $\frac{1}{2}$  равна:  $a_{-3} = 0$ .

После третьего умножения мы пришли к дробному числу  $\frac{1}{2}$ . Из этого следует, что дальнейшее умножение приведет к повторению предыдущих результатов, то есть,  $a_{-4} = a_{-1} = 0$ ;  $a_{-5} = a_{-2} = 1$ ;  $a_{-6} = a_{-3} = 0$  и т.д. Следовательно, десятичная дробь  $\frac{1}{2}$  представляется в КЗП (10.2) как периодическая дробь:

$$. \quad 0.5 = 0.010010010\dots = \Phi^{-2} + \Phi^{-5} + \Phi^{-8} + \dots \quad (10.16)$$

Важно подчеркнуть, что рассмотренный выше алгоритм преобразования правильной дроби в КЗП (10.2) всегда приводит к ЗЦЗ, представленной в минимальной форме, то есть преобразование является контролируемым.

**Преобразование целых чисел.** Преобразование целых чисел в КЗП (10.2) основывается на следующей теореме.

**Теорема 10.1.** Существует единственное представление любого действительного числа  $A$  в виде:

$$A = \Phi^n + r, \quad (10.17)$$

где

$$0 \leq r < \Phi^{n-1}. \quad (10.18)$$

Для доказательства этой теоремы рассмотрим последовательность степеней золотой пропорции:

$$\left\{ \dots, \Phi^{n+2}, \boxed{\Phi^{n+1}, \Phi^n}, \Phi^{n-1}, \dots \right\}. \quad (10.19)$$

В последовательности (10.19) всегда может быть найдена единственная пара соседних степеней золотой пропорции, связанных с числом  $A$  соотношением:

$$\Phi^n \leq A < \Phi^{n+1}. \quad (10.20)$$

Вычитая из всех членов неравенства (10.20) число  $\Phi^n$ , получим:

$$0 \leq A - \Phi^n < \Phi^{n+1} - \Phi^n = \Phi^{n-1}. \quad (10.21)$$

Учитывая, что  $A - \Phi^n = r$ , из (10.21) вытекает неравенство (10.18).

Теорема доказана.

С использованием Теоремы 10.1 процесс преобразования целого числа  $N$  сводится к последовательному сравнению числа  $N$  и возникающих при этом остатков со степенями золотой пропорции (10.19), начиная со старшей степени.

**Пример 10.2.** Преобразовать целое число 4 в КЗП (10.2).

Решение:

Представим степени золотой пропорции  $\Phi^i$  ( $i = 0, \pm 1, \pm 2, \pm 3, \dots$ ) в явном виде (Табл.10.2).

Таблица 10.2. Степени золотой пропорции

$n$	3	2	1	0	-1	-2	-3
$\Phi^n$	$\frac{4+2\sqrt{5}}{2}$	$\frac{3+\sqrt{5}}{2}$	$\frac{1+\sqrt{5}}{2}$	$\Phi^0 = 0$	$\frac{-1+\sqrt{5}}{2}$	$\frac{3-\sqrt{5}}{2}$	$\frac{-4+2\sqrt{5}}{2}$
<i>Д.э.</i>	4.236	2.618	1.618	1	0.618	0.382	0.236

Здесь  $D.э.$  означает «десятичный эквивалент».

#### Первый шаг преобразования

Сравнивая целое число 4 со степенями золотой пропорции из Табл.10.2, мы находим первую пару  $\Phi^3 = (4 + 2\sqrt{5})/2 = 4.236$  и  $\Phi^2 = (3 + \sqrt{5})/2 = 2.618$ , которая связана с числом 4 следующим неравенством:

$$(3 + \sqrt{5})/2 = 2.618 \leq 4 < (4 + 2\sqrt{5})/2 = 4.236. \quad (10.22)$$

Из (10.22) вытекает, что первая значащая двоичная цифра ЗЦЗ числа 4 появляется во 2-м разряде, то есть,  $a_2 = 1$ .

#### Второй шаг преобразования

Представим число 4 следующим образом:

$$4 = \left[ (3 + \sqrt{5})/2 + r_1 \right]. \quad (10.23)$$

Из (10.23) мы можем вычислить остаток  $r_1$ :

$$r_1 = 4 - \left[ (3 + \sqrt{5})/2 \right] = (5 - \sqrt{5})/2 = 1.382. \quad (10.24)$$

#### Третий шаг преобразования

Сравнивая остаток (10.24) с последующими степенями золотой пропорции из Табл.10.2, мы можем найти следующую пару степеней золотой пропорции  $\Phi^1 = (1 + \sqrt{5})/2 = 1.618$  и  $\Phi^0 = 1$ , связанных с остатком  $r_1 = 1.382$  следующим неравенством:

$$1 \leq 1.382 < (1 + \sqrt{5})/2 = 1.618. \quad (10.25)$$

Это означает, что следующая значащая двоичная цифра ЗЦЗ числа 4 должна появиться в 0-м разряде, то есть,  $a_0 = 1$ .

#### Четвертый шаг преобразования

Представим остаток  $r_1$  следующим образом:

$$r_1 = (5 - \sqrt{5})/2 = \Phi^0 + r_2 = 1 + r_2, \quad (10.26)$$

где второй остаток  $r_2$  равен:

$$r_2 = r_1 - 1 = \left[ (5 - \sqrt{5})/2 \right] - 1 = (3 - \sqrt{5})/2 = 0.382. \quad (10.27)$$

#### Пятый шаг преобразования

Сравнивая остаток (10.27) со степенями золотой пропорции из Табл.10.2, мы можем найти следующую пару чисел,  $\Phi^{-1} = (-1 + \sqrt{5})/2 = 0.618$  и  $\Phi^{-2} = (3 - \sqrt{5})/2 = 0.382$ , связанных с остатком  $r_2 = 0.382$  следующим неравенством:

$$(3 - \sqrt{5})/2 = 0.382 \leq r_2 = 0.382 < (-1 + \sqrt{5})/2 = 0.618 \quad (10.28).$$

Это означает, что следующая значащая двоичная цифра ЗЦЗ числа 4 должна появиться в  $(-2)$ -м разряде, то есть,  $a_{-2} = 1$ .

#### Шестой шаг преобразования

Представим остаток  $r_2 = (3 - \sqrt{5})/2 = 0.382$  следующим образом:

$$r_2 = (3 - \sqrt{5})/2 + r_3, \quad (10.29)$$

где

$$r_3 = r_2 - \left[ (3 - \sqrt{5})/2 \right] = 0. \quad (10.30)$$

Так как остаток  $r_3 = 0$ , это означает, что процесс преобразования закончился, и результат преобразования выглядит следующим образом:

$$4 = 101.01 = \Phi^2 + \Phi^0 + \Phi^{-2}. \quad (10.31)$$

Существенно подчеркнуть, что рассмотренный выше алгоритм преобразования любого натурального числа  $N$  в КЗП (10.2) всегда приводит к представлению числа в минимальной форме, то есть, процесс преобразования является контролируемым. Но существует еще один способ контроля результата преобразования, основанный на  $Z$ -свойстве. Действительно, если в выражении

(10.31) заменить все степени золотой пропорции на соответствующие числа Фибоначчи, то мы получим сумму:

$$F_2 + F_0 + F_{-2} = 1 + 0 + (-1) = 0,$$

что свидетельствует о том, что число 4 является целым числом.

**Преобразование смешанных дробей в код золотой пропорции.** Если исходное число  $A$  является смешанной дробью, то есть, действительным числом, состоящим из целой и дробной части, то процесс преобразования осуществляется отдельно для целой и дробной части, а затем результаты преобразования складываются.

**Пример 10.3.** Преобразовать смешанную дробь 4.5 в код золотой пропорции (10.2).

Решение:

Для этого разделим смешанную дробь 4.5 на целую часть 4 и дробную часть 0.5. В примерах 10.1 и 10.2 получены следующие ЗЦЗ чисел 0.5 и 4, задаваемые (10.16) и (10.31). Для получения ЗЦЗ числа 4.5 необходимо сложить ЗЦЗ чисел 0.5 и 4, задаваемые (10.16) и (10.31):

$$\begin{array}{r} 4 = 101.01 \\ 0.5 = 0.010010010\dots \\ \hline 4.5 = 101.100110010\dots \end{array}$$

Результат сложения приводится к минимальной форме:

$$4.5 = 01\boxed{01.1}0\boxed{011}0010\dots = 0110.001000010\dots$$

Заметим, что рассмотренные выше примеры 10.1-10.3 являются основой для разработки машинных алгоритмов преобразования чисел в КЗП (10.2).

## 10.7. «Золотая» арифметика

**Представление чисел с плавающей запятой.** «Мультипликативное» соотношение (10.6) лежит в основе еще одного важного свойства КЗП (10.2) –

представления чисел с плавающей запятой, что сближает КЗП (10.2) с классической двоичной системой.

Рассмотрим следующую «золотую» дробь:

$$A = 0.000010101001 = \Phi^{-5} + \Phi^{-7} + \Phi^{-9} + \Phi^{-12}. \quad (10.32)$$

Используя «мультипликативное» свойство (10.6), мы можем представить «золотую» дробь в виде:

$$A = (0.10101001) \times \Phi^{-4} = (\Phi^{-1} + \Phi^{-3} + \Phi^{-5} + \Phi^{-8}) \times \Phi^{-4}. \quad (10.33)$$

Это и есть представление «золотой» дроби (10.32) в форме с плавающей запятой. Из (10.33) вытекает, что мантисса и экспонента «золотой» дроби (10.32), соответственно, равны:  $M(A) = 0.10101001$  и  $e(A) = -4$ .

Возможны и другие варианты представления «золотой» дроби (10.32) в форме с плавающей запятой. Например,

$$A = (10101001) \times \Phi^{-12} = (\Phi^7 + \Phi^5 + \Phi^3 + \Phi^0) \times \Phi^{-12}. \quad (10.34)$$

Из (10.34) вытекает:  $M(A) = 10101001$  и  $e(A) = -12$ .

**«Золотое» сложение и вычитание.** Арифметические операции сложения и вычитания в кодах (10.1) и (10.2) выполняются с использованием введенных выше базовых микроопераций:

Свертка:  $\boxed{011 \uparrow 100}$  Развертка:  $\boxed{100 \rightarrow 011}$

Перемещение:  $\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \rightarrow & = \\ \hline 0 & 1 \\ \hline \end{array}$  Поглощение:  $\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \updownarrow & = \\ \hline 1 & 0 \\ \hline \end{array}$

**Пример 10.4.** Произвести сложение чисел  $5+4$  в КЗП (10.2):

1. Представление чисел 5 и 4 в КЗП имеет вид (Табл.10.1):

$$5 = 1000.101 \text{ и } 4 = 101.01.$$

2. Выполним микрооперации «перемещения» над ЗЦЗ чисел 5 и 4:

$$\begin{array}{r}
 5 = 1000.1001 \\
 + \quad \downarrow \quad \downarrow \quad \downarrow \\
 4 = 0101.0100 \\
 \hline
 5 + 4 = 1101.1101
 \end{array}$$

3. Приведем результат сложения к минимальной форме:

$$5 + 4 = \boxed{011}\boxed{01.1}101 = 10010.0101 = \Phi^4 + \Phi^1 + \Phi^{-2} + \Phi^{-4}. \quad (10.35)$$

Поскольку исходные числа 5 и 4 являются целыми числами, то результат их сложения должен быть целым числом. Тогда, используя  $Z$ -свойство, мы можем проверить правильность результата сложения  $5+4=9$  (10.35):

$$F_4 + F_1 + F_{-2} + F_{-4} = 3 + 1 + (-1) + (-4) = 0.$$

**Пример 10.5.** Вычесть число 10 из числа 3 в КЗП.

Решение:

1. Представим числа 3 и 10 в КЗП (Табл.10.1):

$$3 = 100.01 \text{ и } 10 = 10100.0101.$$

2 Выполним микрооперацию «поглощения»:

$$\begin{array}{r}
 3 = 00100.0100 \\
 - \quad \quad \uparrow \quad \uparrow \\
 10 = 10100.0101 \\
 \hline
 D = 10000.0001
 \end{array}$$

Так как результат вычитания  $D = 3 - 10$  находится в нижнем регистре, это означает, что результат вычитания  $D$  имеет знак «минус», то есть,

$$3 - 10 = -10000.0001.$$

**«Золотое» умножение.** Из «мультипликативного» соотношения (10.6) вытекает следующее тождество:

$$\Phi^{n+m} = \Phi^n \times \Phi^m, \quad (10.36)$$

которое задает правило «золотого» умножения (Табл.10.3).



Таблица 10.3. Правило «золотого» умножения

0×0	=	0
0×1	=	0
1×0	=	0
1×1	=	1

Ясно, что правило «золотого» умножения по своей форме полностью совпадает с аналогичным правилом в классической двоичной системе счисления.

**Пример 10.6.** Произведем «золотое» умножение правильных дробей:

$$A = 0.010010 \text{ и } B = 0.0001010.$$

Умножение осуществляется в следующем порядке:

1. Перед умножением сомножители представляются в форме с плавающей запятой:

$$A = 010010 \times \Phi^{-6}; B = 001010 \times \Phi^{-6}.$$

Это означает, что мантиссы и порядки сомножителей  $A$  и  $B$  равны, соответственно:

$$m(A) = 010010; e(A) = -6 \text{ и } m(B) = 001010; e(B) = -6.$$

2. Далее осуществляется умножение мантисс:

$$\begin{array}{r}
 \phantom{0} 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\
 \phantom{0} 0 \ 0 \ 1 \ 0 \ 1 \ 0 \\
 \hline
 \phantom{0} 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \phantom{0} 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\
 \phantom{0} 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \phantom{0} 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\
 \phantom{0} 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \phantom{0} 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \hline
 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

Заметим, что вначале, согласно Табл.10.3, осуществляется формирование частных произведений, которые затем складываются, согласно правилам фибоначчиевой арифметики. После сложения порядков  $e(A) + e(B) = -12$  и приведения результата умножения мантисс к минимальной форме получим результат умножения в форме с плавающей запятой:

$$A \times B = 00100000100 \times \Phi^{-12}.$$

**«Золотое» деление.** Чтобы сформулировать правило «золотого» деления, мы можем использовать аналогию между классической двоичной арифметикой и «золотой» арифметикой. Как известно, классическое двоичное деление сводится к операциям сдвига, сравнения чисел и вычитания. Можно показать, что эти же операции лежат в основе «золотого» деления. Так как «золотое» сравнение осуществляется над числами, представленными в минимальной форме, отсюда следует, что на каждом этапе деления все сравниваемые ЗЦЗ чисел должны быть приведены к минимальной форме.

**Пример 10.7.** Разделить число  $9 = 10010.0101 = \Phi^4 + \Phi^1 + \Phi^{-2} + \Phi^{-4}$  на число  $3 = 100.01 = \Phi^2 + \Phi^{-2}$ .

«Золотое» деление осуществляется в следующем порядке:

1. Представление чисел в форме с плавающей запятой:

$$A = 9 \times \Phi^{-4} = 100100101 \quad (10.37)$$

$$B = 3 \times \Phi^{-4} = 1000100. \quad (10.38)$$

2. Первый шаг деления состоит в выделении в ЗЦЗ делимого (10.37) минимально возможной группы разрядов  $A_1$  (со стороны старших разрядов), превышающей численно делитель  $B$  или равной ему:

$$A = \underbrace{1001001}_{A_1}01.$$

Это осуществляется путем сдвига делителя  $B$  на два разряда влево (в сторону старших разрядов) и поразрядного сравнения с делимым  $A$ .

Так как выделение такой группы разрядов в данном случае возможно, то во 2-й разряд частного  $Q$  записывается двоичная цифра 1, что соответствует сдвигу делителя на 2 разряда влево, то есть, тем самым мы определили значение старшего значащего разряда частного  $a_2 = 1$ .

3. После этого осуществляется вычитание из ЗЦЗ числа  $A_1 = 1001001$  делителя  $B = 1000100$ :

$$\begin{array}{r} A_1 = 1001001 \\ - B = 1000100 \\ \hline A_2 = 0000011 \end{array}$$

Полученный промежуточный результат  $A_2 = 011$  приводится к минимальной форме:

$$A_2 = \boxed{011} = 100. \quad (10.39)$$

4. После этого к младшему разряду числа (10.39) приписывается следующая цифра делимого:

$$A_3 = 1000.$$

Число  $A_3$  сравнивается с делителем  $B = 1000100$ . Поскольку  $A_3 < B$ , то следующая двоичная цифра частного  $Q$  равна:  $a_1 = 0$ .

5. К младшему разряду числа (10.39) приписывается следующая (последняя) цифра делимого

$$A_4 = 10001 \quad (10.40)$$

и осуществляется сравнение числа  $A_4$  с делителем  $B = 1000100$ . Поскольку  $A_4 < B$ , то следующая двоичная цифра частного  $a_0 = 0$ .

Так в делимом использованы все цифры, то на этом завершается определение цифр «целой» части частного.

6. На следующем шаге начинается определение «дробной» части частного  $Q$ . Для этого к младшему разряду числа (10.40) приписывается двоичная цифра 0

$$A_5 = 100010 \quad (10.41)$$

и число  $A_5$  сравнивается с делителем  $B = 1000100$ . Поскольку  $A_5 < B$ , то следующая двоичная цифра частного (первая после запятой)  $a_{-1} = 0$ .

7. На следующем шаге к младшему разряду числа (10.41) приписывается двоичная цифра 0

$$A_6 = 1000100 \quad (10.42)$$

и число  $A_6$  сравнивается с делителем  $B = 1000100$ . Так как  $A_6 = B$ , то в следующий разряд частного записывается двоичная цифра 1 ( $a_{-2} = 1$ ) и делитель  $B$  вычитается

из числа  $A_6$ . Поскольку последний остаток  $A_7 = 0$ , то на этом процесс деления заканчивается.

Частное  $Q$  имеет вид:  $Q = 100.01 = \Phi^2 + \Phi^{-2}$ .

**Пример 10.8.** Разделить число  $5 = 1000.1001 = \Phi^3 + \Phi^{-1} + \Phi^{-3}$  (делимое) на число  $10 = 10100.0101 = \Phi^4 + \Phi^2 + \Phi^{-2} + \Phi^{-4}$  (делитель) в КЗП (10.2).

«Золотое» деление осуществляется в следующем порядке:

1. Представление делимого  $5$  и делителя  $10$  в форме с плавающей запятой:

$$A = 5 \times \Phi^{-4} = 10001001 \quad (10.43)$$

$$B = 10 \times \Phi^{-4} = 101000101. \quad (10.44)$$

2. Сравнивая числа  $A$  и  $B$ , обнаруживаем, что  $A < B$ . Это означает, что результат деления является правильной дробью, то есть, двоичная цифра 0-го разряда частного  $Q$  равна:  $a_0 = 0$ .

3. Путем приписывания двоичной цифры 0 к младшему разряду числа  $A$  образуем новое число:

$$A_1 = 100010010. \quad (10.45)$$

Сравнивая число  $A_1$  с делителем  $B = 101000101$ , получаем:  $A_1 < B$ . Это означает, что первый (после запятой) разряд частного равен:  $a_{-1} = 0$ .

4. Приписывая двоичную цифру 0 к младшему разряду числа (10.45), образуем новое число:

$$A_2 = 1000100100. \quad (10.46)$$

Так как  $A_2 > B$ , это означает, что следующая цифра частного:  $a_{-2} = 1$ .

В этом случае мы должны вычесть делитель  $B = 101000101$  из числа  $A_2 = 1000100100$ . Для этого необходимо осуществить «развертку» ЗЦЗ числа  $A_2$ :

$$A_2 = \boxed{100}0\boxed{100}\boxed{100} = 01\boxed{100}11011 = 0101111011.$$

Сопоставляя число  $A_2$  с делителем  $B = 101000101$  и выполняя «поглощения», получим:

$$\begin{array}{r}
 A_2 = 0101111011 \\
 \quad \quad \quad \updownarrow \updownarrow \quad \updownarrow \\
 B = \underline{101000101} \\
 A_3 = 0000111010 \\
 B_1 = 000000100
 \end{array}$$

Осуществляя развертку числа  $A_3$ , получим:

$$A_3 = 1110\boxed{10.0} = 11\boxed{100}1.1 = 110111.1.$$

Сопоставляя число  $A_3$  с числом  $B_1 = 100$  и выполняя «поглощения», получим результат вычитания  $A_4$ :

$$\begin{array}{r}
 A_3 = 110111.1 \\
 \quad \quad \quad \updownarrow \\
 B = \underline{100} \\
 A_4 = 110011.1 \\
 B_2 = 000000.0
 \end{array}$$

Приведем результат вычитания  $A_4$  к минимальной форме:

$$A_4 = \boxed{011}0\boxed{011}.1 = 1000100.1. \quad (10.47)$$

Особенность промежуточного результата  $A_4$  состоит в том, что ЗЦЗ этого числа содержит 1 в дробной части.

Сравнивая число  $A_4 = 1000100.1$  с делителем  $B = 101000101$ , получаем следующий результат:  $A_4 < B$ . Это означает, что следующая цифра частного:  $a_{-3} = 0$ .

Таким образом, после выполнения указанных операций мы установили значения первых трех двоичных разрядов ЗЦЗ частного:

$$Q = 0.010\dots \quad (10.48)$$

5. Поскольку результат вычитания  $A_4$  не равен нулю, то «золотое» деление может быть продолжено. Для этого сдвинем ЗЦЗ числа  $A_4 = 1000100.1$  на один разряд влево:

$$A_5 = 10001001. \quad (10.49)$$

Сравнивая число  $A_5 = 10001001$  с делимым  $A = 10001001$ , мы приходим к заключению, что они совпадают, то есть,  $A_5 = A$ . Это значит, что, начиная с этого шага, процесс будет периодически повторяться, то есть, следующие двоичные цифры ЗЦЗ частного будут равны соответственно:  $a_{-4} = a_{-1} = 0; a_{-5} = a_{-2} = 1; a_{-6} = a_{-3} = 0$  и так далее.

Из этих рассуждений вытекает, что частное от «золотого» деления чисел  $5:10$  может быть представлено в виде следующей «золотой» цифровой записи:

$$Q = 0.010010010\dots = \Phi^{-2} + \Phi^{-5} + \Phi^{-8} + \dots,$$

то есть, для этого случая «золотая» цифровая запись частного  $Q = 5:10 = 1:2$  представляет собой периодическую дробь.

## 10.8. Сравнение КФ и КЗП с классической двоичной системой

**Ограничение разрядности и избыточность КЗП.** Ясно, что в микропроцессоре Фибоначчи разрядность КЗП ограничена некоторым числом  $n$ . И это приводит к некоторым особенностям КЗП (10.2). Рассмотрим КЗП (10.1) для случая ограниченной разрядности:

$$A = a_m \Phi^m + a_{m-1} \Phi^{m-1} + \dots + a_1 \Phi^1 + a_0 \Phi^0 + a_{-1} \Phi^{-1} + a_{-2} \Phi^{-2} + \dots + a_{-k} \Phi^{-k}. \quad (10.50)$$

Коду золотой пропорции (10.50) соответствует следующая ЗЦЗ с ограниченной разрядностью:

$$A = a_m a_{m-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-k}. \quad (10.51)$$

Такая ЗЦЗ состоит из  $n = m + k + 1$  двоичных разрядов, включая  $m + 1$  разрядов с неотрицательными индексами  $(0, 1, 2, 3, \dots, m)$  и  $k$  разрядов с отрицательными индексами  $(-1, -2, -3, \dots, -k)$ .

В качестве примера рассмотрим 4-разрядный КЗП (10.50), который может быть использован для «золотого» представления правильных дробей. ЗЦЗ (10.51) такого КЗП может быть представлена в следующем виде:

$$A = a_{-1} a_{-2} a_{-3} a_{-4}. \quad (10.52)$$

Предположим, что мы используем «минимальную форму» в (10.52). Тогда мы можем представить отображение «золотых» цифровых записей (минимальных форм) (10.52) на некоторое множество действительных чисел, как показано в Табл.10.4.

Таблица 10.4. Отображение ЗЦЗ на множество действительных чисел

$i$	$a_{-1}$	$a_{-1}$	$a_{-1}$	$a_{-1}$	$A_i$	$\Delta_i = A_i - A_{i-1}$
0	0	0	0	0	0	–
1	0	0	0	1	$\Phi^{-4}$	$\Phi^{-4}$
2	0	0	1	0	$\Phi^{-3}$	$\Phi^{-5}$
3	0	1	0	0	$\Phi^{-2}$	$\Phi^{-4}$
4	0	1	0	1	$\Phi^{-2} + \Phi^{-4}$	$\Phi^{-4}$
5	1	0	0	0	$\Phi^{-1}$	$\Phi^{-5}$
6	1	0	0	1	$\Phi^{-1} + \Phi^{-4}$	$\Phi^{-4}$
7	1	0	1	0	$\Phi^{-1} + \Phi^{-3}$	$\Phi^{-5}$

Анализ Табл.10.4 приводит нас к интересным выводам, касающимся КЗП с ограниченной разрядностью:

1. КЗП с ограниченной разрядностью, в отличие от классического двоичного кода, обеспечивают неравномерное квантование действительных чисел. При этом шаг квантования принимает два значения, равные либо весу младшего разряда  $\Phi^{-4}$  либо весу следующего за ним разряда  $\Phi^{-5}$  (для примера (10.52)). Для общего случая (10.50) шаг квантования также принимает два значения:  $\Phi^{-k}$  или  $\Phi^{-k-1}$ .
2. Количество действительных чисел  $A_i$ , которые могут быть представлены с помощью 4-разрядного КЗП (10.52) в минимальной форме совпадает с количеством минимальных форм 4-разрядного КЗП. Для примера (10.52) оно равно 8, то есть, числу Фибоначчи  $F_6 = 8$ . Для  $n$ -разрядного КЗП (10.50) количество действительных чисел, которые могут быть представлены в минимальной форме, равно числу Фибоначчи  $F_{n+2}$ .

Полученные результаты позволяют определить предельное значение относительной избыточности КЗП (10.50), которая совпадает по величине с относительной избыточностью кода Фибоначчи и равна 0.44 (44%).

**Сравнение систем счисления.** Арифметические и схемотехнические основы «компьютеров Фибоначчи» изложены в книгах [9,12]. В этих работах показано, что в КФ (10.1) и КЗП (10.2) можно выполнять все логические и арифметические операции, необходимые для реализации микропроцессора Фибоначчи. Таблица 10.5 дает представление о преимуществах КФ и КЗП в сравнении с классическим двоичным кодом.

Таблица 10.5. Сравнение КФ и КЗП с классическим двоичным кодом

Тип кода	Двоичный код	← КЗП ⇒	КФ
Формула	$A = \sum_i a_i 2^i$	$\Leftarrow A = \sum_i a_i \Phi^i \Rightarrow$	$N = \sum_{i=1}^n a_i F_i$
Связь между разрядами: <i>мультипликат. аддитивная</i>	$2^i = 2 \times 2^{i-1}$ $2^i = 2^{i-1} + 2^{i-1}$	$\Leftarrow \Phi^i = \Phi \times \Phi^{i-1}$ $\Phi^i = \Phi^{i-1} + \Phi^{i-2} \rightarrow$	$F_i = F_{i-1} + F_{i-2}$
Сдвиг кода	+	+	-
Представление с плавающей запятой	+	+	-
Округление чисел	+	+	-
Избыточность	0	0.44	0.44

Из Табл.10.5 вытекает, что КЗП сохраняет все качественные характеристики классического двоичного кода: двоичный характер представления информации, простота правил выполнения арифметических операций, сдвиг кода, представление с плавающей запятой и др. При этом КЗП обладает такой же избыточностью, как и КФ (44%), сохраняет все свойства КФ (свертка и развертка двоичных разрядов, приведение кода к минимальной и максимальной форме и др.) и обладает свойством контроля всех преобразований информации в микропроцессоре.



## 10.9. Контроль электронной памяти и регистров

**Электронная память и регистры.** Выше было показано, что относительная кодовая избыточность КФ (10.1) и КЗП (10.2) не зависит от длины разрядной сетки и в пределе равна 0,44 (44%). Это означает, что длина разрядной сетки КФ и КЗП по сравнению с классическим двоичным кодом должна быть увеличена примерно в 1,5 раза. Это и есть «плата» за возможность непрерывного оперативного контроля всех устройств и информационных преобразований в микропроцессоре.

Микропроцессор содержит различные по своему функциональному назначению устройства: регистры, электронная память, счетчики, таймеры, системы связи, арифметико-логические устройства (АЛУ), наконец, аналого-цифровые и цифроаналоговые преобразователи (в сигнальных микропроцессорах) и др. Эти устройства реализуют различные преобразования информации в микропроцессоре: передача, запись, хранение и считывание информации, счет и вычитание единиц, выполнение логических и арифметических операций, аналого-цифровое и цифроаналоговое преобразование.

Электронная память – это устройство для хранения, записи и считывания данных. Данные хранятся в ячейках памяти. Запись и считывание данных, хранящихся в ячейках, осуществляется с помощью адресов, которые представляют собой  $n$ -разрядные двоичные коды, которые поступают на дешифратор адреса. Для записи или считывания данных в конкретную ячейку памяти используется всего один вывод дешифратора. Когда на шине адреса установится нужный адрес ячейки памяти, то, хотя сигнал записи будет подан на все ячейки, запись произойдет только в ту ячейку, которая будет в данный момент выбрана (адресована). Примерно так же осуществляется считывание данных из ячейки памяти.

Регистр - это разновидность электронной памяти. Регистр представляет собой логическое устройство, используемое для хранения  $n$  – разрядных данных и выполнения преобразований над ними; по существу регистр процессора - это сверхбыстрая память внутри процессора, предназначенная для хранения адресов и промежуточных результатов вычислений (регистр общего назначения/регистр данных) или данных, необходимых для работы самого процессора.

Электронная память применяется практически во всех подсистемах РС, выступая в качестве оперативной памяти, кэш-памяти, постоянной памяти, полупостоянной памяти, буферной памяти, внешней памяти.

Основная, или оперативная, память (main memory) компьютера используется для оперативного обмена информацией (командами и данными) между процессором, внешней памятью (например, дисковой) и периферийными подсистемами (графика, ввод-вывод, коммуникации и т. п.). Ее другое название - ОЗУ (оперативное запоминающее устройство) - примерно соответствует английскому термину RAM (Random Access Memory — память с произвольным доступом). Произвольность доступа подразумевает возможность операций записи в любую ячейку ОЗУ или чтения любой ячейки ОЗУ в произвольном порядке.

**Контроль электронной памяти, основанный на свойстве минимальной формы.** Такие данные, как адреса и команды в электронной памяти или регистре микропроцессора Фибоначчи кодируются в КФ, числовая информация кодируется либо в КФ (натуральные числа), либо в КЗП (действительные числа). При этом для контроля используется две формы представления информации – минимальная или максимальная формы. Контроль кодов данных осуществляется путем проверки их соответствия признакам минимальной или максимальной формы.

Как упоминалось, минимальная форма имеет два контрольных признака:

- 1) две единицы рядом в коде не встречаются;
- 2) младший разряд кода в минимальной форме всегда равен 0.

Максимальная форма также имеет два контрольных признака:

- 1) начиная со старшего значащего разряда (в сторону младших разрядов) два нуля рядом в коде не встречаются;
- 2) младший разряд кода всегда равен 1.

На Рис.10.2 изображена комбинационная логическая схема для контроля минимальной формы. Схема состоит из  $n$  двухвходовых логических элементов И. На их входы поступают данные, представленные в минимальной форме. Выходы элементов И соединены со входами многовходового элемента ИЛИ. Если исходная кодовая комбинация имеет две соседние двоичные единицы или двоичную 1 в

младшем разряде, тогда логическая 1 возникает на выходе хотя бы одного элемента И. Это приводит к возникновению логической 1 на выходе элемента ИЛИ (check output), что и является сигналом ошибки.

Схема на Рис.10.2 гарантирует обнаружение всех ошибок, которые приводят к искажению минимальной формы (minimal form) исходного кода. Если в минимальной форме произошла ошибка, связанная с переходом исходной минимальной формы в другую минимальную форму, то этот случай относится к разряду необнаруживаемых ошибок.

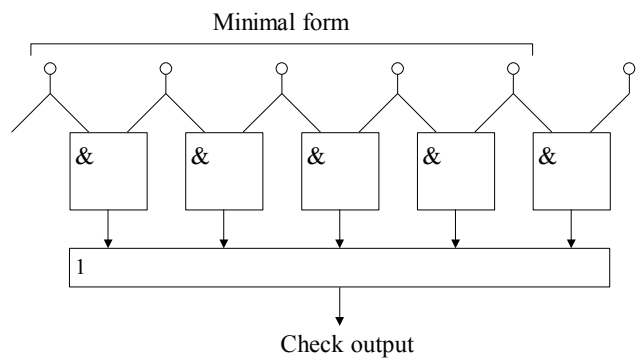


Рис. 10.2. Комбинационная логическая схема для контроля минимальной формы

### Ошибкообнаруживающая способность минимальной формы КФ.

Эффективность способа контроля данных по признаку минимальной формы оценивается с помощью коэффициента обнаружения ошибок [115]. Если, например, для представления числовой информации используется минимальная форма, то коэффициент обнаружения ошибок может быть вычислен на основе следующих соображений.

Как показано в главе 8, с помощью  $n$ -разрядного КФ (10.1) в минимальной форме может быть представлено  $F_{n+1}$  целых чисел в диапазоне от 0 до  $F_{n+1} - 1$ , где  $F_{n+1} - (n + 1)$  – е число Фибоначчи.

Заметим, что минимальные формы являются «разрешенными» кодовыми комбинациями для рассматриваемого способа кодирования. Обозначим их количество через  $N_p$ . Ясно, что  $N_p = F_{n+1}$ . Число всех возможных  $n$ -разрядных

двоичных кодовых комбинаций (КК) равно  $N_0 = 2^n$ . Таким образом, все  $n$ -разрядные двоичные КК разбиваются на две непересекающиеся группы: «разрешенные» КК (минимальные формы), число которых равно  $N_p = F_{n+1}$ , и «запрещенные» КК, число которых равно  $N_0 - N_p$ .

Под воздействием «помех» каждая из  $N_p$  «разрешенных» КК может перейти в одну из  $N_0 = 2^n$  КК. При этом число всех возможных переходов  $\{N_p\} \rightarrow \{N_0\}$  равно  $N_p \times N_0$ . Из всех возможных переходов только один переход является «правильным», когда минимальная форма переходит сама в себя, то есть, число «правильных» переходов равно  $N_p = F_{n+1}$ . Все остальные переходы являются «ошибочными». При этом ошибка обнаруживается в  $N_p \times (N_0 - N_p)$  случаях из  $N_p \times N_0$  возможных случаев.

Тогда доля обнаруживаемых переходов

$$S_{обн} = \frac{N_p \times (N_0 - N_p)}{N_p \times N_0} = 1 - \frac{N_p}{N_0} \quad (10.53)$$

и характеризует ошибкообнаруживающую способность КФ (10.1) при использовании минимальной формы, то есть, коэффициент обнаружения ошибок равен:

$$S_{обн} = 1 - \frac{F_{n+1}}{2^n}. \quad (10.54)$$

Представление о диапазоне изображаемых чисел  $F_{n+1}$  и коэффициенте обнаружения ошибок для КФ различной разрядности  $n$  дает Табл.10.6.

