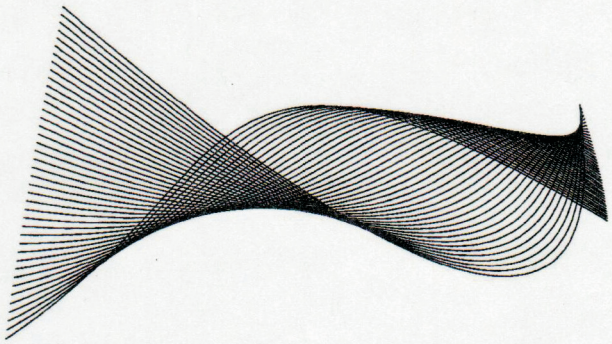
О.А. ФИНЬКО

Модулярная арифметика параллельных ЛОГИЧЕСКИХ ВЫЧИСЛЕНИЙ



РОССИЙСКАЯ АКАДЕМИЯ НАУК
МИНИСТЕРСТВО ОБОРОНЫ РОССИЙСКОЙ ФЕДЕРАЦИИ

О.А. Финько

МОДУЛЯРНАЯ АРИФМЕТИКА ПАРАЛЛЕЛЬНЫХ ЛОГИЧЕСКИХ ВЫЧИСЛЕНИЙ

Под редакцией *В.Д. МАЛЮГИНА*

МОСКВА · КРАСНОДАР · 2003

УДК 519.7
ББК 22.18
Ф 59

Р е ц е н з е н т ы:

лауреат Государственной премии СССР,
академик нац. АН республики Казахстан В. М. АМЕРБАЕВ
(Московский государственный ин-т электронной техники
(технический ун-т))

д-р техн. наук, проф. Е. К. ЛЕБЕДЕВ
(Чувашский государственный ун-т)

Финько, О. А.

Модулярная арифметика параллельных логических вычислений: Монография/ Финько О.А.; Под ред. В. Д. Малюгина; — М.: Ин-т проблем управл. им. В. А. Трапезникова РАН; Краснодар: Краснодар. воен. ин-т, 2003. — 224 с.: ил.

Монография посвящена новому направлению математической логики — реализации параллельных логических вычислений посредством арифметико-логических форм. Впервые рассматривается отображение классической логики на модулярную арифметику, которое открывает уникальные возможности по достижению высоких уровней производительности и отказоустойчивости средств гибких логических вычислений.

Для специалистов в области защиты информации, прикладной математики, математической кибернетики, информатики и вычислительной техники, адъюнктов (аспирантов), курсантов и слушателей соответствующих специальностей.

ББК 22.18

Рекомендовано к изданию Учеными советами института проблем управления им. В. А. Трапезникова Российской академии наук и Краснодарского военного института

Оглавление

Предисловие редактора

Нынешнюю информатизацию общества справедливо связывают с развитием вычислительной техники.

Как известно, все схемы, используемые при создании ЭВМ, основаны на математическом аппарате двоичной булевой алгебры или на его расширении — теории конечных автоматов.

Кажется, что и дальнейшее развитие вычислительной техники должно быть поддержано развитием именно аппарата булевой алгебры, столь успешно зарекомендовавшей себя на предыдущих этапах.

Тем не менее все годы триумфального шествия булевой алгебры предпринимались многочисленные попытки предложить такое математическое описание функциональных зависимостей, которое давало бы какой-то положительный вычислительный эффект при технической реализации дискретных устройств. Другая причина предпринимаемых усилий заключается в том, что в аппарате булевой алгебры, в частности, нет средств отражения *параллелизма вычислений*, широко используемого в современных ЭВМ.

Совершенствование исходных математических моделей осуществлялось путем введения таких понятий, как время (рекуррентные и временные функции), производные, секвенции, k-значный алфавит, арифметические и псевдобулевы функции, конечные матрицы и др.

Особо отметим предложения о применении *модулярной арифметики* и *спектральных* методов для описания логических функций.

Предлагаемая читателю работа Олега Анатольевича Финько нетрадиционна и оригинальна. Всякая книга характеризуется сочетанием общеизвестного и авторского. В данной книге,

несмотря на многочисленные ссылки, авторское исследование, безусловно, превалирует.

Ценность монографии состоит в подъеме на принципиально новый уровень применения методов модулярной арифметики для решения задач логического управления. Это открывает новые перспективы как в вопросах реализации параллельных логических вычислений, так и для самой модулярной арифметики. Читатель же вводится в новую и перспективную проблематику параллельных вычислений многомерных многоместных функций логического управления.

Доктор технических наук, профессор
В. Д. МАЛЮГИН,

Москва, ноябрь 2003 г.

Введение

Моей жене — Та́нюшке
и дочери — Таню́шке

1. Из множества алгоритмов, используемых в современных информационно-вычислительных и телекоммуникационных системах, можно выделить классы, реализация которых чаще, чем другие вызывает затруднения. К таким классам следует отнести алгоритмы, основу которых составляют *логические вычисления*. К „потребителям“ логических вычислений в первую очередь следует отнести системы автоматизированного проектирования (САПР) интегральных микросхем (ИМС) и системы криптографической защиты информации [15, 19, 45, 46, 47, 81, 94, 138].

Для первых острота проблемы вызвана постоянным усложнением дискретных устройств (увеличением количества вентилей, усложнением топологии ИМС, проблемой межсоединений и планаризации структур) и, следовательно, существенным возрастанием *размерности* решаемых задач [140, 193, 194].

Для вторых проблема реализации логических вычислений стала очевидной благодаря широкому внедрению криптографических методов в системы защиты электронной информации. В компьютерных системах требуется получить высокие скорости шифрования при использовании микропроцессоров общего применения. Для высокой защищенности обработки данных необходимо обеспечить *„глобальное“* [75] шифрование, то есть шифрование всех данных, хранимых в энергонезависимой памяти.

В общем случае, обработка данных связана с произвольными запросами на чтение и запись данных, поэтому требуется обеспечить параллельное независимое шифрование отдельных блоков данных. Современные ЭВМ характеризуются большим объемом

постоянной памяти и высокой скоростью записи и считывания. Эти факторы и определяют высокие требования, как к шифрам, так и к способам их реализации.

Перспективным способом повышения стойкости криптосистем является внесение элементов гибкости в некоторую часть струтуры криптоалгоритма. Для практической осуществимости этого способа необходимо, чтобы часть алгоритма шифрования была легко сменяемым элементом, то есть стала бы элементом секретного ключа. Такие шифры в [75] названы *недетерминированными*.

При использовании недетерминированных шифров перед сеансом шифрования выполняются предвычисления — генерация алгоритма шифрования. Наличие этапа предвычислений является фактором, существенно ужесточающим требования к *скорости* реализации логических вычислений.

В [74, 75] данная проблема решается путем разработки скоростных шифров, учитывающих специфику обработки информации в микропроцессорах общего применения или предполагающих применение предельно простых аппаратных средств. Однако, возможен и другой путь — *совершенствование методов обработки логической информации*. Второй путь более *универсален*, так как позволяет *расширить класс* эффективно реализуемых шифров на основе стандартных технических средств.

2. Реализация алгоритмов логического управления традиционно решалась в рамках теории конечных автоматов как задача синтеза схемы минимальной сложности. Несмотря на гибкость программной реализации, непосредственный перенос описания схемы в память ЭВМ как правило оказывается *нерациональным*, — всякое описание алгоритмов и схем на языке булевой алгебры не соответствует непрерывно возрастающим возможностям современных ЭВМ. При написании логической программы, как правило, используются лишь логические операторы и команды условного перехода. При этом, такие ресурсы ЭВМ как весь набор команд (а не только логические), аппаратно поддерживаемая разрядность целочисленных данных и др. используются крайне ограниченно.

Парадоксальность сложившейся ситуации заключается в том, что в основе построения всей вычислительной техники лежат все

те же законы алгебры логики, которые с помощью этой техники не удается эффективно реализовать.

Справедливо будет отметить тот факт, что схемная реализация логических алгоритмов, все еще продолжает активно использоваться во многих практических приложениях. Этому, в частности, способствует широкое распространение ПЛИС — программируемые пользователем ИМС. Аппаратная реализация используется в системах с криптографической защитой информации с целью поддержания высокой скорости обмена данными. Логические расширители могут быть использованы для наращивания вычислительных возможностей ЭВМ при ее проблемной ориентации на решение заданного класса задач. Ценой высокой скорости обработки при этом является предельное *снижение гибкости* вычислений, которая не всегда допустима (например, при реализации недетерминированных шифров).

Кроме того, из-за небольших объемов выпуска таких устройств, в стремлении снизить экономические и временные затраты, связанные с проектированием ИМС, разработчики (или заказчики) часто „жертвуют“ качеством решения достаточно важных для прикладной области вопросов верификации ИМС и обеспечением технического сервиса (контроля ошибок вычислений, тестирования, восстановления работоспособности) [93]. Применение ПЛИС, разумеется, существенно упрощает разработку специализированных вычислителей, однако также очевидно, что ПЛИС — это всего лишь средство, которое отражает и все *недостатки проектирования*.

В то же время промышленностью давно освоены капиталоемкие технологии производства широкой номенклатуры высококачественных специализированных вычислителей цифровой обработки (ЦО) сигналов, лишенных указанных недостатков. Однако многие из них не могут быть эффективно использованы для реализации логических вычислений. При использовании ПЛИС желательно применять высокоразвитый научный аппарат, наиболее полно учитывающий особенности целей проектирования.

3. Из практики логического программирования известны примеры, когда для реализации вычислительно трудных логических операций используют приемы замены этих операций другими — менее сложными операциями для заданных условий реали-

зации. Так, в 50-х годах XX-го столетия профессор Г. Айкен (Aiken, USA) использовал арифметическую форму задания логических функций в компьютерах MARK 3 и MARK 4 [144]. Для реализации подстановок большого размера в некоторых шифрах используют операцию возведения в дискретную степень и операцию дискретного логарифмирования в поле вычетов по заданному модулю (например, по модулю 257 в шифре SAFER).

В 70–80-х годах XX-го столетия в теории логического управления были заложены основы нового научного направления, основывающегося на *арифметических аспектах двоичной логики*[1] [18, 65, 66, 67, 68]. Сквозная идея развиваемого в этих работах подхода — *совместное описание систем логических функций произвольной размерности посредством арифметических полиномов.*

Последнее означает переход от двоичной логики к целочисленной арифметике (рис. 1). Как результат такой „арифметизации“ следует рассматривать обеспечение принципиально новых возможностей по применению высокоразвитых методов арифметических вычислений и парка серийной вычислительной техники, обеспеченного „отработанными“ методами технического сервиса, для решения задач, связанных с реализацией параллельных логических вычислений.

Теоретическую зрелость арифметические аспекты двоичной

[1] Идея описания логических функций методами арифметики была использована еще на этапе становления логического исчисления его основателем — Джоржем Булем (см. кн. Дж. Буль. „Исследование законов мышления“, 1854 г.) и другими основоположниками логики [52, 147]. Готтлоб Фруге в стремлении объединить логику и математику в книге „Основы арифметики“ (Die Grundlagen der Arithmetik, 1884 г.) напротив — арифметические понятия выводит из логики. Высшей точкой этих исследований „докомпьютерной“ эпохи в двадцатом столетии были „Основы математики“ (Principia Mathematica) Бертрана Рассела и Альфреда Уайтхеда (1910–1913 гг.), которые повлияли на позднейшие работы Гильберта по основам арифметики и по устранению парадоксов бесконечного [30, 31, 99]. При исследовании вопросов формализации, в частности, строились понятия элементарной математики (например:

$$a + b = 0 \sim a = 0 \wedge b = 0 \quad \text{или} \quad a \cdot b = 0 \sim a = 0 \vee b = 0;$$

формализм в теории сравнений:

$$a \equiv b \pmod{m} \sim \exists x (a = b + x \cdot m \vee b = a + x \cdot m),$$

где $\sim$ — отношение эквивалентности [31] С. 441).

Нетрадиционные подходы при решении логических уравнений, намного опередившие науку логики своего времени, использовал русский астроном и математик П. С. Порецкий. Он, в частности, ввел такие операции как логическое вычитание и логическое деление [89].

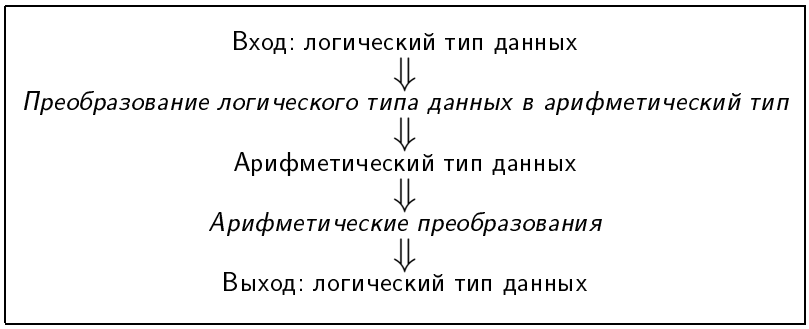


Рис. 1: Обобщенная схема логических вычислений посредством арифметических преобразований

логики приобрели в трудах сотрудника ин-та проблем управления им. В. А. Трапезникова Российской академии наук д-ра техн. наук, проф. В. Д. Малюгина [66, 67, 68, 71]. Полученные положения явились естественным продолжением прикладного направления математической логики, заложенного трудами В. Н. Шестакова, К. Шеннона, И. И. Жегалкина, М. А. Гаврилова. В настоящее время это направление продолжает активно развиваться в работах В. П. Шмерко, А. А. Шалыто, В. А. Артюхова, Л. А. Залманова, В. Н. Кондратьева, Г. А. Кухарева, В. М. Антоненко, С. Н. Янушкевич, Н. Х. Аслановой, Р. Г. Фараджева, В. С. Выхованец и др.

4. В условиях достижения значительных успехов в области организации параллельных вычислений, перевод логических вычислений в плоскость арифметического счета является фундаментальным по своей сути, но еще недостаточным условием для эффективной реализации *параллельных логических вычислений* в *современных* условиях.

Наиболее ярким явлением в области организации параллельных арифметических вычислений является *модулярная арифметика* [13, 14, 16, 21, 39, 35, 41, 50, 51, 59, 60, 61, 62, 95, 96, 102, 158, 181, 184][2] основанная на теории сравнений (см. Приложение А)

[2] Другое распространенное название — арифметика в системе остаточных классов — дословный, но математически неадаптированный перевод с англ. — residue number system arithmetic (правильный термин — „классы вычетов“ [25], однако *классы* вычетов не являются доминирующим понятием модулярной арифметики).

и, в частности, Китайской теореме об остатках (теорема А.28, с. 199)[3].

Уникальной особенностью средств ЦО, использующих в архитектуре и алгоритмах функционирования модулярную арифметику, является низкий, по сравнению с другими методами, *прирост аппаратурных затрат* при достижении предельного для используемой технологии изготовления микросхем уровня *производительности*. Другим не менее важным достоинством модулярной арифметики является развитость методов *контроля ошибок* вычислений и обеспечения *отказоустойчивости* средств вычислений.

Большой вклад в развитие теории модулярной машинной арифметики внесли отечественные ученые: И. Я. Акушский, Д. И. Юдицкий, В. М. Амербаев[4], В. А. Торгашев, М. В.

Идея использования модулярной арифметики для представления целочисленных данных в ЭВМ была выдвинута М. Валахом [189] в начале 50-х годов XX столетия и, затем, обобщена А. Свободой [95] для операций с рациональными числами. После этого в СССР и США начались активные исследования, посвященные этому перспективному направлению [13, 14, 102, 158, 159].

[3]Вопрос приоритетов этой теоремы (рассмотрения системы сравнений 1-й степени) до сих пор не выяснен. Первой известной работой считается работа китайского математика Сунь-Цзу, жившего примерно I–III веке нашей эры. В своей книге „Суань-Цзинь“ („Математический трактат“) Сунь-Цзы дает правило „тья-тен“ (большое обобщение) определения натурального числа, дающего остатки 2, 3, 2 при делении на 3, 5, 7 соответственно. Однако, вполне вероятно, что у китайских математиков эти сведения могли оказаться благодаря торговым путям, лежащим через Китай. Так, в [12] утверждается, что упоминание о подобной задаче было обнаружено в книге „Введение в арифметику“ греческого математика и пифагорейского философа Никомах из Герассы, жившего в I веке н. э. Независимо от китайских математиков способ решения задач такого рода был дан индусским математиком Брамегупта (588—660 г. н. э.).

Положения теории сравнений, наиболее близкие к современному пониманию принадлежат К. Гауссу (Carl Friedrich Gauss; 1777–1855).

[4]Первым в СССР в середине 50-х годов прошлого столетия на модулярную арифметику обратил внимание Ф.В. Лукин, тогда — главный инженер КБ-1 Минрадиопрома, получивший справку о подобных работах в США. В 1960 году Ф.В Лукин (в это время уже директор НИИ-37 — позже НИИ дальней радиосвязи) пригласил Д.И. Юдицкого и И.Я. Акушского возглавить разработку специализированных ЭВМ для радиолокационных станций систем противоракетной обороны (ПРО). Там основываясь на принципах модулярной арифметики, были построены супер-ЭВМ Т-340А и К-340А. Последняя обладала быстродействием 1,2 млн. двойных оп/с или 2,4 обычных оп/с. Эта ЭВМ первой в мире перешагнула барьер быстродействия в 1 млн. оп/с. При этом она обладала самой низкой стоимостью выполнения операций — 25 коп/оп и исключительно высокой надежностью. Эти ЭВМ и по настоящее время — более 40 лет (!) — продолжают работать в системах.

В июне 1972 года на боевое дежурство встала первая очередь системы ПРО Московского региона — система А-35 (генеральный конструктор — Г.В. Кисунько, ОКБ „Вымпел“ Минрадиопрома). Это была первая в стране крупная система, в которой цифровая ЭВМ, помимо традиционной функции вычислителя, взяла на себя задачу управления системой в реальном масштабе времени. Но к моменту создания А-35 у вероятного противника появились ракеты с разделяющимися боеголовками с индивидуальным наведением на цели.

Синьков, И.Т. Пак, А.А. Коляда, В.Г. Евстигнеев, Н.И. Червяков, Е.К. Лебедев[5], В.А. Краснобаев.

5. Следует отметить, что модулярная арифметика эффективна при реализации только определенных (модульных) арифметических операций, обуславливая проблемно-ориентированный характер этой арифметики. Логические операции традиционно относились к неэффективным операциям, так как для их выполнения требовался выход в вычислительную среду, основанную на „внешней“ позиционной арифметике.

Однако последние достижения по „арифметизации“ логических вычислений открывают уникальные возможности по реализации достоинств модулярной арифметики в области организации *параллельных логических вычислений.*

Таким образом, ***основной задачей дальнейших исследований будем считать получение теоретических положений, устанавливающих взаимосвязь между методами модулярной арифметики и методами реализации логических функций арифметическими полиномами.***

Решение этой задачи позволит *адаптировать* высокоразвитые математический аппарат и высокопроизводительные, отказоустойчивые технические средства ЦО, основанные на модулярной арифметике, например, средства ЦО сигнальной информации, для реализации *параллельных гибких логических вычислений.*

6. Исходя из определенной задачи исследований, структура монографии строится следующим образом. *В главе 1* кратко излагаются основные положения арифметической логики. Сформулирована теорема о единственности представления си-

---

Поэтому еще в 1965 году Г.В. Кисунько задумал вторую очередь А-35. Общими усилиями многих предприятий во главе с Ф.В. Лукиным и под научным руководством Д.И. Юдицкого и И.Я. Акушского был создан проект ЭВМ „Алмаз“, обладавшей производительностью — 8 млн. алгоритмических оп/с.

К 27 февраля 1971 года был создан 8-процессорный (4 модулярных и 4 двоичных процессора) комплекс 5Э53, обладавший фантастической для того времени вычислительной мощностью — 0,5 млрд. оп/с. Большой личный вклад в разработку модулярных алгоритмов 5Э53 внес В.М. Амербаев (сведения любезно предоставлены Б.М. Малашевичем [72], см. также [42, 69]).

[5]Изделия „Вычет-1“ и „Вычет-2“, разработанные под руководством Е.К. Лебедева, предназначенные для вычисления корреляционной функции, были удостоены серебряной медали ВДНХ СССР в 1989 г. Эти изделия нашли применение в радиолокации — системах распознавания радиоголографических образов.

стем булевых функций одним арифметическим полиномом. Далее рассмотрены алгебраический и матричный способы получения арифметического полинома. Дается определение линейного арифметического полинома. Рассмотрены основные пути развития арифметической логики.

В главах 2 — 4 вводятся модулярные (полиномиальные и матричные) арифметико-логические формы, основанные на одномодульной арифметике и Китайской теореме об остатках. *В главе 5* полученные модулярные формы обобщены на область k-значной логики.

В *главе 6* впервые обсуждаются вопросы построения альтернативных арифметико-логических форм, основанных на единственности разложения на сомножители и на Китайской теореме об остатках.

В *главе 7* решается сопутствующая проблема восстановления позиционной формы числа по его остаткам (варианта Китайской теоремы об остатках). Особенность приводимого алгоритма — минимальность величины результатов промежуточных вычислений.

Для реализации модулярных арифметико-логических форм требуется выполнение операции s-арного (группового) суммирования. *В главе 8* впервые для модулярной арифметики эта задача решается по принципу „вертикального“ (с помощью счета одноименных разрядов) суммирования операндов.

Наконец, *в главе 9* демонстрируются пути обеспечения контроля ошибок вычислений в модулярных арифметико-логических формах и создания высокоотказоустойчивых средств логических вычислений.

Сведения из теории чисел, необходимые для понимания последующих результатов — даны в *приложениях*.

Основные результаты, рассматриваемые в монографии прошли широкую апробацию на Международных конференциях: IEEE AIS'02, IEEE AIS'03 („Интеллектуальные системы“); CAD-2002, CAD-2003 („Интеллектуальные САПР“); SICPRO'03 („Идентификация систем и задачи управления“); DSPA-2003, DSPA-2004 („Цифровая обработка сигналов и ее применение“); МКПУ-2 („Вторая международная конференция по проблемам управления“, 2003 г.); на XIV Междунар. школе-семинаре „Син-

тез и сложность управляющих систем“, проводимой под руководством академика РАН О.Б. Лупанова [113, 114, 117] — [122, 124, 127].

Автор считает своим долгом выразить глубокую признательность *профессору В.Д. Малюгину* (ин-т проблем управления им. В.А. Трапезникова РАН) за бесконечную доброжелательность и высокое внимание к работе; рецензентам — *лауреату Государственной премии СССР, академику национальной АН республики Казахстан В.М. Амербаеву* (Московский государственный ин-т электронной техники (технический ун-т)) и *профессору Е.К. Лебедеву* (Чувашский государственный ун-т), чей труд и замечания способствовали значительному улучшению книги; научным консультантам — *профессору В.А. Цимбалу* (Серпуховский военный ин-т ракетных войск) и *профессору В.И. Ключко* (Кубанский государственный технологический ун-т) — за практическую помощь и ряд полезных советов в определении прикладной области результатов исследований; *профессору В.П. Шмерко* (ун-т Калгари, Канада) — за многочисленные замечания и предложения по улучшению работы, представленной в материалах статьи специального выпуска журнала Автоматика и телемеханика [128]; *Б.М. Малашевичу* (г. Зеленоград) — за любезно предоставленный очерк об истории создания вычислительной техники в Зеленограде; первому наставнику — *профессору Н.И. Червякову* (Ставропольский государственный ун-т) — за приглашение автора к исследованиям в столь замечательной области как модулярная арифметика; *адъюнкту А.Б. Сизоненко* (Краснодарский военный ин-т) — за ряд полезных замечаний и практическую помощь, оказанную при выполнении экспериментальной части исследований.

Автор признателен *командованию Краснодарского военного института, начальнику кафедры „Безопасности информации“ старшему научному сотруднику М.И. Бухонскому* и *профессорско-преподавательскому составу кафедры* — за предоставленную возможность выполнения исследований.

Особая благодарность *моей жене — Чайкиной Татьяне Александровне* — за б е с к о н е ч н о е терпение и заботу.

Обозначения

В скобках указаны страницы, на которых введены или впервые встречаются эти обозначения.

$a_1 \odot a_2 \odot \ldots \odot a_n$	—	кортеж элементов a_1, a_2, …, a_n, где $\odot$ — разделительный знак (стр. 19).
$(a_k\, a_{k-1}\, \ldots\, a_1)_b$	—	представление числа в позиционной системе счисления с основанием b: $\sum_k a_k b^k$ (стр. 19).
$\Xi^t\{Y\}$	—	оператор маскирования, определяет t-й разряд (выход) представления Y (стр. 25).
$\bigotimes$	—	символ кронекеровского умножения (стр. 27).
$\mathrm{abs}(a)$	—	абсолютная величина a (стр. 30).
$\lfloor a \rfloor$	—	наибольшее целое число, не превосходящее a (стр. 30).
$\square$	—	окончание доказательства (стр. 36).
$\lceil a \rceil$	—	наименьшее целое число $\geq a$ (стр. 50).
$\mathbf{CRT}_{k=1}^{v}\phi_k \pmod{m_k}$	—	применение к вычетам ϕ_k ($k = 1, 2, \ldots, v$) Китайской теоремы об остатках (стр. 60).
$\mathrm{ind}_a b$	—	индекс b по рассматриваемому модулю m и основанию a (стр. 147).
$b \mid a$	—	b делитель a (стр. 192).
$a \equiv b \pmod{m}$	—	a сравнимо с b по модулю m (стр. 192).
$\gcd(a, b)$	—	наибольший общий делитель чисел a, b (стр. 193).
$\mathrm{lcm}(a_1, a_2, \ldots, a_v)$	—	наименьшее общее кратное чисел a_1, a_2, …, a_v (стр. 196).
$\overline{a}$	—	класс по рассматриваемому модулю m (стр. 197).
$b = \lvert a \rvert_m$	—	b наименьший неотрицательный вычет a по модулю m a (стр. 197).
$\varphi(m)$	—	функция Эйлера (стр. 197).
$b \not\mid a$	—	b не делитель a (стр. 198).
$L(m)$	—	обобщенная функция Эйлера (стр. 198).

Глава 1

Основные положения арифметической логики

1.1 Представление булевых функций арифметическими полиномами

Пусть дана d-выходная булева функция $f(X)$ (система булевых функций — $f_1(X)$, $f_2(X)$, …, $f_d(X)$) от n переменных $X = x_1, x_2, \ldots, x_n$:

$$\begin{cases} y_1 = f_1(X), \\ y_2 = f_2(X), \\ \vdots \\ y_d = f_d(X), \end{cases} \tag{1.1}$$

где y_j — значение, принимаемое j-й булевой функцией $f_j(X)$; x_i, $y_j \in \{0, 1\}$ ($i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, d$). При этом кортеж значений булевых функций $y_d \odot y_{d-1} \odot \ldots \odot y_1$, где $\odot$ — разделительный знак, интерпретируется как код целого неотрицательного числа Y, представленного в двоичной системе счисления:

$$(y_d\, y_{d-1}\, \ldots\, y_1)_2 = Y = \sum_{j=1}^{d} y_j 2^{j-1}.$$

Пример 1.1
Представление Y, соответствующее системе булевых функций

$$\left\{ \begin{array}{l} f_1(X) = \overline{x_1 \oplus x_2}, \\ f_2(X) = \overline{x_1 \vee x_2}, \end{array} \right. \tag{1.2}$$

приведено в таблице 1.1 (здесь и далее $\vee$, $\wedge$, $\oplus$, $\neg$ — символы операций логического сложения, умножения, сложения по модулю 2 и инверсии соответственно).

Таблица 1.1: Пример числового представления значений системы булевых функций, соответствующих двоичному полусумматору

x_1	x_2	y_2	y_1	Y (десятичная запись)
0	0	1	1	3
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

Теорема 1.1
Произвольный кортеж булевых функций $y_d \odot y_{d-1} \odot \ldots \odot y_1$ может быть представлен арифметическим полиномом

$$Y = D(X) = \sum_{i=0}^{2^n-1} c_i \, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \tag{1.3}$$

где здесь и далее по тексту

$$\begin{array}{rcl} (i_1 i_2 \ldots i_n)_2 & = & \displaystyle\sum_{u=1}^{n} i_u 2^{n-u} \ (i_u \in 0, 1); \\ x_u^{i_u} & = & \left\{ \begin{array}{l} x_u, \ i_u = 1, \\ 1, \ i_u = 0; \end{array} \right. \end{array}$$

$c_i \in \mathcal{Z}$ $(i = 0, 1, \ldots, 2^n - 1)$ *и притом единственным образом [66, 67, 70, 71].*

Рис. 1.1: Схема вычисления произвольной системы булевых функций над кольцом $\mathcal{Z}$

Доказательство теоремы 1.1 дано в [71].

Принципы реализации логических вычислений посредством теоремы 1.1 поясняются с помощью рис. 1.1 и 1.2.

Схема вычисления, представленная на рис. 1.1, обеспечивает вычисление систем булевых функций, сведения о которых становятся известными в процессе решения задачи. Это безусловно обеспечивает высокую гибкость этого метода. Недостаток — увеличенные временные затраты, обусловленные необходимостью вычисления коэффициентов арифметического полинома. Этого недостатка лишена схема, поясняемая с помощью рис. 1.2. В этом случае коэффициенты арифметического полинома вычислены заранее и хранятся в памяти ЭВМ. Выбор схемы вычисления определяется особенностями конкретной решаемой задачи.

1.2 Преобразование булевых формул в арифметические полиномы

Алгебраический метод получения арифметического полинома (1.3) заключается в реализации следующего алгоритма.

Алгоритм 1.1

Шаг 1. *Получение арифметического полинома $P_j(X)$ для каж-*

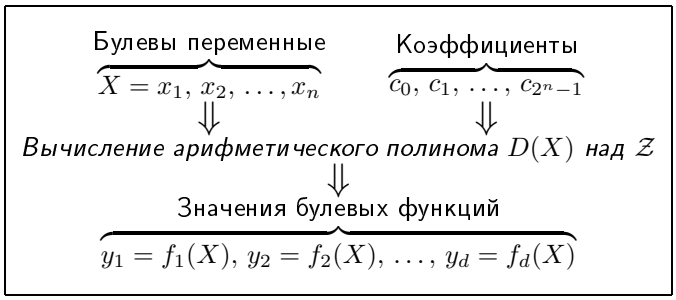


Рис. 1.2: Схема вычисления заданной системы булевых функций над кольцом $\mathcal{Z}$

дой булевой функции $y_j = f_j(X)$, $j = 1, 2, \ldots, d$:

$$\begin{cases} f_1(X) = P_1(X) = \sum\limits_{i=0}^{2^n-1} r_{1,i}\, x_1^{i_1} x_2^{i_2} \ \ldots\ x_n^{i_n}, \\ f_2(X) = P_2(X) = \sum\limits_{i=0}^{2^n-1} r_{2,i}\, x_1^{i_1} x_2^{i_2} \ \ldots\ x_n^{i_n}, \\ \vdots \\ f_d(X) = P_d(X) = \sum\limits_{i=0}^{2^n-1} r_{d,i}\, x_1^{i_1} x_2^{i_2} \ \ldots\ x_n^{i_n}. \end{cases} \tag{1.4}$$

Шаг 2. *Умножение арифметических полиномов на веса:*

$$\begin{cases} P_1'(X) = P_1(X)2^0 = \sum\limits_{i=0}^{2^n-1} r'_{1,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ P_2'(X) = P_2(X)2^1 = \sum\limits_{i=0}^{2^n-1} r'_{2,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ \vdots \\ P_d'(X) = P_d(X)2^{d-1} = \sum\limits_{i=0}^{2^n-1} r'_{d,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \end{cases} \tag{1.5}$$

где $r'_{j,i} = r_{j,i}2^{j-1} \quad (j = 1, 2, \ldots, d;\ i = 0, 1, \ldots, 2^n - 1)$.

Шаг 3. *Получение искомого арифметического полинома* $D(X)$

(1.3) путем суммирования коэффициентов арифметического полинома $P_j^{'}(X)$ *для всех* $j = 1, 2, \ldots, d$*:*

$$D(X) = \sum_{i=0}^{2^n-1} \sum_{j=1}^{d} r_{j,\,i}^{'}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = \sum_{i=0}^{2^n-1} c_i\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \quad (1.6)$$

где $c_i = \sum\limits_{j=1}^{d} r_{j,\,i}^{'} \quad (i = 0, 1, \ldots, 2^n - 1)$.

Структура алгоритма 1.1 поясняется с помощью рис. 1.3

Рис. 1.3: Схема алгебраического метода построения арифметического полинома $D(X)$

Пример 1.2
Для системы булевых функций (1.2) реализация алгоритма 1.1 имеет вид:

Шаг 1. *Используя соотношения*

$$\begin{aligned} x_1 \wedge x_2 &= x_1 x_2, \\ x_1 \vee x_2 &= x_1 + x_2 - x_1 x_2, \\ x_1 \oplus x_2 &= x_1 + x_2 - 2x_1 x_2, \\ \overline{x} &= 1 - x, \end{aligned}$$

получим

$$\begin{aligned} f_1(X) &= P_1(X) = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1x_2, \\ f_2(X) &= P_2(X) = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1x_2. \end{aligned}$$

Шаг 2.

$$\begin{aligned} P_1'(X) &= 2^0(1 - x_1 - x_2 + 2x_1x_2) = 1 - x_1 - x_2 + 2x_1x_2, \\ P_2'(X) &= 2^1(1 - x_1 - x_2 + x_1x_2) = 2 - 2x_1 - 2x_2 + 2x_1x_2. \end{aligned}$$

Шаг 3.

$$\begin{aligned} D(X) &= 1 + 2 - (1 + 2)x_1 - (1 + 2)x_2 + (2 + 2)x_1x_2 = \\ &= 3 - 3x_1 - 3x_2 + 4x_1x_2. \end{aligned}$$

Из этого примера можно видеть, что числовой диапазон, требуемый для представления коэффициентов и результатов промежуточных вычислений арифметических полиномов может значительно превосходить числовой диапазон, достаточный для представления Y. В рассмотренном случае для представления Y достаточно двух двоичных разрядов ($0 \le Y \le 2^2 - 1$). В то же время для представления коэффициентов $c_1 \dots c_4$ арифметического полинома (1.3) требуется четыре двоичных разряда ($-3 \le c_{1\dots4} \le 5$), а результаты промежуточных вычислений при $x_1 = x_2 = 1$ могут принимать значения от -6 до 7.

1.3 Линейные арифметические полиномы

Важное значение для представления d-выходных булевых функций $f(X)$ имеют линейные арифметические полиномы $L(X)$, которые определяются выражением

$$U = L(X) = d_0 + \sum_{i=1}^{n} d_i\, x_i = d_0 + d_1x_1 + \ldots + d_nx_n, \qquad (1.7)$$

где коэффициенты d_0, d_1, …, d_n — целые числа [71, 140, 193].

При вычислении $f_j(X)$ используется оператор маскирования $\Xi^t\{U\}$ [140, 193], служащий для определения t-го двоичного разряда (выхода) представления

$$U = (a_r \ \ldots \ a_t \ \ldots \ a_2 \, a_1)_2 \, ,$$

то есть $\Xi^t\{U\} = a_t$.

Пример 1.3
Для линеаризации $P_j(X) = x_1 + x_2 - x_1x_2$, соответствующего булевой функции $f_j(X) = x_1 \vee x_2$ используется введение дополнительной (избыточной) булевой функции $f_j^{(1)}(X)$. При этом образуется система функций:

$$\begin{aligned} f_j^{(1)}(X) &= 1 \oplus x_1 \oplus x_2, \\ f_j^{(2)}(X) &= x_1 \vee x_2. \end{aligned}$$

Тогда

$$U = L(X) = 2^1 f_j^{(2)}(X) + 2^0 f_j^{(1)}(X) = 1 + x_1 + x_2$$

и $f_j(X) = \Xi^2\{U\}$ [193].

Таким образом, для представления систем булевых функций (1.1) с помощью линейных арифметических полиномов $L(X)$ используется тот же принцип взвешивания представлений булевых функций с помощью весов 2^i ($i = 0, 1, \ldots$), что и при построении арифметических полиномов $D(X)$ (1.3). Однако значения i при этом выбираются с учетом введенных дополнительных булевых функций.

Пример 1.4
Дана система булевых функций [193]:

$$\left\{ \begin{array}{l} f_A(X) = x_1 \wedge x_3, \\ f_B(X) = \overline{x}_1 \wedge x_2, \\ f_C(X) = \overline{x}_2 \wedge x_3, \end{array} \right. \tag{1.8}$$

Для обеспечения линейности результирующего арифметического полинома добавляют вспомогательные булевы функции $f_A^{(1)}(X)$, $f_B^{(3)}(X)$, $f_C^{(5)}(X)$ *и получают систему булевых функций:*

$$\left\{\begin{array}{l} f_A^{(1)}(X) = x_1 \oplus x_3, \\ f_A^{(2)}(X) = f_A(X), \end{array}\right. \quad \left\{\begin{array}{l} f_B^{(3)}(X) = \overline{x}_1 \oplus x_2, \\ f_B^{(4)}(X) = f_B(X), \end{array}\right.$$

$$\left\{\begin{array}{l} f_C^{(5)}(X) = \overline{x}_2 \oplus x_3, \\ f_C^{(6)}(X) = f_C(X). \end{array}\right.$$

Далее, в соответствии с (1.7) и примером 1.3 имеем:

$$\begin{aligned} U_A &= L_A^{'}(X) = 2^1 f_A^{(1)}(X) + 2^0 f_A^{(2)}(X) = x_1 + x_3, \\ U_B &= L_B^{'}(X) = 2^1 f_B^{(3)}(X) + 2^0 f_B^{(4)}(X) = 1 - x_1 + x_2, \\ U_C &= L_C^{'}(X) = 2^1 f_C^{(5)}(X) + 2^0 f_C^{(6)}(X) = 1 - x_2 + x_3. \end{aligned}$$

Получаем линейный арифметический полином:

$$\begin{aligned} U &= L(X) = 2^0 L_A^{'}(X) + 2^2 L_B^{'}(X) + 2^4 L_C^{'}(X) = \\ &= 20 - 4x_1 - 12x_2 + 16x_3. \end{aligned} \tag{1.9}$$

Для определения t*-й булевой функции воспользуемся оператором маскирования* $\Xi^t\{U\}$*:*

$$\left\{\begin{array}{l} f_A(X) = \Xi^2\{U\}, \\ f_B(X) = \Xi^4\{U\}, \\ f_C(X) = \Xi^6\{U\}. \end{array}\right.$$

Отметим, что линейная форма арифметического полинома (1.9) достигнута за счет введения избыточных булевых функций и увеличения числового диапазона, необходимого для представления U в 2^3 раза.

1.4 Конъюнктивные преобразования

Под прямым и обратным матричным преобразованием (логическим дискретным преобразованием Фурье — ЛДПФ) понимают

соответственно пару преобразований [71, 140, 139]:

$$\mathbf{C} = \mathbf{A}_{2^n}\mathbf{Y}, \tag{1.10}$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{C}, \tag{1.11}$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования размерности $2^n \times 2^n$ (базис преобразования); $\mathbf{Y}$ — вектор истинности d-выходной булевой функции $f(X)$ такой, что

$$\mathbf{Y} = [\mathbf{Y}_d|\mathbf{Y}_{d-1}|\dots|\mathbf{Y}_1]^T = \left[Y^{(0)}Y^{(1)}\dots Y^{(2^n-1)}\right]^T,$$

где T — символ транспонирования; $Y^{(i)}$ — числовое значение, принимаемое d-выходной булевой функцией $f(X)$ на i-м наборе булевых аргументов обычной таблицы истинности (см. пример 1.1); $\mathbf{C} = [c_0\, c_1\, \dots\, c_{2^n-1}]^T$ — вектор коэффициентов арифметического полинома (1.3) или арифметический спектр булевой функции.

Матрица

$$\mathbf{A}_{2^n} = \left[\begin{array}{c|c} \mathbf{A}_{2^{n-1}} & 0 \\ \hline -\mathbf{A}_{2^{n-1}} & \mathbf{A}_{2^{n-1}} \end{array}\right]$$

является n-й кронекеровской степенью

$$\mathbf{A}_{2^n} = \bigotimes_{j=1}^{n} \mathbf{A}_1$$

базовой матрицы

$$\mathbf{A}_1 = \left[\begin{array}{cc} 1 & 0 \\ -1 & 1 \end{array}\right];$$

$$\mathbf{A}_{2^n}^{-1} = \bigotimes_{j=1}^{n} \mathbf{A}_1^{-1},$$

где $\mathbf{A}_1^{-1} = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array}\right]$ — базовая матрица обратного преобразования. Матрица $-\mathbf{A}_{2^{n-1}}$ образуется из $\mathbf{A}_{2^{n-1}}$ заменой знаков единичных элементов на противоположные.

Матричные преобразования хорошо алгоритмизируются и удобны для практического применения.

Пример 1.5
Пусть задана трехвыходная булева функция, векторы принимаемых значений, которой имеют вид:

$$\begin{array}{l}\mathbf{Y}_1 = [0\ 1\ 0\ 1\ 1\ 0\ 1\ 1]^T,\\ \mathbf{Y}_2 = [0\ 1\ 1\ 0\ 0\ 1\ 1\ 1]^T,\\ \mathbf{Y}_3 = [0\ 1\ 1\ 0\ 1\ 0\ 0\ 1]^T,\end{array}$$

Тогда

$$\mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}.$$

Выполняя прямое ЛДПФ (1.10), получим

$$\mathbf{C} = \mathbf{A}_{2^3}\mathbf{Y} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ 7 \\ 6 \\ -12 \\ 5 \\ -10 \\ -8 \\ 19 \end{bmatrix} \begin{array}{l} \\ x_3 \\ x_2 \\ x_2x_3 \\ x_1 \\ x_1x_3 \\ x_1x_2 \\ x_1x_2x_3 \end{array}.$$

Наиболее распространенной операцией криптографических алгоритмов блочного шифрования является операция *подстановки* σ, определяемая как биективное отображение конечного множества из k элементов Ω на себя:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & \dots & k \\ \sigma_1 & \sigma_2 & \sigma_3 & \dots & \sigma_k \end{pmatrix},$$

где $\sigma_i \in \{1, 2, \dots, k\}$ ($\sigma_i \neq \sigma_j$ при $i \neq j$); k — длина подстановки.

Пример 1.6
Рассмотрим одну из четырех операций подстановки, S_2-блока криптографического алгоритма американского стандарта DES:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \end{pmatrix}.$$

Прямое ЛДПФ (1.10) примет вид:

$$\mathbf{C} = \mathbf{A}_{2^4}\mathbf{Y} =$$

$$= \mathbf{A}_{2^4} \cdot \begin{bmatrix} 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \end{bmatrix}^T =$$

$$= \begin{bmatrix} 15 \\ -14 \\ -7 \\ 20 \\ -9 \\ 19 \\ 4 \\ -24 \\ -6 \\ 12 \\ 0 \\ -7 \\ 12 \\ -29 \\ 20 \\ 28 \end{bmatrix} \begin{matrix} \\ x_4 \\ x_3 \\ x_3x_4 \\ x_2 \\ x_2x_4 \\ x_2x_3 \\ x_2x_3x_4 \\ x_1 \\ x_1x_4 \\ x_1x_3 \\ x_1x_3x_4 \\ x_1x_2 \\ x_1x_2x_4 \\ x_1x_2x_3 \\ x_1x_2x_3x_4 \end{matrix}.$$

где матрица $\mathbf{A}_{2^4}$ имеет вид:

$$\left[\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{array}\right]$$

Из анализа структуры матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ следует, что максимальное количество единичных элементов находится в последней строке обеих матриц. Причем количество единичных элементов с одинаковыми знаками в нижней строке матрицы $\mathbf{A}_{2^n}$ равно 2^{n-1}. Учитывая, что максимальное значение, принимаемое

элементами матрицы $\mathbf{Y}$ равно $2^d - 1$ (d — количество реализуемых одновыходных булевых функций), можно сделать вывод о том, что в результирующей матрице $\mathbf{C}$ максимальное абсолютное значение может иметь коэффициент $\mathrm{abs}(c_{2^n-1}) = 2^{n-1}(2^d-1)$. Для его представления в двоичной системе счисления с учетом необходимости представления знака числа потребуется

$$N_{\mathbf{C}} = \lfloor \log(2^{n-1}(2^d - 1)) + 2 \rfloor = n + d \qquad (1.12)$$

двоичных разрядов ($\lfloor x \rfloor$ — наибольшее целое число, не превосходящее x).

Для линейных арифметических полиномов проблема больших коэффициентов является еще более критичной. Однако в этом случае причиной большой величины коэффициентов является, прежде всего, большое количество реализуемых булевых функций, что в свою очередь вызвано необходимостью введения избыточных булевых функций, имеющих вспомогательный (служебный) характер.

1.5 Арифметический аналог логического ряда Тейлора

Разложение Тейлора — аналог ряда Тейлора в математическом анализе. Его достоинство — удобное для анализа представление коэффициентов арифметического полинома (арифметические производные).

Определение 1.1
Булевой производной функции $f(X)$ от n переменных по i-й переменной x_i ($i = 1, 2, \ldots, n$) называется логическая разность [145]:

$$\frac{df(X)}{dx_i} = f(x_1, \ldots, x_i, \ldots, x_n) \oplus f(x_1, \ldots, \overline{x}_i, \ldots, x_n).$$

Определение 1.2
Арифметический аналог булевой производной булевой функции

$f(X)$ от n переменных по i-й переменной x_i ($i = 1, 2, \ldots, n$) определяется в виде [143, 192]:

$$\frac{\widetilde{d}f(X)}{\widetilde{d}x_i} = -f(x_1, \ldots, \overline{x}_i, \ldots, x_n) + f(x_1, \ldots, x_i, \ldots, x_n). \quad (1.13)$$

Определение 1.3 [1]
Арифметический аналог логического ряда Тейлора [145] булевой функции $f(X)$ в точке

$$x_1 = w_1, x_2 = w_2, \ldots, x_n = w_n$$

определяется выражением

$$P_W(X) = \frac{1}{2^n} \sum_{j=0}^{2^n-1} p_W^{(j)} (x_1 \oplus w_1)^{j_1} (x_2 \oplus w_2)^{j_2} \ \ldots \ (x_n \oplus w_n)^{j_n}, \quad (1.14)$$

где двоичное представление параметра W — полярность (определяет правило записи булевых переменных: с отрицанием или без отрицания);

$$\begin{aligned} W &= (w_1\, w_2 \ \ldots \ w_n)_2, \\ j &= (j_1\, j_2 \ \ldots \ j_n)_2, \\ p_W^{(j)} &= \frac{\widetilde{d}^n f_{(W)}(X)}{\widetilde{d}x_1^{j_1} \widetilde{d}x_2^{j_2} \ldots \widetilde{d}x_n^{j_n}}. \end{aligned}$$

На основе (1.14) может быть найдено 2^n арифметических выражений, соответствующих 2^n полярностям.

Пример 1.7 [2]
Арифметический полином полярности $W = 3 = (0\,1\,1)_2$ произвольной булевой функции трех переменных представляется рядом (1.14) следующим образом:

$$P_3(X) = \tfrac{1}{8} \textstyle\sum_{j=0}^{7} p_3^{(j)} (x_1 \oplus 0)^{j_2} (x_2 \oplus 0)^{j_2} (x_3 \oplus 0)^{j_3} =$$

[1] В. П. Шмерко
[2] В. П. Шмерко

$$= \frac{1}{8}\left(f_3(X) + \frac{\widetilde{d}f_3(X)}{\widetilde{d}x_3}\overline{x}_3 + \frac{\widetilde{d}f_3(X)}{\widetilde{d}x_2}\overline{x}_2 + \frac{\widetilde{d}^2 f_3(X)}{\widetilde{d}x_2\widetilde{d}x_3}\overline{x}_2\overline{x}_3 + \right.$$
$$\left. + \frac{\widetilde{d}f_3(X)}{\widetilde{d}x_1}\overline{x}_1 + \frac{\widetilde{d}^2 f_3(X)}{\widetilde{d}x_1\widetilde{d}x_3}x_1\overline{x}_3 + \frac{\widetilde{d}^2 f_3(X)}{\widetilde{d}x_1\widetilde{d}x_2}x_1\overline{x}_2 + \frac{\widetilde{d}^3 f_3(X)}{\widetilde{d}x_1\widetilde{d}x_2\widetilde{d}x_3}x_1\overline{x}_2\overline{x}_3\right).$$

Расчетные значения коэффициентов арифметических полиномов (арифметические производные в точках) для булевой функции $f(X) = x_2\overline{x}_2 \vee \overline{x}_3$ *представлены в таблице 1.2.*

Таблица 1.2: Расчетные значения коэффициентов арифметических полиномов (арифметические производные в точках)

W	$P_W(X)$	000 X	001 $\frac{\widetilde{d}f(X)}{\widetilde{d}x_3}$	010 $\frac{\widetilde{d}f(X)}{\widetilde{d}x_2}$	011 $\frac{\widetilde{d}^2 f(X)}{\widetilde{d}x_2\widetilde{d}x_3}$	100 $\frac{\widetilde{d}f(X)}{\widetilde{d}x_1}$	101 $\frac{\widetilde{d}^2 f(X)}{\widetilde{d}x_1\widetilde{d}x_3}$	110 $\frac{\widetilde{d}^2 f(X)}{\widetilde{d}x_1\widetilde{d}x_2}$	111 $\frac{\widetilde{d}^3 f(X)}{\widetilde{d}x_1\widetilde{d}x_2\widetilde{d}x_3}$
0	$P_0(X)$	1	1	0	0	0	1	0	1
1	$P_1(X)$	0	−1	0	0	0	−1	−1	−1
2	$P_2(X)$	1	1	0	0	0	0	0	−1
3	$P_3(X)$	0	−1	0	0	0	0	1	1
4	$P_4(X)$	1	0	0	−1	0	−1	0	−1
5	$P_5(X)$	1	0	1	1	1	1	1	1
6	$P_6(X)$	1	1	0	1	0	0	0	1
7	$P_7(X)$	0	−1	−1	−1	0	0	−1	−1

Алгебраическая запись будет иметь вид:

$$\begin{aligned}
P_0(X) &= 1 + x_3 + x_1x_3 + x_1x_2x_3,\\
P_1(X) &= -\overline{x}_3 - x_1 - x_1\overline{x}_3 - x_1x_2 - x_1x_2\overline{x}_3,\\
P_2(X) &= 1 + x_3 - x_1\overline{x}_2x_3,\\
P_3(X) &= -\overline{x}_3 + x_1\overline{x}_2 + x_1\overline{x}_2\overline{x}_3,\\
P_4(X) &= 1 - x_2x_3 - \overline{x}_1x_3 - \overline{x}_1x_2x_3,\\
P_5(X) &= 1 + x_2 + x_2\overline{x}_3 + \overline{x}_1 + \overline{x}_1\overline{x}_3 + \overline{x}_1x_2 + \overline{x}_1x_2\overline{x}_3,\\
P_6(X) &= 1 + x_3 + \overline{x}_2x_3 + \overline{x}_1\overline{x}_2x_3,
\end{aligned}$$

$$P_7(X) = -\overline{x}_3 - \overline{x}_2 - \overline{x}_2\overline{x}_3 - \overline{x}_1\overline{x}_2 - \overline{x}_1\overline{x}_2\overline{x}_3.$$

Разложение Тейлора удобно для анализа свойств исходной функции в терминах *изменения* [143]. Ограничения связаны с невозможностью представления *системы* булевых функций.

1.6 Арифметические полиномы и полиномы Жегалкина

Перепишем матричные преобразования (1.10) и (1.11) для случая арифметических полиномов (6.4), задающих одну булеву функцию:

$$\mathbf{R} = \mathbf{A}_{2^n}\mathbf{Y}, \tag{1.15}$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{R}, \tag{1.16}$$

где $\mathbf{R} = [r_0\, r_1\, \ldots\, r_{2^n-1}]^T$ — вектор коэффициентов арифметического полинома (6.4); $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и обратного преобразования из отношений (1.10) и (1.11).