Таблица 10.6. Коэффициент обнаружения ошибок

$n$	$F_{n+1}$	$S_{обн}$ %
10	89	91.32
20	10946	98.95
30	1346269	99.87

Как следует из таблицы, ошибкообнаруживающая способность 20- и 30-разрядного КФ соответственно равна около 99% и 99,9%.

Интересное решение при кодировании команд программы, выполняемой микропроцессором Фибоначчи, приведено в [18]. Все данные предлагается представлять в минимальной форме КФ (10.1), что само по себе позволяет контролировать эти данные. Однако, при записи данных в ячейки памяти с четными адресами используется минимальная форма представления данных в КФ, а в случае записи данных в ячейки памяти с четными адресами они преобразовываются в максимальную форму. Это обеспечивает довольно эффективный контроль данных при их считывании из памяти, так как данные, считываемые из ячеек с четными адресами, должны быть представлены в минимальной форме, а из ячеек с нечетными адресами – в максимальной форме. Нарушение этого условия свидетельствует об ошибке в адресной части (данные считаны не с той ячейки). После считывания данных из ячеек памяти с нечетными адресами они приводятся к минимальной форме с помощью устройства приведения КФ к минимальной форме.

## **10.10. Устройство для приведения КФ и КЗП к минимальной форме**

**Регистр «свертки».** Как вытекает из анализа фибоначчиевой арифметики (см. главу 8), микрооперации свертки, развертки и приведения КФ к минимальной форме являются основными специфическими операциями фибоначчиевой арифметики. Поскольку в КФ (10.1) и КЗП (10.2) для представления чисел используется двоичный алфавит  $\{0,1\}$ , то в качестве основы для устройства свертки может быть использован классический двоичный регистр, состоящий из двоичных триггеров. Этот регистр имеет специальные логические схемы для выполнения операций свертки. Каждая разряд регистра содержит Т-триггер и логические элементы. Операция «свертки» ( $011 \rightarrow 100$ ) может быть представлена

как инверсия состояний соответствующих триггеров. Инверсия состояния триггера выполняется очень легко для триггера с "установочным" и "счетным" входами.

Один из возможных вариантов регистра свертки показан на Рис.10.3.

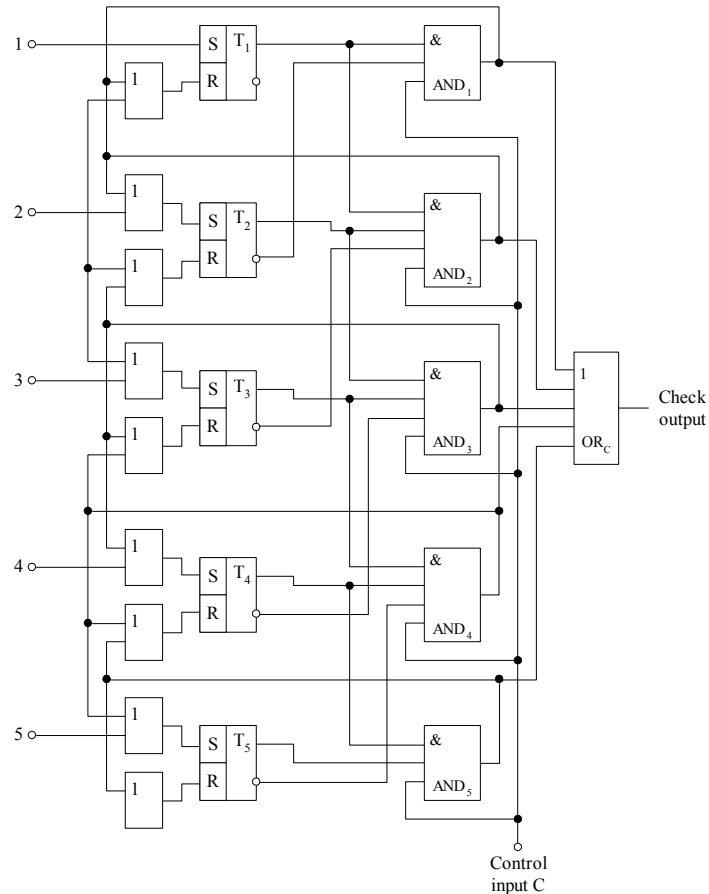


Рис. 10.3. «Фибоначчиевый» регистр «свертки»

Существенно отметить, что именно устройство на Рис.10.3 признано в СССР пионерным изобретением [69]. И это же устройство взято за основу при патентовании заявки «Устройство для приведения  $p$ -кода Фибоначчи» [70-83] во всех странах патентования (США, Япония, Англия, Франция, Канада, ФРГ, ГДР, Польша) и это устройство также зарегистрировано в этих странах как пионерное изобретение.

Регистр «свертки» состоит из пяти  $(R-S)$ - триггеров и логических элементов  $AND, OR(И, ИЛИ)$ , которые предназначены для реализации «свертки». Операция «свертки» выполняется, начиная с младшего триггера  $T_1$  к

старшему триггеру  $T_5$  с помощью логических элементов  $AND_1 - AND_5$  и соответствующих логических элементов  $OR(ИЛИ)$ , расположенных перед  $R$ - и  $S$ - входами соответствующих триггеров. Логический элемент  $AND_1$  реализует «свертку» 1-го разряда во 2-й разряд регистра. Его два входа связаны с «прямым» выходом триггера  $T_1$  и «инверсным» выходом триггера  $T_2$ . Его 3-й вход связан с сигналом синхронизации  $C$ . Логический элемент  $AND_1$  анализирует состояния  $Q_1$  и  $Q_2$  триггеров  $T_1$  и  $T_2$ . Если  $Q_1 = 1$  и  $Q_2 = 0$ , это означает, что условие «свертки» для 1-го и 2-го разрядов выполняется. Синхросигнал  $C = 1$  является причиной появления логической 1 на выходе элемента  $AND_1$ . Эта логическая 1 вызывает переключение триггеров  $T_1$  и  $T_2$  в противоположное состояние. Это приводит к выполнению операции «свертки» над первыми двумя разрядами регистра.

Логический элемент  $AND_k$   $k$ -го разряда ( $k = 2, 3, 4, 5$ ) реализует операцию «свертки»  $(k - 1)$ -го и  $k$ -го разрядов в  $(k + 1)$ -й разряд. Его три входа связаны с «прямыми» выходами триггеров  $T_{k-1}$  и  $T_k$  и «инверсным» выходом триггера  $T_{k+1}$ , четвертый вход связан с синхросигналом  $C$ . Логический элемент  $AND_k$  анализирует состояния  $Q_{k-1}, Q_k, Q_{k+1}$  триггеров  $T_{k-1}, T_k$  и  $T_{k+1}$ . Если  $Q_{k-1} = 1, Q_k = 1, Q_{k+1} = 0$ , это означает, что условие «свертки» выполняется.

Синхросигнал  $C = 1$  приводит к переключению триггеров  $T_{k-1}, T_k$  и  $T_{k+1}$  в противоположные состояния. При этом выполняется операция «свертки» над соответствующими разрядами ( $011 \rightarrow 100$ ).

Заметим, что все элементы  $AND_1 - AND_5$  связаны через общий логический элемент  $OR(ИЛИ)$  с контрольным выходом регистра «свертки».

Регистр «свертки» на Рис.10.3 функционирует следующим образом. Входная кодовая информация поступает на входы 1–5 регистра «свертки» и далее поступает на  $S$ -входы триггеров через соответствующие логические элементы  $OR(ИЛИ)$ .

Пусть исходное состояние регистра «свертки» следующее:

5	4	3	2	1
0	1	<b>0</b>	<b>1</b>	<b>1</b>

Ясно, что условие «свертки» выполняется только для 1-го, 2-го и 3-го разрядов. Синхросигнал  $C=1$  приводит к переходу регистра «свертки» в следующее состояние:

5	4	3	2	1
<b>0</b>	<b>1</b>	<b>1</b>	0	0

В этом состоянии условие «свертки» выполняется только для 3-го, 4-го и 5-го разрядов (выделено жирным шрифтом). Очередной синхросигнал  $C=1$  приводит к переходу регистра «свертки» в следующее состояние:

5	4	3	2	1
1	0	0	0	0

«Свертка» исходной кодовой комбинации 01011 закончилась.

**Регистр «свертки» как самоконтролирующееся устройство.** Регистр «свертки» является самоконтролирующимся устройством микропроцессора. Рассмотрим регистр «свертки» на Рис.10.3 с точки зрения самоконтроля. Выходы логических элементов  $AND_1 - AND_5$  регистра «свертки» на Рис.10.3 связаны с контрольным выходом регистра через общий логический элемент  $OR(ИЛИ)$ . Этот выход играет роль контрольного выхода регистра.

Из принципа функционирования регистра свертки вытекает, что логическая 1 возникает на контрольном выходе только для двух ситуаций:

1. Двоичная комбинация, записанная в регистр «свертки», не представлена в «минимальной» форме. Это означает, что условие «свертки» выполняется хотя бы для одной тройки соседних триггеров регистра. Это вызывает появление логический 1 на выходе соответствующего элемента  $AND(I)$ . Следовательно, в этом случае появление логический 1 на контрольном выходе регистра «свертки» указывает на то, что процесс «свертки» не завершен. Это означает, что мы можем



контролировать процесс «свертки» посредством наблюдения за контрольным выходом регистра «свертки».

2. Возникновение постоянной логической 1 на контрольном выходе регистра «свертки» является свидетельством отказа в регистре «свертки». Следовательно, регистр «свертки» является «естественным» самоконтролирующимся устройством.

## 10.11. Счетчик Фибоначчи для минимальной формы

**Задача счета.** Задача счета – это исторически наиболее древняя математическая задача, известная еще в Древнем Риме, для решения которой был придуман особый вычислительный прибор – римский абак. Это был своего рода универсальный счетчик, способный решать практически все арифметические задачи, возникающие в ходе хозяйственной жизни римлян. Значительно позже, в 30-х годах двадцатого столетия молодой английский математик Алан Тьюринг предложил вычислительную машину в виде универсального счетчика, способного решать любую математическую задачу. Этот счетчик считывал информацию в виде символов с потенциально бесконечной ленты и обрабатывал ее по определенному алгоритму. Другим английским ученым Черчем был выдвинут тезис, названный тезисом Черча-Тьюринга, согласно которому любая вычислительная процедура, какой сложности бы она не была, может быть реализована на машине Тьюринга. Это значит, что любой компьютер может быть заменен машиной Тьюринга и соответственно построен на счетчиках.

Оказывается такой прецедент уже был, когда в конце 60-х годов прошлого столетия в СССР была разработана машина «Проминь», использующая в своей основе алгоритмы счета. Но ее быстродействие было значительно ниже быстродействия обычных компьютеров, потому что счетчики, как и машины Тьюринга, перебирают состояния последовательно. Поэтому, невзирая на свою простоту и надежность, данное направление построения компьютеров не развивалось, хотя множество приборов вычислительной техники широко использует счетчики. Например, на их основе разрабатываются электронные часы, управляющие устройства, дальномеры и многое другое.

Однако, существует вполне реальная перспектива построения в будущем наноэлектронных компьютеров на счетчиках. Такие компьютеры будут иметь очень высокое быстродействие за счет чрезвычайно высокой скорости работы своих основных логических элементов, реализованных на атомном уровне. В результате скорость работы счетчиков не будет лимитировать быстродействие нанокomпьютеров, которое будет вполне достаточным для многих практических задач. Но при этом можно будет получать очень надежные и простые структуры узлов нанокomпьютеров, что чрезвычайно важно в условиях высоких скоростей выполнения ими операций. Поэтому поиск структур счетчиков, которые одновременно были бы и быстродействующими и обладали повышенной надежностью, важен как с точки зрения современной техники, так и будущей нанотехники. Такой счетчик и описывается ниже.

**Алгоритм функционирования счетчика Фибоначчи.** Исследования, проведенные на кафедре электроники и компьютерной техники Сумского государственного университета, привели к разработке оригинального алгоритма фибоначчьевого счета (авторы разработки – Алексей Борисенко и Алексей Стахов) [127]. Особенность этого алгоритма состоит в том, что счет ведется в минимальной форме.

Прежде всего, сделаем одно важное замечание. Поскольку в младшем разряде минимальной формы любой ФЦЗ всегда находится 0, то в этом случае мы можем исключить младший разряд из рассмотрения и рассмотреть «укороченный» код Фибоначчи (10.1) с весами разрядов  $1, 2, 5, 8, \dots, F_n$ , то есть,

$$N = a_n F_n + a_{n-1} F_{n-1} + \dots + a_i F_i + \dots + a_3 F_3 + a_2 F_2. \quad (10.55)$$

Например, 5-разрядный код Фибоначчи (10.55) при таком условии будет выглядеть следующим образом:

$$N = a_6 \times 8 + a_5 \times 5 + a_4 \times 3 + a_3 \times 2 + a_2 \times 1 = a_6 a_5 a_4 a_3 a_2. \quad (10.56)$$

Заметим, что последовательную смену минимальных форм в ФЦЗ (10.56) в счетчике Фибоначчи можно представить так, как это показано в Табл. 10.7.

Таблица 10.7. Минимальные формы кода Фибоначчи

6	5	4	3	2	$n$
8	5	3	2	1	$F_n$
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
0	0	1	0	0	3
0	0	1	0	1	4
0	1	0	0	0	5
0	1	0	0	1	6
0	1	0	1	0	7
1	0	0	0	0	8
1	0	0	0	1	9
1	0	0	1	0	10
1	0	1	0	0	11
1	0	1	0	1	12

Анализ Табл.10.7 позволяет установить следующие общие правила, которым подчиняется функционирование такого счетчика Фибоначчи:

**Правило 1.** Очередная единица записывается в младший разряд ФЦЗ (10.56) (то есть, в разряд с номером 2) только в том случае, если двоичные цифры первых двух разрядов равны 0 (случаи, соответствующие числам 0,3,5,8,11).

**Правило 2.** Пусть минимальная форма, представляющая фибоначчьево цифровое представление числа  $N$ , имеет вид:

$n$	$n-1$	...	$2k+3$	$2k+2$	$2k+1$	$2k$	$2k-1$	$2k-2$	$2k-3$	...	4	3	2
$a_n$	$a_{n-1}$	...	$a_{2k+3}$	<b>0</b>	<b>0</b>	1	0	1	0	...	1	0	1

В этой кодовой комбинации имеется характерная группа двоичных разрядов, начиная со 2-го и заканчивая  $(2k+2)$ -м разрядом. Эта группа разрядов состоит из 2-х частей:

1. Группа разрядов от 2-го до  $2k$ -го, в которой младший, то есть, 2-й разряд равен 1, а остальные разряды представляют собой чередующиеся 1 и 0; при этом 1 содержатся во всех разрядах с четными номерами: 2, 4, 6, ...,  $2k-2, 2k$ .
2. Группа из двух разрядов -  $(2k+1)$ -й и  $(2k+2)$ -й, в которых содержатся два последовательных 0 (выделены жирным шрифтом).

Заметим, что для остальных разрядов от  $(2k + 3)$ -го до  $n$ -го выполняется условие минимальной формы (двух единиц рядом не встречается).

Тогда смена ФЦЗ числа  $N$  на ФЦЗ следующего числа  $N + 1$  (при поступлении на вход счетчика очередной счетной единицы) происходит согласно следующему правилу: в  $(2k + 1)$ -й разряд записывается 1, а все остальные разряды справа от  $(2k + 1)$ -го обнуляются; возникающее при этом ФЦЗ имеет вид:

$n$	$n-1$	...	$2k+3$	$2k+2$	$2k+1$	$2k$	$2k-1$	$2k-2$	$2k-3$	...	4	3	2
$a_n$	$a_{n-1}$	...	$a_{k+3}$	0	1	0	0	0	0	...	0	0	0

Докажем, что полученная ФЦЗ Фибоначчи, действительно, представляет число  $N + 1$ . Для этого достаточно вспомнить [2-4], что вес  $(2k + 1)$ -го разряда кода Фибоначчи (10.55) равен  $F_{2k+1}$ , а сумма всех весов разрядов с четными индексами равна:

$$F_2 + F_4 + F_6 + \dots + F_{2k} = F_{2k+1} - 1. \quad (10.57)$$

Из проведенных рассуждений вытекает, что новая ФЦЗ, действительно, является представлением числа  $N + 1$ .

**Правило 3.** Пусть ФЦЗ, соответствующее числу  $N$ , имеет вид:

$n$	$n-1$	...	$2k+2$	$2k+1$	$2k$	$2k-1$	$2k-2$	$2k-3$	$2k-4$	...	4	3	2
$a_n$	$a_{n-1}$	...	$a_{2k+2}$	<b>0</b>	<b>0</b>	1	0	1	0	...	0	1	0

В этой кодовой комбинации имеется характерная группа двоичных разрядов, начиная со 2-го и заканчивая  $(2k + 1)$ -м разрядом. Эта группа разрядов состоит из 2-х частей:

1. Группа разрядов от 2-го до  $(2k - 1)$ -го, в которой младший, то есть, 2-й разряд равен 0, а остальные разряды представляют собой чередующиеся 1 и 0; при этом 1 содержатся во всех разрядах с нечетными номерами: 3, 5, 7, ...,  $2k - 3, 2k - 1$ .
2. Группа из двух разрядов с номерами  $2k$  и  $2k + 1$ , в которых содержатся два последовательных 0 (выделены жирным шрифтом).

Заметим, что для остальных разрядов от  $(2k + 2)$ -го до  $n$ -го выполняется условие минимальной формы (двух единиц рядом не встречается).

Тогда смена ФЦЗ числа  $N$  на ФЦЗ числа  $N+1$  происходит согласно следующему правилу: в  $2k$ -й разряд записывается 1, а все остальные разряды справа от  $2k$ -го обнуляются; возникающее при этом новая ФЦЗ имеет вид:

$n$	$n-1$	...	$2k+2$	$2k+1$	$2k$	$2k-1$	$2k-2$	$2k-3$	$2k-4$	...	4	3	2
$a_n$	$a_{n-1}$	...	$a_{k+2}$	0	1	0	0	0	0	...	0	0	0

Докажем, что полученная ФЦЗ, действительно, представляет число  $N+1$ . Для этого достаточно вспомнить [2-4], что, вес  $2k$ -го разряда кода Фибоначчи (10.55) равен  $F_{2k}$ . С другой стороны, сумма всех весов разрядов с нечетными индексами равна:

$$F_1 + F_3 + F_5 + \dots + F_{2k-1} = F_{2k}. \quad (10.58)$$

Теперь вспомним, что в коде Фибоначчи (10.55) мы исключили из рассмотрения 1-й разряд, значение которого в минимальной форме всегда тождественно равно 0. Тогда для нашего случая мы должны переписать соотношение (10.58) следующим образом:

$$F_3 + F_5 + \dots + F_{2k-1} = F_{2k} - F_1 = F_{2k} - 1. \quad (10.59)$$

Таким образом, числовой эквивалент обнуляемых разрядов равен  $F_{2k} - 1$ , в то время как числовой эквивалент  $2k$ -го разряда равен  $F_{2k}$ . Из проведенных рассуждений вытекает, что новая ФЦЗ, действительно, представляет число  $N+1$ .

Новый алгоритм фибоначчиевского счета показывает, что переход от ФЦЗ числа  $N$  к ФЦЗ числа  $N+1$  осуществляется за один такт, который включает в себя запись 1 в соответствующий разряд и обнуление группы разрядов справа от этого разряда. Это создает предпосылки для создания быстродействующего счетчика Фибоначчи. Но самое главное, что в таком счетчике используются только представление в минимальной форме, которая является основной «контрольной» формой кода Фибоначчи. Контроль минимальной формы осуществляется с помощью несложной комбинационной схемы, представленной на Рис.10.2.

Схема на Рис.10.2 гарантирует обнаружение всех ошибок, которые приводят к искажению минимальной формы исходной ФЦЗ. Если в минимальной форме произошла ошибка, связанная с переходом исходной минимальной формы в другую минимальную форму, то этот случай относится к разряду необнаруживаемых ошибок. Эффективность такого контроля демонстрируется с

помощью Табл.10.6. Это означает, что рассматриваемый счетчик Фибоначчи позволяет решить сразу две важные задачи:

- повысить быстродействие счетчика Фибоначчи;
- обеспечить непрерывный контроль счетчика на всех этапах преобразования информации в счетчике.

Для удобства описания структурной схемы счетчика Фибоначчи мы произведем перенумерацию разрядов в «укороченном» коде Фибоначчи (10.55) и будем использовать код Фибоначчи, в котором нумерация разрядов происходит с 1-го разряда:

$$N = a_{n-1}F_n + a_{n-2}F_{n-1} + \dots + a_{i-1}F_i + \dots + a_2F_3 + a_1F_2. \quad (10.60)$$

**Структурная реализация счетчика Фибоначчи.** Анализ приведенного выше алгоритма счета показывает, что для реализации счетчика Фибоначчи достаточно использовать 5 блоков: Регистр со схемой его управления, Блок анализа, Блок диспозиций, Блок контроля и Блок установки нуля (Рис.10.4). Эти блоки, объединяясь определенным образом между собой, позволяют построить помехоустойчивый быстродействующий счетчик Фибоначчи любой разрядности. Его помехоустойчивость достигается за счет наличия в нем запрещенных состояний, а его быстродействие является следствием отсутствия переносов, которые необходимы в двоичных счетчиках, и сверток и разверток, которые присущи известным фибоначчиевым счетчикам. Эти свойства придают рассматриваемому счетчику указанные выше качества, недоступные в совокупности известным счетчикам.

Рассмотрим работу предлагаемого счетчика Фибоначчи на конкретном примере построения пятиразрядного счетчика Фибоначчи (см. Рис. 10.4).

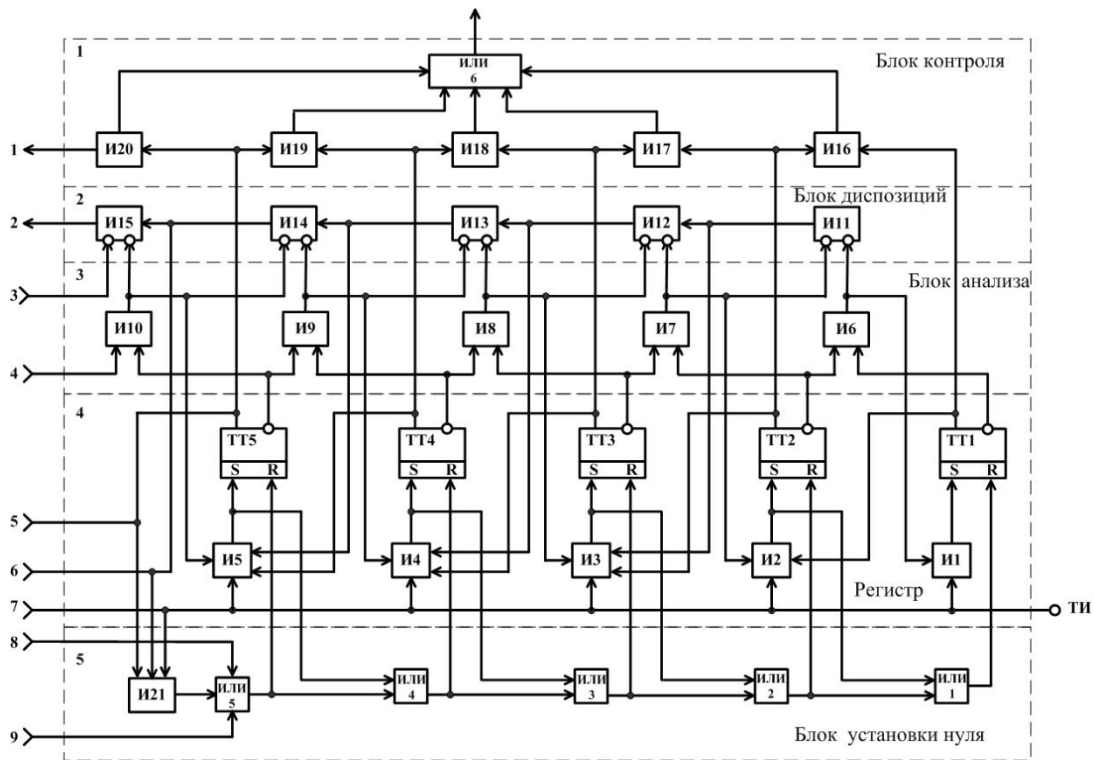


Рис.10.4. Счетчик Фибоначчи

Счетчик состоит из Регистра, образуемого 5 двухступенчатыми  $RS$ –триггерами  $ТТ1–ТТ5$  со схемами их управления  $И1–И5$ , Блока анализа, реализуемого на схемах  $И6–И10$ , Блока диспозиций, (логические схемы  $И11–И15$ ), Блока контроля со схемами  $И16–И20$ , объединяемых схемой  $ИЛИ6$ , Блока установки нуля, состоящего из схемы  $И21$  и схем  $ИЛИ1–ИЛИ5$ . Все эти блоки имеют регулярную структуру и могут быть неограниченной длины. Соответственно и счетчик Фибоначчи на Рис.10.4 может иметь неограниченную разрядность.

Регистр со схемами управления триггерами  $И1–И5$  необходим для запоминания состояний счетчика. В рассматриваемом примере счетчика используются  $RS$ –триггеры, как наиболее простые, хотя могут использоваться и любые другие схемы триггеров, выполняющие их функцию, например,  $JK$ –триггеры. Выходы  $RS$ –триггеров связаны с входами схем  $И6–И10$  блока анализа, а входы – с блоком установки в ноль и шиной тактовых импульсов. Блок анализа контролирует выходы пар соседних триггеров регистра, начиная с  $ТТ1$  и  $ТТ2$ , на предмет их нахождения в одном из трех разрешенных состояний 00, 01 или 10. Состояние же соседних триггеров 11 в соответствии с алгоритмом работы счетчика запрещено. Для выявления запрещенного состояния в счетчик введен Блок контроля, который сигнализирует об ошибочном состоянии счетчика

Фибоначчи. Для анализа расположения единиц в сформированном в счетчике числе введен Блок диспозиций, дающий возможность совместно с Блоком анализа управлять триггерами регистра с целью получения следующего правильного состояния счетчика. Блок установки нуля устанавливает после перехода какого-либо триггера в состояние 1 все триггеры младших разрядов в нуль и в исходное нулевое состояние после окончания цикла счета.

**Исходное состояние счетчика.** Счетчик Фибоначчи на Рис.10.4 функционирует следующим образом. Вначале счетчик Фибоначчи находится в начальном состоянии, когда все триггеры  $ТТ1–ТТ5$  Регистра установлены в состояние 0. В этом случае логические сигналы 0 с прямых выходов триггеров  $ТТ1–ТТ4$  (см. Рис.10.4) поступают на входы логических схем  $И2–И5$ , соответственно. Заметим, что инверсные выходы триггеров  $ТТ1, ТТ2$  связаны со входами логической схемы  $И6$ , и инверсные выходы триггеров  $ТТ2, ТТ3$  связаны со входами логической схемы  $И7$  и т.д. до логической схемы  $И10$ , как показано на Рис.10.4. Это означает, что на выходах логических элементов  $И6–И10$  формируются сигналы 1, которые поступают на соответствующие входы логических схем  $И1–И5$ , соответственно.

**Первый шаг.** Однако, только триггер  $ТТ1$  сможет установиться по первому тактовому импульсу в единичное состояние, так как схемы  $И2–И5$  будут блокированы нулевыми сигналами, поступающими с прямых выходов установленных в 0 предшествующих им триггеров, и соответственно тактовый импульс не сможет пройти на  $S$  – входы триггеров  $ТТ2–ТТ5$ . В результате после прихода первого тактового импульса состояние счетчика станет 00001, то есть, в счетчике будет сформирована фибоначчиева цифровая запись числа 1.

**Второй шаг.** Переход триггера  $ТТ1$  в состояние 1 приведет к тому, что на одном из входов схеме  $И1$  появится запрещающий сигнал 0 с логической схемы  $И6$ , который не разрешает прохождение следующего тактового импульса на  $S$  – вход триггера  $ТТ1$ . Однако, схема  $И2$  пропустит очередной (второй) тактовый импульс на  $S$  – вход триггера  $ТТ2$ , так как на ее входах появляются разрешающие логические сигналы 1 с выхода схемы  $И7$  и прямого выхода триггера  $ТТ1$ , находящегося в состоянии 1. Это значит, что следующий (второй) тактовый импульс приведет к появлению логической 1 на выходе логической



схемы *И2*. Эта логическая 1 установит триггер *ТТ2* в состояние 1 и одновременно через логическую схему *ИЛИ1* установит триггер *ТТ1* в состояние 0. В результате возникнет увеличенное на 1 состояние счетчика – 00010, равное числу 2.

**Третий шаг.** Рассмотрим состояние логических сигналов на входах схемы *И3* после того, как счетчик перейдет в состояние 00010. Для этого напомним, что с выхода логической схемы *И8* на первый вход схемы *И3* поступает разрешающий сигнал 1. Такой же разрешающий сигнал 1 поступает на вход схемы *И3* с прямого выхода триггера *ТТ2*, который находится в состоянии 1. Поскольку триггер *ТТ2* перешел в состояние 1, то на его инверсном выходе появляется логический сигнал 0, что приводит к тому, что на выходе логической схемы *И6* появится сигнал 0. Этот сигнал приведет к появлению логических сигналов 0 на выходах логических схем *И6* и *И7* Блока анализа. Это, в свою очередь, приведет к возникновению логического сигнала 1 на выходе схемы *ИИ1* Блока диспозиций. Этот сигнал поступит еще на один вход схемы *И3*. Таким образом, на трех первых входах схемы *И3* после второго шага формируются разрешающие сигналы 1. В результате третий тактовый импульс проходит через схему *И3* на *S*–вход триггера и переключает его в состояние 1. Этот же сигнал через логическую схему *ИЛИ2* поступает на *R*–вход триггера *ТТ2* и переключает его в состояние 0. В результате счетчик устанавливается в состояние 00100, соответствующее фибоначчиевой цифровой записи числа 3.

**Четвертый шаг.** Так как триггеры *ТТ1* и *ТТ2* находятся в состоянии 0, а триггер *ТТ3* – в состоянии 1, то это приводит к изменению логических сигналов на соответствующих входах и выходах логических схем *И6, И1, И2*. Во-первых, логический сигнал 0 с прямого выхода триггера *ТТ1* поступает на вход схемы *И2* и запрещает прохождение следующего (четвертого) тактового импульса на *S*–вход триггера *ТТ2*. Кроме того, на входы логической схемы *И6* поступают логические сигналы 1 с инверсных выходов триггеров *ТТ1* и *ТТ2*, что приводит к появлению логической 1 на выходе схемы *И6*. Этот сигнал поступает на вход схемы *И1* и подготавливает эту схему к прохождению очередного (четвертого) тактового импульса. Кроме того, этот же сигнал поступает на инверсный вход схемы *ИИ1* Блока диспозиций, что приводит к появлению логического 0 на ее выходе. Это затем приводит к появлению сигналов 0 на выходах всех логических

схем *ИИ12–ИИ14* Блока диспозиций. Эти сигналы поступают на соответствующие входы схем *И3–И5* и запрещают прохождение очередного (четвертого) тактового импульса на *S*–входы триггеров *ТТ3–ТТ5*. Таким образом, четвертый тактовый импульс пройдет только через схему *И1* и установит триггер *ТТ1* в состояние 1. Состояния других триггеров *ТТ2–ТТ5* останутся неизменными. В результате счетчик установится в состояние 00101, что соответствует фибоначчиевой цифровой записи числа 4.

**Пятый шаг.** После четвертого шага состояния логических сигналов на входах и выходах логических схем и триггеров регистра изменятся следующим образом. Во-первых, поскольку триггер *ТТ1* находится в состоянии 1, то это приведет к появлению логического 0 на выходе схемы *И8*. Этот логический 0 является запрещающим сигналом для схемы *И1* и состояние триггера *ТТ1* не изменится после поступления очередного (пятого) тактового импульса. Далее. Поскольку триггер *ТТ3* находится в состоянии 1, то это приведет к появлению запрещающих сигналов 0 на входах логических схем *И2* и *И3* и, следовательно, очередной (пятый) тактовый импульс не изменит состояния триггеров *ТТ2*, *ТТ3*. Проанализируем теперь логические сигналы на входе схемы *И4*. Поскольку триггеры *ТТ1* и *ТТ3* находятся в состоянии 1, то на их инверсных выходах появляются логические сигналы 0. Эти сигналы приведут к появлению логических 0 на выходах схем *И6–И8*. Это, в свою очередь, приведет к появлению логических сигналов 1 на выходе схемы *И11* и затем на выходе схемы *И12*. Разрешающий сигнал 1 с выхода схемы *И12* поступает на вход логической схемы *И4*. На два других входа этой же схемы *И4* поступают также разрешающие сигналы 1 с прямого выхода триггера *ТТ3*, который находится в состоянии 1, и с выхода схемы *И9*. Таким образом, логическая схема *И4* открыта для прохождения тактового импульса. Пятый тактовый импульс проходит через схему *И4* на *S*–вход триггера *ТТ4* и переключает его в состояние 1. Одновременно через логические схемы *ИЛИ3, ИЛИ2, ИЛИ1* триггеры *ТТ3–ТТ1* будут установлены в состояние 0. Как следствие, будет получено фибоначчиева цифровая запись 01000, соответствующая числу 5.

**Шестой шаг.** Нетрудно показать, что в состоянии 01000 только одна схема *И1* будет открыта для прохождения очередного (шестого) тактового импульса. Все остальные логические схемы *И2–И5* будут закрыты. После прихода

шестого тактового импульса счетчик перейдет в состояние 01001, представляющее фибоначчиеву цифровую запись числа 6.

**Седьмой, восьмой и последующие шаги.** Далее по аналогии будет получено число 01010, равное 7, после которого появятся единицы на всех трех разрешающих входах схемы *И5* одновременно и соответственно триггер *ТТ5* установится восьмым тактовым импульсом в состояние 1, а все остальные триггеры через схемы *ИЛИ4–ИЛИ1* будут установлены в состояние 0. Затем процесс счета будет идти до тех пор, пока не будет получено максимальное фибоначчиево число 10101, равное 12. В этом случае на схему *И21*, принадлежащей Блоку установки нуля, будут поданы логические 1 с прямого выхода триггера *ТТ5* и логической схемы *И4* Блока диспозиций. Это позволит тактовому импульсу через схемы *ИЛИ5–ИЛИ1* установить все триггеры *ТТ1–ТТ5* в состояние 0. После чего счетчик будет готов к новому циклу счета.

**Увеличение разрядности счетчика.** Дополнительные входы и выходы счетчика 1-9 предназначены для дальнейшего неограниченного наращивания старших разрядов счетчика, хотя можно наращивать и младшие разряды справа, используя для этого стандартную структуру одного разряда счетчика. Она состоит из двухразрядной схемы *ИЛИ* Блока установки нуля, 4-входовой схемы *И* соответствующего ей триггера Регистра, схем *И* Блока анализа и Блока диспозиций. Такая универсальная реализация счетчика позволяет, с одной стороны, упростить его проектирование, особенно при реализации на ПЛИС, а с другой повысить его надежность, так как некоторые связи в схеме с точки зрения функционирования счетчика являются излишними. Но именно они позволяют сделать схему счетчика универсальной и регулярной, и при выходе некоторых связей из строя автоматически заменять их другими, организовывая, таким образом, по сути, дублирование этих связей. А надежность на сегодня – это один из главных показателей качества работы любого цифрового устройства, в том числе и счетчиков.