Для полиномома Жегалкина [1, 43, 44]

$$G(X) = \bigoplus_{i=0}^{2^n-1} g_i \wedge \left(x_1^{i_1} \wedge x_2^{i_2} \wedge \ldots \wedge x_n^{i_n}\right); \quad G(X),\ g_i \in \{0,1\}$$

могут быть заданы матричные преобразования:

$$\mathbf{G} = \mathbf{A}_{2^n} * \mathbf{Y}, \tag{1.17}$$

$$\mathbf{Y} = \mathbf{A}_{2^n} * \mathbf{G}, \tag{1.18}$$

где $\mathbf{G} = [g_0\, g_1\, \ldots\, g_{2^n-1}]^T$ — вектор коэффициентов полинома Жегалкина; $*$ — символ операции умножения матриц, в которой арифметическое сложение заменяется логическим сложением по модулю два [1].

Подставляя вектор $\mathbf{Y}$, выраженный в виде (1.16), в (1.18), можно получить [18, 70, 135]

$$\mathbf{G} = \mathbf{A}_{2^n} * \mathbf{A}_{2^n}\mathbf{R}.$$

Выражая вектор $\mathbf{Y}$ в виде (1.16), получим

$$\mathbf{R} = \mathbf{A}_{2^n}\mathbf{A}_{2^n} * \mathbf{G}.$$

Теорема 1.2 [3]
Пусть дана система булевых функций — $f_1(X)$, $f_2(X)$, ..., $f_d(X)$ от n переменных $X = x_1, x_2, \ldots, x_n$ и соответствующее им представление полиномами Жегалкина:

$$\begin{cases} G_1(X) = \bigoplus\limits_{i=0}^{2^n-1} g_{1,i} \wedge \left(x_1^{i_1} \wedge x_2^{i_2} \wedge \ldots \wedge x_n^{i_n}\right), \\ G_2(X) = \bigoplus\limits_{i=0}^{2^n-1} g_{2,i} \wedge \left(x_1^{i_1} \wedge x_2^{i_2} \wedge \ldots \wedge x_n^{i_n}\right), \\ \vdots \\ G_d(X) = \bigoplus\limits_{i=0}^{2^n-1} g_{d,i} \wedge \left(x_1^{i_1} \wedge x_2^{i_2} \wedge \ldots \wedge x_n^{i_n}\right). \end{cases} \tag{1.19}$$

Тогда система булевых функций может единственным образом представлена одним логическим полиномом:

$$\begin{aligned} G^{(o)}(X) &= \bigoplus_{i=0}^{2^n-1} \left(g_{d,\,i} \ldots g_{1,\,i}\, g_{0,\,i}\right)_2 x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = \\ &= \bigoplus_{i=0}^{2^n-1} b_i\; x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \end{aligned} \tag{1.20}$$

где $b_i = \left(g_{d,\,i} \ldots g_{1,\,i}\, g_{0,\,i}\right)_2 \in \{0,\, 1, \ldots, 2^d - 1\}$; $g_{j,\,i} \in \{0,\, 1\}$ — коэффициенты полиномов Жегалкина (1.19).

Доказательство
По аналогии с шагом 1 и шагом 2 алгоритма 1.1 используем принцип взвешивания применительно к полиномам (1.19):

[3] На возможность обобщения полинома Жегалкина для случая реализации системы булевых функций указано в [71] с. 169.

- умножение полиномов на веса 2^{j-1} $(j = 1, 2, \ldots, d)$:

$$\begin{cases} G_1(X)2^0 = \bigoplus\limits_{i=0}^{2^n-1} b'_{1,i}\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}, \\ G_2(X)2^1 = \bigoplus\limits_{i=0}^{2^n-1} b'_{2,i}\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}, \\ \vdots \\ G_d(X)2^{d-1} = \bigoplus\limits_{i=0}^{2^n-1} b'_{d,i}\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}, \end{cases} \tag{1.21}$$

где $b'_{j,i} = 2^{j-1}g_{j,i} = (a_{j,i}^{(d-1)}\ a_{j,i}^{(d-2)}\ \ldots\ a_{j,i}^{(1)}\ a_{j,i}^{(0)})_2$; $a_{j,i}^{(k)} \in \{0, 1\}$; $i = 1, 2, \ldots, 2^n - 1$; $k, j = 1, 2, \ldots, d$.

- получение искомого полинома (1.20) путем суммирования коэффициентов полиномов (1.21) для всех $j = 1, 2, \ldots, d$:

$$\begin{aligned} G^{(o)}(X) &= \bigoplus_{i=0}^{2^n-1} \bigvee_{j=1}^{d} b'_{j,i}\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n} = \\ &= \bigoplus_{i=0}^{2^n-1} \bigvee_{j=1}^{d} (a_{j,i}^{(d-1)}\ \ldots\ a_{j,i}^{(1)}\ a_{j,i}^{(0)})_2\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n} = \\ &= \bigoplus_{i=0}^{2^n-1} b_i\ x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}, \end{aligned}$$

где $b_i = \sum_{j=1}^{d} g_{j,i}2^{j-1} = (a_{d-1,\,i}\ \ a_{d-2,\,i}\ \ \ldots\ \ a_{1,\,i}\ \ a_{0,\,i})_2$; a_j, $i \in \{0, 1\}$; $j = 1, 2, \ldots, d$; $i = 1, 2, \ldots, 2^n - 1$.

Принимая во внимание, что в записи $b'_{j,i} = (a_{j,i}^{(d-1)} \ldots a_{j,i}^{(1)}\ a_{j,i}^{(0)})_2$ двоичные цифры $a_{j,i}^{(k)}$ имеют значения

$$a_{j,i}^{(k)} = \begin{cases} 0 & \text{при} \quad k \neq j; \\ g_{j,i} & \text{при} \quad k = j, \end{cases} \tag{1.22}$$

получим

$$\begin{array}{rllll} b'_{0,\,i} = & (0 & 0 & \ldots\ 0 & a^{(0)}_{0,\,i})_2 \\ b'_{1,\,i} = & (0 & 0 & \ldots\ a^{(1)}_{1,\,i} & 0)_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b'_{d-1,\,i} = & (0 & a^{(d-2)}_{d-2,\,i} & \ldots\ 0 & 0)_2 \\ b'_{d,\,i} = & (a^{(d-1)}_{d-1,\,i} & 0 & \ldots\ 0 & 0)_2 \\ \hline b_i = & (a_{d-1,\,i} & a_{d-2,\,i} & \ldots\ a_{1,\,i} & a_{0,\,i})_2, \end{array} \tag{1.23}$$

где

$$a_{j,\,i} = \bigvee_{k=0}^{d-1} a^{(k)}_{j,i}$$

($j = 1,\, 2,\, \ldots,\, d;\ i = 1,\, 2,\, \ldots,\, 2^n - 1$).

Окончательно, учитывая (1.22), получим

$$b_i = (g_{d-1,\,i}\ g_{d-2,\,i}\ \ldots\ g_{1,\,i}\ g_{0,\,i})_2\,,$$

где $i = 1,\, 2,\, \ldots,\, 2^n - 1$.

□

Определение 1.4
Полином (1.20) называется обобщенным полиномом Жегалкина [71].

1.7 Выводы

- Существующий парк вычислительной техники традиционно не ориентирован на реализацию массовых логических вычислений.
- Технический прогресс привел к существенным количественным и качественным изменениям в составе и содержании решаемых задач: интеллектуализация управления, распараллеливание процессов управления обработкой информации, возрастание объемов, ценности обрабатываемой и передаваемой информации и, как следствие, массовое применение

криптографических методов защиты информации, развитие автоматизированных систем проектирования и лавинообразное возрастание сложности интегральных микросхем.

- Ключ к решению сложившегося противоречия — распараллеливание логических вычислений на основе арифметических методов представления и обработки логических данных.
- Логические функции и системы логических функций могут быть представлены в виде одного арифметического полинома, причем единственным образом.
- Существуют алгебраический и матричный методы построения арифметических полиномов. Матричный метод (разновидность логического дискретного преобразования Фурье) наиболее приспособлен для алгоритмизации методов арифметической логики.
- Существенное препятствие на пути использования методов арифметической логики — высокая сложность арифметических полиномов, большая величина коэффициентов и результатов промежуточных вычислений.
- Линеаризация арифметических полиномов — один из наиболее перспективных путей уменьшения сложности вычислений. Однако для линейных арифметических полиномов еще в более острой форме свойственна проблема больших весовых коэффициентов и результатов промежуточных преобразований.
- Перспективный путь развития арифметической логики и решения проблемы больших весовых коэффициентов и результатов промежуточных вычислений — применение методов модулярной арифметики.
- Разработка методов реализации логических вычислений, основанных на модулярной арифметике, позволит задействовать в указанных целях высокоразвитый математический аппарат и технические средства ЦО сигналов.

Глава 2

Введение в модулярные формы арифметической логики

В данной главе[1] рассмотрим новые принципы задания булевых функций, основанные на отображении алгебры логики на арифметику конечного кольца Z_m.

2.1 Модулярные полиномиальные формы

Одномодульной арифметикой будем называть арифметику кольца вычетов Z_m, где m — значение модуля. Согласно [12, 16, 50] модулярные преобразования обладают рядом полезных свойств, позволяющих ограничить величину результатов промежуточных вычислений заданного выражения. При этом схема вычислений имеет вид (рис. 2.1). Воспользуемся этой схемой для реализации логических вычислений посредством арифметических полиномов.

Теорема 2.1
Если $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то произвольный кортеж булевых функций может

[1] Основное содержание данной главы впервые апробировано и опубликовано (принято к опубликованию) в [118, 120, 121, 122, 127, 128, 130].

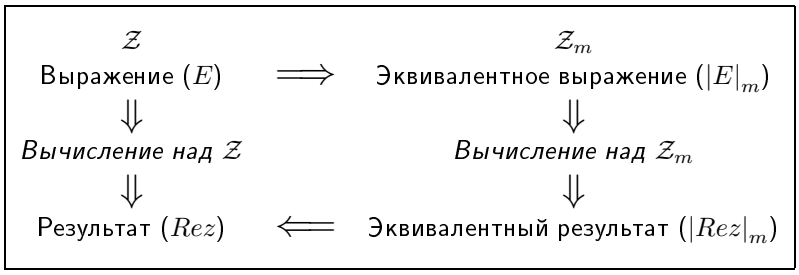


Рис. 2.1: Принцип ограничения величины промежуточных результатов вычисления выражения(E) над $\mathcal{Z}$ путем вычисления эквивалентного выражения ($|E|_m$) над $\mathcal{Z}_m$ [12]

быть представлен арифметическим полиномом:

$$Y = \mu(X) = \left| \sum_{i=0}^{2^n-1} \psi_i \, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m , \tag{2.1}$$

где $\psi_i = |c_i|_m$ $(i = 0, 1, \dots, 2^n - 1)$.

Доказательство

Рассмотрим два варианта доказательств, которые определят два варианта алгоритма получения (2.1).

(i) Пусть дан арифметический полином $D(X)$ (1.3). Тогда согласно теории сравнений [25] справедливо:

$$|Y|_m = \left| \sum_{i=0}^{2^n-1} c_i \, x_1^{i_1} \dots x_n^{i_n} \right|_m = \left| \sum_{i=0}^{2^n-1} |c_i|_m \, x_1^{i_1} \dots x_n^{i_n} \right|_m . \tag{2.2}$$

Но при условии $Y < m$ (Y — неотрицательное) выполняется $|Y|_m = Y$. Следовательно, при $Y < m$:

$$Y = \left| \sum_{i=0}^{2^n-1} |c_i|_m \, x_1^{i_1} \dots x_n^{i_n} \right|_m = \left| \sum_{i=0}^{2^n-1} \psi_i \, x_1^{i_1} \dots x_n^{i_n} \right|_m , \tag{2.3}$$

где $\psi_i = |c_i|_m$.

□

Отсюда вытекает алгоритм получения (2.1), заключающийся в выполнении следующих шагов (рис. 2.2):

Шаг. 1. Построение (1.3) посредством алгоритма 1.1.

Шаг. 2. Определение (2.1) посредством выражения (2.3).

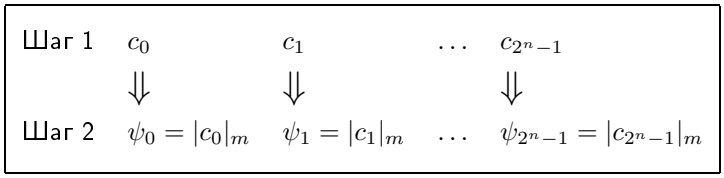


Рис. 2.2: Пояснения к первому варианту алгоритма получения (2.1)

(ii) Так как $f_j(X) \in \{0, 1\}$ и $m > Y$ с учетом предыдущих рассуждений, полином (6.4) можно представить в виде:

$$f_j(X) = \mu'_j(X) = \left| \sum_{i=0}^{2^n-1} \psi'_{j,\,i}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m , \qquad (2.4)$$

где $\psi'_{j,\,i} = |r_{j,\,i}|_m$ ($j = 1, 2, \dots, d$; $i = 0, 1, \dots, 2^n - 1$). Далее, приведением (1.5) и (1.6) по модулю m получим:

$$\mu''_j(X) = \sum_{i=0}^{2^n-1} \psi''_{j,\,i}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \qquad (2.5)$$

где $\psi''_{j,\,i} = \left|\psi'_{j,\,i} b_j\right|_m$, $b_j = |2^{j-1}|_m$, при $j = 1, 2, \dots, d$; $i = 0, 1, \dots, 2^n - 1$;

$$Y = \mu(X) = \left| \sum_{i=0}^{2^n-1} \sum_{j=1}^{d} \psi''_{j,\,i}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m . \qquad (2.6)$$

Преобразование (2.6) дает (2.1).

□

Рис. 2.3: Пояснения ко второму варианту алгоритма получения (2.1)

Таким образом, второй вариант алгоритма получения (2.1) состоит в выполнении трех шагов, заключающихся соответственно в построении (2.4), (2.5) и (2.6) (рис. 2.3).

Замечание 2.1

$$m \geq 2^d.$$

Определение 2.1
Выражение (2.1) будем называть представлением булевой функции $f(X)$ на основе модулярной формы арифметического полинома.

Сравнительный анализ арифметических полиномов $D(X)$ и $\mu(X)$ можно выполнить на примере некоторых элементарных булевых функций (табл. 2.1).

Используя схему вычислений, представленную на рис. 2.1, схемы вычислений — рис. 1.1 и 1.2 — можно представить в виде рис. 2.4 и 2.5.

Следствие 2.1
Коэффициенты ψ_i ($i = 0, 1, \dots, 2^n - 1$) арифметического полинома $\mu(X)$ (2.1) лежат в области целых неотрицательных чисел, а их числовой диапазон равен значению модуля m.

Таблица 2.1:

$f(X)$	$D(X)$	$\mu(X)$
$\overline{x}_i$	$1 - x_i$	$\lvert 1 + (m-1)x_i \rvert_m$
$x_1 \wedge x_2$	$x_1 x_2$	$x_1 x_2$
$x_1 \vee x_2$	$x_1 + x_2 - x_1 x_2$	$\lvert x_1 + x_2 + (m-1)x_1 x_2 \rvert_m$
$x_1 \oplus x_2$	$x_1 + x_2 - 2x_1 x_2$	$\lvert x_1 + x_2 + (m-2)x_1 x_2 \rvert_m$
$\overline{x_1 \wedge x_2}$	$1 - x_1 x_2$	$\lvert 1 + (m-1)x_1 x_2 \rvert_m$
$\overline{x_1 \vee x_2}$	$1 - x_1 - x_2 + x_1 x_2$	$\lvert 1 + (m-1)x_1 + (m-1)x_2 + x_1 x_2 \rvert_m$

Рис. 2.4: Схема вычисления произвольной системы булевых функций над $\mathcal{Z}_m$

Следствие 2.2
Если для одной и той же системы булевых функций заданы два арифметических полинома $D(X)$ (1.3) и $\mu(X)$ (2.1), а K_1 и K_2 — количество членов этих полиномов, то $K_2 \leq K_1$.

Для пояснения следствия 2.2 рассмотрим следующий пример.

Пример 2.1
Рассмотрим систему булевых функций (1.2), которой согласно выражению (1.3) (пример 1.2) соответствует арифметический полином

$$Y = D(X) = 3 - 3x_1 - 3x_2 + 4x_1 x_2.$$

Рис. 2.5: Схема вычисления заданной системы булевых функций над $\mathcal{Z}_m$

Применение теоремы 2.1 в общем случае дает:

$$Y = \mu(X) = |3 + (m-3)x_1 + (m-3)x_2 + 4x_1x_2|_m .$$

При $m = 4$ *получим*

$$\mu(X) = |3 + x_1 + x_2|_4 .$$

Таким образом, следствие 2.2 указывает на то, что модулярная форма арифметического полинома (2.1) как минимум не усложняет полиномиальной формы представления систем булевых функций по показателям K_1 и K_2, а как максимум — позволяет уменьшить сложность арифметических полиномов за счет сокращения коэффициентов, кратных m. Следовательно, значение модуля m может выбираться не только по критерию собственной минимальности, но и по критерию минимальности K_2.

Лемма 2.1
Если кортеж булевых функций (1.1) задан линейным арифметическим полиномом (1.7), то при $m > U_{\max}$ *справедлива модулярная форма линейного арифметического полинома:*

$$\begin{aligned} U = \lambda(X) &= \left| \omega_0 + \sum_{i=1}^{n} \omega_i x_i \right|_m = \\ &= |\omega_0 + \omega_1 x_1 + \ldots + \omega_n x_n|_m , \end{aligned} \qquad (2.7)$$

где $\omega_j = |d_j|_m$ $(j = 0, 1, \ldots, n)$.

Доказательство
Рассмотрим арифметический полином

$$\begin{aligned} U &= \left| \omega_0^\star \sum_{i=1}^{s} \omega_i^\star z_i \right|_m = \\ &= |\omega_0^\star + \omega_1^\star z_1 + \ldots + \omega_n^\star z_n|_m \,, \end{aligned}$$

который при подстановке $\omega_0^\star = \omega_0$, $\omega_i^\star = \omega_i$, $z_i = x_i$, $s = n$ соответствует полиному (2.7). Пусть $U = Y$, $\omega_0^\star = \psi_0$, $\omega_i^\star = \psi_i$, $z_i = x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$, $s = 2^n - 1$, тогда доказательство (2.7) сводится к повторению п. (i) доказательства теоремы 2.1.

□

Замечание 2.2
Значения параметра t *оператора* $\Xi^t\{U\}$ *при переходе от (1.7) к (2.7) не изменяются.*

Определение 2.2
Выражение (2.7) будем называть представлением булевой функции на основе модулярной формы линейного арифметического полинома.

Пример 2.2
Для системы булевых функций (1.8), заданной линейным арифметическим полиномом (1.9), параметр t *оператора* $\Xi^t\{U\}$ *имеет максимальное значение* $t_{\max} = 6$ *и* $U_{\max} = 36$. *Выберем* $m = 2^6 > 36$. *Тогда*

$$U = |20 + 60x_1 + 52x_2 + 16x_3|_{64} \,.$$

Пусть $x_1 \odot x_2 \odot x_3 = 0 \odot 1 \odot 1$. *Следовательно,*
$U = |88|_{64} = (24)_{10} = (0\,1\,1\,0\,0\,0)_2$. *Окончательно имеем:*

$$\begin{aligned} f_A(X) &= \Xi^2\{(0\,1\,1\,0\,0\,0)_2\} = 0, \\ f_B(X) &= \Xi^4\{(0\,1\,1\,0\,0\,0)_2\} = 1, \\ f_C(X) &= \Xi^6\{(0\,1\,1\,0\,0\,0)_2\} = 0. \end{aligned}$$

Связь оператора $\Xi^t\{U\}$ с модулярной арифметикой устанавливается отношением:

$$\Xi^t\{U\} = \left|\left\lfloor \frac{U}{2^t} \right\rfloor\right|_2 .$$

Замечание 2.3
Если для получения U используются избыточные булевы функции с номерами, превышающими $t_{\max}$ — максимальное значение параметра t оператора $\Xi^t\{U\}$, то модулю m можно присвоить значение $2^{t_{\max}}$. В этом случае вместо U в (2.7) следует писать $u = |U|_{2^{t_{\max}}}$, при этом $u \le U$.

Для пояснения замечания 2.3 рассмотрим следующий пример.

Пример 2.3
Для системы булевых функций

$$\begin{aligned} f_A(X) &= \overline{x_1 \wedge x_2 \wedge x_3} = \Xi^3\{6 - x_1 - x_2 - x_3\}, \\ f_B(X) &= \overline{x_1 \oplus x_2 \oplus x_3} = \Xi^1\{1 + x_1 + x_2 + x_3\} \end{aligned}$$

линейный арифметический полином $L(X)$ имеет вид:

$$U = 2^3 f_B(X) + 2^0 f_A(X) = 14 + 7x_1 + 7x_2 + 7x_3.$$

Так как $t_{\max} = 4$, то в соответствии с замечанием 2.3 получим $m = 16$ и

$$u = \lambda(X) = |14 + 7x_1 + 7x_2 + 7x_3|_{16} .$$

Пусть $x_1 \odot x_2 \odot x_3 = 1 \odot 1 \odot 1$, тогда $u = |35|_{16} = (2)_{10} = (0\,0\,1\,0)_2$ и

$$\begin{aligned} f_A(X) &= \Xi^3\{(0\,0\,1\,0)_2\} = 0, \\ f_B(X) &= \Xi^4\{(0\,0\,1\,0)_2\} = 0. \end{aligned}$$

Т. е. вместо шести разрядов, необходимых для представления U в соответствии с (1.7) и (2.7), замечание 2.3 позволяет обойтись только четырьмя разрядами.

Таким образом основным свойством модулярной формы полинома (2.1) является уменьшение числового диапазона, требуемого для его вычисления. Прежде чем сделать более точную оценку числового диапазона, рассмотрим принципы реализации матричных преобразований, основанных на модулярной арифметике.

2.2 Конъюнктивные модулярные преобразования

Теорема 2.2
Если для d-выходной булевой функции $f(X)$ задана пара ЛДПФ (1.10) и (1.11) и $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то справедлива следующая модулярная форма преобразований:

$$\mathbf{\Psi} = \mathbf{A}_{2^n}\mathbf{Y} \pmod m, \tag{2.8}$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi} \pmod m, \tag{2.9}$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования; $\mathbf{Y}$ и $\mathbf{\Psi}$ — соответственно вектор истинности булевой функции $f(X)$ и вектор коэффициентов модулярной формы арифметического полинома $\mu(X)$ (2.1). Запись $\bmod m$ означает, что арифметические операции, используемые при произведении матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ на вектор-столбец $\mathbf{Y}$ или $\mathbf{\Psi}$, выполняются по модулю m.

Для доказательства теоремы 2.2 необходимо учесть взаимооднозначность матричной (1.10), (1.11) и полиномиальной (1.3) формам представления системы булевых функций [71]. Тогда справедливость (2.8) и (2.9) вытекает из справедливости (2.1).

□

Полученная пара преобразований имеет много общего с теоретико-числовыми преобразованиями методов цифровой обработки сигналов (number theoretic transforms) [57, 64, 181].

Определение 2.3
Преобразования (2.8) и (2.9) будем соответственно называть модулярной формой прямого и обратного матричного арифметического преобразования или логическими теоретико-числовыми преобразованиями (ЛТЧП, logical number theoretic transforms).

Учитывая, что $|-1|_m = m-1$, выражение (2.8) можно переписать в другой форме:

$$\mathbf{\Psi} = \mathbf{M}_{2^n}\mathbf{Y} \pmod m, \tag{2.10}$$

где $\mathbf{M}_{2^n} = |\mathbf{A}_{2^n}|_m$. Запись $|\mathbf{A}_{2^n}|_m$ означает, что отрицательные элементы (единицы) матрицы $\mathbf{A}_{2^n}$ заменяются на $m-1$.

Принципы реализации прямого и обратного ЛТЧП (2.8) и (2.9) можно пояснить с помощью графов (соответственно рис. 2.6 и рис. 2.7), структура которых аналогична структурам графов матричных преобразований, предложенных Малюгиным в [71]. Из этих графов видно, что 1) в обоих преобразованиях применяются операции модулярного сложения; 2) все результаты промежуточных преобразований неотрицательны и не превышают значения модуля m; 3) для уменьшения вычислительных затрат к (2.8) и (2.9) могут быть применены методы факторизации матриц [161]

$$\mathbf{A}_{2^n} = \mathbf{A}_{2^n}^{(1)}\mathbf{A}_{2^n}^{(2)}\ldots\mathbf{A}_{2^n}^{(n)}$$

и

$$\mathbf{A}_{2^n}^{-1} = \widetilde{\mathbf{A}}_{2^n}^{(1)}\widetilde{\mathbf{A}}_{2^n}^{(2)}\ldots\widetilde{\mathbf{A}}_{2^n}^{(n)},$$

предложенные в [71]. При этом ожидается снижение вычислительной сложности преобразования в

$$\xi = \frac{3^n - 2^n}{n2^{n-1}}$$

раз [71].

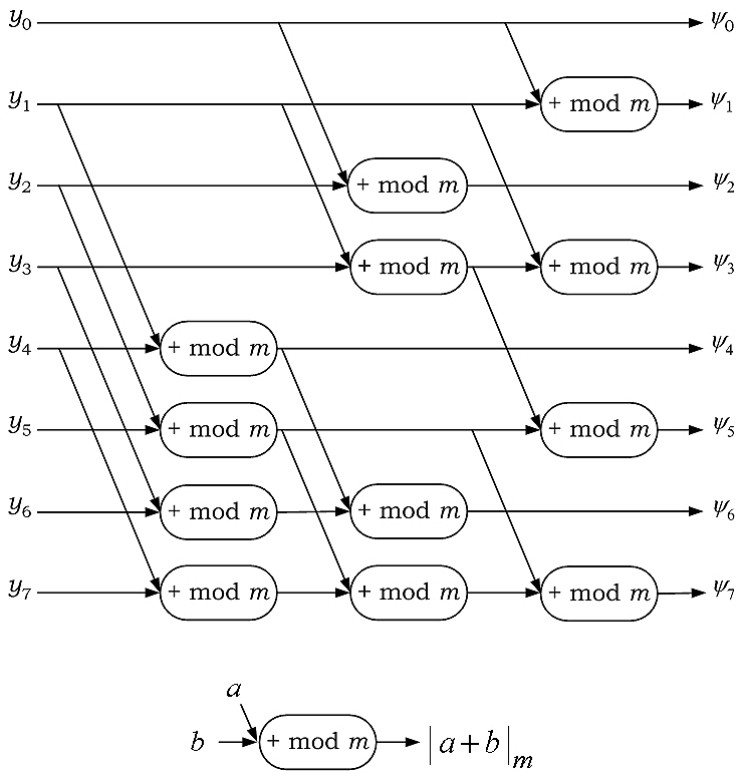


Рис. 2.6: Граф алгоритма прямого ЛТЧП

Пример 2.4

Продемонстрируем применение ЛТЧП (2.8) и (2.9) к двухвыходной булевой функции (1.2) с матрицей истинности, заданной табл. 1.1 (см. для сопоставления пример 1.2 на с. 23):

$$\mathbf{\Psi} = \mathbf{A}_{2^2}\mathbf{Y} \ (\mathbf{mod}\, 2^2) = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \hline -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \ (\mathrm{mod}\, 2^2) =$$

$$= \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ x_2 \\ x_1 \\ x_1x_2 \end{matrix},$$

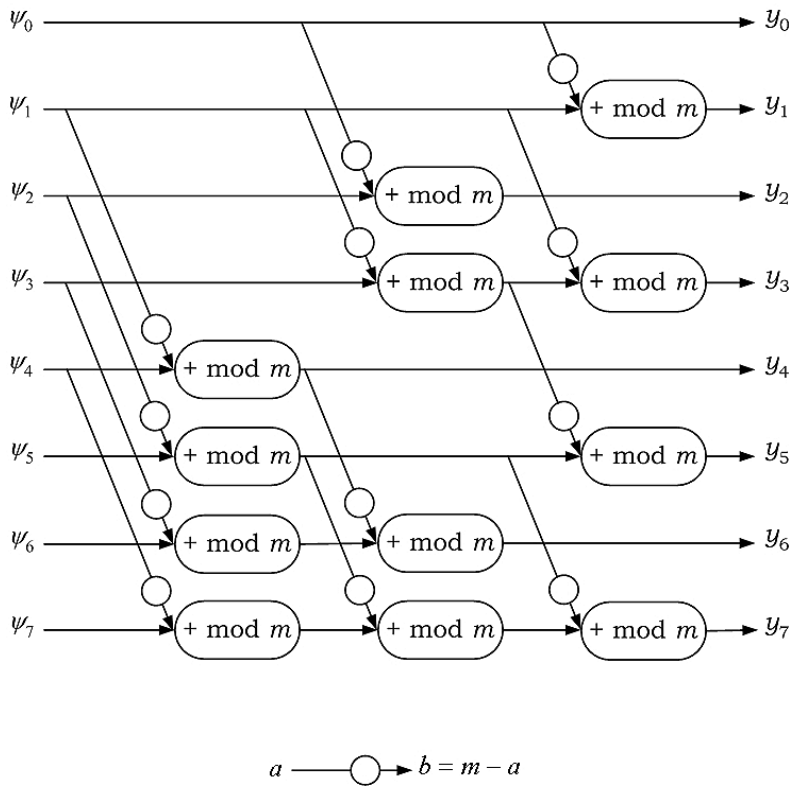


Рис. 2.7: Граф алгоритма обратного ЛТЧП

$$\mathbf{Y} = \mathbf{A}_{2^2}^{-1}\mathbf{\Psi} \pmod{2^2} = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \pmod{2^2} =$$

$$= \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Пример 2.5

Применение прямого ЛТЧП (2.10) при $m = 2^3$ *к трехвыходной*

булевой функции из примера 1.5 (с. 28) дает результат:

$$\begin{aligned}
\mathbf{\Psi} &= \mathbf{M}_{2^3}\mathbf{Y} \pmod{2^3} = \\
&= \left[\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 7 & 7 & 1 & 0 & 0 & 0 & 0 \\
\hline
7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 7 & 0 & 0 & 7 & 1 & 0 & 0 \\
1 & 0 & 7 & 0 & 7 & 0 & 1 & 0 \\
7 & 1 & 1 & 7 & 1 & 7 & 7 & 1
\end{array}\right] \cdot \left[\begin{array}{c} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{array}\right] \pmod{2^3} = \\
&= \left[\begin{array}{c} 0 \\ 7 \\ 6 \\ 4 \\ 5 \\ 6 \\ 0 \\ 3 \end{array}\right] \begin{array}{l} x_3 \\ x_2 \\ x_2x_3 \\ x_1 \\ x_1x_3 \\ x_1x_2 \\ x_1x_2x_3 \end{array} .
\end{aligned}$$

Пример 2.6

Для примера 1.6 (с. 29) реализации первой операции подстановки блока S_2 криптографического алгоритма стандарта DES прямое ЛТЧП (2.10) при $m = 2^4$ примет вид:

$$\begin{aligned}
\mathbf{\Psi} &= \mathbf{M}_{2^4}\mathbf{Y} \pmod{2^4} = \\
&= \mathbf{M}_{2^4} \cdot \left[\begin{array}{cccccccccccccccc} 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \end{array}\right]^T = \\
&= \left[\begin{array}{cccccccccccccccc} 15 & 2 & 9 & 4 & 7 & 3 & 4 & 8 & 10 & 12 & 0 & 9 & 12 & 3 & 4 & 12 \end{array}\right]^T .
\end{aligned}$$

Таким образом, значения коэффициентов вектора $\mathbf{\Psi}$ лежат в интервале $0 \le \psi_i < m$, $i = 0, 1, \ldots, 2^n - 1$, который сохраняется и при выполнении промежуточных вычислений (сравним с $-28 \le c_i \le 28$ в примере 1.5).

По аналогии с ЛДПФ в качестве оценки сложности ЛТЧП выберем размер матрицы-спектра. Для представления элементов матрицы $\mathbf{\Psi}$ потребуется $N_{\mathbf{\Psi}} = \lceil \log_2 m \rceil$ ($\lceil x \rceil$ — наименьшее целое число равное или превышающее x) двоичных разрядов или, при $m = 2^d$, $N_{\mathbf{\Psi}} = d$ двоичных разрядов, что в

$$\frac{N_{\mathbf{C}}}{N_{\mathbf{\Psi}}} = \frac{n}{d} + 1 \tag{2.11}$$

раз меньше по сравнению с количеством разрядов $N_{\mathbf{C}}$, необходимых для представления элементов матрицы $\mathbf{C}$ (1.12).

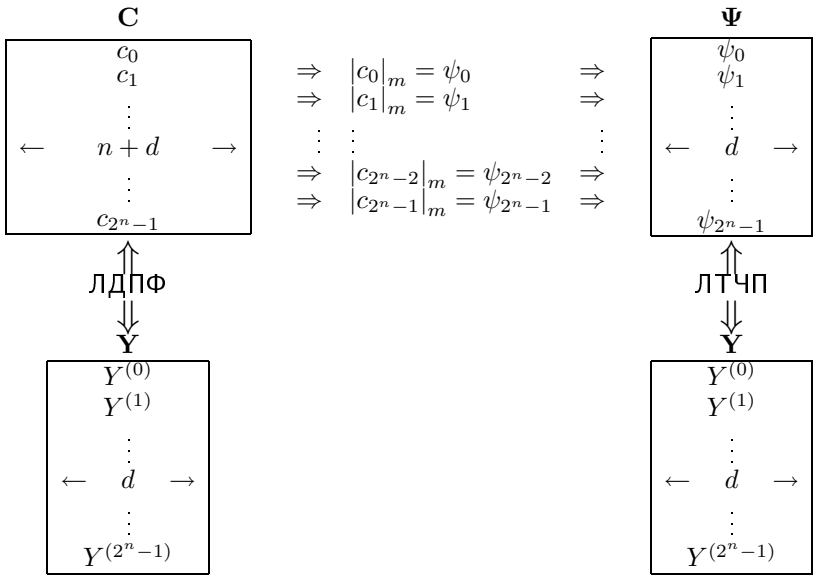


Рис. 2.8: Геометрическая интерпретация ЛТЧП и его связь с ЛДПФ

Так как $N_{\mathbf{C}}$ и $N_{\mathbf{\Psi}}$ — это максимальные размерности (количество двоичных разрядов) коэффициентов арифметических полиномов (1.3) и (2.1) соответственно, то оценка (2.11) применима и к арифметическому полиному (2.1).

На рис. 2.8 представлена геометрическая интерпретация получаемого выигрыша в виде представления матриц $\mathbf{Y}$, $\mathbf{C}$ и $\mathbf{\Psi}$ (здесь ширина матриц означает количество двоичных символов, которые необходимы для представления элементов матриц-столбцов, ПЛДПФ и ОЛДПФ — соответственно прямое и обратное ЛДПФ, а ПЛТЧП и ОЛТЧП — соответственно прямое и обратное ЛТЧП).

Однако этот выигрыш не удается сохранить для линейных арифметических полиномов, для которых числовой диапазон представления коэффициентов гарантированно можно уменьшить только в два раза — за счет переноса вычислений в область неотрицательных чисел (в некоторых случаях может быть использовано также замечание 2.3). Препятствием для дальнейшего уменьшения, используемого числового диапазона является

большая величина модуля m.

2.3 Модулярная форма логического ряда Тейлора

Определение 2.4
Модулярная форма булевой производной булевой функции $f(X)$ от n переменных по i-й переменной x_i ($i = 1, 2, \ldots, n$) определяется в виде:

$$\frac{\widetilde{d}f(X)}{\widetilde{d}x_i} = |(m-1)f(x_1, \ldots, \overline{x}_i, \ldots, x_n) + f(x_1, \ldots, x_i, \ldots, x_n)|_m , \tag{2.12}$$

где $m > 1$.

Определение 2.5
Модулярная форма логического ряда Тейлора булевой функции $f(X)$ в точке

$$x_1 = w_1, x_2 = w_2, \ldots, x_n = w_n$$

определяется выражением

$$MP_W(X) = \left| \xi(n) \sum_{j=0}^{2^n-1} p_W^{(j)} (x_1 \oplus w_1)^{j_1} (x_2 \oplus w_2)^{j_2} \ldots (x_n \oplus w_n)^{j_n} \right|_m , \tag{2.13}$$

где

$$\begin{aligned} W &= (w_1\, w_2 \ldots w_n)_2, \\ j &= (j_1\, j_2 \ldots j_n)_2, \\ p_W^{(j)} &= \left| \frac{\widetilde{d}^n f_{(W)}(X)}{\widetilde{d}x_1^{j_1} \widetilde{d}x_2^{j_2} \ldots \widetilde{d}x_n^{j_n}} \right|_m ; \\ \xi(n) &= 2^{n\varphi(m)-1} \pmod m; \end{aligned}$$

$\varphi(A)$ — функция Эйлера от числа A; модуль $m > 1$ — простое число.

Пример 2.7 *(продолжение примера 1.7)*
Модулярная форма арифметического полинома полярности $W = 3 = (0\,1\,1)_2$ *произвольной булевой функции трех переменных при* $m = 3$ *представляется рядом (2.13) следующим образом:*

$$P_3(X) = \Big|2\Big(f_3(X) + \tfrac{\tilde{d}f_3(X)}{\tilde{d}x_3}\overline{x}_3 + \tfrac{\tilde{d}f_3(X)}{\tilde{d}x_2}\overline{x}_2 + \tfrac{\tilde{d}^2f_3(X)}{\tilde{d}x_2\tilde{d}x_3}\overline{x}_2\overline{x}_3 + \\ + \tfrac{\tilde{d}f_3(X)}{\tilde{d}x_1}\overline{x}_1 + \tfrac{\tilde{d}^2f_3(X)}{\tilde{d}x_1\tilde{d}x_3}x_1\overline{x}_3 + \tfrac{\tilde{d}^2f_3(X)}{\tilde{d}x_1\tilde{d}x_2}x_1\overline{x}_2 + \tfrac{\tilde{d}^3f_3(X)}{\tilde{d}x_1\tilde{d}x_2\tilde{d}x_3}x_1\overline{x}_2\overline{x}_3\Big)\Big|_3 .$$

Для булевой функции $f(X) = x_2\overline{x}_2 \vee \overline{x}_3$ *модулярная форма арифметических полиномов (2.13) с учетом таблицы 1.2 имеет вид:*

$$\begin{aligned}
P_0(X) &= 1 + x_3 + x_1x_3 + x_1x_2x_3,\\
P_1(X) &= |2\overline{x}_3 + 2x_1 + 2x_1\overline{x}_3 + 2x_1x_2 + 2x_1x_2\overline{x}_3|_3 ,\\
P_2(X) &= |1 + x_3 + 2x_1\overline{x}_2x_3|_3 ,\\
P_3(X) &= |2\overline{x}_3 + x_1\overline{x}_2 + x_1\overline{x}_2\overline{x}_3|_3 ,\\
P_4(X) &= |1 + 2x_2x_3 + 2\overline{x}_1x_3 + 2\overline{x}_1x_2x_3|_3 ,\\
P_5(X) &= 1 + x_2 + x_2\overline{x}_3 + \overline{x}_1 + \overline{x}_1\overline{x}_3 + \overline{x}_1x_2 + \overline{x}_1x_2\overline{x}_3,\\
P_6(X) &= 1 + x_3 + \overline{x}_2x_3 + \overline{x}_1\overline{x}_2x_3,\\
P_7(X) &= |2\overline{x}_3 + 2\overline{x}_2 + 2\overline{x}_2\overline{x}_3 + 2\overline{x}_1\overline{x}_2 + 2\overline{x}_1\overline{x}_2\overline{x}_3|_3 .
\end{aligned}$$

2.4 Выводы

- Предложен метод реализации систем булевых функций посредством модулярной формы арифметического полинома, которая классифицирована как *арифметическое обобщение полинома Жегалкина* [1, 43].

- Установлено, что модулярная форма арифметического полинома позволяет существенно ослабить проблему больших коэффициентов и ограничить величину результатов промежуточных вычислений.

- Предложено прямое аналитическое преобразование систем булевых функций непосредственно в модулярную форму арифметического полинома.
- Установлена взаимосвязь между немодулярной и модулярной формами арифметического полинома.
- Построены прямое и обратное матричные преобразования между системами булевых функций и модулярной формой арифметических полиномов. Полученные матричные преобразования определены как *логические теоретико-числовые преобразования*.
- Показано, что логические теоретико-числовые преобразования могут иметь структуру, аналогичную быстрым конъюнктивным преобразованиям, предложенным В. Малюгиным.
- Предложена модулярная форма логического ряда Тейлора. Эта форма не имеет преимуществ перед арифметическим аналогом логического ряда Тейлора, однако, может быть полезной для обеспечения возможности ее реализации техническими средствами, функционирующими в модулярной арифметике.

Глава 3

Многомерные формы, основанные на Китайской теореме

В данной главе[1] введенные ранее модулярные арифметико-логические формы развиваются в многомерные отображения в кольце $Z_{m_1, m_2, ..., m_v}$.

3.1 Полиномиальные многомерные формы, основанные на Китайской теореме

При моделировании реальных цифровых устройств значения коэффициентов линейных арифметических полиномов могут превышать величину 2^{100} [140, 193]. В таких случаях требуются более радикальные пути уменьшения числовых диапазонов.

Пусть модуль m для (2.1) и (2.7) обладает свойством

$$m = m_1 m_2 \dots m_k,$$

причем $\gcd(m_i, m_j) = 1$; $i, j = 1, 2, \dots, v$; $i \neq j$ (здесь и далее $\gcd(a, b)$ — наибольший общий делитель a и b). Тогда в соответствии с Китайской теоремой об остатках Y можно взаимно однозначно отобразить в v-мерный вектор $\{Y\} = (\phi_1, \phi_2, \dots, \phi_v)$, где

[1] Основное содержание данной главы впервые апробировано и опубликовано (принято к опубликованию) в [113, 118, 120, 121, 122, 127, 128, 130].

$\phi_k = |Y|_{m_k}$ $(k = 1, 2, \ldots, v)$ [12, 16, 50]. При этом $Y \in Z_m$. Применение для каждого вычета ϕ_k $(k = 1, 2, \ldots, v)$ рассмотренного выше подхода позволяет получить следующие положения.

Теорема 3.1
Если $m > Y_{\max}$, *причем* $m = \prod_{k=1}^{v} m_k$ *и* $\gcd(m_i, m_j) = 1$ $(i, j = 1, 2, \ldots, v; i \neq j)$, *то произвольный кортеж булевых функций может быть однозначно представлен системой модулярных форм арифметических полиномов:*

$$\begin{cases} \phi_1 = \mu_1(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,1}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_1}, \\ \phi_2 = \mu_2(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,2}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_2}, \\ \vdots \\ \phi_v = \mu_v(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,v}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_v}, \end{cases} \tag{3.1}$$

где $\psi_{i,k} = |c_i|_{m_k}$ $(i = 0, 1, \ldots, 2^n - 1;\ k = 1, 2, \ldots, v)$.

Доказательство
По аналогии с доказательством теоремы 2.1 рассмотрим два варианта доказательств, определяющих два варианта построения алгоритма получения (3.1).

(i) Так как (2.2) справедливо для любых значений модуля $m > 1$, то справедливы и v записей (2.2) для различных значений модуля $m > 1$:

$$\begin{aligned} |Y|_{m_1} &= \left| \sum_{i=0}^{2^n-1} c_i\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_1}, \\ |Y|_{m_2} &= \left| \sum_{i=0}^{2^n-1} c_i\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_2}, \\ &\vdots \\ |Y|_{m_v} &= \left| \sum_{i=0}^{2^n-1} c_i\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_v} \end{aligned}$$

или

$$\begin{aligned}
|Y|_{m_1} &= \left| \sum_{i=0}^{2^n-1} |c_i|_{m_1} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_1}, \\
|Y|_{m_2} &= \left| \sum_{i=0}^{2^n-1} |c_i|_{m_2} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_2}, \\
&\vdots \\
|Y|_{m_v} &= \left| \sum_{i=0}^{2^n-1} |c_i|_{m_v} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_v}.
\end{aligned} \tag{3.2}$$

На основании Китайской теоремы об остатках система вычетов

$$\phi_1 = |Y|_{m_1},\ \phi_2 = |Y|_{m_2},\ \dots,\ \phi_v = |Y|_{m_v}$$

имеет единственное решение Y, если выполнены условия: $Y_{\max} < \prod_{k=1}^{v} m_k$; $\gcd(m_i, m_j) = 1$ $(i, j = 1, 2, \dots, v;\ i \neq j)$. Так как эти условия в теореме 3.1 определены, то и (3.1) имеет единственное решение Y.

□

Из данного доказательства следует, что построить (3.1) можно посредством выполнения двух шагов:

Шаг. 1. Построение (1.3) посредством алгоритма 1.1.

Шаг. 2. Построение системы (3.1) посредством (7.9).

(ii) Используем (2.4) для построения системы арифметических полиномов по v заданным модулям, отвечающим требованиям однозначности представления (3.1):

$$P_j(X) = \begin{cases}
\mu'_{j,1}(X) = \left| \sum_{i=0}^{2^n-1} \psi'_{j,i,1} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_1}, \\
\mu'_{j,2}(X) = \left| \sum_{i=0}^{2^n-1} \psi'_{j,i,2} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_2}, \\
\vdots \\
\mu'_{j,v}(X) = \left| \sum_{i=0}^{2^n-1} \psi'_{j,i,v} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_v},
\end{cases} \tag{3.3}$$

где $\psi'_{j,\,i,\,k} = |r_{j,\,i}|_{m_k}$ ($j = 1, 2, \ldots, d$; $i = 1, 2, \ldots, 2^n - 1$; $k = 1, 2, \ldots, v$). Аналогично используя (2.5), получим:

$$\begin{cases} \mu''_{j,\,1}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,\,i,\,1}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ \mu''_{j,\,2}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,\,i,\,2}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ \vdots \\ \mu''_{j,\,v}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,\,i,\,v}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \end{cases} \tag{3.4}$$

где

$$\psi''_{j,\,i,\,k} = \left|\psi'_{j,\,i,\,k} b_{j,\,k}\right|_{m_k};$$
$$b_{j,\,1} = |2^{j-1}|_{m_1},\ b_{j,\,2} = |2^{j-1}|_{m_2},\ \ldots\ b_{j,\,s} = |2^{j-1}|_{m_s}$$

($j = 1, 2, \ldots, d$; $i = 0, 1, \ldots, 2^n - 1$; $k = 1, 2, \ldots, v$). Окончательно, обобщая (2.5) для модулей $m_1, m_2, \ldots, m_v$ получают (3.1) с помощью выражений

$$\psi_{j,\,k} = \left|\sum_{j=1}^{d} \psi''_{j,\,i,\,k}\right|_{m_k} \tag{3.5}$$

($i = 0, 1, \ldots, 2^n - 1$; $k = 1, 2, \ldots, v$).

□

Алгоритм получения (3.1) состоит в выполнении трех шагов, заключающихся соответственно в построении систем полиномов (7.10), (3.4) и собственно (3.1) при помощи (3.5).

Достоинством первых вариантов алгоритмов формирования арифметических полиномов (2.1) и (3.1) является простота и наглядность. Особенностью вторых вариантов является формирование соответствующих арифметических полиномов, обеспечивающих модулярность преобразований на всех этапах реализации алгоритма, что дает „удержание“ промежуточных результатов вычислений в заданном числовом диапазоне.

Определение 3.1
Систему арифметических полиномов (3.1) будем называть полиномиальной многомерной формой представления булевых функций, основанной на Китайской теореме об остатках.

Замечание 3.1
Модулярные формы (3.1) и (2.1) связаны отношениями:

$$\begin{aligned} \{Y\} &= (\phi_1, \phi_2, \ldots, \phi_v), \\ (\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,v}) &= \{\psi_i\} = |c_i|_m \ (i = 0, 1, \ldots, 2^n - 1). \end{aligned}$$

Для каждого арифметического полинома системы (3.1) справедливы и следствия 2.1 и 2.2 (при этом вместо m необходимо рассматривать соответствующий модуль m_j $(j = 1, 2, \ldots, v)$.

Лемма 3.1
Если кортеж булевых функций (1.1) задан линейным арифметическим полиномом $L(X)$ (1.7), то при $m > U_{max}$, где $m = \prod_{k=1}^{v} m_k$, причем $\gcd(m_i, m_j) = 1$ $(i, j = 1, 2, \ldots, v;\ i \neq j)$, справедлива следующая модулярная форма линейного арифметического полинома:

$$\begin{cases} \phi_1 = \lambda_1(X) = |\omega_{0,1} + \omega_{1,1}x_1 + \ldots + \omega_{n,1}x_n|_{m_1}, \\ \phi_2 = \lambda_2(X) = |\omega_{0,2} + \omega_{1,2}x_1 + \ldots + \omega_{n,2}x_n|_{m_2}, \\ \vdots \\ \phi_v = \lambda_v(X) = |\omega_{0,v} + \omega_{1,v}x_1 + \ldots + \omega_{n,v}x_n|_{m_v}, \end{cases} \tag{3.6}$$

где $\omega_{j,k} = |d_i|_{m_k}$ $(j = 0, 1, \ldots, n;\ k = 1, 2, \ldots, v)$.

Справедливость (3.6) следует из применения доказательства справедливости (2.7) для каждого номера модуля (3.6) в отдельности и из Китайской теоремы об остатках.

Определение 3.2
Систему арифметических полиномов (3.6) будем называть линейной полиномиальной многомерной формой представления булевых функций, основанной на Китайской теореме об остатках.

Замечание 3.2
Модулярные формы (3.6) и (2.7) связаны отношениями:

$$\begin{aligned} \{U\} &= (\phi_1, \phi_2, \ldots, \phi_v), \\ (\omega_{j,1}, \omega_{j,2}, \ldots, \omega_{j,v}) &= \{\omega_j\} = \{\omega_j\} = |d_j|_m \ (j = 0, 1, \ldots, n). \end{aligned}$$

Замечание 2.3 к (3.6) не применимо.

Для упрощения изложения в дальнейшем не будем различать числа Y и U.

Решение системы равенств

$$\begin{cases} Y = \phi_1 \pmod{m_1}, \\ Y = \phi_2 \pmod{m_2}, \\ \vdots \\ Y = \phi_v \pmod{m_v} \end{cases}$$

дает Китайская теорема об остатках. Для этого будем использовать запись

$$Y = \mathbf{CRT}_{k=1}^{v} \phi_k \pmod{m_k}. \tag{3.7}$$

В современной трактовке Китайской теоремы об остатках для вычисления (3.7) используется формула

$$Y = \mathbf{CRT}_{k=1}^{v} \phi_k \pmod{m_k} = |\phi_1 B_1 + \phi_2 B_2 + \ldots + \phi_v B_v|_m, \tag{3.8}$$

где $B_k = q_k m m_k^{-1}$; q_k находится из $q_k m m_k^{-1} \equiv 1 \pmod{m_k}$ $(k = 1, 2, \ldots, v)$.

Рассмотрим далее два примера реализации систем булевых функций с использованием Китайской теоремы об остатках.

Пример 3.1
Рассмотрим реализацию универсального многополюсника ((2, 16)-полюсника), арифметическая форма выходных функций

которого имеет вид [71]:

$$
\begin{array}{ll}
f_1 = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1x_2, & f_9 = \overline{x_1} \wedge x_2 = x_2 - x_1x_2, \\
f_2 = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1x_2, & f_{10} = x_1 \wedge x_2 = x_1x_2, \\
f_3 = \overline{x_1 \wedge \overline{x_2}} = 1 - x_1 + x_1x_2, & f_{11} = \overline{x_1} = 1 - x_1, \\
f_4 = \overline{\overline{x_1} \wedge x_2} = 1 - x_2 + x_1x_2, & f_{12} = x_1, \\
f_5 = \overline{x_1 \wedge x_2} = 1 - x_1x_2, & f_{13} = \overline{x_2} = 1 - x_2, \\
f_6 = x_1 \oplus x_2 = x_1 + x_2 - 2x_1x_2, & f_{14} = x_2, \\
f_7 = x_1 \vee x_2 = x_1 + x_2 - x_1x_2, & f_{15} = 1, \\
f_8 = x_1 \wedge \overline{x_2} = x_1 - x_1x_2, & f_{16} = 0.
\end{array}
$$

Результирующий арифметический полином в соответствии с (1.3) имеет вид [71]:

$$D(X) = 21535 + 1241x_1 + 4437x_2.$$

Максимальное значение, принимаемое полином при $x_1 = 1$ *и* $x_2 = 1$, *равно* 27213. *Поэтому выберем попарно простые модули* $m_1 = 11$, $m_1 = 13$, $m_1 = 15$ *и* $m_1 = 16$ *такие, что* $M = 11 \cdot 13 \cdot 15 \cdot 16 = 34320 > 27213$. *Используя один из алгоритмов, вытекающих из доказательства теоремы 3.1, получим модулярные формы арифметических полиномов в соответствии с (3.1):*

$$
\begin{aligned}
\mu_1(X) &= |8 + 9x_1 + 4x_2 + 0x_1x_2|_{11}\,, \\
\mu_2(X) &= |7 + 6x_1 + 4x_2 + 0x_1x_2|_{13}\,, \\
\mu_3(X) &= |10 + 11x_1 + 12x_2 + 0x_1x_2|_{15}\,, \\
\mu_4(X) &= |15 + 9x_1 + 5x_2 + 0x_1x_2|_{16}\,.
\end{aligned}
$$

Далее легко построить таблицу расчетных значений ϕ_1, ϕ_2, ϕ_3, ϕ_4 *и, с помощью (3.8), —* Y *для всех наборов булевых переменных (табл. 3.1).*

Кортежи значений выходов $(2, 16)$*-полюсника* $f_{16} \odot \ldots \odot f_1$ *на всех наборах булевых переменных* $x_1 \odot x_2$ *получаются из представлений* Y *в двоичной системе счисления:*

$$
\begin{aligned}
(21535)_{10} &= (0101010000011111)_2, \\
(25972)_{10} &= (0110010101110100)_2,
\end{aligned}
$$

Таблица 3.1: Таблица расчетных значений модулярных форм арифметических полиномов на всех наборах булевых переменных

$x_1 \odot x_2$	ϕ_1	ϕ_2	ϕ_3	ϕ_4	Y
$0 \odot 0$	8	7	10	15	21535
$0 \odot 1$	1	11	7	4	25972
$1 \odot 0$	6	0	6	8	22776
$1 \odot 1$	10	4	3	13	27213

$$\begin{aligned}(22776)_{10} &= (0101100011111000)_2,\\ (27213)_{10} &= (0110101001001101)_2.\end{aligned}$$

Пример 3.2
Рассмотрим линейный арифметический полином [193]:

$$Y = L(X) = 8 + 137x_1 - 7x_2 + 129x_3 + 136x_4 + 64x_5,$$

реализующий систему булевых функций:

$$\begin{aligned}f_1(X) &= x_1 \oplus x_2 \oplus x_3,\\ f_2(X) &= x_1 \oplus \overline{x}_2 \oplus x_3,\\ f_3(X) &= x_5,\\ f_4(X) &= x_1 \oplus x_3 \oplus x_4.\end{aligned}$$

Причем

$$\begin{aligned}f_1(X) &= \Xi^1\{Y\},\\ f_2(X) &= \Xi^4\{Y\},\\ f_3(X) &= \Xi^7\{Y\},\\ f_4(X) &= \Xi^8\{Y\}.\end{aligned}$$

Так как $t_{\max} = 8$ *и* $Y_{\max} = 474$, *выберем модули* $m_1 = 7$, $m_2 = 8$, $m_3 = 9$, *обеспечивающие выполнение условия*

$$m = m_1 m_2 m_3 = 504 > 474.$$

Тогда согласно (3.6):

$$\begin{aligned}\lambda_1(X) &= |1+4x_1+3x_3+3x_4+x_5|_7\,,\\ \lambda_2(X) &= |x_1+x_2+x_3|_8\,,\\ \lambda_3(X) &= |8+2x_1+2x_2+3x_3+x_4+x_5|_9\,.\end{aligned}$$

Пусть $x_1 \odot x_2 \odot x_3 \odot x_4 \odot x_5 = 1 \odot 0 \odot 0 \odot 1 \odot 1$. *Тогда*

$$\begin{aligned}\lambda_1(X) &= 2,\\ \lambda_2(X) &= 1,\\ \lambda_3(X) &= 3.\end{aligned}$$

Используя формулу (3.8) и учитывая, что $B_1 = 288$, $B_2 = 441$, $B_3 = 280$, *получим*

$$Y = |2\cdot 288 + 1\cdot 441 + 3\cdot 280|_{504} = |1857|_{504} = (101011001)_2.$$

Несмотря на классический вид формулы (3.8) она не всегда удобна для практического использования, в частности, из-за необходимости обеспечения большого числового диапазона. Другие, более приемлемые методы восстановления позиционной формы числа, будут рассмотрены в главе 7.

Схема, поясняющая принцип реализации булевых функций посредством модулярных форм арифметических полиномов, основанных на Китайской теореме об остатках, представлена на рис. 3.1.

3.2 Конъюнктивные многомерные преобразования, основанные на Китайской теореме

Лемма 3.2
Если для d*-выходной булевой функции* $f(X)$ *задана пара ЛТЧП (2.8) и (2.9) и* $m > Y_{max}$, *причем* $m = \prod_{k=1}^{v} m_k$ *и* $\gcd(m_i, m_j)$ $(i, j = 1, 2, \ldots, v;\ i \neq j)$, *то справедлива следующая модулярная арифметико-логическая форма преобразований:*

$$\begin{cases} \mathbf{\Psi}_1 = \mathbf{A}_{2^n}\mathbf{\Phi}_1 \pmod{m_1},\\ \mathbf{\Psi}_2 = \mathbf{A}_{2^n}\mathbf{\Phi}_2 \pmod{m_2},\\ \vdots\\ \mathbf{\Psi}_v = \mathbf{A}_{2^n}\mathbf{\Phi}_v \pmod{m_v}; \end{cases} \tag{3.9}$$

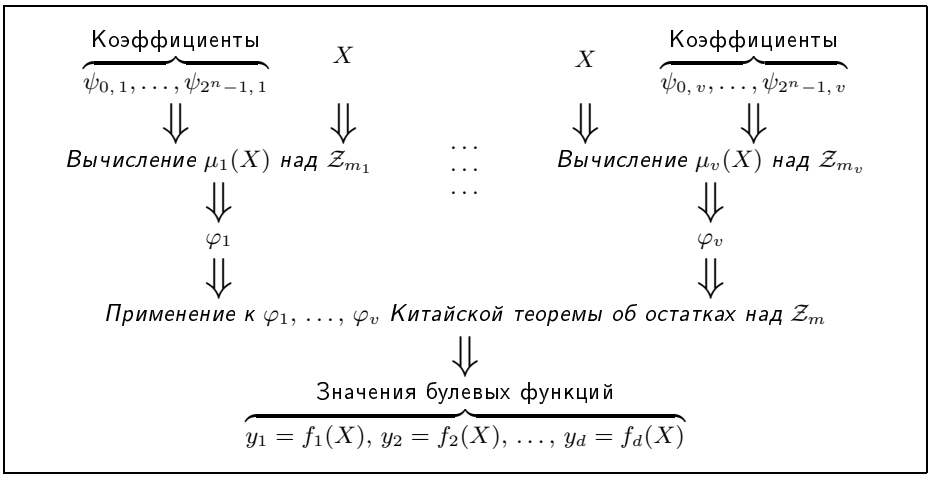


Рис. 3.1: Схема параллельного вычисления заданной системы булевых функций над $Z_{m_1,m_2,\ldots,m_v}$ на основе Китайской теоремы об остатках

$$\begin{cases} \mathbf{\Phi}_1 = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_1 \pmod{m_1}, \\ \mathbf{\Phi}_2 = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_2 \pmod{m_2}, \\ \vdots \\ \mathbf{\Phi}_v = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_v \pmod{m_v}, \end{cases} \tag{3.10}$$

где $\mathbf{A}_{2^n}$ *и* $\mathbf{A}_{2^n}^{-1}$ *— соответственно матрицы прямого и инверсного арифметического преобразования;*

$$\mathbf{\Phi}_k = [\phi_{0,k}\,\phi_{1,k}\,\ldots\,\phi_{2^n-1,k}]^T;\quad \phi_{i,k} = \left|Y^{(i)}\right|_{m_k};\quad k=1,\ldots,v;$$

$$\mathbf{\Psi}_k = [\psi_{0,k}\,\psi_{1,k}\,\ldots\,\psi_{2^n-1,k}]^T;\quad k=1,\ldots,v.$$

Справедливость леммы и (3.9) и (3.10) следует 1) из взаимооднозначности связи матричной (1.10) и (1.11) и полиномиальной (1.3) форм представления булевых функций [71] и 2) из доказательства справедливости полиномиальной формы представления (3.1), основанной на Китайской теореме об остатках.

Определение 3.3
Системы матричных преобразований (3.9) и (3.10) будем назы-

вать конъюнктивными многомерными преобразованиями, основанными на Китайской теореме об остатках или ЛТЧП, основанными на Китайской теореме об остатках (ЛТЧП КТО).

Продемонстрируем применение ЛТЧП КТО и некоторые полезные свойства этого преобразования, связанные с сокращением длин соответствующих арифметических полиномов, на примере получения спектров коэффициентов арифметических полиномов, соответствующих операции подстановки криптографического алгоритма стандарта DES (пример 1.6 с. 29).

Пример 3.3
Для подстановки

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \end{pmatrix}.$$

в соответствии с леммой 3.2 необходимо, чтобы $m_1 m_2 \geq 2^4$.

Для выбора модулей, целесообразных по критерию сложности получаемых арифметических полиномов, построим таблицу 3.2 (для рассматриваемой подстановки), где коэффициент уменьшения сложности полинома определим как $\frac{K}{2^n}$; K *— количество ненулевых коэффициентов. Учитывая требование обеспечения попарной простоты модулей, выберем* $m_1 = 3$, $m_2 = 7$. *Тогда*

$$\begin{cases} \mathbf{\Psi}_1 = \mathbf{A}_{2^4}\mathbf{\Phi}_1 \pmod 3, \\ \mathbf{\Psi}_2 = \mathbf{A}_{2^4}\mathbf{\Phi}_2 \pmod 7, \end{cases}$$

где

$$\mathbf{\Phi}_1 = \begin{bmatrix} 0 & 1 & 2 & 2 & 0 & 2 & 0 & 1 & 0 & 1 & 2 & 1 & 0 & 0 & 2 & 1 \end{bmatrix}^T \pmod 3,$$
$$\mathbf{\Phi}_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 6 & 4 & 3 & 4 & 2 & 0 & 2 & 6 & 5 & 0 & 5 & 3 \end{bmatrix}^T \pmod 7.$$

Тогда

$$\mathbf{\Psi}_1 = \begin{bmatrix} 3 & 1 & 2 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 \end{bmatrix}^T \pmod 3,$$
$$\mathbf{\Psi}_2 = \begin{bmatrix} 1 & 0 & 0 & 6 & 5 & 5 & 4 & 4 & 1 & 12 & 0 & 0 & 5 & 5 & 6 & 0 \end{bmatrix}^T \pmod 7.$$

Таблица 3.2: Оценка зависимости сложности арифметического полинома от значения модуля

Значение модуля	Длина полинома (сложность)	Коэффициент уменьшения
2	7	0,56
3	9	0,47
4	8	0,5
5	12	0,25
6	11	0,31
7	11	0,31
8	14	0,12
9	14	0,12
10	13	0,19
11	15	0,06
12	13	0,19
13	15	0,06
14	13	0,19
15	15	0,06
16	15	0,06

Замечание 3.3
ЛТЧП КТО (3.9) и (3.10) связаны с ЛТЧП (2.8) и (2.9) следующими отношениями

$$\mathbf{\Psi} = \mathbf{CRT}_{k=1}^{v}\mathbf{\Psi}_k \pmod{m_k},$$

$$\mathbf{Y} = \mathbf{CRT}_{k=1}^{v}\mathbf{\Phi}_k \pmod{m_k}.$$

Структура графов матричных преобразований для каждого номера k модуля m_k ЛТЧП КТО аналогична структурам графов, представленным на рис. 2.6 и рис. 2.7. Следовательно, к ЛТЧП КТО применимы методы факторизации матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ (методы быстрых конъюнктивных преобразований), применимые и к ЛДПФ [71] и ЛТЧП.

На рис. 3.2 показана геометрическая интерпретация ЛТЧП КТО и его взаимосвязь с ЛДПФ и ЛТЧП. Согласно этой диаграмме смысл ЛТЧП КТО сводится к разложению каждой из матриц $\mathbf{Y}$ и $\mathbf{C}$ на v матриц меньшей „ширины“ — $l_1, l_2, \ldots, l_v$, где $l_k = \lceil \log_2 m_k \rceil$, что позволяет упростить преобразование для каждой из полученных матриц $\mathbf{\Psi}_k$ или $\mathbf{\Phi}_k$ в отдельности. Полученные результаты затем восстанавливаются с помощью Китай-

ской теоремы об остатках. При этом спектр $\mathbf{\Psi}$ является матрицей ЛТЧП по модулю $m = \prod_{k=1}^{v} m_k$.

Взаимосвязь ЛТЧП КТО и ЛТЧП поясняется с помощью рис. 3.3.

Будем исходить из предположения, что преобразования для всех модулей ЛТЧП КТО выполняются параллельно. Тогда наибольшую сложность будет иметь преобразование ЛТЧП КТО для наибольшего модуля m_v. Максимальная „ширина“ (необходимое количество двоичных разрядов для представления элементов матрицы) соответствует матрице $\mathbf{\Psi}_v$ по наибольшему модулю m_v и составит

$$l_v = \lceil \log_2 m_v \rceil$$

двоичных разрядов. Учитывая, что для большинства практических задач значение l_v не превышает $6 \div 7$, выигрыши в сравнении с ЛДПФ ЛТЧП соответственно составят

$$\frac{N_{\mathbf{C}}}{l_v} \approx \frac{n+d}{6 \div 7} \text{ и } \frac{N_{\mathbf{\Psi}}}{l_v} \approx \frac{d}{6 \div 7} \text{ раз.}$$

Например, при $n = d = 40$ и $m_v = 128$ выигрыши будут соответственно 11, 4 и 5, 7 раз. Например, набор из 17 модулей (табл. 3.3) обеспечивает представление коэффициентов арифметического полинома (1.3) или (1.7) с верхней границей — 2^{105}. Полученные оценки сохраняют силу и для модулярных форм

Таблица 3.3:

m_1 67	m_2 71	m_3 73	m_4 79	m_5 83	m_6 89	m_7 91	m_8 97	m_9 101	m_{10} 103
m_{11} 107	m_{12} 109	m_{13} 111	m_{14} 113	m_{15} 115	m_{16} 127	m_{17} 128			

полиномиальных преобразований, основанных на Китайской теореме об остатках. При этом для сравнения с линейным арифметическим полиномом необходимо использовать отношение $\frac{d^\star}{6 \div 7}$, где $d^\star$ — общее количество (с учетом избыточных) реализуемых булевых функций или максимальное количество разрядов, необходимых для представления коэффициентов

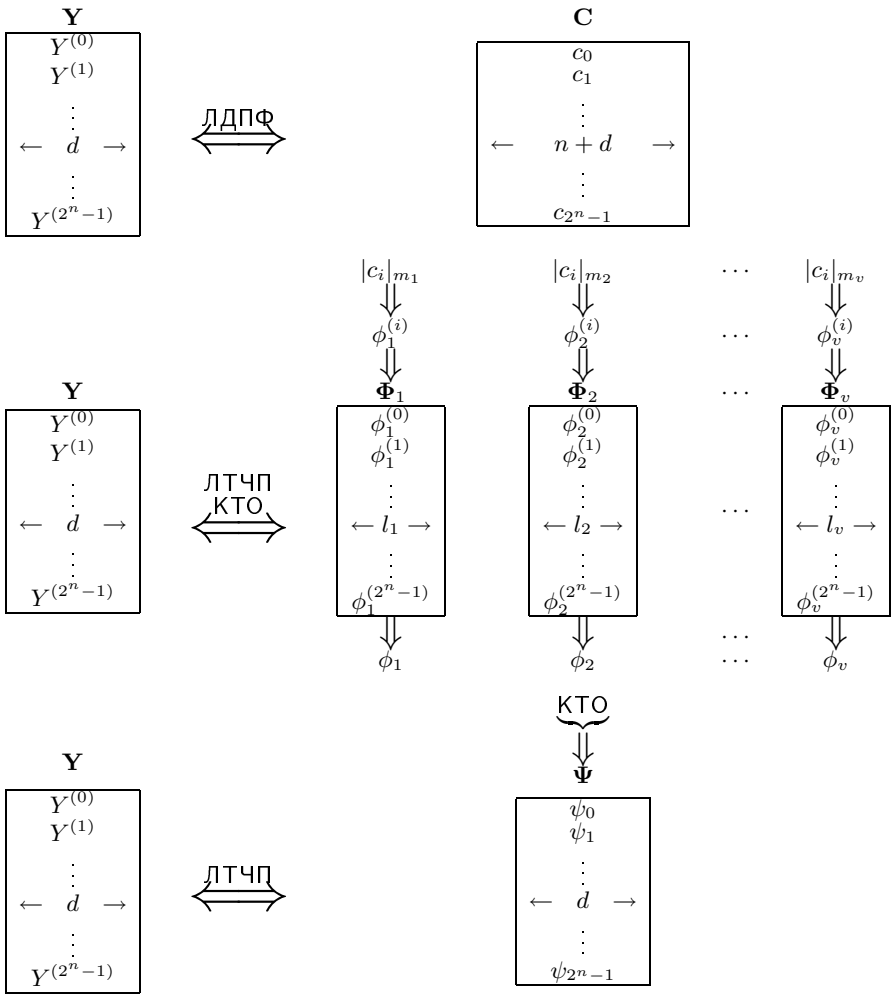


Рис. 3.2: Геометрическая интерпретация ЛТЧП КТО и его взаимосвязь с ЛДПФ и ЛТЧП

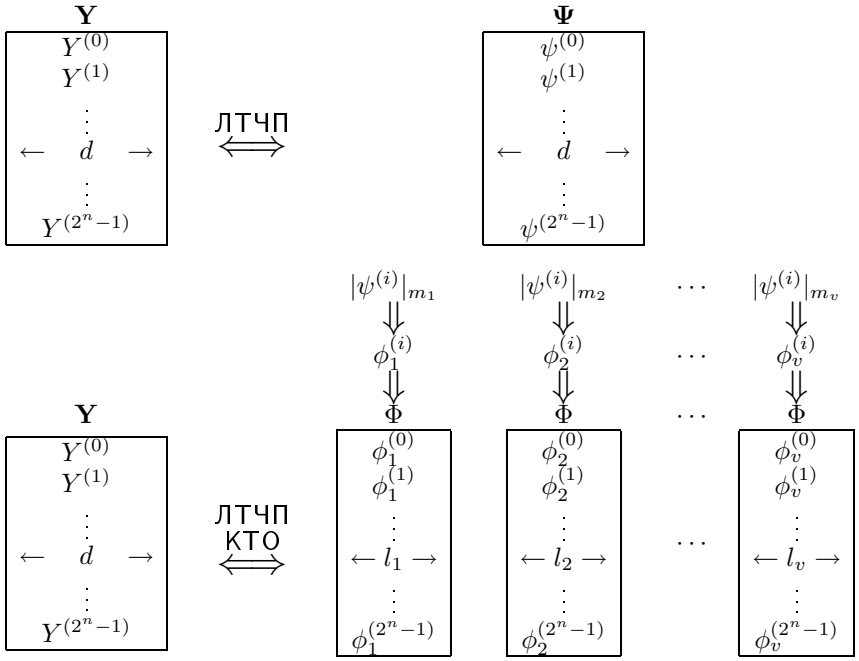


Рис. 3.3: Геометрическая интерпретация взаимосвязи ЛТЧП и ЛТЧП КТО

полинома. Относительно $d^\star = 2^{100}$ линейного арифметического полинома выигрыш в количестве используемых двоичных разрядных цифр в пределах одного канала (модуля) преобразования составит 14, 3 раза.

3.3 Выводы

- Предложен метод реализации систем булевых функций посредством арифметической полиномиальной формы, основанной на Китайской теореме об остатках, которая представляет собой систему модулярных форм арифметических полиномов по системе специальным образом выбранных модулей.

- Показано, что предложенный метод, основанный на Китай-

ской теореме об остатках, позволяет полностью решить проблему больших коэффициентов как нелинейных, так и линейных арифметических полиномов, при условии обеспечения распараллеливания вычислений по системе выбранных модулей.

- Предложено прямое аналитическое преобразование систем булевых функций непосредственно в арифметическую полиномиальную форму, основанную на Китайской теореме об остатках.
- Установлена взаимосвязь между немодулярной, модулярной и арифметической полиномиальной формой, основанной на Китайской теореме об остатках.
- Построены прямое и обратное матричные преобразования между системами булевых функций и арифметической полиномиальной формой, основанной на Китайской теореме об остатках. Полученные матричные преобразования определены как *логические теоретико-числовые преобразования, основанные на Китайской теореме об остатках.*
- Указано, что по аналогии с быстрыми конъюнктивными преобразованиями к логическим теоретико-числовым преобразованиям, основанным на Китайской теореме об остатках, могут быть применены методы факторизации матриц.

Глава 4

Организация вычислений в больших числовых диапазонах

Необходимость реализации систем логических функций больших размерностей ставит задачу организации параллельных вычислений в больших числовых диапазонах — $2^{100} \dots 2^{1000}$ и более. Здесь под *большим* числовым диапазоном понимается числовой диапазон, который на несколько порядков превышает *типовой* числовой диапазон. Эффективное решение задачи таких вычислений состоит в организации логических вычислений на основе *многопроцессорных* или *многомашинных* вычислительных систем.

4.1 Принцип больших модулей

Если сопоставить каждому модулю вычислитель, который бы функционировал по этому модулю, то совокупность таких вычислителей, объединенных связями, может представлять собой вычислительную систему, функционирующую в числовом диапазоне, равному произведению этих модулей. Ясно, что количество таких вычислителей должно быть минимизировано (без существенной потери производительности). Добиться этого можно, если значения модулей, сопоставимы с числовыми диапазонами, аппаратурно поддерживаемыми этими вычислителями (например $\approx 2^k$, где $k = 16, 32, 64 \dots$).

Пример организации такой вычислительной системы представлен на рис. 4.1. Восстановление окончательного результата

(позиционной формы числа) может быть выполнено как на основе одного из вычислителей (или некоторой совокупности вычислителей), так и на основе вычислителя, специально выделенного для этой цели.

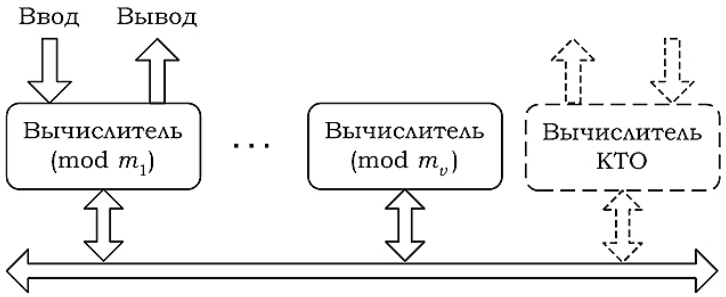


Рис. 4.1: Система, построенная на основе вычислителей, функционирующих по большим модулям (здесь „Вычислитель КТО“ — вычислитель, реализующий Китайскую теорему об остатках)

Изучению принципов построения таких вычислительных систем был посвящен ряд работ [36, 48, 49]. В частности в [48, 49] исследовались принципы вычислений в квадратичных больших и сверхбольших числовых диапазонах на основе модулярной арифметики, заданной квадратами модулей.

Достоинством принципа больших модулей является возможность использования для организации вычислительной системы не только специализированных, но и *серийных* — „универсальных“ вычислителей, а также *IBM-совместимых PC*.

Несмотря на общий параллелизм вычислений, образованный организацией параллельного функционирования некоторой совокупности вычислителей, в рассмотренном случае *не задействуется* параллелизм вычислений на уровне *каждого* вычислителя.

4.2 Принцип групп модулей

В ЦО сигналов широко используются специализированные вычислители для реализации теоретико-числовых преобразований

и числовой свертки (рис. 4.2). Возьмем за основу этот вычислитель (фрагмент структуры) для организации многопроцессорной вычислительной системы, предназначенной для реализации систем логических функций большой размерности.

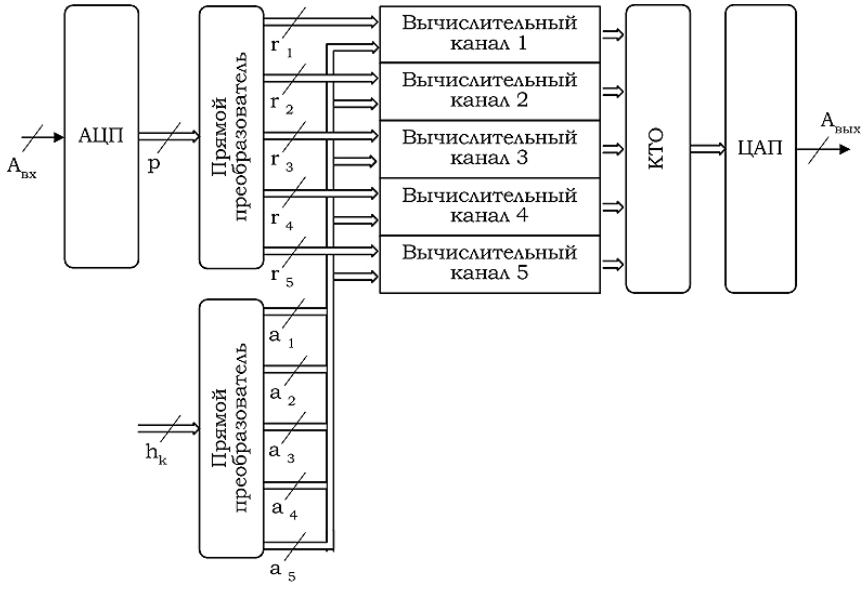


Рис. 4.2: Пример типового специализированного вычислителя цифровой обработки сигналов, функционирующего в пятимодульной арифметике, предназначенного для обработки аналоговых сигналов (АЦП — аналого-цифровой преобразователь, ЦАП — цифроаналоговый преобразователь, КТО — устройство восстановления позиционной формы числа в соответствии с Китайской теоремой об остатках)

4.2.1 Использование полностью параллельных вычислителей

Рассмотрим принцип построения вычислительной системы, на основе специализированных вычислителей, каждый из которых функционирует по системе из s модулей (рис. 4.3). На входы специализированных вычислителей помимо коэффициентов модулярных форм арифметических полиномов подаются результа-

ты вычисления конъюнкций булевых переменных или *непосредственно булевы переменные* в случае использования *линейных форм* арифметических полиномов.

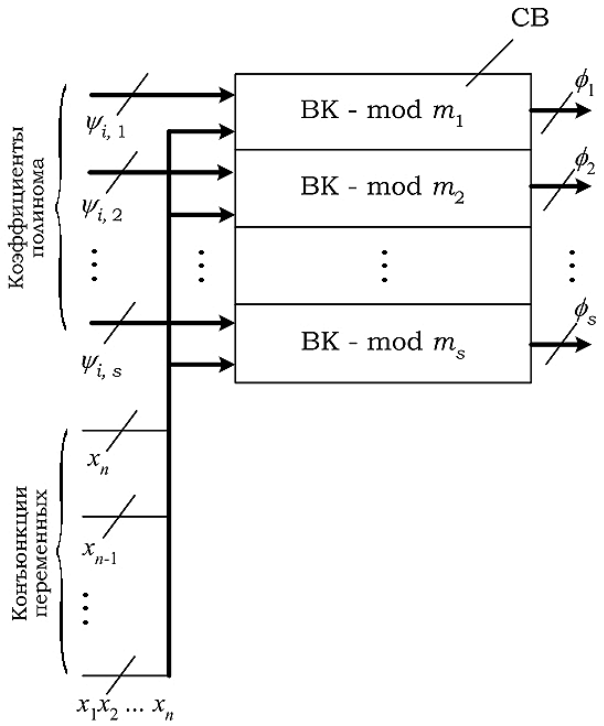


Рис. 4.3: Специализированный вычислитель, функционирующий по системе s модулей $m_1, m_2, \ldots, m_s$ (ВК — вычислительный канал)

Пусть задана система модулей

$$m_1, m_2, \ldots, m_v,$$

свойства которых удовлетворяют теореме 3.1. Поставим им в соответствие вычеты

$$\phi_1, \phi_2, \ldots, \phi_v,$$

определяющие значение искомого результата

$$Y = \mathbf{CRT}_{i=1}^{v} \phi_i \pmod{m_i}.$$

Разобьем систему модулей $m_1, m_2, \ldots, m_v$ на t групп по $s = \frac{v}{t}$ модулей:

$$\overbrace{m_1, m_2, \ldots, m_s}^{1};$$
$$\overbrace{m_{s+1}, m_{s+2}, \ldots, m_{2s}}^{2};$$
$$\vdots$$
$$\overbrace{m_{ts-s}, m_{ts-s+1}, \ldots, m_{st}}^{t}.$$

В соответствие группам модулей сопоставим t групп вычетов

$$\overbrace{\phi_1, \phi_2, \ldots, \phi_s}^{1};$$
$$\overbrace{\phi_{s+1}, \phi_{s+2}, \ldots, \phi_{2s}}^{2};$$
$$\vdots$$
$$\overbrace{\phi_{ts-s}, \phi_{ts-s+1}, \ldots, \phi_{st}}^{t}.$$

Таким образом, вычислительная система может быть построена на основе t специализированных вычислителей, каждый из которых функционирует по группе s модулей (рис. 4.3). Результат вычислений всех специализированных вычислителей преобразуется в позиционную двоичную форму (значения булевых функций) с помощью выходного специализированного вычислителя, ориентированного на реализацию Китайской теоремы об остатках.

Важным свойством данной вычислительной системы является высокий уровень параллелизма, образованный как на уровне организации специализированных вычислителей, так и на уровне функционирования каждого специализированного вычислителя.

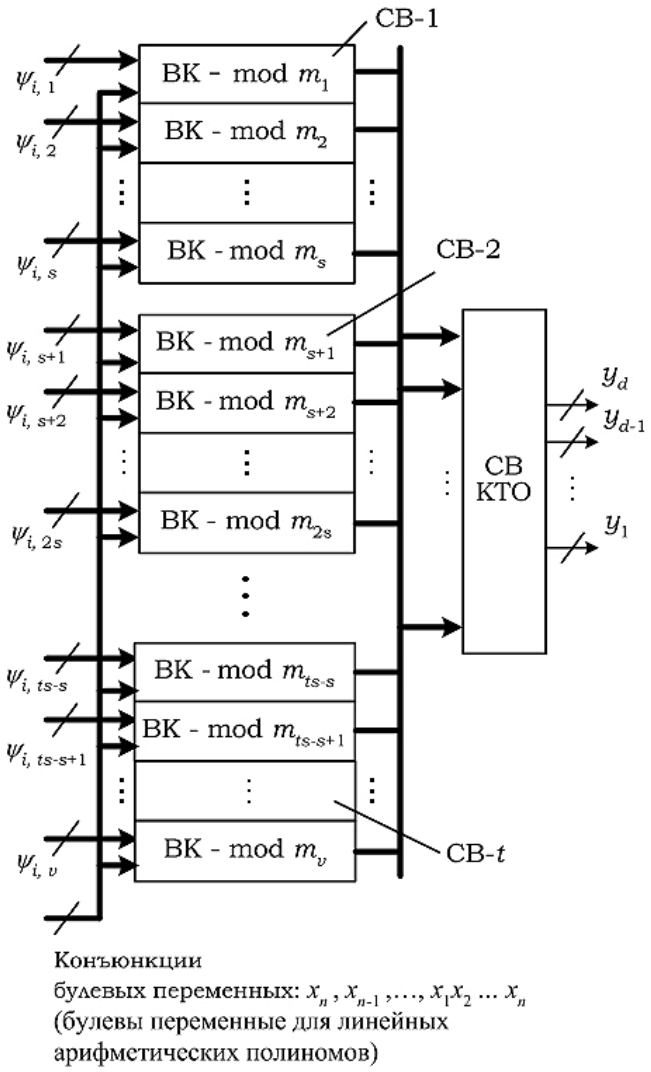


Рис. 4.4: Многопроцессорная система (фрагмент), использующая принцип разбиения системы модулей на группы модулей (СВ — специализированный вычислитель; ВК — вычислительный канал, СВ КТО — специализированный вычислитель, ориентированный на восстановление позиционной формы числа в соответствии с Китайской теоремой об остатках)

4.2.2 Использование промежуточных устройств восстановления числа

Для того чтобы для построения вычислительной системы использовались специализированные вычислители, структура которых бы максимально приближалась к структуре *серийных специализированных вычислителей ЦО сигналов*, функционирующих в модулярной арифметике, выполнение части алгоритма восстановления позиционной формы числа можно распределить среди специализированных вычислителей за счет использования в них встроенных *устройств восстановления позиционной формы числа по группе модулей функционирования* (рис. 4.5). При этом достигается *уменьшение количества связей* в вычислительной системе, а также *сокращение объема оборудования* выходного специализированного вычислителя (СВ КТО), ориентированного на восстановление позиционной формы числа.

Применим к каждой группе вычетов Китайскую теорему об остатках:

$$\begin{cases} \phi_1^* = \mathbf{CRT}_{i=1}^{s}\phi_i \pmod{m_i}, \\ \phi_2^* = \mathbf{CRT}_{i=s+1}^{2s}\phi_i \pmod{m_i}, \\ \vdots \\ \phi_t^* = \mathbf{CRT}_{i=ts-s}^{v}\phi_i \pmod{m_i}. \end{cases} \tag{4.1}$$

При этом отметим, что

$$\phi_1^* = |Y|_{m_1^*}, \quad \phi_2^* = |Y|_{m_2^*}, \quad \ldots, \quad \phi_t^* = |Y|_{m_t^*},$$

где

$$m_1^* = \prod_{i=1}^{s} m_i, \quad m_2^* = \prod_{i=s+1}^{2s} m_i, \quad \ldots, \quad m_t^* = \prod_{i=ts-s}^{v} m_i.$$

Поэтому

$$Y = \mathbf{CRT}_{i=1}^{t}\phi_i^* \pmod{m_i^*}.$$

Таким образом, схема вычисления Y может быть представлена в соответствии с рис. 4.6. Вариант вычислительной системы, построенной в соответствии с рис. 4.6, представлен на рис. 4.7.

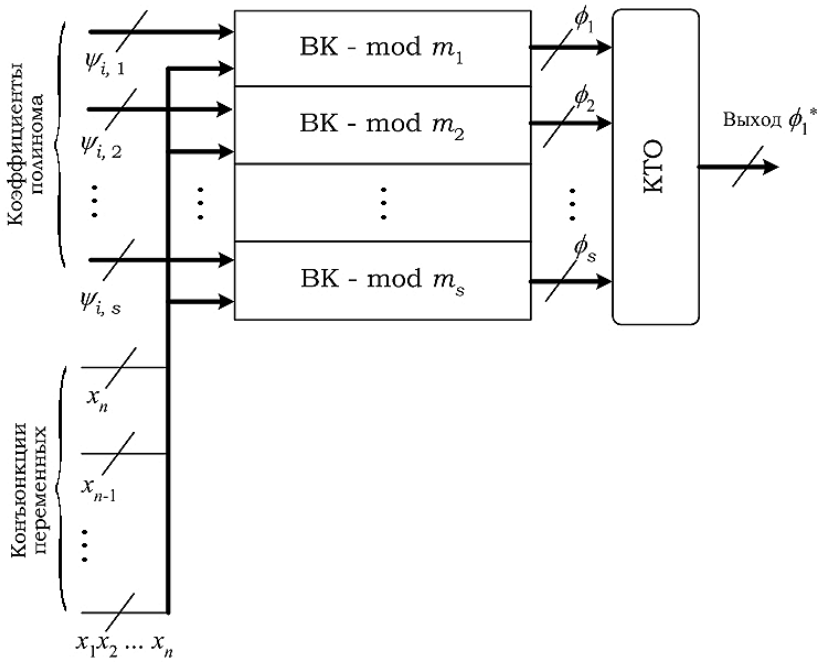


Рис. 4.5: Специализированный вычислитель, функционирующий по системе s модулей m_1, m_2, ..., m_s, содержащий устройство восстановления позиционной формы числа (ВК — вычислительный канал, КТО — устройство для восстановления позиционной формы числа в соответствии с Китайской теоремой об остатках)

Принцип поэтапного восстановления позиционной формы числа в литературе по модулярной арифметике известен. Мы лишь адаптировали этод принцип к условиям нашей задачи.

4.3 Многоступенчатые многомерные формы

4.3.1 Полиномиальные многоступенчатые многомерные формы

Рассмотренный выше принцип организации многопроцессорной вычислительной системы требует использования набора специ-

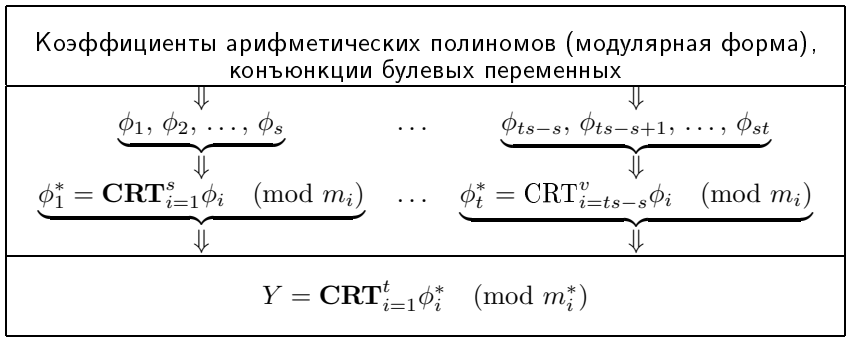


Рис. 4.6: Схема вычисления Y, использующая принцип разбиения системы модулей на группы модулей

ализированных вычислителей, каждый из которых функционирует по „индивидуальному“ набору модулей. Это не всегда удобно. Например, в некоторых случаях требуется повысить однородность оборудования за счет применения набора *одинаковых* специализированных вычислителей.

В предыдущем случае это препятствие возможно обойти путем применения специальных приемов выполнения модулярных операций (по произвольному модулю). Однако такое решение достигается как правило за счет введения дополнительных временных и аппаратурных затрат и сужает область допустимых технических решений. Поэтому применять вычислительные схемы по произвольному модулю целесообразно в случаях, когда имеется некоторый резерв производительности, но требуется обеспечить высокие показатели надежности и живучести вычислительной системы.

В теории модулярной арифметики известен метод сведения вычислений над большими вычетами к вычислениям над маленькими вычетами, предусматривающее вторичное (или многократное) кодирование вычетов, основанное на Китайской теореме об остатках [13]. Этот метод, как правило, предлагается как дополнительное средство для уменьшения объема табличных операционных устройств и не получил широкого распространения.

Однако, в нашем случае этот метод приобретает особую целесообразность в силу необходимости организации вычислений

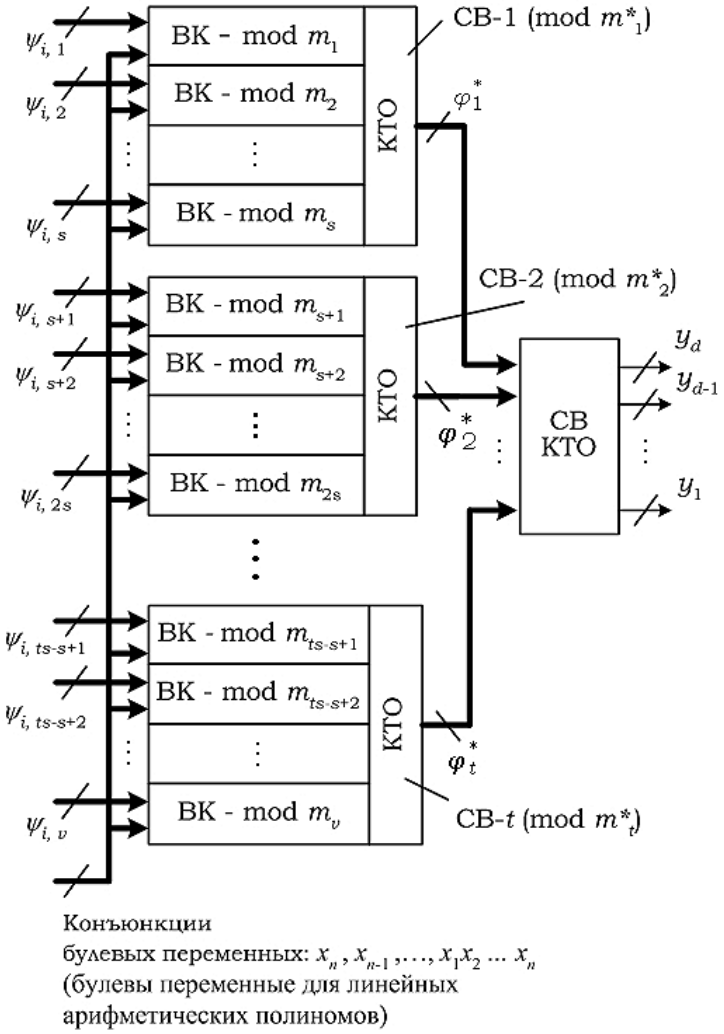


Рис. 4.7: Многопроцессорная система (фрагмент), использующая принцип разбиения системы модулей на группы модулей (СВ — специализированный вычислитель; ВК — вычислительный канал, КТО — устройство для восстановления позиционной формы числа в соответствии с Китайской теоремой об остатках)

в *больших* числовых диапазонах, которые не используются в обычных сферах приложения модулярной арифметки (цифровая обработка сигналов). Более того, как будет показано далее, благодаря отсутствию операций арифметического умножения требования к величине вторичного числового диапазона могут быть существенно снижены (имеется в виду числовой диапазон, достаточный для представления не только конечных, но и промежуточных результатов вычислений, которые при операциях умножения значительно превышают промежуточные результаты, получаемые при сложении).

Принцип многоступенчатого кодирования поясним на примере двухступенчатого кодирования (рис. 4.8).

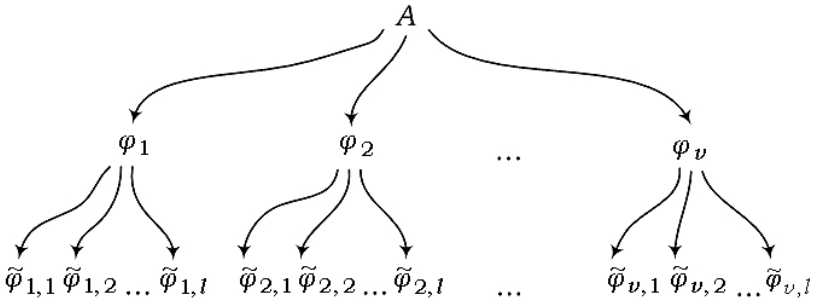


Рис. 4.8: Пояснение принципа двухступенчатого модулярного кодирования числа A, основанного на применении Китайской теоремы об остатках (здесь $\phi_i = |A|_{m_i}$ $(i = 1, 2, \ldots, v)$ — вычеты первой ступени, $\widetilde{\phi}_{j,\,i} = |A|_{\widetilde{m}_j}$ $(j = 1, 2, \ldots, l)$ — вычеты второй ступени модулярного кодирования)

Теорема 4.1
Пусть даны модули $m_1 < m_2 < \ldots < m_v$ *такие, что* $\gcd(m_i, m_j)$ $(i \neq j;\ i, j = 1, \ldots, v)$;

$$\prod_{i=1}^{v} m_i = m \geq 2^d,$$

где d *— количество реализуемых булевых функций, а также даны модули* $\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_l$ *такие, что* $\gcd(\widetilde{m}_i, \widetilde{m}_j)$

$(i \neq j;\ i, j = 1, \ldots, l;)$,

$$\prod_{i=1}^{l} \widetilde{m}_i > 2^n(m_v - 1), \tag{4.2}$$

где n — количество булевых переменных. Тогда произвольный кортеж булевых функций может быть однозначно представлен системой модулярных форм арифметических полиномов:

$$\begin{cases} \widetilde{\phi}_{1,1} = \widetilde{\mu}_{1,1}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{1,i,1}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_1}, \\ \widetilde{\phi}_{1,2} = \widetilde{\mu}_{1,2}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{1,i,2}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_2}, \\ \vdots \\ \widetilde{\phi}_{1,l} = \widetilde{\mu}_{1,l}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{1,i,l}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_l}; \end{cases}$$

$$\begin{cases} \widetilde{\phi}_{2,1} = \widetilde{\mu}_{2,1}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{2,i,1}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_1}, \\ \widetilde{\phi}_{2,2} = \widetilde{\mu}_{2,2}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{2,i,2}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_2}, \\ \vdots \\ \widetilde{\phi}_{2,l} = \widetilde{\mu}_{2,l}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{2,i,l}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_l}; \end{cases} \tag{4.3}$$

$$\vdots$$

$$\begin{cases} \widetilde{\phi}_{v,1} = \widetilde{\mu}_{v,1}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{v,i,1}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_1}, \\ \widetilde{\phi}_{v,2} = \widetilde{\mu}_{v,2}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{v,i,2}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_2}, \\ \vdots \\ \widetilde{\phi}_{v,l} = \widetilde{\mu}_{v,l}(X) = \left| \sum\limits_{i=0}^{2^n-1} \widetilde{\psi}_{v,i,l}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{\widetilde{m}_l}. \end{cases}$$

где $\widetilde{\psi}_{k,i,l} = |\psi_{k,i}|_{m_j}$; $i = 0, 1, \ldots, 2^n - 1$; $k = 1, 2, \ldots, v$;

$j = 1, 2, \ldots, l$; $\psi_{k,i}$ — *коэффициенты полиномиальной арифметической формы, основанной на Китайской теореме об остатках (3.1).*

Доказательство

Пусть дан результат вычисления полинома $\mu_k(X)$ — вычет $0 \le \phi_{k,1} < m_k$. На основании Китайской теоремы об остатках вычет $0 \le \phi_{k,1} < m_k$ может быть представлен системой вычетов $\widetilde{\phi}_{k,1}, \widetilde{\phi}_{k,2}, \ldots, \widetilde{\phi}_{k,l}$ по системе попарно простых модулей $\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_l$. Это представление будет единственным, если

$$\prod_{i=1}^{l} \widetilde{m}_i \ge m_k \quad (k = 1, 2, \ldots, v). \tag{4.4}$$

Тогда так как, согласно теореме 3.1 заданная система булевых функций может быть единственным способом представлена системой вычетов $\phi_1, \phi_2, \ldots, \phi_k$, то система вычетов

$$\begin{array}{rcccc}
\phi_1 = & (\widetilde{\phi}_{1,1}, & \widetilde{\phi}_{1,2}, & \ldots, & \widetilde{\phi}_{1,l}); \\
\phi_2 = & (\widetilde{\phi}_{2,1}, & \widetilde{\phi}_{2,2}, & \ldots, & \widetilde{\phi}_{2,l}); \\
\vdots & \vdots & \vdots & & \vdots \\
\phi_v = & (\widetilde{\phi}_{v,1}, & \widetilde{\phi}_{v,2}, & \ldots, & \widetilde{\phi}_{v,l})
\end{array}$$

также единственным образом соответствует заданной системе булевых функций.

Примем во внимание то, что максимальный результат, получаемый при вычислении полинома

$$\sum_{i=0}^{2^n-1} \psi_{i,k}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},$$

с учетом $0 \le \psi_{i,k} < m_k$ равен

$$2^n(m_k - 1).$$

Но, так как $\max(m_1, m_2, \ldots, m_v) = m_v$, то условие (4.4) должно быть заменено на условие (4.2). Если (4.2) выполняется, то и представление (4.3) единственно.

□

Определение 4.1
Систему арифметических полиномов (4.3) будем называть двухступенчатой полиномиальной формой представления булевых функций, основанной на Китайской теореме об остатках.

Из (4.2) следует, что при использовании нелинейных арифметических полиномов величина числового диапазона, обеспечиваемая второй ступенью модулярного кодирования может быть *существенной*. Уменьшить его можно, если применять Китайскую теорему после более коротких участков цепочки вычислений. Однако это резко снизит производительность вычислений.

„Естественным“ путем уменьшения числового диапазона второй ступени модулярного кодирования является применение идеи многоступенчатого кодирования к *линейным арифметическим полиномам.*

Лемма 4.1
Пусть кортеж булевых функций (1.1) задан линейным арифметическим полиномом $L(X)$ (1.7). Даны модули

$$m_1 < m_2 < \ldots < m_v$$

такие, что $\gcd(m_i, m_j)$ $(i \neq j;\ i, j = 1, \ldots, v)$;

$$\prod_{i=1}^{v} m_i = m > U_{max}.$$

Также даны модули

$$\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_l$$

такие, что $\gcd(\widetilde{m}_i, \widetilde{m}_j)$ $(i \neq j;\ i, j = 1, \ldots, l;)$,

$$\prod_{i=1}^{l} \widetilde{m}_i > (n+1)(m_v - 1), \tag{4.5}$$

где n — количество булевых переменных. Тогда произвольный кортеж булевых функций может быть однозначно представ-

лен системой модулярных форм линейных арифметических полиномов:

$$
\begin{cases}
\widetilde{\phi}_{1,1} = \widetilde{\lambda}_{1,1}(X) = |\widetilde{\omega}_{1,0,1} + \widetilde{\omega}_{1,1,1}x_1 + \ldots + \widetilde{\omega}_{1,n,1}x_n|_{\widetilde{m}_1}, \\
\widetilde{\phi}_{1,2} = \widetilde{\lambda}_{1,2}(X) = |\widetilde{\omega}_{1,0,2} + \widetilde{\omega}_{1,1,2}x_1 + \ldots + \widetilde{\omega}_{1,n,2}x_n|_{\widetilde{m}_2}, \\
\vdots \\
\widetilde{\phi}_{1,l} = \widetilde{\lambda}_{1,l}(X) = |\widetilde{\omega}_{1,0,l} + \widetilde{\omega}_{1,1,l}x_1 + \ldots + \widetilde{\omega}_{1,n,l}x_n|_{\widetilde{m}_l};
\end{cases}
$$

$$
\begin{cases}
\widetilde{\phi}_{2,1} = \widetilde{\lambda}_{2,1}(X) = |\widetilde{\omega}_{2,0,1} + \widetilde{\omega}_{2,1,1}x_1 + \ldots + \widetilde{\omega}_{2,n,1}x_n|_{\widetilde{m}_1}, \\
\widetilde{\phi}_{2,2} = \widetilde{\lambda}_{2,2}(X) = |\widetilde{\omega}_{2,0,2} + \widetilde{\omega}_{2,1,2}x_1 + \ldots + \widetilde{\omega}_{2,n,2}x_n|_{\widetilde{m}_2}, \\
\vdots \\
\widetilde{\phi}_{2,l} = \widetilde{\lambda}_{2,l}(X) = |\widetilde{\omega}_{2,0,l} + \widetilde{\omega}_{2,1,l}x_1 + \ldots + \widetilde{\omega}_{2,n,l}x_n|_{\widetilde{m}_l};
\end{cases}
\qquad (4.6)
$$

$$\vdots$$

$$
\begin{cases}
\widetilde{\phi}_{v,1} = \widetilde{\lambda}_{v,1}(X) = |\widetilde{\omega}_{v,0,1} + \widetilde{\omega}_{v,1,1}x_1 + \ldots + \widetilde{\omega}_{v,n,1}x_n|_{\widetilde{m}_1}, \\
\widetilde{\phi}_{v,2} = \widetilde{\lambda}_{v,2}(X) = |\widetilde{\omega}_{v,0,2} + \widetilde{\omega}_{v,1,2}x_1 + \ldots + \widetilde{\omega}_{v,n,2}x_n|_{\widetilde{m}_2}, \\
\vdots \\
\widetilde{\phi}_{v,l} = \widetilde{\lambda}_{v,l}(X) = |\widetilde{\omega}_{v,0,l} + \widetilde{\omega}_{v,1,l}x_1 + \ldots + \widetilde{\omega}_{v,n,l}x_n|_{\widetilde{m}_l}.
\end{cases}
$$

где $\widetilde{\omega}_{k,i,l} = |\omega_{k,i}|_{m_j}$*;*
$i = 0, 1, \ldots, 2^n - 1$; $k = 1, 2, \ldots, v$; $j = 1, 2, \ldots, l$; $\omega_{k,i}$ *— коэффициенты линейной полиномиальной арифметической формы, основанной на Китайской теореме об остатках (3.6).*

Доказательство
Если условие (4.4) выполнено, то на основании справедливости леммы 3.1 система вычетов

$$
\begin{array}{rcccc}
\phi_1 = & (\widetilde{\phi}_{1,1}, & \widetilde{\phi}_{1,2}, & \ldots, & \widetilde{\phi}_{1,l}); \\
\phi_2 = & (\widetilde{\phi}_{2,1}, & \widetilde{\phi}_{2,2}, & \ldots, & \widetilde{\phi}_{2,l}); \\
\vdots & \vdots & \vdots & & \vdots \\
\phi_v = & (\widetilde{\phi}_{v,1}, & \widetilde{\phi}_{v,2}, & \ldots, & \widetilde{\phi}_{v,l})
\end{array}
$$

единственна. Максимальный результат, получаемый при вычислении линейного полинома

$$\widetilde{\omega}_{0,k} + \widetilde{\omega}_{1,k}x_1 + \ldots + \widetilde{\omega}_{n,k}x_n$$

равен

$$(n+1)(m_k-1).$$

Так как условие (4.5) оговорено, то лемма 4.1 справедлива.