**Блок контроля.** Блок контроля является, по сути, внешним блоком по отношению к счетчику и не влияет на алгоритм счета. Его задача – это выявление ошибок, возникающих в процессе счета. Появление случайно двух единиц в рядом стоящих триггерах Регистра приведет к тому, что одна их схем *И* Блока

контроля выдаст на своем выходе 1, которая через схему *ИЛИ* поступит на выход Блока контроля, сигнализируя, таким образом, об ошибке. В случае обнаружения ошибки счетчик может начать новый цикл счета или, если есть дублирующий его резервный счетчик, переключиться на него.

**Быстродействие счетчика.** Обратим также внимание на то, что в данном счетчике отсутствуют переносы из разряда в разряд, что повышает его быстродействие по сравнению с известными двоичными помехоустойчивыми счетчиками. Задержка сигнала в рассматриваемом счетчике возникает только в цепочке схем *И* в Блоке диспозиций и схем *ИЛИ* в Блоке установки нуля. При этом, чем выше разрядность счетчика, тем эта задержка выше. Однако, она происходит в течение одного такта параллельно с переключением одного триггера, а не за ряд тактов, как это обычно происходит при помехоустойчивом двоичном счете. По быстродействию данный счетчик сравним с обычным двоичным счетчиком со сквозным переносом, но отличается от него высокой помехоустойчивостью. Данный счетчик был реализован в виде компьютерной модели программой Веркбэнч и полностью подтвердил свою работоспособность.

Таким образом, предлагаемый счетчик является помехоустойчивым, достаточно быстродействующим, отличается регулярной структурой и высокой надежностью. Поэтому его имеет смысл использовать для реализации в виде ПЛИС в различных схемах цифровых устройств, требующих высокого быстродействия, помехоустойчивости и надежности, таких, например, как частотомеры и управляющие устройства.

**Микроконтроллеры Фибоначчи.** В основу специализированного микроконтроллера Фибоначчи может быть положен рассмотренный выше помехоустойчивый счетчик Фибоначчи с встроенной системой контроля, который может обнаруживать и исправлять более 99 процентов ошибок, возникающих в микропроцессоре под влиянием внешних и внутренних воздействий (см. Табл.10.6), причем обнаружение ошибок осуществляется на каждом такте работы процессора, без снижения его скорости. Каждый счетчик дублируется, что позволяет оперативно исправлять возникшую ошибку по «принципу дублирования». Такое техническое решение существенно увеличивает информационную надежность микроконтроллеров систем управления. Как упоминалось, предлагаемый счетчик Фибоначчи (Рис.10.4) является

помехоустойчивым, быстродействующим, отличается регулярной структурой, что важно для микроэлектронной реализации, позволяет обнаруживать ошибки в момент их возникновения и поэтому обладает достаточно высокой информационной надежностью.

Счетчик может стать основой для создания микроконтроллера Фибоначчи, который построен на счетном принципе. Все функциональные узлы микроконтроллера Фибоначчи (сумматоры, вычитатели, умножители, делители) основываются на счетчиках Фибоначчи. Это резко упрощает их техническую реализацию. Обеспечение высокого быстродействия осуществляется за счет использования оригинальных технических решений. Главным преимуществом такого подхода является создание микроконтроллеров для специальных приложений, обеспечивающих непрерывный самоконтроль и высокую помехозащищенность.

## **10.12. Помехоустойчивое АЛУ Фибоначчи**

**Проблема помехоустойчивых вычислений.** Как известно, компьютерная программа реализуется с помощью процессора. Процессор состоит из триггеров, связанных комбинационной логикой. В этом случае реализация программы сводится к переключению триггеров. К сожалению, невозможно полностью исключить ошибки, возникающие в результате неисправной работы компьютерных элементов. Но при этом необходимо различать два типа неисправной работы компьютерных элементов. Первый тип называется постоянным отказом, когда элемент, однажды отказавший, «выбывает из игры». Второй тип неисправной работы называется случайным отказом, когда элемент «отказывает» временно, то есть, в случайные моменты времени, а в другие моменты времени элемент работает исправно. Последний тип отказов называется сбоем. Сбои в процессоре возникают под влиянием различных внешних и внутренних факторов, которые приводят к сбоям в компьютерных элементах и их электронных цепях.

Но какой же вид неисправной работы является наиболее характерным для электронных элементов? Сейчас уже установлено, что сбои в электронных элементах, в частности, в триггерах, возникают значительно более часто, чем отказы. Существует два режима работы электронных элементов:

а) режим, когда элемент находится в стабильном (устойчивом) состоянии и

б) режим переключения, когда элемент переключается из одного устойчивого состояния в другое.

Экспериментально доказано, что интенсивность сбоев триггеров в режиме переключения больше на 2-3 порядка, чем интенсивность сбоев элементов, находящихся в устойчивом, то есть, стабильном состоянии. Отсюда вытекает, что сбои триггеров в режиме переключения являются наиболее вероятной причиной ненадежного функционирования процессоров. Вот почему проектирование самоконтролирующихся цифровых автоматов, гарантирующих эффективный контроль сбоев триггеров, является одной из важнейших проблем проектирования самоконтролирующихся процессоров.

Итак, мы пришли к идее создания процессора, позволяющего обнаруживать сбои триггеров, возникающих в момент их переключения. Такой процессор мы будем называть помехоустойчивым процессором.

**Контроль «базовых микроопераций».** Выше было показано, что все арифметические операции в АЛУ Фибоначчи сводятся к выполнению четырех «базовых микроопераций»: свертка, развертка, перемещение, поглощение.

Оценим теперь потенциальную возможность обнаружения ошибок в «базовых микрооперациях», которые могут возникнуть при их выполнении.

Известно, что потенциальный коэффициент обнаружения ошибок – это отношение числа всех обнаруживаемых ошибок к числу всех возможных ошибок. Попытаемся объяснить сущность процесса обнаружения ошибок на примере микрооперации «свертки»:

$$011 \rightarrow 100. \quad (10.61)$$

Микрооперация «свертка» (10.61) выполняется над трехразрядной кодовой комбинацией (10.61). В общем случае существует  $2^3 = 8$  возможных переходов при выполнении микрооперации (10.61). Только один из них, а именно переход (10.61) является правильным переходом. Кодовые комбинации

$$\{011, 100\}, \quad (10.62)$$

участвующие в правильном переходе (10.61), называются разрешенными КК.

Остальные КК

$$\{000, 001, 010, 101, 110, 111\} \quad (10.63)$$

называются запрещенными КК.

Идея обнаружения ошибок состоит в следующем. Если в результате выполнения микрооперации (10.61) возникает одна из «запрещенных» кодовых комбинаций (10.63), этот факт сам по себе является индикатором ошибки при выполнении данной микрооперации. Заметим, что, если возникает ошибочный переход в другую разрешенную кодовую комбинацию, то это является свидетельством необнаруживаемой ошибки.

Рассмотрим теперь все возможные ситуации, которые могут возникнуть при выполнении микрооперации (10.61). Ясно, что переходы

$$011 \rightarrow \left. \begin{array}{l} 011 \\ 000 \\ 001 \\ 010 \\ 101 \\ 110 \\ 111 \end{array} \right\} \quad (10.64)$$

являются ошибочными.

Проанализируем теперь ошибочные переходы (10.64). Мы видим, что ошибочный переход

$$011 \rightarrow 011 \quad (10.65)$$

является необнаруживаемым, поскольку кодовые комбинации (10.62) являются разрешенными для микрооперации «свертка». Рассмотрим теперь переход (10.65) с арифметической точки зрения. Ясно, что этот ошибочный переход состоит в сохранении исходной кодовой комбинации. И если мы проанализируем этот переход с арифметической точки зрения, мы можем увидеть, что этот переход не разрушает числовую информацию и не влияет на конечный результат арифметической операции. Следовательно, ошибочный переход (10.65) не приводит к арифметической ошибке катастрофического характера. Это означает, что переход (10.65) является ошибочным, но эта ошибка не влияет на конечный результат вычислений. Эта ошибка только задерживает процесс обработки данных. Все остальные ошибочные переходы из (10.64) приводят к разрушению информации и, следовательно, для этих случаев ошибки носит катастрофический характер.

Основной вывод, вытекающий из этих рассуждений, состоит в том, что множество «катастрофических» кодовых комбинаций из (10.64) совпадает со множеством «обнаруживаемых» кодовых комбинаций из (10.63). Это означает,

что все «катастрофические» переходы для операции «свертка» являются обнаруживаемыми. Следовательно, из этих рассуждений вытекает, что контроль «фибоначчиевой» схемы, основанный на контроле операции «свертка», обладает «абсолютной» (то есть 100%-й) способностью к обнаружению всех «катастрофических» ошибок, которые могут возникнуть при переключении триггеров, участвующих в микрооперации «свертка».

Подобный вывод можно сделать для всех «базовых микроопераций». Но реализация любого алгоритма обработки данных в микропроцессоре Фибоначчи сводится к выполнению определенной последовательности «базовых микроопераций». Так как контроль «фибоначчиевых» схем или блоков, основанный на «базовых микрооперациях», обладает «абсолютной» ошибкообнаруживающей способностью, из рассмотренной выше рассуждений вытекает возможность разработки арифметических самоконтролирующихся микропроцессоров, обладающих «абсолютной» ошибкообнаруживающей способностью по отношению к «катастрофическим» ошибкам, которые могут возникнуть в микропроцессоре Фибоначчи при переключении триггеров. Этот факт дает нам право назвать такой микропроцессор Фибоначчи помехоустойчивым.

Таким образом, мы, наконец, поняли основное преимущество «помехоустойчивой» фибоначчиевой арифметики, основанной на «базовых микрооперациях», в сравнении с классической двоичной арифметикой, которая лишена каких-либо механизмов контроля арифметических преобразований. Мы доказали, что существует возможность создать микропроцессор Фибоначчи для «помехоустойчивых» вычислений и что такой микропроцессор Фибоначчи гарантирует обнаружение всех возможных «сбоев» триггеров, возникающих при их переключениях!

### **10.13. Самоконтролирующийся «фибоначчиевый» модуль**

**Принцип «причина-следствие».** Аппаратная реализация «фибоначчиевого» модуля основана на принципе "причина - следствие". Сущность принципа состоит в следующем. Исходная информация, которая подвергается обработке, преобразуется в некоторый результат с использованием некоторой "базовой микрооперации". После этого проверяется соответствие



«результата» некоторой «причине», то есть, «результат» должен соответствовать своей «причине».

Например, при выполнении микрооперации «свертка» над кодовой комбинации 011 ("причина") мы получаем новую кодовую комбинацию 100 ("результат"), который является необходимым условием для выполнения микрооперации «развертка». Это означает, что правильное выполнение «свертки» приводит к условию для выполнения «развертки». Аналогично правильное выполнение «развертки» приводит к условию «свертки». Из этого рассуждения вытекает, что микрооперации «свертка» и «развертка» являются взаимно контролируемыми!

Рассмотрим другие микрооперации с этой точки зрения. Результатом правильного выполнения микрооперации «перемещение»

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \rightarrow = & \\ \hline 0 & 1 \\ \hline \end{array}$$

является

комбинация  $\begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline \end{array}$ . Кодовые комбинации  $\begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array}$  и  $\begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline \end{array}$  являются «ошибочными» для

«перемещения». Микрооперация «поглощение»

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline \updownarrow = & \\ \hline 1 & 0 \\ \hline \end{array}$$

является правильной,

если мы получаем в качестве результата кодовую комбинацию

$$\begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array}$$

. Кодовые

комбинации  $\begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \end{array}$  и  $\begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline \end{array}$  являются «ошибочными» для «поглощения».

**Техническая реализация.** При «регистровой интерпретации» установление соответствия между «причиной» и «результатом» реализуется с помощью «контрольного триггера». «Причина» устанавливает соответствующий «контрольный триггер» в состояние 1, а правильное выполнение микрооперации («результат» соответствует «причине») переключает «контрольный триггер» в состояние 0. Если «результат» не соответствует «причине» (микрооперация выполнена неправильно), тогда «контрольный триггер» остается в состоянии 1 и этот факт является индикатором ошибки.

Проверка «записи» информации в регистр выполняется посредством сравнения выходной информации регистра после «записи» с входной информацией регистра.

Если мы проанализируем «причины» и «результаты» для каждой базовой микрооперации, мы заключаем, что каждый «результат» является инверсией «причины», то есть, все микрооперации могут быть сведены к «инвертированию» разрядов или триггеров, участвующих в микрооперации.

Блок-схема «фибоначчиевого» модуля для реализации принципа «причина-следствие» показана на Рис.10.5.

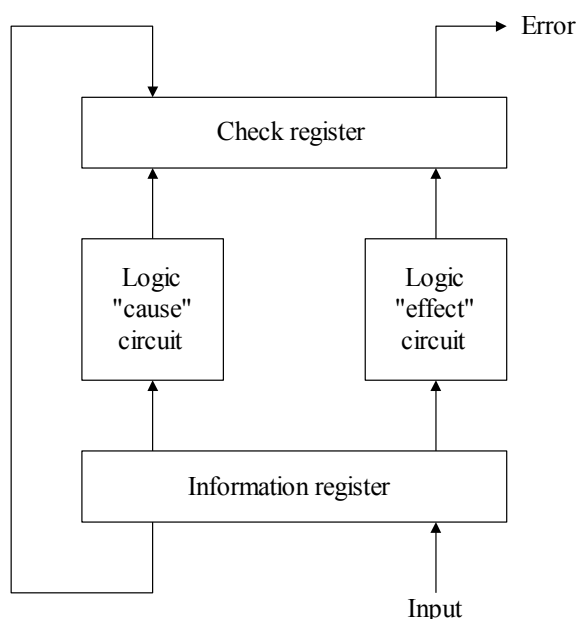


Рис.10.5. Блок-схема «фибоначчиевого» модуля

Модуль на Рис.10.5 состоит из «информационного» и «контрольного» регистров (information and check registers), которые связаны с помощью логических схем “cause” («причина») и “effect” («следствие»). Кодовая информация, поступающая в информационный регистр, анализируется логической схемой “cause” («причина»).

Таким образом, если мы получили «нулевую» кодовую комбинацию в контрольном регистре после завершения всех микроопераций, это означает что все «причины» соответствовали своим «результатам», то есть, все микрооперации были выполнены правильно. Если контрольный регистр содержит хотя бы одну двоичную 1 в некотором триггере, это означает что, по крайней мере, хотя бы одна «базовая микрооперация» выполнена неправильно. Двоичные единицы в

триггерах контрольного регистра вызывают сигнал ошибки на выходе «Ошибка» (“Error”) модуля на Рис.10.5.

Наиболее важное преимущество принципа «причина – следствие», который реализован в «фибоначчиевом» модуле на Рис.10.5, состоит в том, что обнаружение ошибки осуществляется в момент ее появления. Исправление ошибки в микрооперации осуществляется повторением этой микрооперации.

Следовательно, рассматриваемый подход, основанный на принципе «причина-следствие», позволяет обнаруживать и затем исправлять посредством повторения все арифметические ошибки, возникающие в «фибоначчиевом» модуле в момент переключения триггеров с гарантией 100 %.

Рассмотренный выше принцип «причина-следствие» был положен в основу самоконтролирующегося «фибоначчиевого» модуля, который был реализован в виде «большой интегральной схемы» (БИС) [18]. Модуль выполнял следующие микрооперации: запись, чтение, свертка, развертка, перемещение, поглощение, «фибоначчево» сложение, вычитание, приведение к минимальной форме, циклический сдвиг, конъюнкция, дизъюнкция, сложение по модулю 2. Наличие контрольного выхода «Ошибка» (“Error”) является важным преимуществом «фибоначчиевого» модуля. Если на контрольном выходе модуля формируется двоичный сигнал 1 (наличие «ошибки»), то все информационные выходы модуля блокируются. Чтобы исправить «ошибку», необходимо повторить предшествующую микрооперацию. Если при повторении на контрольном выходе возникает двоичный сигнал 0 (отсутствие «ошибки»), то это означает, что «ошибка» возникла в результате «случайного сбоя», и блокирование информационных выходов снимается. Если при повторении микрооперации двоичный сигнал 1 снова появляется на выходе «ошибка», это означает, что «ошибка» является следствием отказа в модуле и блокирование информационных выходов остается.

## **10.14. «Золотые» аналого-цифровые и цифро-аналоговые преобразователи**

Микропроцессоры, используемые в системах управления, связаны с другими устройствами управляемой системы с помощью аналого-цифровых и цифроаналоговых преобразователей (АЦП и ЦАП). Еще одно достоинство кодов золотой  $p$ -пропорции

$$A = \sum_i a_i \Phi_p^i, \quad (10.66)$$

частным случаем которого является КЗП (10.1), состоит в том, что на их основе могут быть созданы самоконтролирующиеся и самокорректирующиеся АЦП и ЦАП, что чрезвычайно важно для систем управления сложными техническими объектами.

Основы теории «золотых» АЦП и ЦАП впервые изложены автором в работах [63,64]. На основе идей, изложенных в этих работах, в Таганрогском радиотехническом институте (Россия) и затем в Винницком политехническом институте (Украина) под руководством автора было выполнено 10 кандидатских диссертаций. Защищена также одна докторская диссертация (А.Д. Азаров, Винницкий государственный университет).

**«Золотые» резистивные делители.** Основу ЦАП и АЦП для электрических величин составляют резистивные делители или аттенюаторы. В работах [63,64] изложена теория «золотых» резистивных делителей (Рис.10.6).

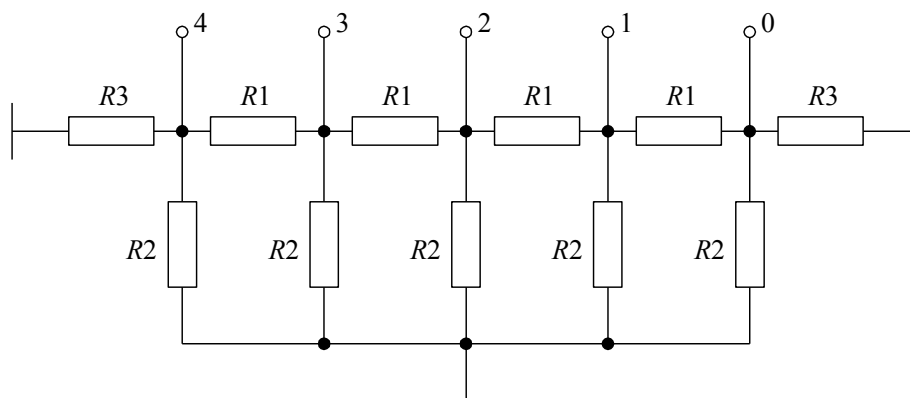


Рис.10.6. «Золотые» резистивные делители

Делитель на Рис.10.6 состоит из резисторов трех типов  $R1, R2, R3$ , номиналы которых выбраны следующим образом:

$$R1 = \Phi_p^{-p} R; \quad R2 = \Phi_p^{p+1} R; \quad R3 = \Phi_p R, \quad (10.67)$$

где  $R$  – некоторое эталонное сопротивление,  $\Phi_p$  – золотая  $p$ -пропорция, ( $p = 0, 1, 2, 3, \dots$ ).

Ясно, что делитель на Рис.10.6 задает теоретически бесконечное число различных делителей, так как каждое  $p$  «генерирует» свой собственный

делитель. В частности, при  $p = 0$  золотая  $p$ -пропорция  $\Phi_p = \Phi_0 = 2$  и резисторы  $R1, R2, R3$  принимают следующие значения:

$$R1 = R; R2 = 2R; R3 = 2R, \quad (10.68)$$

откуда вытекает, что при  $p = 0$  делитель на Рис.10.6 сводится к классическому двоичному делителю, на котором основана вся современная цифровая метрология и техника аналого-цифрового и цифроаналогового преобразования.

Для случая  $p = 1$  резисторы  $R1, R2, R3$  принимают следующие значения:

$$R1 = \Phi^{-1}R; R2 = \Phi^2R; R3 = \Phi R, \quad (10.69)$$

где  $\Phi = (1 + \sqrt{5})/2$  - классическая золотая пропорция.

Для вывода математических соотношений, лежащих в основе «золотого» резистивного делителя на Рис.10.6, будем использовать следующие тождества для золотой  $p$ -пропорции:

$$\Phi_p = 1 + \Phi_p^{-p} \quad (10.70)$$

$$\Phi_p^{p+2} = \Phi_p^{p+1} + \Phi_p. \quad (10.71)$$

Используя закон Ома и тождество (10.71), легко вычислить эквивалентное сопротивление резисторной цепи на Рис.10.6 справа от точки 0 и слева от точки 4:

$$R_{e1} = \frac{R2 \times R3}{R2 + R3} = \frac{\Phi_p^{p+1}R \times \Phi_p R}{\Phi_p^{p+1}R + \Phi_p R} = R. \quad (10.72)$$

Используя выражение (10.72) и тождество (10.70), легко вычислить эквивалентное сопротивление резисторной цепи на Рис.10.6 справа от точки 1 и слева от точки 3:

$$R_{e2} = \Phi_p^{-p}R + R = \Phi_p R. \quad (10.73)$$

Используя выражения (10.69), (10.70) и тождества (10.67), (10.68), легко показать, что эквивалентное сопротивление резисторной цепи на Рис.10.5 слева или справа от любой точки 1,2,3 всегда задается выражением (10.72).

Рассматривая эквивалентную схему резисторной цепи на Рис.10.6 в любой точке 0,1,2,3,4 как параллельное соединение «вертикального» резистора  $R2 = \Phi^{p+1}R$  и двух эквивалентных резисторов справа и слева от точек 0,1,2,3,4, значения которых задаются выражением (10.72), мы можем вычислить эквивалентное сопротивление резисторной цепи на Рис.10.6 в любой из точек 0,1,2,3,4:

$$R_{e3} = \frac{\Phi_p R \times R}{\Phi_p R + R} = \frac{\Phi_p}{\Phi_p + 1} R = \frac{1}{1 + \Phi_p^{-1}} R. \quad (10.74)$$

Легко найти коэффициент передачи по напряжению между соседними точками 0–1, 1–2, 2–3, 3–4:

$$K_U = \frac{U}{\Phi_p}, \quad (10.75)$$

то есть, коэффициент передачи по напряжению между соседними точками 0,1,2,3,4 в резистивной цепи на Рис.10.6 обратно пропорционален золотой  $p$ -пропорции  $\Phi_p$ .

Заметим, что при  $p = 0$   $\Phi_p = 2$  и выражения (10.74) и (10.75) сводятся к хорошо известным выражениям для классических двоичных делителей типа  $R-2R$ :

$$R_{e3} = \frac{2}{3} R \text{ и } K_U = \frac{U}{2}, \quad (10.76)$$

Таким образом, «золотые» резистивные делители, основанные на золотой  $p$ -пропорции  $\Phi_p$ , являются основой «золотой» цифровой метрологии [63] и «золотых» цифро-аналоговых и аналогоцифровых преобразователей [64].

**«Золотые» цифроаналоговые преобразователи.** Структурная схема «золотого» ЦАП, основанного на «золотом» резистивном делителе (Рис.10.6), представлена на Рис.10.7.

ЦАП на Рис.10.7 осуществляет преобразование «золотого» 5-разрядного кода (10.1) в электрическое напряжение  $U_{out}$  на выходе ЦАП. Однако, число разрядов ЦАП может быть легко увеличено до произвольного числа  $n$  путем расширения «золотого» резистивного делителя слева направо. «Золотой» ЦАП содержит 5 ( $n$  в общем случае) генераторов стандартного электрического тока  $I_0$  и 5 ( $n$  в общем случае) электрических ключей  $K_0 - K_4$ . Состояния ключей управляются двоичными цифрами  $a_4 a_3 a_2 a_1 a_0$  кода золотой  $p$ -пропорции (10.66). Для случая  $a_i = 1$  ключ  $K_i$  ( $i = 0, 1, 2, 3, \dots, n-1$ ) находится в замкнутом положении, для случая  $a_i = 0$  - в разомкнутом. Можно показать, что замкнутый ключ  $K_i$  приводит к появлению следующего напряжения в точке  $i$ :

$$U_i = \beta_p I_0 R, \quad (10.77)$$

где  $\beta_p = 1/(1 + \Phi_p^{-1})$ .

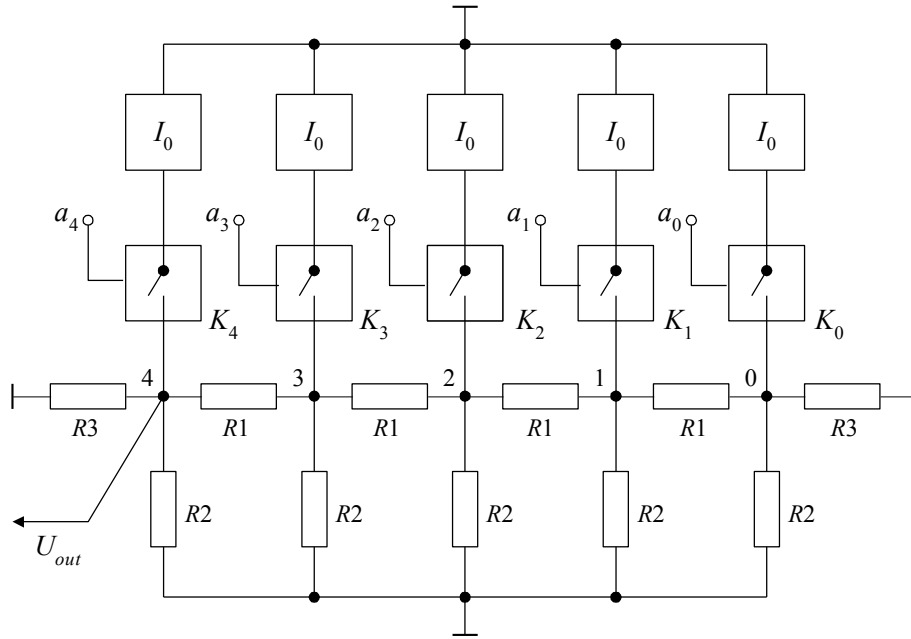


Рис.10.7. «Золотой» ЦАП

Так как напряжение (10.77) передается от  $i$ -й точки к  $(i+1)$ -й точке с коэффициентом передачи  $1/\Phi_p$ , тогда на выходе ЦАП возникнет следующее напряжение:

$$U_{out} = \frac{\beta_p I_0 R}{\Phi_p^{n-i-1}} = \frac{\beta_p I_0 R}{\Phi_p^{n-1} \times \Phi_p^{-1}} = \frac{\beta_p I_0 R}{\Phi_p^{n-1}} \times \Phi_p^i.$$

Используя принцип суперпозиции, легко вычислить напряжение на выходе ЦАП при подаче на вход ЦАП  $n$ -разрядного кода золотой  $p$ -пропорции  $a_{n-1}a_{n-2}\dots a_2a_1a_0$ :

$$U_{out} = B_p \sum_{i=0}^{n-1} a_i \Phi_p^i, \quad (10.78)$$

где  $B_p = \beta_p I_0 R / \Phi_p^{n-1}$ .

Как следует из выражения (10.78), ЦАП на Рис.10.7 преобразует код золотой  $p$ -пропорции  $a_{n-1}a_{n-2}\dots a_2a_1a_0$  в эквивалентное электрическое напряжение  $U_{out}$  с точностью до постоянного коэффициента  $B_p$ .

**Метрологический контроль «золотого» ЦАП.** В цифровой метрологии [63] существует проблема контроля линейности ЦАП в процессе производства и эксплуатации. Для классического двоичного ЦАП для контроля линейности

используется следующее математическое соотношение, связывающее веса двоичных разрядов:

$$2^n = \sum_{i=0}^{n-1} 2^i + 1.$$

Математические свойства золотой пропорции дают более широкие возможности для контроля линейности ЦАП. В частности, для случая  $p=1$  контроль линейности «золотого» ЦАП основывается на следующих соотношениях:

$$\Phi^n = \Phi^{n-1} + \Phi^{n-2} = \Phi^{n-1} + \Phi^{n-3} + \Phi^{n-4} = \Phi^{n-1} + \Phi^{n-3} + \Phi^{n-5} + \Phi^{n-6} = \dots \quad (10.79)$$

Контроль линейности ЦАП осуществляется путем последовательной развертки (слева направо) следующей входной «золотой» кодовой комбинации:

$$\begin{array}{cccccccc} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{array}$$

При этом выходное напряжение ЦАП  $U_{out}$  должно оставаться неизменным.

Используя операции свертки и развертки, мы можем контролировать линейность ЦАП для любого ЗЦЗ на входе ЦАП, что обеспечивает возможность эффективного контроля ЦАП как в процессе производства, так и эксплуатации.

**«Золотой» самокорректирующийся аналого-цифровой преобразователь.** Наиболее широкую известность получили инженерные разработки СКТБ "Модуль" (Винницкий политехнический институт) по проектированию «золотых» самокорректирующихся аналого-цифровых и цифроаналоговых преобразователей. Исследования в этой области, выполненные в Винницком техническом университете, показали, что применение кода золотой пропорции позволяет одновременно улучшить все технические параметры АЦП и ЦАП, в частности, точность, быстродействие и самое главное - температурную и временную метрологическую стабильность АЦП и ЦАП. Улучшение всех технических параметров таких АЦП и ЦАП достигалось за счет использования свойства "многозначности" «золотого» представления одной и той же величины.

В СКТБ «Модуль» Винницкого политехнического института с 1986 по 1989 гг. было налажено мелкосерийное производство «золотых» АЦП и ЦАП. Наиболее важной технической характеристикой таких АЦП и ЦАП являлась их высокая метрологическая стабильность и низкие требования к технологической



точности изготовления аналоговых элементов АЦП. Это достигалось с помощью системы встроенного контроля аналоговой части АЦП, реализуемой с помощью специального микропроцессора. Благодаря встроенному контролю в АЦП осуществлялась периодическая автонастройка АЦП на заданную точность, чем и поддерживались высокие точностные параметры АЦП. По мнению известных измерительных фирм СССР, куда поставляло свои разработки СКТБ "Модуль", в СССР на тот период не было подобных разработок. Подробное описание этих разработок можно найти в брошюре [18].

**Еще раз о преимуществах КФ и КЗП.** Важно еще раз подчеркнуть, что КФ (10.1) и КЗП (10.2) являются позиционными двоичными системами счисления, которые сохраняют все известные преимущества классической двоичной системы счисления. С другой стороны, они являются избыточными кодами, сохраняющими все свойства классических избыточных кодов, и позволяют, как показано выше, достаточно эффективно контролировать все информационные преобразования в микропроцессоре Фибоначчи. Но в качественном отношении КФ (10.1) и КЗП (10.2) превосходят как другие избыточные коды, так и все известные позиционные системы счисления, прежде всего потому, что они совмещают в себе свойства позиционных систем счисления и избыточных кодов одновременно.

Применение этих кодов в компьютерах не приводит к необходимости преобразования исходного кода в избыточный, что автоматически решает проблему кодирования-декодирования и приводит к упрощению вычислительных структур.

При этом эти коды позволяют создавать компьютерные и измерительные устройства и структуры, обладающие следующими свойствами:

1. Свойство самоконтроля всех компьютерных структур (счетчиков, регистров, таймеров, электронной памяти, АЛУ и др.)
2. Свойство самосинхронизации двоичных сигналов при их последовательной передаче по каналам связи [126]
3. Уменьшение числа единиц для хранения одного и того же диапазона чисел, которое вместе со свойством минимальной формы позволяют уменьшить рассеиваемую мощность в электронной памяти.
4. КЗП (10.2) обладает исключительными преимуществами при реализации АЦП и ЦАП и позволяет создавать на его основе самоконтролирующиеся АЦП и ЦАП

для сигнальных микропроцессоров и самокорректирующиеся АЦП и ЦАП для метрологических систем, нечувствительные к погрешностям изготовления аналоговых элементов АЦП и ЦАП, их старению и влияниям температуры («вечные» АЦП и ЦАП).

## Глава 11

### «ЗОЛОТАЯ» ТРОИЧНАЯ ЗЕРКАЛЬНО-СИММЕТРИЧНАЯ АРИФМЕТИКА

#### 11.1. Троичное зеркально-симметричное представление

Впервые «золотая» троичная зеркально-симметричная арифметика описана автором в статье «Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic», опубликованной международным журналом «The Computer Journal» в 2002 г. [83]. Статья вызвала положительную реакцию представителей западного компьютерного сообщества. Первым из американских ученых, поздравивших автора с ее публикацией, стал выдающийся американский специалист Дональд Эрвин Кнут, автор всемирно известного бестселлера «Искусство программирования».

**Преобразование «золотого» двоичного представления в «золотое» троичное представление.** В главе 9 было введено понятие  $\Phi$ -кода:

$$N = \sum_i a_i \Phi^i, \quad (11.1)$$

где  $a_i \in \{0,1\}$  - двоичная цифра,  $\Phi^i$  - вес  $i$ -го разряда,  $\Phi$  - основание системы счисления (11.1) («золотая пропорция»).

Цифровая запись  $\Phi$ -кода (11.1) имеет следующий вид:

$$N = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}. \quad (11.2)$$

В дальнейшем мы будем использовать так называемую минимальную форму представления (11.2). Это означает, что каждый бит  $a_k = 1$  в «золотом» представлении (11.2) всегда «окружен» двумя соседними нулями, то есть  $a_{k-1} = a_{k+1} = 0$ .

Как известно, степени золотой пропорции  $\Phi$  связаны следующим соотношением:

$$\Phi^{k+1} = \Phi^k + \Phi^{k-1}; \quad (k = 0, \pm 1, \pm 2, \pm 3, \dots). \quad (11.3)$$

Тождество (11.3) можно представить в виде:

$$\Phi^k = \Phi^{k+1} - \Phi^{k-1}. \quad (11.4)$$

Тождество (11.4) имеет следующую кодовую интерпретацию:

$$\begin{array}{cccccccc} k+1 & k & k-1 & k+1 & k & k-1 & & \\ 0 & 1 & 0 & = & 1 & 0 & \bar{1} & , \end{array} \quad (11.5)$$

где  $\bar{1}$  - отрицательная единица, то есть,  $\bar{1} = -1$ .

Из (11.5) вытекает, что положительная двоичная 1  $k$ -го разряда может быть преобразована в две троичные единицы: положительную единицу 1  $(k+1)$ -го разряда и отрицательную единицу  $\bar{1}$   $(k-1)$ -го разряда.

Кодовое преобразование (11.5) может быть использовано для перевода «минимальной» формы «золотого» двоичного представления (11.2) в «золотое» троичное представление.

Рассмотрим «золотое» двоичное представление числа 5 в  $\Phi$ -коде (11.1):

$$\begin{array}{cccccccc} & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \\ 5 = & 0 & 1 & 0 & 0 & 0, & 1 & 0 & 0 & 1 \\ & \uparrow & \downarrow & \uparrow & & \uparrow & \downarrow & \uparrow & & \end{array} \quad (11.6)$$

Здесь стрелками показано преобразование, которое осуществляется над тройкой соседних двоичных цифр 010, согласно (11.5).

Заметим, что представление (11.6) является «золотой» цифровой записью (ЗЦЗ) следующей суммы:

$$\begin{aligned} 5 = \Phi^3 + \Phi^{-1} + \Phi^{-4} &= \frac{L_3 + F_3\sqrt{5}}{2} + \frac{L_{-1} + F_{-1}\sqrt{5}}{2} + \frac{L_{-4} + F_{-4}\sqrt{5}}{2} = \\ &= \frac{(L_3 + L_{-1} + L_{-4}) + (F_3 + F_{-1} + F_{-4})\sqrt{5}}{2} = \frac{(4-1+7) + (2+1-3)\sqrt{5}}{2} = \frac{10}{2}. \end{aligned}$$

Преобразуем «минимальную» форму ЗЦЗ (11.6) в «золотое» троичное представление. Чтобы это сделать, мы применим кодовое преобразование (11.5) одновременно ко всем нечетным разрядам ( $k = 2m+1$ ), значения которых равны 1. Ясно, что в ЗЦЗ (11.6) кодовое преобразование (11.5) можно применить только к 3-му и (-1)-му разрядам, значения которых равны 1. В ЗЦЗ (10.6) это преобразование показано стрелками. В результате мы получаем «золотое» троичное представление числа 5:

$$\begin{array}{cccccccc} & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \\ 5 = & 1 & 0 & \bar{1} & 0 & 1. & 0 & \bar{1} & 0 & 1. \end{array} \quad (11.7)$$

Нетрудно видеть из (11.7), что все разряды с нечетными индексами тождественно равны 0, а разряды с четными индексами принимают троичные

значения из множества  $\{\bar{1}, 0, 1\}$ . Это означает, что все разряды с нечетными индексами в представлении (11.7) являются «неинформативными», так как их значения всегда заведомо равны 0. Если теперь исключить из (11.7) все «неинформативные» разряды, мы получим следующее «золотое» троичное представление исходного числа  $N$ :

$$N = \sum_i c_{2i} \Phi^{2i}, \quad (11.8)$$

где  $c_{2i}$  – троичная цифра  $2i$ -го разряда.

Введем следующую перенумерацию разрядов «золотого» троичного представления (11.8). Каждая троичная цифра  $c_{2i}$  заменяется троичной цифрой  $b_i$ . В результате такой перенумерации выражение (11.8) может быть представлено в следующей форме:

$$N = \sum_i b_i \Phi^{2i}, \quad (11.9)$$

где  $b_i \in \{\bar{1}, 0, 1\}$  – троичная цифра,  $\Phi^{2i}$  – вес  $i$ -го разряда.

Сокращенная троичная цифровая запись числа  $N$  в системе счисления (11.9) имеет вид:

$$N = b_m b_{m-1} \dots b_2 b_1 b_0 \cdot b_{-1} b_{-2} \dots b_{-(m-1)} b_{-m}. \quad (11.10)$$

С учетом выражения (11.9) троичная ЗЦЗ (11.7) принимает следующий вид:

$$5 = 1 \begin{matrix} 2 & 1 & 0 & -1 & -2 \\ \bar{1} & 1 & \bar{1} & 1 & \bar{1} \end{matrix}. \quad (11.11)$$

Троичная ЗЦЗ (11.11) имеет следующую числовую интерпретацию:

$$\begin{aligned} 5 &= 1 \times \Phi^4 + \bar{1} \times \Phi^2 + 1 \times \Phi^0 + \bar{1} \times \Phi^{-2} + 1 \times \Phi^{-4} = \\ &= \frac{L_4 + F_4 \sqrt{5}}{2} + \frac{L_2 + F_2 \sqrt{5}}{2} + 1 + \frac{L_{-2} + F_{-2} \sqrt{5}}{2} + \frac{L_{-4} + F_{-4} \sqrt{5}}{2} = \\ &L_4 - L_2 + 1 = 7 - 3 + 1. \end{aligned} \quad (11.12)$$

Если над всеми нечетными разрядами минимальной формы числа  $N$ , задаваемой (11.2), выполнить кодовое преобразование (11.5), то может быть найден очень простой способ преобразования минимальной формы  $\Phi$ -кода (11.1) числа  $N$  в его троичное представление. Такое преобразование осуществляется с помощью простой комбинационной логической схемы, которая преобразует все тройки двоичных разрядов типа

$$a_{2i+1} a_{2i} a_{2i-1}, \quad (11.13)$$

взяты из минимальной формы  $\Phi$ -кода (11.1), в троичную цифру  $c_{2i} = b_i$  троичного кода (11.10) в соответствии с Табл. 11.1.

Таблица 11.1. Преобразование двоичного «золотого» кода в троичный «золотой» код

	$a_{2i+1}$	$a_{2i}$	$a_{2i-1}$	$b_i$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	1	0	0	$\bar{1}$
5	1	0	1	0

Заметим, что в Табл.11.1 используется только 5 двоичных кодовых комбинаций типа (11.13), поскольку исходная ЗЦЗ (11.2) является минимальной формой и поэтому кодовые комбинации 011,110,111 являются запрещенными.

Дадим пояснения преобразования двоичных троек (11.13) в троичную цифру  $b_i$ , представленным в Табл.11.1. Прежде всего, заметим, что значение цифры  $b_i$  совпадает со значением разряда  $a_{2i}$ , которое получается в результате применения преобразования (11.5) к двоичным разрядам  $a_{2i-1}, a_{2i+1}$ . В случае преобразований под номерами 1 и 3 разряды  $a_{2i-1} = a_{2i+1} = 0$  и к ним преобразование (11.5) не применяется; поэтому значение разряда  $b_i$  в случаях 1 и 3 совпадает со значением разряда  $a_{2i}$ . В случае 2 к разряду  $a_{2i-1} = 1$  применяется преобразование (11.5); при этом значение разряда  $a_{2i}$  становится равным 1, поэтому  $b_i = 1$ . В случае 4 к разряду  $a_{2i+1} = 1$  применяется преобразование (11.5); при этом значение разряда  $a_{2i}$  становится равным  $\bar{1}$ , поэтому  $b_i = \bar{1}$ . Наконец, в случае 5 преобразование (11.5) применяется к разрядам  $a_{2i-1} = a_{2i+1} = 1$  одновременно. При этом в результате преобразования (11.5) от разряда  $a_{2i-1} = 1$  в разряд  $a_{2i}$  поступает положительная 1, а от разряда  $a_{2i+1} = 1$  - отрицательная единица  $\bar{1}$ . В результате их суммирования получаем:  $1 + \bar{1} = 0$ , то есть, результирующее значение разряда  $a_{2i}$  остается без изменений, следовательно,  $b_i = 0$ .

На Рис.11.1 представлена комбинационная схема для преобразования «золотого» двоичного кода (11.2) в «золотой» троичный код (11.10).

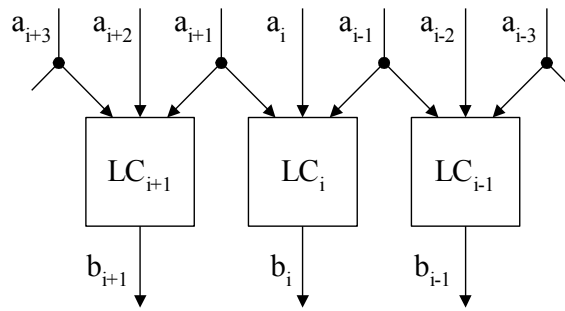


Рис.11.1. Преобразователь «золотого» двоичного кода в «золотой» троичный код

**Представление отрицательных чисел.** Подобно троичной симметричной системе счисления, рассмотренной в Главе 7, важное преимущество системы счисления (11.9) заключается в возможности представления как положительных, так и отрицательных чисел в «прямом» коде. Код отрицательного числа  $(-N)$  получается из троичной ЗЦЗ исходного числа  $N$  (11.10) посредством применения правила «троичной инверсии»:

$$1 \rightarrow \bar{1}, 0 \rightarrow 0, \bar{1} \rightarrow 1. \quad (11.14)$$

Применяя это правило к троичной ЗЦЗ (11.11), мы получим троичную ЗЦЗ отрицательного числа  $(-5)$ :

$$-5 = \bar{1} \ 1 \ 1. \ 1 \ \bar{1}.$$

**Свойство зеркальной симметрии.** Рассматривая троичную ЗЦЗ (11.11), мы можем установить, что левая часть  $(1\bar{1})$  троичной ЗЦЗ (11.11) зеркально симметрична по отношению к его правой части  $(\bar{1}1)$  относительно 0-го разряда.

Рассмотрим теперь другие примеры троичных ЗЦЗ натуральных чисел, приведенные в Табл.11.2. Из Табл.11.2 вытекает, что все троичные ЗЦЗ натуральных чисел обладают так называемым свойством «зеркальной симметрии», которое возникает при их представлении в «золотой» троичной системе (11.9). Но если теперь каждое натуральное число, представленное в системе счисления (11.9), преобразовать в соответствующее ему отрицательное число, используя правило (11.14), то свойство «зеркальной симметрии» при этом не будет нарушено. Отсюда вытекает, что все множество целых чисел (положительных и отрицательных) обладает свойством «зеркальной симметрии» при их представлении в «золотой» троичной системе счисления (11.9). Именно

поэтому «золотая» троичная система (11.9) названа в работе [83] троичной зеркально-симметричной системой счисления.

Таблица 11.2. Троичные золотые цифровые записи натуральных чисел

$i$	3	2	1	0	-1	-2	-3
$N$	$\Phi^6$	$\Phi^4$	$\Phi^2$	$\Phi^0$	$\Phi^{-2}$	$\Phi^{-4}$	$\Phi^{-6}$
0	0	0	0	0.	0	0	0
1	0	0	0	1.	0	0	0
2	0	0	1	$\bar{1}$ .	1	0	0
3	0	0	1	0.	1	0	0
4	0	0	1	1.	1	0	0
5	0	1	$\bar{1}$	1.	$\bar{1}$	1	0
6	0	1	0	$\bar{1}$ .	0	1	0
7	0	1	0	0.	0	1	0
8	0	1	0	1.	0	1	0
9	0	1	1	$\bar{1}$ .	1	1	0
10	0	1	1	0.	1	1	0

Отметим еще одно необычное свойство троичной зеркально-симметричной системы счисления (11.9). В традиционных позиционных системах счисления (десятичной, двоичной) увеличение числа сопровождается расширением цифровой записи числа в сторону старших разрядов. Из Табл.11.2 вытекает, что в троичной зеркально-симметричной системе счисления (11.9) расширение цифровой записи числа при его увеличении происходит симметрично, как в сторону старших, так и в сторону младших разрядов.

**Основание троичной зеркально - симметричной системы счисления.** Из выражения (11.9) вытекает, что основанием троичной зеркально-симметричной системы счисления (11.9) является квадрат золотой пропорции:

$$\Phi^2 = \frac{3 + \sqrt{5}}{2} \approx 2.618.$$

Это означает, что система счисления (11.9) относится к разряду систем счисления с иррациональными основаниями.

Любопытно, что основание системы счисления (11.9) имеет следующее традиционное представление:

$$\Phi^2 = 10.$$



## 11.2. Троичные зеркально-симметричные $F$ - и $L$ -коды

В Главе 9 мы ввели понятия  $F$ - и  $L$ -кодов (9.37), (9.40), с помощью которых можно представлять все натуральные числа; при этом  $F$ - и  $L$ -коды натурального числа  $N$  получаются из  $\Phi$ -кода (10.22) этого же натурального числа  $N$  путем простой замены всех степеней золотой пропорции  $\Phi^i$  на соответствующие числа Фибоначчи и Люка  $F_{i+1}$  и  $L_{i+1}$ .

Произведем теперь в троичном зеркально-симметричном представлении (11.9) следующие замены:

$$F_{2i+1} \rightarrow \Phi^{2i} \quad (11.15)$$

$$L_{2i+1} \rightarrow \Phi^{2i}. \quad (11.16)$$

В результате мы получим новые троичные зеркально-симметричные представления того же натурального числа  $N$ :

$$N = \sum_i b_i F_{2i+1} \quad (11.17)$$

$$N = \sum_i b_i L_{2i+1}. \quad (11.18)$$

Первое из этих представлений (11.17) мы будем называть троичным зеркально-симметричным  $F$ -кодом числа  $N$ , а второе (11.18) - троичным зеркально-симметричным  $L$ -кодом числа  $N$ .

В качестве примеров рассмотрим троичные зеркально-симметричные  $F$ - и  $L$ -коды числа 5,  $\Phi$ -код которого задается выражением (11.12). Если в выражении (11.12) произвести замены (11.15) и (11.16), то получим троичные зеркально-симметричные  $F$ - и  $L$ -коды числа 5:

$$5 = 1 \times F_5 + \bar{1} \times F_3 + 1 \times F_1 + \bar{1} \times F_{-1} + 1 \times F_{-3} = 5 - 2 + 1 - 1 + 2 \quad (11.19)$$

$$5 = 1 \times L_5 + \bar{1} \times L_3 + 1 \times L_1 + \bar{1} \times L_{-1} + 1 \times L_{-3} = 11 - 4 + 1 + 1 - 4. \quad (11.20)$$

**Сравнение чисел.** В троичной зеркально-симметричной системе счисления (11.9) сохраняется то же правило сравнения чисел по величине, которое характерно для всех позиционных представлений. Сравнение троичных ЗЦЗ чисел начинается со старших разрядов до момента получения несовпадающих разрядов. Большим считается то число, которое имеет большую цифру в первом из несовпадающих разрядов. Сравним, например, цифровые записи чисел 5 и 6 из Табл.11.2:

$i$	3	2	1	0	-1	-2	-3
5	0	1	$\bar{1}$	1.	$\bar{1}$	1	0
6	0	1	$\mathbf{0}$	$\bar{1}$	$\mathbf{0}$	1	0

При сравнении со стороны старших разрядов (слева направо) первым разрядом, в котором наблюдается несовпадение, является 1-й разряд. Но поскольку в цифровой записи чисел 5 и 6 в этом разряде соответственно записаны  $\bar{1}$  и 0 (выделены жирным шрифтом), то это означает, что  $6 > 5$ .

Заметим, что в соответствии со свойством зеркальной симметрии сравнение чисел в системе счисления (11.9) можно вести и с младших разрядов, двигаясь справа налево к старшим разрядам. Это обстоятельство может быть использовано для контроля операции сравнения зеркально-симметричных чисел по величине, когда сравнение чисел ведется одновременно со стороны старших и со стороны младших разрядов. Результаты обеих сравнений должны совпасть; в противном случае результат сравнения чисел является ошибочным.

В рассмотренном выше примере при сравнении со стороны младших разрядов первым разрядом, в котором наблюдается несовпадение, является (-1)-й разряд (значения разрядов выделены жирным шрифтом). Но поскольку в цифровой записи чисел 5 и 6 в этом разряде соответственно записаны  $\bar{1}$  и 0, соответственно, то это означает, что  $6 > 5$ , то есть, результат сравнения со стороны младших разрядов совпадает с результатом сравнения со стороны старших разрядов.

#### **Необычность троичной зеркально-симметричной системы счисления.**

А теперь обсудим полученные выше результаты. В результате проведенных выше рассуждений мы пришли к весьма необычной системе счисления. Прежде всего, эта система является троичной системой, использующей троичные цифры  $\{\bar{1}, 0, 1\}$ , и в ней в «прямом» коде могут быть представлены как положительные, так и отрицательные целые числа. Во-вторых, она имеет иррациональное основание  $\Phi^2 = \frac{3 + \sqrt{5}}{2}$ . Таким образом, эта система является своеобразным синтезом троичной симметричной системой счисления, использованной Николаем Брусенцовым при создании троичного компьютера «Сетунь» [107], и первой в

истории науки системы счисления с иррациональным основанием  $\Phi = \frac{1+\sqrt{5}}{2}$ , введенной в 1957 г. американским математиком Джорджем Бергманом [55].

Но наиболее неожиданным свойством такой системы является свойство зеркальной симметрии, возникающее при представлении чисел. Какое же практическое значение имеет обнаруженное выше свойство? Ясно, что это свойство является отличительной особенностью только для целых чисел (положительных и отрицательных) и мы можем использовать свойство зеркальной симметрии в компьютерах, если будем представлять целые числа в троичной «золотой» системе (11.9). Но наиболее важный практический результат состоит в том, что это свойство, как будет показано ниже, может быть использовано в компьютерах для контроля арифметических операций.

### 11.3. Диапазон представления и избыточность троичного зеркально-симметричного представления

**Диапазон представления чисел.** Исследуем вопрос о диапазоне представления целых чисел в троичной зеркально-симметричной системе счисления (11.9). Рассмотрим троичную ЗЦЗ (11.10), состоящую из  $2m+1$  троичных разрядов. Ясно, что с помощью такой ЗЦЗ можно представить все целые числа в диапазоне от

$$N_{\max} = \underset{m}{11\dots11}\underset{m}{11\dots11}1 \quad (11.21)$$

до

$$N_{\min} = \bar{\underset{m}{11\dots11}}\bar{\underset{m}{11\dots11}}\bar{1}. \quad (11.22)$$

Ясно, что минимальное число (11.22) может быть получено из максимального числа (11.21) с использованием правила троичной инверсии (11.14), то есть, числа  $N_{\min}$  и  $N_{\max}$  равны по абсолютной величине:

$$|N_{\min}| = N_{\max}.$$

Из этих рассуждений вытекает, что с помощью  $(2m+1)$ -разрядной троичной ЗЦЗ (11.10) можно представить

$$2N_{\max} + 1 \quad (11.23)$$

целых чисел, включая число 0.

Для вычисления  $N_{\max}$  воспользуемся понятием троичного зеркально-симметричного  $L$ -кода, задаваемого выражением (11.18). Тогда с учетом (11.16) мы можем интерпретировать троичную ЗЦЗ (11.21) как следующую сумму:

$$N_{\max} = L_{2m+1} + L_{2m-1} + \dots + L_3 + L_1 + L_{-1} + L_{-3} + \dots + L_{-(2m-1)}. \quad (11.24)$$

Известно (см. Главу 2), что для нечетных индексов  $i = 2k - 1$  числа Люка связаны следующим соотношением:

$$L_{2m-1} = -L_{-(2m-1)}. \quad (11.25)$$

Учитывая свойство (11.25), легко усмотреть, что все числа Люка в сумме справа от наибольшего числа Люка  $L_{2m+1}$

$$L_{2m-1} + \dots + L_3 + L_1 + L_{-1} + L_{-3} + \dots + L_{-(2m-1)}$$

согласно (11.25) взаимно уничтожаются, так как

$$L_{2m-1} + L_{-(2m-1)} = 0; \dots; L_3 + L_{-3} = 0; L_1 + L_{-1} = 0.$$

Следовательно, значение суммы (11.24) равно:

$$L_{\max} = L_{2m+1}. \quad (11.26)$$

Тогда с учетом (11.23) результаты проведенных рассуждений можно сформулировать в виде следующей теоремы.

**Теорема 11.1.** С помощью  $(2m+1)$ -разрядного троичного зеркально-симметричного кода (11.9) можно представить  $2L_{2m+1} + 1$  целых чисел в диапазоне от  $(-L_{2m+1})$  до  $(+L_{2m+1})$ , где  $L_{2m+1}$  – число Люка.

### **Избыточность троичной зеркально-симметричной системы счисления.**

Сравним теперь систему счисления (11.9) с троичной симметричной системой счисления. Как показано в Главе 7, с помощью  $n$  разрядов в троичной симметричной системе счисления (7.5) можно представить  $3^n$  целых чисел в

$$\text{диапазоне от } N_{\min} = -\frac{3^n - 1}{2} \text{ до } N_{\max} = \frac{3^n - 1}{2}.$$

Для представления такого диапазона чисел, задаваемого Теоремой 1.1, в троичной симметричной системе счисления (7.5), должно быть выполнено следующее неравенство:

$$3^n \geq 2L_{2m+1} + 1 \approx 2L_{2m+1}. \quad (11.27)$$

В «теории чисел Фибоначчи» [2-4] известна формула Бине, выражающая числа Люка через степени золотой пропорции; для достаточно больших  $m$  мы можем записать следующее приближенное равенство для числа Люка  $L_{2m+1}$ :

$$L_{2m+1} \approx \Phi^{2m+1}. \quad (11.28)$$

Подставляя (11.28) в (11.27) и беря логарифм по основанию 3 от обеих частей неравенства (11.27), мы получим следующее неравенство:

$$n \geq (2m+1)\log_3 \Phi + \log_3 2. \quad (11.29)$$

С увеличением  $m$  неравенство (11.29) превращается в следующее приближенное равенство:

$$n \approx (2m+1)\log_3 \Phi. \quad (11.30)$$

Равенство (11.31) может быть использовано для вычисления избыточности троичного зеркально-симметричного представления (11.9). Относительная избыточность  $R$  может быть вычислена по следующей формуле:

$$R = \frac{k-n}{n} = \frac{k}{n} - 1,$$

где  $k$  и  $n$  – число разрядов соответственно избыточной и неизбыточной системы счисления, используемой для представления одного и того же диапазона чисел.

Для представления чисел в избыточной троичной зеркально-симметричной системе счисления мы использовать  $k = 2m + 1$  троичных разрядов, в то время как для представления этого же диапазона чисел в неизбыточной троичной симметричной системе счисления (7.5) мы будем использовать  $n$  троичных разрядов. Тогда из этих рассуждений вытекает следующее выражение для вычисления относительной избыточности троичного зеркально-симметричного представления (11.9):

$$R = \frac{1 - \log_3 \Phi}{\log_3 \Phi} = 1,283 = 128,3\%.$$

Таким образом, относительная избыточность троичной зеркально-симметричной системы счисления (11.9) достаточно велика, то есть, ее применение потребует увеличение разрядности представления более чем на 100% по сравнению с троичной симметричной системой счисления (7.5), с использованием которой был разработан первый в компьютерной истории троичный компьютер «Сетунь». Однако, этот недостаток компенсируется рядом интересных возможностей системы счисления (11.9), о которых мы расскажем ниже.

## 11.4. Троичное зеркально-симметричное сложение и вычитание

Таблица троичного зеркально-симметричного сложения. В основе зеркально-симметричного сложения лежат следующие тождества для степеней золотой пропорции:

$$2\Phi^{2k} = \Phi^{2(k+1)} - \Phi^{2k} + \Phi^{2(k-1)} \quad (11.31)$$

$$3\Phi^{2k} = \Phi^{2(k+1)} + 0 + \Phi^{2(k-1)} \quad (11.32)$$

$$4\Phi^{2k} = \Phi^{2(k+1)} + \Phi^{2k} + \Phi^{2(k-1)}, \quad (11.33)$$

где  $k = 0, \pm 1, \pm 2, \pm 3, \dots$

Тождество (11.31) легко доказывается, если над его правой частью выполнить следующие преобразования:

$$\Phi^{2(k+1)} - \Phi^{2k} + \Phi^{2(k-1)} = \Phi^{2k+1} + \Phi^{2k} - \Phi^{2k} + \Phi^{2(k-1)} = \Phi^{2k} + \Phi^{2k-1} + \Phi^{2k-2} = \Phi^{2k} + \Phi^{2k} = 2\Phi^{2k}$$

Тождества (11.32) и (11.33) являются следствием (11.31).

Тождество (11.31) является математической основой для зеркально-симметричного сложения двух одноразрядных троичных цифр и задает правило формирования переноса при сложении (Табл.11.3).

Таблица 11.3. Правило зеркально-симметричного сложения одноразрядных чисел

$b_k + a_k$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}\bar{1}$	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$\bar{1}\bar{1}$

Главной особенностью Табл.11.3 является правило сложения двух единиц с одинаковыми знаками, то есть:

$$\begin{aligned} a_k + b_k &= c_k \quad s_k \quad c_k \\ 1 + 1 &= 1 \quad \bar{1} \quad 1 \\ \bar{1} + \bar{1} &= \bar{1} \quad 1 \quad \bar{1} \end{aligned}$$

Таким образом, при зеркально-симметричном сложении троичных единиц одного знака возникает промежуточная сумма  $s_k$ , равная 1, взятой с противоположным знаком, и перенос  $c_k$ , равный 1, взятой с тем же знаком, что и слагаемые. Особенность состоит в том, что перенос из  $k$ -го разряда распространяется

одновременно в два соседние разряда, а именно, в соседний слева, то есть, в  $(k+1)$ -й разряд, и соседний справа, то есть в  $(k-1)$ -й разряд.

### Одноразрядные троичные зеркально-симметричные сумматоры.

Таблица 11.3 описывает функционирование простейшего троичного сумматора, называемого одноразрядным полусумматором. Последний представляет собой комбинационную логическую схему (Рис.11.2-а), имеющую два троичных входа  $a_k$  и  $b_k$  и два троичных выхода  $s_k$  и  $c_k$  и функционирующую в соответствии с Табл.11.3.

Так как перенос из  $k$ -го разряда сумматора распространяется в сторону соседних левого и правого разрядов, то это означает, что полный зеркально - симметрический одноразрядный сумматор должен иметь два входа для переносов, поступающих из  $(k-1)$ -го и  $(k+1)$ -го разрядов в  $k$ -й разряд. Таким образом, полный троичный зеркально-симметричный одноразрядный сумматор представляет собой комбинационную логическую схему, имеющую четыре троичных входа и два троичных выхода (Рис.11.2-б). Обозначим через  $2\Sigma$  троичный зеркально-симметричный одноразрядный полусумматор с двумя входами и через  $4\Sigma$  троичный зеркально-симметричный одноразрядный полный сумматор с четырьмя входами.

Опишем логическое действие троичного зеркально-симметричного полного одноразрядного сумматора  $4\Sigma$ . Заметим, прежде всего, что число всех возможных 4-разрядных троичных входных комбинаций троичного зеркально-симметрического полного сумматора равно  $3^4 = 81$ . Значения выходных переменных  $s_k$  и  $c_k$  являются некоторыми дискретными функциями алгебраической суммы  $S$  входных троичных переменных  $a_k, b_k, c_{k-1}, c_{k+1}$ , то есть:

$$S = a_k + b_k + c_{k-1} + c_{k+1}. \quad (11.34)$$

Сумма (11.34) принимает значения из множества:  $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ .

Правило функционирования троичного зеркально-симметрического полного сумматора  $4\Sigma$  состоит в следующем. Сумматор формирует выходную троичную кодовую комбинацию  $c_k s_k$  в соответствии со значением суммы (11.34):

$$-4 = \bar{1}\bar{1}; -3 = \bar{1}0; -2 = \bar{1}\bar{1}; -1 = 0\bar{1}; 0 = 00; 1 = 01; 2 = \bar{1}\bar{1}; 3 = 10; 4 = 11. \quad (11.35)$$

Младшая цифра 2-разрядного троичного представления (11.35) представляет собой значение промежуточной суммы  $s_k$ , а старшая цифра - значение переноса

$c_k$ , который распространяется в соседние разряды (справа и слева от  $k$ -го разряда).

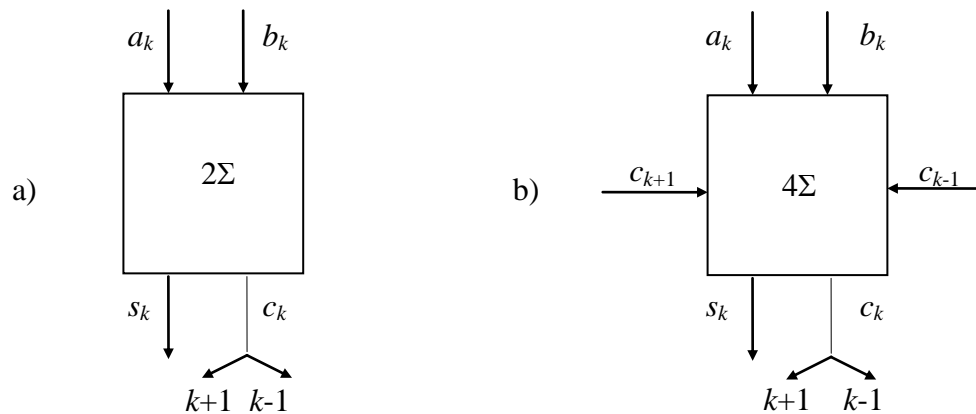


Рис.11.2. Троичные зеркально-симметричные одноразрядные сумматоры:  
(а) полусумматор; (б) полный сумматор

**Многоразрядный троичный зеркально-симметричный сумматор.** Многоразрядный троичный зеркально-симметричный сумматор, реализующий сложение двух  $(2m+1)$ -разрядных зеркально-симметричных чисел, представляет собой комбинационную логическую схему, состоящую из  $(2m+1)$  троичных зеркально-симметричных полных одноразрядных сумматоров типа  $4\Sigma$  (Рис.11.3).

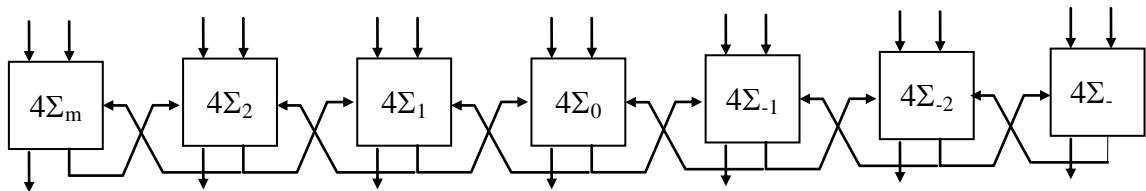


Рис.11.3. Многоразрядный троичный зеркально-симметричный сумматор

В качестве примера рассмотрим сложение двух чисел  $5 + 10$  в троичной зеркально-симметричной системе счисления:

$$\begin{array}{r}
 5 = 0 \ 1 \ \bar{1} \ 1, \ \bar{1} \ 1 \ 0 \\
 10 = 0 \ 1 \ 1 \ 0, \ 1 \ 1 \ 0 \\
 \quad 0 \ 1 \ 0 \ 1, \ 0 \ 1 \ 0 \\
 \quad 1 \leftrightarrow 1 \quad 1 \leftrightarrow 1 \\
 \hline
 15 = 1 \ \bar{1} \ 1 \ 1, \ 1 \ \bar{1} \ 1
 \end{array}$$



Заметим, что стрелка  $\leftrightarrow$  обозначает процесс распространения переноса (влево и вправо).

Важно подчеркнуть, что результат сложения (число 15) представлен в зеркально-симметричной форме!

Как было отмечено выше, важное преимущество троичной зеркально-симметричной системы счисления состоит в возможности суммирования всех целых чисел (положительных и отрицательных) в «прямом» коде, то есть, без использования понятий инверсного и дополнительного кодов. В качестве примера рассмотрим сложение отрицательного числа (-24) с положительным числом 15:

$$\begin{array}{rcccccc}
 -24 = & \bar{1} & \bar{1} & 0 & 1, & 0 & \bar{1} & \bar{1} \\
 15 = & 1 & \bar{1} & 1 & 1, & 1 & \bar{1} & 1 \\
 & & 0 & 1 & 1 & \bar{1}, & 1 & 1 & 0 \\
 & & & \downarrow & 1 & \leftrightarrow & 1 & \downarrow \\
 & & \bar{1} & \leftrightarrow & \bar{1} & & \bar{1} & \leftrightarrow & \bar{1} \\
 -9 = & \bar{1} & 1 & 1 & \bar{1}, & 1 & 1 & \bar{1}
 \end{array}$$

Важно подчеркнуть, что результат сложения (отрицательное число -9) представлен в зеркально-симметричной форме!

**Троичное зеркально-симметричное вычитание.** Вычитание двух троичных зеркально-симметричных чисел  $N_1 - N_2$  сводится к троичному зеркально-симметричному сложению, если мы представим их разность в следующем виде:

$$N_1 - N_2 = N_1 + (-N_2). \quad (11.36)$$

Как следует из (11.36), перед вычитанием необходимо взять троичную инверсию от вычитаемого  $N_2$  и затем выполнить троичное зеркально-симметричное сложение.

**Счет и вычитание единиц.** Заметим, что изображенный на Рис.11.3 многоразрядный зеркально-симметричный сумматор является базовым элементом троичного зеркально-симметричного процессора и выполняет при этом также функции троичного зеркально-симметричного счетчика.

При счете единиц на вход сумматора 0-го разряда поступают «положительные» или «отрицательные» единицы, которые суммируются с предыдущим числом, записанным в сумматоре, например:

$$\begin{array}{r}
0 = 000.00 \\
\underline{\quad 1 \quad} \\
1 = 001.00 \\
\underline{\quad 1 \quad} \\
2 = 01\bar{1}.10 \\
\underline{\quad 1 \quad} \\
3 = 010.10 \\
\underline{\quad 1 \quad} \\
4 = 011.10
\end{array}$$

**Явление «качелей».** Просуммируем два равных числа  $5+5$ , представленных в троичной зеркально-симметричной системе счисления:

$$\begin{array}{r}
5 = 0 \ 1 \ \bar{1} \ 1, \ \bar{1} \ 1 \ 0 \\
5 = 0 \ 1 \ \bar{1} \ 1, \ \bar{1} \ 1 \ 0 \\
\hline
0 \ \bar{1} \ 1 \ \bar{1}, \ 1 \ \bar{1} \ 0 \\
\quad \downarrow \ 1 \ \leftrightarrow \ 1 \ \downarrow \\
1 \ \leftrightarrow \ 1 \quad \quad 1 \ \leftrightarrow \ 1 \\
\quad \bar{1} \ \leftrightarrow \ \bar{1} \ \downarrow \\
\quad \quad \quad \bar{1} \ \leftrightarrow \ \bar{1} \\
\hline
1 \ 1 \ 0 \ 0, \ 0 \ 1 \ 1 \\
\quad \quad \bar{1} \ \leftrightarrow \ \bar{1} \\
\quad \quad 1 \ \leftrightarrow \ 1 \ \downarrow \\
\quad \downarrow \quad \quad 1 \ \leftrightarrow \ 1 \\
\bar{1} \ \leftrightarrow \ \bar{1} \quad \quad \bar{1} \ \leftrightarrow \ \bar{1} \\
\hline
0 \ \bar{1} \ 1 \ \bar{1}, \ 1 \ \bar{1} \ 0 \\
\quad \quad \quad 1 \ \leftrightarrow \ 1 \\
\quad \quad \bar{1} \ \leftrightarrow \ \bar{1} \ \downarrow \\
\quad \downarrow \quad \quad \bar{1} \ \leftrightarrow \ \bar{1} \\
1 \ \leftrightarrow \ 1 \quad \quad 1 \ \leftrightarrow \ 1
\end{array}$$

Из этого примера вытекает, что мы имеем специальный случай сложения, называемый «качелями». Если процесс суммирования продолжить, то, начиная с некоторого шага, процесс формирования переноса оказывается периодически повторяющимся и, следовательно, сложение становится бесконечным. Явление «качелей» является разновидностью «гонок», возникающих в цифровых автоматах, когда элементы начинают переключаться.

Чтобы исключить явление «качелей», можно использовать следующий эффективный "технический" прием. Задержим входные сигналы одноразрядных сумматоров с нечетными индексами ( $k = \pm 1, \pm 2, \pm 3, \dots$ ) на один такт. С учетом этого сумматоры с четными индексами ( $k = 0, \pm 2, \pm 4, \dots$ ) формируют на первом

также сложения промежуточные суммы и переносы в одноразрядные сумматоры с нечетными индексами. На втором такте сложения переносы, сформированные на первом шаге, суммируются с соответствующими троичными переменными всех нечетных разрядов слагаемых чисел. Благодаря такому подходу явление «качелей» автоматически устраняется.