□

Определение 4.2
Систему арифметических полиномов (4.6) будем называть двухступенчатой линейной полиномиальной формой представления булевых функций, основанной на Китайской теореме об остатках.

Следствие 4.1

$$Y = \mathbf{CRT}_{i=1}^{v}\phi_i \pmod{m_i},$$

где

$$\phi_1 = \left|\widetilde{\phi}_1\right|_{m_1}, \quad \phi_2 = \left|\widetilde{\phi}_2\right|_{m_2}, \quad \ldots, \quad \phi_v = \left|\widetilde{\phi}_v\right|_{m_v};$$

$$\begin{aligned}
\widetilde{\phi}_1 &= \mathbf{CRT}_{i=1}^{l}\widetilde{\phi}_{1,\,i} \pmod{\widetilde{m}_i},\\
\widetilde{\phi}_2 &= \mathbf{CRT}_{i=1}^{l}\widetilde{\phi}_{2,\,i} \pmod{\widetilde{m}_i},\\
&\vdots\\
\widetilde{\phi}_v &= \mathbf{CRT}_{i=1}^{l}\widetilde{\phi}_{v,\,i} \pmod{\widetilde{m}_i}.
\end{aligned}$$

Следствие 4.2

$$Y = \left|\widetilde{\phi}_1 B_1 + \widetilde{\phi}_2 B_2 + \ldots + \widetilde{\phi}_v B_v\right|_m,$$

где B_k $(k = 1, 2, \ldots v)$ — ортогональные базисы из выражения (3.7).

Следствие 4.3
Из (4.2) (4.5) следует, что результат вычисления модулярной формы линейного полинома второй ступени (4.6) в

$$\frac{2^n}{n+1}$$

раз меньше по сравнению с результатом вычисления модулярной формы полинома второй ступени (4.3). Поэтому применение модулярной формы линейного полинома более целесообразно.

Замечание 4.1
Использование в теореме 4.3.2 и лемме 4.1 единой системы модулей $\widetilde{m}_1, \widetilde{m}_2, \ldots, \widetilde{m}_l$ позволяет при технической реализации использовать комплект из одинаковых специализированных вычислителей.

4.3.2 Конъюнктивные многоступенчатые многомерные формы

Существование полиномиальных модулярных многоступенчатых форм дает основание для построения многоступенчатых теоретико-числовых преобразований.

Лемма 4.2
Для d-выходной булевой функции $f(X)$ справедлива пара матричных преобразований:

$$
\begin{cases}
\begin{cases}
\widetilde{\mathbf{\Psi}}_{1,1} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Psi}}_{1,2} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Psi}}_{1,l} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,l} \pmod{\widetilde{m}_l};
\end{cases}\\
\begin{cases}
\widetilde{\mathbf{\Psi}}_{1,1} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Psi}}_{1,2} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Psi}}_{1,l} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,l} \pmod{\widetilde{m}_l};
\end{cases}\\
\vdots\\
\begin{cases}
\widetilde{\mathbf{\Psi}}_{1,1} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Psi}}_{1,2} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Psi}}_{1,l} = \mathbf{A}_{2^n}\widetilde{\mathbf{\Phi}}_{1,l} \pmod{\widetilde{m}_l}
\end{cases}
\end{cases}
\tag{4.7}
$$

(прямое преобразование);

$$
\begin{array}{l}
\left\{\begin{array}{l}
\widetilde{\mathbf{\Phi}}_{1,1} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Phi}}_{1,2} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Phi}}_{1,l} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,l} \pmod{\widetilde{m}_l};
\end{array}\right.\\
\left\{\begin{array}{l}
\widetilde{\mathbf{\Phi}}_{1,1} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Phi}}_{1,2} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Phi}}_{1,l} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,l} \pmod{\widetilde{m}_l};
\end{array}\right.\\
\vdots\\
\left\{\begin{array}{l}
\widetilde{\mathbf{\Phi}}_{1,1} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,1} \pmod{\widetilde{m}_1},\\
\widetilde{\mathbf{\Phi}}_{1,2} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,2} \pmod{\widetilde{m}_2},\\
\vdots\\
\widetilde{\mathbf{\Phi}}_{1,l} = \mathbf{A}_{2^n}^{-1}\widetilde{\mathbf{\Psi}}_{1,l} \pmod{\widetilde{m}_l}
\end{array}\right.
\end{array}
\tag{4.8}
$$

(обратное преобразование),
где $\mathbf{A}_{2^n}$ *и* $\mathbf{A}_{2^n}^{-1}$ *— соответственно матрицы прямого и инверсного арифметического преобразования;*

$$
\begin{aligned}
\widetilde{\mathbf{\Phi}}_{k,j} &= \left[\widetilde{\phi}_{0,k,j},\, \widetilde{\phi}_{1,k,j},\, \ldots,\, \widetilde{\phi}_{2^n-1,k,j}\right]^T;\\
\widetilde{\mathbf{\Psi}}_{k} &= \left[\widetilde{\psi}_{k,0,j},\, \widetilde{\psi}_{k,1,j},\, \ldots,\, \widetilde{\psi}_{k,2^n-1,j}\right]^T;\\
\widetilde{\phi}_{i,k,j} &= \left|\phi_k^{(i)}\right|_{\widetilde{m}_j};\\
\phi_k^{(i)} &= |Y^{(i)}|_{m_k};\\
i &= 0,\, 1,\, \ldots,\, 2^n-1;\\
k &= 1,\, \ldots,\, v;\\
j &= 1,\, \ldots,\, l.
\end{aligned}
$$

С п р а в е д л и в о с т ь леммы и (4.7) и (4.8) следует 1) из взаимооднозначности связи матричной (1.10) и (1.11) и полиномиальной (1.3) форм представления булевых функций [71]

и 2) из доказательства справедливости полиномиальной формы представления (4.3).

□

Определение 4.3
Системы матричных преобразований (4.7) и (4.8) будем называть конъюнктивными двухступенчатыми многомерными преобразованиями, основанными на Китайской теореме об остатках или двухступенчатыми ЛТЧП, основанными на Китайской теореме об остатках (двухступенчатыми ЛТЧП КТО).

Пример вычислительной системы, основанной на использовании теоремы и леммы 4.1, представлен на рис. 4.9

Таким образом, метод применения многоступенчатых многомерных арифметико-логических форм позволяет значительно повысить однородность вычислительной системы (то есть строить ее на одинаковых специализированных вычислителях).

4.4 Выводы

- Рассмотрены принципы построения *многопроцессорных вычислительных систем*, обеспечивающих интенсивную реализацию логических функций большой размерности в *больших числовых диапазонах.*
- Принцип *больших модулей* обеспечивает *доступность* технической реализации на основе серийных универсальных и специализированных микропроцессоров и 32—64-х разрядных IBM-совместимых PC.
- Принцип *групп модулей* позволяет задействовать для построения параллельной вычислительной системы специализированные параллельные вычислители, идентичные специализированным вычислителям, применяемым в ЦО сигналов для реализации ТЧП. При этом достигается *высокий уровень параллелизма*, обеспечиваемый параллельной организацией параллельных специализированных вычислителей.

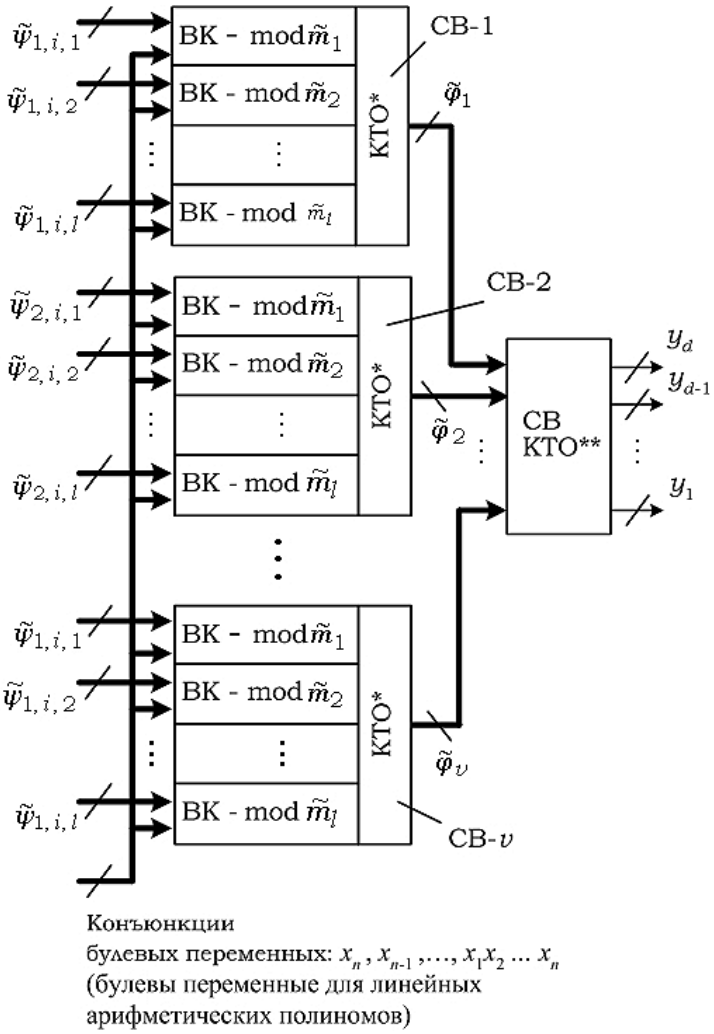


Рис. 4.9: Многопроцессорная система (фрагмент), использующая двухступенчатые многомерные формы, основанные на Китайской теореме об остатках (СВ — специализированный вычислитель; ВК — вычислительный канал, КТО — устройство для восстановления позиционной формы числа в соответствии с Китайской теоремой об остатках, здесь звездочка $*$ означает, что это устройство может быть наделено функцией приведения результата преобразования $\widetilde{\phi}_i$ по модулю m_i; две звездочки $**$ означают, что выходной преобразователь может учитывать не применение вышеуказанной функции приведения по модулю)

- Предложены *полиномиальные* и *конъюнктивные* многомерные многоступенчатые арифметико-логические формы. Они реализуют принцип больших модулей с обеспечением параллельного характера вычислений „внутри“ вычетов. При этом как и при использовании принципа групп модулей для построения многопроцессорной системы применяются параллельные специализированные вычислители, обеспечивая, тем самым, чрезвычайно высокий уровень параллелизма. Однако в отличие от принципа групп модулей достигается более высокая однородность оборудования за счет применения идентичных специализированных вычислителей.

Глава 5

Обобщение модулярных форм на k-значную логику

Глава[1] посвящена обобщению арифметической логики на формы и функции многозначной логики. Общая схема погружения функции алгебры логики в полиномиальное кольцо сохраняется. Однако при этом возникает специфика, которая состоит в том, что коэффициенты целозначных полиномов теперь могут быть рациональными числами. Как будет показано далее этот факт необходимо учитывать при выборе модулей модулярного представления форм и функций многозначной алгебры логики.

5.1 Полиномиальная арифметика k-значной логики

Под многозначной $f^{(k)}(X)$ функцией алгебры логики (ФАЛ) n переменных $X = x_1, x_2, \ldots, x_n$ будем понимать логическую функцию, заданную на множестве $\{0, 1, \ldots, k-1\}$, значения аргументов которой принадлежат этому же множеству, где k — значность ФАЛ.

Число всевозможных ФАЛ равно k^{k^n}.

Как известно, любую многозначную ФАЛ можно представить в виде арифметического полинома [58]:

$$P^{(k)}(X) = \sum_{i=0}^{k^n-1} p_i^{(k)} \, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \tag{5.1}$$

[1] Материалы данной главы впервые представлены к опубликованию в [129, 130].

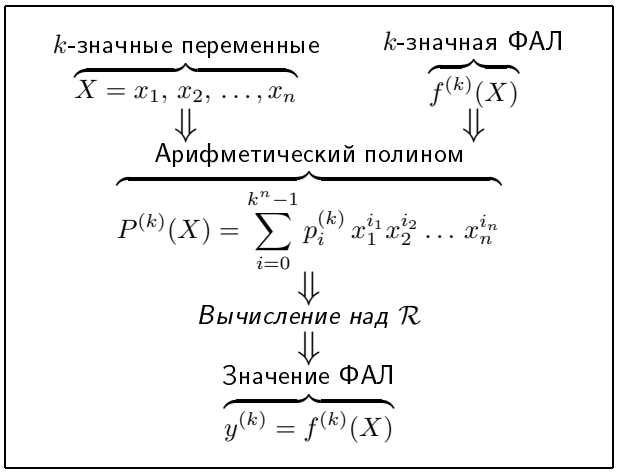


Рис. 5.1: Схема вычисления произвольной k-значной ФАЛ над $\mathcal{R}$

где $p_i^{(k)}$ — коэффициенты АП, такие, что $p_i^{(k)} \in \mathcal{R}$ ($\mathcal{R}$ — множество рациональных чисел); $X = x_1, x_2, \ldots, x_n$ — аргументы ФАЛ $x_u \in \{0, 1, \ldots, k-1\}$ $(u = 1, 2, \ldots, n)$; $(i_1\, i_2\, \ldots\, i_n)_k$ — представление параметра i в k-ичной системе счисления:

$$
\begin{aligned}
(i_1\, i_2\, \ldots\, i_n)_k &= \sum_{u=1}^{n} i_u k^{n-u} \ (i_u \in \{0, 1, \ldots, k-1\}); \\
x_u^{i_u} &= \begin{cases} x_u, & i_u \neq 0, \\ 1, & i_u = 0. \end{cases}
\end{aligned}
$$

Так же как и в случае вычисления булевых функций для k-значных ФАЛ можно построить две схемы вычисления посредством арифметических полиномов (рис. 5.1 и 5.2). В отличие от схем, представленных на рис. 1.1 и 1.2, вычисление производится над множеством рациональных чисел $\mathcal{R}$.

Аналогично двоичной логике в k-ичной логике могут быть определены алгебраический и матричный методы построения арифметического полинома (5.1) [58].

Прямое и обратное матричное преобразование (ЛДПФ — в дальнейшем k-ЛДПФ) определяется соответственно выражени-

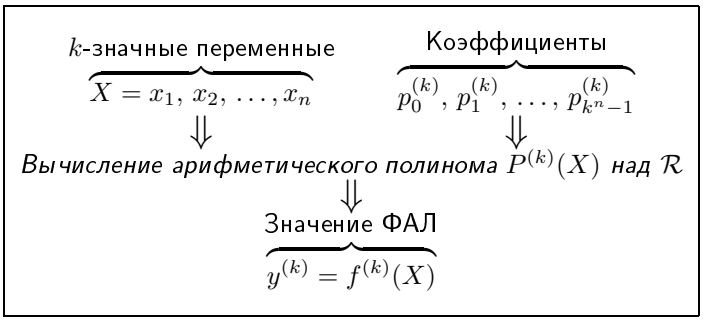


Рис. 5.2: Схема вычисления заданной k-значной ФАЛ над $\mathcal{R}$

ями

$$\mathbf{P}^{(k)} = N_k^{-1}\mathbf{K}_{k^n}\mathbf{S}, \tag{5.2}$$

$$\mathbf{S} = \mathbf{K}_{k^n}^{-1}\mathbf{P}^{(k)}, \tag{5.3}$$

где N_k — нормализующий множитель; $\mathbf{K}_{k^n}$ и $\mathbf{K}_{k^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования размерности $k^n \times k^n$ (базис преобразования); $\mathbf{S}$ — вектор истинности k-значной ФАЛ;

$$\mathbf{S} = \left[S^{(0)}S^{(1)}\ldots S^{(k^n-1)}\right]^T,$$

где $S^{(i)}$ — числовое значение, принимаемое k-значной ФАЛ на i-м наборе переменных; вектор коэффициентов (спектр) арифметического полинома (5.1):

$$\mathbf{P}^{(k)} = \left[p_0^{(k)}p_1^{(k)}\ldots p_{k^n-1}^{(k)}\right].$$

Матрицы $\mathbf{K}_{k^n}$ и $\mathbf{K}_{k^n}^{-1}$ как и базис преобразования двоичной логики определяется кронекеровским возведением в степень:

$$\mathbf{K}_{k^n} = \bigotimes_{j=1}^{n}\mathbf{K}_k; \quad \mathbf{K}_{k^n}^{-1} = \bigotimes_{j=1}^{n}\mathbf{K}_k^{-1},$$

где $\mathbf{K}_k$ и $\mathbf{K}_k^{-1}$ — базовые матрицы прямого и обратного преобразования (см. табл. 5.1 — для $k = 2\ldots6$).

Таблица 5.1:

k	$\mathbf{K}_k$	$\mathbf{K}_k^{-1}$
2	$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
3	$\begin{bmatrix} 2 & 0 & 0 \\ -3 & 4 & -1 \\ 1 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$
4	$\begin{bmatrix} 6 & 0 & 0 & 0 \\ -11 & 18 & -9 & 2 \\ 6 & -15 & 12 & -3 \\ -1 & 3 & -3 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix}$
5	$\begin{bmatrix} 24 & 0 & 0 & 0 & 0 \\ -50 & 96 & -72 & 32 & -6 \\ 35 & -104 & 114 & -56 & 11 \\ -10 & 36 & -48 & 28 & -6 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}$
6	$\begin{bmatrix} 120 & 0 & 0 & 0 & 0 & 0 \\ -274 & 660 & -600 & 400 & -150 & 24 \\ 225 & -770 & 1070 & -780 & 305 & -20 \\ -85 & 355 & -590 & 490 & -205 & 35 \\ 15 & -70 & 130 & -120 & 55 & -10 \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 \\ 1 & 3 & 9 & 27 & 81 & 243 \\ 1 & 4 & 16 & 64 & 256 & 1024 \\ 1 & 5 & 25 & 125 & 625 & 3125 \end{bmatrix}$

Пример 5.1
Пусть $k = 3$ и вектор принимаемых значений ФАЛ при $n = 2$ имеет вид: $\mathbf{S} = [2\ 0\ 2\ 1\ 1\ 1\ 0\ 1\ 2]^T$. Тогда прямое преобразование (5.6) примет вид:

$$\mathbf{P}^{(3)} = \frac{1}{4}\mathbf{K}_{3^2}\mathbf{S} =$$

$$= \frac{1}{4}\begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 8 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 0 & 8 & 0 & 0 & -2 & 0 & 0 \\ 9 & -12 & 3 & -12 & 16 & -4 & 3 & -4 & 1 \\ -3 & 6 & -3 & 4 & -8 & 4 & -1 & 2 & -1 \\ 2 & 0 & 0 & -4 & 0 & 0 & 2 & 0 & 0 \\ -3 & 4 & -1 & 6 & -8 & 2 & -3 & 4 & -1 \\ 1 & -2 & 1 & -2 & 4 & -2 & 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} =$$

$$= \frac{1}{4}\begin{bmatrix} 8 \\ -16 \\ 8 \\ -4 \\ 22 \\ -12 \\ 0 \\ -6 \\ 4 \end{bmatrix} \begin{matrix} x_2 \\ x_2^2 \\ x_1 \\ x_1x_2 \\ x_1x_2^2 \\ x_1^2 \\ x_1^2x_2 \\ x_1^2x_2^2 \end{matrix}.$$

Таким образом на основании (5.1) алгебраическая форма будет иметь вид:

$$\begin{aligned} P^{(3)}(X) &= \frac{1}{4}(8 - 16x_2 + 8x_2^2 - 4x_1 + 22x_1x_2 - 12x_1x_2^2 + \\ &+ 0x_1^2 - 6x_1^2x_2 + 4x_1^2x_2^2) = \\ &= 2 - 4x_2 + 2x_2^2 - x_1 + \frac{11}{2}x_1x_2 - 3x_1x_2^2 - \\ &- \frac{3}{2}x_1^2x_2 + x_1^2x_2^2. \end{aligned}$$

Например, пусть $x_1 = 1$, $x_2 = 2$, тогда значение ФАЛ определится как:

$$\begin{aligned} P^{(3)}(X) &= \frac{1}{4}(8 - 16 \cdot 2 + 8 \cdot 2^2 - 4 \cdot 1 + 22 \cdot 1 \cdot 2 - 12 \cdot 1 \cdot 2^2 - \\ &- 6 \cdot 1^2 \cdot 2 + 4 \cdot 1^2 \cdot 2^2) = \frac{1}{4}4 = 1. \end{aligned}$$

Обратное преобразование (5.7) будет иметь вид:

$$\mathbf{S} = \mathbf{K}_{3^2}^{-1}\mathbf{P}^{(3)} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 \\ 1 & 2 & 4 & 2 & 4 & 8 & 4 & 8 & 16 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ -4 \\ 2 \\ -1 \\ \frac{11}{2} \\ -3 \\ 0 \\ -\frac{3}{2} \\ 1 \end{bmatrix} \begin{matrix} x_2 \\ x_2^2 \\ x_1 \\ x_1x_2 \\ x_1x_2^2 \\ x_1^2 \\ x_1^2x_2 \\ x_1^2x_2^2 \end{matrix} =$$

$$= \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}.$$

Недостатком представления (5.1) является возможность принятия коэффициентами $p_i^{(k)}$ $(i = 0, 1, \ldots, k^n - 1)$ как неотрицательных, так и отрицательных значений, что требует удваивания числового диапазона по сравнению с использованием неотрицательных коэффициентов, а их абсолютные значения и, тем более абсолютные значения промежуточных результатов вычисления (5.1), могут многократно превышать значение k.

Известно представление многозначной ФАЛ с помощью логического полинома [4]:

$$F^{(k)}(X) = \left| \sum_{i=0}^{k^n-1} f_i^{(k)} \, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_k, \tag{5.4}$$

где $f_i^{(k)}$ — коэффициенты, такие что $f_i^{(k)} \in \{0, 1, \ldots, k-1\}$;

$$x_u^{i_u} = \begin{cases} x_u, \; i_u \neq 0, \\ 1, \; i_u = 0. \end{cases}$$

Недостатком (5.4) является необходимость выполнения вычислений по „жестко“ заданному модулю k (k — *простое число*), обязательно совпадающему со значностью ФАЛ, что не обеспечивает требуемую свободу выбора средств вычисления ФАЛ.

5.2 Модулярные формы k-значной логики

Теорема 5.1
Пусть $m \geq k$, где k — значность логики и m — простое, тогда произвольная ФАЛ может быть представлена арифметическим полиномом:

$$\mu^{(k)}(X) = \left| \sum_{i=0}^{k^n-1} \rho_i \, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_m, \tag{5.5}$$

где $\rho_i, h_u^{i_u} \in \mathcal{Z}_m$;

$$x_u^{i_u} = \begin{cases} x_u, & i_u \neq 0, \\ 1, & i_u = 0. \end{cases}$$

Доказательство теоремы 5.1 строится аналогично доказательству теоремы 2.1.

Замечание 5.1
Арифметические полиномы (5.1) и (5.5) связаны отношением $\rho_i = |p_i^{(k)}|_m$.

Определение 5.1
Выражение (5.5) будем называть представлением ФАЛ k-значной логики на основе модулярной формы арифметического полинома.

Примитивный алгоритм (рис. 5.3) получения (5.5), который состоит из двух шагов и заключается в 1) вычислении арифметического полинома (5.1) и 2) определении арифметического полинома (5.5) на основании замечания 5.1.

Шаг 1	$p_0^{(k)}$	$p_1^{(k)}$	$\dots$	$p_{k^n-1}^{(k)}$
	$\Downarrow$	$\Downarrow$		$\Downarrow$
Шаг 2	$\rho_0 = \lvert p_0^{(k)}\rvert_m$	$\rho_1 = \lvert p_1^{(k)}\rvert_m$	$\dots$	$\rho_{k^n-1} = \lvert p_{k^n-1}^{(k)}\rvert_m$

Рис. 5.3: Пояснения к варианту построения алгоритма получения (5.5) на основании замечания 5.1

К (5.5) справедливы следствия 2.1 и 2.2.

Основные схемы вычисления k-значной ФАЛ посредством модулярной формы арифметического полинома представлены на рис. 5.4 и 5.5, из которых следует, что вычисления из поля рациональных чисел $\mathcal{R}$ перенесены в простое поле $PG(m)$ целых чисел.

Полученный результат можно наглядно продемонстрировать на следующих примерах.

Рис. 5.4: Схема вычисления произвольной k-значной ФАЛ над $PG(m)$

Пример 5.2
Выберем значения $k = 2, 3, 4$ *и* $n = 2$. *Определим модуль* $m = 5$, *величина которого удовлетворяет всем выбранным значениям* k *(то есть* $m \geq k$*). Используем полиномы (5.1) и (5.5) для представления следующих элементарных ФАЛ:*

$\overline{x} = (k-1) - x$ — *инверсия;*
$\hat{x} = |x+1|_k$ — *циклическое отрицание (*$\overline{x} = x \oplus 1$, *при* $k = 2$*);*
$x_1 \vee x_2 = \max(x_1, x_2)$ — *дизъюнкция;*
$x_1 \wedge x_2 = \min(x_1, x_2)$ — *конъюнкция;*
$|x_1 + x_2|_k$ — *сложение по модулю* k *(*$x_1 \oplus x_2$, *при* $k = 2$*);*
$|x_1 x_2|_k$ — *умножение по модулю* k *(*$x_1 \wedge x_2$, *при* $k = 2$*);*
$x_1 \uparrow x_2 = |(x_1 \vee x_2) + 1|_k$ — *функция Вебба (*$\overline{x_1 \vee x_2} = (x_1 \vee x_2) \oplus 1$ — *„стрелка Пирса“, при* $k = 2$*);*
$x_1 \downarrow x_2 = \overline{x_1 \wedge x_2}$ — *„штрих Шеффера“, при* $k = 2$.

Результаты использования (5.1) и (5.5) для $k = 2, 3, 4$ *представлены соответственно в табл. 5.2—5.4.*

Для пояснения принципа преобразования представления ФАЛ из (5.1) в (5.5) рассмотрим следующий пример.

Рис. 5.5: Схема вычисления заданной k-значной ФАЛ над $PG(m)$

Таблица 5.2:

$f(\mathbf{X})$	$P(\mathbf{X})$	$\mu(\mathbf{X})$ при $m = 5$
$\overline{x}_i$	$1 - x_i$	$\lvert 1 + 4x_i \rvert_5$
$x_1 \wedge x_2$	$x_1 x_2$	$\lvert x_1 x_2 \rvert_5$
$x_1 \vee x_2$	$x_1 + x_2 - x_1 x_2$	$\lvert x_1 + x_2 + 4x_1 x_2 \rvert_5$
$x_1 \oplus x_2$	$x_1 + x_2 - 2x_1 x_2$	$\lvert x_1 + x_2 + 3x_1 x_2 \rvert_5$
$x_1 \downarrow x_2$	$1 - x_1 x_2$	$\lvert 1 + 4x_1 x_2 \rvert_5$
$x_1 \uparrow x_2$	$1 - x_1 - x_2 + x_1 x_2$	$\lvert 1 + 4x_1 + 4x_2 + x_1 x_2 \rvert_5$

Пример 5.3
При $k = 3$ получить модулярную форму арифметического полинома (5.5) на основе данного арифметического полинома вида (5.1) ФАЛ $|x_1 + x_2|_k$ для произвольного m и $m = 5$ (табл. 5.3):

$$
\begin{aligned}
& |x_1 + x_2|_3 = \\
= \; & \Big| x_1 + x_2 + |21|4^{\varphi(m)-1}|_m|_m x_1 x_2 + \left(m - |15|4^{\varphi(m)-1}|_m|_m\right) x_1 x_2^2 + \\
& + \left(m - |15|4^{\varphi(m)-1}|_m|_m\right) x_1^2 x_2 + |9|4^{\varphi(m)-1}|_m|_m x_1^2 x_2^2 \Big|_m = \\
= \; & \left| x_1 + x_2 + 4x_1 x_2 + x_1^2 x_2^2 \right|_5 .
\end{aligned}
$$

где $\varphi(m)$ — функция Эйлера [25].

Таблица 5.3:

$f(\mathbf{X})$	$P(\mathbf{X})$	$\mu(\mathbf{X})$ при $m=5$
$\overline{x}_i$	$2-x_i$	$\lvert 2+4x_i\rvert_5$
$\hat{x}_i$	$1+5x_i/2-3x_i^2/2$	$\left\lvert 1+x_i^2\right\rvert_5$
$x_1 \wedge x_2$	$5x_1x_2/2-x_1x_2^2-x_1^2x_2+x_1^2x_2^2/2$	$\left\lvert 4x_1x_2^2+4x_1^2x_2+3x_1^2x_2^2\right\rvert_5$
$\lvert x_1+x_2\rvert_3$	$x_1+x_2+21x_1x_2/4-15x_1x_2^2/4-$ $-15x_1^2x_2/4+9x_1^2x_2^2/4$	$\left\lvert x_1+x_2+4x_1x_2+x_1^2x_2^2\right\rvert_5$
$\lvert x_1x_2\rvert_3$	$x_1x_2/4+3x_1x_2^2/4+$ $+3x_1^2x_2/4-3x_1^2x_2^2/4$	$\left\lvert 4x_1x_2+2x_1x_2^2+\right.$ $\left.+2x_1^2x_2+3x_1^2x_2^2\right\rvert_5$
$x_1 \uparrow x_2$	$1+5x_2/2-3x_2^2/2+5x_1/2-$ $-7x_1x_2/4+x_1x_2^2/4-3x_1^2/2+$ $+x_1^2x_2/4+x_1^2x_2^2/4$	$\left\lvert 1+x_2^2+2x_1x_2+4x_1x_2^2+\right.$ $\left.+x_1^2+4x_1^2x_2+4x_1^2x_2^2\right\rvert_5$

Пример 5.4
При $k=3$ получить модулярную форму арифметического полинома из примера 5.2 для произвольного m и $m=5$:

$$\begin{aligned}
\mu^{(3)}(X) &= \Big|2-4x_2+2x_2^2-x_1+\frac{11}{2}x_1x_2-3x_1x_2^2- \\
&\quad -\frac{3}{2}x_1^2x_2+x_1^2x_2^2\Big|_m = \\
&= \left|2+(m-|4|_m)\,x_2+2x_2^2+(m-1)x_1+\right. \\
&\quad +|11|_m 2^{\varphi(m)-1}x_1x_2+ \\
&\quad \left.+(m-3)x_1x_2^2+(m-3)2^{\varphi(m)-1}x_1^2x_2+x_1^2x_2^2\right|_m = \\
&= \left|2+x_2+2x_2^2+4x_1+3x_1x_2+2x_1x_2^2+x_1^2x_2+x_1^2x_2^2\right|_5 .
\end{aligned}$$

Таблица 5.4:

$f(\mathbf{X})$	$P(\mathbf{X})$	$\mu(\mathbf{X})$ при $m=5$
$\overline{x}_i$	$3-x_i$	$\lvert 3+4x_i\rvert_5$
$\hat{x}_i$	$1-x_i/3+2x_i^2-2x_i^3/3$	$\lvert 1+3x_i+2x_i^2+x_i^3\rvert_5$
$x_1 \wedge x_2$	$29x_1x_2/6-15x_1x_2^2/4-$ $-15x_1^2x_2/4+3x_1x_2^3/4+$ $+3x_1^3x_2/4+7x_1^2x_2^2/2-$ $-3x_1^2x_2^3/4-3x_1^3x_2^2/4+x_1^3x_2^3/6$	$\lvert 4x_1x_2+2x_1x_2^3+$ $+2x_1^3x_2+x_1^2x_2^2+$ $+3x_1^2x_2^3+3x_1^3x_2^2+x_1^3x_2^3\rvert_5$
$\lvert x_1+x_2\rvert_4$	$x_1+x_2-121x_1x_2/9+$ $+49x_1x_2^2/3+49x_1^2x_2/3-$ $-38x_1^3x_2/9-38x_1x_2^3/9-$ $-19x_1^2x_2^2+14x_1^2x_2^3/3+$ $+14x_1^3x_2^2/3-10x_1^3x_2^3/9$	$\lvert x_1+x_2+x_1x_2+3x_1x_2^2+$ $+3x_1^2x_2+3x_1^3x_2+3x_1x_2^3+$ $+x_1^2x_2^2+3x_1^2x_2^3+3x_1^3x_2^2\rvert_5$
$x_1 \uparrow x_2$	$1-x_2/3+2x_2^2-2x_2^3/3-$ $-x_1/3-79x_1x_2/18+$ $+37x_1x_2^2/12-19x_1x_2^3/36+$ $+2x_1^2+37x_1^2x_2/12-$ $-15x_1^2x_2^2/6+5x_1^2x_2^3/12-$ $-2x_1^3/3-19x_1^3x_2/36+$ $+5x_1^3x_2^2/12-x_1^3x_2^3/18$	$\lvert 1+3x_2+2x_2^2+x_2^3+$ $+3x_1+2x_1x_2+x_1x_2^2+$ $+x_1x_2^3+2x_1^2+x_1^2x_2+$ $+x_1^3+x_1^3x_2+3x_1^3x_1^3\rvert_5$

5.3 Теоретико-числовые преобразования на k-значной логике

Введем понятие модулярного базиса ЛТЧП на k-значной логике (в дальнейшем k-ЛТЧП). Обозначим как $\mathbf{\Delta}_k$ и $\mathbf{\Delta}_k^{-1}$ соответственно базовые матрицы размерности $k \times k$ прямого и обратного k-ЛТЧП. Причем, если обозначить элементы базовых матриц k-ЛДПФ $\mathbf{K}_k$ и $\mathbf{K}_k^{-1}$ соответственно как k_{ij} и $k_{ij}^{(-1)}$, то элементы

Таблица 5.5: Расчетные $\mathbf{\Delta}_k$ и $\mathbf{\Delta}_k^{-1}$ при $m = 5$

k	$\mathbf{\Delta}_k$	$\mathbf{\Delta}_k^{-1}$
2	$\begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
3	$\begin{bmatrix} 2 & 0 & 0 \\ 2 & 4 & 4 \\ 1 & 3 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$
4	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 4 & 3 & 1 & 2 \\ 1 & 0 & 2 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 3 & 4 & 2 \end{bmatrix}$
5	$\begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 4 \\ 0 & -1 & 4 & 4 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 & 1 \\ 1 & 3 & 4 & 2 & 1 \\ 1 & 4 & 1 & 4 & 1 \end{bmatrix}$

δ_{ij} и $\delta_{ij}^{(-1)}$ матриц $\mathbf{\Delta}_k$ и $\mathbf{\Delta}_k^{-1}$ образуются из элементов матриц $\mathbf{K}_k$ и $\mathbf{K}_k^{-1}$ соответственно как $\delta_{ij} = |k_{ij}|_m$ и $\delta_{ij}^{(-1)} = \left|k_{ij}^{(-1)}\right|_m$. Тогда модулярный базис k-ЛТЧП — матрицы $\mathbf{\Delta}_{k^n}$ и $\mathbf{\Delta}_{k^n}^{-1}$ — определятся кронекеровским возведением в степень:

$$\mathbf{\Delta}_{k^n} = \bigotimes_{j=1}^{n} \mathbf{\Delta}_k \pmod m;$$

$$\mathbf{\Delta}_{k^n}^{-1} = \bigotimes_{j=1}^{n} \mathbf{\Delta}_k^{-1} \pmod m,$$

где запись $\bmod m$ означает, что все арифметические операции с элементами матриц выполняются по модулю m.

Пример 5.5
Зададим значения модуля $m = 5$ и $m = 7$. Тогда базовые матрицы $\mathbf{\Delta}_k$ и $\mathbf{\Delta}_k^{-1}$ для различных значений k будут иметь вид, представленный в табл. 5.5 и 5.6.

Теорема 5.2
Если для ФАЛ задана пара матричных преобразований (ЛДПФ)

Таблица 5.6: Расчетные $\mathbf{\Delta}_k$ и $\mathbf{\Delta}_k^{-1}$ при $m = 7$

k	$\mathbf{\Delta}_k$	$\mathbf{\Delta}_k^{-1}$
2	$\begin{bmatrix} 1 & 0 \\ 6 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$
3	$\begin{bmatrix} 2 & 0 & 0 \\ 4 & 4 & 6 \\ 1 & 5 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$
4	$\begin{bmatrix} 6 & 0 & 0 & 0 \\ 3 & 4 & 5 & 2 \\ 6 & 6 & 5 & 4 \\ 6 & 3 & 4 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 \\ 1 & 3 & 2 & 6 \end{bmatrix}$
5	$\begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 6 & 5 & 5 & 4 & 1 \\ 0 & 1 & 2 & 0 & 4 \\ 4 & 1 & 1 & 0 & 1 \\ 1 & 3 & 1 & 3 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 \\ 1 & 3 & 2 & 6 & 4 \\ 1 & 4 & 2 & 1 & 4 \end{bmatrix}$
6	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 2 & 2 & 1 & 4 & 3 \\ 1 & 0 & 6 & 4 & 4 & 1 \\ 6 & 5 & 5 & 0 & 5 & 0 \\ 1 & 0 & 4 & 6 & 6 & 4 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 3 & 2 & 6 & 4 & 5 \\ 1 & 4 & 2 & 1 & 4 & 2 \\ 1 & 5 & 3 & 6 & 2 & 3 \end{bmatrix}$

(5.6) и (5.7) и $m \geq k$ (m — простое), то справедливы преобразования:

$$\mathbf{\Psi}^{(k)} = R_k \mathbf{\Delta}_{k^n} \mathbf{S} \pmod{m}, \tag{5.6}$$

$$\mathbf{S} = \mathbf{\Delta}_{k^n}^{-1} \mathbf{\Psi}^{(k)} \pmod{m}, \tag{5.7}$$

где $R_k = \left|\frac{1}{N}\right|_m$ — модулярная форма нормализующего множителя; $\mathbf{\Delta}_{k^n}$ и $\mathbf{\Delta}_{k^n}^{-1}$ — соответственно модулярная форма матриц прямого и инверсного арифметического преобразования размерности $k^n \times k^n$ (базис преобразования); $\mathbf{S}$ — вектор истинности k-значной ФАЛ.

Доказательство

Для доказательства теоремы 5.2 необходимо учесть взаимооднозначность связи между матричной (1.10), (1.11) и полиномиальной (1.3) формами представления системы булевых функций [71]. Тогда справедливость (2.8) и (2.9) вытекает из справедливости (2.1).

Определение 5.2
Пару преобразований (2.8) и (2.9) будем соответственно называть модулярной формой прямого и обратного матричного арифметического преобразования или ЛТЧП на k-значной логике (k-ЛТЧП).

Пример 5.6
Продемонстрируем применение 3-ЛТЧП к условиям, использованным в примере 5.1. Выберем модуль $m = 5$. Предварительно вычислим матрицы $\mathbf{\Delta}_{k^n}$ и $\mathbf{\Delta}_{k^n}^{-1}$, воспользовавшись табл. 5.5:

$$\mathbf{\Delta}_{3^2} = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 4 & 4 \\ 1 & 3 & 1 \end{bmatrix} \otimes \begin{bmatrix} 2 & 0 & 0 \\ 2 & 4 & 4 \\ 1 & 3 & 1 \end{bmatrix} \pmod 5 =$$

$$= \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 \\ 4 & 3 & 3 & 3 & 1 & 1 & 3 & 1 & 1 \\ 2 & 1 & 2 & 4 & 2 & 4 & 4 & 2 & 4 \\ 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 2 & 4 & 4 & 1 & 2 & 2 & 2 & 4 & 4 \\ 1 & 3 & 1 & 3 & 4 & 3 & 1 & 3 & 1 \end{bmatrix};$$

$$\mathbf{\Delta}_{3^2}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \pmod 5 =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 \\ 1 & 2 & 4 & 2 & 4 & 3 & 4 & 3 & 1 \end{bmatrix}.$$

Прямое 3-ЛТЧП (5.6), при с учетом $R_3 = 4$, примет вид:

$$\mathbf{\Psi}^{(3)} = R_3 \cdot \mathbf{\Delta}_{3^2}\mathbf{S} \pmod 5 =$$

$$= 4 \cdot \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 \\ 4 & 3 & 3 & 3 & 1 & 1 & 3 & 1 & 1 \\ 2 & 1 & 2 & 4 & 2 & 4 & 4 & 2 & 4 \\ 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 2 & 4 & 4 & 1 & 2 & 2 & 2 & 4 & 4 \\ 1 & 3 & 1 & 3 & 4 & 3 & 1 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} \pmod 5 =$$

$$= \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 0 \\ 1 \\ 1 \end{bmatrix} \begin{matrix} x_2 \\ x_2^2 \\ x_1 \\ x_1x_2 \\ x_1x_2^2 \\ x_1^2 \\ x_1^2x_2 \\ x_1^2x_2^2 \end{matrix} .$$

Обратное преобразование (5.7) будет иметь вид:

$$\mathbf{S} = \boldsymbol{\Delta}_{3^2}^{-1}\boldsymbol{\Psi}^{(3)} \pmod 5 =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 4 & 4 & 4 \\ 1 & 2 & 4 & 2 & 4 & 3 & 4 & 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 0 \\ 1 \\ 1 \end{bmatrix} \begin{matrix} x_2 \\ x_2^2 \\ x_1 \\ x_1x_2 \\ x_1x_2^2 \\ x_1^2 \\ x_1^2x_2 \\ x_1^2x_2^2 \end{matrix} \pmod 5 =$$

$$= \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} .$$

По аналогии с методикой расчета преимуществ модулярных форм арифметической логики для $k = 2$ (2.11) можно определить выигрыш от использования модулярных форм для k-значной логики. В частности для $k = 3$ выигрыш (размерность данных) от использования модулярных форм составит $\frac{3n}{\lceil \log_2 3^d \rceil}$ раз. Из рис. 5.6 можно видеть, что преимущества модулярных форм с увеличением k возрастают.

Метод вычисления многозначных ФАЛ на основе одномодульной арифметики позволяет реализовать гибкие логические вычисления с помощью аппаратных средств, функционирующих по заданному значению модуля (а не по модулю k). В качестве модулей определенные преимущества имеют простые числа Мерсенна и числа Ферма. Размерность промежуточных результатов при использовании модулярных форм уменьшается, что

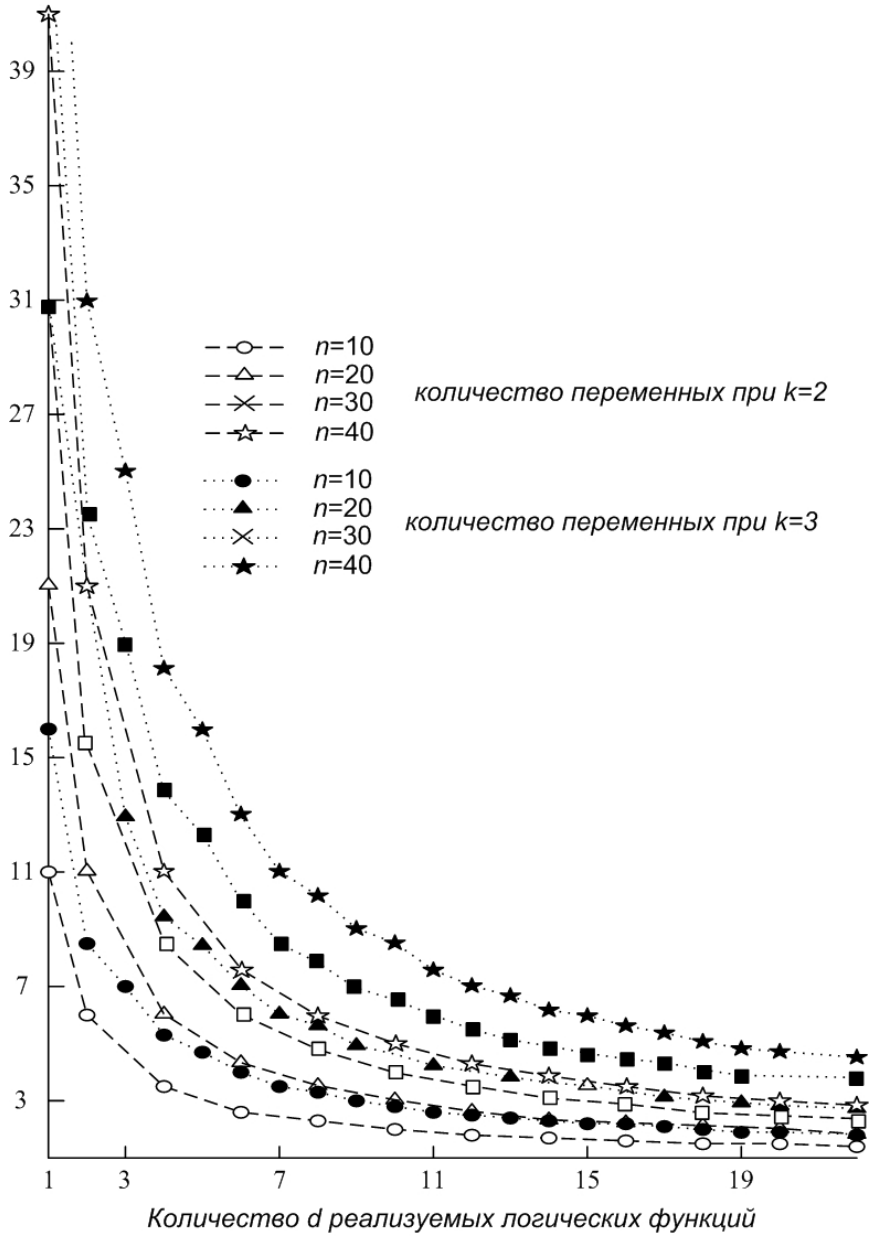


Рис. 5.6: Теоретический выигрыш (количество раз) — уменьшение максимальной размерности (количества бит) коэффициентов арифметического полинома для $k = 2$ и $k = 3$

позволяет отказаться от использования арифметики многократной точности и сократить время логических вычислений.

5.4 Реализация систем k-значных функций

Не затрагивая вопросов представления систем k-значных ФАЛ [37] линейными арифметическими полиномами, рассмотрим некоторые принципиальные вопросы для нелинейных арифметических полиномов, решение которых могло бы дать обобщение и на линейные полиномы.

5.4.1 Арифметический полином для реализации систем k-значных функций

Будем использовать аналогию с алгоритмом 1.1.

Алгоритм 5.1

Шаг 1. *Поставим в соответствие ФАЛ*

$$f_1^{(k)}(X),\ f_2^{(k)}(X),\ \ldots,\ f_d^{(k)}(X),$$

арифметические полиномы вида (5.1):

$$\begin{aligned}
P_1^{(k)}(X) &= \sum_{i=0}^{k^n-1} p_{1,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},\\
P_2^{(k)}(X) &= \sum_{i=0}^{k^n-1} p_{2,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},\\
&\vdots\\
P_d^{(k)}(X) &= \sum_{i=0}^{k^n-1} p_{d,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}.
\end{aligned}$$

Шаг 2. *Умножим эти полиномы на веса* k^{j-1} $(j = 1, 2, \ldots, d)$*:*

$$P_1^{*(k)}(X) = k^0 P_1^{(k)}(X) = \sum_{i=0}^{k^n-1} p_{1,\,i}^{*(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},$$

$$\begin{aligned} P_2^{*(k)}(X) &= k^1 P_1^{(k)}(X) = \sum_{i=0}^{k^n-1} p_{2,i}^{*(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \\ &\vdots \\ P_d^{*(k)}(X) &= k^{d-1} P_1^{(k)}(X) = \sum_{i=0}^{k^n-1} p_{d,i}^{*(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \end{aligned}$$

где $p_{j,i}^{*(k)} = k^{j-1} p_{j,i}^{(k)}$. $(j = 1, 2, \dots, d;\ i = 0, 1, \dots, k^n - 1)$

Шаг 3. *Получение арифметического полинома*

$$\begin{aligned} D^{(k)}(X) &= \sum_{i=0}^{k^n-1} \sum_{j=1}^{d} p_{j,i}^{*(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} = \\ &= \sum_{i=0}^{k^n-1} c_i^{(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \end{aligned} \tag{5.8}$$

где $c_i^{(k)} = \sum_{j=1}^{d} p_{j,i}^{*(k)}$ $(i = 0, 1, \dots, k^n - 1)$.

Если результат вычисления (5.8), получить в k-ичной системе счисления, то он будет представлять собой кортеж значений искомых ФАЛ

$$y_1^{(k)}(X) \odot y_2^{(k)}(X) \odot \dots \odot y_d^{(k)}(X),$$

интерпретируемый как код целого неотрицательного числа:

$$(y_d^{(k)}\, y_{d-1}^{(k)}\, \dots\, y_1^{(k)})_k = Y^{(k)} = \sum_{j=1}^{d} y_j^{(k)} k^{j-1}.$$

Однако, в силу того, что результат $Y^{(k)}$ представлен в двоичной системе счисления, для значений ФАЛ применяется оператор маскирования $\Xi^t\{Y^{(k)}\}$, служащий для определения t-го k-ичного разряда (выхода) представления

$$Y^{(k)} = (y_d^{(k)}\ \dots\ y_t^{(k)}\ \dots\ y_1^{(k)})_k,$$

т. е. $\Xi^t\{Y^{(k)}\} = y_t^{(k)}$.

Для вычисления оператора $\Xi^t\{Y^{(k)}\}$ используется формула:

$$\Xi^t\{Y^{(k)}\} = \left|\left\lfloor \frac{Y^{(k)}}{k^t} \right\rfloor\right|_k. \tag{5.9}$$

5.4.2 Модулярная форма для реализации систем k-значных функций

Теорема 5.3

Если $m > Y_{\max}^{(k)}$, где $Y_{\max}^{(k)}$ — максимальное значение, принимаемое $Y^{(k)}$, то произвольный кортеж k-значных ФАЛ может быть представлен арифметическим полиномом:

$$Y^{(k)} = \Omega^{(k)}(X) = \left| \sum_{i=0}^{k^n-1} \omega_i^{(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m, \tag{5.10}$$

где $\omega_i^{(k)} = \left|c_i^k\right|_m$ $(i = 0,\, 1,\, \dots,\, k^n - 1)$.

Доказательство

В качестве доказательства построим алгоритм.