Продемонстрируем изложенный выше метод на примере сложения 5+5:

$$\begin{array}{r}
 5 = 0 \ 1 \ \bar{1} \ 1, \ \bar{1} \ 1 \ 0 \\
 5 = 0 \ 1 \ \bar{1} \ 1, \ \bar{1} \ 1 \ 0 \\
 \hline
 0 \ \bar{1} \ \bar{1} \ \bar{1}, \ \bar{1} \ \bar{1} \ 0 \\
 0 \ \downarrow \ \bar{1} \ \downarrow \ \bar{1} \ \downarrow \ 0 \\
 \quad \downarrow \ 1 \ \leftrightarrow \ 1 \ \downarrow \\
 \quad \quad \quad 1 \ \leftrightarrow \ 1 \quad \quad 1 \ \leftrightarrow \ 1 \\
 \hline
 10 = 1 \ \bar{1} \ 0 \ \bar{1}, \ 0 \ \bar{1} \ 1
 \end{array}$$

Первый такт зеркально-симметричного сложения состоит в формировании переносов от всех разрядов с четными индексами  $(0, 2, -2)$ . Сумматоры всех разрядов с нечетными индексами  $(1, 3, -1, -3)$  не функционируют на первом такте. Второй такт состоит в сложении переносов, возникших на первом такте, с троичными переменными разрядов с нечетными индексами.

**«Дублирование» троичного зеркально-симметричного сумматора.** Заметим, что сумматор на Рис.11.3 состоит из двух зеркально-симметричных частей относительно 0-го одноразрядного сумматора  $4\Sigma_0$ . При этом левая часть сумматора на Рис.11.3 как бы несет основную числовую нагрузку, в то время как его правая часть используется для контроля результата согласно свойству «зеркальной симметрии». Существует возможность уменьшения «структурной избыточности» сумматора на Рис.11.3, если использовать «принцип дублирования», как показано на Рис.11.4.

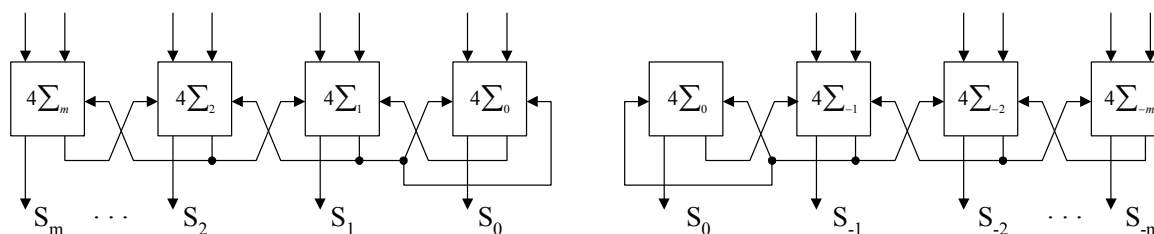


Рис.11.4. Использование «принципа дублирования» для троичного зеркально-симметричного сумматора

Троичный зеркально-симметричный сумматор на Рис.11.4 разбит на две независимые части. Каждая часть имеет собственный сумматор типа  $4\Sigma_0$ . При этом выход переноса одноразрядного сумматора  $4\Sigma_1$  левой части соединен с двумя входами одноразрядного сумматора  $4\Sigma_0$ , в то время как выход переноса одноразрядного сумматора  $4\Sigma_{-1}$  соединен с двумя входами своего одноразрядного сумматора  $4\Sigma_0$ . Это обеспечивает корректное зеркально-симметричное сложение в правой и левой части сумматора на Рис.11.4.

Это означает, что в случае отказа одной из частей сумматора на Рис.11.4 мы можем использовать другую часть сумматора для вычислений. На заключительной стадии вычислений мы можем восстановить конечное троичное зеркально-симметричное представление, пользуясь свойством «зеркальной симметрии». Это означает, что сумматор на Рис.11.4 по существу является отказоустойчивым сумматором. Таким образом, теоретически отказ левой или правой частей сумматора на Рис.11.4 не оказывает влияния на результат вычисления, поскольку вся информация может быть восстановлена согласно принципу «зеркальной симметрии».

**Преимущества троичного зеркально-симметричного сложения-вычитания.** Мы можем подчеркнуть ряд важных преимуществ троичного зеркально-симметричного сложения-вычитания с «технической» точки зрения:

1. Троичное зеркально-симметричное вычитание сводится к троичному зеркально-симметричному сложению путем использования правила троичной инверсии (11.14).
2. Троичное зеркально-симметричное сложение и вычитание выполняются в «прямом» коде, то есть, без использования понятий «инверсного» и «дополнительного» кодов.
4. Знак результата арифметической операции определяется автоматически, поскольку он совпадает со знаком старшего значащего разряда результата.
5. Результат сложения-вычитания всегда представляется в зеркально-симметричной форме, что позволяет контролировать процесс зеркально-симметричного сложения-вычитания.

## 11.5. Троичное зеркально-симметричное умножение и деление

**Троичное зеркально-симметричное умножение.** В основе троичного зеркально-симметричного умножения лежит следующее тривиальное тождество, связывающее степени золотой пропорции:

$$\Phi^{2n} \times \Phi^{2m} = \Phi^{2(n+m)}. \quad (11.37)$$

Правило троичного зеркально-симметричного умножения задается Табл.11.4.

Таблица 11.4. Таблица троичного зеркально-симметричного умножения

$b_k \times a_k$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	0	$\bar{1}$
0	0	0	0
1	$\bar{1}$	0	1

Умножение выполняется в «прямом» коде. Общий алгоритм умножения двух троичных зеркально-симметричных чисел сводится к формированию частичных произведений в соответствии с Табл.11.4 и их сложению в соответствии с правилом троичного зеркально-симметричного сложения. В качестве примера умножим отрицательное число ( $-6 = \bar{1}01.0\bar{1}$ ) на положительное число ( $2 = 1\bar{1}.1$ ):

$$\begin{array}{r}
 \bar{1} \ 0 \ 1, \ 0 \ \bar{1} \\
 \phantom{\bar{1} \ 0 \ 1, \ 0 \ \bar{1}} 1 \ \bar{1}, \ 1 \\
 \hline
 \bar{1} \ 0, \ 1 \ 0 \ \bar{1} \\
 \phantom{\bar{1} \ 0, \ 1 \ 0 \ \bar{1}} 1 \ 0 \ \bar{1}, \ 0 \ 1 \\
 \bar{1} \ 0 \ 1 \ 0, \ \bar{1} \\
 \hline
 \bar{1} \ 1 \ 0 \ \bar{1}, \ 0 \ 1 \ \bar{1}
 \end{array}$$

Результат умножения в рассматриваемом примере формируется как сумма трех частных произведений. Первое частное произведение ( $\bar{1}0.10\bar{1}$ ) есть результат умножения троичного зеркально-симметричного числа ( $-6 = \bar{1}01.0\bar{1}$ ) на младшую положительную единицу троичного зеркально-симметричного числа ( $2 = 1\bar{1}.1$ ), второе частное произведение ( $10.\bar{1}01$ ) есть результат умножения того

же самого числа  $(-6 = \bar{1}01.0\bar{1})$  на среднюю отрицательную единицу числа  $(2 = 1\bar{1}.1)$  и, наконец, третье частное произведение  $(\bar{1}010.\bar{1})$  есть результат умножения того же самого числа  $(-6 = \bar{1}01.0\bar{1})$  на старшую положительную единицу числа  $(2 = 1\bar{1}.1)$ .

Заметим что произведение  $(-12 = \bar{1}10\bar{1}.01\bar{1})$  сохраняет свойство «зеркальной симметрии». Так как старшая цифра произведения является отрицательной единицей, отсюда вытекает, что произведение является отрицательным числом.

**Троичное зеркально-симметричное деление.** Троичное зеркально-симметричное деление выполняется в соответствии с правилами деления в классической троичной симметричной системе счисления (7.5). Общий алгоритм троичного зеркально-симметричного деления сводится к последовательному вычитанию из делимого сдвинутого делителя, умноженного на очередную троичную цифру промежуточного частного.

Продemonстрируем троичное зеркально-симметричное деление на примере деления троичного зеркально-симметричного числа  $(24 = 110\bar{1}.011)$  на троичное зеркально-симметричное число  $(2 = 1\bar{1}.1)$ .

#### Первый такт

Первый такт деления состоит в сдвиге делителя  $(2 = 1\bar{1}.1)$  на два разряда влево. В результате мы получим сдвинутый делитель в виде  $(\bar{1}\bar{1}10.000)$ . Сравнивая цифровую запись сдвинутого делителя с делимым, то есть, цифровые записи  $(24 = 110\bar{1}.011)$  и  $(\bar{1}\bar{1}10.000)$ , мы устанавливаем, что знаки сравниваемых цифровых записей совпадают. В этом случае мы записываем положительную троичную единицу 1 в 3-й разряд частного.

Затем мы должны вычесть сдвинутый делитель из делимого. Как установлено выше, вычитание двух троичных зеркально-симметричных чисел сводится к троичному зеркально-симметричному сложению путем троичной инверсии вычитаемого. Следовательно, на первом шаге мы должны сложить два

«золотых» зеркально-симметричных числа, а именно, делимое и троичную инверсию сдвинутого делителя, то есть,

$$\begin{array}{r|l} 1 \ 1 \ 0 \ \bar{1}, \ 0 \ 1 \ 1 & 1 \ \bar{1}, \ 1 \\ \bar{1} \ 1 \ \bar{1} \ 0, \ 0 \ 0 \ 0 & \hline 0 \ \bar{1} \ \bar{1} \ \bar{1}, \ 0 \ 1 \ 1 & 1 \ 0 \ 0, \ 0 = Q_1 \\ 1 \leftrightarrow 1 & \\ \hline D_1 = 1 \ \bar{1} \ 0 \ \bar{1}, \ 0 \ 1 \ 1 & \end{array}$$

Следовательно, на первом такте деления мы получили первое промежуточное частное

$$Q_1 = 100.0$$

и первый промежуточное делимое

$$D_1 = 1\bar{1}00\bar{1},011.$$

#### Второй такт

Второй такт деления является повторением первого такта, в котором в качестве делимого используется число  $(D_1 = 1\bar{1}00\bar{1},011)$ , полученное на предыдущем такте. Заметим, что число  $D_1$  содержит положительную единицу в том же разряде, что и исходное делимое. Это означает, что делитель  $(2 = 1\bar{1}.1)$  должен быть сдвинут на 2 разряда влево и после этого мы должны сравнить сдвинутый делитель и первый промежуточное делимое, то есть, цифровые записи  $(D_1 = 1\bar{1}00\bar{1},011)$  и  $(1\bar{1}10.000)$ . Так как знаки сравниваемых чисел совпадают, то это означает, что мы должны сформировать на втором шаге второе промежуточное частное

$$Q_2 = 100.0$$

и осуществить сложение числа  $(D_1 = 1\bar{1}00\bar{1},011)$  с троичной инверсией сдвинутого делителя. Следовательно, второй такт деления состоит в следующем:

$$\begin{array}{r|l} 1 \ \bar{1} \ 0 \ \bar{1}, \ 0 \ 1 \ 1 & 1 \ \bar{1}, \ 1 \\ \bar{1} \ 1 \ \bar{1} \ 0, \ 0 \ 0 \ 0 & \hline D_2 = 0 \ 0 \ \bar{1} \ \bar{1}, \ 0 \ 1 \ 1 & 1 \ 0 \ 0, \ 0 = Q_2 \end{array}$$

Таким образом, в результате второго такта мы получили второе промежуточное частное

$$Q_2 = 100.0$$

и второе промежуточное делимое

$$D_2 = 00\bar{1}\bar{1}.011.$$

### Третий такт

Третий такт деления состоит в следующем. Сравнивая отрицательное число  $(D_2 = 00\bar{1}\bar{1}.011)$  (отрицательное число) с делителем  $(2 = 1\bar{1}.1)$  (положительное число), мы должны сформировать третье промежуточное частное в виде

$$Q_3 = \bar{1}.0$$

и затем вычесть делитель  $(2 = 1\bar{1}.1)$ , умноженный на отрицательную единицу  $\bar{1}$ , то есть, число  $(-2 = \bar{1}\bar{1}.1)$  из числа  $(D_2 = 00\bar{1}\bar{1}.011)$ . Учитывая, что вычитание сводится к сложению троичной инверсии числа  $(-2 = \bar{1}\bar{1}.1)$ , это означает, что в случае получения отрицательного частного на предыдущем шаге деление на этом такте сводится к сложению делителя с предыдущим промежуточным делимым  $(D_2 = 00\bar{1}\bar{1}.011)$ , то есть,

$$\begin{array}{r} \bar{1} \quad \bar{1}. \quad 0 \quad 1 \quad 1 \quad | \quad 1 \quad \bar{1}, \quad 1 \\ 1 \quad \bar{1}. \quad 1 \quad \quad \quad | \quad 1, \quad 0 = Q_3 \\ \hline 0 \quad 1. \quad 1 \quad 1 \quad 1 \\ \bar{1} \quad \leftrightarrow \quad \bar{1} \\ \hline D_3 = \bar{1} \quad 1. \quad 0 \quad 1 \quad 1 \end{array}$$

### Четвертый такт

$$\begin{array}{r} \bar{1} \quad 1, \quad 0 \quad 1 \quad 1 \quad | \quad 1 \quad \bar{1}, \quad 1 \\ 1 \quad \bar{1}, \quad 1 \quad \quad \quad | \quad 1, \quad 0 = Q_4 \\ \hline D_4 = 0 \quad 0, \quad 1 \quad 1 \quad 1 \end{array}$$

### Пятый такт

$$\begin{array}{r} 0, \quad 1 \quad 1 \quad 1 \quad | \quad 1 \quad \bar{1}, \quad 1 \\ 0, \quad \bar{1} \quad 1 \quad \bar{1} \quad | \quad 0, \quad 0 \quad 1 = Q_5 \\ \hline 0, \quad 0 \quad \bar{1} \quad 0 \\ 1 \quad \leftrightarrow \quad 1 \\ \hline D_5 = 0, \quad 1 \quad \bar{1} \quad 1 \end{array}$$



Шестой такт

$$\begin{array}{r|l} 0, 1 \bar{1} 1 & 1 \bar{1}, 1 \\ 0, \bar{1} 1 \bar{1} & 0, 0 1 = Q_6 \\ \hline 0, 0 0 0 & \end{array}$$

Деление закончено, так как промежуточное делимое на 6-м шаге стало равным 0. В общем случае деление продолжается либо до получения нулевого делимого, либо до получения промежуточного делимого, меньшего делителя. Частное от деления формируется как результат сложения всех промежуточных частных, то есть,

$$\begin{array}{r} 1 \quad 0 \quad 0, \quad 0 \\ 1 \quad 0 \quad 0, \quad 0 \\ \quad \quad \quad \bar{1}, \quad 0 \\ \quad \quad \quad \bar{1}, \quad 0 \\ \quad \quad \quad 0, \quad 0 \quad 1 \\ \quad \quad \quad 0, \quad 0 \quad 1 \\ \hline \bar{1} \quad 0 \quad 1, \quad 0 \quad \bar{1} \\ \downarrow \quad \bar{1} \leftrightarrow \bar{1} \quad \downarrow \\ 1 \leftrightarrow 1 \quad \quad 1 \leftrightarrow 1 \\ \hline 1 \quad \bar{1} \quad 0 \quad 1, \quad 0 \quad \bar{1} \quad 1 \end{array}$$

Заметим, что результат деления  $(12 = 1\bar{1}01.0\bar{1}1)$  оказался представленным в зеркально-симметричной форме! Это означает, что все арифметические операции в троичной зеркально-симметричной арифметике являются контролируруемыми по принципу «зеркальной симметрии».

**Обсуждение полученных результатов.** Мы получили весьма любопытную троичную систему счисления, которая обладает необычными свойствами. Во-первых, все арифметические операции в этой системе счисления выполняются в "прямом" коде! И мы можем забыть об инверсном и дополнительном кодах для представления отрицательных чисел!

Во-вторых, результаты всех арифметических операций контролируются согласно свойству «зеркальной симметрии»!

## 11.6. Матричный и конвейерный зеркально-симметричные сумматоры

**Матричный троичный зеркально-симметричный сумматор.** Известно, что цифровая обработка сигналов выдвигает высокие требования к быстродействию арифметических устройств. Для этой цели разработаны различные специальные структуры (матричные, конвейерные и т.д.). Покажем, что троичная зеркально-симметричная арифметика содержит в себе интересные возможности для реализации быстрых арифметических процессоров для цифровой обработки сигналов.

На Рис.11.5 представлен матричный многоразрядный троичный зеркально-симметричный сумматор.

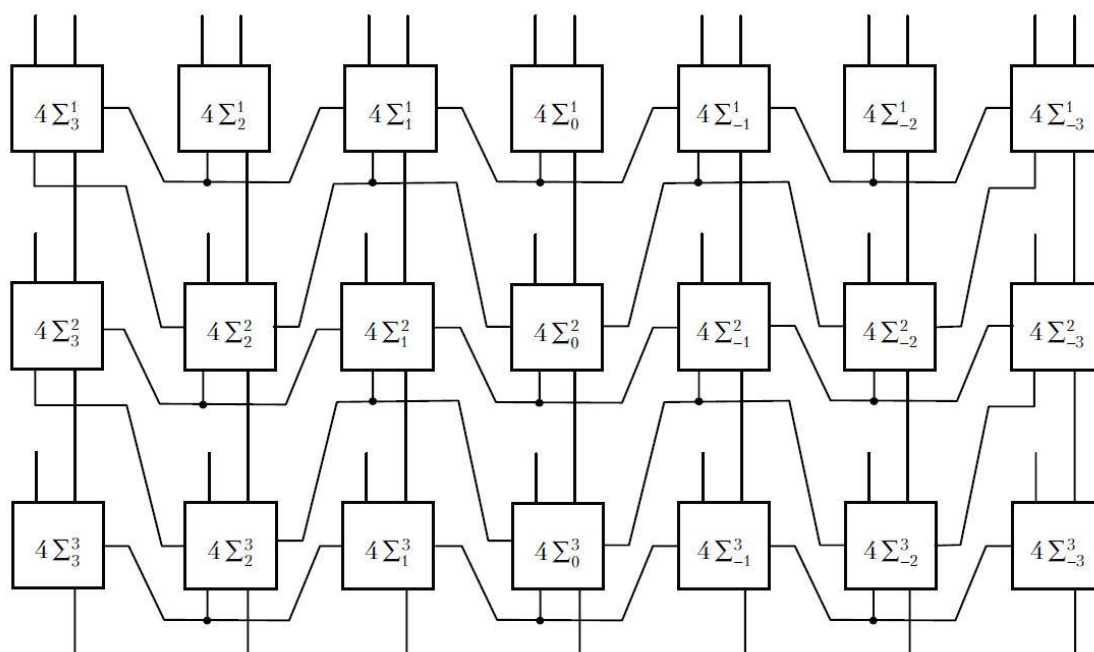


Рис.11.5. Матричный троичный зеркально-симметричный сумматор

Каждая ячейка матричного сумматора на Рис.11.5 является одноразрядным зеркально-симметричным сумматором типа  $4\Sigma$ , имеющим 4 троичных входа и 2 троичных выхода (см. Рис.11.1-b). Матричный сумматор на Рис.11.5 состоит из 21 одноразрядных сумматоров, расположенных в форме матрицы размером  $7 \times 3$ , то есть, сумматор предназначен для обработки 7-разрядного троичного зеркально-симметричного кода, а сам матричный сумматор состоит из трех рядов 7-разрядных сумматоров. Каждый одноразрядный сумматор имеет обозначение вида  $4\Sigma_i^k$ , где число 4 перед символом суммы означает, что сумматор имеет 4 троичных входа, нижний индекс  $i$  знака суммы означает, что сумматор относится

к  $i$ -му разряду троичного зеркально-симметричного кода (11.9), а верхний индекс  $k$  означает номер ряда матричного сумматора на Рис.11.5.

Троичные входы одноразрядных сумматоров

$$4\Sigma_3^1, 4\Sigma_2^1, 4\Sigma_1^1, 4\Sigma_0^1, 4\Sigma_{-1}^1, 4\Sigma_{-2}^1, -4\Sigma_{-3}^1$$

первой строки образуют многоразрядные троичные входы матричного троичного зеркально-симметричного сумматора. Выходы промежуточной суммы каждого одноразрядного сумматора связаны с соответствующим входом следующего одноразрядного зеркально-симметричного сумматора, находящегося в том же столбце.

Выходы промежуточной суммы одноразрядных сумматоров

$$4\Sigma_3^3, 4\Sigma_2^3, 4\Sigma_1^3, 4\Sigma_0^3, 4\Sigma_{-1}^3, 4\Sigma_{-2}^3, 4\Sigma_{-3}^3$$

последнего ряда матричного сумматора образуют многоразрядный выход матричного зеркально-симметричного сумматора.

Основной особенностью матричного зеркально-симметричного сумматора на Рис.11.5 является специальная организация связей между выходами переносов одноразрядных сумматоров и входами соседних одноразрядных сумматоров. Выходы переносов всех одноразрядных сумматоров с четными индексами  $(2, 0, -2)$  связаны с соответствующими входами соседних одноразрядных сумматоров, расположенных в том же ряду, в то время как выходы переносов всех одноразрядных сумматоров с нечетными индексами  $(3, 1, -1, -3)$  связаны с соответствующими входами соседних одноразрядных сумматоров, расположенных в нижнем ряду. Заметим, что такая организация связей переносов позволяет исключить вышеупомянутое явление «качелей».

Рассмотрим теперь действие матричного зеркально-симметричного сумматора на конкретном примере. Пусть необходимо просуммировать два одинаковых зеркально-симметричных числа:  $A = 0111.110$  и  $B = 0111.10$ .

Сложение выполняется в 2 этапа. Каждая этап реализуется с помощью одноразрядных сумматоров, расположенных в одном ряду матричного сумматора и состоит из двух тактов.

### Первый этап

Первый такт. В соответствии с Рис.11.5 первый такт сложения состоит в следующем. Одноразрядные сумматоры первого ряда с четными нижними

индексами  $(4\Sigma_2^1, 4\Sigma_0^1, 4\Sigma_{-2}^1)$  формируют промежуточные суммы, которые поступают на входы сумматоров второго ряда, и переносы, которые поступают на соответствующие входы одноразрядных сумматоров с нечетными нижними индексами первого ряда  $(4\Sigma_3^1, 4\Sigma_1^1, 4\Sigma_{-1}^1, 4\Sigma_{-3}^1)$ . Указанное выше преобразование информации может быть представлено в следующем виде:

$$\begin{array}{ccccccc}
 0 & 1 & 1 & 1, & 1 & 1 & 0 \\
 0 & 1 & 1 & 1, & 1 & 1 & 0 \\
 \hline
 & \bar{1} & & \bar{1} & & \bar{1} & \\
 & \downarrow & 1 & \leftrightarrow & 1 & \downarrow & \\
 1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1
 \end{array}$$

Следовательно, первый такт состоит в формировании промежуточных сумм и переносов на выходах одноразрядных сумматоров первого ряда с четными нижними индексами  $(2, 0, -2)$ .

Второй такт. На втором такте вступают в действие одноразрядные сумматоры первого ряда с нечетными нижними индексами  $(3, 1, -1, -3)$ . В соответствии с поступившими переносами они формируют промежуточные суммы и переносы, поступающие на входы одноразрядных сумматоров нижнего ряда, то есть,

$$\begin{array}{ccccccc}
 0 & 1 & 1 & 1, & 1 & 1 & 0 \\
 0 & 1 & 1 & 1, & 1 & 1 & 0 \\
 \hline
 & \bar{1} & & \bar{1} & & \bar{1} & \\
 & \downarrow & 1 & \leftrightarrow & 1 & \downarrow & \\
 1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1 \\
 \hline
 1 & \bar{1} & 1 & \bar{1}, & 1 & \bar{1} & 1 \\
 & 1 & \leftrightarrow & 1 & \downarrow & & \\
 & & & & 1 & \leftrightarrow & 1
 \end{array}$$

Первый этап закончен. Мы видим, что результатом первого этапа является формирование некоторых промежуточных сумм и переносов, поступивших на входы сумматоров нижнего ряда.

Существенно отметить, что для формирования кодовой информации на выходе сумматоров первого ряда потребуется время  $2\Delta\tau$ , где  $\Delta\tau$  - время задержки одноразрядного зеркально-симметричного сумматора, так как этап реализуется за два шага.

## Второй этап

Одноразрядные сумматоры второго ряда с четными нижними индексами  $(4\Sigma_2^2, 4\Sigma_0^2, 4\Sigma_{-2}^2)$  формируют промежуточные суммы, поступающие на соответствующие входы сумматоров нижнего ряда, и переносы, поступающие на соответствующие входы сумматоров этого же ряда с нечетными нижними индексами  $(4\Sigma_3^2, 4\Sigma_1^2, 4\Sigma_{-1}^2, 4\Sigma_{-3}^2)$ , то есть,

$$\begin{array}{cccccc}
 1 & \bar{1} & 1 & \bar{1}, & 1 & \bar{1} & 1 \\
 & & 1 & \leftrightarrow & 1 & \downarrow & \\
 & & & & 1 & \leftrightarrow & 1 \\
 \hline
 1 & 0 & 1 & 1, & 1 & 0 & 1
 \end{array}$$

Так как все переносы, сформированные на этом этапе, равны 0, это означает, что процесс сложения закончился на втором этапе (это справедливо только для рассматриваемого примера). Полученная сумма поступает на входы сумматоров нижнего ряда  $4\Sigma_3^3 - 4\Sigma_{-3}^3$  и затем возникает на выходе многозарядного сумматора.

### Конвейерный троичный зеркально-симметричный сумматор.

Существует два направления расширения функциональных возможностей матричного зеркально-симметричного сумматора на Рис.11.5. Если мы установим троичные регистры памяти между соседними рядами сумматоров, то рассмотренный выше матричный сумматор превращается в так называемый конвейерный зеркально-симметричный сумматор. Действительно, кодовая информация из предыдущего ряда сумматоров запоминается в соответствующих регистрах и тогда на следующем шаге предыдущий ряд сумматоров становится готовым для обработки новой информации. И пока сумматоры нижнего ряда обрабатывают кодовую информацию, поступившую из регистра, одновременно верхний ряд сумматоров обрабатывает новую информацию. Это означает, что, начиная с этого момента, мы будем одновременно обрабатывать суммы троичных чисел  $A_1 + B_1, A_2 + B_2, \dots, A_n + B_n$ , последовательно поступающих на вход многозарядного сумматора через время  $2\Delta\tau$ , где  $\Delta\tau$  - время задержки одноразрядного троичного зеркально-симметричного сумматора.

**Конвейерный зеркально-симметричный умножитель.** Еще одна возможность расширения функциональных возможностей сумматора на Рис.11.5

состоит в следующем. Мы можем видеть из Рис.11.5, что одноразрядные сумматоры нижних рядов имеют «свободные» входы, которые в совокупности образуют собой новые многоразрядные входы сумматора на Рис.11.5. И тогда мы можем использовать эти новые многоразрядные входы для расширения функциональных возможностей конвейерного сумматора и превратить этот сумматор в конвейерный умножитель. В этом случае зеркально-симметричное умножение двух зеркально-симметричных чисел  $A(1) \times B(1)$  осуществляется следующим образом. Первый ряд сумматоров суммирует два частных произведения  $P_1^1 + P_2^1$ . Эта кодовая информация запоминается в регистрах памяти и поступает для дальнейшей обработки в следующий ряд сумматоров. Если теперь мы пошлем 3-е частное произведение  $P_3^1$ , возникающее при умножении чисел  $A(1) \times B(1)$  на «свободный» многоразрядный вход сумматоров второго ряда, то на выходах 2-го ряда мы получим кодовую информацию о сумме частных произведений  $P_1^1 + P_2^1 + P_3^1$ . При этом первый ряд сумматоров начинает суммировать первые два частных произведения следующей пары умножаемых чисел  $A(2) \times B(2)$ . На следующем шаге умножения «свободные» входы 3-го ряда сумматоров используются для получения следующего частного произведения  $P_4^1$  первой пары умножаемых чисел  $A(1) \times B(1)$ , а на «свободные» входы второго ряда сумматоров поступает третье частное произведение второй пары умножаемых чисел. Таким образом, в конвейерном сумматоре на Рис.11.5 может одновременно обрабатываться в конвейерном режиме большое количество произведений зеркально-симметричных чисел, частные произведения которых последовательно поступают на «свободные» входы сумматоров. При этом скорость умножения будет определяться временем  $2\Delta\tau$ , где  $\Delta\tau$  - время задержки одноразрядного зеркально-симметричного сумматора.

## 11.7. Троичная логика и троичные элементы

**Основные функции троичной логики.** Троичная логика является специальным видом так называемой  $k$ -значной логики ( $k=2,3,4,5,\dots$ ) для случая  $k=3$ . Для согласования с троичной зеркально-симметричной системой счисления

будем предполагать, что логические переменные принимают значения из множества  $\{\bar{1}, 0, 1\}$ .

Тогда основные логические функции одной троичной переменной определяются следующим образом:

Функция «троичной инверсии»	Циклическое отрицание
$f(v) = \bar{v} = \begin{cases} \bar{1} & \text{для } v = 1 \\ 0 & \text{для } v = 0 \\ 1 & \text{для } v = \bar{1} \end{cases}$	$f(v) = \tilde{v} = \begin{cases} \bar{1} & \text{для } v = 0 \\ 0 & \text{для } v = 1 \\ 1 & \text{для } v = \bar{1} \end{cases}$

Рассмотрим теперь следующие важнейшие функции от двух логических переменных:

(1) Троичная конъюнкция  $f(v_1, v_2) = \min(v_1, v_2) = v_1 \wedge v_2$

$\wedge$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$
0	$\bar{1}$	0	0
1	$\bar{1}$	0	1

(2) Троичная дизъюнкция  $f(v_1, v_2) = \max(v_1, v_2) = v_1 \vee v_2$

$\vee$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	0	1
0	0	0	1
1	1	1	1

(3) Сложение по модулю 3  $f(v_1, v_2) = v_1 \oplus v_2 \pmod{3}$

$\oplus$	$\bar{1}$	0	1
$\bar{1}$	1	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$\bar{1}$

(4) Умножение по модулю 3  $f(v_1, v_2) = v_1 \otimes v_2 \pmod{3}$

$\otimes$	$\bar{1}$	0	1
$\bar{1}$	1	0	$\bar{1}$
0	0	0	0
1	$\bar{1}$	0	1

Существует следующие тождества, связывающие между собой введенные выше троичные логические функции:

$$\begin{aligned} \bar{\bar{v}} &= v; v \wedge v = v; v \wedge \bar{1} = \bar{1}; v \wedge 1 = v; \\ \bar{\bar{v}} &= v; v \wedge v = v; v \wedge \bar{1} = \bar{1}; v \wedge 1 = v; \\ v \oplus 0 &= v; v \otimes 0 = 0; v \otimes 1 = v; v \otimes \bar{1} = \bar{v}. \end{aligned}$$

Троичные функции конъюнкции, дизъюнкции и инверсии связаны между собой с помощью формул Моргана:

$$\overline{v_1 \wedge v_2} = \bar{v}_1 \vee \bar{v}_2; \quad \overline{v_1 \vee v_2} = \bar{v}_1 \wedge \bar{v}_2.$$

Подобно двоичной («булевой») логике существуют различные варианты функционально полных систем троичной логики. Для синтеза троичных логических элементов мы будем использовать логические функции так называемой «модулярной логики». Система «модулярной логики» включает следующие логические функции:

$$\overline{v_1 \wedge v_2} = \bar{v}_1 \vee \bar{v}_2; \quad \overline{v_1 \vee v_2} = \bar{v}_1 \wedge \bar{v}_2. \quad (11.38)$$

Добавим к «модулярным функциям» (11.38) следующую специальную логическую функцию:

$$f(v_1, v_2) = v_1 \ominus v_2. \quad (11.39)$$

Функция (11.39) задает правило формирования переноса при сложении одноразрядных троичных чисел. Логическая таблица, задающая эту функцию, имеет следующий вид:

$\ominus$	$\bar{1}$	0	1
$\bar{1}$	1	0	0
0	0	0	0
1	0	0	1

**Логические схемы инверсии и циклического отрицания.** Множество троичных функций (11.38), (11.39) составляют функционально полное множество троичных логических функций, которое использовано для синтеза троичных логических схем. Очень просто доказать, что функции «троичной инверсии»  $\bar{v}$  и «циклического отрицания»  $\bar{\bar{v}}$  реализуются с помощью логической функции сложения по модулю 3 (Рис.11.6).



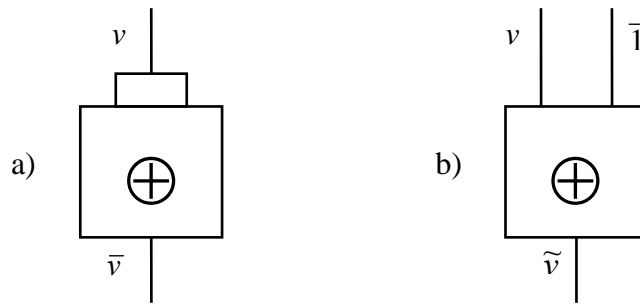


Рис.11.6. Логические схемы троичной инверсии (а) и циклического отрицания (b)

**Троичные одноразрядные сумматоры и умножители.** Троичный одноразрядный полусумматор  $2\Sigma$  реализуется с использованием логических схем  $\oplus$  и  $\ominus$  (Рис.11.6-а), а троичный одноразрядный умножитель представляет собой логическую схему  $\otimes$  (Рис.11.4-б).

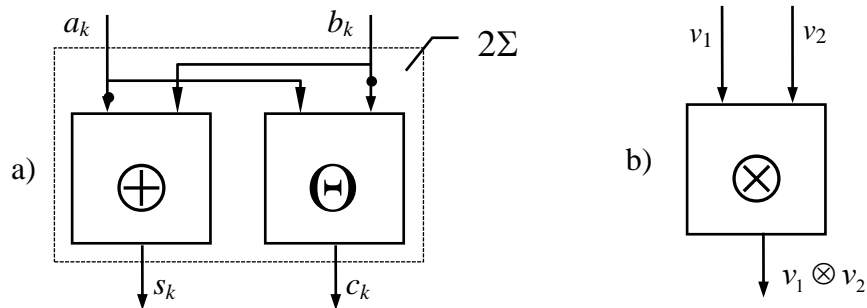


Рис.11.7. Троичный одноразрядный полусумматор (а) и умножитель (b)

Если использовать троичные одноразрядные полусумматоры  $2\Sigma$  (Рис.11.7-а) в качестве базовой логической схемы для реализации троичных зеркально-симметричных устройств, то можно доказать, что 5 одноразрядных полусумматоров на Рис.11.7-а образуют полный троичный одноразрядный сумматор  $4\Sigma$  (Рис.11.8).

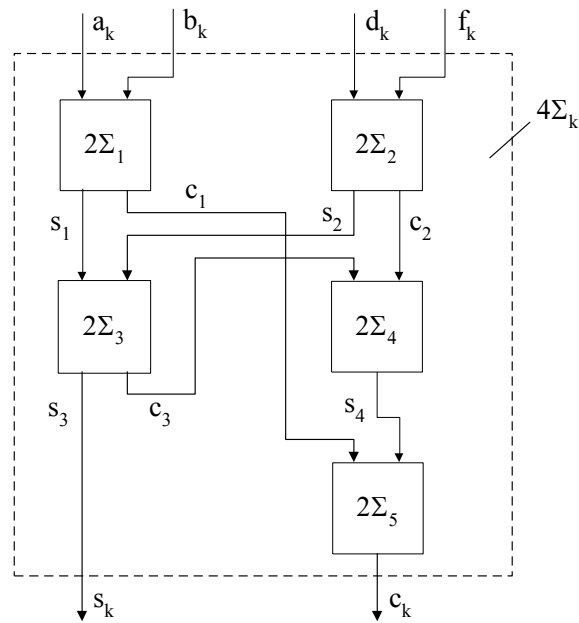


Рис.11.8. Полный троичный одноразрядный сумматор

При микросхемотехнической реализации троичных элементов в виде большой интегральной схемы (БИС) мы можем использовать следующий способ двоичного кодирования троичных переменных (Табл.11.5)

Таблица 11.5. Двоичное кодирование троичных переменных

$v$	$x_1$	$x_2$
$\bar{1}$	1	0
0	0	0
1	0	1

Тогда каждая логическая схема троичной арифметики, например, троичные сумматоры на Рис.11.7-а и 11.8 и троичный умножитель на Рис.11.7-б могут быть спроектированы в виде БИС, имеющих троичные входы и выходы. Все троичные переменные на каждом входе БИС преобразуются в двоичные переменные в соответствии с Табл.11.5 и все пары двоичных переменных преобразуются в соответствующие троичные переменные на выходах БИС в соответствии с Табл.11.5. И тогда проблема проектирования троичных схем сводится к проектированию традиционных двоичных БИС.

Заметим, что некоторые троичные функции при таком подходе реализуются проще, чем аналогичные функции булевой логики. Например, логическая функция «троичной инверсии»  $f(v) = \bar{v}$  при двоичном кодировании троичных переменных ( $v = x_1x_2$  и  $\bar{v} = x_2x_1$ ) реализуется, как показано на Рис.11.9.

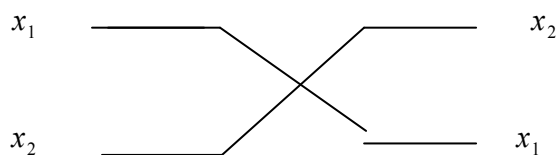


Рис.11.9. Логическая схема троичной инверсии

**Троичный элемент памяти (“flip-flap-flap”).** Такой же «двоичный подход» может быть использован для проектирования троичного элемента памяти, называемого “flip-flap-flap”. Известно, что основу классического двоичного элемента памяти (“flip-flop”) составляют два логических элемента 1 и 2 типа *OR–NO*, которые связаны между собой логическими обратными связями (Рис.11.10-а).

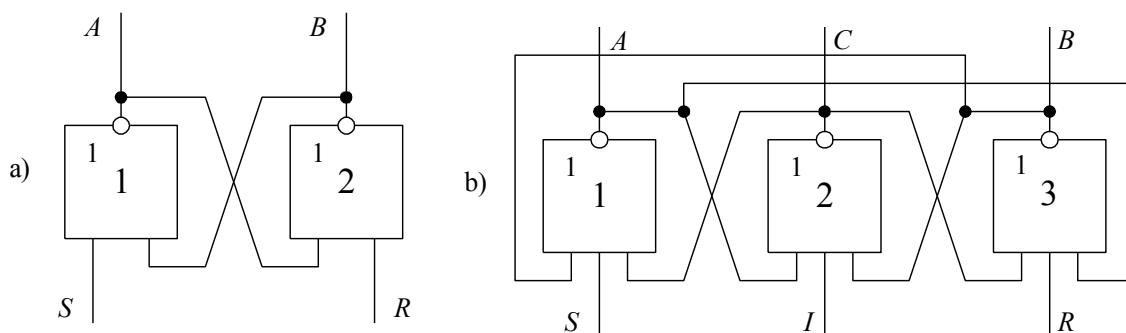


Рис.11.10. “Flip-flop” (а) и “flip-flap-flap” (b)

Рассмотрим теперь логическую схему, состоящую из трех логических элементов 1,2,3 типа *OR–NO* (Рис.11.10-б). Будем считать, что логические элементы 2 и 3 являются «соседними» к логическому элементу 1, логические элементы 3 и 1 являются «соседними» к логическому элементу 2 и логические элементы 1 и 2 являются «соседними» к логическому элементу 3. Каждый логический элемент *OR–NO* связан с «соседними» логическими элементами логическими обратными связями. Эти логические связи являются причиной трех стабильных состояний логической схемы на Рис.11.10-б.

Действительно, предположим, что мы имеем логическую 1 на выходе *C* логического элемента 2. Эта логическая 1 поступает на соответствующие входы «соседних» логических элементов 1 и 3 и образуют логические 0 на их выходах *A* и *B*. Эти логические 0 по цепям обратных связей поступают на входы логического элемента 2 и поддерживают логическую 1 на выходе *C*. Следовательно, это

состояние схемы на Рис.11.10-б является ее первым стабильным состоянием. Это стабильно состояние соответствует кодовой комбинации 010 на выходах  $A, C, B$ . Можно показать, что схема на Рис.11.10-б имеет два дополнительных стабильных состояния, соответствующих кодовым комбинациям 100 и 001 на выходах  $A, C, B$ . Мы можем использовать упомянутые стабильные состояния схемы на Рис.11.10-б для двоичного кодирования троичных переменных в соответствии со следующей таблицей:

		$A$	$C$	$B$
$\bar{1}$	=	1	0	0
0	=	0	1	0
1	=	0	0	1

Если мы исключим из рассмотрения двоичную переменную, соответствующую среднему выходу  $C$ , мы получим на выходах  $A$  и  $B$  двоичное кодирование троичных переменных, соответствующее Табл.11.5.

Таким образом, логическая схема на Рис.11.10-б может рассматриваться как троично-двоичный элемент памяти, называемый “flip-flap-flop”. Рассмотрим функционирование этого элемента, который может находиться в трех стабильных состояниях:  $\bar{1}$ , 0 и 1. Пусть “flip-flap-flop” на Рис.11.10-б находится в состоянии  $Q=0(010)$ . Это означает, что выход  $C=1$ , а другие выходы  $A=B=0$ . Если нам необходимо установить “flip-flap-flop” в следующее состояние  $Q=1(001)$ , мы должны подать на входы  $S, I, R$  “flip-flap-flop” следующие логические сигналы:  $S=1, I=1, R=0$ . Сигналы  $S=1, I=1$  вызывают появление логических 0 на выходах  $A$  и  $C$ . Эти логические 0 поступают на входы «соседнего» логического элемента 3 и вместе с логическим сигналом  $R=0$  вызывают возникновение логической 1 на выходе элемента  $B$ .

Аналогично можно показать, что входные сигналы  $S=0, I=1, R=1$  устанавливают “flip-flap-flop” на Рис.11.10-б в третье стабильное состояние  $Q=\bar{1}(100)$ .

## 11.8. Троичный зеркально-симметричный цифроаналоговый преобразователь

«Золотой» резистивный делитель. Выше (Глава 10) мы рассмотрели различные варианты «золотых» резистивных делителей. Все эти делители имеют

одну и ту же электрическую схему (Рис.10.5). Различие между резистивными делителями определяется выбором значений резисторов на Рис.10.5. В настоящем параграфе мы рассмотрим новый вид «золотого» резистивного делителя, схема которого представлена на Рис.11.11.

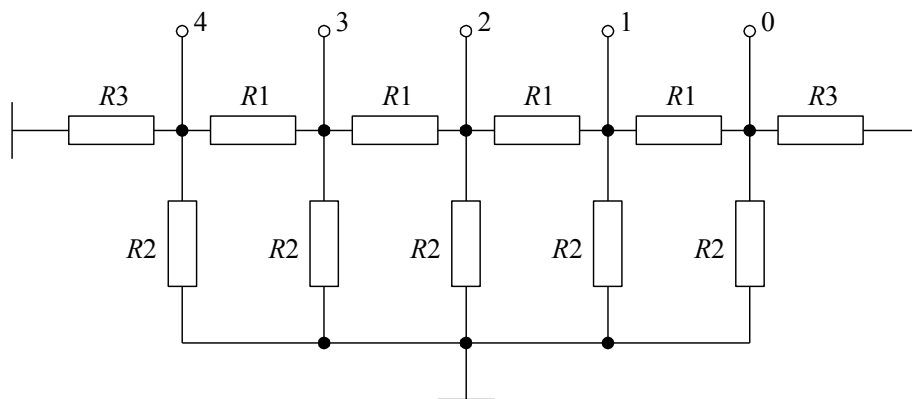


Рис.11.11. «Золотой» резистивный делитель для троичного зеркально-симметричного кода

В делителе на Рис.11.11 значения резисторов  $R_1, R_2, R_3$  выбраны следующим образом:

$$R_1 = R; R_2 = R; R_3 = \Phi R, \quad (11.40)$$

где  $\Phi = (1 + \sqrt{5})/2$  - золотая пропорция.

Вычислим эквивалентное сопротивление  $R_{e1}$  параллельного соединения резисторов  $R_2 = R$  и  $R_3 = \Phi R$  справа от точки 0 и слева от точки 4 (Рис.11.11):

$$R_{e1} = \frac{R_2 \times R_3}{R_2 + R_3} = \frac{R \times \Phi R}{R + \Phi R} = \Phi^{-1} R. \quad (11.41)$$

Если теперь вычислить значение эквивалентного сопротивления последовательного соединения сопротивлений  $R_1 = R$  и эквивалентного сопротивления  $R_{e1} = \Phi^{-1} R$  справа от точки 1 и слева от точки 3, то получим:

$$R_{e2} = R_1 + R_{e1} = R + \Phi^{-1} R = \Phi R. \quad (11.42)$$

Продолжая эти рассуждения, нетрудно показать, в любой точке 0,1,2,3,4 значение эквивалентного сопротивления резистивной цепи слева или справа от каждой точки всегда равно  $R_{e2} = \Phi R$ . Таким образом, в любой точке 0,1,2,3,4 как бы объединяются три сопротивления – «вертикальное» сопротивление  $R_2 = R$  и два эквивалентных сопротивления  $R_{e2} = \Phi R$ . Поскольку параллельное соединение сопротивления  $R_2 = R$  и эквивалентного сопротивления  $R_{e2} = \Phi R$

равно  $R_{e1} = \Phi^{-1}R$ , то легко вычислить эквивалентное сопротивление резистивной цепи в любой из точек 0,1,2,3,4, используя формулу:

$$R_{e3} = \frac{\Phi R \times \Phi^{-1}R}{\Phi R + \Phi^{-1}R} = \frac{R^2}{\Phi^2 R} = \Phi^{-2}R. \quad (11.43)$$

Если к одной из точек, например, к средней точке 2 подключить генератор эталонного тока  $I_0$ , то в этой точке появится электрическое напряжение

$$U_0(2) = I_0 \Phi^{-2}R. \quad (11.44)$$

Точка 2 соединена с точками 1 и 3 с помощью резисторов типа  $R1 = R$ . Вспомним теперь, что эквивалентные сопротивления резисторной цепи справа от точки 1 и слева точки 3 задается выражением (11.42) и равны  $R_{e1} = \Phi^{-1}R$ . В этом случае, используя «закон Ома», можно легко вычислить напряжение в точках 1 и 3:

$$U_0(1) = U_0(3) = \frac{U_0(2)}{R + \Phi^{-1}R} \times \Phi^{-1}R = U_0(2) \times \Phi^{-2} = I_0 \Phi^{-4}R. \quad (11.45)$$

Сравнивая выражения (11.44) и (11.45), можно установить, что коэффициент передачи по напряжению между двумя соседними точками 2 и 3, 2 и 1 равен величине, обратной 2-й степени золотой пропорции  $\Phi^{-2}$ .

**Троичный зеркально-симметричный ЦАП.** «Золотой» резистивный делитель на Рис.11.11, в котором значения резисторов выбраны в соответствии с (11.40) может быть использован для проектирования троичного зеркально-симметричного цифроаналогового преобразователя (ЦАП), представленного на Рис.11.12. Этот ЦАП состоит из 5 (в общем случае  $n$ ) разрядов. Средняя точка соответствует 0-му разряду  $a_0$  входного троичного зеркально-симметричного кода  $a_2 a_1 a_0 a_{-1} a_{-2}$  ( $a_m a_{m-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$  - в общем случае) числа  $N$ . Троичные разряды  $a_i$  ( $i = 0, \pm 1, \pm 2, \dots, \pm m$ ) управляют эталонными токами  $I_0$ , которые через трехпозиционные ключи подключены к соответствующим точкам «золотого» резистивного делителя. Эти ключи управляются троичными цифрами  $a_i$  согласно следующим правилам. Если  $a_i = 1$ , тогда эталонный ток  $I_0$  подключается к соответствующей точке «золотого» троичного делителя в «позитиве»,  $+I_0$ . Если  $a_i = -1$ , тогда эталонный ток  $I_0$  подключается к соответствующей точке

«золотого» троичного делителя в «негативе»,  $-I_0$ . Наконец, если  $a_i = 0$ , тогда эталонный ток  $I_0$  не подключается к соответствующей точке делителя.

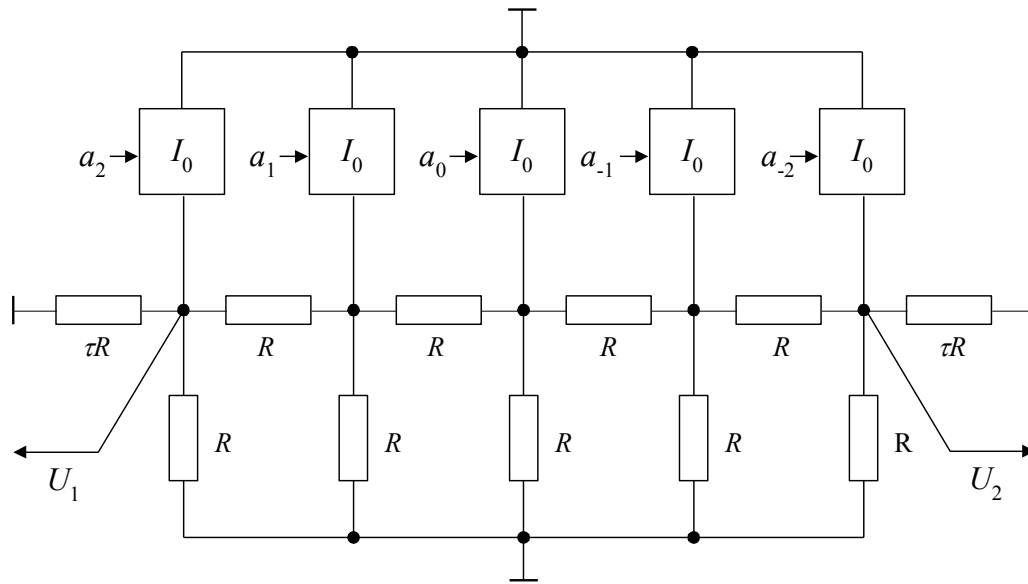


Рис.11.12. «Золотой» троичный зеркально-симметричный ЦАП

«Золотой» троичный зеркально-симметричный ЦАП имеет два «зеркально-симметричных» выхода,  $U_1$  и  $U_2$ . Если на входы ЦАП подать троичный зеркально-симметричный код  $a_m a_{m-1} \dots a_2 a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-(m-1)} a_{-m}$ , то, используя принцип суперпозиции, можно показать, что на выходах  $U_1$  и  $U_2$  появится напряжение, пропорциональное троичному зеркально-симметричному коду (11.9):

$$U_1 = U_2 = I_0 \Phi^{-2} R \sum_{i=-m}^m a_i \Phi^{2i}. \quad (11.46)$$

Фундаментальное контрольное свойство «золотого» зеркально-симметричного ЦАП на Рис.11.12 состоит в следующем:

$$\boxed{U_1 = U_2}. \quad (11.47)$$

Нарушение равенства (11.47) является признаком ошибки в ЦАП. Таким образом, «золотой» зеркально-симметричный ЦАП на Рис.11.12 является самоконтролирующимся ЦАП, в котором осуществляется непрерывный контроль ЦАП в процессе его функционирования.

В системах управления существует так называемая проблема «последнего каскада». Ее суть состоит в том, что ЦАП, основанные на классическом двоичном коде, являются неконтролируемыми, в силу «нулевой» избыточности двоичной

системы счисления. «Золотой» зеркально-симметричный ЦАП на Рис.11.12 решает очень важную проблему «последнего каскада» в системах управления.



## Глава 12

# ПЕРСПЕКТИВНЫЕ ИДЕИ В РАЗВИТИИ ФИБОНАЧЧИЕВОГО КОДИРОВАНИЯ

### 12.1. Троичный код Фибоначчи

**Определение.** В главе 11 предложено троичное зеркально-симметричное представление натуральных чисел, которое является синтезом системы Бергмана и классической троичной симметричной системы счисления. Однако, существует еще одно направление развития теории кодов Фибоначчи, основанное на использовании троичного алфавита  $\{\bar{1}, 0, 1\}$ .

Рассмотрим следующее позиционное представление, которое будем называть троичным кодом Фибоначчи:

$$N = c_n F_n + c_{n-1} F_{n-1} + \dots + c_i F_i + \dots + c_1 F_1, \quad (12.1)$$

где  $c_i = \{\bar{1}, 0, 1\}$  троичная цифра  $i$ -го разряда.

**Троичная фибоначчиева цифровая запись.** Заметим, что выбор троичного алфавита  $\{\bar{1}, 0, 1\}$  приводит к позиционному представлению, которое может быть использовано для представления всех целых чисел (положительных и отрицательных). Сокращенная троичная фибоначчиева цифровая запись (ФЦЗ) позиционного представления (12.1) имеет вид:

$$N = c_n c_{n-1} \dots c_i \dots c_1. \quad (12.2)$$

Применяя к троичной ФЦЗ числа  $N$ , задаваемой (12.2), правило троичной инверсии

$$\bar{1} \rightarrow 1; 0 \rightarrow 0; 1 \rightarrow \bar{1} \quad (12.3)$$

мы можем получить троичную ФЦЗ отрицательного числа  $(-N)$ .

**Пример 12.1.** Пусть задана троичная ФЦЗ некоторого числа  $(N = 100\bar{1}10\bar{1}01\bar{1})$ . Применяя правило (12.3), получим троичную ФЦЗ отрицательного числа:  $(-N = \bar{1}001\bar{1}010\bar{1}1)$ .

**Микрооперации «свертки» и «развертки».** Для преобразования троичных ФЦЗ одного и того же целого числа  $N$ , мы можем использовать следующие рекуррентные соотношения:

$$F_{n-2} = F_n - F_{n-1} \quad (12.4)$$

$$F_{n-1} = F_n - F_{n-2}. \quad (12.5)$$

Рекуррентные соотношения (12.4) и (12.5) имеют следующие кодовые интерпретации, соответственно:

$$[010] \rightarrow [10\bar{1}]; \quad [10\bar{1}] \rightarrow [010] \quad (12.6)$$

$$[001] \rightarrow [1\bar{1}0]; \quad [1\bar{1}0] \rightarrow [001]. \quad (12.7)$$

Будем называть троичные преобразования  $[10\bar{1}] \rightarrow [010]$  и  $[1\bar{1}0] \rightarrow [001]$  троичной сверткой-1 (ТС-1) и троичной сверткой-2 (ТС-2), соответственно.

Троичные преобразования  $[010] \rightarrow [10\bar{1}]$  и  $[001] \rightarrow [1\bar{1}0]$  будем называть троичной разверткой-1 (ТР-1) и троичной разверткой-2 (ТР-2), соответственно.

Ясно, что выполнение микроопераций ТС и ТР в троичной ФЦЗ (12.2) не изменяет числа  $N$ , представляемого троичной ФЦЗ (12.2).

Учитывая, что мы используем троичный алфавит  $\{\bar{1}, 0, 1\}$ , с помощью правила троичной инверсии (12.3) мы можем представить микрооперации (12.6) и (12.7) следующим образом:

$$[0\bar{1}0] \rightarrow [\bar{1}01]; \quad [\bar{1}01] \rightarrow [0\bar{1}0] \quad (12.8)$$

$$[00\bar{1}] \rightarrow [\bar{1}10]; \quad [\bar{1}10] \rightarrow [00\bar{1}]. \quad (12.9)$$

**Микрооперация «обнуление».** Введем теперь для троичного кода Фибоначчи (12.1) важную микрооперацию «обнуление». Чтобы объяснить это понятие, запишем основное рекуррентное соотношение для чисел Фибоначчи  $F_n = F_{n-1} + F_{n-2}$  в следующем виде:

$$F_n - F_{n-1} - F_{n-2} = 0; \quad -F_n + F_{n-1} + F_{n-2} = 0. \quad (12.10)$$

Равенства (12.10) имеют следующие троичные интерпретации:

$$[\bar{1}\bar{1}\bar{1}] \rightarrow [000] \quad (12.11)$$

$$[\bar{1}11] \rightarrow [000]. \quad (12.12)$$

Микрооперация обнуления, так же, как и микрооперации троичной свертки позволяют уменьшить количество единиц (положительных или отрицательных) при выполнении арифметических операций, что особенно важно при умножении.

**Пример 12.2.** Применить микрооперации обнуления и троичной свертки для уменьшения числа единиц в следующей троичной ФЦЗ:

$$1\bar{1}\bar{1}10\bar{1}\bar{1}\bar{1}111. \quad (12.13)$$

Анализ троичной ФЦЗ (12.13) показывает, что существует несколько путей для уменьшения числа единиц. Например, применяя микрооперацию ТС-1  $[10\bar{1}] \rightarrow [010]$  и микрооперации обнуления (12.11), мы можем выполнить следующее преобразование троичной ФЦЗ (12.13):

$$\boxed{\bar{1}\bar{1}\bar{1}} \boxed{10\bar{1}} \boxed{\bar{1}\bar{1}\bar{1}} 111 = 000010000111. \quad (12.14)$$

Затем мы можем уменьшить число единиц в (12.14), используя микрооперацию свертки  $[011] \rightarrow [100]$  для двоичного кода Фибоначчи:

$$00001000 \boxed{011} 1 = 000010001001. \quad (12.15)$$

Таким образом, в результате проведенных преобразований мы получили ФЦЗ (12.15), которая содержит 3 единицы вместо 11 в исходной ФЦЗ (12.13).

Заметим, что микрооперация «обнуление» может быть использована для контроля троичной ФЦЗ. Действительно, выполнение этой микрооперации приводит к трем подряд идущим 0 в соответствующих разрядах троичной ФЦЗ (12.2). Это может быть проконтролировано специальной логической схемой. Важно подчеркнуть, что «обнуление» осуществляется над группой троичных разрядов типа  $\bar{1}11$  или  $1\bar{1}\bar{1}$ . Схемотехнически это осуществляется путем переключения соответствующих троичных триггеров (flip-flap-flop) из состояния 1 или  $\bar{1}$  в состояние 0. Это создает предпосылки для реализации эффективного контроля случайных сбоев троичных элементов памяти, которые могут возникнуть в процессе их переключения.

**Формы представления чисел в троичном коде Фибоначчи.** Существуют различные формы представления чисел в троичном коде Фибоначчи (12.1):

1. Однородная форма, когда мы используем только две цифры: 0 и 1 или 0 и  $\bar{1}$ . Например,  $A = 0100111001$  или  $B = \bar{1}00\bar{1}\bar{1}00\bar{1}$ .
2. Смешанная форма, когда мы используем троичные цифры  $\{\bar{1}, 0, 1\}$ . Например,  $A = 10\bar{1}10010\bar{1}$ .
3. Минимальная форма, когда две единицы рядом (положительные или отрицательные) в однородной форме не встречаются. Например,  $A = 01001010010$  или  $B = \bar{1}00\bar{1}0\bar{1}00\bar{1}00$ .
4. Максимальная форма, когда два нуля рядом, начиная со старшего значащего разряда (1 или  $\bar{1}$ ) в однородной форме не встречаются. Например,  $A = 0110111011$  или  $B = 00\bar{1}0\bar{1}\bar{1}\bar{1}0\bar{1}$ .

В процессе выполнения арифметических операций необходимо преобразовывать смешанные формы в однородные. Такие преобразования могут выполняться путем использования введенных выше троичных микроопераций ТС-1, ТР-1, ТС-2, ТР-2, и микрооперации «обнуление». Например, преобразуем смешанную форму в однородную:

$$0\boxed{\bar{1}\bar{1}}\boxed{\bar{1}\bar{1}0}\boxed{0\bar{1}\bar{1}}\boxed{011} = 00000010000100. \quad (12.16)$$

В процессе преобразования (12.16) выполняются следующие микрооперации (слева направо): «обнуление» типа  $[\bar{1}\bar{1}] \rightarrow [000]$ , троичная свертка типа  $[\bar{1}\bar{1}0] \rightarrow [001]$ , «обнуление» типа  $[\bar{1}11] \rightarrow [000]$ , двоичная свертка типа  $[011] \rightarrow [100]$ . Такое преобразование приводит к уменьшению числа единиц в троичной ФЦЗ (12.6) от 10 до 2.

## 12.2. Троичная арифметика Фибоначчи

**Сравнение чисел.** Если сравниваемые числа представлены в смешанной форме, мы должны преобразовать их в однородную форму и затем использовать правило сравнения чисел в двоичном коде Фибоначчи.

**Сложение и вычитание.** Правило сложения чисел в троичном коде Фибоначчи (12.1) основано на использовании следующих тождеств:

$$F_n + F_n = F_{n+1} + F_n - F_{n-1}; \quad -F_n - F_n = -F_{n+1} - F_n + F_{n-1}. \quad (12.17)$$

Формулы (12.17) имеет следующие кодовые интерпретации:

$$1+1=1\bar{1}\bar{1} \quad (12.18)$$

$$\bar{1}+\bar{1}=\bar{1}\bar{1}1. \quad (12.19)$$

Используя (12.18) и (12.19), мы можем сконструировать следующую таблицу троичного фибоначчиевого сложения:

+	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}\bar{1}1$	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$1\bar{1}\bar{1}$

(12.20)

Отличительной особенностью таблицы (12.20) являются правила троичного фибоначчиевого сложения двух троичных единиц с одинаковыми знаками, задаваемыми (12.18) и (12.19).

Мы видим, что при фибоначчиевом сложении троичных единиц  $k$ -го разряда  $a_k + b_k$  с одинаковыми знаками  $(1+1)$  и  $(\bar{1}+\bar{1})$  возникает сумма  $s_k$ , равная 1 того же знака, что и слагаемые единицы, и два единичных переноса в соседние разряды – левый единичный перенос  $c_k^l$  того же знака в  $(k+1)$ -й разряд, и правый единичный перенос  $c_k^r$  противоположного знака.

Ясно, что троичное фибоначчиево вычитание чисел  $A-B$  сводится к троичному фибоначчиевому сложению, если к вычитаемому  $B$  применить правило троичной инверсии (12.3), то есть,

$$A - B = A + (-B). \quad (12.21)$$

Рассмотрим примеры троичного фибоначчиевого сложения и вычитания чисел, представленных в однородной минимальной форме.

**Пример 12.3.** Сложить два троичные ФЦЗ чисел  $(A+B)$ , представленных в минимальной форме:

$n$	=	11	10	9	8	7	6	5	4	3	2	1
$A$	=	0	<b>1</b>	0	<b>1</b>	0	0	1	0	<b>1</b>	0	0
$B$	=	0	<b>1</b>	0	<b>1</b>	0	1	0	0	<b>1</b>	0	0
$A+B$	=	1	1	0	1	$\bar{1}$	1	1	1	1	$\bar{1}$	0

Дадим объяснение процесса сложения и полученного результата. Мы видим, что в 3-м, 8-м и 10-м разрядах мы имеем ситуацию (1+1) (выделено жирным шрифтом). Для ситуации (1+1) в 3-м разряде формируется сумма  $s_3 = 1$  и возникают два единичных переноса в соседние (4-й и 2-й) разряды. Так как слагаемые  $A$  и  $B$  представлены в минимальной форме, это означает, что переносы из 3-го разряда могут быть сразу помещены в 4-й и 2-й разряды суммы, то есть  $s_4 = 1$  и  $s_2 = \bar{1}$ .

Рассмотрим теперь ситуацию (1+1) в 8-м и 10-м разрядах. В ситуации (1+1) для 8-го разряда в этом разряде формируется сумма  $s_8 = 1$  и возникают два единичных переноса в соседние разряды, перенос отрицательной единицы  $\bar{1}$  в 7-й разряд и перенос положительной единицы 1 в 9-й разряд. Первый перенос  $\bar{1}$  сразу же помещается в 7-й разряд суммы, то есть  $s_7 = \bar{1}$ .

В ситуации (1+1) для 10-го разряда в этом разряде формируется сумма  $s_{10} = 1$  и возникают два единичных переноса в соседние разряды, перенос отрицательной единицы  $\bar{1}$  в 9-й разряд и перенос положительной единицы 1 в 11-й разряд. Перенос положительной единицы 1 из 10-го разряда сразу помещается в 11-й разряд суммы, то есть  $s_{11} = 1$ . Ясно также, что перенос положительной единицы 1 из 8-го в 9-й разряд и перенос отрицательной единицы  $\bar{1}$  из 10-го разряда суммируются  $(1 + \bar{1} = 0)$  и поэтому мы должны записать цифру 0 в 9-й разряд суммы, то есть,  $s_9 = 0$ .

Значения 1-го, 5-го и 6-го разрядов суммы формируются согласно таблице (12.20), то есть,  $s_1 = 0+0=0$ ,  $s_5 = 1+0=1$ ,  $s_6 = 0+1=1$ .

Наиболее важное свойство сложения двух чисел, представленных в однородной минимальной форме, состоит в том, что для любых значений слагаемых, представленных в минимальной форме, сложение выполняется за один такт, что является предпосылкой для проектирования супербыстрых арифметических устройств.

Полученный результат  $A+B=1101\bar{1}1111\bar{1}0$  представлен в смешанной форме. Этот результат имеет следующую числовую интерпретацию:

$F_n$	=	89	55	34	21	13	8	5	3	2	1	1		$\Sigma$
$A+B$	=	1	1	0	1	$\bar{1}$	1	1	1	1	$\bar{1}$	0	=	169

Мы можем преобразовать его в однородную форму, используя следующие микрооперации: «обнуление»  $[\bar{1}11] \rightarrow [000]$  и «троичная свертка»  $[\bar{1}\bar{1}0] \rightarrow [001]$ :

$$A+B=1101\boxed{\bar{1}11}1\boxed{\bar{1}\bar{1}0}=11010001001.$$

После этого полученный результат приводится к минимальной форме:

$$A+B=\boxed{011}0100010\boxed{01}=100010001010.$$

**Пример 12.4.** Найти разность двух чисел  $(A-B)$ , представленных в минимальной форме:  $A=100101001010$  и  $B=010001010010$ .

Представим разность  $(A-B)$  в виде  $A+(-B)$ . Для этого применим правило троичной инверсии (12.3) к числу  $B$ :

$B$	=	0	1	0	0	0	1	0	1	0	0	1	0
$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
$-B$	=	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	$\bar{1}$	0

Выполним операцию сложения чисел  $A+(-B)$ :

$n$	=	12	11	10	9	8	7	6	5	4	3	2	1
$A$	=	1	0	0	1	0	1	0	0	1	0	1	0
$-B$	=	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0	0	$\bar{1}$	0
$A+(-B)$	=	1	$\bar{1}$	0	1	0	0	0	$\bar{1}$	1	0	0	0

Результат сложения  $A+(-B)=1\bar{1}01000\bar{1}1000$  представлен в смешанной форме. Мы можем преобразовать его сначала в однородную форму, а затем в минимальную форму, используя следующие преобразования:

$$A+(-B)=\boxed{1\bar{1}0}1000\boxed{\bar{1}10}00=0\boxed{011}00000100=0100000100$$

**Троичное фибоначчьево умножение и деление.** Эти арифметические операции выполняются над числами, которые предварительно приводятся к

минимальной форме, и аналогичны подобным арифметическим операциям в двоичной фибоначчией арифметике.

**Преимущества троичной фибоначчией арифметики.** Троичная фибоначчией арифметика имеет следующие важные преимущества:

1. Первое преимущество состоит в возможности представления всех целых чисел (положительных и отрицательных) и выполнения над ними арифметических операций в прямом коде. Это означает, что в троичном коде Фибоначчи (12.1) фундаментально решается проблема знака. При этом существует очень простой алгоритм преобразования положительного числа в отрицательное, основанный на использовании правила троичной инверсии (12.3).

2. Троичное фибоначчией сложение является главной арифметической операцией для троичной арифметики Фибоначчи; все остальные арифметические операции (вычитание, умножение и деление) сводятся к сложению.

3. Основное преимущество троичной арифметики Фибоначчи содержит необходимые предпосылки для проектирования супербыстрых арифметических устройств, поскольку основная арифметическая операция (троичное фибоначчией сложение) выполняется за один такт (без длинных переносов), если слагаемые представлены в однородной минимальной форме. Конечно, главной проблемой при проектировании таких арифметических устройств является проектирование устройств для быстрого преобразования троичного кода Фибоначчи (12.1) из смешанной формы в однородную и затем в минимальную формы.

4. Троичный код и арифметика Фибоначчи обладают рядом «естественных» контрольных свойств. Во-первых, в однородной форме троичный код Фибоначчи обладает всеми контрольными свойствами двоичного кода Фибоначчи (минимальная и максимальная формы). Важным контрольным свойством обладает микрооперация «обнуление». Если мы суммируем два числа, представленные в однородной минимальной форме, и в некоторый разряд одновременно возникают переносы единиц из двух соседних разрядов, то эти единицы должны быть противоположных знаков; нарушение этого правила является признаком ошибки при сложении.



## 12.3. Новая теория кодирования, основанная на матрицах Фибоначчи

**Алгебраическая теория корректирующих кодов: принципы обнаружения и коррекции ошибок.** В современной информатике широко используются так называемые алгебраические корректирующие коды. Основная идея таких кодов (код Хемминга, коды Рида-Соломона, циклические коды и др.) [115] состоит в следующем. Пусть исходная кодовая комбинация (КК), называемая информационной, состоит из  $n$  битов. Для образования корректирующего кода мы добавляем к информационной  $n$ -разрядной КК  $m$  корректирующих битов и образуем  $k$ -разрядную КК корректирующего  $(k, n)$ -кода, где  $k = n + m$ .

Корректирующие биты образуются из информационной КК путем сложения по «модулю 2» определенных (проверочных) групп информационных разрядов. Все КК корректирующего  $(k, n)$ -кода называются разрешенными КК. Ясно, что число разрешенных КК совпадает с числом всех возможных информационных КК и равно  $2^n$ . С другой стороны, длина каждой разрешенной КК  $a_1 a_2 a_3 \dots a_k$  равна  $k = n + m$ . Это означает, что число всех возможных КК длины  $k = n + m$  равно  $2^k = 2^{n+m}$ .

Разделим множество всех возможных  $k$ -разрядных КК на две непересекающиеся группы. К первой группе отнесем  $2^n$  разрешенных КК, ко второй – все остальные  $(2^k - 2^n)$  КК, которые мы будем называть запрещенными КК.

При передаче информации мы посылаем в канал связи одну из  $2^n$  разрешенных КК. Под влиянием помех, действующих в канале, эта разрешенная КК может перейти в одну из  $2^k$  возможных КК. Если рассмотреть все возможные варианты, то мы можем вычислить потенциальное число всех возможных переходов, которые могут возникнуть при передаче всех разрешенных КК корректирующего кода. Оно равно:  $N_0 = 2^n \times 2^k$ .