Алгоритм 5.2

Шаг 1. *Поставим d ФАЛ:*

$$f_1^{(k)}(X),\ f_2^{(k)}(X),\ \dots,\ f_d^{(k)}(X)$$

в соответствие арифметические полиномы вида (5.5):

$$\begin{aligned}
\mu_1^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{1,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m, \\
\mu_2^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{2,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m, \\
&\vdots \\
\mu_d^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{d,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m.
\end{aligned}$$

Шаг 2. *Вычисление модулярных произведений:*

$$\begin{aligned}
\left|l^0\mu_1^{(k)}(X)\right|_m &= \left|\sum_{i=0}^{k^n-1}\rho_{1,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m,\\
\left|l^1\mu_1^{(k)}(X)\right|_m &= \left|\sum_{i=0}^{k^n-1}\rho_{1,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m,\\
&\vdots\\
\left|l^{d-1}\mu_1^{(k)}(X)\right|_m &= \left|\sum_{i=0}^{k^n-1}\rho_{1,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m,
\end{aligned}$$

где $\rho_{j,i}^{*(k)} = \left|l^{j-1}\rho_{j,i}^{(k)}\right|_m$; $l = l^{j-1} = |k^{j-1}|_m$; $j = 1, 2, \ldots, d$; $i = 0, 1, \ldots, k^n - 1$.

Шаг 3.

$$\begin{aligned}
\Omega^{(k)}(X) &= \left|\sum_{i=0}^{k^n-1}\sum_{j=1}^{d}\rho_{j,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m =\\
&= \left|\sum_{i=0}^{k^n-1}\omega_i^{(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m,
\end{aligned}$$

где $\omega_i = \sum_{j=1}^{d}\omega_i^{*(k)}$ $(i = 0, 1, \ldots, k^n - 1)$.

Структура алгоритма 5.2 аналогична структуре 1.3.

□

5.5 Выводы

- Получено обобщение модулярных форм арифметических полиномов на область k-значной логики. При этом достигаются преимущества по ограничению величины коэффициентов арифметического полинома и величины промежуточных результатов вычислений.

- Показано, что посредством модулярной формы арифметического полинома вычисления k-значных логических функций из поля рациональных чисел $\mathcal{R}$ могут быть перенесены на простое поле Галуа $PG(m)$.
- Установлена взаимосвязь между немодулярной и модулярной формами арифметического полинома на k-значной логике.
- Построены теоретико-числовые преобразования на k-значной логике.
- Модулярные формы обобщены для реализации систем k-значных ФАЛ.
- Указано, что предложенные модулярные формы позволяют повысить гибкость логических вычислений за счет обеспечения возможности применения аппаратных или программных средств, функционирующих не по „жестко“ заданному значению модуля k, а и по другим значениям модуля. Или наоборот — когда необходимо выполнять вычисления для различных значностей логики — k на основе аппаратных или программных средств, функционирующих по одному, заранее заданному значению модуля.

Глава 6

Альтернативные арифметико-логические формы

Все рассмотренные выше методы построения арифметических форм основаны на принципе взвешивания — умножения полиномов, соответствующих отдельным логическим функциям, на степени значности k, или их модулярные эквиваленты. Необходимо отметить, что принцип взвешивания — не единственный прием, который может быть использован для построения (кодирования) арифметических форм и последующего различения (маскирования) логических функций. Рассмотрим два альтернативных метода кодирования, основанные на 1) единственности представления числа его разложением на простые множители и 2) Китайской теореме об остатках. Рассмотрим также случаи, в которых с помощью этих методов можно было бы получить некоторую пользу[1].

6.1 Мультипликативно-кодированные арифметико-логические формы

Теорема 6.1
Произвольный кортеж булевых функций

$$y_d \odot y_{d-1} \odot \ldots \odot y_1$$

[1] Материалы данной главы публикуются впервые.

может быть единственным способом задан арифметическим выражением

$$N(X) = \prod_{i=1}^{2^n-1} N_0 N_i^{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}, \tag{6.1}$$

где $N_i > 0$, $i = 0, 1, \ldots, 2^n - 1$.

Доказательство

Сопоставим булевым функциям $f_i(X)$ числа $m_i^{f_i(X)}$ ($i = 1, 2, \ldots, d$), где m_i ($i = 1, 2, \ldots, d$) — простые числа:

$$\begin{array}{cccc} f_1(X) & f_2(X) & \cdots & f_d(X) \\ \downarrow & \downarrow & & \downarrow \\ m_1^{f_1(X)} & m_2^{f_2(X)} & \cdots & m_d^{f_d(X)}. \end{array}$$

Построим с учетом (1.3) произведение

$$\begin{gathered} N(X) = m_1^{f_1(X)} \times \ldots \times m_d^{f_d(X)} = \\ = m_1^{\sum_{i=0}^{2^n-1} r_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}} \times \ldots \times m_d^{\sum_{i=0}^{2^n-1} r_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}. \end{gathered} \tag{6.2}$$

Далее, (6.2) можем переписать в виде

$$\begin{array}{rcccccl} N(X) = & \left(m_1^{r_{1,0}}\right. & \times & m_1^{r_{1,1} x_n} & \times \ldots \times & \left. m_1^{r_{1,2^n-1} x_1 x_2 \ldots x_n}\right) & \times \\ \times & \left(m_2^{r_{2,0}}\right. & \times & m_2^{r_{2,1} x_n} & \times \ldots \times & \left. m_2^{r_{2,2^n-1} x_1 x_2 \ldots x_n}\right) & \times \\ & \vdots & & & & & \vdots \\ \times & \left(m_d^{r_{d,0}}\right. & \times & m_d^{r_{d,1} x_n} & \times \ldots \times & \left. m_d^{r_{d,2^n-1} x_1 x_2 \ldots x_n}\right) & = \end{array}$$

$$\begin{array}{rcccl} = & \left(m_1^{r_{1,0}}\right. & \times \ldots \times & \left. m_d^{r_{d,0}}\right) & \times \\ \times & \left(m_1^{r_{1,1} x_n}\right. & \times \ldots \times & \left. m_d^{r_{d,1} x_n}\right) & \times \\ & \vdots & & & \vdots \\ \times & \left(m_1^{r_{d,2^n-1} x_1 x_2 \ldots x_n}\right. & \times \ldots \times & \left. m_d^{r_{d,2^n-1} x_1 x_2 \ldots x_n}\right) & = \end{array}$$

$$= N_0 N_1^{x_n} \ldots N_{2^n-1}^{x_1 x_2 \ldots x_n},$$

где

$$\begin{aligned} N_0 &= m_1^{r_{1,0}} \times \ldots \times m_d^{r_{d,0}}, \\ N_1 &= m_1^{r_{1,1}x_n} \times \ldots \times m_d^{r_{d,1}\,x_n}, \\ &\vdots \\ N_{2^n-1} &= m_1^{r_{d,2^n-1}\,x_1x_2\ldots x_n} \times \ldots \times m_d^{r_{d,2^n-1}\,x_1x_2\ldots x_n}. \end{aligned}$$

Единственность (6.1) следует из 1) основной теоремы арифметики о единственности разложения числа на простые множители и 2) из единственности представления (1.3) в соответствии с теоремой 1.1.

□

Определение 6.1
Выражение (6.1) будем называть мультипликативно-кодированной арифметико-логической формой.

Свойство 6.1
В результате вычисления (6.1) значение t-й булевой определяется проверкой условия делимости $N(X)$ на число m_t:

$$f_t(X) = \begin{cases} 1, & \textit{если} \quad m_t | N(X), \\ 0, & \textit{если} \quad m_t \not| N(X) \end{cases} \tag{6.3}$$

или

$$f_t(X) = \begin{cases} 1, & \textit{если} \quad |N(X)|_{m_t} = 0, \\ 0, & \textit{если} \quad |N(X)|_{m_t} \neq 0. \end{cases}$$

Для проверки выполнимости (6.3) могут быть использованы признаки делимости [25, 28].

Свойство 6.2

$$\mu\left(N(X)\right) = (-1)^d, \quad \textit{при} \quad \bigvee_{i=1}^{d} y_i = 1,$$

$$\mu\left(N(X)\right) = 1, \quad \textit{при} \quad \bigvee_{i=1}^{d} y_i = 0,$$

где $\mu(a)$ — функция Мебиуса от a [25].

Можно предложить следующий алгоритм получения (6.1).

Алгоритм 6.1

Шаг 1. *Определение коэффициентов $r_{1,i}, r_{2,i}, \ldots, r_{d,i}$ путем получения арифметического полинома $P_j(X)$ для каждой булевой функции $y_j = f_j(X)$, $j = 1, 2, \ldots, d$:*

$$\begin{cases} f_1(X) = P_1(X) = \sum\limits_{i=0}^{2^n-1} r_{1,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ f_2(X) = P_2(X) = \sum\limits_{i=0}^{2^n-1} r_{2,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\ \vdots \\ f_d(X) = P_d(X) = \sum\limits_{i=0}^{2^n-1} r_{d,i}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}. \end{cases} \tag{6.4}$$

Шаг 2. *Выбор d простых чисел $m_1, m_2, \ldots, m_d$ и вычисление $N_0, N_1, \ldots, N_{2^n-1}$:*

$$\begin{aligned} N_0 &= m_1^{r_{1,0}} \times \ldots \times m_d^{r_{d,0}}, \\ N_1 &= m_1^{r_{1,1} x_n} \times \ldots \times m_d^{r_{d,1} x_n}, \\ &\vdots \\ N_{2^n-1} &= m_1^{r_{d,2^n-1} x_1 x_2 \ldots x_n} \times \ldots \times m_d^{r_{d,2^n-1} x_1 x_2 \ldots x_n}. \end{aligned}$$

Пример 6.1
Используем условия таблицы 1.1 (пример 1.1, стр. 19).

Шаг 1. *Согласно примеру 1.2 (стр. 23):*

$$\begin{aligned} f_1(X) &= P_1(X) = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1x_2, \\ f_2(X) &= P_2(X) = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1x_2. \end{aligned}$$

Шаг 2. *Выберем первые два простых числа (порядок следования простых чисел произволен)* $m_1 = 2$, $m_2 = 3$ *и вычислим числа* $N_0, \ldots, N_3$*:*

$$\begin{aligned} N_0 &= 2^1 \cdot 3^1 = 6, \\ N_1 &= 2^{-x_2} \cdot 3^{-x_2} = \left(\frac{1}{6}\right)^{x_2}, \\ N_2 &= 2^{-x_1} \cdot 3^{-x_1} = \left(\frac{1}{6}\right)^{x_1}, \\ N_3 &= 2^{2x_1x_2} \cdot 3^{x_1x_2} = 12^{x_1x_2}. \end{aligned}$$

Таким образом

$$N(X) = 6 \times \left(\frac{1}{6}\right)^{x_2} \times \left(\frac{1}{6}\right)^{x_1} \times 12^{x_1x_2}.$$

Результат реализации функций $f_1(X)$ *и* $f_2(X)$ *на полном наборе булевых переменных* x_1 *и* x_2 *представлен в табл. 6.1.*

Таблица 6.1: Пример реализации булевых функций (двоичный полусумматор) с помощью мультипликативно-кодированной арифметико-логической формы

x_2	x_1	$N(X)$	Проверка условия (6.3)		y_2	y_1
0	0	6	$\|6\|_3 = 0$	$\|6\|_2 = 0$	1	1
0	1	1	$\|1\|_3 \neq 0$	$\|1\|_2 \neq 0$	0	0
1	0	1	$\|1\|_3 \neq 0$	$\|1\|_2 \neq 0$	0	0
1	1	2	$\|2\|_3 \neq 0$	$\|2\|_2 = 0$	0	1

6.2 Модулярно-кодированные арифметико-логические формы

Как следует из (5.9) для реализации оператора $\Xi^t\{Y^{(k)}\}$ требуется деление, округление и нахождение наименьшего неотрица-

тельного вычета. Это обстоятельство уменьшает скорость вычислений на этапе реализации ФАЛ. Рассмотрим альтернативный принцип построения арифметического полинома, который в отличие принципа *декомпозиции*, использованного в разделе 3, основан на *синтезе* одного арифметического полинома, обладающего свойством упрощения оператора маскирования.

Пусть дана система d ФАЛ:

$$f_1^{(k)}(X),\ f_2^{(k)}(X),\ \ldots,\ f_d^{(k)}(X),$$

где k — значность ФАЛ и, поставленная в соответствие им, система арифметических полиномов вида (5.5):

$$\mu_1^{(k)}(X),\ \mu_2^{(k)}(X),\ \ldots,\ \mu_d^{(k)}(X).$$

Выберем простые модули: $m_1, m_2, \ldots, m_d$ такие, что $m_i \geq k$, где $i = 1, 2, \ldots, d$. Соблюдение условия простоты модулей необходимо для обеспечения вычислений в конечном поле.

Будем отождествлять полиномы $\mu_1^{(k)}(X),\ \mu_2^{(k)}(X),\ \ldots,\ \mu_d^{(k)}(X)$ с вычетами $\phi_1, \phi_2, \ldots, \phi_d$ в формуле (3.8):

$$Y = \left|\phi_1 B_1 + \phi_2 B_2 + \ldots + \phi_d B_d\right|_m,$$

где $m = m_1 m_2 \ldots m_d$.

В связи с этим рассмотрим следующую теорему.

Теорема 6.2
Если даны простые модули $m_1, m_2, \ldots, m_d$ такие, что $m_i \geq k$, где $i, j = 1, 2, \ldots, d$, то произвольный кортеж k-значных ФАЛ может быть представлен арифметическим полиномом:

$$T^{(k)} = \Theta^{(k)}(X) = \left|\sum_{i=0}^{k^n-1} \zeta_i^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}\right|_m, \qquad (6.5)$$

где $0 \leq \zeta_i^{(k)} < m$, причем

$$y_1^{(k)} = \left|T^{(k)}\right|_{m_1},\ y_2^{(k)} = \left|T^{(k)}\right|_{m_2},\ \ldots,\ y_d^{(k)} = \left|T^{(k)}\right|_{m_d}.$$

Доказательство
Справедливость теоремы 6.2 вытекает из следующего алгоритма.

Алгоритм 6.2

Шаг 1. *Построение полиномов вида (5.5) для представления каждой ФАЛ:*

$$\begin{aligned}
\mu_1^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{1,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_1}, \\
\mu_2^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{2,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_2}, \\
&\vdots \\
\mu_d^{(k)}(X) &= \left| \sum_{i=0}^{k^n-1} \rho_{d,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|_{m_d};
\end{aligned}$$

и запись в виде:

$$\begin{aligned}
\mu_1^{*(k)}(X) &= \sum_{i=0}^{k^n-1} \rho_{1,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\
\mu_2^{*(k)}(X) &= \sum_{i=0}^{k^n-1} \rho_{2,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \\
&\vdots \\
\mu_d^{*(k)}(X) &= \sum_{i=0}^{k^n-1} \rho_{d,\,i}^{(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}.
\end{aligned}$$

Шаг 2. *Выполнение модулярных умножений:*

$$\left| B_1 \mu_1^{*(k)}(X) \right|_m = \sum_{i=0}^{k^n-1} \zeta_{1,\,i}^{*(k)}\, x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},$$

$$\begin{aligned}
\left|B_2\mu_1^{*(k)}(X)\right|_m &= \sum_{i=0}^{k^n-1}\zeta_{2,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n},\\
&\vdots\\
\left|B_d\mu_1^{*(k)}(X)\right|_m &= \sum_{i=0}^{k^n-1}\zeta_{d,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n},
\end{aligned}$$

где $\zeta_{j,i}^{*(k)}(X) = \left|B_j\rho_{j,i}^{(k)}\right|_m$; $m = m_1m_2\ldots m_d$; $B_i = q_imm_i^{-1}$; q_i *находится из* $q_imm_k^{-1} \equiv 1 \pmod{m_i}$ $(i = 1, 2, \ldots, d)$.

Шаг 3.

$$\begin{aligned}
\Theta^{(k)}(X) &= \left|\sum_{i=0}^{k^n-1}\sum_{j=1}^{d}\zeta_{j,i}^{*(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m =\\
&= \left|\sum_{i=0}^{k^n-1}\zeta_i^{(k)}\,x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n}\right|_m,
\end{aligned}$$

где $\zeta_i = \sum_{j=1}^{d}\zeta_{j,i}^{*(k)}$ $(i = 0, 1, \ldots, k^n - 1)$.

Корректность вычислений в конечном поле обеспечивается соблюдением условия выбора простых модулей.

□

Определение 6.2
Выражение (6.5) будем называть модулярно-кодированной арифметико-логической формой.

Пример 6.2
Рассмотрим применение теоремы 6.2 для случая реализации двух 3-х значных ФАЛ, векторы значений которых при $n = 2$, *имеют вид:*

$$\begin{aligned}
\mathbf{S}_1 &= [0\ 0\ 2\ 2\ 1\ 1\ 0\ 0\ 2]^T,\\
\mathbf{S}_2 &= [2\ 0\ 2\ 1\ 1\ 1\ 0\ 1\ 2]^T.
\end{aligned}$$

Им соответствуют арифметические полиномы вида (5.1):

$$\begin{aligned} P_1^{(3)}(X) &= -x_2 + x_2^2 + 4x_1 - x_1x_2 - x_1x_2^2 - 2x_1^2 + \\ &\quad + \frac{1}{2}x_1^2x_2 + \frac{1}{2}x_1^2x_2^2, \\ P_2^{(3)}(X) &= 2 - 4x_2 + 2x_2^2 - x_1 + \frac{11}{2}x_1x_2 - 3x_1x_2^2 - \\ &\quad - \frac{3}{2}x_1^2x_2 + x_1^2x_2^2. \end{aligned}$$

В соответствии с теоремой 6.2 выберем модули: $m_1 = 3$, $m_2 = 5$. *Тогда* $m = m_1m_2 = 15$, *значения констант* B_1 *и* B_2 *в соответствии с (3.8) составят:*

$$\begin{aligned} B_1 &= \frac{q_1m}{m_1} = \frac{2 \cdot 15}{3} = 10, \\ B_2 &= \frac{q_2m}{m_2} = \frac{2 \cdot 15}{5} = 6. \end{aligned}$$

Применение алгоритма 6.2 в этом случае будет выглядеть следующим образом.

Шаг 1. *Построение полиномов вида (5.5) (пример 5.6):*

$$\begin{aligned} \mu_1^{(3)}(X) &= \left|2x_2 + x_2^2 + x_1 + 2x_1x_2 + 2x_1x_2^2 + x_1^2 + \right. \\ &\quad \left. + 2x_1^2x_2 + 2x_1^2x_2^2\right|_3, \\ \mu_2^{(3)}(X) &= \left|2 + x_2 + 2x_2^2 + 4x_1 + 3x_1x_2 + 2x_1x_2^2 + \right. \\ &\quad \left. + x_1^2x_2 + x_1^2x_2^2\right|_5 \end{aligned}$$

и запись их в немодулярном виде:

$$\begin{aligned} \mu_1^{*(3)}(X) &= 2x_2 + x_2^2 + x_1 + 2x_1x_2 + 2x_1x_2^2 + x_1^2 + \\ &\quad + 2x_1^2x_2 + 2x_1^2x_2^2, \\ \mu_2^{*(3)}(X) &= 2 + x_2 + 2x_2^2 + 4x_1 + 3x_1x_2 + 2x_1x_2^2 + \\ &\quad + x_1^2x_2 + x_1^2x_2^2. \end{aligned}$$

Шаг 2. *Выполнение модулярных умножений:*

$$\begin{aligned}\left|10\mu_1^{*(3)}(X)\right|_{15} &= |10\cdot 2|_{15}x_2 + 10x_2^2 + 10x_1 + |10\cdot 2|_{15}x_1x_2 + \\ &\quad +|10\cdot 2|_{15}x_1x_2^2 + 10x_1^2 + |10\cdot 2|_{15}x_1^2x_2 + \\ &\quad +|10\cdot 2|_{15}x_1^2x_2^2 = \\ &= 5x_2 + 10x_2^2 + 10x_1 + 5x_1x_2 + 5x_1x_2^2 + \\ &\quad +10x_1^2 + 5x_1^2x_2 + 5x_1^2x_2^2, \\ \left|6\mu_1^{*(3)}(X)\right|_{15} &= |6\cdot 2|_{15} + 6x_2 + |6\cdot 2|_{15}x_2^2 + |6\cdot 4|_{15}x_1 + \\ &\quad +|6\cdot 3|_{15}x_1x_2 + |6\cdot 2|_{15}x_1x_2^2 + 6x_1^2x_2 + \\ &\quad +6x_1^2x_2^2 = \\ &= 12 + 6x_2 + 12x_2^2 + 9x_1 + 3x_1x_2 + 12x_1x_2^2 + \\ &\quad +6x_1^2x_2 + 6x_1^2x_2^2.\end{aligned}$$

Шаг 3.

$$\begin{aligned}\Theta^{(3)}(X) &= |0+12|_{15} + |5+6|_{15}x_2 + |10+12|_{15}x_2^2 + \\ &\quad +|10+9|_{15}x_1 + |5+3|_{15}x_1x_2 + \\ &\quad +|5+12|_{15}12x_1x_2^2 + |10+0|_{15}x_1^2 + \\ &\quad +|5+6|_{15}x_1^2x_2 + |5+6|_{15}x_1^2x_2^2.\end{aligned}$$

Таким образом

$$\begin{aligned}\Theta^{(3)}(X) = 12 + 11x_2 + 7x_2^2 + 4x_1 + 8x_1x_2 + 2x_1x_2^2 + \\ +10x_1^2 + 11x_1^2x_2 + 11x_1^2x_2^2.\end{aligned}$$

Для проверки рассмотрим два случая. Первый случай — $x_1 = 0$ и $x_2 = 0$ (первая строка таблицы истинности). Второй случай — $x_1 = 2$ и $x_2 = 2$ (последняя строка таблицы истинности).
В первом случае получим:

$$\Theta^{(3)}(X) = |12|_{15} = 12.$$

Результат (см. исходные условия):

$$\begin{aligned} y_1^{(3)} &= |12|_3 = 0, \\ y_2^{(3)} &= |12|_5 = 2. \end{aligned}$$

Во втором случае:

$$\begin{aligned} \Theta^{(3)}(X) &= \big|12 + |11 \cdot 2|_{15} + |7 \cdot 2^2|_{15} + |4 \cdot 2|_{15} + \\ &\quad + |8 \cdot 2 \cdot 2|_{15} + |2 \cdot 2 \cdot 2^2|_{15} + \\ &\quad + |10 \cdot 2^2|_{15} + |11 \cdot 2^2 \cdot 2|_{15} + \\ &\quad + |11 \cdot 2^2 \cdot 2^2|_{15}\big|_{15} = \\ &= |12 + 7 + 13 + 8 + 2 + 1 + 10 + 13 + 11|_{15} = \\ &= |77|_{15} = 2. \end{aligned}$$

Результат:

$$\begin{aligned} y_1^{(3)} &= |2|_3 = 2, \\ y_2^{(3)} &= |2|_5 = 2. \end{aligned}$$

6.3 Выводы

- Предложены мультипликативно-кодированные арифметико-логические формы. Реализация систем булевых функций с помощью мультипликативно-кодированных арифметико-логических форм может быть полезной при выполнении вычислений в недвоичной системе счисления. Обобщение (6.1) для k-значной логики и использование для формирования (6.1) линейных арифметических полиномов приведет к необходимости факторизации числа $N(X)$, что существенно усложнит вычисления.

- Предложены модулярно-кодированные арифметико-логические формы k-значных ФАЛ, основанные на Китайской теореме об остатках. Эти формы являются результатом синтеза, в отличие от арифметико-логических

форм (3.1) и (3.6), основанных на декомпозиции исходного полинома. Преимущество данных форм — упрощение реализации оператора маскирования.

Глава 7

Восстановление позиционной формы числа

Важное место в формах арифметической логики, основанных на многомодульной арифметике, занимают методы восстановления позиционной формы числа. Китайская теорема об остатках в своем классическом варианте (теорема А.28) далеко не всегда дает оптимальное для технической реализации решение. Это стимулирует поиск новых вариантов Китайской теоремы об остатках, удовлетворяющих тем или иным требованиям технической реализации [119][1].

7.1 Теорема о восстановлении для двух модулей

Рассмотрим классическую формулу восстановления позиционной формы числа для случая двух модулей:

$$A = |\phi_1 m_2 q_1 + \phi_2 m_1 q_2|_M \,, \tag{7.1}$$

где q_1 и q_2 — решения сравнений $m_2 q_1 \equiv 1 \pmod{m_1}$ и $m_1 q_2 \equiv 1 \pmod{m_2}$, $M = m_1 m_2$.

Для аппаратурной реализации алгоритма восстановления позиционной формы числа в соответствии с (7.1) потребуется два умножителя на константы $m_2 q_1$ и $m_1 q_2$ и один сумматор, функ-

[1] Оригинальные материалы этой главы впервые опубликованы в [4, 8, 85, 86, 104, 105, 107].

ционирующий по модулю $M = m_1 m_2$ — числовому диапазону модулярной арифметики (рис. 7.1).

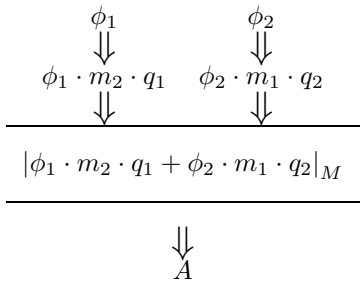


Рис. 7.1: Схема вычисления позиционной формы числа по двум остаткам в соответствии с (7.1)

Теорема 7.1
Целое неотрицательное число $A < m_1 m_2$, представленное в двухмодульной арифметике вычетами ϕ_1 и ϕ_2 по системе взаимно простых модулей m_1 и m_2, таких, что $m_1 < m_2$, может быть однозначно восстановлено по формуле

$$A = m_1 \left| \delta \left| \phi_1 - \phi_2 \right|_{m_2} \right|_{m_2} + \phi_1, \tag{7.2}$$

где $\delta = |(m_2 - m_1)^{-1}|_{m_2}$.

Доказательство
В соответствии с алгоритмом Евклида число A можно представить в виде

$$A = m_1 n_1 + \phi_1, \tag{7.3}$$

или

$$A = (m_2 - b) n_1 + \phi_1,$$

где $b = m_2 - m_1$.

Поскольку $A \equiv \phi_2 \pmod{m_2}$, то

$$(m_2 - b) n_1 + \phi_1 \equiv \phi_2 \pmod{m_2}.$$

Учитывая, что $m_2 n_1 \equiv 0 \pmod{m_2}$, получаем

$$b n_1 \equiv \phi_1 - \phi_2 \pmod{m_2}$$

или

$$n_1 \equiv b^{-1}(\phi_1 - \phi_2) \pmod{m_2}. \tag{7.4}$$

Согласно формулы Эйлера (следствие А.2):

$$b^{-1} \equiv b^{\varphi(m_2)-1} \pmod{m_2}, \tag{7.5}$$

где $\varphi(m_2)$ — функция Эйлера (теорема А.22). Подставляя выражение (7.4) в (7.3), с учетом (7.5) получаем (7.2).

□

Пример 7.1

Рассмотрим пару ЛТЧП КТО (3.9) и (3.10) применительно к условиям, использованным в примерах 1.5 и 2.5. При этом выберем значения модулей $m_1 = 3$ и $m_2 = 5$, что должно обеспечить числовой диапазон $M = m_1 m_2 = 15 \geq 2^3$. Тогда

$$|\mathbf{Y}|_3 = \left|\begin{bmatrix} 0&0&0\\1&1&1\\1&1&0\\0&0&1\\1&0&1\\0&1&0\\0&1&1\\1&1&1 \end{bmatrix}\right|_3 = \left|\begin{bmatrix} 0\\7\\6\\1\\5\\2\\3\\7 \end{bmatrix}\right|_3 = \begin{bmatrix} 0\\1\\0\\1\\2\\2\\0\\1 \end{bmatrix},$$

$$|\mathbf{Y}|_5 = \left|\begin{bmatrix} 0&0&0\\1&1&1\\1&1&0\\0&0&1\\1&0&1\\0&1&0\\0&1&1\\1&1&1 \end{bmatrix}\right|_5 = \left|\begin{bmatrix} 0\\7\\6\\1\\5\\2\\3\\7 \end{bmatrix}\right|_5 = \begin{bmatrix} 0\\2\\1\\1\\0\\2\\3\\2 \end{bmatrix}.$$

Преобразование (3.9) примет вид:

$$\begin{aligned} \mathbf{\Psi}_1 &= \mathbf{A}_{2^3}\mathbf{\Phi}_1 \pmod 3 = \mathbf{A}_{2^3} \cdot [0\ 1\ 0\ 1\ 2\ 2\ 0\ 1]^T \pmod 3 = \\ &= [0\ 1\ 0\ 0\ 2\ 2\ 1\ 1]^T, \\ \mathbf{\Psi}_2 &= \mathbf{A}_{2^3}\mathbf{\Phi}_2 \pmod 5 = \mathbf{A}_{2^3} \cdot [0\ 2\ 1\ 1\ 0\ 2\ 3\ 2]^T \pmod 5 = \\ &= [0\ 2\ 1\ 3\ 0\ 0\ 2\ 4]^T. \end{aligned}$$

Для получения Y используем (7.2):

$$Y^{(i)} = m_1 \left|\delta_1 \left|\phi_{i,1} - \phi_{i,2}\right|_{m_2}\right|_{m_2} + \phi_{i,1} = 2\left|2\left|\phi_{i,1} - \phi_{i,2}\right|_5\right|_5 + \phi_{i,1},$$

где $i = 0, 1, \ldots, 2^n - 1$. *В результате получим:*

$$Y^{(0)} = 2\,|2\,|0-0|_5|_5 + 0 = 0, \quad Y^{(4)} = 2\,|2\,|2-0|_5|_5 + 2 = 5,$$

$$Y^{(1)} = 2\,|2\,|1-2|_5|_5 + 1 = 7, \quad Y^{(5)} = 2\,|2\,|2-2|_5|_5 + 2 = 2,$$

$$Y^{(2)} = 2\,|2\,|0-1|_5|_5 + 0 = 6, \quad Y^{(6)} = 2\,|2\,|0-3|_5|_5 + 0 = 3,$$

$$Y^{(3)} = 2\,|2\,|1-1|_5|_5 + 1 = 1, \quad Y^{(7)} = 2\,|2\,|1-2|_5|_5 + 1 = 7.$$

Таким образом:

$$\begin{aligned} \mathbf{\Psi} &= \mathbf{CRT}^2_{k=1}\left[\psi_{0,k},\, \psi_{1,k},\, \ldots,\, \psi_{2^3-1,k}\right]^T \pmod{m_k} = \\ &= [0\ 7\ 6\ 8\ 5\ 0\ 2\ 9]^T, \\ \mathbf{Y} &= \mathbf{CRT}^2_{k=1}\left[\phi_{0,k},\, \phi_{1,k},\, \ldots,\, \phi_{2^3-1,k}\right]^T \pmod{m_k} = \\ &= [0\ 7\ 6\ 1\ 5\ 2\ 3\ 7]^T. \end{aligned}$$

Аппаратурная реализация формулы (7.2) требует использования вычитателя, функционирующего по модулю m_2, для вычисления разности $|\phi_1 - \phi_2|_{m_2}$, умножителя полученной разности на константу δ по модулю m_2, позиционного умножителя на модуль m_1 и позиционного сумматора для суммирования ϕ_1 [4].

7.2 Применение специальных модулей

Как правило для представления вычетов в модулярной арифметике используется двоичная система счисления. Это обстоятельство может быть использовано для упрощения как программной, так и аппаратной реализаций алгоритмов восстановления позиционной формы числа. В качестве модулей удобно использование чисел $2^i \pm 1$, $D_i = 2^i$ и др. [82, 163, 164, 195].

Поскольку модуль вида 2^i имеет в своем каноническом разложении единственный простой множитель 2, он не может быть использован в качестве числового диапазона модулярной арифметики. Пусть $m_1 = 2^i - 1$ и $m_2 = 2^i + 1$. Тогда $M = m_1 m_2 = 2^{2i} - 1$, а операцию суммирования по модулю $M = 2^{2i} - 1$ — числовому

диапазону модулярной арифметики — можно свести к суммированию дополнительного кода. Это дает наиболее простое решение задачи восстановления позиционной формы числа в двухмодульной арифметике в рамках классической теоремы об остатках.

Теперь рассмотрим формулу (7.2). Сократим δ за счет применения модулей, обладающих свойством $m_2 - m_1 = 1$. Далее, используя модули D_i и $2^i + 1$, получим

$$A = D_i \left|\phi_1 - \phi_2\right|_{2^i+1} + \phi_1. \tag{7.6}$$

Вычеты ϕ_1, ϕ_2 и результат восстановления числа A представлены в системе счисления с основанием 2:

$$\begin{aligned}
\phi_1 &= \left(a^{(1)}_{\lceil \log_2 m_1 \rceil}\, a^{(1)}_{\lceil \log_2 m_1 \rceil - 1} \ldots a^{(1)}_1\right)_2; \\
\phi_2 &= \left(a^{(2)}_{\lceil \log_2 m_2 \rceil}\, a^{(2)}_{\lceil \log_2 m_2 \rceil - 1} \ldots a^{(2)}_1\right)_2; \\
A &= \left(a^{(3)}_{\lceil \log_2 M \rceil}\, a^{(3)}_{\lceil \log_2 M \rceil - 1} \ldots a^{(3)}_1\right)_2,
\end{aligned}$$

где $a^{(i)}_j$ — двоичные цифры.

Аппаратурная реализация формулы (7.6) содержит один вычитатель $\sum^-$ разрядности $\lceil \log_2 m_2 \rceil$, функционирующий по модулю $2^i + 1$ (7.2). Поскольку умножение числа, представленного в двоичной системе счисления, на D_i реализуется путем сдвига представления числа на i разрядов в сторону старших разрядов, умножение разности $|\phi_1 - \phi_2|_{2^i+1}$, получаемой на выходе $\sum^-$, на модуль D_i и последующее суммирование результата умножения с ϕ_1 выполняется соответствующим распределением старших и младших разрядов на выходе устройства.

Если вычитатель $\sum^-$ имеет табличную реализацию, то формулу (7.6) можно усилить не усложняя устройства:

$$A = D_i \left|\delta \left|\phi_1 - \phi_2\right|_m\right|_m + \phi_1, \tag{7.7}$$

где

$$\delta = \left|(m - D_i)^{-1}\right|_m;$$

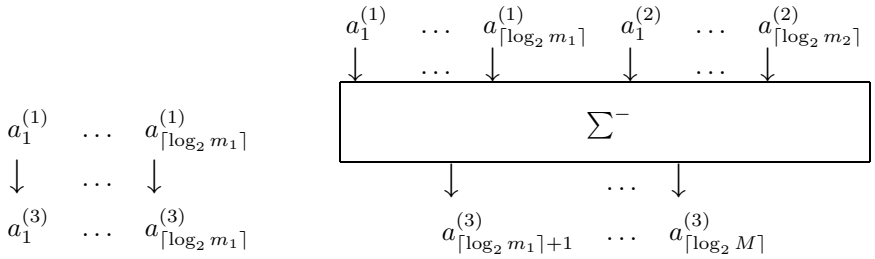


Рис. 7.2: Схема вычисления позиционной формы числа по двум остаткам в соответствии с теоремой 7.1

m — нечетный модуль, взаимно простой с D_i и больший D_i.

Таблица вычитателя $\sum^-$ строится в соответствии с выражением

$$|\delta\,|\phi_1 - \phi_2|_m|_m\,.$$

Пример 7.2
Пусть арифметика задана модулями D_4 и $m = 21$. Тогда

$$\delta = \left|(21-16)^{-1}\right|_{21} = \left|5^5\right|_{21} = 17.$$

Даны вычеты $\phi_1 = 13$ и $\phi_2 = 18$, тогда в соответствии с (7.7) получим

$$A = 16\,|17\,|13-18|_{21}|_{21} + 13 = 333.$$

Действительно, $|333|_{16} = 13$, $|333|_{21} = 18$.

7.3 Обобщения для произвольного количества модулей

Теорема 7.2
Целое неотрицательное число $Y < m = \prod_{k=1}^{v} m_k$, представленное вычетами $\phi_1, \phi_2, \ldots, \phi_v$ по системе попарно простых модулей $m_1 < m_2 < \ldots < m_v$ может быть однозначно восста-

новлено посредством рекурсии

$$\begin{cases} h_1 = \phi_2, \\ h_2 = m_1 \left| \delta_1 \left| \phi_1 - h_1 \right|_{M_1} \right|_{M_1} + \phi_1, \\ h_3 = m_3 \left| \delta_2 \left| \phi_3 - h_2 \right|_{M_2} \right|_{M_2} + \phi_3, \\ h_4 = m_4 \left| \delta_3 \left| \phi_4 - h_3 \right|_{M_3} \right|_{M_3} + \phi_4, \\ \vdots \\ h_v = Y = m_v \left| \delta_{v-1} \left| \phi_v - h_{v-1} \right|_{M_{v-1}} \right|_{M_{v-1}} + \phi_v, \end{cases} \tag{7.8}$$

где

$$\begin{aligned} M_1 &= m_2; \\ M_i &= \prod_{j=1}^{i} m_j; \\ \delta_1 &= \left| (M_1 - m_1)^{-1} \right|_{M_1}; \\ \delta_i &= \left| (M_i - m_{i+1})^{-1} \right|_{M_i}. \end{aligned}$$

Доказательство справедливости (7.8) основано на обобщении теоремы 7.2 путем поэтапного перехода к составным модулям и укрупнения числового диапазона.

Для получения Y с помощью (7.8) потребуется v шагов.

Преимуществом (7.8) по отношению к теореме А.28 является ограничение промежуточных результатов вычислений верхней границей M_v.

При обеспечении возможности распараллеливания вычислений возможен вариант построения алгоритма восстановления позиционной формы числа, основанный на теореме 7.2, требующий $\lceil \log_2 v \rceil$ шагов преобразования [104, 105, 107]. При этом используется принцип рекурсивного сдваивания.

Например, число A представлено вычетами ϕ_1, ϕ_2, ϕ_3, ϕ_4 по попарно взаимно простым и упорядочинным модулям $m_1 < m_2 < m_3 < m_4$.

Тогда алгоритм восстановления числа A состоит из двух шагов (рис. 7.3, здесь $\phi_{11} = |A|_{m_1 m_2}$ и $\phi_{12} = |A|_{m_3 m_4}$).

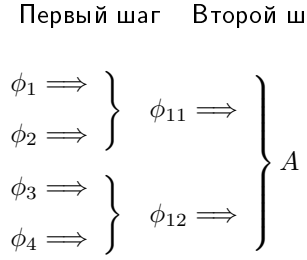


Рис. 7.3: Схема вычисления позиционной формы числа по четырем остаткам с упорядоченными номерами модулей

Алгоритм 7.1

Шаг 1. *В соответствии с (7.2) имеем*

$$\begin{aligned} \phi_{11} &= m_1 \left| \delta_{11} \left| \phi_1 - \phi_2 \right|_{m_2} \right|_{m_2} + \phi_1; \\ \phi_{12} &= m_3 \left| \delta_{12} \left| \phi_3 - \phi_4 \right|_{m_4} \right|_{m_4} + \phi_3, \end{aligned}$$

где

$$\begin{aligned} \delta_{11} &= \left| (m_2 - m_1)^{-1} \right|_{m_2}; \\ \delta_{12} &= \left| (m_4 - m_3)^{-1} \right|_{m_4}. \end{aligned}$$

Шаг 2.

$$A = m_1 m_2 \left| \delta \left| \phi_{11} - \phi_{12} \right|_{m_3 m_4} \right|_{m_3 m_4} + \phi_{11},$$

где

$$\delta = \left| (m_3 m_4 - m_1 m_2)^{-1} \right|_{m_3 m_4}.$$

Номера модулей в рассмотренной схеме и алгоритме восстановления позиционной формы числа могут быть изменены. Для понимания сказанного рассмотрим частный случай, демонстрируемый с помощью рис. 7.4. При этом полагаем, что выполняются условие: $m_1 m_4 > m_2 m_3$.

Такой способ вычисления полезен для выравнивания величины и получения необходимых свойств промежуточных укрупненных модулей, образованных произведениями модулей предыдущих ступеней. При этом алгоритм восстановления будет выглядеть следующим образом.

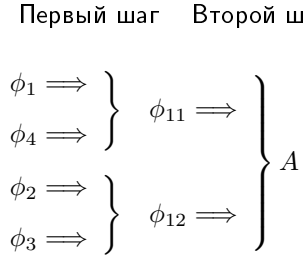


Рис. 7.4: Схема вычисления позиционной формы числа по четырем остаткам с переупорядочиванием номеров модулей

Алгоритм 7.2

Шаг 1. *В соответствии с (7.2) имеем*

$$\begin{aligned} \phi_{11} &= m_1 \left| \delta_{11} \left| \phi_1 - \phi_4 \right|_{m_4} \right|_{m_4} + \phi_1; \\ \phi_{12} &= m_2 \left| \delta_{12} \left| \phi_2 - \phi_3 \right|_{m_3} \right|_{m_3} + \phi_2, \end{aligned}$$

где

$$\begin{aligned} \delta_{11} &= \left| (m_4 - m_1)^{-1} \right|_{m_4}; \\ \delta_{12} &= \left| (m_3 - m_2)^{-1} \right|_{m_3}. \end{aligned}$$

Шаг 2.

$$A = m_1 m_4 \left| \delta \left| \phi_{11} - \phi_{12} \right|_{m_2 m_3} \right|_{m_2 m_3} + \phi_{11},$$

где

$$\delta = \left| (m_2 m_3 - m_1 m_4)^{-1} \right|_{m_2 m_3}.$$

Вычеты ϕ_{11} и ϕ_{12} могут быть получены и любым другим методом, например в соответствии с Китайской теоремой об остатках или прямой дешифрацией.

Формула (7.2) впервые была рассмотрена в [4]. Ее обобщение (7.8) для произвольного количества модулей было получено в [8]. Метод рекурсивного сдваивания и метод применения специальных модулей опубликован в [107, 154]. Случаи применения модулей, обладающих свойствами $m_2 - m_1 = 1$ и $m_2 > m_1 = 2^i$ рассмотрены соответственно в [85] и [86].

Похожие на приведенные выше результаты были позднее опубликованы в [80]. Так, для восстановления позиционной формы числа по двум модулям предлагается формула [80]

$$A = m_1\zeta \left|\phi_2 - \phi_1\right|_{m_2} + \phi_1, \tag{7.9}$$

где $\zeta = \left|m_1^{-1}\right|_{m_2}$.

Обобщение (7.9) для произвольного количества модулей дает, согласно [80], рекурсию:

$$\begin{cases} h_1 = \phi_1, \\ h_2 = M_2\zeta_2 \left|\phi_2 - h_1\right|_{m_2} + \phi_1, \\ h_3 = M_3\zeta_3 \left|\phi_3 - h_2\right|_{m_3} + \phi_2, \\ h_4 = M_4\zeta_4 \left|\phi_4 - h_3\right|_{m_4} + \phi_3, \\ \vdots \\ h_v = Y = M_v\zeta_v \left|\phi_v - h_{v-1}\right|_{m_v} + \phi_{v-1}, \end{cases} \tag{7.10}$$

где $M_i = \prod_{j=1}^{i} m_j$; $\zeta_i = \left|M_i^{-1}\right|_{m_i}$.

Использование схемы в соответствии с рис. 7.3 и модификация алгоритма 7.1 с учетом (7.9) даст следующий результат.

Алгоритм 7.3

Шаг 1. *В соответствии с (7.9) имеем*

$$\begin{aligned} \phi_{11} &= m_1\zeta_{11} \left|\phi_2 - \phi_1\right|_{m_2} + \phi_2; \\ \phi_{12} &= m_3\zeta_{12} \left|\phi_4 - \phi_3\right|_{m_4} + \phi_4, \end{aligned}$$

где

$$\begin{aligned} \zeta_{11} &= \left|m_1^{-1}\right|_{m_2} \\ \zeta_{12} &= \left|m_3^{-1}\right|_{m_4}. \end{aligned}$$

Шаг 2.

$$A = m_1m_2\zeta \left|\phi_{12} - \phi_{11}\right|_{m_3m_4} + \phi_{12},$$

где

$$\zeta = \left|(m_1m_2)^{-1}\right|_{m_3m_4}.$$

Для схемы, представленной на рис. 7.4 алгоритм 7.3 примет вид.

Алгоритм 7.4

Шаг 1. *В соответствии с (7.9) имеем*

$$\begin{aligned}\phi_{11} &= m_1\zeta_{11}\left|\phi_4-\phi_1\right|_{m_4}+\phi_4;\\ \phi_{12} &= m_2\zeta_{12}\left|\phi_3-\phi_2\right|_{m_3}+\phi_3,\end{aligned}$$

где

$$\begin{aligned}\zeta_{11} &= \left|m_1^{-1}\right|_{m_4}\\ \zeta_{12} &= \left|m_2^{-1}\right|_{m_3}.\end{aligned}$$

Шаг 2.

$$A = m_1m_4\zeta\left|\phi_{12}-\phi_{11}\right|_{m_2m_3}+\phi_{12},$$

где

$$\zeta = \left|(m_1m_4)^{-1}\right|_{m_2m_3}.$$

Рассмотренные алгоритмы 7.2 и 7.4 могут быть обобщены для произвольного количества модулей и ступеней преобразования. При этом сортировка исходных и промежуточных модулей может выполняться не только на первой, но и на любой последующей ступени преобразования.

7.4 Использование полиадической системы счисления

Представление числа A с помощью полиадической системы счисления имеет вид:

$$A = z_vq_{v-1}\ldots q_2q_1+\ldots+z_2q_1+z_1, \tag{7.11}$$

где $0 \le z_i < q_i$ $(i = 1, 2, \ldots, v)$; $q_1, q_2, \ldots, q_v$ — основания системы счисления (поэтому ее еще называют системой счисления со смешанными основаниями).

Если в (7.11) выбрать такие основания, что $m_i = q_i$, где $i = 1, 2, \ldots, v$, то представление

$$A = z_v m_{v-1} \ldots m_2 m_1 + \ldots + z_2 m_1 + z_1, \tag{7.12}$$

где $0 \leq z_i < m_i$ $(i = 1, 2, \ldots, v)$, может быть использовано для восстановления позиционной формы числа [16]. При этом используется схема вычислений, представленная на рис 7.5.

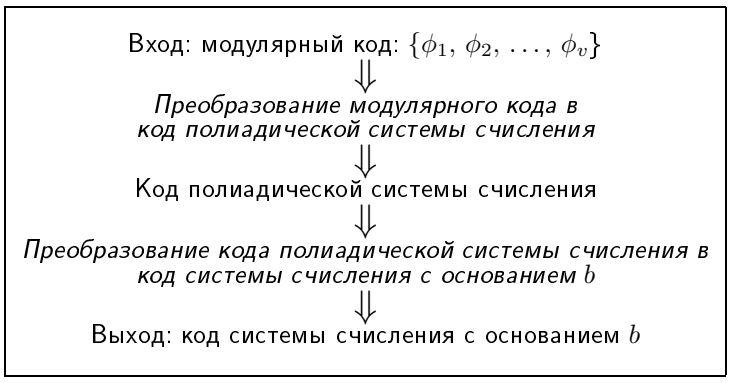


Рис. 7.5: Принцип использования полиадической системы счисления для восстановления позиционной формы числа в заданной системе счисления

Вычислить разрядные цифры $z_1, z_2, \ldots, z_v$ полиадической системы счисления можно, например, с помощью рекурсии [80]:

$$\begin{cases} z_1 = \phi_1, \\ z_2 = \zeta_2 \left|\phi_2 - z_1\right|_{m_2}, \\ z_3 = \zeta_3 \left|\phi_3 - (M_2 z_2 + z_1)\right|_{m_3}, \\ \vdots \\ z_v = \zeta_v \left|\phi_v - (M_{v-1} z_{v-1} + \ldots + M_2 z_2 + z_1)\right|_{m_v}, \end{cases}$$

где числа ζ_i и M_i $(i = 1, 2, \ldots, v)$ взяты из формулы 7.10.

Так как представления весов полиадической системы счисления с помощью b-ичной системы счисления — известные константы (например $b = 2$):

$$m_1 \;=\; (a_{g-1}^{(1)} \ldots a_1^{(1)} a_0^{(1)})_2,$$

$$
\begin{aligned}
m_1 m_2 &= (a_{g-1}^{(2)} \ldots a_1^{(2)} a_0^{(2)})_2, \\
&\vdots \\
m_1 m_2 \ldots m_{v-1} &= (a_{g-1}^{(v-1)} \ldots a_1^{(v-1)} a_0^{(v-1)})_2,
\end{aligned}
$$

то вычисление (7.12) с представлением цифр z_i ($i = 1, 2, \ldots, v$) и весов m_1, $m_1 m_2$, $\ldots$, $m_1 m_2 \ldots m_{v-1}$ в итоговой системе счисления с основанием b даст результат в системе счисления с основанием b. Здесь следует указать, что разрядные цифры итогового представления в системе счисления с основанием b, могут формироваться синхронно (с запаздыванием на одну цифру) с формированием разрядных цифр полиадической системы счисления.

Помимо сказанного, полиадическая система счисления весьма удобна для выполнения немодульных операций в собственном (полиадическом) представлении. Однако эти возможности выходят за рамки обсуждаемой темы.

7.5 Выводы

- Предложены варианты Китайской теоремы об остатках для случаев применения двух и произвольного количества модулей, позволяющие по сравнению с классической формулой существенно ограничить величину результатов промежуточных вычислений.
- Получено два варианта построения алгоритма восстановления позиционной формы числа для произвольного количества модулей, позволяющие получить соответственно v (количество модулей) и $\lceil \log_2 v \rceil$ шагов преобразования.
- Предложен метод упрощения реализации формулы для восстановления позиционной формы числа, основанный на применении специальных модулей.
- Равномерность величины промежуточных модулей алгоритма восстановления позиционной формы числа может быть обеспечена переупорядочиванием номеров исходных и промежуточных модулей.

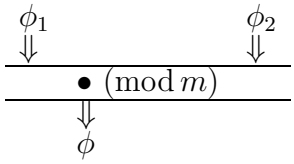
Глава 8

Оператор s-арного модулярного суммирования

Операция s-арного (группового) модулярного суммирования является одной из основных при реализации модулярных форм арифметических полиномов. Однако традиционные методы реализации оператора группового суммирования по модулю m не ориентированы на обработку больших массивов числовых данных, образуемых при реализации интенсивных логических вычислений[1].

8.1 Традиционный „горизонтальный“ метод s-арного суммирования

Существующие алгоритмы модулярной арифметики ориентированны на реализацию бинарных операций $\phi = |\phi_1 \bullet \phi_2|_m$ (символ $\bullet$ означает модулярную операцию — сложение, вычитание, умножение):



s-арной модульной операцией мы будем называть операцию

[1] Оригинальные решения, использованные в этой главе, опубликованы в [7, 10, 110, 114, 124].

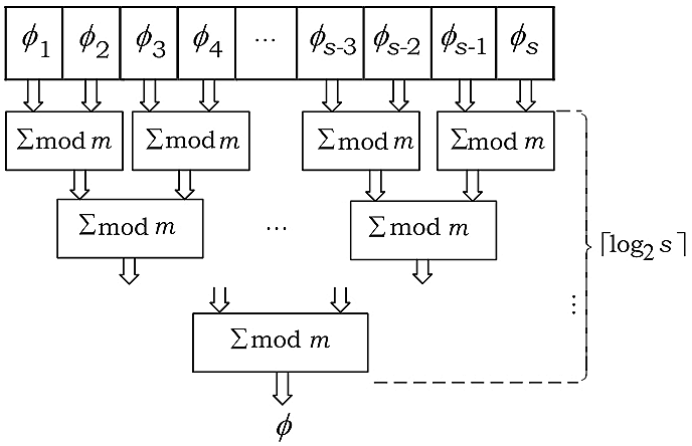
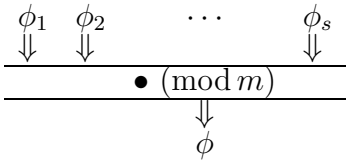


Рис. 8.1: s-арный параллельный сумматор по модулю m, использующий традиционный „горизонтальный“ метод

$\phi = |\phi_1 \bullet \phi_2 \bullet \ldots \bullet \phi_s|_m$ вида:



Будем, также, считать, что вычеты ϕ_1, ϕ_2, ..., ϕ_s представлены с помощью двоичной системы счисления

$$\phi_j = (a_{g-1,\,j} \ \ldots \ a_{1,\,j}\, a_{0,\,j})_2 = a_{g-1,\,j} 2^{g-1} + \ldots + a_{1,\,j} 2^1 + a_{0,\,j},$$

где $a_{i,\,j} \in \{0,\, 1\}$, $i = 0,\, 1,\, \ldots,\, g-1$; $j = 1,\, 2,\, \ldots,\, s$, или в унитарном коде. Здесь значение модуля m и разрядность g связаны отношением $g = \lceil \log_2 m \rceil$.

Традиционный s-арный сумматор по модулю m параллельного типа имеет древовидную структуру с $\lceil \log_2 s \rceil$ уровнями и $s-1$ бинарными (двухвходовыми) сумматорами $(\Sigma \bmod m)$ по модулю m (рис. 8.1).

Время задержки такого сумматора составляет

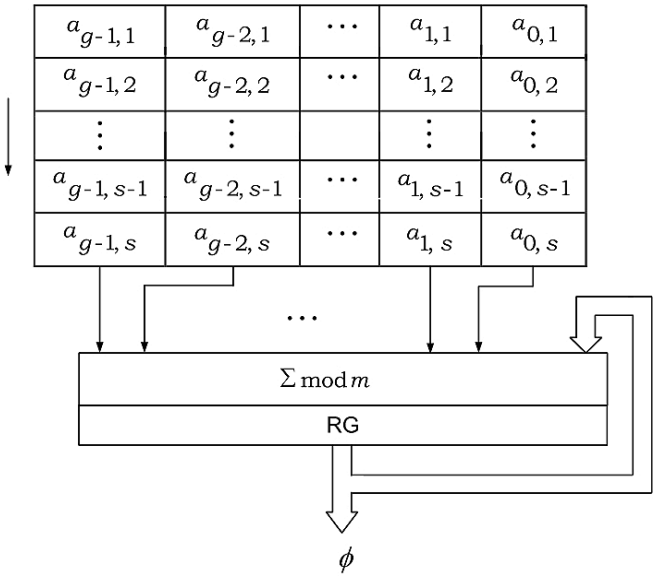


Рис. 8.2: s-арный поразрядно-групповой сумматор по модулю m, использующий традиционный „горизонтальный“ метод

$T_\Sigma = \tau_\Sigma \lceil \log_2 s \rceil$, где τ_Σ — время задержки бинарного сумматора по модулю m.

Традиционный алгоритм реализации s-арного суммирования по модулю m будем называть „горизонтальным“.

Традиционный s-арный сумматор поразрядно-группового действия состоит из параллельно-последовательного регистра сдвига и накапливающего бинарного сумматора $(\Sigma \bmod m)$ по модулю m (рис. 8.2).

Количество синхроимпульсов, необходимых для выполнения операции $\phi = \left|\sum_{j=1}^{s} \phi_j\right|_m$, равно количеству слагаемых — s и всегда одинаково.

8.2 „Вертикальный“ метод s-арного суммирования

Рассмотрим алгоритм s-арного суммирования по модулю m, в котором вместо чисел ϕ_j ($j = 1, 2, \ldots, s$), использованы разрядные цифры $a_{i,1}, a_{i,2}, \ldots, a_{i,s}$ ($i = 0, 1, \ldots, g-1$).

Алгоритм 8.1

Шаг 1. *Выполнить g операций счета количества единиц по модулю m:*

$$\xi_{g-1} = \left| \sum_{j=1}^{s} a_{g-1,j} \right|_m , \ \ldots , \ \xi_1 = \left| \sum_{j=1}^{s} a_{1,j} \right|_m , \ \xi_0 = \left| \sum_{j=1}^{s} a_{0,j} \right|_m .$$

Шаг 2. *Выполнить модулярные умножения значений ξ_i на веса 2^i ($i = 0, 1, \ldots, g-1$):*

$$\widehat{\xi}_{g-1} = \left| \xi_{g-1} 2^{g-1} \right|_m , \ \ldots , \ \widehat{\xi}_1 = |\xi_1 2|_m , \ \widehat{\xi}_0 = |\xi_0|_m .$$

Шаг 3. *Вычислить результат s-арного суммирования:*

$$\phi = \left| \sum_{j=0}^{g-1} \widehat{\xi}_j \right|_m .$$

Структура алгоритма 8.1 поясняется с помощью рис. 8.3.

Устройство для параллельного выполнения алгоритма 8.1 (рис. 8.4) содержит регистр хранения двоичных разрядных цифр

$$\begin{array}{llll} a_{0,1}, & a_{0,2}, & \ldots, & a_{0,s}; \\ a_{1,1}, & a_{1,2}, & \ldots, & a_{1,s}; \\ \vdots & \vdots & & \vdots \\ a_{g-1,1}, & a_{g-1,2}, & \ldots, & a_{g-1,s}, \end{array}$$

g параллельных счетчиков единиц (ST-$g \bmod m$, …, ST-1 $\bmod m$) по модулю m и $g-1$ бинарных сумматоров ($\Sigma \bmod m$) по модулю m. Для обычно используемых значений модуля $2 \le m < 128$

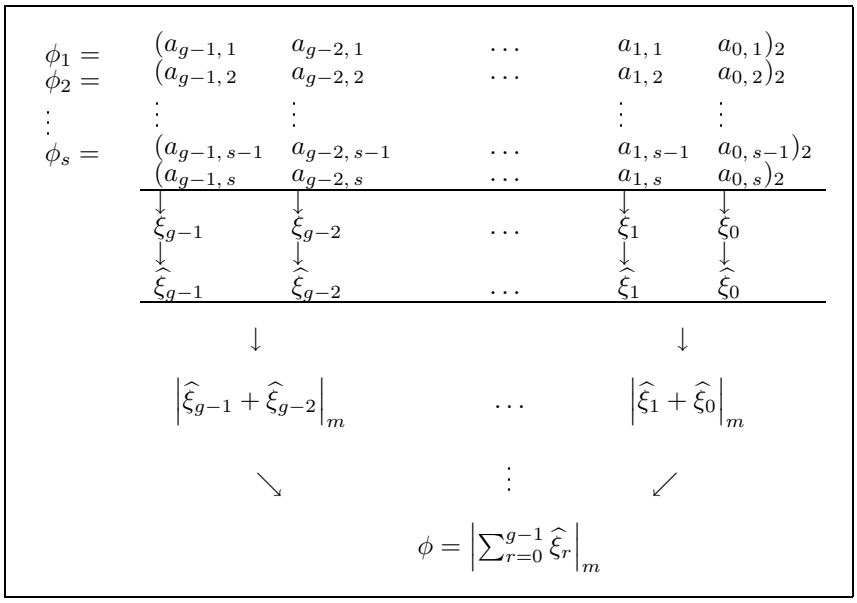


Рис. 8.3: Пояснения к алгоритму 8.1

дерево бинарных сумматоров будет иметь 1—3 уровня. Основное устройство рассматриваемого сумматора — счетчик единиц (ST-$i \bmod m$) по модулю m.

Пример 8.1
Сложить 39 операндов (табл. 8.1) по модулю 13. *Принцип сложения и результат представлены на рис. 8.5.*

Для построения s-арного модулярного сумматора поразрядно-группового действия используем схему Горнера:

$$\phi = \left|\xi_{g-1}2^{g-1} + \xi_{g-2}2^{g-2} + \ldots + \xi_1 2^1 + \xi_0\right|_m =$$
$$= \Big|\Big|\Big(\ldots\Big|\Big(\Big|\big(\big|\xi_{g-1}2\big|_m + \xi_{g-2}\big)\,2\Big|_m + \xi_{g-3}\Big)\,2\Big|_m + \ldots$$
$$\ldots + \xi_1\big)\,2\big|_m + \xi_0\Big|_m,$$

где числа ξ_{g-1}, …, ξ_1, ξ_0 взяты из алгоритма 8.1.

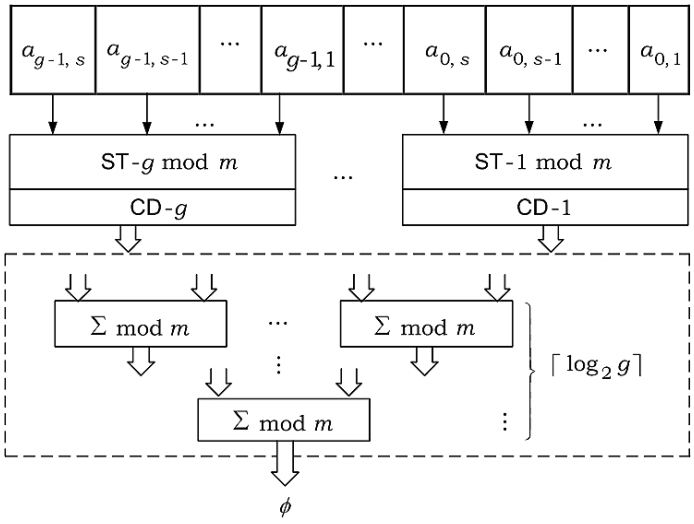


Рис. 8.4: Сумматор по модулю m поразрядно-группового действия, использующий „вертикальный“ метод

Устройство, реализующее эту формулу представлено на рис. 8.6 и содержит параллельно-последовательный регистр сдвига, счетчик единиц (ST $\bmod m$) по модулю m выходной результат которого, представлен в унитарном коде (шифратор — CD не используется), бинарный накапливающий сумматор $\Sigma \bmod m$ по модулю m и умножитель на число 2 по модулю m, функционирующие в унитарном коде. Последний такт работы устройства используется для сложения. Необходимость использования счетчика единиц с представлением результата в унитарном коде объясняется простотой реализации в этом случае умножителя на константу по модулю m.

Пример такого умножителя на число 2 по модулю 5 представлен на рис. 8.7. Если используется счетчик единиц с выводом результата в двоичной системе счисления, то перед входом умножителя необходимо установить дешифратор и использовать сумматор по модулю, функционирующий в двоичной системе счисления.

Таблица 8.1: Исходные данные к примеру 8.1

8	8	10
0	11	1
11	10	8
2	7	8
5	8	5
8	9	8
8	1	0
7	6	4
7	10	8
1	0	10
4	0	7
0	7	5
5	2	8

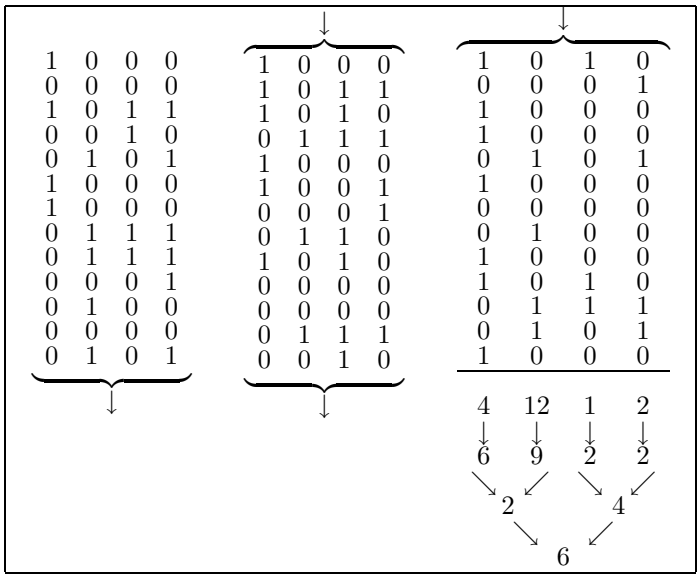


Рис. 8.5: Численный пример реализации оператора группового суммирования 39 операндов по модулю 13

8.3 „Вертикальный“ метод s-арного преобразования

Преобразование позиционного представления числа в вычет (вычеты) по модулю (модулям) применяется при использовании

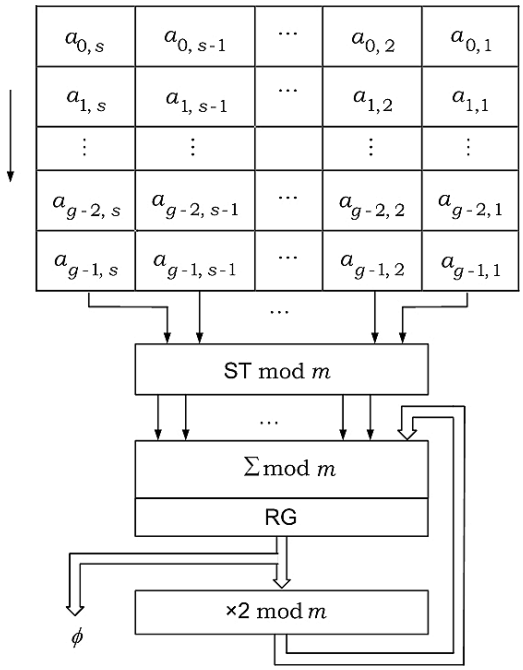


Рис. 8.6: Сумматор по модулю m поразрядно-группового действия, использующий схему Горнера

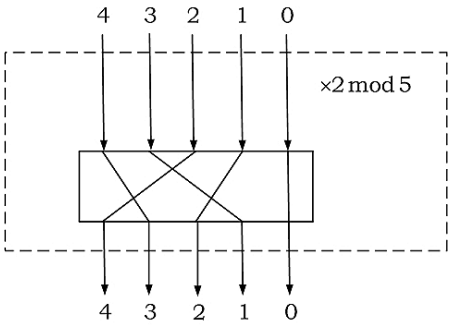


Рис. 8.7: Структура умножителя на число 2 по модулю 5

многомодульной арифметики вычетов, основанной на Китайской теореме об остатках. Рассмотрим оператор следующего вида:

$$\phi = \left| \sum_{j=1}^{z} Y_j \right|_m ,$$

где

$$\begin{aligned}
Y_1 &= b_{t-1,\,1} 2^{t-1} + b_{t-2,\,1} 2^{t-2} + \ldots + b_{1,\,1} 2^1 + b_{0,\,1}, \\
Y_2 &= b_{t-1,\,2} 2^{t-1} + b_{t-2,\,2} 2^{t-2} + \ldots + b_{1,\,2} 2^1 + b_{0,\,2}, \\
&\vdots \\
Y_{z-1} &= b_{t-1,\,z-1} 2^{t-1} + b_{t-2,\,z-1} 2^{t-2} + \ldots + b_{1,\,z-1} 2^1 + b_{0,\,z-1}, \\
Y_z &= b_{t-1,\,z} 2^{t-1} + b_{t-2,\,z} 2^{t-2} + \ldots + b_{1,\,z} 2^1 + b_{0,\,z};
\end{aligned}$$

$b_{i,\,j} \in \{0,\, 1\}$ $(i = 0,\, 1,\, \ldots,\, t-1;\ j = 1,\, 2,\, \ldots,\, z)$.

Числа $Y_1,\, Y_2,\, \ldots,\, Y_z$ удовлетворяют условию:

$$0 \le \sum_{j=1}^{z} Y_j < M = m_1\, m_2\, \ldots\, m_v,$$

здесь $m_1\, m_2\, \ldots\, m_v$ — набор модулей.

Алгоритм 8.2

Шаг 1. *Выполнить t операций счета количества единиц по модулю m:*

$$\beta_{t-1} = \left| \sum_{j=1}^{z} b_{t-1,\,j} \right|_m ,\ \ldots,\ \beta_1 = \left| \sum_{j=1}^{z} a_{1,\,j} \right|_m ,\ \beta_0 = \left| \sum_{j=1}^{z} a_{0,\,j} \right|_m .$$

Шаг 2. *Выполнить модулярные умножения значений β_i на веса 2^i $(i = 0,\, 1,\, \ldots,\, t-1)$:*

$$\widehat{\beta}_{t-1} = \left| \beta_{t-1} 2^{t-1} \right|_m ,\ \ldots,\ \widehat{\beta}_1 = \left| \beta_1 2 \right|_m ,\ \widehat{\beta}_0 = \left| \beta_0 \right|_m .$$

Шаг 3. *Присвоить*

$$\widehat{\beta}_0 = \phi_0,\ \widehat{\beta}_1 = \phi_1,\ \ldots,\ \widehat{\beta}_{t-1} = \phi_s$$

и выполнить алгоритм 8.1.

Структура алгоритма 8.2 поясняется с помощью рис. 8.8.

Блок-схема параллельного суммирующего преобразователя, реализующего алгоритм 8.2 представлена на рис. 8.9.

Алгоритм преобразования кода для устройства поразрядно-группового действия определим формулой (схема Горнера):

$$\phi = \left|\beta_{t-1}2^{t-1} + \beta_{t-2}2^{t-2} + \ldots + \beta_1 2^1 + \beta_0\right|_m =$$
$$= \Big|\Big|\big(\ldots\big|\big(\big|\big(\big|\beta_{t-1}2\big|_m + \beta_{t-2}\big)\,2\big|_m + \beta_{t-3}\big)\,2\big|_m + \ldots$$
$$\ldots + \beta_1\big)\,2\big|_m + \beta_0\Big|_m .$$

Структурная схема устройства преобразования поразрядно-группового действия совпадает со схемой сумматора поразрядно-группового действия, представленного на рис. 8.6. Однако размеры последовательно-параллельного регистра сдвига и количество входов счетчиков единиц по модулю будут другими. Для выполнения алгоритма преобразования устройству поразрядно-группового действия потребуется t тактов функционирования.

8.4 s-арное модулярное умножение

Предложенные выше алгоритмы s-арного суммирования и способы их технической реализации могут быть положены в основу алгоритма и его технической реализации s-арного модулярного умножения. Для этого используем свойство А.2 индексов:

$$\left|\operatorname{ind}_g \prod_{j=1}^{s} \phi_j\right|_{\varphi(m)} = \left|\sum_{j=1}^{s} \operatorname{ind}_g \phi_j\right|_{\varphi(m)},$$

где g — первообразный корень по модулю m, $\varphi(m)$ — функция Эйлера от m (определение А.4). При этом алгоритм умножения строится следующим образом.

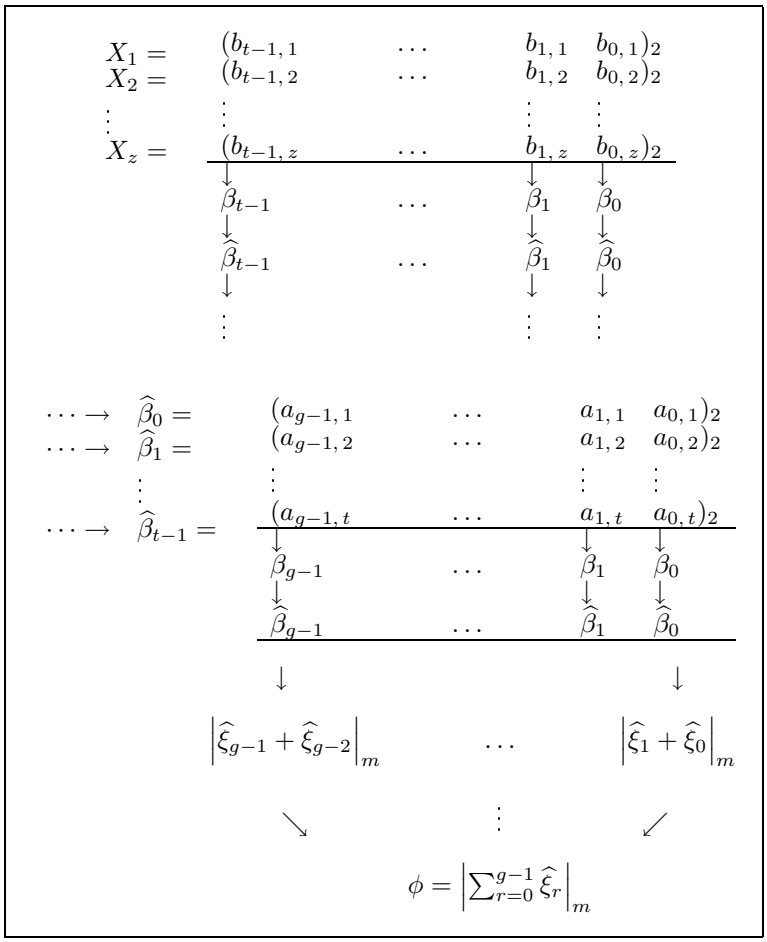


Рис. 8.8: Пояснения к алгоритму 8.2

Алгоритм 8.3

Шаг 1. *Определить значения индексов от* $\phi_1, \phi_2, \dots, \phi_s$ *для модуля* m*:*

$$q_1 = \operatorname{ind}_g \phi_1, \ q_2 = \operatorname{ind}_g \phi_2, \ \dots, \ q_s = \operatorname{ind}_g \phi_s.$$

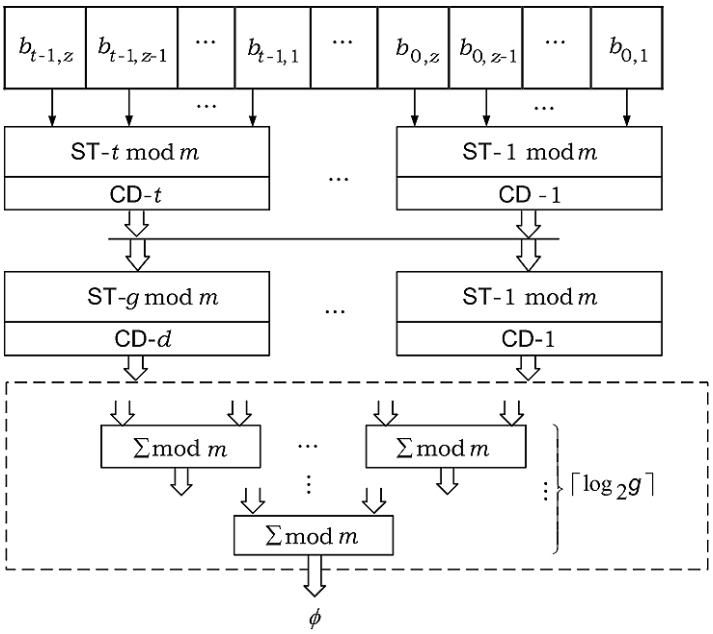


Рис. 8.9: z-арный преобразователь параллельного действия, использующий "вертикальный" метод

Шаг 2. *Вычислить сумму*

$$\phi = \left| \sum_{j=1}^{s} \mathrm{ind}_g \phi_j \right|_{\varphi(m)}$$

в соответствии с алгоритмом 8.1, но для значения модуля $\varphi(m)$.

Шаг 3. *Определить конечный результат* D *умножения по значению индекса* ϕ *по обратной таблице индексов исходя из того, что*

$$g^D \equiv \phi \pmod{m},$$

где g *— первообразный корень по модулю* m.

Алгоритм 8.3 поясняется с помощью рис. 8.10.

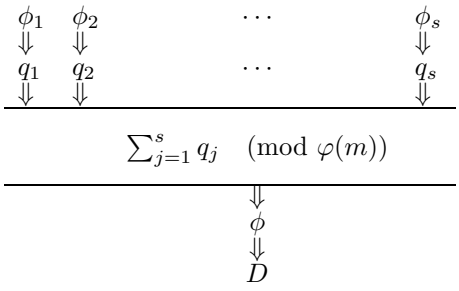


Рис. 8.10: Пояснения к алгоритму 8.3

Пример 8.2
Найти произведение 39 операндов (табл. 8.2) по модулю 13. *Применяя алгоритм 8.3 получим:*

Шаг 1. *С помощью приложения В определяем значения индексов. Результат представлен в таблице 8.1 примера 8.1.*

Шаг 2. *В соответствии с алгоритмом 8.1 вычисляем сумму операндов, находящихся в таблице 8.1, по модулю* $\varphi(13) = 12$ *(в отличие от примера 8.1). Сумма равна 11.*

Шаг 3. *С помощью приложения В находим результат произведения — величину обратную индексу — 6.*

Структурная схема устройства реализации алгоритма 8.3 представлена на рис. 8.11 и содержит регистр хранения вычетов-множителей ϕ_1, ϕ_2, $\ldots$, ϕ_s, запоминающие устройства — ЗУ-1. 1, ЗУ-1. 2, ..., ЗУ-1. s, предназначенные для хранения значений индексов в соответствии с шагом 1 алгоритма 8.3, s-арный сумматор, функционирующий по модулю $\varphi(m)$, выходное ЗУ-2, предназначенное для хранения обратных значений индексов в соответствии с шагом 3 алгоритма 8.3.

8.5 Счетчики единиц по модулю

Основным устройством аппаратурных средств реализации s-арных модулярных операций является параллельный счетчик

Таблица 8.2: Исходные данные к примеру 8.2

9	9	10
1	6	11
6	10	9
4	2	9
7	9	7
9	8	9
9	11	1
2	12	3
2	10	9
11	1	10
3	1	2
1	2	7
7	4	9

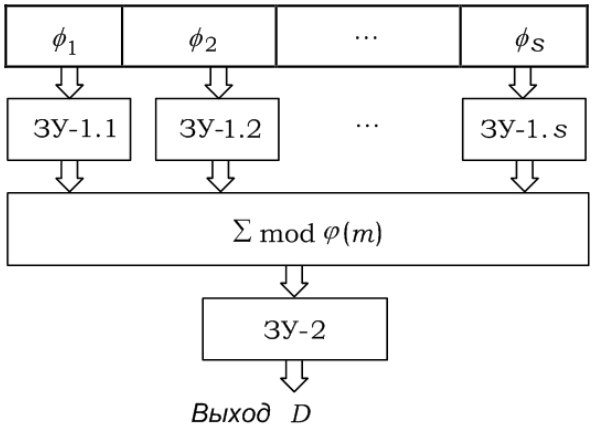


Рис. 8.11: Схема аппаратурной реализации параллельного s-арного умножителя по модулю m

единиц по заданному модулю.

Известно, что в классе схем из функциональных элементов параллельные счетчики реализуются с линейной относительно числа переменных сложностью [2, 40, 63]. При этом схема счетчика содержит не более $(n-1)$ полных сумматоров. Вопросы синтеза параллельных счетчиков подробно исследовались в [149, 150].

„Изящный“ вариант построения устройства для подсчета количества единиц, предложен в описании к патенту 1479596 (Великобритания) [136]. Принцип построения этого устройства поясняется с помощью рис. 8.12.

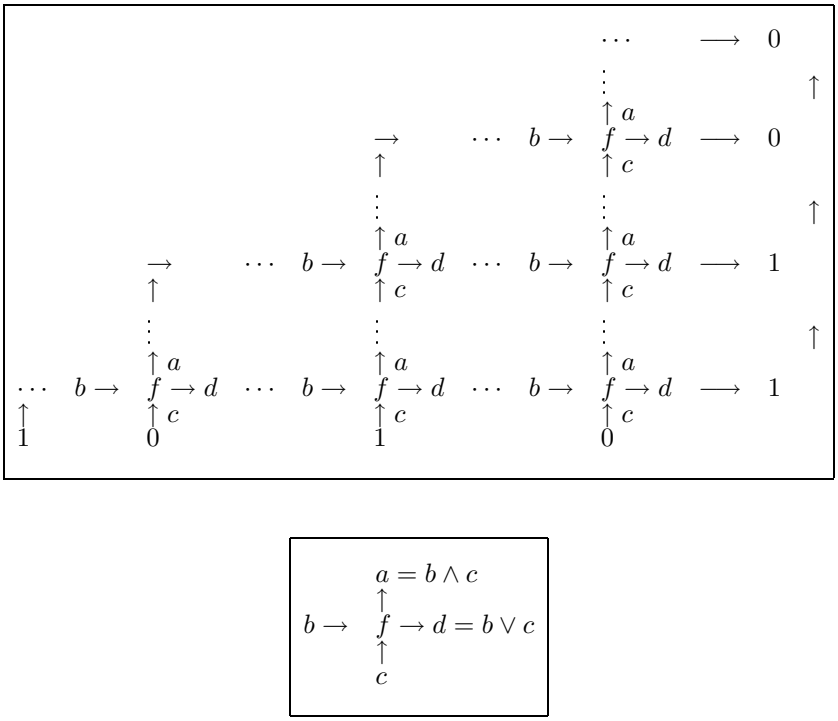


Рис. 8.12: Пояснение принципа построения устройства, для подсчета количества единиц

Для приведения результата на выходах устройства, использующего данный принцип, в код по модулю m можно построить схему, реализующую выражения:

$$f_1 = x_1 \bigwedge_{i=1}^{m} x_i \vee x_{m+1} \bigwedge_{i=m+1}^{2m} x_i \vee \ldots \vee x_{(k-1)m+1} \bigwedge_{i=(k-1)m+1}^{km} x_i;$$

$$f_2 = x_2 \bigwedge_{i=1}^{m} x_i \vee x_{m+2} \bigwedge_{i=m+1}^{2m} x_i \vee \ldots \vee x_{(k-1)m+2} \bigwedge_{i=(k-1)m+1}^{km} x_i;$$

$$\vdots$$

$$f_m = x_m \bigwedge_{i=1}^{m} x_i \vee x_{2m} \bigwedge_{i=m+1}^{2m} x_i \vee \ldots \vee x_{mk} \bigwedge_{i=(k-1)m+1}^{km} x_i,$$

где $x_1, x_2, \ldots, x_{mk}$ — выходы исходного счетчика единиц.

Более экономные решения позволяют получить менее быстродействующие счетчики с памятью, принцип построения которых предложен в [7]. Однако, максимизация уровня быстродействия для этих последовательных схем, обеспечивается использованием асинхронного принципа функционирования.

Наиболее совершенная на наш взгляд методика синтеза параллельных счетчиков единиц, функционирующих в унитарных кодах по модулю, изложена в [78, 79]. В данных работах для уменьшения сложности параллельных счетчиков предлагается использовать два метода: метод декомпозиции [76] и метод факторизации [77], разработанные для элементарных симметрических функций. В силу большой важности этих методов для реализации рассмотренных выше методов s-арного модулярного суммирования остановимся на них более подробно.

1. *Декомпозиционный метод.* Унитарный параллельный код вычета по модулю m количества единиц двоичного кода $A = \{a_1, a_2, \ldots, a_g\}$ определим как код $Y = \left(y_g^{(0)}, y_g^{(1)}, \ldots, y_g^{(m-1)}\right)_1$, удовлетворяющий условию

$$y_g^{(\alpha)}(A) = \begin{cases} 1, & \text{при} \quad \sum_{i=1}^{g} a_i \pmod m = \alpha; \\ 0, & \text{при} \quad \sum_{i=1}^{g} a_i \pmod m \neq \alpha. \end{cases}$$

Согласно [76] унитарный параллельный позиционный код $Y(A)$ может быть представлен композицией элементарных симметрических функций или периодических симметрических

функций:

$$y_g^{(\alpha)}(A) = \bigvee_{i=0}^{l} f_g^{(\alpha+im)}(A) = f_g^{(\alpha+im)}(A), \tag{8.1}$$

где $\alpha = 0, 1, 2, \ldots, m-1$; $f_g^{(\alpha+im)}(A)$ — периодическая симметрическая функция, являющаяся дизъюнкцией элементарных симметрических функций $f_g^{(\alpha)}(A)$, индексы которых принимают значения $\alpha = \alpha, \alpha + m, \ldots, \alpha \lfloor g/m \rfloor m$, причем $\alpha \leq g$.

Любая элементарная симметрическая ФАЛ может быть представлена композицией функций разложения, зависящих от меньшего числа переменных вида [76]

$$f_g^{(\alpha)}(A) = \bigvee_{\{C_r\}} f_{m_1}^{(\alpha_1)}(A_1) f_{m_2}^{(\alpha_2)}(A_2) \ldots f_{m_r}^{(\alpha_r)}(A_r), \tag{8.2}$$

где $f_{m_i}^{(\alpha_i)}(A_i)$ — функции разложения, которые являются элементарными симметрическими функциями; r — параметр разложения; $C_r = \alpha_1, \ldots, \alpha_r$ — r-мерный вектор; α_i — элемент r-мерного вектора C_r, являющегося индексом функции $f_{m_i}^{(\alpha_i)}(A_i)$; $\{C_r\}$ — множество всех r-мерных векторов, удовлетворяющих условию

$$\sum_{i=1}^{r} \alpha_i = \alpha \quad (\alpha_i \geq 0);$$

A_i — множества переменных функций разложения $f_{m_i}^{(\alpha_i)}(A_i)$, удовлетворяющие условиям:

$$A_1 \cup A_2 \cup \ldots \cup A_r = A,$$
$$A_i \cap A_j = \emptyset, \quad \text{при} \quad i \neq j.$$

С учетом разложения (8.2) уравнение (8.1) согласно [79] может быть представлено в виде:

$$y_g^{(\alpha)}(A) = \bigvee_{\{C_r\}} y_{m_1}^{(\alpha_1)}(A_1) y_{m_2}^{(\alpha_2)}(A_2) \ldots y_{m_r}^{(\alpha_r)}(A_r), \tag{8.3}$$

где $\{C_r\}$ — множество r-мерных векторов $C_r = \alpha_1, \ldots, \alpha_r$ элементы, которых удовлетворяют условию:

$$\sum_{t=0}^{r} \alpha_t \pmod m = \alpha. \tag{8.4}$$

Таким образом, согласно [79] схема формирования вычета от количества единиц двоичного кода A по модулю m может быть синтезирована путем последовательной совместной декомпозиции системы функций $Y(A)$, описывающей работу схемы, на основе обобщенного разложения (8.4).

Глубина и сложность схем синтезируемых параллельных счетчиков будут зависеть от параметров разложения r и мощности множеств A_i на каждом шаге декомпозиции.

Схема параллельного счетчика, синтезированная методом разложения по переменным содержит [79]:

$$L\left(Y(g)\right) = \begin{cases} (3g/2+2)(g-1), & \text{при} \quad g < m; \\ (3m+1)(m-2)/2 + 3m(g-m+1), & \text{при} \quad g \geq m \end{cases}$$

элементов И, ИЛИ.

Схема параллельного счетчика, синтезированная методом регулярной детерминированной декомпозиции с параметром $r = 2$ содержит [79]:

$$L\left(Y(g)\right) =$$

$$= \begin{cases} (g+3)(g-1) + g\left\lfloor \log_2 g \right\rfloor - \\ -2^{\left\lfloor \log_2 g \right\rfloor + 1} + 2, & \text{при} \quad g < m, \\ L\left(Y(\lceil g/2 \rceil)\right) + L\left(Y(\lfloor g/2 \rfloor)\right) + \\ +2\left(\lceil g/2 \rceil + 1\right)\left(\lfloor g/2 \rfloor + 1\right) - m, & \text{при} \quad m \leq g \leq 2m-2, \\ L\left(Y(m_1)\right) + L\left(Y(m_2)\right) + 2m^2 - m, & \text{при} \quad g > 2m-2 \end{cases}$$

элементов И, ИЛИ, где m_1, $m_2 \geq m-1$.

2. *Факторизационный метод* [77] может быть применен для синтеза параллельных счетчиков единиц, функционирующих по модулю m, благодаря свойствам рассматриваемых функций,

аналогичным свойствам элементарных симметрических функций [79]:

$$
\begin{aligned}
y_g^{(\alpha)}(A)y_g^{(\beta)}(A) &= \\
&= f_g^{(\alpha+jm)}(A)f_g^{(\beta+jm)}(A) = \\
&= \begin{cases} y_g^{(\alpha)}(A) = f_g^{(\alpha+jm)}(A), & \text{при} \quad \alpha = \beta, \\ 0, & \text{при} \quad \alpha \neq \beta, \end{cases} \\
y_g^{(\alpha)}(A)\overline{y_g^{(\beta)}(A)} &= \\
&= f_g^{(\alpha+jm)}(A)\overline{f_g^{(\beta+jm)}(A)} = \\
&= \begin{cases} y_g^{(\alpha)}(A) = f_g^{(\alpha+jm)}(A), & \text{при} \quad \alpha \neq \beta, \\ 0, & \text{при} \quad \alpha = \beta. \end{cases}
\end{aligned}
$$

Факторизация заключается в преобразовании системы логических уравнений вида (8.3) в соответствии с представлением:

$$
\begin{aligned}
&\bigvee_{\{C_r'\}} y_{m_1}^{(\alpha_1)}(A_1)y_{m_2}^{(\alpha_2)}(A_2)\ldots y_{m_r}^{(\alpha_r)}(A_r) = \\
&= \bigwedge_{i=1}^{t} F_r^{(\gamma_i)}\left(y_{m_1}^{(\beta_i)}(A_1),\, y_{m_2}^{(\beta_i)}(A_2),\, \ldots,\, y_{m_r}^{(\beta_i)}(A_r)\right) = \\
&= \bigwedge_{i=1}^{t} F_r^{(\gamma_i)}(Y_i),
\end{aligned}
$$

где $F_r^{(\alpha)}(A)$ — пороговая равновесная функция r переменных с порогом α, A_r' — множество r-мерных векторов $A_r = \alpha_1, \alpha_2, \ldots, \alpha_r$ одного типа, получаемых перестановкой элементов фиксированного множества индексов $\alpha_1, \alpha_2, \ldots, \alpha_r$,

причем:

$$\sum_{j=1}^{t} \gamma_j \beta_j = \sum_{i=1}^{r} \alpha_i = \alpha.$$

Схема параллельного счетчика, синтезированная методом факторизации ($r = 2$) по переменным содержит [79]:

$$L\left(Y(g)\right) = L\left(Y(m_1)\right) + L\left(Y(m_2)\right) + l_1$$

элементов И, ИЛИ, НЕ, где

$$l_1 = \begin{cases} g^2/4 + g + 2, & \text{при} \quad g < m, \quad g = 2\lceil g/2 \rceil, \\ (g^2+1)/4 + \\ +3g/2 + 1, & \text{при} \quad g < m, \quad g \neq 2\lceil g/2 \rceil, \\ (g^2+1)/4 + \\ +5g/2 - m + 2, & \text{при} \quad m \leq g \leq 2m-2, \quad g \neq 2\lceil g/2 \rceil, \\ m^2 + m, & \text{при} \quad m \leq g \leq 2m-2, \quad g > 2m-2, \end{cases}$$

где $m_1 = g \neq \lceil g/2 \rceil, \quad m_2 = g \neq \lfloor g/2 \rfloor, \quad m_1, m_2 > m.$

Быстродействие синтезируемых параллельных счетчиков определяется глубиной схемы. При этом под глубиной схемы понимается максимальное количество последовательно соединенных логических элементов от входа схемы к ее выходу. Приведем оценки глубины схемы, полученные в [79].

Глубина схемы, синтезированной методом декомпозиции определяется рекуррентным выражением

$$H(Y(g)) = H\left(Y\left(\lceil g/r \rceil\right)\right) + 2.$$

В зависимости от различных случаев декомпозиции могут быть получены и различные значения глубины схемы. Так, в случае применения метода декомпозиции, представляющем собой разложение по переменным, имеет место максимальная глубина схемы, которая зависит от количества переменных g и параметра

разложения r и определяется оценкой:

$$H(Y(g)) =$$
$$= \begin{cases} 2\lfloor g/(r-1)\rfloor + 2, & \text{при} \quad (t+1)(r-1) > g \geq t(r-1)+2 \\ & \text{и} \quad r > 2, \\ 2\lfloor g/(r-1)\rfloor, & \text{при} \quad t(r-1)+1 > g \geq t(r-1) \\ & \text{и} \quad r > 2, \\ 2(g-1), & \text{при} \quad r = 2, \end{cases}$$

где t — целое число.

Схема минимальной глубины будет получена при использовании метода регулярной детерминированной декомпозиции, когда на каждом шаге мощность подмножеств переменных функций разложения одинакова или отличается не более, чем на единицу. Глубина схемы при этом определяется оценкой

$$H(Y(g)) = H\left(Y\left(\lceil g/r\rceil\right)\right) + 2 \approx 2\log_r g.$$

При синтезе схем методом факторизации глубина схемы по сравнению с методом декомпозиции возрастает. Это обусловлено необходимостью реализации системы пороговых равновесных или элементарных симметрических функций. Глубина схемы при этом определяется оценкой

$$H(Y(g)) = H\left(Y\left(\lceil g/r\rceil\right)\right) + r + 1.$$

Для случая факторизации логических уравнений вида (8.3), являющихся результатом регулярной детерминированной декомпозиции, оценка глубины схемы имеет вид

$$H(Y(g)) \approx (r+1)\log_r g.$$

Таким образом наименьшую глубину имеют схемы, получаемые в результате регулярной детерминированной декомпозиции. Увеличение параметра разложения приводит к уменьшению глубины схемы за счет увеличения ее сложности.

8.6 Использование избыточных позиционных представлений чисел

В [71] для ускорения реализации оператора группового суммирования предложено использовать принцип ортогонализации представления коэффициентов (констант) арифметического полинома. Это принцип можно развить далее, используя промежуточные избыточные представления коэффициентов [110, 114].

Пусть коэффициенты $A_0, A_1 \ldots, A_{2^n-1}$ (будем считать их неотрицательными) арифметического полинома

$$D(X) = A_0 v_0 + A_1 v_1 + \cdots + A_{2^n-1} v_{2^n-1}, \tag{8.5}$$

где $v_0 = 1$, $v_1 = x_n$, $\ldots$, $v_{2^n-1} = x_1 x_2 \ldots x_n$, представлены в двоичной системе счисления:

$$A_i = a_{k-1,\,i} 2^{k-1} + a_{k-2,\,i} 2^{k-2} + \cdots + a_{0,\,1}, \tag{8.6}$$

$i = 0, 1, \ldots, 2^n - 1$; $a_{j,\,i} \in \{0, 1\}$ $(j = 0, 1, \ldots, k-1)$ — разрядные цифры.

Подставляя (8.6) в (8.5) получим

$$Y = \sum_{i=0}^{2^n-1} \sum_{j=0}^{k-1} a_{j,\,i} v_i 2^j. \tag{8.7}$$

Принцип параллельного сложения заключается в использовании ортогональной формы представления констант. Пусть слагаемые

$$\begin{aligned} A_1 &= (a_{1,\,k-1}\, a_{1,\,k-2} \ldots a_{1,\,1}\, a_{1,\,0})_2, \\ A_2 &= (a_{2,\,k-1}\, a_{2,\,k-2} \ldots a_{2,\,1}\, a_{2,\,0})_2 \end{aligned}$$

удовлетворяют условию

$$a_{1,\,j} \wedge a_{2,\,j} = 0 \quad (j = 0, 1, \ldots, k-1). \tag{8.8}$$

Тогда сложение $A_1 + A_2$ можно реализовать параллельным логическим сложением

$$a_{k-1} = a_{1,\,k-1} \vee a_{2,\,k-1},$$

$$\begin{aligned} a_{k-2} &= a_{1,\,k-2} \vee a_{2,\,k-2}, \\ &\vdots \\ a_1 &= a_{1,\,1} \vee a_{2,\,1}, \\ a_0 &= a_{1,\,0} \vee a_{2,\,0}. \end{aligned}$$

Аналогично бинарной операции может быть определена и 2^n-арная операция сложения:

$$A = \sum_{i=0}^{2^n-1} A_i. \tag{8.9}$$

Если известно, что операнды A_i $(i = 0,\, 1,\, \ldots,\, 2^n - 1)$ удовлетворяют условию

$$\sum_{i=0}^{2^n-1} a_{i,\,j} = w_j \in \{0,\, 1\} \quad (j = 0,\, 1,\, \ldots,\, k-1), \tag{8.10}$$

где w_j $(j = 0,\, 1,\, \ldots,\, k-1)$ — вес кодовой комбинации (количество единичных символов), то (8.9) можно вычислить следующим образом:

$$\left[\begin{array}{rccccc} A_{n-1} = & a_{n-1,\,k-1} & a_{n-1,\,k-2} & \ldots & a_{n-1,\,1} & a_{n-1,\,0} \\ A_{n-2} = & a_{n-2,\,k-1} & a_{n-2,\,k-2} & \ldots & a_{n-2,\,1} & a_{n-2,\,0} \\ \vdots\ \ \vdots & \vdots & \vdots & & \vdots & \vdots \\ A_1 = & a_{1,\,k-1} & a_{1,\,k-2} & \ldots & a_{1,\,1} & a_{1,\,0} \\ A_0 = & a_{0,\,k-1} & a_{0,\,k-2} & \ldots & a_{0,\,1} & a_{0,\,0} \\ \hline A = & a_{k-1} & a_{k-2} & \ldots & a_1 & a_0 \end{array}\right], \tag{8.11}$$

где a_j $(j = 0,\, 1,\, \ldots,\, k-1)$ — цифры представления A, полученные с помощью параллельного логического сложения

$$a_j = \bigvee_{i=0}^{2^n-1} a_{i,\,j} \quad (j = 0,\, 1,\, \ldots,\, k-1). \tag{8.12}$$

Соответственно для получения Y выражение (8.11) следует переписать в виде:

$$\left[\begin{array}{rcll} v_{2^n-1}A_{n-1} = & v_{2^n-1}a_{n-1,\,k-1} & \dots & v_{2^n-1}a_{n-1,\,0} \\ v_{2^n-2}A_{n-2} = & v_{2^n-2}a_{n-2,\,k-1} & \dots & v_{2^n-2}a_{n-2,\,0} \\ \vdots & \vdots & & \vdots \\ v_1A_1 = & v_1a_{1,\,k-1} & \dots & v_1a_{1,\,0} \\ v_0A_0 = & v_0a_{0,\,k-1} & \dots & v_0a_{0,\,0} \\ \hline Y = & a_{k-1} & \dots & a_0 \end{array}\right], \qquad (8.13)$$

где $a_j = \bigvee_{i=0}^{2^n-1} v_i a_{i,\,j}$, причем $a_j = w_j$ $(j = 0,\, 1,\, \dots,\, k-1)$.

Устройство преобразования в этом случае может быть реализовано простейшей разводкой проводников на кристалле, либо матрицей ПЗУ, и требует минимальных временных и аппаратурных затрат.

Существенным недостатком этого метода является его ограниченность низкой вероятностью того, что форма кода всех операндов A_i $(j = 0,\, 1,\, \dots,\, 2^n - 1)$ будет удовлетворять условию (8.10).

Представим A_i $(j = 0,\, 1,\, \dots,\, 2^n - 1)$ с помощью избыточной многозначной системы счисления:

$$A_i = \sum_{j=0}^{k-1} a'_{i,\,j} 2^j \quad (j = 0,\, 1,\, \dots,\, 2^n - 1),$$

где $a'_{i,\,j} \in \{-d_2,\, \dots,\, -1,\, 0,\, 1,\, \dots,\, d_1\}$ $(j = 0,\, 1,\, \dots,\, 2^n - 1)$;

$$d_1 = \max\{w_0^{(1)},\, w_1^{(1)},\, \dots,\, w_{k-1}^{(1)}\}, \quad w_j^{(1)} = \sum_{i=0}^{2^n-1} a'_{i,\,j},$$

для $a'_{i,\,j} > 0$ $(j = 0,\, 1,\, \dots,\, k-1)$;

$$d_2 = \min\{w_0^{(2)},\, w_1^{(2)},\, \dots,\, w_{k-1}^{(2)}\}, \quad w_j^{(2)} = \sum_{i=0}^{2^n-1} a'_{i,\,j},$$

для $a'_{i,\,j} < 0$ $(j = 0,\, 1,\, \dots,\, k-1)$.

Матрицу $(a'_{i,j})_{2^n,k}$ представим суммой из $d_1 + \mathrm{abs}(d_2)$ матриц:

$$\left(a'_{i,j}\right)_{2^n,k} = \sum_{s_1=1}^{d_1} \left(a_{i,j}^{(s_1)}\right)_{2^n,k} + \sum_{s_2=1}^{\mathrm{abs}(d_2)} \left(a_{i,j}^{(s_2)}\right)_{2^n,k}, \qquad (8.14)$$

где

$$\begin{aligned} a_{i,j}^{(s_1)} &\in \{0, 1\} \quad (s_1 = 1, \ldots, d_1); \\ a_{i,j}^{(s_2)} &\in \{0, -1\} \quad (s_2 = 1, \ldots, \mathrm{abs}(d_2)). \end{aligned}$$

При этом

$$\begin{cases} w_j^{(s_1)} = \sum\limits_{i=0}^{2^n-1} a_{i,j}^{(s_1)} \in \{0, 1\}, \\ \\ w_j^{(s_2)} = \sum\limits_{i=0}^{2^n-1} a_{i,j}^{(s_2)} \in \{0, -1\}, \end{cases} \qquad (8.15)$$

где $s_1 = 1, \ldots, d_1$; $s_2 = 1, \ldots, \mathrm{abs}(d_2)$; $j = 0, 1, \ldots, k-1$.

Перепишем (8.14) в виде

$$\left(v_i a'_{i,j}\right)_{2^n,k} = \sum_{s_1=1}^{d_1} \left(v_i a_{i,j}^{(s_1)}\right)_{2^n,k} + \sum_{s_2=1}^{\mathrm{abs}(d_2)} \left(v_i a_{i,j}^{(s_2)}\right)_{2^n,k}.$$

Матрицам $\left(v_i a'_{i,j}\right)_{2^n,k}$, $\left(v_i a_{i,j}^{(s_1)}\right)_{2^n,k}$ $(s_1 = 1, \ldots, d_1)$ и $\left(v_i a_{i,j}^{(s_2)}\right)_{2^n,k}$ $(s_2 = 1, \ldots, \mathrm{abs}(d_2))$ поставим в соответствие числа Y, $Y^{(s_1)}$ $(s_1 = 1, \ldots, d_1)$ и $Y^{(s_2)}$ $(s_2 = 1, \ldots, \mathrm{abs}(d_2))$. Причем

$$\begin{aligned} Y &= \sum_{i=0}^{2^n-1} \sum_{j=0}^{k-1} v_i a'_{i,j} 2^j = \sum_{j=0}^{k-1} a_j 2^j; \\ Y^{(s_1)} &= \sum_{i=0}^{2^n-1} \sum_{j=0}^{k-1} v_i a_{i,j}^{(s_1)} 2^j = \sum_{j=0}^{k-1} a_j^{(s_1)} 2^j \quad (s_1 = 1, \ldots, d_1), \end{aligned}$$

где $a_j^{(s_1)} \in \{0, 1\}$ $(s_1 = 1, \ldots, d_1)$,

$$Y^{(s_2)} = \sum_{i=0}^{2^n-1} \sum_{j=0}^{k-1} v_i a_{i,j}^{(s_2)} 2^j = \sum_{j=0}^{k-1} a_j^{(s_2)} 2^j \quad (s_2 = 1, \ldots, \mathrm{abs}(d_2)),$$

где $a_j^{(s_2)} \in \{0, -1\}$ $(s_2 = 1, \ldots, \mathrm{abs}(d_2))$;
цифры $a_{i,j}^{(s_1)}$ и $a_{i,j}^{(s_2)}$ $(i = 0, 1, \ldots, 2^n - 1; \quad j = 0, 1, \ldots, k-1)$ выбираются таким образом, чтобы выполнялось условие

$$Y^{(s_1)} \geq Y^{(s_2)}. \tag{8.16}$$

Если условия (8.15) выполнены, то для вычисления цифр $a_j^{(s_1)}$ $(s_1 = 1, \ldots, d_1)$ и $a_j^{(s_2)}$ $(s_2 = 1, \ldots, \mathrm{abs}(d_2))$ можно воспользоваться ранее рассмотренным методом:

$$a_j^{(s_1)} = \bigvee_{i=0}^{2^n-1} a_{i,j}^{(s_1)} \quad (j = 0, 1, \ldots, k-1;\ s_1 = 1, \ldots, d_1),$$

$$a_j^{(s_2)} = \bigvee_{i=0}^{2^n-1} a_{i,j}^{(s_2)} \quad (j = 0, 1, \ldots, k-1;\ s_2 = 1, \ldots, \mathrm{abs}(d_2)).$$

Таким образом, окончательный результат преобразования определяется выражением

$$\begin{aligned} Y &= \sum_{s_1=1}^{d_1} Y^{(s_1)} - \sum_{s_2=1}^{\mathrm{abs}(d_2)} Y^{(s_2)} = \\ &= \sum_{s_1=1}^{d_1} \sum_{j=1}^{k-1} a_j^{(s_1)} 2^j - \sum_{s_2=1}^{\mathrm{abs}(d_2)} \sum_{j=1}^{k-1} a_j^{(s_2)} 2^j, \end{aligned} \tag{8.17}$$

который в силу (8.16) будет неотрицательным.

Пример 8.3
Необходимо найти сумму коэффициентов полинома:

$$P(X) = 8 + 9x_1 + 2x_2 + 11x_3.$$

Матрица, построенная из двоичных неотрицательных коэффициентов полинома и ее представление в виде суммы матриц, удовлетворяющих заданным требованиям, имеет вид

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \\ + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Для нахождения суммы потребуется выполнить две операции арифметического сложения.