Принцип обнаружения ошибок состоит в фиксации запрещенной КК, которая может появиться на выходе канала связи. Все возможные переходы можно разделить на обнаруживаемые и необнаруживаемые. Обнаруживаемый переход – это такой переход, который приводит к возникновению на выходе канала запрещенной КК. Необнаруживаемые переходы – это такие, которые

приводят к появлению на выходе канала другой разрешенной КК. Ясно, что число всех обнаруживаемых переходов для каждой КК корректирующего  $(k, n)$ -кода совпадает с числом запрещенных КК и равно  $(2^k - 2^n)$ . Если рассмотреть ситуацию для всего множества разрешенных КК, то можно вычислить число всех обнаруживаемых переходов; оно равно  $N_d = 2^n \times (2^k - 2^n)$ . Если теперь вычислить отношение  $N_d / N_0$ , то мы получим первую важную характеристику корректирующего кода, называемую потенциальным коэффициентом обнаружения ошибок:

$$S_d = \frac{N_d}{N_0} = \frac{2^n (2^k - 2^n)}{2^n 2^k} = 1 - \frac{1}{2^m} \quad (12.22)$$

где  $m = k - n$  – число корректирующих разрядов.

Принцип коррекции ошибок состоит в следующем. Все  $(2^k - 2^n)$  запрещенные КК разбиваются на  $2^n$  непересекающихся подмножеств  $M_1, M_2, M_3, \dots, M_{2^n}$ , где  $2^n$  – число разрешенных кодовых комбинаций. Каждая разрешенная КК приписывается к одному из  $2^n$  подмножеств:  $A_1 \rightarrow M_1, A_2 \rightarrow M_2, A_3 \rightarrow M_3, \dots$ . Тогда принцип коррекции состоит в следующем. Если мы получим запрещенную КК, которая принадлежит к подмножеству  $M_i$ , мы считаем, что была передана разрешенная КК  $A_i$ . В этом случае мы корректируем правильно только те ошибочные КК из множества  $M_i$  которые были образованы из разрешенной КК  $A_i$ . В противном случае коррекция ошибки будет выполняться неправильно. Ясно, что число корректируемых ошибочных переходов  $N_c$  равно числу всех запрещенных КК, то есть,  $N_c = 2^k - 2^n$ .

Потенциальный коэффициент коррекции ошибок вычисляется как отношение всех корректируемых переходов  $N_c$  к числу всех обнаруживаемых переходов, то есть,

$$S_c = \frac{N_c}{N_d} = \frac{2^k - 2^n}{2^n (2^k - 2^n)} = \frac{1}{2^n} \quad (12.23)$$

где  $n$  – число информационных битов в КК корректирующего  $(k, n)$ -кода.

Анализ формул (12.32) и (12.33) позволяет сделать некоторые важные выводы, касающиеся эффективности корректирующих алгебраических кодов. Формула (12.32) показывает, что потенциальный коэффициент обнаружения

ошибок увеличивается экспоненциально и стремится к 100% по мере увеличения числа корректирующих разрядов  $m$ . Этот факт подтверждает высокую эффективность корректирующих кодов в обнаружении ошибок, которая увеличивается с увеличением числа корректирующих разрядов. В то же время формула (12.33) показывает, что потенциальный коэффициент коррекции ошибок экспоненциально уменьшается по мере увеличения числа информационных разрядов  $n$ . Например, (15,11)-код Хемминга, гарантирующий исправление всех одиночных ошибок в 15-разрядной КК, позволяет обнаружить  $2^{11} \times (2^{15} - 2^{11}) = 62\,914\,560$  ошибочных переходов. При этом он может гарантировать правильную коррекцию только  $2^{15} - 2^{11} = 30\,720$  из всех ошибочных переходов, то есть, он может потенциально правильно корректировать только  $\frac{30720}{62\,914\,560} = 0.0004882 (0.04882\%)$  переходов из общего числа ошибочных переходов.

Естественно возникает вопрос о целесообразности практического использования корректирующих алгебраических кодов с такой низкой потенциальной корректирующей способностью для коррекции ошибок. Специалисты в области корректирующих кодов, как правило, уходят от ответа на этот вопрос. Однако, во всех учебниках по теории кодирования они описывают рассмотренный выше (15,11)-код Хемминга, как одно из высших достижений алгебраической теории кодирования. Заметим, что рассуждения по поводу ошибкообнаруживающей и корректирующей способности алгебраических кодов, проведенные в книге [115] выдающегося советского специалиста в этой области академика А.А. Харкевича, носят общий характер и применимы ко всем алгебраическим корректирующим кодам.

Из проведенных рассуждений вытекает первый существенный недостаток алгебраической теории корректирующих кодов [115]: алгебраические корректирующие коды обладают очень низкой потенциальной корректирующей способностью, которая стремится к нулю с увеличением длины информационной кодовой комбинации.

Второй их недостаток состоит в том, что объектом коррекции являются очень мелкие информационные элементы – биты или их комбинации.

Возникает вопрос: нельзя ли создать новую теорию корректирующих кодов, которые позволяют корректировать не биты, а значительно более крупные информационные объекты, например, числа или даже файлы.

Одно из направлений создания новой теории корректирующих кодов основано на использовании матриц Фибоначчи [84]. Настоящий параграф содержит набросок новой теории избыточного кодирования, основанной на матрицах Фибоначчи.

**Невырожденные или несингулярные матрицы.** Квадратная матрица  $A$  называется невырожденной или несингулярной, если существует квадратная матрица  $B$  такая, что

$$AB = I \text{ и } BA = I,$$

где  $I$  есть единичная матрица. Матрица  $B$  называется обратной к матрице  $A$  и обозначается  $B = A^{-1}$ .

Основное свойство невырожденной матрицы  $A$  состоит в том, что ее детерминант не равен 0, то есть,  $\det A \neq 0$

Рассмотрим общий случай квадратной невырожденной (2×2)-матрицы:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad (12.24)$$

где  $a_{11}, a_{12}, a_{21}, a_{22}$  - некоторые действительные числа. Ясно, что для невырожденной матрицы (12.24) всегда должно выполняться условие:

$$\det A = a_{11}a_{22} - a_{12}a_{21} \neq 0. \quad (12.25)$$

Обратная матрица  $A^{-1}$  может быть получена из матрицы  $A$  следующим образом:

$$A^{-1} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{-1} = \frac{1}{\det A} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix} \quad (12.26)$$

Ясно, что выражение (12.26) имеет смысл только для случая  $\det A \neq 0$ , то есть, когда матрица  $A$  является невырожденной.

**Матрицы Фибоначчи.** В главе 2 мы рассмотрели так называемые  $Q$ -матрицы Фибоначчи, изученные в книге [3], и  $Q_p$ -матрицы Фибоначчи, введенные в статье [84]. Напомним основные свойства этих матриц.

Под  $Q$ -матрицей Фибоначчи понимается простейшая квадратная матрица типа

$$Q = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad (12.27)$$

детерминант которой равен:

$$\det Q = -1 \quad (12.28)$$

Если  $Q$ -матрицу (12.24) возвести в  $n$ -ю степень, то получаем следующую квадратную матрицу:

$$Q^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} \quad (12.29)$$

элементами которой являются числа Фибоначчи  $F_{n-1}, F_n, F_{n+1}$ .

Легко доказать, что выражение для детерминанта этой матрицы

$$\det(Q^n) = F_{n-1}F_{n+1} - F_n^2 = (-1)^n \quad (12.30)$$

совпадает с формулой Кассини.

В главе 2 выведены также выражения для обратных  $Q$ -матриц Фибоначчи, которые зависят от четности степени матрицы (12.26) :

$$Q^{-2k} = \begin{bmatrix} F_{2k-1} & F_{2k} \\ F_{2k} & F_{2k+1} \end{bmatrix} \quad (12.31)$$

$$Q^{-2k-1} = \begin{bmatrix} F_{2k} & F_{2k+1} \\ F_{2k+1} & F_{2k+2} \end{bmatrix} \quad (12.32)$$

Напомним также, что при заданном  $p = 0, 1, 2, 3, \dots$  под  $Q_p$ -матрицей Фибоначчи [84] понимается квадратная  $(p+1) \times (p+1)$ - матрица следующего вида:

$$Q_p = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad (12.33)$$

Если возвести матрицу (12.33) в  $n$ -ю степень, то обнаруживается ее связь с  $p$ -числами Фибоначчи:

$$Q_p^n = \begin{bmatrix} F_p(n+1) & F_p(n) & \cdots & F_p(n-p+2) & F_p(n-p+1) \\ F_p(n-p+1) & F_p(n-p) & \cdots & F_p(n-2p+2) & F_p(n-2p+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ F_p(n-1) & F_p(n-2) & \cdots & F_p(n-p) & F_p(n-p-1) \\ F_p(n) & F_p(n-1) & \cdots & F_p(n-p+1) & F_p(n-p) \end{bmatrix} \quad (12.34)$$

Наиболее неожиданным свойством матриц (12.33) и (12.34) является удивительно простые выражения для их детерминантов:

$$\det Q_p = (-1)^p \quad (12.35)$$

$$\det Q_p^n = (-1)^{pn}, \quad (12.36)$$

где  $p = 0, 1, 2, 3, \dots; n = 0, \pm 1, \pm 2, \pm 3, \dots$ .

Заметим, что все рассмотренные выше  $Q$ -матрицы и  $Q_p$ -Фибоначчи являются невырожденными матрицами, поскольку их детерминанты равны либо  $(+1)$  либо  $(-1)$ .

**Метод кодирования-декодирования, основанный на матрицах Фибоначчи.** Разобьем исходное сообщение  $M$  на 4 части  $M = m_1 m_2 m_3 m_4$  и представим его в виде квадратной  $(2 \times 2)$ -матрицы:

$$M = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix}. \quad (12.37)$$

Заметим, что элементы матрицы  $M$  являются целыми неотрицательными числами. Потребуем, чтобы элементы  $m_1, m_2, m_3, m_4$  были выбраны таким образом, чтобы матрица  $M$  была невырожденной матрицей с детерминантом

$$\det M = m_1 m_2 - m_3 m_4 \neq 0. \quad (12.38)$$

Будем использовать для кодирования  $Q$ -матрицу Фибоначчи  $Q^n$ , задаваемую (12.29). Для декодирования будем использовать обратные  $Q$ -матрицы Фибоначчи  $Q^{-n}$ , задаваемые (12.31) или (12.32). Тогда процесс кодирования-декодирования состоит в следующем:

Кодирование	Декодирование	(12.39)
$M \times Q^n = E$	$E \times Q^{-n} = M$	

Таким образом, кодирование состоит в умножении исходной матрицы (12.37) на кодирующую матрицу  $Q^n$ ; в результате мы получаем кодовую матрицу  $E$ :

$$E = M \times Q^n = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \times \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix}, \quad (12.40)$$

элементы которой задаются следующими выражениями:

$$\begin{aligned} e_1 &= F_{n+1}m_1 + F_n m_2; & e_2 &= F_n m_1 + F_{n-1}m_2; \\ e_3 &= F_{n+1}m_3 + F_n m_4; & e_4 &= F_n m_3 + F_{n-1}m_4. \end{aligned} \quad (12.41)$$

Заметим, что при  $n \geq 1$  числа Фибоначчи всегда являются неотрицательными целыми числами; тогда из (12.37) вытекает, что элементы кодовой матрицы  $e_1, e_2, e_3, e_4$  всегда являются целыми неотрицательными числами.

Декодирование состоит в умножении кодовой матрицы  $E$  на декодирующую матрицу  $Q^{-n}$ . После декодирования, исходная матрица  $M$  получается из кодовой матрицы  $E$  в результате следующих преобразований:

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} = E \times Q^{-n} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \times Q^{-n}. \quad (12.42)$$

Для случая  $n = 2k + 1$  мы можем использовать в качестве декодирующей обратную матрицу (12.32) и тогда формула (12.42) может быть представлена в следующем виде:

$$M = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \\ e_3 & e_4 \end{pmatrix} \times \begin{pmatrix} -F_{n-1} & F_n \\ F_n & -F_{n+1} \end{pmatrix}. \quad (12.43)$$

Из (12.43) вытекает, что элементы матрицы  $M$  могут быть выражены через элементы кодовой матрицы  $E$  следующим образом:

$$\begin{aligned} m_1 &= -F_{n-1}e_1 + F_n e_2; & m_2 &= F_n e_1 - F_{n+1}e_2; \\ m_3 &= -F_{n-1}e_3 + F_n e_4; & m_4 &= F_n e_3 - F_{n+1}e_4. \end{aligned} \quad (12.44)$$

**Основные контрольные соотношения.** Вычислим детерминант кодовой матрицы  $E = M \times Q^n$ :

$$\det E = \det(M \times Q^n) = \det M \times \det Q^n. \quad (12.45)$$

Тогда, используя тождество (12.30), мы можем записать:

$$\boxed{\det E = \det M \times (-1)^n}. \quad (12.46)$$

Это и есть основное контрольное соотношение для нового метода избыточного кодирования, задаваемого таблицей (12.39). Его суть состоит в том, что детерминант кодовой матрицы  $E$  полностью определяется детерминантом исходной матрицы  $M$ ; при этом при четном  $n = 2k$  детерминанты матриц  $E$  и  $M$  совпадают, а при нечетном  $n = 2k + 1$  – противоположны по знаку.

Для вывода еще одного контрольного соотношения обратимся к выражению (12.41), связывающему элементы матрицы  $E$  с элементами матрицы  $M$ . Так как элементы  $m_1 m_2 m_3 m_4$  исходной матрицы (12.37) всегда являются неотрицательными целыми числами, мы можем переписать равенства (12.44) в виде следующих неравенств:

$$-F_{n-1}e_1 + F_n e_2 \geq 0 \quad (12.47)$$

$$F_n e_1 - F_{n+1} e_2 \geq 0 \quad (12.48)$$

$$-F_{n-1}e_3 + F_n e_4 \geq 0 \quad (12.49)$$

$$F_n e_3 - F_{n+1} e_4 \geq 0. \quad (12.50)$$

Неравенства (12.47) и (12.48) могут быть переписаны следующим образом:

$$\frac{F_{n+1}}{F_n} e_2 \leq e_1 \leq \frac{F_n}{F_{n-1}} e_2. \quad (12.51)$$

По аналогии, мы можем переписать неравенства (12.49) и (12.50) следующим образом:

$$\frac{F_{n+1}}{F_n} e_4 \leq e_3 \leq \frac{F_n}{F_{n-1}} e_4. \quad (12.52)$$

Так как отношение соседних чисел Фибоначчи стремится к «золотой пропорции»  $\Phi = \frac{1+\sqrt{5}}{2}$  по мере увеличения индекса  $n$  в (12.51) и (12.52), из неравенств (12.51) и (12.52) вытекают следующие приближенные равенства, которые связывают элементы кодовой матрицы (12.40):

$$\boxed{e_1 \approx \Phi e_2; \quad e_3 \approx \Phi e_4}, \quad (12.53)$$

где  $\Phi = \frac{1+\sqrt{5}}{2}$  - «золотая пропорция». Заметим, что приближенные равенства (12.53) становятся все более точными по мере увеличения степени  $n$  кодирующей матрицы  $Q^n$ .

По аналогии для случая  $n = 2k$  мы можем использовать «обратную» матрицу (12.31) в качестве декодирующей и затем мы можем вывести приближенные равенства, подобные (12.53) для соседних элементов  $e_1$  и  $e_3$ ,  $e_2$  и  $e_4$  кодовой матрицы (12.40).



**Понятие «ошибки».** Таким образом, метод кодирования-декодирования, задаваемый таблицей (12.39), обладает важными контрольными соотношениями (12.46) и (12.53), которые могут быть использованы для эффективного обнаружения и исправления «ошибок», возникающих в кодовой матрице  $E$  в процессе ее передачи по каналу связи. Для проверки этих контрольных соотношений детерминант исходной матрицы  $\det M$  должен быть направлен в канал связи вслед за элементами кодовой матрицы  $E$ .

Прежде всего, мы должны уточнить понятие «ошибки», используемое в новом методе кодирования. Заметим, что в данном случае мы имеем дело не с битами, которые являются основными информационными объектами обнаружения и исправления ошибок в классической теории избыточных кодов [115], а с числами, которые являются элементами кодовой матрицы  $E$ . При этом под «ошибкой» понимается отклонение элемента матрицы от его номинального значения. Например, если один из элементов кодовой матрицы, вычисленный согласно (12.40), равен числу 235, то любое его отклонение от этого значения, которое может возникнуть в канале связи под воздействием «помех», может рассматриваться как «ошибка». Ясно, что в кодовой  $(2 \times 2)$ -матрице  $E$  могут возникать одиночные, двойные, тройные и четверные ошибки в указанном выше смысле.

Одиночные ошибки

$$\begin{pmatrix} x & e_2 \\ e_3 & e_4 \end{pmatrix}; \begin{pmatrix} e_1 & y \\ e_3 & e_4 \end{pmatrix}; \begin{pmatrix} e_1 & e_2 \\ z & e_4 \end{pmatrix}; \begin{pmatrix} e_1 & e_2 \\ e_3 & v \end{pmatrix}, \quad (12.54)$$

Двойные ошибки

$$\begin{pmatrix} x & y \\ e_3 & e_4 \end{pmatrix}; \begin{pmatrix} x & e_2 \\ z & e_4 \end{pmatrix}; \begin{pmatrix} x & e_2 \\ e_3 & v \end{pmatrix}; \begin{pmatrix} e_1 & y \\ z & e_4 \end{pmatrix}; \begin{pmatrix} e_1 & y \\ e_3 & v \end{pmatrix}; \begin{pmatrix} e_1 & e_2 \\ z & v \end{pmatrix}. \quad (12.55)$$

Тройные ошибки

$$\begin{pmatrix} x & y \\ z & e_4 \end{pmatrix}; \begin{pmatrix} e_1 & y \\ e_3 & v \end{pmatrix}; \begin{pmatrix} x & e_2 \\ z & v \end{pmatrix}; \begin{pmatrix} e_1 & y \\ z & v \end{pmatrix}. \quad (12.56)$$

Четверная ошибка

$$\begin{pmatrix} x & y \\ z & v \end{pmatrix}. \quad (12.57)$$

**Обнаружение и исправление ошибок.** Для обеспечения возможности обнаружения и исправления ошибок в канал связи вслед за элементами кодовой матрицы  $e_1, e_2, e_3, e_4$  посылается «контрольный элемент»  $\det M$ . Этот

«контрольный элемент» вносит наибольший вклад в кодую избыточность метода кодирования. Как показано в [85], нижняя оценка относительной избыточности составляет 0,33 (33%), что соизмеримо, например, с избыточностью (15,11)-кода Хемминга.

Для обнаружения ошибок используется «основное контрольное соотношение» (12.46), связывающее информационную матрицу  $M$  с кодовой матрицей  $E$ . Для этого с использованием элементов  $e_1, e_2, e_3, e_4$ , полученных на выходе канала связи, на приемной стороне вычисляется детерминант кодовой матрицы

$$\det E = e_1 \times e_4 - e_2 \times e_3, \quad (12.58)$$

который затем сравнивается с «контрольным элементом»  $\det M$  в соответствии с соотношением (12.46). Если соотношение (12.46) выполняется, то это является признаком отсутствия ошибки в кодовой матрице. В противном случае считается, что в кодовой матрице  $E$  или в «контрольном элементе»  $\det M$  возникла ошибка, которая должна быть исправлена.

Для исправления ошибок используются контрольные соотношения (12.46) и (12.53). При этом основой для исправления служит «контрольный элемент»  $\det M$ , с использованием которого могут быть вычислены «номинальные значения» искаженных элементов матрицы. Именно поэтому должны быть предприняты дополнительные меры для защиты «контрольного элемента»  $\det M$ . Например, мы можем ввести дополнительную кодую избыточность в элемент  $\det M$ , чтобы обнаруживать ошибки в элементе  $\det M$  с использованием классических избыточных кодов.

Основная идея исправления ошибок состоит в следующем. Если мы убедились в «правильности» элемента  $\det M$ , то после этого мы можем выдвинуть некоторые гипотезы относительно характера возможных ошибок в элементах кодовой матрицы  $E$ . Наиболее вероятной является гипотеза о наличии «одиночной» ошибки в кодовой матрице  $E$ . В этом случае мы должны рассмотреть четыре гипотезы, задаваемые (12.54). Если наша гипотеза об «одиночной» ошибке верна, одна из «одиночных» ошибок (12.54) должна соответствовать реальной ситуации. Для нахождения «правильной» гипотезы из четырех гипотез, задаваемых (12.54), мы можем использовать «основное контрольное соотношение» (12.46). Из него вытекают четыре алгебраических уравнения, которые связывают элементы матриц (12.54):

$$xe_4 - e_2e_3 = (-1)^n \det M (e_1 - \text{ошибочный элемент}) \quad (12.59)$$

$$e_1e_4 - ye_3 = (-1)^n \det M (e_2 - \text{ошибочный элемент}) \quad (12.60)$$

$$e_1e_4 - e_2z = (-1)^n \det M (e_3 - \text{ошибочный элемент}) \quad (12.61)$$

$$e_1v - e_2e_3 = (-1)^n \det M (e_4 - \text{ошибочный элемент}), \quad (12.62)$$

откуда вытекает четыре формулы для вычисления возможных «одиночных» ошибок:

$$x = \frac{(-1)^n \det M + e_2e_3}{e_4} \quad (12.63)$$

$$y = \frac{-(-1)^n \det M + e_1e_4}{e_3} \quad (12.64)$$

$$z = \frac{-(-1)^n \det M + e_1e_4}{e_2} \quad (12.65)$$

$$v = \frac{(-1)^n \det M + e_2e_3}{e_1}. \quad (12.66)$$

При этом мы должны учесть, что элементы  $e_1, e_2, e_3, e_4$  кодовой матрицы  $E$  всегда являются целыми числами! Это условие является дополнительным «контрольным соотношением», то есть, мы должны выбирать «правильные» решения  $x, y, z, v$  только среди целочисленных значений. Это означает, что уравнения (12.59)-(12.62) являются Диофантовыми! Если ни одно из решений (12.59)-(12.62) не является целым числом, это означает, что наша гипотеза об «одиночных» ошибках является некорректной, и дальше мы должны рассмотреть гипотезы о «двойных» или «тройных» ошибках в кодовой матрице  $E$ .

Поясним, что такое «Диофантово уравнение». Такое название происходит от имени древнегреческого математика Диофанта, который впервые начал рассматривать алгебраические уравнения или системы алгебраических уравнений с целыми коэффициентами, имеющие число неизвестных, превосходящее число уравнений, и у которых разыскиваются целые или рациональные решения.

Диофантовы уравнения называются также неопределёнными. Простейшее Диофантово уравнение  $ax + by = 1$ , где  $a$  и  $b$  — целые взаимно простые числа, имеет бесконечно много решений: если  $x_0$  и  $y_0$  — одно решение, то числа  $x = x_0 + bn$ ,  $y = y_0 - an$  ( $n$  — любое целое число) тоже будут решениями. Другим примером Диофантова уравнения является  $x^2 + y^2 = z^2$ . Целые положительные решения этого уравнения представляют собой длины катетов  $x, y$  и гипотенузы  $z$

прямоугольных треугольников с целочисленными длинами сторон и называются пифагоровыми числами.

Таким образом, метод избыточного кодирования, основанный на использовании матриц Фибоначчи, обладает тремя «контрольными соотношениями», справедливыми для кодовой матрицы  $E$ :

- (1) основное контрольное соотношение (12.46) задает связь детерминантов исходной и кодовой матриц  $M$  и  $E$ ;
- (2) дополнительные контрольные соотношения (12.53) задают приближенные равенства между элементами кодовой матрицы  $E$ ;
- (3) наконец, третье контрольное соотношение состоит в том, что уравнения (12.59)-(12.62) являются Диофантовыми.

Однако, может возникнуть ситуация, когда не одно, а несколько из уравнений (12.59)-(12.62) приводят к целочисленным решениям. Это означает, что сразу несколько матриц типа (12.54) могут претендовать на роль «корректных» матриц. Как быть в этой ситуации? Здесь на помощь нам приходят дополнительные контрольные соотношения (12.53). Вычисляя для каждой из таких матриц приближенные равенства (12.53), мы можем выбрать ту «корректную» матрицу, для которой приближенные равенства выполняются в наибольшей степени.

**Коррекция двойных и тройных ошибок.** Эти же контрольные соотношения могут быть использованы для коррекции «двойных» и «тройных» ошибок. В случае установления ошибочности гипотезы об «одиночной» ошибке в кодовой матрице  $E$  (все формулы типа (12.63)-(12.66) не приводят к целым числам), мы можем рассмотреть гипотезу о «двойных» ошибках в кодовой матрице  $E$ . Согласно (12.55) количество таких «ошибок» равно 6. Для выяснения того, какой из этих вариантов соответствует реальной ситуации, мы используем «основное контрольное соотношение» (12.46) для составления 6 алгебраических уравнений с двумя неизвестными. Например, «двойной» ошибке типа

$$E = \begin{pmatrix} x & e_2 \\ e_3 & v \end{pmatrix}, \quad (12.67)$$

соответствует следующее алгебраическое уравнение:

$$xv - e_2e_3 = \det M (-1)^n. \quad (12.68)$$

Как упоминалось выше, хотя бы одно уравнение из 6 возможных алгебраических уравнений, соответствующих «двойным» ошибкам (12.55),

должно быть Диофантовым, то есть, иметь целочисленные решения. В этом случае среди всех возможных Диофантовых решений этого уравнения, мы должны выбрать решение, которое в наибольшей степени удовлетворяет дополнительным контрольным соотношениям (12.53). Для случая (12.67) между элементами матрицы (12.68) должны выполняться следующие приближенные равенства:

$$x \approx \Phi e_2; \quad e_3 \approx \Phi v.$$

Таким же путем может быть проверена гипотеза о «тройной» ошибке в кодовой матрице  $E$ .

**О корректирующей способности предлагаемого метода избыточного кодирования.** Из проведенных рассуждений вытекает, что при малых значениях  $n$  приближенные равенства (12.53) являются слишком «грубыми» и их использование для оценки соотношений между элементами кодовой матрицы  $E$ , задаваемое (12.53), является слишком «грубым». Поэтому необходимо увеличивать значение степени  $n$  кодирующей матрицы (12.29). Однако, увеличение  $n$  приводит к увеличению избыточности данного метода кодирования. Поэтому возникает проблема выбора оптимального значения  $n$ , которое бы, с одной стороны давало бы высокую достоверность правильности выбора «корректной» матрицы среди других матриц, но, с другой стороны, обеспечивало бы приемлемую избыточность. При этом для случая возможных одиночных ошибок метод гарантирует (правильнее сказать, обеспечивет высокую вероятность) «правильного» исправления любой из 4 «одиночных» ошибок из возможных 15, задаваемых (12.54)-(12.57). В этом случае мы можем оценить «корректирующую способность» предложенного метода с помощью соотношения:

$$S_{cor} = \frac{4}{15} = 0.2667 = 26.67\%. \quad (12.69)$$

Мало это или много? Для ответа на этот вопрос достаточно сравнить значение (12.69) с потенциальной корректирующей способностью (15,11)-кода Хемминга, равной 0.0004882 (0.04882%). Отношение  $A = \frac{0.2667}{0.0004882} \approx 546$  дает представление о довольно внушительном преимуществе предложенного метода по сравнению с (15,11)-кодом Хемминга.

Таким образом, общая идея коррекции ошибок в предлагаемом методе избыточного кодирования основана на поиске «правильных» ошибочных ситуаций типа (12.64)-(12.66) с использованием сформулированных выше трех контрольных условий.

Вопрос о выборе оптимального значения  $n$  для практических приложений данного метода является предметом дальнейших научных исследований.

### **Преимущества и области применения нового способа кодирования.**

Основным преимуществом предложенного метода кодирования является существенное повышение корректирующей способности (на несколько порядков) по сравнению с классическими алгебраическими кодами.

Другое преимущество состоит в том, что метод позволяет исправлять не биты и их сочетания («пачки ошибок»), а элементы матриц, которые могут быть числами достаточно большой величины!

Как известно, в современных компьютерных системах алгебраические корректирующие коды в действительности не используются для коррекции ошибок, потому что корректирующая способность этих кодов ничтожно мала. В реальных приложениях алгебраические коды (в частности, циклические коды) широко используются для обнаружения ошибок; при этом исправление обнаруженных ошибок осуществляется по «принципу обратной связи», когда к передатчику по каналу обратной связи направляется «запрос» с просьбой повторить сообщение; это продолжается до тех пор, пока не будет получено сообщение без ошибок. Этот способ исправления ошибок является эффективным, когда расстояние между передатчиком и приемником сравнительно невелико и временная задержка, связанная с направлением «запроса» по каналу обратной связи, не влияет существенно на качество передачи информации. Однако, существуют области, для которых применение «принципа обратной связи» становится проблематичным. Речь идет, прежде всего, о космических системах связи, когда расстояние между передатчиком и приемником увеличивается до космических масштабов и задержка в передаче «запроса» по каналу обратной связи увеличивается до нескольких минут и даже часов. Поэтому метод может быть использован для передачи особо важных сообщений на космические расстояния. Область космических систем является той реальной областью, где применение нового метода кодирования, основанного на матрицах Фибоначчи, может оказаться эффективным.

Ясно, что метод может иметь и много других приложений, одним из которых является создание отказустойчивой памяти компьютеров. Эти и другие приложения потребуют дополнительных исследований, так же, как и использование более сложных кодирующих «матриц Фибоначчи», задаваемых, например, (12.33), (12.34). Автор надеется, что развитие нового метода избыточного кодирования станет привлекательным направлением для многих молодых исследователей в этой области.





## Послесловие ко второй части

Настоящая книга является второй частью трехтомной книги автора «Основы математики гармонии и ее приложения». Главная цель трехтомной книги – популяризация «математики гармонии» как нового междисциплинарного направления современной науки и ее приложений в современной науке. При этом главной целью второй части является обсуждение приложений «математики гармонии» в информатике. В центре этой книги – поиски ответа на вопрос: являются ли коды Фибоначчи и коды золотой пропорции альтернативой классической двоичной системе счисления?

Главная задача второй части настоящей книги состоит в том, чтобы провести критический анализ применения классической двоичной системы счисления в специализированных компьютерах, используемых в системах управления без участия человека.

Основной тезис состоит в следующем. Классическая двоичная система счисления обладает **НУЛЕВОЙ ИЗБЫТОЧНОСТЬЮ** и в ней отсутствуют механизмы, позволяющие обнаруживать сбои в электронных элементах, что уже привело к катастрофам национального уровня в ракетной технике (Россия, США и другие страны). Таким образом, человечество становится заложником классической двоичной системы счисления, которая лежит в основе современных микропроцессоров и информационных технологий. Современные микропроцессоры, основанные на двоичной системе счисления, ненадежны с информационной точки зрения. Поэтому дальнейшее развитие микропроцессорной техники и основанной на ней информационной технологии на основе классической двоичной системы счисления применительно к специальным областям приложений следует считать бесперспективным направлением. Двоичная система не может служить информационной и арифметической основой специализированных компьютерных и измерительных систем (космос, управление транспортом и сложными технологическими объектами), а также наноэлектронных систем, где проблемы надежности, помехоустойчивости, контролеспособности, стабильности, живучести систем выходят на передний план».

Главный вывод из проведенного исследования состоит в следующем.

Коды Фибоначчи и золотой пропорции сохраняют все известные преимущества классической двоичной системы счисления, но при этом обладают избыточностью (44%), достаточной, чтобы осуществлять эффективный и непрерывный контроль всех преобразований информации в компьютерах и микропроцессорах; они являются альтернативой классической двоичной системе счисления и должны стать основой при проектировании специализированных компьютеров и микропроцессоров повышенной информационной надежности и помехоустойчивости.

К этому выводу автор шел в течение более 40 лет научной работы, результатом которой стало создание нескольких прикладных математических теорий, которые затрагивают основания современной математики и информатики, написание ряда книг в этой области [9,10,12,18,37,42,47,126], уникальное по своим масштабам патентование изобретений по направлению «компьютеры Фибоначчи» за рубежом (65 патентов США, Японии, Англии, Франции, ФРГ, Канады и др. стран). Эти патенты защищают приоритет советской науки (и автора настоящей книги) в области «компьютеров Фибоначчи». Часть 2 книги написана с использованием в основном оригинальных прикладных математических и технических результатов автора. При этом необходимо отметить наиболее важные научные результаты настоящего исследования:

**Алгоритмическая теория измерения.** Толчком к созданию этой теории стали практические задачи, направленные на улучшение технических характеристик аналого-цифровых преобразователей алгоритмическим путем. Решение этих задач привело в 60-е годы 20 в. к разработке математической теории оптимальных алгоритмов аналого-цифрового преобразования [57]. Хотя на начальном этапе эти исследования не выходили за узкие рамки аналого-цифровых преобразователей [57], дальнейшие исследования показали, что речь идет о новой математической теории измерения – алгоритмической теории измерения [9], которая затрагивает основания математики, информатики и теоретической метрологии.

Алгоритмическая теория измерения является новой и необычной для математики теорией, направленной на изучение алгоритмов измерения; такая задача по существу никогда не являлась предметом серьезных математических исследований (хотя постановка этой задачи содержится в «задаче о выборе

наилучшей системы гирь», сформулированной Фибоначчи еще в 13 в. и известной в современной науке под названием задачи Баше-Менделеева).

**Классические алгоритмы измерения как фундамент математики и информатики.** Как известно [52], математика была создана под влиянием двух практических задач – счета и измерения. Простейшие методы счета и измерения и были положены в основу ее фундаментальных понятий, одним из которых является понятие натурального числа. В течение более чем двух тысячелетий математика (а вслед за ней и информатика) вполне удовлетворялась в основном двумя классическими алгоритмами измерения, широко используемыми в практике измерений. Первый из них – это алгоритм счета, который исторически лежит в основе «Евклидоваго определения» натурального числа и «элементарной теории чисел», а также аксиомы измерения (аксиомы Евдокса-Архимеда), которая лежит в основе математической теории измерения. Таким образом, этот алгоритм исторически порождает две фундаментальные математические теории – «элементарную теорию чисел» и «математическую теорию измерения». Вторым алгоритмом – это «двоичный» алгоритм или алгоритм поразрядного кодирования, который имеет фундаментальную значимость для информатики, поскольку порождает двоичную систему счисления – основу современных компьютеров.

**Биномиальные алгоритмы измерения.** В рамках алгоритмической теории измерения получен необычный математический результат – синтезирован новый класс «оптимальных» алгоритмов измерения, основанных на таблице биномиальных коэффициентов, называемой «арифметическим квадратом». Поскольку данный алгоритм является обобщением алгоритма счета, лежащего в основе элементарной теории чисел, то вполне правомерным является постановка следующего вопроса: не могут ли биномиальные алгоритмы измерения способствовать развитию «элементарной теории чисел»? Прикладное значение биномиальных алгоритмов состоит в том, что на их основе уже созданы биномиальные системы счисления, которые использованы для создания биномиальных счетчиков и биномиальных контроллеров [97].

**Принцип асимметрии измерения, фибоначчиевые алгоритмы измерения и новая машинная арифметика.** Анализ «двоичного» алгоритма взвешивания на рычажных весах (задача Баше-Менделеева) привел к

формулировке принципа асимметрии измерения [58]. Решение задачи синтеза оптимальных алгоритмов измерения с учетом сформулированного принципа привело к открытию так называемых «фибоначчиевых алгоритмов измерения», в которых «оптимальная» система гирь задается с помощью  $p$ -чисел Фибоначчи. Данные алгоритмы измерения являются обобщением «двоичного» алгоритма, который порождает двоичную систему счисления. Такой взгляд на фибоначчиевые алгоритмы измерения привел к открытию новых способов двоичного позиционного представления натуральных чисел –  $p$ -кодов Фибоначчи и созданию новой машинной арифметики – «арифметики Фибоначчи» [59-61,65,67].