Преобразуем исходную матрицу в матрицу, в которой использованы разрядные цифры избыточной системы счисления, и выполним суммирование

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \to \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} = \\ = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \qquad (8.18)$$

В этом случае потребуется только одна операция арифметического сложения. Устройство для реализации рассматриваемого полинома в соответствии с (8.16) представлено на рис. 8.13.

В общем случае (для произвольного количества членов полинома) структурная схема устройства содержит (рис. 8.14): распределитель разрядных цифр по группам разрядов в соответствии с (8.14) и (8.16), d_1 и $\mathrm{abs}(d_2)$-арные сумматоры Σ^+, выходной вычитатель Σ^-, обеспечивающий выполнение выражения (8.18).

Метод группового сложения, основанный на использовании промежуточной избыточной системы счисления, позволяет

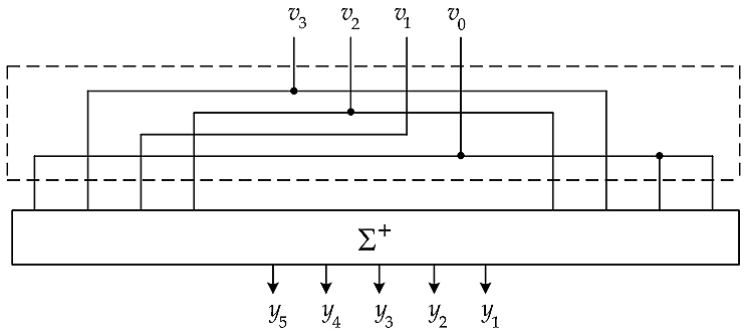


Рис. 8.13: Устройство для вычисления коэффициентов полинома $P(X) = 8 + 9x_1 + 2x_2 + 11x_3$ (Σ^+ — двоичный бинарный сумматор)

уменьшить количество входов группового сумматора в

$$\frac{2^n - t}{d_1} + \frac{t}{\mathrm{abs}(d_2)}$$

раз, где t — количество отрицательных членов арифметического полинома. Недостатком этого метода является сложность его алгоритмизации.

8.7 Выводы

- Оператор s-арного модулярного суммирования является основным оператором, используемым при реализации логических функций посредством модулярных форм арифметических полиномов. Однако известные методы s-арного модулярного суммирования не ориентированы на большие значения параметра s, которые характерны для рассматриваемых арифметико-логических форм.
- Разработан „вертикальный“ метод реализации оператора s-арного модулярного суммирования, который в отличие от известных методов основан на действиях с одноименными разрядными цифрами. Построены параллельная и поразрядно-групповая схемы аппаратурной реализации метода.

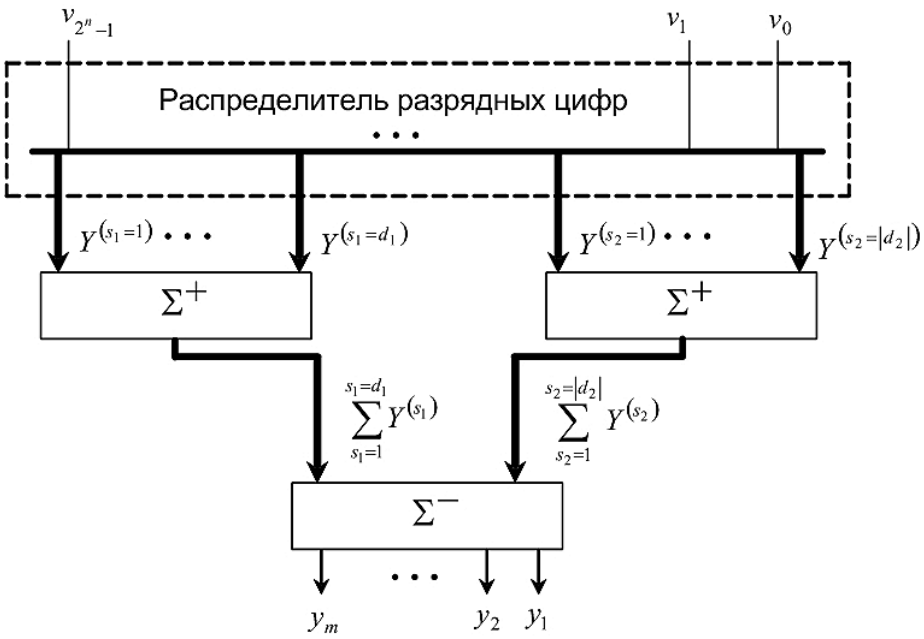


Рис. 8.14: Фрагмент устройства для вычисления полинома с произвольным количеством членов

- Предложен поразрядно-групповой метод s-арного модулярного суммирования, основанный на схеме Горнера и рассмотрена его аппаратурная реализация.
- Разработанный „вертикальный“ метод реализации оператора s-арного модулярного суммирования развит для выполнения s-арного суммирующего преобразования позиционного кода в код по модулю m.
- Получено обобщение метода s-арного модулярного суммирования для реализации s-арного модулярного умножения.
- Основной операцией разработанных методов является операция счета количества единиц по модулю.
- Известны эффективные методы декомпозиции и факторизации для синтеза логических схем для реализации опера-

ций параллельного счета количества единиц по модулю [79]. Схемы, синтезируемые на основе этих методов, имеют высокое быстродействие, поскольку их параметр глубины схемы носит логарифмический характер от количества g входов. Данные методы позволяют регулировать параметры сложности и быстродействия синтезируемых схем. Их использование целесообразно при высоких требованиях к быстродействию схемы и при малых значениях модуля m.

- При больших значениях модуля m могут быть использованы схемы на полных сумматорах и экономичный последовательный счетчик асинхронного действия, предложенный в [7].

- Предложен метод s-арного суммирования констант, представленных в позиционной двоичной системе счисления, основанный на использовании промежуточной избыточной системы счисления, позволяющий сократить количество входов группового сумматора.

Глава 9

Контроль ошибок логических вычислений и отказоустойчивые структуры

Осуществление контроля ошибок логических вычислений вызывает множество затруднений из-за необходимости организации действий над битами [92]. В то же время известно, что методы модулярной арифметики обладают уникальными возможностями по обеспечению контроля ошибок вычислений и повышению отказоустойчивости вычислительных структур.

Исследования в области контроля и синтеза отказоустойчивых модулярных структур активно проводятся с 60-х гг. XX столетия как в нашей стране [13, 16, 23, 34, 39, 51, 55, 56, 102, 111, 131, 155] так и за рубежом [146, 151, 160, 166, 167, 168, 170, 171, 173, 174, 176, 180, 182, 183, 185].

Решение в предыдущих главах проблемы *отображения логических вычислений на модулярную арифметику* открывает доступ к использованию накопленных знаний и испытанных высокоэффективных методов по обеспечению контроля ошибок и повышению отказоустойчивости средств цифровой обработки.

В данной главе[1] мы коснемся только вопросов *адаптации* этих методов к проблеме реализации интенсивного логического счета.

[1]Оригинальные решения, использованные в этой главе, опубликованы в [6, 111, 116, 126, 155]. Дополнительный оригинальный материал опубликован в [100, 101, 188].

9.1 Контроль ошибок избыточными модулярными кодами

9.1.1 Контроль ошибок использованием логарифмической погрешности модулярного представления

Так как результаты вычислений модулярной арифметико-логической формы (2.1), полученные при различных значениях модуля m по определению равны, то имеет место равенство:

$$\mu^{(a)}(X) = \mu^{(b)}(X),$$

где $\mu^{(a)}(X)$ и $\mu^{(b)}(X)$ модулярные формы арифметических полиномов (2.1) при различных значениях модуля — соответственно 2^d и m, то есть:

$$\left| \sum_{i=0}^{2^n-1} \psi_i^{(a)} \; x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{2^d} = \left| \sum_{i=0}^{2^n-1} \psi_i^{(b)} \; x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m , \qquad (9.1)$$

где $\psi_i^{(a)} = |c_i|_{2^d}$, $\psi_i^{(b)} = |c_i|_m$ $(i = 0, 1, \dots, 2^n - 1)$ — соответственно коэффициенты полиномов $\mu^{(a)}(X)$ и $\mu^{(b)}(X)$ модулярной арифметико-логической формы (2.1) при соотношении модулей

$$m \geq 2^d.$$

c_i $(i = 0, 1, \dots, 2^n - 1)$ — коэффициенты полинома $D(X)$ арифметико-логической формы (1.3).

Пусть

$$m > 2^d.$$

Тогда результат вычисления (2.1) должен быть представлен с помощью $d + l$ двоичных разрядов, где значение l определяется выражением

$$l = \lceil \log_2 m \rceil - d,$$

называется *логарифмической погрешностью* представления результата преобразования и является количеством избыточных двоичных разрядов представления результата вычисления Y.

Так как $Y_{\max} = 2^d - 1$, то получаемые в результате вычисления Y двоичные разряды

$$y_d,\ y_{d-1},\ \ldots,\ y_1$$

будем считать информационными, а оставшиеся старшие разряды

$$y_{d+l},\ \ldots,\ y_{d+1}$$

контрольными разрядами, так как при отсутствии ошибок вычислений

$$0 \le Y < 2^d,$$

в противном случае

$$2^d \le Y < 2^{d+l}.$$

Контрольное выражение при этом будет выглядеть следующим образом:

$$\bigvee_{i=d+1}^{d+l} y_i = 0. \tag{9.2}$$

Особенности технической реализации данного метода контроля поясняются с помощью рис. 9.1. Данный метод контроля обеспечивает вероятностный характер обнаружения ошибок и может быть рекомендован для контроля ошибок, вызванных устойчивыми неисправностями (в течении некоторого периода их действия) на основе минимальных аппаратурных затрат контрольного оборудования. Более высокую эффективность этот метод контроля может показать при организации тестового контроля, так как тестовое воздействие можно построить с учетом особенностей метода.

9.1.2 Контроль ошибок L-кодами

Модулярные коды с избыточностью, образованной использованием модулей с общими множителями, изучались в [102] и названы L-кодами.

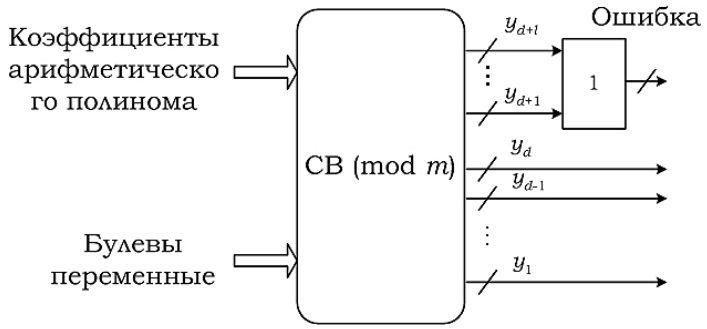


Рис. 9.1: Схемотехническая реализация контроля ошибок в модулярных арифметико-логических формах на основе использования логарифмической погрешности представления (СВ ($\mathrm{mod}\, m$)) — специализированный вычислитель, функционирующий по модулю m)

Контроль одномодульных арифметико-логических форм

Покажем возможность применения этих кодов к одномодульным арифметико-логическим формам (2.1) как предельный случай многомодульного кодирования.

Пусть в представлении (2.1) модуль m — составное число, такое, что $m > Y_{max}$, $m = m_1 m_2$, при этом $\gcd(m_1, m_2) = 1$. Тогда полиному (2.1) можно сопоставить систему модулярных форм арифметических полиномов:

$$\begin{cases} Y = \mu(X) = \left| \sum_{i=0}^{2^n-1} \psi_i \, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m , \\ \mu^{(\mathrm{contr})}(X) = \left| \sum_{i=0}^{2^n-1} \psi_i^{(\mathrm{contr})} \, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_k} , \end{cases} \tag{9.3}$$

где $\psi_i^{(\mathrm{contr})} = |\psi_i|_{m_k} = |c_i|_{m_k}$; $i = 0, 1, \dots, 2^n - 1$; $k \in \{1, 2\}$.

Полученное соотношение (9.3) позволяет обеспечить контроль ошибок. Признаком обнаруживаемой ошибки является выполнение неравенств:

$$\left| \mu(X) - \mu_i^{(\mathrm{contr})} \right|_{m_k} \neq 0 \quad \text{или} \quad \mu_i^{(\mathrm{contr})} \neq |\mu(X)|_{m_k} . \tag{9.4}$$

Принцип схемотехнической реализации этого метода поясняется с помощью рис. 9.2. Данный код позволяет обнаруживать арифметические ошибки, не кратные контрольному модулю m_k и во многом совпадает с широко известным арифметическим кодом контроля по модулю [142].

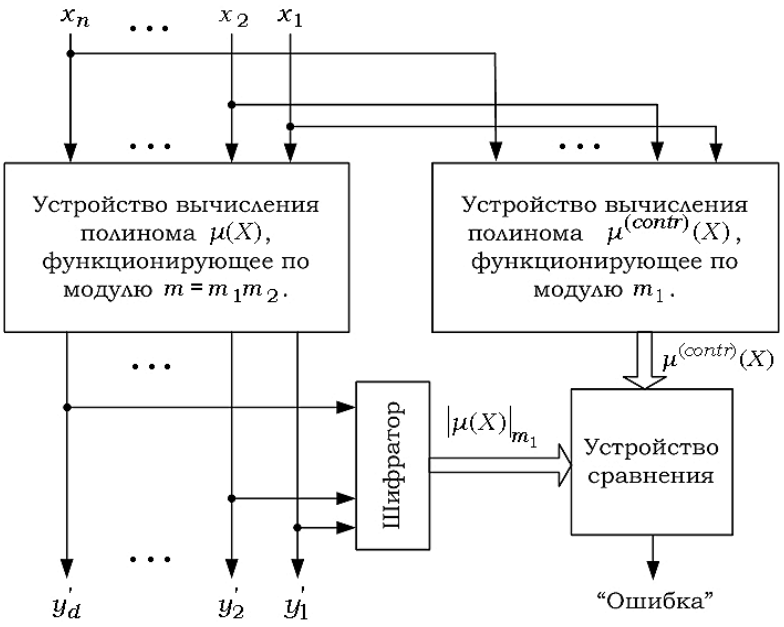


Рис. 9.2: Пример схемотехнической реализации контроля ошибок на основе выражений (9.4)

Рассмотренный случай можно обобщить и для многомодульной арифметики.

Контроль многомодульных арифметико-логических форм

Частный случай L-кодов с одним „ведущим“ модулем был предложен в [6] для контроля аналого-цифрового преобразователя.

Пусть L-код задан системой модулей:

$$m_1, m_2, \ldots, m_v$$

таких, что

$$\begin{aligned} m_1 &= g_1, \\ m_2 &= g_1 g_2, \\ &\vdots \\ m_v &= g_1 g_v; \end{aligned}$$

то есть

$$\begin{aligned} \gcd(m_1, m_2, \ldots, m_v) &= g_1, \\ \gcd(g_1, g_2, \ldots, g_v) &= 1. \end{aligned}$$

Информационный (рабочий) числовой диапазон в этом случае определяется как:

$$M_v = \frac{1}{g_1^{v-2}} \prod_{k=2}^{v} m_k.$$

Получаемый в результате функционирования вычислительной системы результат, который может содержать ошибки, обозначим следующим образом:

$$Y^{'} = \{\phi_1^{'} = \beta_1, \phi_2^{'}, \ldots, \phi_v^{'}\},$$

где $\beta_1 = |Y^{'}|_{g_1}$, $\phi_2^{'} = |Y^{'}|_{m_2}$, $\ldots$, $\phi_v^{'} = |Y^{'}|_{m_v}$.

Контрольные отношения имеют вид [6]:

$$\begin{cases} |\beta_2 - \beta_1|_{g_1} = 0, \\ |\beta_3 - \beta_1|_{g_1} = 0, \\ \vdots \\ |\beta_v - \beta_1|_{g_1} = 0 \end{cases} \tag{9.5}$$

или

$$\begin{cases} \beta_2 = \beta_1, \\ \beta_3 = \beta_1, \\ \vdots \\ \beta_v = \beta_1, \end{cases} \tag{9.6}$$

где $\beta_k = \left|\phi_k^{'}\right|_{g_1}$.

Для оценки корректирующих свойств L-кодов используется две метрики.

Определение 9.1
Весом $\omega(\{Y\})$ кодового слова $\{Y\}$ называется количество его ненулевых символов (вычетов). Расстояние (геометрическое) между модулярными кодами $\{Y_1\}$ и $\{Y_2\}$ определяется как вес их разности $\omega(\{Y_1 - Y_2\})$.

Определение 9.2
Весом $\omega^{(A)}(\{\phi_k\})$ вычета ϕ_k будем называть собственно числовое значение ϕ_k. Расстояние (арифметическое) между модулярными вычетами $\phi_k^{'}$ и $\phi_k^{''}$ определяется как арифметический вес их разности $\omega^{(A)}(\{\phi_k^{'} - \phi_k^{''}\})$.

Признаком однократной (в геометрическом смысле) обнаруживаемой ошибки в вычете ϕ_k является выполнение неравенства

$$\omega^{(A)}(\{\beta_k - \beta_1\}) \neq 0,$$

где $k \in \{2, \ldots, v\}$. При этом обнаруживаются ошибки, имеющие арифметический вес

$$g_1 \not| \omega^{(A)}(\{\beta_k - \beta_1\}).$$

Признаком обнаруживаемой ошибки в вычете $\phi_1^{'}$ можно считать выполнения неравенств $\omega^{(A)}(\{\beta_k - \beta_1\}) \neq 0$ для всех $k = 2, \ldots, v$. Для обнаружения ошибок большей кратности ($\frac{v-2}{2}$, при четном v) можно использовать мажоритарный алгоритм декодирования, предложенный в [6].

Многомодульные L-коды могут быть использованы и для исправления ошибок.

9.1.3 Контроль ошибок R-кодами

Определение 9.3
Расширенным модулярным кодом (R-кодом) называется код

$$\{Y\} = (\phi_1,\ \phi_2,\ \ldots,\ \phi_v,\ \ldots,\ \phi_{v+r}),$$

полученный по расширенной системе попарно простых модулей

$$m_1,\ m_2,\ \ldots,\ m_v,\ \ldots,\ m_{v+r},$$

таких, что

$$m_1 < m_2 < \ldots < m_v < \ldots < m_{v+r}, \tag{9.7}$$

$0 \le Y < m = \prod_{i=1}^{v} m_i$ *— информационный (рабочий) числовой диапазон,* $\widehat{m} = \prod_{i=1}^{v+r} m_i$ *— общий числовой диапазон.*

Так как используемая система модулей упорядочена по величине, для восстановления числа Y по вычетам $(\phi_1,\ \phi_2,\ \ldots,\ \phi_v,\ \ldots,\ \phi_{v+r})$ достаточно знать любые v вычетов. Остальные r вычетов являются избыточными.

Одиночной ошибкой в слове R-кода называется произвольное искажение одного из вычетов. t-кратная ошибка — произвольное искажение t вычетов.

Для того, чтобы R-код исправлял t или менее ошибок необходимо и достаточно, чтобы для минимального кодового расстояния D выполнялось условие:

$$D \ge 2t + 1.$$

Для определения признаков обраруживаемой ошибки при вычислениях в модулярных арифметико-логических формах немного модифицируем теорему, рассмотренную в [23, 172].

Теорема 9.1
R-код обнаруживает все одиночные ошибки, если $r \ge 1$, при этом признаком обнаруживаемой ошибки является выполнение условия

$$Y' \ge 2^d,$$

где d — количество реализуемых булевых функций, Y' — результат вычисления.

Доказательство
Пусть в результате цифровой обработки (вычисление, хранение, передача данных) должно быть получено правильное кодовое слово

$$(\phi_1,\ \phi_2,\ \ldots,\ \phi_v,\ \ldots,\ \phi_{v+r}),$$

однако реальное кодовое слово имеет вид

$$(\phi_1^{'},\ \phi_2^{'},\ \ldots,\ \phi_v^{'},\ \ldots,\ \phi_{v+r}^{'}),$$

$\{e\} = (0,\ \ldots,\ e_j,\ \ldots,\ 0)$ — слово-ошибка.

По слову $\{Y^{'}\}$ определим

$$F = \sum_{i=1}^{v} M_i d_i^{'}, \tag{9.8}$$

где $d_i^{'}$ — решение сравнения

$$M_i d_i^{'} \equiv \phi_i^{'} \pmod{m_i}, M_i = \frac{\widehat{m}}{m_i} = \prod_{i=1}^{i-1} m_i \prod_{j=1}^{j+1} m_j.$$

Учитывая, что

$$Y = \left| \sum_{i=1}^{v} M_i d_i \right|_{\widehat{m}},$$

где $0 \le Y < m$ и $d_i^{'} = d_i$ для всех $i \ne j$, представим (9.8) в виде

$$F = Y + M_j d_j^{''}, \tag{9.9}$$

где $d_j^{''}$ есть решение сравнения

$$M_j x \equiv e_j \pmod{m)_j}.$$

В (9.8) число $M_j\ d_j^{''} > m$, так как $0 < d_j^{''} < m_j$ и на основании (9.7) $M_j \ge m$. Но тогда $F > m$. Имеем

$$F_{max} \le m - 1 + \frac{m m_v (m_v - 1)}{m_v} = \widehat{m} - 1.$$

Таким образом,

$$m \le F < \widehat{m}.$$

Учтем, что если $Y_{max} = 2^d - 1$, где d — количество реализуемых булевых функций, то в качестве признака обнаруживаемой ошибки запишем неравенство:

$$Y' \geq 2^d$$

или

$$\bigvee_{i=d+1}^{d+l} y_i \neq 0, \tag{9.10}$$

где $l = \lceil \log_2 \widehat{m} \rceil - d$.

□

Пример специализированного вычислителя с обнаружением ошибок функционирования представлен на рис. 9.3. Для обеспечения обнаружения ошибок в этом вычислителе использован один избыточный вычислительный канал. Для проверки условия (9.11) используется элемент ИЛИ.

Вопросам построения алгоритмов декодирования ошибок в R-кодах посвящено большое количество литературы [13, 16, 23, 51, 102, 173, 174].

9.2 Реконфигурация вычислительных структур, функционирующих в R и RL кодах

Принципы использования избыточных кодов в вычислительных системах существенно отличаются от принципов их использования для защиты от ошибок каналов связи. В отличие от канала связи вычислительную систему возможно реконфигурировать таким образом, чтобы временно вывести из ее состава часть оборудования, в котором обнаружен отказ.

Примеры вычислительных систем, функционирующих в избыточных R-кодах представлены на рис 9.4 и 9.5. Важной особенностью этих систем является раздельность оборудования, закрепленного за контролируемыми вычислительными каналами.

В вычислительной системе, представленной на рис. 9.4, используется r избыточных вычислителей, а вычислитель (или

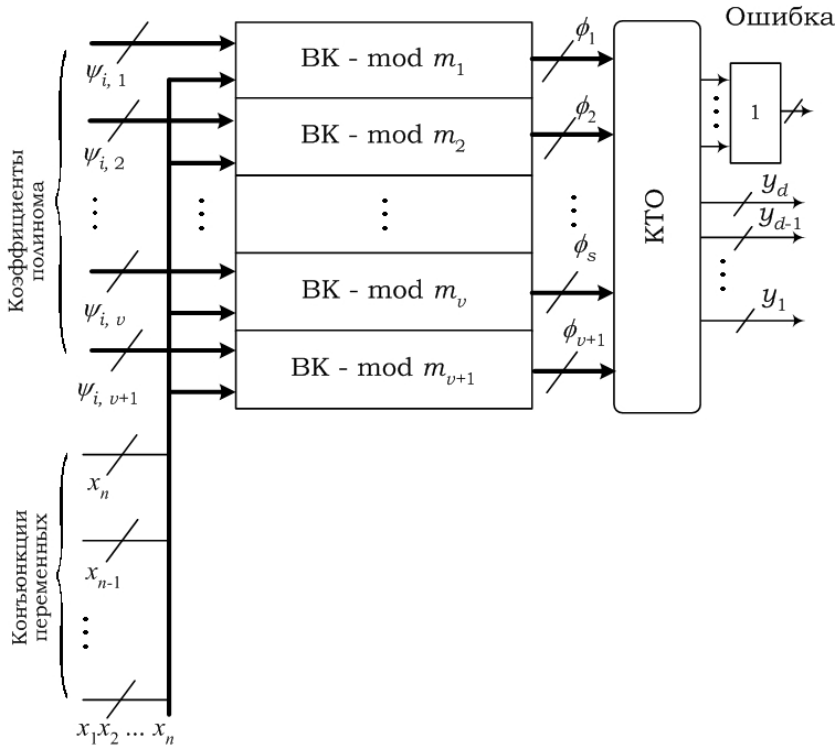


Рис. 9.3: Специализированный вычислитель с одним избыточным каналом и цепями контроля ошибок вычислений

программа), обеспечивающий восстановление позиционной формы числа, наделен функцией декодирования ошибок и управления реконфигурацией системы.

Важной особенностью вычислительных систем, функционирующих в R-кодах является отсутствие разделения вычислительный каналов на информационные и контрольные. По мере деградации вычислительной системы из ее состава может быть выведено до $(r-1)$ вычислителей, соответствующих любым модулям.

Более того, возможно продолжение функционирования системы даже после выхода из строя r и более вычислителей. В этом

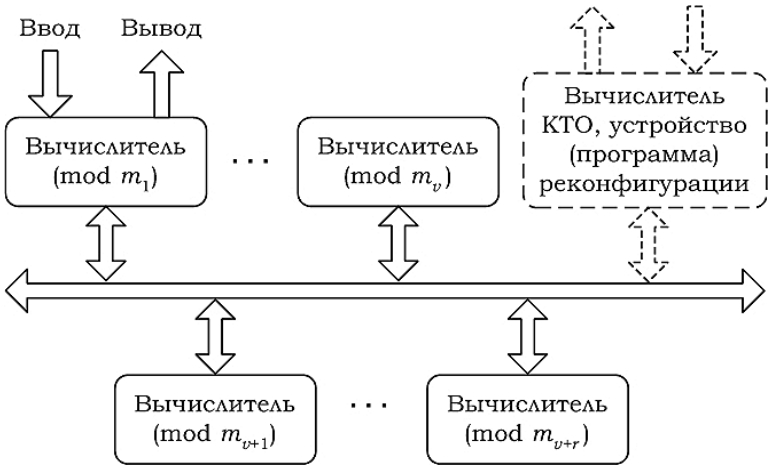


Рис. 9.4: Отказоустойчивая многопроцессорная вычислительная система с r избыточными вычислителями

случае возможен обмен между потерей производительности системы и снижением точности вычислений. Таким образом речь идет о возможности обеспечения повышенной живучести вычислительной системы. В пределе вычислительная система продолжит вычисления даже при наличии только одного исправного вычислителя, который будет последовательно отрабатывать задания по заданной системе модулей и алгоритм восстановления позиционной формы числа.

Выполнение реконфигурации возможно двумя способами: 1) исключением отказавших вычислительных каналов из функционирования путем коммутации с изменением состава модулей в алгоритме восстановления позиционной формы числа и 2) принудительным обнулением вычетов на выходе отказавших вычислительных каналов с внесением изменений в параметры алгоритма восстановления позиционной формы числа.

Достоинством первого варианта является минимизация количества модулей, используемых в алгоритме восстановления позиционной формы числа. Второй вариант позволяет выполнить коммутацию „математическим“ способом.

В вычислительной системе с коммутацией каналов (рис. 9.5)

используется $\frac{v+r}{s}$ избыточных специализированных вычислителей. Процесс декодирования ошибок может быть организован на уровне групп модулей. Целесообразность этого обусловлена тем, что неисправности в специализированных вычислителях могут возникать не только в отдельных вычислительных каналах, но в вычислителе в целом, например, в результате изменения параметров цепей питания или механических воздействиях на микросхему. Кроме того, таким образом решается задача контроля ошибок в устройствах восстановления позиционной формы числа.

Рассмотрим вариант с „математической“ коммутацией вычислительных каналов. В основу алгоритма восстановления позиционной формы числа положим наиболее известный метод ортонормированных базисов, базирующийся на Китайской теореме об остатках (теорема А.28). Будем рассматривать случай преобразования R-кода. Тогда формулу (3.8) можно представить в следующем виде:

$$Y'' = \left| \sum_{\substack{k=1,\,2,\,\ldots,\,v+r;\\ k\neq j_1,\,j_2,\,\ldots,\,j_h}} B_k^{(j_1,j_2,\ldots,j_h)}\phi'_k \right|_{M_{v+r}^{(j_1,j_2,\ldots,j_h)}}, \qquad (9.11)$$

где

$$M_{v+r}^{(j_1,j_2,\ldots,j_h)} = \prod_{\substack{k=1,\,2,\,\ldots,\,v+r;\\ k\neq j_1,\,j_2,\,\ldots,\,j_h}} m_k;$$

$$B_k^{(j_1,\ldots,j_h)} = \begin{cases} \dfrac{q_k^{(j_1,\ldots,j_h)} M_{v+r}^{(j_1,\ldots,j_h)}}{m_k}, \text{ при } k=1,\ldots,v+r;\ k\neq j_1,\ldots,j_h, \\ 0, \quad \text{при} \quad k=j_1,\,j_2,\,\ldots,\,j_h \end{cases}$$

являются ортонормированными базисами; коэффициенты $q_k^{(j_1,\ldots,j_h)} = 1,\,2,\,\ldots,\,m_k-1$ находятся из сравнений

$$\frac{q_k^{(j_1,\ldots,j_h)} M_{v+r}^{(j_1,\ldots,j_h)}}{m_k} \equiv 1 \pmod{m_k}$$

для $k = 1,\,2,\,\ldots,\,v+r$.

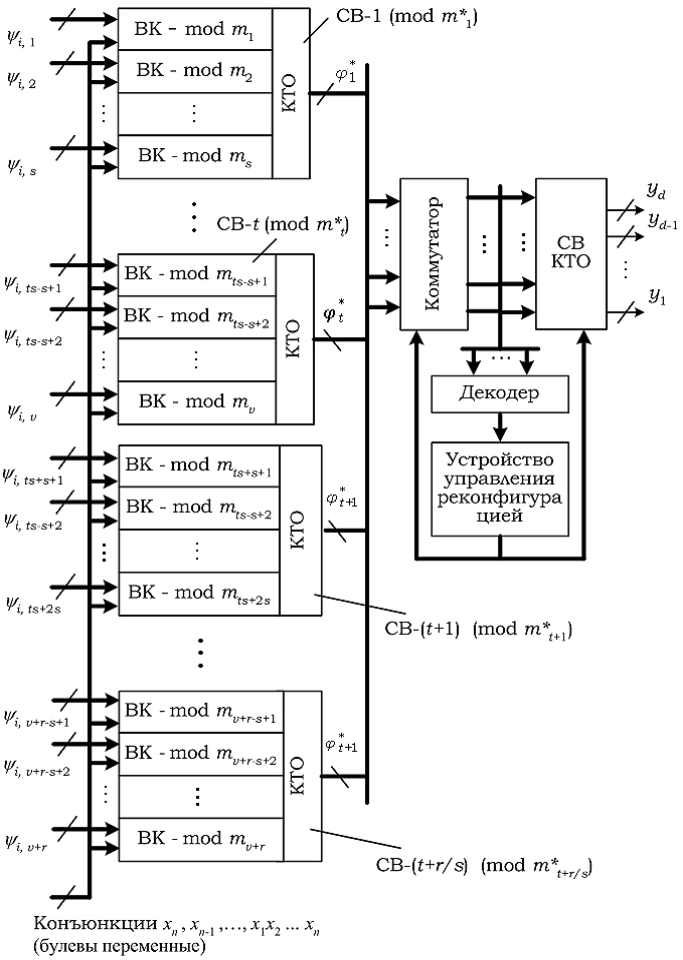


Рис. 9.5: Отказоустойчивая многопроцессорная система с r избыточными специализированными вычислителями

В отличие от формулы (3.8) ортонормированные базисы $B_k^{(j_1, j_2, \ldots, j_h)}$ и модуль преобразования $M_{v+r}^{(j_1, j_2, \ldots, j_h)}$ являются „настраиваемыми“ величинами. При этом j_1, j_2, …, j_h — номера модулей, исключенных из процесса восстановления позиционной формы числа.

При аппаратурной реализации выражения (9.11) устройство восстановления позиционной формы числа будет содержать (рис. 9.6) табличные устройства — ТУ-1,…, ТУ-$(v+r)$, предназначенные для хранения результатов произведений

$$B_k^{(j_1,\ldots,j_h)}\phi_k' \quad (k = 1, 2, \ldots, v+r),$$

табличное устройство ТУ, предназначенное для хранения констант $M_{v+r}^{(j_1, j_2, \ldots, j_h)}$ и $v+r$-входовый сумматор $\Sigma^+ (\mathrm{mod}\, M_{v+r}^{(j_1, j_2, \ldots, j_h)})$, функционирующий по модулю $M_{v+r}^{(j_1, j_2, \ldots, j_h)}$.

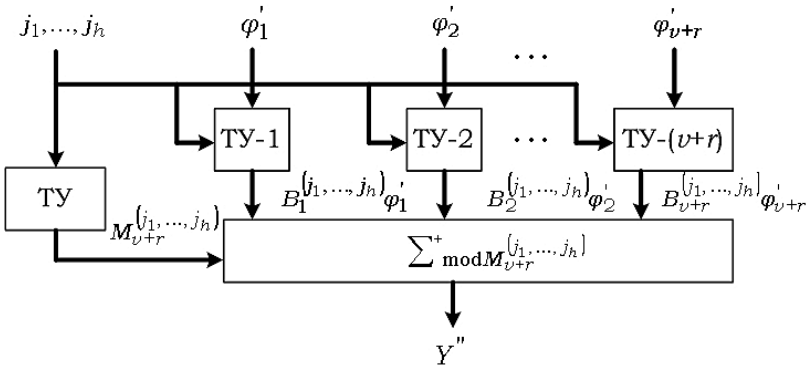


Рис. 9.6: Устройство восстановления позиционной формы числа, обеспечивающее „математическую“ коммутацию вычислительных каналов

Пример расчетных значений ортонормированных базисов $B_i^{(j)}$ и общих модулей $M_5^{(j)}$, составленных для системы модулей $m_1 = 3$, $m_2 = 5$, $m_3 = 7$, $m_4 = 8$, $m_5 = 11$ при $r = 1$ представлен в табл. 9.1.

L-код, заданный расширенной системой $v + r$ модулей, называется RL-кодом. В этом случае свойства L-кода могут быть

Таблица 9.1: Пример расчетных значений ортонормированных базисов $B_i^{(j)}$ и общих модулей $M_5^{(j)}$ для R-кода с одним избыточным модулем

j	$B_1^{(j)}$	$B_2^{(j)}$	$B_3^{(j)}$	$B_4^{(j)}$	$B_5^{(j)}$	$M_5^{(j)}$
1	0	616	2640	385	2520	3080
2	616	0	792	1617	672	1848
3	880	1056	0	825	1200	1320
4	385	231	165	0	210	1155
5	280	336	120	105	0	840

использованы для обнаружения ошибок в вычислительных каналах, а свойства R-кода могут быть использованы для реконфигурации системы.

Пример устройства, обеспечивающего контроль ошибок и реконфигурацию вычислительной системы представлен на рис. 9.7. Здесь контроль ошибок на основе отношений (9.6) обеспечивается с помощью устройств сравнения.

Для организации контроля ошибок „внутри“ вычислительных каналов вместо L-кода можно использовать и другие числовые коды, например избыточные позиционные системы счисления [100, 101].

9.3 Контроль дефектов методом тестирования

В случаях, когда требуется реализовать функцию контроля ценой минимальных избыточных аппаратурных средств, полезны методы тестового и комбинированного — кодового и тестового контроля.

В соответствии с ранее приведенными положениями появление в кодовом слове R-кода

$$\{Y^{'}\}(\phi_1^{'},\ \phi_2^{'},\ \ldots,\ \phi_v^{'},\ \ldots,\ \phi_{v+r}^{'})$$

ошибки с вектором

$$\{e\} = (0,\ \ldots,\ e_{j_1},\ \ldots,\ e_{j_h},\ \ldots,\ 0),$$

где $h \leq t$, всегда приводит к выходу результата восстановления $Y^{'}$ из рабочей области $0 \leq Y^{'} < M_v$, где $M_v = \prod_{k=1}^{v} m_k$ и

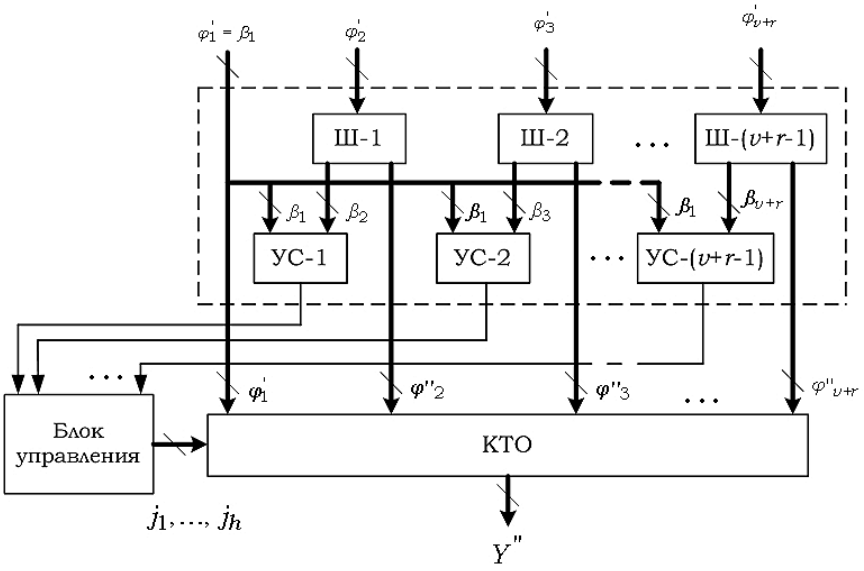


Рис. 9.7: Устройство контроля и реконфигурации вычислительной системы, функционирующей в RL-коде (УС — устройство сравнения, Ш — шифраторы, КТО — устройство восстановления позиционной формы числа)

попадания его недопустимую область:

$$M_v \leq Y' < M_{v+r},$$

где $M_{v+r} = \prod_{k=1}^{v+r} m_k$, или

$$Y' \geq 2^d,$$

что является признаком обнаруживаемой ошибки.

Как было показано выше для организации контроля достаточно использовать элемент ИЛИ.

Для того, чтобы восстановить утраченные при этом возможности исправления ошибок, при обнаружении факта ошибки выполняется останов рабочего цикла функционирования вычислительной системы и производится ее тестирование с целью выявления отказавшего вычислительного канала. Если отказ носит устойчивый характер, то такой вычислительный канал будет

выявлен и после изменения параметров алгоритма восстановления позиционной формы числа правильный (с заданной вероятностью) результат будет восстановлен. Если причина отказа — сбой, цикл вычислений повторяется.

Процесс тестирования при этом заключается в параллельном выполнении тестовой последовательности для всех вычислительных каналов с последовательным исключением одного или нескольких вычислительных каналов (в зависимости от кратности ошибки) из процесса функционирования, чем и достигается реконфигурация оборудования вычислительной системы или вычислительного устройства.

Если тестовая последовательность — простой перебор всех допустимых кодовых комбинаций на входах всех вычислительных каналов, то при небольшой величине максимального значения модуля m_{v+r} для однократной ошибки функционирования, длина тестовой последовательности не превышает величины nm_{v+r} [111, 116, 126].

Таким образом, комбинированный метод позволяет локализовать до $(v + r - 1)$ вычислительных каналов за счет потерь времени, необходимых для выполнения тестовой последовательности и введения указанных ограничений на причины возникновения отказа. Если же наложить ограничение и на причины возникновения отказов, обнаруживая отказы, вызванные только устойчивыми неисправностями, то контроль можно осуществить не вводя избыточных модулей [111, 116, 126].

9.4 Выводы

- Использование методов модулярной арифметики для реализации логического счета помимо повышения производительности вычислений позволяет получить не менее важные преимущества, связанные с обеспечением контроля ошибок логических вычислений и обеспечением отказоустойчивости вычислительных структур.

- Предложен принцип контроля вычислительных структур, предназначенных для реализации арифметико-логических форм, основанных на одномодульной арифметике. Показа-

но, что техническая реализации такого контроля приближается к хорошо изученной классической схеме контроля по модулю.

- Рассмотрены принципы контроля вычислений над многомодульными арифметико-логическими формами посредством L и R избыточных модулярных кодов.

- Показано, что вычислительная система, функционирующая в избыточном модулярном коде, является отказоустойчивой, способной сохранять работоспособность при выходе из строя до r произвольных вычислительных каналов.

- Высокая живучесть вычислительной системы, функционирующей в модулярной арифметике, обеспечивается сохранением общей работоспособности системы при постепенной деградации оборудования с понижением показателей производительности или точности вычислений. Вычислительная система, функционирующая в модулярной арифметике, является исключительно однородной, приближаясь по своим свойствам к живым организмам. При этом высокая однородность обеспечивается не только аналогичностью или идентичностью используемых вычислительных каналов (вычислителей), но и отсутствием четкой грани между „информационным“ и „контрольным“ оборудованием.

- Предложены технические решения обеспечения контроля и реконфигурации вычислительной системы на основе выходного устройства (вычислителя) восстановления позиционной формы числа [111, 116, 126, 155].

- Показано, что методы тестового контроля обеспечивают высокую экономичность оборудования, позволяя обнаруживать устойчивые неисправности даже в вычислительной системе, не содержащей избыточные вычислительные каналы. Показана экономическая целесообразность применения комбинированных (кодовых и тестовых) методов контроля и реконфигурации оборудования [111, 116, 126, 155].

Заключение

Итоговые выводы и замечания сформулируем в виде следующих положений.

1. Основы вычислительных методов алгебры логики, используемые в настоящее время, были созданы в „докомпьютерную“ эпоху и *плохо согласуются* с методами организации вычислений в современной компьютерной технике. Напротив, арифметическая логика полностью *соответствует* принципам построения современных и перспективных ЭВМ и позволяет раскрыть неиспользуемый в настоящее время потенциал вычислительной техники по реализации высокопроизводительных, гибких, параллельных логических вычислений.

2. Модулярные арифметико-логические формы (общая классификация представлена на рис. 9.8) обладают рядом новых полезных свойств и ориентированы на *воплощение в современную и перспективную практику цифровой обработки информации идей арифметической логики на основе высокоразвитого и прогрессивного научно-методического аппарата модулярной арифметики.*

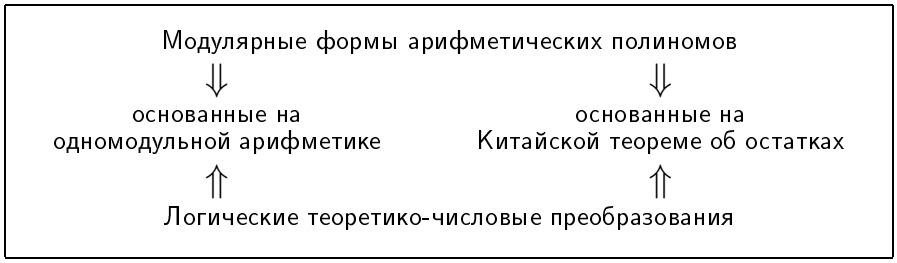


Рис. 9.8: Классификация модулярных форм

Достоинствами модулярных арифметико-логических форм являются:

- высокая степень *параллелизма* логических вычислений, которая может быть классифицирована как сверхпараллелизм;
- уникальные возможности по обеспечению *отказоустойчивости* и *живучести* средств логических вычислений;
- обеспечение *контроля* и *коррекции* ошибок на всех стадиях обработки, хранения, а также передачи информации;
- создание благоприятных условий для приоритетного использования быстродействующих *табличных* операционных устройств (в том числе на базе программируемой логики) за счет существенного уменьшения (по сравнению с двоичной системой счисления — на порядки [13]) объема таблиц;
- уменьшение *сложности* представления логических функций;
- возможность *многоцелевого* использования средств логических вычислений, которая, в свою очередь, может быть использована для обеспечения отказоустойчивости и живучести вычислительной системы [22, 33, 38, 84, 103] или для сокращения аппаратурных затрат за счет разделения решения задач во времени;
- повышение *защищенности* средств обработки от методов инженерной разведки за счет существенного усложнения побочных электромагнитных излучений и наводок.

3. В настоящее время модулярная арифметика широко применяется в методах и средствах цифровой обработки сигнальной информации. Модулярные арифметико-логические формы позволяют *задействовать* высокоразвитый математический аппарат и совершенные технические средства цифровой обработки сигналов, базирующиеся на методах модулярной арифметики, для высококачественной реализации параллельных логических вычислений.

4. Таким образом, модулярные арифметико-логические формы, по-видимости, позволяют преодолеть главное *противоречие* двух основных способов реализации логических алгоритмов: *программного* (гибкого) и *аппаратного* (жесткого). *Логические вычисления, обладающие достоинствами программной реализации, становится возможным реализовать специализированными вычислительными средствами*, характеризующимися требуемым комплексом технических характеристик.

Наглядный пример сочетания ряда противоречивых требований — специализированные вычислители, устанавливаемые на малых космических аппаратах с длительными сроками активного существования (7–10 лет и более) [98]. Проблемы, возникающие при проектировании бортовых средств цифровой обработки заключаются в обеспечении заданной производительности, определяемой высокой скоростью и большими объемами передаваемой информации (например, оптической и радиоголографической), а также надежности функционирования в условиях размещения приборов на стенках негерметизированных сотовых панелей в открытом космосе с общей защитой до 0, 3 г/см^2, при жестких ограничениях на энергопотребление и массогабаритные показатели.

Методы арифметической логики в этих условиях позволяют существенно упростить поиск оптимального решения[2].

5. Известной трудностью современной практики программирования криптографических алгоритмов, реализуемых на основе сигнальных процессоров, является максимизация использования возможностей процессора для реализации специфических криптографических функций. Труд программиста, в этом случае, требует высокой (дорогостоящей) квалификации и приближается в большей степени к *искусству*, нежели к *технологии*. Использование методов арифметической логики позволяет свести разнообразие используемых логических функций к реализации однотипных арифметических полиномов, что делает процесс программирования более *технологичным* (повышает качество, сокращает сроки и стоимость программирования).

[2]Методы модулярной арифметики были применены для повышения надежности бортовой вычислительной машины аэрокосмической ракеты STAR [56]

6. Структура технического средства (даже при использовании средств программируемой логики), основанного на методах арифметической логики, *не отражает* реализуемый алгоритм, что может быть использовано в качестве дополнительного средства повышения стойкости шифрования.

7. Известно, что *проговая функция* может быть реализована линейным арифметическим полиномом и условным оператором [24]. В [135] этот результат развит на область реализации логических функций линейными арифметическими полиномами с *маскированиями*. Таким образом и *система* пороговых функций может быть задана *одним* линейным арифметическим полиномом с маскированиями. Этот фундаментальный результат, в частности, может быть использован для моделирования нейронных сетей методами арифметической логики, где каждому слою нейронной сети поставлен в соответствие линейный арифметический полином.

Модулярные арифметико-логические формы могут составить модулярную арифметико-логическую модель нейронной сети. Таким образом, преимущества модулярных арифметико-логических форм могут быть распространены и на область нейровычислений.

8. Методы модулярной арифметики продолжают развиваться и находят все более широкое применение, прежде всего, в алгоритмах и средствах цифровой обработки сигнальной информации [51, 57, 60, 64, 82, 91, 132, 181], а также в задачах ассиметричной криптографии [106, 108, 112, 123, 169, 177, 178, 179].

Исследуются вопросы построения высокопроизводительных модулярных структур на перспективной оптической и электрооптической элементной базе [11, 83, 109, 175].

Развиваются методы и средства ввода и вывода информации, повышающие эффективность специализированных вычислителей за счет уменьшения этапов промежуточных преобразований. Так, в [3, 5, 6] предложен ряд технических решений, обеспечивающих аналого-цифровое преобразование непосредственно в код модулярной арифметики. В [109, 115] предложены аналого-цифровые преобразователи в код модулярной арифметики, основанные на электрооптической и сверхпроводниковой (впервые)

элементной базе.

Отмечается интерес к *нейроподобным*[3] модулярным вычислительным структурам [29, 133, 134, 190, 191, 196, 197, 198, 199], которые помимо рассмотренных в книге средств могут быть использованы в целях аппаратной поддержки методов арифметической логики (в широком смысле), в том числе, и для аппаратной поддержки нейровычислений (в узком смысле), основанных на реализации модулярных арифметико-логических форм.

Поэтому, хотя модулярная арифметика является в известном смысле проблемно-ориентированным математическим средством, расширение решаемых на ее основе задач, принадлежащих, как ранее считалось, к диаметрально противоположным классам, позволяет сделать вывод о *становлении единой математической и технической базы, основанной на методах модулярной арифметики*, предназначенной для решения широкого спектра задач интенсивной, достоверной, гибкой обработки информации в различных сферах практического применения.

[3]Здесь имеются принципиальные отличия от указанного выше способа построения модулярных арифметических моделей нейронных сетей, которые *инвариантны* по отношению к способу их технической реализации. Напротив, *нейроподобные* (приближающиеся к структуре нейронных сетей) модулярные структуры — вполне определенный способ *технической реализации* модулярных вычислений, имеющий преимущества по показателю отказоустойчивости, которые достигаются органическим сочетанием свойств неронных сетей и свойств модулярной арифметики.

Приложение А

Необходимые положения теории чисел

Данное приложение носит справочный характер и составлено преимущественно на основе [25] с использованием [12, 16, 26] и введением некоторых дополнительных обозначений.

Определение А.1 *Целые числа a и b называются сравнимыми по модулю m, если разность $a-b$ делится на m, т. е. если $m|a-b$.*

Для обозначения этого соотношения используется запись

$$a \equiv b \pmod{m},$$

a и b называются соответственно левой и правой частями сравнения, а число $m \geq 1$ — модулем. В дальнейшем нас будут интересовать только модули $m > 1$.

Пример А.1 $137 \equiv 16 \pmod{11}$.

Теорема А.1 *a сравнимо с b тогда и только тогда, когда a и b имеют одинаковые остатки при делении на m.*

Доказательство. Пусть остатки от деления a и b на m равны, т. е. $a = mq_1 + r$ и $b = mq_2 + r$; тогда $a - b = m(q_1 - q_2)$, $m|a-b$ и $a \equiv b \pmod{m}$.

Теорема А.2 *Отношение сравнимости обладает свойствами:*

- *рефлексивности:*

$$a \equiv a \pmod{m};$$

- *симметричности: если*

$$a \equiv b \pmod{m}, \quad \textit{то} \quad b \equiv a \pmod{m};$$

- *транзитивности: если*

$$a \equiv b \pmod{m}, \quad b \equiv c \pmod{m}, \quad \textit{то} \quad a \equiv c \pmod{m}.$$

Теорема А.3 *Если* $a \equiv b \pmod{m}$ *и* k *— произвольное целое число, то*

$$ka \equiv kb \pmod{m}.$$

Доказательство. Если $a \equiv b \pmod{m}$, то $m|a-b$, $m|k(a-b)$, $m|ka-kb$, $ka \equiv kb \pmod{m}$.

Теорема А.4 *Если* $ka \equiv kb \pmod{m}$ *и* $\gcd(k, m) = 1$, *то*

$$a \equiv b \pmod{m}.$$

Доказательство. Если $ka \equiv kb \pmod{m}$, то $m|ka-kb$, $m|k(a-b)$, то на основании справедливости утверждения $c|b$, если $c|ab$ и $\gcd(a, c) = 1$, условие $\gcd(k, m) = 1$ дает $m|a-b$, т. е. $a \equiv b \pmod{m}$.

Теорема А.5 *Если* $a \equiv b \pmod{m}$ *и* k *— произвольное натуральное число, то* $ka \equiv kb \pmod{km}$.

Доказательство. Если $a \equiv b \pmod{m}$, то $m|a-b$, $km|ka-kb$, $ka \equiv kb \pmod{km}$.

Теорема А.6 *Если* $ka \equiv kb \pmod{km}$, *где* k *и* m *— произвольные натуральные числа, то* $a \equiv b \pmod{m}$.