**Изоморфизм между «размножением кроликов» и «принципом асимметрии измерения».** Установлен изоморфизм между «размножением кроликов» и «принципом асимметрии измерения». Этот изоморфизм привел к обобщению задачи о «размножении кроликов» и формулировке обобщенного «принципа асимметрии размножения» в живой природе. Согласно этому принципу, размножение в живой природе является асимметричным и подчиняется математической закономерности  $p$ -чисел Фибоначчи. Этот факт подтверждается публикацией американских исследователей [102].

**Изоморфизм между алгоритмической теорией измерения и позиционными системами счисления.** Подобно тому, как алгоритм счета порождает способ представления натуральных чисел, известный под названием «Евклидоваго определения» натурального числа, а «двоичный» алгоритм – классическую двоичную систему, «оптимальные» алгоритмы измерения, полученные в рамках алгоритмической теории измерения [9], порождают все известные позиционные системы счисления (включая вавилонскую 60-ричную систему, десятичную и двоичную). Но самое главное, что новые «оптимальные» алгоритмы измерения порождают новые позиционные системы счисления, которые были неизвестны в математике и информатике, в частности, биномиальные алгоритмы порождают биномиальные системы [97], а фибоначчиевые алгоритмы – фибоначчиевые системы счисления [59-61,65,67].

**Система счисления Бергмана как выдающееся математическое достижение в области позиционных систем счисления.** В 1957 г. юный (12-

летний) американский вундеркинд Джордж Бергман в виде математической игры предложил необычную двоичную позиционную систему счисления, основанием которой является золотая пропорция [55]. Система Бергмана является первой в истории науки системой счисления с иррациональным основанием. Система Бергмана является крупнейшим современным открытием в области систем счисления, сравнимым разве что с открытием позиционного принципа представления чисел, а также десятичной и двоичной систем счисления. Она переворачивает наши представления о системах счисления; более того - соотношение между рациональными и иррациональными числами. В этой системе на первый план выдвигается иррациональное число «золотая пропорция», которое становится основанием всех чисел, так как с его помощью может быть представлено любое действительное число!

**Коды золотой  $p$ -пропорции как обобщение системы Бергмана и классической двоичной системы счисления.** Обобщение системы Бергмана дано в работах автора настоящей книги [12,56]. Речь идет о так называемых кодах золотой  $p$ -пропорции, которые включают в себя в качестве частных случаев классическую двоичную систему счисления ( $p = 0$ ) и систему Бергмана ( $p = 1$ ). Это обобщение расширяет количество систем счисления с иррациональными основаниями до бесконечности и открывает новые пути для развития теории чисел и компьютерной арифметики.

**«Золотая» теория чисел.** Система Бергмана и коды золотой  $p$ -пропорции можно рассматривать как новые конструктивные определения действительных чисел и такой подход естественным образом приводит к так называемой «золотой» теории чисел [117]. В ее рамках получено ряд необычных математических результатов, касающихся, например, натуральных чисел. В частности, доказано, что представления натуральных чисел в системе Бергмана и любом коде золотой  $p$ -пропорции являются конечными, то есть, любое натуральное число может быть представлено в виде конечной суммы степеней золотой  $p$ -пропорции с двоичными коэффициентами  $\{0,1\}$  (такое представление натурального числа называется  $\Phi$ -кодом или  $\Phi_p$ -кодом). Для любого натурального числа  $N$  его  $\Phi$ -код (или  $\Phi_p$ -код) обладает уникальным свойством,

названным  $Z$ -или  $Z_p$ -свойством, соответственно. Его суть состоит в следующем. Если в  $\Phi$ -коде все степени золотой пропорции  $\Phi^i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$  заменить на соответствующее числа Фибоначчи  $F_i$ , либо в  $\Phi_p$ -коде все степени золотой  $p$ -пропорции  $\Phi_p^i (i = 0, \pm 1, \pm 2, \pm 3, \dots)$  заменить на соответствующие  $p$ -числа Фибоначчи  $F_p(i)$ , то возникающая при этом сумма тождественно равна 0 независимо от исходного натурального числа  $N$ . Поскольку это справедливо только для натуральных чисел, то  $Z$ -и  $Z_p$ -свойства можно рассматривать, как новые свойства натуральных чисел. Это означает, что изучение систем счисления с иррациональными основаниями привело к открытию новых свойств натуральных чисел, которые изучаются в математике свыше двух тысячелетий.

**Патентование изобретений в области «компьютеров Фибоначчи».** В 70-80-е годы в Советском Союзе проведено беспрецедентное по своим масштабам патентование изобретений в области «компьютеров Фибоначчи» за рубежом. 65 патентов США, Японии, Англии, Франции, Канады и др. стран [70-83] являются официальными юридическими документами, подтверждающими советский приоритет в этом направлении.

**Микропроцессоры Фибоначчи как одна из базисных инноваций будущего технологического уклада, изменяющих уровень информационной безопасности систем.** Осознание того, что современные двоичные микропроцессоры ненадежны и не обладают достаточной помехоустойчивостью для их использования в вычислительных комплексах специализированных систем управления (академик Я.А. Хетагуров) является одним из не очень радостных итогов развития компьютерной техники, основанной на классической двоичной системе счисления. Именно поэтому создание микропроцессоров Фибоначчи, основанных на кодах Фибоначчи и золотой пропорции, обладающих избыточностью, необходимой для обнаружения ошибок, является актуальной задачей на современном этапе развития информационных технологий. Использование "золотого сечения" в современных информационных технологиях означает возврат к Пифагору, Платону и Евклиду. Это означает использование «естественных законов природы» для улучшения информационных технологий. Важно подчеркнуть, что это направление родилось в Советском Союзе

(Таганрогский радиотехнический институт и Винницкий политехнический институт).

**Счетчик Фибоначчи для минимальной формы как начало нового этапа в развитии «компьютеров Фибоначчи» и основа помехоустойчивых микроконтроллеров для специализированных систем управления.** Публикация англоязычной книги автора [47] вызвало новую волну интереса к идее «компьютеров Фибоначчи». Исследования в этой области, проведенные на кафедре электроники и компьютерной техники Сумского государственного университета, привели к разработке уникальной структуры счетчика Фибоначчи (разработчики – Алексей Борисенко и Алексей Стахов) [127]. Отличительная особенность этого счетчика состоит в том, что счет единиц осуществляется только в минимальной форме. Любые другие фибоначчиевые представления в счетчике запрещены, что и является принципиально новым свойством помехоустойчивого счетчика, в котором, за счет избыточности кода Фибоначчи, обнаруживается свыше 99% всех возможных ошибок. Предлагаемый счетчик является помехоустойчивым, достаточно быстродействующим, отличается регулярной структурой и высокой надежностью. Поэтому его имеет смысл использовать для реализации в виде ПЛИС в различных схемах цифровых устройств, требующих высокого быстродействия, помехоустойчивости и надежности, таких, например, как частотомеры и управляющие устройства.

Однако, разработанный счетчик, который представляет собой патентно-чистую структуру, открывает еще один путь для развития «компьютеров Фибоначчи». Речь идет о создании микроконтроллеров Фибоначчи на таких счетчиках. Известно, что теоретически любое вычислительное устройство или микроконтроллер может быть построен на счетчиках, то есть, базовой структурой такого микроконтроллера Фибоначчи является счетчик Фибоначчи. Это резко упрощает техническую реализацию такого микроконтроллера. Обеспечение высокого быстродействия осуществляется за счет использования оригинальных технических решений и выбора быстродействующей элементной базы. Главным преимуществом такого подхода является создание микроконтроллеров для специальных приложений, обеспечивающих непрерывный самоконтроль всех процессов преобразования информации и обеспечивая таким путем высокую помехозащищенность.

**Троичная зеркально-симметричная система счисления.** Эта система счисления является весьма необычной системой счисления по сравнению с традиционными системами счисления (двоичной, троичной, десятичной и др.). Она является синтезом системы Бергмана [55] и троичной симметричной системы счисления [105], использованной Николаем Брусенцовым при создании троичного компьютера «Сетунь». Ее основанием является квадрат «золотой пропорции»  $\Phi^2 = \frac{3+\sqrt{5}}{2}$ , то есть, эта система счисления относится к разряду систем счисления с иррациональными основаниями. Эта система позиционного представления целых чисел обладает уникальным математическим свойством – свойством зеркальной симметрии. В соответствии с этим свойством представление любого целого числа разбивается на две зеркально-симметричные части относительно нулевого разряда; это свойство является инвариантом относительно всех арифметических операций, то есть, результаты всех арифметических операций (сложение, вычитание, умножение и деление) обладают свойством зеркальной симметрии. Это означает, что найден новый способ контроля всех преобразований информации в компьютере (свойство зеркальной симметрии).

**Троичный код Фибоначчи.** Если в классическом Фибоначчи вместо двоичного алфавита  $\{0,1\}$  использовать троичный алфавит  $\{\bar{1},0,1\}$ , то при этом возникает оригинальная троичная система счисления, которая обладает необходимой избыточностью, достаточной для обеспечения эффективного контроля всех преобразований информации в компьютере. Однако, главное свойство такой троичной арифметики Фибоначчи состоит в том, что сложение двух троичных фибоначиевых представлений, использующих минимальную форму, осуществляется за один такт, что создает предпосылки для создания сверхбыстрых арифметических процессоров.

**Теория избыточного кодирования, основанная на матрицах Фибоначчи.** Как известно, алгебраические корректирующие коды (Хемминга, циклические, турбо-коды и др.) [115] являются основным средством защиты информации в системах передачи и хранения информации. Следует отметить, что потенциальная корректирующая способность этих кодов настолько мала, что на практике эти коды никогда не используются как корректирующие. На самом деле,



в подавляющем количестве приложений эти коды используются как коды, обнаруживающие ошибки. При этом для исправления обнаруженной ошибки используется «принцип обратной связи», согласно которому в случае обнаружения ошибки посылается запрос на повторение сообщения по каналу «обратной связи» и это делается до тех пор, пока сообщение будет принято без ошибок. Поэтому прилагательное «корректирующие», употребляемое в названии «алгебраические корректирующие коды», является, строго говоря, большим преувеличением.

Разработанная автором теория избыточного кодирования, основанная на матрицах Фибоначчи [85], обеспечивает широкие возможности не только для обнаружения ошибок, но и для их коррекции. Суть нового метода кодирования состоит в умножении исходного (информационного) сообщения  $M$ , представленного в матричном виде, на специальную кодирующую матрицу -  $Q$ -матрицу Фибоначчи типа  $Q^n$  [3] или  $Q_p$ -матрицу Фибоначчи типа  $Q_p^n$  [84], в результате чего формируется кодовая матрица  $E$ , направляемая в канал связи. Декодирование кодовой матрицы осуществляется путем ее умножения на обратные матрицы типа  $Q^{-n}$  или  $Q_p^{-n}$ . В качестве «контрольного сообщения» используется детерминант информационной матрицы  $\det M$ , направляемый в канал связи вслед за кодовой матрицей  $E$ . При этом достигается эффект в корректирующей способности, на несколько порядков превышающий корректирующую способность классических алгебраических кодов. Еще одно достоинство «матричного кода Фибоначчи» состоит в том, что объектом коррекции являются не биты, а числа, являющиеся элементами кодовой матрицы  $E$ .

В заключение отметим, что главная цель материалов, изложенных во второй части книги, состоит в том, чтобы разбудить исследовательскую мысль студентов и молодых ученых в использовании новых позиционных систем счисления и новых методов избыточного кодирования в современных компьютерных и телекоммуникационных системах, основанных на использовании «золотой пропорции» и чисел Фибоначчи. Приведенные в книге материалы могут стать источником бакалаврских, магистерских, кандидатских и докторских диссертаций.



## Список литературы

1. Coxeter, H. S. M. Introduction to Geometry New York: John Wiley and Sons, 1961.
2. Воробьев Н.Н. Числа Фибоначчи. М.: Наука, 1984. – 144 с. (первое издание - 1961).
3. Hoggat V. E. Jr. Fibonacci and Lucas Numbers. - Boston, MA: Houghton Mifflin, 1969.
4. Vajda S. Fibonacci & Lucas Numbers, and the Golden Section. Theory and Applications. - Ellis Harwood Limited, 1989.
5. Gardner Martin. Mathematics, Magic and Mystery. New York: Publishing House “Dover”, 1952.
6. Brousseau Alfred. An Introduction to Fibonacci Discovery. San Jose, California: Fibonacci Association, 1965.
7. Huntley H. E. The Divine Proportion: a Study in Mathematical Beauty. Dover Publications, 1970.
8. Ghyka Matila. The Geometry of Art and Life. Dover Publications, 1977.
9. Стахов А.П. Введение в алгоритмическую теорию измерения. М.: Советское Радио, 1977. – 288 с.
10. Стахов А.П. Алгоритмическая теория измерения. М.: Знание, 1979. – 64 с. (Новое в жизни, науке и технике. Серия «Математика и кибернетика», 6).
11. Реньи Альфред. Трилогия математики (пер. с венг.). М.: Мир, 1980. – 376 с.
12. Стахов А.П. Коды золотой пропорции. М.: Радио и связь, 1984. – 152 с.
13. Сороко Э.М. Структурная гармония систем. Минск: Наука и техника, 1984. – 264 с.
14. Grzedzielski Jan. Energetycno-geometryczny kod Przyrody. Warszawa: Warszawskie centrum studenckiego ruchu naukowego, 1986 (in Polen). – 140 p.
15. Шевелев И.Ш. Принцип пропорции. М.: Стройиздат, 1986.- 200 с.
16. Garland T.H. Fascinating Fibonacci: Mystery and Magic in Numbers. Dale Seymour, 1987.
17. Ковалев Ф.В. Золотое сечение в живописи. Киев: Высшая школа, 1989. – 143 с.
18. Стахов А.П. (редактор). Помехоустойчивые коды: Компьютер Фибоначчи, Москва, Знание, серия «Радиоэлектроника и связь», вып.6, 1989 г. – 64 с.
19. Васютинский Н.А. Золотая пропорция. М.: Молодая Гвардия», 1990. – 238 с.
20. Шевелев И.Ш., Марутаев М.А., Шмелев И.П. Золотое сечение. Три взгляда на гармонию природы. М.: Стройиздат, 1990. – 343 с.

21. Runion G.E. The Golden Section. Dale Seymour, 1990.
22. Fisher Robert, Fibonacci Applications and Strategies for Traders. New York: John Wiley & Sons, Inc., 1993.
23. Шмелев И.П. Феномен Древнего Египта. Минск: РИТС, 1993. – 343 с.
24. Боднар О.Я. Золотое сечение и неевклидова геометрия в природе и искусстве. Львов: СВИТ, 1994. – 204 с.
25. Dunlap R.A. The Golden Ratio and Fibonacci Numbers. World Scientific, 1997.
26. Цветков В.Д. Сердце, золотая пропорция и симметрия. Пущино: ОНТИ РНЦ РАУ, 1997. – 155 с.
27. Коробко В.И. Золотая пропорция и проблемы гармонии систем. М.: Изд-во Ассоциации строительных вузов стран СНГ, 1998. – 373.
28. Herz-Fischler, Roger. A Mathematical History of the Golden Number. New York: Dover Publications, Inc., 1998. – 195 p.
29. Vera W. de Spinadel. From the Golden Mean to Chaos. Nueva Libreria, 1998 (second edition, Nobuko, 2004).
30. Gazale Midhat J. Gnomon. From Pharaohs to Fractals. Princeton, New Jersey: Princeton University Press, 1999 (Русский перевод: Мидхат Газале. Гномон. От фараонов до фракталов. Москва-Ижевск: Институт компьютерных исследований, 2002. – 272 с.).
31. Prechter, Robert R. The Wave Principle of Human Social Behaviour and the New Science of Socionomics. Gainesville, Georgia: New Classics Library, 1999.
32. Шевелев И.Ш. Метаязык живой природы. М.: Воскресенье, 2000. – 352 с.
33. Kappraff Jay. Connections. The geometric bridge between Art and Science. Second Edition. Singapore, New Jersey, London, Hong Kong: World Scientific, 2001. – 490 p.
34. Kappraff Jay. “Beyond Measure. A Guided Tour Through Nature, Myth and Number”. Singapore, New Jersey, London, Hong Kong: World Scientific, 2002. – 584 p.
35. Koshy, T. Fibonacci and Lucas Numbers with Applications. New York: Wiley, 2001.
36. Livio, M. The Golden Ratio: The Story of Phi, the World's Most Astonishing Number. New York: Broadway Books, 2002.
37. Стахов А.П. Новая математика для живой природы. Гиперболические функции Фибоначчи и Люка. Винница: ІТІ, 2003. – 264 с.
38. Стахов А.П. Под знаком «Золотого Сечения». Исповедь сына студбатовца. Винница: ІТІ, 2003. – 284 с.

39. Боднар О.Я. Золотий переріз і неевклідова геометрія в науці та мистецтві. Львів: Українські Технології, 2005. – 197 с.
40. Петруненко В.В. Золотое сечение квантовых состояний и его астрономические и физические проявления. Минск: Право и экономика, 2005. – 390 с.
41. Сороко Э.М. Золотое сечение, процессы самоорганизации и эволюции систем. Введение в общую теорию гармонии систем. Москва: URSS, 2006. – 264 с.
42. Стахов А., Слученкова А., Щербаков И. Код да Винчи и ряды Фибоначчи. Санкт-Петербург: Питер, 2006. – 320 с.
43. Olsen Scott. The Golden Section: Nature's Greatest Secret. New York: Walker Publishing Company, 2006. – 58 p.
44. Петухов С.В. Матричная генетика, алгебры генетического кода, помехоустойчивость. Москва-Ижевск: НИЦ «Регулярная и хаотическая динамика», 2008.-316 с.
45. Шевелев И.Ш. Основы гармонии. Визуальные и числовые образы реального мира. М.: Луч, 2009. – 360 с.
46. Южанников А.Ю. Золотое сечение и техноценозов в системах электроснабжения. Красноярск: Поликор, 2009. – 288 с.
47. Stakhov A.P. The Mathematics of Harmony. From Euclid to Contemporary Mathemartics and Computer Science. New Jersey, London, Singapore, Hong Kong: World Scientific, 2009. – 697 p.
48. HRH Charles The Prince of Wales. Harmony: A New Way of Looking at Our World. Harper Publisher, 2010. – 330 p.
49. Аракелян Грант. Теория ЛМФ и принцип золотого сечения. В 4 частях. Академия Тринитаризма, 2011 (электронная публикация).
50. Григорьев Ю., Мартыненко Г. Типология последовательностей Фибоначчи: Теория и приложения. Введение в математику гармонии. LAMBERT Academic Pudlishing Gmbh & Co.KG. Saarbruecken, Germany, 2012. – 298 с.
51. Клайн М. Математика. Утрата определенности (пер. с англ). М.: Мир, 1984. – 434 с.
52. Колмогоров А.Н. Математика в ее историческом развитии. М.: Наука, 1991. – 224 с.
53. Harmony of spheres. The Oxford dictionary of philosophy, Oxford University Press, 1994, 1996, 2005.
54. Dimitrov Vladimir. A new kind of social science. Study of self-organization of human dynamics. Morrisville Lulu Press, 2005.

55. Bergman G. A number system with an irrational base // *Mathematics Magazine*, 1957, No 31: 98-119.
56. Стахов А.П. «Золотая» пропорция в цифровой технике. *Автоматика и вычислительная техника*, №1, 1980 г. – с.27-33.
57. Витенько И.В., Стахов А.П. Теория оптимальных алгоритмов аналого-цифрового преобразования. – В кн. *Приборы и системы автоматизики*, вып. 11. Харьков, Изд-во Харьковского университета, 1970. – с.20-35.
58. Стахов А.П. Принцип асимметрии логики измерения. *Проблемы передачи информации*, №3, 1976 г.
59. Стахов А.П. Избыточные двоичные позиционные системы счисления. В кн. *Однородные цифровые вычислительные и интегрирующие структуры*, вып.2. Изд-во Таганрогского радиотехнического института, 1974 г. – с.5-41
60. Стахов А.П. Использование естественной избыточности «фибоначчиевых» систем счисления для контроля вычислительных систем. *Автоматика и вычислительная техника*, №6, 1975 г. – с.80-87.
61. Стахов А.П. «Фибоначчиевые» двоичные позиционные системы счисления. В сб. *Кодирование и передача дискретных сообщений в системах связи*. Москва, Наука, 1976 г.
62. Стахов А.П. Методологические аспекты введения кодовой избыточности в цифровые вычислительные машины / А.П. Стахов // *Автоматика и вычислительная техника*. – 1976. - , №15. с. 21-30.
63. Стахов А.П. Цифровая метрология в кодах Фибоначчи и кодах золотой пропорции. В сб. *Современные проблемы метрологии*. Москва, Изд-во Всесоюзного заочного машиностроительного института, 1978 г.
64. Стахов А.П. Перспективы применения систем счисления с иррациональными основаниями в технике аналого-цифрового и цифроаналогового преобразования. *Журнал «Измерения, Контроль, Автоматизация»*, №6, 1981 г. –с.3-40.
65. Стахов А.П. Алгоритмическая теория измерения и основания компьютерной арифметики. *Журнал «Измерения, Контроль, Автоматизация»*, №2, 1988 г. – с.12-23.
66. Stakhov A.P. The Golden Section in the measurement theory. *An International Journal «Computers & Mathematics with Applications»*, Volume 17, No 4-6, 1989. с.613-638.

67. Стахов А.П. Алгоритмическая теория измерения: новый взгляд на теорию позиционных систем счисления и компьютерную арифметику. Управляющие системы и машины, №4-5, 1994. – с.25-41.
68. Стахов А.П. Микропроцессоры Фибоначчи - как одна из базисных инноваций будущего технологического уклада, изменяющих уровень информационной безопасности систем. Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах», 2011, №2. – с.105-119.
69. Стахов А.П., Фомичев А.В. Устройство для приведения p-кодов Фибоначчи к минимальной форме. Авторское свидетельство № 662930, Бюллетень изобретений №18, 1979 г.
70. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of USA No 4187500.
71. Device for reduction of p-Fibonacci codes to the minimal form. Patent certificate of USA No 4290051
72. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of England No 1543302.
73. Device for reduction of p-Fibonacci codes to the minimal form. Patent certificate of England No 2050011.
74. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of Germany No 2732008.
75. Device for reduction of p-Fibonacci codes to the minimal form. Patent certificate of Germany No 2921053.
76. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of Japan No 1118407.
77. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificates of France No 7722036, No 2359460.
78. Device for reduction of p-Fibonacci codes to the minimal form. Patent certificates of France No 7917216, No 2460367.
79. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of Canada No 1134510.
80. Device for reduction of p-Fibonacci codes to the minimal form. Patent certificate of Canada N1132263.
81. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of Poland No 108086.

82. Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of DDR No 150514.
83. Stakhov AP. Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic. The Computer Journal 2002, Vol. 45, No. 2: 222-236.
84. Stakhov AP. A generalization of the Fibonacci Q-matrix. Доклады Академии наук Украины, 1999, №9, с. 46-49.
85. Stakhov A. Fibonacci matrices, a generalization of the "Cassini formula", and a new coding theory. Chaos, Solitons & Fractals, 2006, Volume 30, Issue 1, 56-66
86. Бахвалов С.В., Иваницкая В.П. Основания геометрии. Москва: Издательство «Высшая школа», 1972.
87. Шанин Н.А. О рекурсивном математическом анализе и исчислении арифметических равенств Р.Л. Гудстейна. Вступительная статья к книге Р.Л. Гудстейна «Рекурсивный математический анализ», Москва: Наука, 1970.
88. Марков А.А. О логике конструктивной математики. Москва: Издательство «Знание», 1972.
89. Гильберт Д. О бесконечном. – В книге «Основания геометрии», 1948.
90. Зенкин А.А. Ошибка Георга Кантора // Вопросы философии. 2000, №2.
91. Стахов А.П., Клещев Д.С., Проблема бесконечного в математике и философии от Аристотеля до А.Зенкина // «Академия Тринитаризма», М., Эл № 77-6567, публ.15680, 03.12.2009 <http://www.trinitas.ru/rus/doc/0016/001c/00161588.htm>
92. Клещев Д. Лженаука: болезнь, которую некому лечить // «Академия Тринитаризма», М., Эл № 77-6567, публ.17012, 22.11.2011 <http://www.trinitas.ru/rus/doc/0232/013a/02322041.htm>
93. Стахов А.П. Не стоит ли современная математика на «лженаучном» фундаменте? (В порядке обсуждения статьи Дениса Клещева «Лженаука: болезнь, которую некому лечить») // «Академия Тринитаризма», М., Эл № 77-6567, публ.17034, 28.11.2011 <http://www.trinitas.ru/rus/doc/0232/013a/02322052.htm>
94. Вейль Г. О философии математики. М.-Л., 1934. (Репринт М.: КомКнига, 2005)
95. Уайлд Дж. Методы поиска экстремума. М.: Наука, 1967.
96. Виленкин Н.Я. Комбинаторика. М.: Наука, 1967.
97. Борисенко А.А. Биномиальный счет. Теория и практика. Сумы: Университетская книга, 2004.
98. Бородин А.И., Бугай А.С. Биографический словарь деятелей в области математики. Пер. с укр. Киев: Радянська школа, 1979.- 607 с.607.



99. Давыдов Е.С. Наименьшие группы чисел для образования натуральных рядов. Санкт-Петербург, 1903.
100. Гартц В.Ф. Лучшая система для весовых гирь. Санкт-Петербург, 1910.
101. Динамическая теория биологических популяций. Под ред. Р.А. Полуэктова. М.: Наука, Главная редакция физико-математической литературы. 1974. – 456 с.
102. Spears, C.P., Bicknell-Johnson, M. Asymmetric cell division: binomial identities for age analysis of mortal vs. immortal trees. *Applications of Fibonacci Numbers*, Vol. 7 (1998), 377-391.
103. Кольман Э. История математики в древности. М.: Физматгиз, 1961.
104. Башмакова И.Г., Юшкевич А.П. Происхождение систем счисления. - Энциклопедия Элементарной Математики, том 1 «Арифметика». М.: Гостехиздат, 1951. – с.7-74.
105. Пospelов Д.А. Арифметические основы вычислительных машин дискретного действия. М.: Высшая школа, 1970. – 308 с.
106. Нейгебауэр О. Лекции по истории античных математических наук. Том 1. Догреческая математика. Москва –Ленинград: ОНТИ НКТП СССР, 1937.
107. Николай Петрович Брусенцов - творец первого и единственного в мире троичного компьютера "Сетунь"  
[http://www.icfcst.kiev.ua/MUSEUM/Brusentsov\\_r.html](http://www.icfcst.kiev.ua/MUSEUM/Brusentsov_r.html)
108. Цифровые процессоры сигналов на основе троичных кодов  
[http://www.ci.ru/inform13\\_08/p\\_23.htm](http://www.ci.ru/inform13_08/p_23.htm)
109. Физики создали нанопамять с использованием троичной логики  
<http://www.moyaufa.ru/35590/1/view/news.html>
110. Троичная логика – прогресс в IT-технологиях  
<http://www.izobretenija.ru/vashi/667>
111. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М.: Советское радио, 1968. – 439 с.
112. Swinton Jonathan. Fibonacci phyllotaxis: Turing's problem, 2002  
[www.swintons.net/jonathan/Turing/fibonacci.htm](http://www.swintons.net/jonathan/Turing/fibonacci.htm)
113. Turing A. M. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London*, volume B 237, 1952, - p. 37-72.
114. Turing A. M.. The morphogen theory of phyllotaxis. In Saunders, 1992.
115. Харкевич А.А. Борьба с помехами. М.: Физматгиз, 1963.
116. Петухов С.В. Метафизические аспекты матричного анализа генетического кодирования и золотое сечение. *Метафизика*. Москва, Бином, 2006.-с.216-251.

117. Стахов А.П. Обобщенные золотые сечения и новый подход к геометрическому определению числа. Украинский математический журнал, том. 56, 2004 г.
118. Стахов А.П. О возможной причине участившихся аварий при выводе российских спутников // «Академия Тринитаризма», М., Эл № 77-6567, публ.17146, 26.12.2011 <http://www.trinitas.ru/rus/doc/0023/001a/00231046.htm>
119. Стахов А.П. О возможных причинах участившихся катастроф при запуске российских ракет // «Академия Тринитаризма», М., Эл № 77-6567, публ.17296, 09.02.2012 <http://www.trinitas.ru/rus/doc/0023/001a/00231047.htm>
120. Monteiro P. and R.W. Newcomb (University of Maryland). Minimal and maximal Fibonacci Representations: Boolean Generation. The Fibonacci Quarterly, 1976, v.14, No 1.
121. Licomendes P. and Newcomb R. (University of Maryland). Multilevel Fibonacci Conversion and Addition. The Fibonacci Quarterly, 1984, v.22, No 3.
122. Ligomenides P. and Newcomb R. Equivalence of some Binary, Ternary, and Quaternary Fibonacci Computers”. Proceeding of the Eleventh International Symposium on Multiple-Valued Logik, Norman, Oklahoma, May 1981.
123. Ligomenides P. and Newcomb R.. Complement Representations in the Fibonacci Computer. Proceedings of the Fifth Symposium ob Computer Arithmetic, Ann Arbor, Michigan, May 1981.
124. Newcomb R. Fibonacci Numbers as a Computer Base. Conference Proceedings of the Second Interamerican Conference on Systems and Informatics, Mexico City, November 27, 1974.
125. Hoang V.D. A Class of Arithmetic Burst-Error-Correcting Codes for the Fibonacci Computer. PhD Dissertation, University Maryland, December 1979.
126. Стахов А.П., Лихтциндер Б.Я., Орлович Ю.П., Сторожук Ю.А. Кодирование данных в информационно-регистрирующих системах. Киев, Техника, 1985 г.
127. Борисенко А.А., Стахов А.П. Об одном методе счета в коде Фибоначчи. Вестник Сумского государственного университета, 2011, №3, - с.141-149.

## Научная биография Алексея Стахова

1. **Алексей Стахов** является одним из лидеров мировой науки в области «золотого сечения» и «математики гармонии». С 2003 г. он является Президентом Международного Клуба Золотого Сечения, а с 2005 г. - Директором Института Золотого Сечения Академии Тринитаризма (Россия). Он был инициатором создания так называемой «Славянской Золотой Группы» (Киев, 1992) и научным руководителем Международного Конгресса по Математике Гармонии (Одесса, 2010). Имеет широкие международные связи с учеными США, России, Украины, Беларуси, Армении, Англии, Германии, Аргентины, Бразилии, Турции, Чили и других стран. Владеет русским, украинским и английским языками.
2. **Образование.** В 1961 г. закончил радиотехнический факультет Харьковского авиационного института (сейчас – Национальный аэрокосмический университет),
3. **Ученые степени и звания:** кандидат технических наук в области технической кибернетики (1966), доцент (1968), доктор технических наук в области вычислительной техники (1972), профессор (1974), академик Академии инженерных наук Украины (1992)
4. **Основные этапы научно-преподавательской деятельности:**
  - Заведующий кафедрой информационно-измерительной техники Таганрогского радиотехнического института, 1971 – 1977.
  - Заведующий кафедрой вычислительной техники Винницкого политехнического института (сейчас – Винницкий национальный технический университет), 1977 – 1988.
  - Директор Специального конструкторско-технологического бюро "Модуль" при Винницком политехническом университете, 1986 – 1989.
  - Заведующий кафедрой прикладной математики и вычислительных систем Винницкого технического университета, 1989 – 1995.
  - Заведующий кафедрой информатики Винницкого государственного аграрного университета, 1997-2003.
5. **Подготовка научных кадров:** подготовил 30 кандидатов наук, 4 ученика проф. Стахова защитили докторские диссертации.
6. **Краткая характеристика научной деятельности, основные теоретические достижения:**

6.1. Создал новое направление в теории измерения – алгоритмическую теорию измерения. В этой области имя проф. Стахова стоит рядом с именами признанных ученых в области теоретической метрологии.

6.2. Создал новое направление в развитии вычислительной техники, а именно новые системы счисления, основанные на числах Фибоначчи и золотой пропорции, и выдвинул проект «Компьютеры Фибоначчи». Мировой приоритет в этой области защищен 65 зарубежными патентами США, Японии, Англии, Франции, Германии, Канады и других стран. В 1989 г. выступил с докладом «Компьютеры Фибоначчи» на специальном заседании Президиума Академии наук Украины.

6.3. Разработал «Математику гармонии», как новое междисциплинарное направление, касающееся оснований математики, теоретической физики, компьютерной науки и математического образования. Впервые концепция «математики гармонии» была изложена профессором Стаховым в докладе "The Golden Section and Modern Harmony Mathematics", сделанном на 7-й Международной конференции «Числа Фибоначчи и их приложения» (Австрия, Грац, 1996). Его главным научным достижением в этой области является книга “The Mathematics of Harmony. From Euclid to Contemporary Mathematics and Computer Science” (World Scientific, 2009), вызвавшая большой интерес в современной науке.

6.4. В этой области опубликовал около 500 научных работ, среди них – 15 книг, 65 зарубежных патентов, 130 авторских свидетельств СССР. За период работы в Канаде (2004-2012) опубликовал около 30 статей в известных международных журналах (Chaos, Solitons and Fractals, Applied Mathematics, Congressus Numerantium, Arc Combinatoria, Visual Mathematics и др. ).

## **7. Наиболее важные научные доклады:**

- Доклад «Алгоритмическая теория измерения и основания компьютерной арифметики» на объединенном заседании компьютерного и кибернетического обществ Австрии (Вена, 1976);

- Доклад «Компьютеры Фибоначчи» на заседании Президиума Академии наук Украины (Киев, 1989);
- Доклад "The Golden Section and Modern Harmony Mathematics" на 7-й Международной конференции «Числа Фибоначчи и их приложения» (Австрия, Грац, 1996);
- Доклад «Новый тип элементарной математики и компьютерной науки, основанных на Золотом Сечении" на совместном заседании семинара "Геометрия и Физика" кафедры теоретической физики Московского университета и Междисциплинарного семинара "Симметрии в науке и искусстве" при Институте машиноведения РАН (Москва, МГУ, май 2003).

## **7. Научные награды, работа в зарубежных университетах:**

- 7.1. Работа «Приглашенным профессором»:
  - Венский технический университет (1976),
  - Йенский университет (1986),
  - Дрезденский технический университет (1988),
  - Университет Аль Фатех (Триполи, Ливия, 1995-1997),
  - Университет Эдуардо Мондлане (Мапуту, Мозамбик, 1998-2000),
- 7.2. Премии, награды:
  - Премия Министерства образования Украины в области науки за лучшую научную публикацию (1980)
  - Памятная медаль имени Генриха Баркгаузена, выданная Дрезденским Техническим Университетом как «Приглашенному профессору» кафедры имени Генриха Баркгаузена (1988)
  - Почетное звание «Рыцарь науки и искусств» (Российская Академия Естественных Наук, 2009)
  - Почетное звание «Доктор Священной Геометрии в Математике» (Американское Общество Золотого Сечения, 2010)

## **8. Научные форумы, проведенные под руководством Алексея Стахова:**

- Научный руководитель Международных семинаров «Золотая пропорция и проблемы гармонии систем» (Киев, 1992, 1993).

- Научный руководитель Международной Конференции „Проблемы Гармонии, Симметрии и Золотого Сечения в Природе, Науке и Искусстве" (Винница, октябрь 2003)
- Научный руководитель Международного Конгресса по Математике Гармонии (Одесса, октябрь 2010)
- Научный руководитель Международного online семинара по Математике Гармонии (Институт Золотого Сечения Академии Тринитаризма, ноябрь, декабрь 2011, январь 2012)