Доказательство. Если $ka \equiv kb \pmod{km}$, то $km|k(a-b)$, k — натуральное, т. е. $k \neq 0$, и тогда

$$m|a-b, \quad a \equiv b \pmod{m}.$$

Теорема А.7 *Если* $a \equiv b \pmod{m}$, $c \equiv d \pmod{m}$, *то* $a+c \equiv b+d \pmod{m}$ *и* $a-c \equiv b-d \pmod{m}$.

Доказательство. Если $a \equiv b \pmod{m}$ и $c \equiv d \pmod{m}$, то $m|a-b$ и $m|c-d$. На основании утверждения $c|a+b$ и $c|a-b$, если $c|a$ и $c|d$, имеем

$$m|(a-b) \pm (c-d),\ m|(a \pm c) - (b \pm d),\ a \pm c \equiv b \pm d \pmod{m}.$$

Теорема А.8 *(обобщение теоремы А.7). Если*

$$\begin{aligned} a_1 &\equiv b_1 \pmod m, \\ a_2 &\equiv b_2 \pmod m, \\ &\vdots \\ a_n &\equiv b_n \pmod m, \end{aligned}$$

то $a_1 + a_2 + \ldots + a_n \equiv b_1 + b_2 + \ldots + b_n \pmod m$.

Доказательство теоремы А.8 строится аналогично доказательству теоремы А.7.

Теорема А.9 *Если* $a \equiv b \pmod m$, $c \equiv d \pmod m$, *то*

$$ac \equiv bd \pmod m.$$

Доказательство. Если $a \equiv b \pmod m$, $c \equiv d \pmod m$, то согласно теореме А.3 $ac \equiv bc \pmod m$ и $bc \equiv bd \pmod m$; теорема А.2 (транзитивность сравнений) дает $ac \equiv bd \pmod m$.

Теорема А.10 *(обобщение теоремы А.9). Если*

$$\begin{aligned} a_1 &\equiv b_1 \pmod m, \\ a_2 &\equiv b_2 \pmod m, \\ &\vdots \\ a_n &\equiv b_n \pmod m, \end{aligned}$$

то $a_1 a_2 \ldots a_n \equiv b_1 b_2 \ldots b_n \pmod m$.

Доказательство. Последовательно применяя теорему А.3, получим:

$$\begin{aligned} a_1 a_2 a_3 \ldots a_n &\equiv b_1 a_2 a_3 \ldots a_n \equiv \\ &\equiv b_1 b_2 a_3 \ldots a_n \equiv \ldots \\ \ldots &\equiv b_1 b_2 b_3 \ldots b_n \pmod m. \end{aligned}$$

Теорема А.11 *Если* $a \equiv b \pmod m$, *то при любом целом* $n \geq 0$

$$a^n \equiv b^n \pmod m.$$

Доказательство. При $n = 0$ справедливость утверждения вытекает из теоремы А.2 (рефлексивность сравнимости), а при $n \geq 1$ — из теоремы А.10, полагая при этом

$$a_1 = a_2 = \ldots = a_n = a \quad \text{и} \quad b_1 = b_2 = \ldots = b_n = b.$$

Переход от сравнений

$$a \equiv b \pmod{m}, \quad c \equiv d \pmod{m}$$

к сравнениям

$$\begin{aligned} a \pm c &\equiv b \pm d \pmod{m}, \\ ac &\equiv bd \pmod{m}, \\ a^n &\equiv b^n \pmod{m} \end{aligned}$$

называется соответственно сложением, вычитанием, умножением, возведением в степень сравнений.

Поскольку из сравнения $c \equiv d \pmod{m}$ следует $d \equiv c \pmod{m}$, то из сравнений $a \equiv b \pmod{m}$ и $c \equiv d \pmod{m}$ следует, что $a \pm d \equiv b \pm c \pmod{m}$ и $ad \equiv bc \pmod{m}$.

Теорема А.12 *Если имеет место сравнение* $a \equiv b \pmod{m}$ *и* $f(x) = c_0 + c_1 x + \ldots + c_n x^n$ *— произвольный многочлен с целыми коэффициентами, то*

$$f(a) \equiv f(b) \pmod{m}.$$

Доказательство. Если $a \equiv b \pmod{m}$, то согласно теоремам А.11 и А.3 имеем:

$$a^s \equiv b^s \pmod{m}, \quad c_s a^s \equiv c_s b^s \pmod{m}$$

при $s = 0, 1, \ldots, n$, но тогда по теореме А.8 получим:

$$c_0 + c_1 a + \ldots + c_n a^n \equiv c_0 + c_1 b + \ldots + c_n b^n \pmod{m},$$

т. е.

$$f(a) \equiv f(b) \pmod{m}.$$

Теорема А.13 *Если*

$$\begin{aligned} a_1 &\equiv b_1 \pmod{m}, \\ a_2 &\equiv b_2 \pmod{m}, \\ &\vdots \\ a_n &\equiv b_t \pmod{m}, \end{aligned}$$

и $f(x_1, x_2, \ldots, x_t)$ *— многочлен с целыми коэффициентами, то*

$$f(a_1, a_2, \ldots, a_t) \equiv f(b_1, b_2, \ldots, b_t) \pmod{m}.$$

Доказательство аналогично доказательству теоремы А.12.

Теорема А.14 *Любое слагаемое левой или правой части сравнения можно перенести с противоположным знаком в противоположную часть.*

Доказательство. На основании теоремы А.2 (симметричность сравнимости) достаточно рассмотреть случай, когда дано сравнение

$$a + b \equiv c \pmod{m}.$$

Складывая это сравнение со сравнением

$$-b \equiv -b \pmod{m},$$

получим:

$$a \equiv c - b \pmod{m}.$$

Теорема А.15 *В сравнении можно отбрасывать или добавлять слагаемые, делящиеся на модуль.*

Доказательство. Если $a + c \equiv b \pmod{m}$, то $m|c$, т. е. $-c \equiv 0 \pmod{m}$, то, складывая эти сравнения, получаем $a + c \equiv b \pmod{m}$.

Теорема А.16 *Если* $a \equiv b \pmod{m}$ *и* $d|m$, *то*

$$a \equiv b \pmod{d}.$$

Доказательство. Если $a \equiv b \pmod{m}$, то $m|a - b$. Из $d|m$, $m|a - b$ в силу транзитивности отношения делимости получим $d|a - b$, $a \equiv b \pmod{d}$.

Теорема А.17 *Если* $a \equiv b \pmod{m}$, *то множество общих делителей* a *и* m *совпадает с множеством общих делителей* b *и* m. *В частности,* $\gcd(a, m) = \gcd(b, m)$.

Доказательство. Если $a \equiv b \pmod{m}$, то $m|a - b$, $a - b = mq$, $b = a - mq$, любой общий делитель δ чисел a и m является общим делителем чисел b и m, и, наоборот, если $\delta|b$ и $\delta|m$, то $\delta|a$.

Поскольку пара a, m и пара b, m имеют одни и те же общие делители, то и $\gcd(a, m) = \gcd(b, m)$.

Теорема А.18 *Если*

$$\begin{aligned} a &\equiv b \pmod{m_1}, \\ a &\equiv b \pmod{m_2}, \\ &\vdots \\ a &\equiv b \pmod{m_s}, \end{aligned}$$

то $a \equiv b \pmod{m}$, *где* $m = \operatorname{lcm}(m_1, m_2, \ldots, m_s)$.

Доказательство. Если $a \equiv b \pmod{m_1}$,
$a \equiv b \pmod{m_2}, \ldots, a \equiv b \pmod{m_s}$, то $m_1|a-b$, $m_2|a-b$, $\ldots$, $m_s|a-b$, и согласно свойствам наименьшего общего кратного $m|a-b$.

Определение А.2 *Классом $\overline{a}$ по данному модулю m называется множество всех целых чисел, сравнимых с некоторым данным целым числом a.*

Определение А.3 *Вычетом класса называется любое из чисел, принадлежащих этому классу.*

Теорема А.19 *Наименьший неотрицательный вычет класса $\overline{a}$ по модулю m равен остатку от деления a на m.*

Доказательство. Обозначим через r остаток от деления a на m, т. е. положим $a = mq+r$, $0 \le r < m$. Тогда $a \equiv r \pmod{m}$, $\overline{a} = \overline{r}$. Любое $x \in \overline{a}$ имеет вид $x = r+mt$. При целых отрицательных t будем иметь $r+mt \le r-m < 0$, т. е. положительные числа класса получаются при неотрицательных значениях t, а наименьшее среди них, получающееся при $t = 0$, равно r.

В дальнейшем для обозначения наименьшего неотрицательного вычета класса $\overline{a}$ по модулю m будем использовать запись

$$|a|_m.$$

Теорема А.20 *Пусть r — остаток от деления a на m; тогда наименьший по абсолютной величине вычет класса $\overline{a}$ равен:*
1) r, если $0 \le r < m/2$;
2) $\pm r$, если $r = m/2$;
3) $r - m$, если $m/2 < r < m$.

Доказательство. Если $a = mq + r$ $(0 \le r < m)$ и $x \in \overline{a}$, то $x = r + mt$, где t может равняться $0, \pm1, \pm2, \ldots$. Наименьшее неотрицательное значение x равно r, а наименьшее по абсолютной величине отрицательное значение x равно $r - m$. Если при этом:
1) $0 \le r < m/2$, то $r < \mathrm{abs}(r - m)$. Наименьший по абсолютной величине вычет равен r.
2) $r = m/2$ (только при четном m), то $r = \mathrm{abs}(r - m)$. Имеются два наименьших по абсолютной величине вычета r и $r - m = -r$.
3) $m/2 < r < m$, то $\mathrm{abs}(r - m) < r$. Наименьший по абсолютной величине вычет равен $r - m$.

Обозначим множество классов по модулю m как $\mathcal{Z}_m$.

Теорема А.21 *Множество классов $\mathcal{Z}_m$ по данному модулю m представляет собой коммутативное кольцо.*

Определение А.4 *Функцией Эйлера $\varphi(m)$ называется количество натуральных чисел, не превосходящих m и взаимно простых с m.*

Теорема А.22 *Если* $m = p_1^{\alpha_1}, p_2^{\alpha_2}, \ldots, p_s^{\alpha_s}$ *— каноническое разложение числа* m*, то*

$$\varphi(m) = p_1^{\alpha_1-1}\, p_2^{\alpha_2-1} \ldots p_s^{\alpha_s-1}\,(p_1-1)(p_2-1)\ldots(p_s-1).$$

Теорема А.23 (Ферма) *Для любого простого модуля* m *и любого* $a \geq 1$*, не делящегося на* m*, справедливо сравнение*

$$a^{m-1} \equiv 1 \pmod{m}.$$

Следствие А.1 *Если* m *— простое число, то в кольце* $\mathcal{Z}_m$ *выполняется равенство*

$$\left|a^{-1}\right|_m = \left|a^{m-2}\right|_m.$$

Теорема А.24 (Эйлер) *Для любого модуля* m *и любого* $a \geq 1$*, взаимно простого с* m*, справедливо сравнение*

$$a^{\varphi(m)-1} \equiv 1 \pmod{m}.$$

Следствие А.2 *В кольце* $\mathcal{Z}_m$ *из* $\gcd(a, m) = 1$ *вытекает, что*

$$\left|a^{-1}\right|_m = \left|a^{\varphi(m)-1}\right|_m.$$

Определение А.5 *Элемент* a^{-1} *кольца* $\mathcal{Z}_m$ *будем называть обратной мультипликативной величиной (элементом).*

Для уменьшения значения k, такого, что $a^k \equiv 1 \pmod{m}$ полезной может быть обобщенная функция Эйлера $L(m)$.

Определение А.6 *Обобщенной функцией Эйлера* $L(m)$ *называется функция, определенная для всех натуральных значений* m *следующим образом:* $L(1) = 1$*, а при* $m > 1$

$$L(m) = \operatorname{lcm}\left(p_1^{\alpha_1-1}(p_1-1),\, p_2^{\alpha_2-1}(p_2-1),\, \ldots,\, p_s^{\alpha_s-1}(p_s-1)\right),$$

где $m = p_1^{\alpha_1}, p_2^{\alpha_2}, \ldots, p_s^{\alpha_s}$ *— каноническое разложение* m*.*

Люка доказал, что вместо $L(m)$ можно рассматривать функцию

$$l(m) = \begin{cases} L(m), & \text{при } 8 \nmid m, \\ \frac{1}{2}L(m), & \text{при } 8 \mid m. \end{cases}$$

Теорема А.25 *Кольцо* $\mathcal{Z}_m$ *при простом модуле* m *является полем.*

Рассмотрим сравнение $ax \equiv b \pmod{m}$, которое будем называть сравнением с неизвестной величиной x.

Теорема А.26 *Если* $\gcd(a, m) = 1$, *то сравнение* $ax \equiv b \pmod{m}$ *имеет одно и только одно решение.*

Теорема А.27 *При* $\gcd(a, m) = 1$ *решением сравнения*

$$ax \equiv b \pmod{m}$$

является класс

$$x \equiv ba^{\varphi(m)-1} \pmod{m}.$$

Доказательство. Применяя теорему Эйлера, получаем

$$a(ba^{\varphi(m)-1}) = ba^{\varphi(m)} \equiv b \pmod{m},$$

и тогда согласно теореме А.26 $x \equiv ba^{\varphi(m)-1} \pmod{m}$ — единственное решение сравнения $ax \equiv b \pmod{m}$.

Теперь рассмотрим систему сравнений 1-й степени с неизвестным x

$$\left.\begin{array}{rcll} x & \equiv & c_1 & (\mathrm{mod}\ m_1), \\ x & \equiv & c_2 & (\mathrm{mod}\ m_2), \\ & \vdots & & \\ x & \equiv & c_v & (\mathrm{mod}\ m_v). \end{array}\right\} \tag{A.1}$$

Теорема А.28 *Пусть* $m_1, m_2, \ldots, m_v$ — *попарно взаимно простые числа,* $M = m_1 m_2 \ldots m_v$; $k_1, k_2, \ldots, k_v$ *удовлетворяют сравнениям:*

$$\begin{array}{rcll} k_1 M m_1^{-1} & \equiv & 1 & (\mathrm{mod}\ m_1), \\ k_2 M m_2^{-1} & \equiv & 1 & (\mathrm{mod}\ m_2), \\ & \vdots & & \\ k_v M m_v^{-1} & \equiv & 1 & (\mathrm{mod}\ m_v), \end{array}$$

тогда решение системы (А.1) будет представлять собой класс:

$$x \equiv x_0 \pmod{M}, \tag{A.2}$$

где

$$x_0 = \frac{M}{m_1} k_1 c_1 + \frac{M}{m_2} k_2 c_2 + \ldots + \frac{M}{m_v} k_v c_v.$$

Доказательство. Поскольку

$$m_1 \left| \frac{M}{m_2} \right., \ldots, m_1 \left| \frac{M}{m_v} \right.$$

и $\frac{M}{m_1} k_1 \equiv 1 \pmod{m_1}$, получаем $x_0 \equiv \frac{M}{m_1} k_1 c_1 \equiv c_1 \pmod{m_1}$. Аналогично проверяем, что

$$x_0 \equiv c_2 \pmod{m_2}, \ldots, x_0 \equiv c_v \pmod{m_v},$$

т. е. x_0 удовлетворяет всем сравнениям системы. Так как m_1, m_2, ..., m_v — попарно взаимно просты, то решение этой системы представляет собой класс M, т. е. $x \equiv x_0 \pmod{M}$.

Система сравнений (А.1) имеет своим решением единственное число при

$$0 \leq x < M.$$

Т. е.

$$x = \left| \frac{M}{m_1} k_1 c_1 + \frac{M}{m_2} k_2 c_2 + \ldots + \frac{M}{m_v} k_v c_v \right|_M . \tag{А.3}$$

Определение А.7 *Показателем $P_m(a)$ по модулю m называется наименьший положительный показатель степени a, сравнимой с единицей по модулю m, т. е.*

$$a^{P(a)} \equiv 1 \pmod{m},$$

причем для всех r, таких, что $1 \leq r < P(a)$, $a^r \not\equiv 1 \pmod{m}$.

$$x = \left| \frac{M}{m_1} k_1 c_1 + \frac{M}{m_2} k_2 c_2 + \ldots + \frac{M}{m_v} k_v c_v \right|_M . \tag{А.4}$$

Определение А.8 *Класс $\overline{a}$, где $\gcd(a, m) = 1$, называется первообразным корнем по модулю m, если показатель $\overline{a}$ по этому модулю равен $\varphi(m)$, т. е. если $P(\overline{a}) = \varphi(m)$.*

Определение А.9 *Пусть $\gcd(a, m) = 1$, $\gcd(b, m) = 1$; число s называется индексом b по модулю m и основанию a, если*

$$a^s \equiv b \pmod{m}$$

Для этого при фиксированном значении модуля используется запись: $s = \mathrm{ind}_a b$. Т. е. $a^{\mathrm{ind}_a b} \equiv b \pmod{m}$.

Пусть g — первообразный корень по модулю m и $\varphi(m)$ — функция Эйлера от m, тогда справедливы следующие свойства индексов.

Свойство А.1 *Пусть* $\gcd(b, m) = 1$, *тогда сравнение*

$$c \equiv b \pmod{m}$$

имеет место тогда и только тогда, когда

$$\mathrm{ind}_g c \equiv \mathrm{ind}_g b \pmod{\varphi(m)}.$$

Свойство А.2 *Пусть* $\gcd(a_1, m) = 1$, $\gcd(a_2, m) = 1$, $\ldots$, $\gcd(a_n, m) = 1$, *тогда*

$$\mathrm{ind}_g(a_1 \cdot a_2 \cdot \ldots \cdot a_n) \equiv \mathrm{ind}_g a_1 + \mathrm{ind}_g a_2 + \ldots \mathrm{ind}_g a_n \pmod{\varphi(m)}.$$

Свойство А.3 *Пусть* $\gcd(a, m) = 1$, $n \geq 1$; *тогда*

$$\mathrm{ind}_g a^n \equiv n \,\mathrm{ind}_g a \pmod{\varphi(m)}.$$

Свойство А.4 *Пусть* $\gcd(a, m) = 1$, $n \geq 1$; *тогда*

$$\mathrm{ind}_g \frac{b}{a} \equiv \mathrm{ind}_g b - \mathrm{ind}_g a \pmod{\varphi(m)}.$$

Таблица значений индексов и значений обратных индексам по модулю 13 представлена в приложени В

Приложение В

Таблица индексов для модуля 13

a	$\operatorname{ind}_g a$	$\operatorname{ind}_g a$	a
1	0	0	1
2	7	1	11
3	4	2	4
4	4	3	5
5	3	4	3
6	11	5	7
7	5	6	12
8	9	7	2
9	8	8	9
10	10	9	8
11	1	10	10
12	6	11	6

Литература

[1] *Авсаркисян Г. С., Брайловский Г. С.* Представление логических функций в виде полиномов Жегалкина // Автоматика и вычисл. техника. — 1975. — № 6.

[2] А. с. 450160 СССР. Устройство для параллельного счета количества единиц (нулей) в двоичном числе / Е. Г. Быков // Открытия. Изобрет. — 1974. — № 42.

[3] А. с. 1259487 СССР, МКИ 4 *H* 03 *M* 1/28. Преобразователь перемещения в код системы остаточных классов / С. Н. Хлевной, О. А. Финько // Открытия. Изобрет. — 1986. — № 35.

[4] А. с. 1343553 СССР, МКИ 4 *H* 03 *M* 7/18. Преобразователь кода системы остаточных классов в позиционный код / Н. И. Червяков, О. Е. Коршунов, О. А. Финько // Открытия. Изобрет. — 1987. — № 37.

[5] А. с. 1368989 СССР, МКИ 4 *H* 03 *M* 1/28. Аналого-цифровой преобразователь в код системы остаточных классов /О. А. Финько и др. // Открытия. Изобрет. — 1988. — № 3.

[6] А. с. 1372620 СССР, МКИ 4 *H* 03 *M* 1/28. Аналого-цифровой преобразователь в системе остаточных классов / О. А. Финько и др. // Открытия. Изобрет. — 1988. — № 5.

[7] А. с. 1383365 СССР, МКИ 4 *G* 03 *F* 11/10. Устройство для свертки по модулю / Н. И. Червяков, Н. И. Швецов, О. А. Финько, А. В. Пальцев // Открытия. Изобрет. — 1988. — № 11.

[8] А. с. 1388996 СССР, МКИ 4 *H* 03 *M* 7/18. Преобразователь кода из системы остаточных классов в позиционный код / Н. И. Червяков, О. Е. Коршунов, О. А. Финько // Открытия. Изобрет. — 1988. — № 14.

[9] А. с. 1444961 СССР, МКИ 4 *H* 03 *M* 1/28. Преобразователь числа в модулярный код / О. А. Финько и др. // Открытия. Изобрет. — 1988. — № 46.

[10] А. с. 1557682 СССР, МКИ 4 *H* 03 *M* 7/18. Преобразователь позиционного кода в код системы остаточных классов / О. А. Финько и др. // Открытия. Изобрет. — 1990. — № 14.

[11] *Акаев А. А., Майоров С. А.* Когерентные оптические вычислительные машины. — Л.: Машиностроение, 1977. —440 с.

[12] *Акритас А.* Основы компьютерной алгебры с приложениями: Пер. с англ. — М.: Мир, 1994. — 544 с.

[13] *Акушский И. Я., Юдицкий Д. И.* Машинная арифметика в остаточных классах. — М.: Сов. радио, 1968. — 440 с.

[14] *Акушский И. Я., Амербаев В. М., Пак И. Т.* Основы машинной арифметики комплексных чисел. — Алма-Ата: Наука, 1970. — 248 с.

[15] *Алферов А. П., Зубов А. П., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. — М.: Гелиос АРВ, 2001. — 480 с.

[16] *Амербаев В. М.* Теоретические основы машинной арифметики. — Алма-Ата: Наука, 1976. — 324 с.

[17] *Антоненко В. М., Иванов А. А., Шмерко В. П.* Линейные арифметические формы k-значных логических функций и их реализация на систолических массивах // Автоматика и телемеханика. — 1995. — № 3. — С. 139–155.

[18] *Артюхов В. Л., Кондратьев В. Н., Шалыто А. А.* Реализация булевых функций арифметическими полиномами // Автоматика и телемеханика. — 1988. — № 4. — С. 138–147.

[19] *Бабаш А. В., Шанкин Г. П.* Криптография. — М.: СОЛОН-Р, 2002. — 512 с.

[20] *Балашов Е. П., Негода В. И., Пузанков Д. В. и др.* Информационные системы. Табличная обработка информации/ Под. ред. Е. П. Балашова, В. Б. Смочова. — Л.: Энергоиздат, Л.О., 1985. — 184 с.

[21] *Байоми М. А.* Заказные матрицы СБИС для структур, основанных на системе счисления в остаточных классах // ТИИЭР. — 1987. — Т. 38. — № 12. — С. 134–139.

[22] *Березюк И. Т.* Живучесть микропроцессорных систем. — Киев: Техника, 1989. — 143 с.

[23] *Бояринов И. М.* Помехоустойчивое кодирование числовой информации. — М.: Наука, 1983. — 196 с.

[24] *Бутаков Е. А.* Методы синтеза релейных устройств из пороговых элементов. — М.: Энергия,1970.

[25] *Бухштаб А. А.* Теория чисел. — М.: Просвещение, 1966. — 384 с.

[26] *Виноградов И. М.* Основы теории чисел. — М.: Гл. ред. физ.-мат. лит., 1972. — 168 с.

[27] *Воеводин В. В., Воеводин Вл. В.* Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.

[28] *Воробьев Н. Н.* Признаки делимости. — 4-е изд., испр. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 96 с.

[29] *Галушкин А. И.* Нейрокомпьютеры в разработках военной техники США (обзор по материалам открытой печати). Часть I // Зарубежная радиоэлектроника. — 1995. — № 5. — С. 3–48.

[30] *Гильберт Д., Аккерман В.* Основы теоретической логики. — М.: ИЛ, 1947. / *Hilbert D., Ackermann W.* Gründzuge der theoretischen Logik. — Berlin, 1928.

[31] *Гильберт Д., Бернайс П.* Основания математики. Логические исчисления и формализация логики. — М.: Наука. Гл. ред. физ.-мат. лит., 1979. — 560 с.

[32] *Гольденберг Л. М., Матюшкин Б. Д., Поляк М. Н.* Цифровая обработка сигналов: Справочник. — М.: Радио и связь, 1985. — 312 с.

[33] *Гуляев А. В.* Организация живучих вычислительных систем // Управляющие системы и машины. — 1987. — № 5. — С. 26–29.

[34] *Дадаев Ю. Г.* Теория арифметических кодов. — М.: Радио и связь. 1981. — 272 с.

[35] *Даджион Д., Мерсеро Р.* Цифровая обработка многомерных сигналов: Пер. с англ. — М.: Мир, 1988. — 488 с.

[36] *Дзегеленок И. И., Оцоков Ш. А.* О распараллеливании безошибочных вычислений на ПМК-сети „Курс-2000“ // Вычислительные сети. — 2003. — № 1.

[37] *Дзюжаньски П., Малюгин В., Шмерко В., Янушкевич С.* Линейные модели схем на многозначных элементах // Автоматика и телемеханика. — 2002. — № 6. — С. 99–119.

[38] *Додонов А. И. и др.* Введение в теорию живучести вычислительных систем. — Киев: Наукова думка, 1990. — 184 с.

[39] *Долгов А. И.* Диагностика устройств, функционирующих в системе остаточных классов. — М.: Радио и связь, 1982. — 64 с.

[40] *Дулепов Е. Г.* Об одном методе канонического синтеза базовых симметричных многополюсников // Вопросы теории ЭЦВМ, — Киев, 1969. С. 3–9.

[41] *Евстигнеев В. Г.* Недвоичная машинная арифметика и специализированные процессоры. — М.: МИФИ СЕРВИС, 1992. — 266 с.

[42] *Евстигнеев В. Г.* Компьютерные арифметики. Ретроспективный взгляд // Электроника, наука,технология, бизнес. — 1998. — № 2. — С. 19–22.

[43] *Жегалкин И. И.* О технике вычисления предложений в символической логике // Математический сборник Московского математического общества. — 1927. — Т. 34. — С. 9–28.

[44] *Жегалкин И. И.* Арифметизация символической логики // Математический сборник Московского математического общества. — 1928. — Т. 35. — С. 311–374, 1929. — Т. 36. — С. 205–338.

[45] *Жуков А. Е.* Системы блочного шифрования. Методы анализа и результаты. (Обзор). — М.: МИРАН, 1992.

[46] *Жуков А. Е.* Современные методы блочного шифрования. (Обзор). — М.: МИРАН, 1994.

[47] *Жуков А. Е.* Системы блочного шифрования. Методы анализа и результаты II. (Обзор). — М.: МИРАН, 1994.

[48] *Инютин С. А.* Параллельные вычисления в сверхбольших компьютерных диапазонах // I Междунар. конф. „Параллельные вычисления и задачи управления“ (PACO-2001). Москва, 29—31 января 2001. Сборник трудов. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2001. — С. 76–87.

[49] *Инютин С. А.* Теория и методы моделирования вычислительных структур с параллелизмом машинных операций: Автореф. дис. ... д-ра техн. наук. — М.: МГИЭТ (ТУ), 2001. — 33 с.

[50] *Кнут Д. Э.* Искусство программирования, том 2. Получисленные алгоритмы, 3-е из. — М.: Издательский дом „Вильямс“, 2000.

[51] *Коляда А. А., Пак И. Т.* Модулярные структуры конвейерной обработки цифровой информации. — Мн.: Университетское, 1992. — 256 с.

[52] *Кондаков В. И.* Логический словарь-справочник. — М.: Наука, 1971. — 352 с.

[53] *Кондратьев В. Н., Шалыто А. А.* Реализация систем булевых функций с использованием линейных арифметических полиномов // Автоматика и телемеханика. — 1993. — № 3.

[54] *Кондратьев В. Н., Шалыто А. А.* Реализация булевых функций одним линейным арифметическим полиномом с маскированием // Автоматика и телемеханика. — 1996. — № 1.

[55] *Краснобаев В. А.* Надежностная модель ЭВМ в системе остаточных классноv // Электронное моделирование. — 1985. — Т. 7, № 4. — С. 44–46.

[56] *Краснобаев В. А. и др.* Методы повышения надежности специализированных ЭВМ систем и средств связи. — Харьков: ХВВКМУ РВ, 1990. — 172 с.

[57] *Кравченко В. Ф., Крот А. М.* Методы и микроэлектронные средства цифровой фильтрации сигналов и изображений на основе теоретико-числовых преобразований // Зарубежная радиоэлектроника. Успехи современной радиоэлектроники. — 1997. — № 6. — С. 3–31.

[58] *Кухарев Г. А., Шмерко В. П., Зайцева Е. Н.* Алгоритмы и систолические процессоры для обработки многозначных данных. — Минск: Наука и техника, 1990.

[59] *Лебедев Е. К.* Цифровая фильтрация в системе остаточных классов // Изв. вузов. Радиотехника. — 1985. — Т. 28, № 8. — С. 58–62.

[60] *Лебедев Е. К.* Синтез нелинейных непозиционных устройств обработки марковских сигналов // Изв. вузов. Радиотехника. — 1987. — Т. 30, № 12. — С. 69–72.

[61] *Лебедев Е. К.* Быстрые алгоритмы цифровой обработки сигналов: Монография. — Красноярск, 1989.

[62] *Лебедев Е. К.* Микропроцессорные коррелятоpы: Монография. — Йошкар-Ола, 1996.

[63] *Лупанов О. Б.* Об одном подходе к синтезу управляющих систем — принципе локального кодирования // Проблемы кибернетики. — М.: Наука, 1965. — Вып. 14. — С. 31–110.

[64] *Макклеллан Дж., Рейдер Ч. М.* Применение теории чисел в цифровой обработке сигналов: Пер. с англ. / Под ред. Ю. И. Манина. — М.: Радио и связь, 1983.

[65] *Малюгин В. Д.* Надежность переключательных схем // Автоматика и телемеханика. — 1964. — № 3. — С. 1375–1383.

[66] *Малюгин В. Д.* О полиномиальной реализации кортежа булевых функций // ДАН СССР. — 1982. — Т. 265, № 6. — С. 1338–1341.

[67] *Малюгин В. Д.* Реализация булевых функций арифметическими полиномами // Автоматика и телемеханика. — 1982. — № 4. — С. 84–93.

[68] *Малюгин В. Д.* Реализация кортежей булевых функций посредством линейных арифметических полиномов // Автоматика и телемеханика. — 1984. — № 2. — С. 114–121.

[69] *Малиновский Б.Н.* История вычислительной техники в лицах. — Киев: Фирма КИТ, ПТОО „АСК“, 1995. — 384 с.

[70] *Малюгин В. Д., Кухарев Г. А., Шмерко В. П.* Преобразования полиномиальных форм булевых функций: Препринт. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 1986.

[71] *Малюгин В. Д.* Параллельные логические вычисления посредством арифметических полиномов. — М.: Наука. Физматлит, 1997.

[72] *Малашевич Б. М.* Разработка вычислительной техники в Зеленограде. Неизвестные супер-ЭВМ // Электроника, наука, технология, бизнес. — 2004. — № 2.

[73] *Мерекин Ю. В.* Арифметические формы записи булевых выражений и их применения для расчета надежности схем // В сб. Вычислительные системы. — Вып. 7. Институт математики СО АН СССР. 1963.

[74] *Молдовян А. А., Молдовян Н. А.* Метод скоростного преобразования для защиты информации в АСУ // Автоматика и телемеханика. — 2000. — № 4. — С. 151–165.

[75] *Молдовян А. А., Молдовян Н. А., Гуц Н. Д., Изотов Б. В.* Криптография: скоростные шифры. — СПб.: БВХ-Петербург, 2002. — 496 с.

[76] *Музыченко О. Н., Лукоянов В. П.* Быстродействующий алгоритм синтеза схем симметричных функций алгебры логики для систем автоматизированного проектирования // Вопросы радиоэлектроники. — 1984. — Вып. 12. — С. 120–128.

[77] *Музыченко О. Н.* Факторизационный метод синтеза схем симметричных функций // Вопросы радиоэлектроники. — 1987. — Вып. 7. — С. 111–117.

[78] *Музыченко О. Н.* Использование симметрии переменных для уменьшения сложности логических схем // Автоматика и телемеханика. — 2000. — № 10. — С. 151–163.

[79] *Музыченко О. Н.* Синтез логических схем модульного контроля в унитарных позиционных двоичных кодах // Автоматика и телемеханика. — 2001. — № 3. — С. 158–173.

[80] *Ноден П., Китте К.* Алгебраическая алгоритмика: Пер. с франц. — М.: Мир, 1999. — 720 с.

[81] *Носов В. А.* Специальные главы дискретной математики. Учебное пособие. — М.: 1990. — 156 с.

[82] *Нуссбаумер Г.* Быстрое преобразование Фурье и алгоритмы вычисления сверток: Пер. с англ. — М.: Радио и связь, 1985. — 248 с.

[83] Оптическая передача и обработка информации: Пер. с фр./ *А. Козанне, Ж. Флере, Г. Руссо.* — М.: Мир, 1984. — 504 с.

[84] Отказоустойчивые цифровые системы // ТИИЭР. — 1978. — Т. 66, № 10. — 239 с.

[85] Пол. решение ВНИИГПЭ РФ о выдаче патента на изобрет. по заявке № 97104283/09, МКИ 6 *H* 03 *M* 7/18. Преобразователь кода системы остаточных классов, заданной модулями $p_2 - p_1 = 1$, в позиционный код / О. А. Финько, С. Б. Елесин, В. В. Корниенко // Открытия. Изобрет. — 1998. — № 7.

[86] Пол. решение ВНИИГПЭ РФ о выдаче патента на изобрет. по заявке № 97104346/09, МКИ 6 *H* 03 *M* 7/18. Преобразователь кода системы остаточных классов, заданной модулями $2^i < p$, в позиционный код / О. А. Финько, С. Б. Елесин, В. В. Корниенко // Открытия. Изобрет. — 1998. — № 5.

[87] Пол. решение ВНИИГПЭ РФ о выдаче патента на изобрет. по заявке № 97104347/09, МКИ 6 *G* 06 *F* 1/02. Генератор возвратных последовательностей чисел / О. А. Финько, С. Б. Елесин, В. В. Корниенко // Открытия. Изобрет. — 1998. — № 5.

[88] *Пухов Г. Е., Евдокимов В. Ф., Синьков М. В.* Разрядно-аналоговые вычислительные системы. — М.: Сов. радио, 1978. — 255 с.

[89] *Порецкий П. С.* О способах решения логических равенств и об обратном способе математической логики // Собрание протоколов заседаний секции физ.-мат. наук при Казанском ун-те. — Казань, 1884. — т. 2.

[90] *Прангишвили И. В. и др.* Микропроцессоры и локальные сети микроЭВМ в распределенных системах управления — М.: Энергоатомиздат, 1985.

[91] *Рабинер Л., Гоулд Б.* Теория и применение цифровой обработки сигналов. — М.: Мир, 1978. — 848 с.

[92] *Савельев А. Я.* Прикладная теория цифровых автоматов. — М.: Высшая школа, 1987. — 272 с.

[93] *Садыхов Р. Х., Татур М. М.* Технический сервис однородных вычислительных устройств. — Мн.: Университетское, 2001. — 279 с.

[94] *Сачков В. Н., Солодовников В. И., Федюкин М. В.* Дискретные функции, используемые в криптографии. — М., 1998.

[95] *Свобода А.* Развитие вычислительной техники в Чехословакии. Система счисления остаточных классов // Кибернетический сборник. — М.: Мир, 1964. — Вып. 8. — С. 115–148 / *Svoboda A.* Computer progress in Czechoslovakia. II. The numerical system of residual classes (SRE) in Digital Informations Wandles, 1962.

[96] *Синьков М. В., Губарени Н. М.* Непозиционные представления многомерных числовых систем. — Киев: Наукова думка, 1977. — 149 с.

[97] *Сизоненко А. Б., Кузьменко А. С., Финько О. А.* Нелинейная арифметическая модель рекуррентного регистра сдвига // IV Межведомственная научно-техническая конф. „Проблемы совершенствования систем защиты информации и образовательных технологий подготовки специалистов в области информационной безопасности“. Краснодар, 2003. Сборник трудов. — Краснодар: Краснодарский воен. ин-т, 2003. — С. 144–147.

[98] *Сиренко В. Г., Гришин В. Ю., Еремеев П. М., Зубов Н.,Н.* Проблемы и стратегия проектирования высокопроизводительных бортовых комплексов управления малыми космическими аппаратами с длительными сроками активного существования // Третья научно-техническая конф. „Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М. В. Хруничева“. Москва, 16—18 декабря 2003. Пленарные доклады. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. — С. 51–72.

[99] *Стройк Д. Я.* Краткий очерк истории математики: Пер. с немец. — М.: Наука. Физматлит, 1984. — 285 с.

[100] *Ткаченко А. В., Финько О. А.* Синтез и преобразование сложных структурных кодов // Автоматика и телемеханика. 1995. — № 5. — С. 183–189.

[101] *Ткаченко А. В., Финько О. А.* Согласованные позиционные избыточные счисления // Электронное моделирование. — 1996. — Т. 18, № 5. — С. 72–75.

[102] *Торгашев В. А.* Система остаточных классов и надежность ЦВМ. — М.: Сов. радио, 1973. — 120 с.

[103] *Турута Е. П.* Обеспечение отказоустойчивости управления микропроцессорных систем путем перераспределения задач отказавших модулей/ Системы управления информационных сетей. — М.: Наука, 1983. — С. 187–189.

[104] *Финько О. А.* Применение функции Эйлера для преобразования кода системы остаточных классов в непозиционный код. — НИР по ХД 499 от 30.11.1984 г. Отчет № 14. Научный руководитель В. И. Ключко. Краснодар: КВВКИУ РВ, 1986. — С. 119–131.

[105] *Финько О. А.* Методика распараллеливания сложных операций в специализированных вычислителях АСУ при проблемно-ориентированном представлении данных: Дис. ... канд. техн. наук. — Краснодар, КВВКУ РВ, 1995. — 240 с. (Библиотека КВИ).

[106] *Финько О. А. и др.* Непозиционное представление данных в криптосистемах с открытым ключом // V научно-техническая конференция Ракетных Войск. Краснодар, 17—19 сентября 1997. Тезисы докладов. Часть I. — Краснодар: КВВКИУ РВ, 1997. — С. 13.

[107] *Финько О. А.* Восстановление числа в системе остаточных классов с минимальным количеством оснований // Электронное моделирование. — 1998. — Т. 20, № 3. — С. 56–61.

[108] *Финько О. А., Елесин С. Б.* Принципы построения средств аппаратной поддержки криптосистем с открытым ключом // 4-й сессия Междунар. научно-технической школы-семинара „Передача, обработка и отображение информации“. Сочи-Теберда, март-апрель 1997, Материалы. — Ставрополь: Ставропольский воен. авиац. ин-т, 1998.

[109] *Финько О. А.* Синтез параллельных электрооптических аналого-цифровых преобразователей для вычислителей, функционирующих в модулярной арифметике // Изв. ВУЗов. Приборостроение. — 1999. — Т. 42, № 3–4. — С. 30–32.

[110] *Финько О. А.* Параллельная реализация оператора группового суммирования констант в позиционной арифметике // 5-й сессия Междунар. научно-технической школы-семинара „Передача, обработка и отображение информации“. Сочи-Теберда, март-апрель 1998, Материалы. — Ставрополь: Ставропольский воен. авиац. ин-т, 1999.

[111] *Финько О. А.* Контроль и реконфигурация аналого-цифровых устройств, функционирующих в системе остаточных классов // Электронное моделирование. — 2000. —- Т. 22, № 4. — С. 92–103.

[112] *Финько О. А. и др.* Методика защиты модулярных криптопроцессоров от аппаратных ошибок // Межвузовский сборник научных трудов. — Краснодар: Краснодарский военный ин-т, 2001. — С. 171–175.

[113] *Финько О. А.* Сверхпараллельные логические вычисления методами модулярной арифметики // Междунар. конф. „Искусственные интеллектуальные системы“ (IEEE AIS'02) и „Интеллектуальные САПР“ (CAD-2002). Геленджик, 5–10 сентября 2002. Сборник трудов. — М.: Наука. Физматлит, 2002. — С. 448–455.

[114] *Финько О. А.* Параллельные логические вычисления, использующие избыточные представления чисел // Вторая Междунар. конф. „Идентификация систем и задачи управления“ (SICPRO'03). Москва, 29–31 января 2003. Сборник трудов (CD). — М.: Ин-т проблем управ. им. В. А. Трапезникова РАН, 2003. — С. 1716–1728.

[115] *Финько О. А.* Сверхпроводниковый аналого-цифровой преобразователь для устройств цифровой обработки сигналов, функционирующих в остаточных классах // 5-я Междунар. конф. „Цифровая обработка сигналов и ее применение“ (DSPA-2003). Москва, 12–14 марта 2003. Сборник трудов. — М.: Радиотехника, 2003. — С. 575–579.

[116] *Финько О. А., Кузьменко А. С.* Тестопригодное устройство цифровой обработки сигналов, функционирующее в остаточных классах с простым средством контроля // 5-я Междунар. конф. „Цифровая обработка сигналов и ее применение“ (DSPA-2003). Москва, 12—14 марта 2003. Сборник трудов. — М.: Радиотехника, 2003. — С. 571–574.

[117] *Финько О. А.* Арифметико-кодовая совместимость устройств обработки информации в распределенных АСУ при проблемно-ориентированном представлении данных // Вторая Междунар. конф. по проблемам управ. (МКПУ II). Москва, 17–19 июня 2003. Сборник трудов. — М.: Ин-т проблем управ. им. В. А. Трапезникова РАН, 2003. — С. 139.

[118] *Финько О. А.* Логические вычисления на основе теоретико-числовых преобразований // Вторая Междунар. конф. по проблемам управ. (МКПУ II). Москва, 16–20 июня 2003. Сборник трудов. — М.: Ин-т проблем управ. им. В. А. Трапезникова РАН, 2003. — Т. 2. — С. 159–166.

[119] *Финько О. А.* Варианты Китайской теоремы об остатках, ориентированные на техническую реализацию // Междунар. конгресс „Математика в XXI в. Роль механико-математического факультета Новосибирского гос. ун-та в науке, образовании и бизнесе“. Новосибирск. Академгородок, 25—28 июня 2003. Тез. докл. http://www.sbras.ru/ws/MMF-21/.

[120] *Финько О. А.* Модулярные формы арифметических полиномов для реализации систем булевых функций // Междунар. конф. „Искусственные интеллектуальные системы“ (IEEE AIS ' 03) и „Интеллектуальные САПР“ (CAD-2003). Геленджик, 3—10 сентября 2003. Сборник трудов. — М.: Наука. Физматлит, 2002. — С. 548–560.

[121] *Финько О. А.* Синтез арифметических форм булевых функций посредством теоретико-числовых преобразований // Перспективные информационные технологии и интеллектуальные системы. — Таганрог: Таганрогский гос. радиотехнический ун-т, 2003. — № 15. — С. 45–53.

[122] *Финько О. А.* Вариант классификации арифметических форм представления логических функций // XIV Междунар. школа-семинар „Синтез и сложность управляющих систем“. Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов / Под ред. академика РАН О. Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. — С. 83–84.

[123] *Финько О. А.* Групповой контроль ассиметричных криптосистем методами модулярной арифметики // XIV Междунар. школа-семинар „Синтез и сложность управляющих систем“. Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов/ Под ред. академика РАН О. Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. — С. 85–86.

[124] *Финько О. А.* Методы обработки больших массивов информации на основе арифметики в остаточных классах // Третья научно-техническая конф. „Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М. В. Хруничева“. Москва, 16—18 декабря 2003. Сборник трудов. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. — С. 211–216.

[125] *Финько О. А.* Проблемы арифметико-кодовой совместимости устройств обработки информации в изделиях ракетно-космической техники // Третья научно-техническая конф. „Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М. В. Хруничева“. Москва, 16–18 декабря 2003. Сборник трудов. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. — С. 216–218.

[126] *Финько О. А., Кузьменко А. С.* Простой способ контроля тестопригодного устройства цифровой обработки сигналов, функционирующего в остаточных классах // Приборы и системы. Управление, контроль, диагностика. — 2004. — № 1. — С. 37–39.

[127] *Финько О. А.* Применение цифровой обработки сигналов для реализации интенсивных логических вычислений // 6-я Междунар. конф. „Цифровая обработка сигналов и ее применение“ (DSPA-2004). Москва, 31 марта — 2 апреля 2004. Сборник трудов. — М.: Радиотехника, 2004. — Т. 1. — С. 265–268.

[128] *Финько О. А.* Реализация систем булевых функций большой размерности методами модулярной арифметики // Автоматика и телемеханика. — 2004. — № 6. — С. 37–60. (Специальный выпуск; в печати).

[129] *Финько О. А.* Полиномиальная арифметика функций многозначной логики по заданному модулю // Известия вузов. Приборостроение. — 2004. — Т. 47, № 5. — С. 41–46.

[130] *Финько О. А.* Параллельные логические вычисления методами модулярной арифметики // II Междунар. конф. „Параллельные вычисления и задачи управления“ (PACO-2004). Москва, 4–6 октября 2004. Сборник трудов. — М.: Ин-т проблем управ. им. В. А. Трапезникова РАН, 2004. (В печати).

[131] *Червяков Н. И.* Отказоустойчивые непозиционные процессоры // Управляющие системы и машины. — 1988. — № 3. — С. 3–7.

[132] *Червяков Н. И., Велигороша А. и др.* Цифровые фильтры в системе остаточных классов // Теоретическая радиотехника. — 1995. — Вып. 38, № 8. — С. 11–20.

[133] *Червяков Н. И., Сахнюк П. А. и др.* Модулярные параллельные вычислительные структуры нейропроцессорных систем. — М.: Физматлит, 2003. — 288 с.

[134] *Червяков Н. И., Сахнюк П. А. и др.* Нейрокомпьютеры в остаточных классах. Кн. 11. — М.: Радиотехника, 2003. — 272 с.

[135] *Шалыто А. А.* Логическое управление. Методы аппаратной и программной реализации алгоритмов. — СПб.: Наука, 2000. — 780 с.

[136] *Шевкопляс Б. В.* Микропроцессорные структуры. Инженерные решения: Справочник. — М.: Радио и связь, 1990. — 512 с.

[137] *Шенброт И. М., Антропов М. В., Давиденко К. Я.* Распределенные АСУ технологическими процессами. — М.: Энергоатомиздат, 1985. — 240 с.

[138] *Шеннон К.* Теория связи в секретных системах // В кн.: Работы по теории информации и кибернетике. — М.: ИЛ, 1963. — 480 с.

[139] *Шмерко В. П.* Синтез арифметических форм булевых функций посредством преобразования Фурье // Автоматика и телемеханика. — 1989. — № 5. — С. 134–142.

[140] *Шмерко В. П.* Теоремы Малюгина: новое понимание в логическом управлении, проектировании СБИС и структурах данных для новых технологий // Автоматика и телемеханика. — 2004. — № 6. (Специальный выпуск; в печати).

[141] *Шуба Ю. А.* Оценка целесообразности применения системы остаточных классов в аппаратуре обработки сигналов // Радиотехника. — 1980. — Т. 25, № 1. — С. 75–76.

[142] *Щербаков Н. С.* Достоверность работы цифровых устройств. — М.: Машиностроение, 1989. — 224 с.

[143] *Янушкевич С.* Систолические алгоритмы синтеза арифметических полиномиальных форм k-значных функций алгебры логики // Автоматика и телемеханика. — 1994. — № 12. — С. 128–142.

[144] *Aiken H. H.* The Annals of the Computation Laboratory of Harvard University // Synthesis of electronic Computing and Control Circuits. Cambridge. Massachusetts. Harvard University. — 1951. — Vol. XXVII.

[145] *Akers S. B.* On a Theory of Boolean Functions // Society for Industrial and Applied Mathematics. — 1959. — Vol. 7, № 4. — P. 487—498.

[146] *Beckmann P. E., Musicus Bruce R.* Fast fault-tolerant digital convolution using a polynomial residue number system // IEEE Trans. On Signal Processing. — 1993. — Vol. 41, № 7. — P. 2300–2313.

[147] *Bool G.* An Investigation of the Laws of Thought. — London, Walton, 1984.

[148] *Blum T., Paar C.* Montgomery modular exponentiation on reconfigurable hardware // 14-th IEEE Symp. on computer arithmetic (ARITH14). April, 1999, Proceedings. — Los Alamitos, California: IEEE Computer Soc. Press, 1999. — P. 70–77.

[149] *Dormido S., Canto M. A.* An upper bound for the syntesis of generalized parallel counter // IEEE Trans. Comput. — 1982. — Vol. C-34, № 8. — P. 802–805.

[150] *Earl E., Swartzlander J. R.* Parallel counters // IEEE Trans. Comput. — 1973. — Vol. C-22, № 6. — P. 1021—1042.

[151] *Etzel M. N., Jenkins W. K.* Redundant residue number system for error defection and correction in digital filters // IEEE Trans. On Acoust., and Speech and Signal Processing. — 1980. — FSSP-29, № 5. — P. 538–545.

[152] *Falkowski B. J.* A Note on the Polynomial Form of Boolean Functions and Related Topics. // IEEE Trans. on Computers. — 1999. — Vol. 48, № 8. — P. 860–863.

[153] *Finko O. A.* Concordant Redundant Positional Notations // Engineering Simulation. — 1997. — Vol. 14. — P. 827–832.

[154] *Finko O. A.* Number Restoration in the System of Residual Classes With a Minimum Number of Radices // Engineering Simulation. — 1999. — Vol. 16. — P. 329—334.

[155] *Finko O. A.* Check and Reconfiguration of Analog-to-Digital Devices Operating in the System of Residual Classes // Engineering Simulation. — 2001. — Vol. 18. — P. 631–543.

[156] *Finko O. A.* Methods of problem-oriented representation and data processing in resources of the hardware support of intellectual systems // IEEE Conf. Artificial Intelligence Syst. (AIS '02). Gelendzhik, Russia, September 5–10, 2002. Proceedings. — Los Alamitos, California: IEEE Computer Soc. Press, 2002. — P. 453–454.

[157] *Finko O. A.* Introduction to New Parallel Computer Arithmetics Grounded on Factorizations of Operands // Proc. International Congress "MATHEMATICS in XXI century. The role of the MMD of NSU in science, education, and business." 25–28 June 2003, Novosibirsk, Akademgorodok. http://www.sbras.ru/ws/MMF-21/.

[158] *Garner H. L.* The residue number system // Ire transactions on electronic computers. — 1959. — Vol. 8, № 6. — P. 140–147.

[159] *Garner H. L.* Namber systems and arithmetic // Advances in Computers. — 1965. — Vol. 6. — P. 131–194.

[160] *Goldereich O., Ron D., Sudan M.* Chinese remaindering with errors // IEEE Trans. On Information Theory. — 1999. — Preliminary version appered in proc. of 31st STOC 1999. — P. 225–234.

[161] *Good I. J.* The relationship between two fast Fourier transforms // IEEE Trans. on Computers. — 1971. — № 3.

[162] *Heidtmann K. D.* Arithmetic Spectrum Applied to Fault Detection for Combinational Networks // IEEE Trans. Comput. 1991. — Vol. 40, № 3. — P. 320–324.

[163] *Hiasat A. A., Abdel-Aty-Zohdy S. H.* Residue-to-binary arithmetic converter for the moduli set $(2k;\ 2^{k-1};\ 2^{k-1}-1)$ // IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing. — 1998. — Vol. 45, № 2. — P 204–209.

[164] *Ibrahim Khalid M., Saloum Salat N.* An afficient residue-to-binary converter design // At the same place. — 1988. — Cas.-35, № 9. — P. 1156–1158.

[165] *Jain J., Bitner J., Fussell D. S., Abraham J. A.* Probabilistic Verification of Boolean Functions // Formal Methods in System Design. Kluwer Academic Publishers. — 1992. — Vol. 1. — P. 61–115.

[166] *Jenkins W. K.* The design of error checkers for self-checking residue number arithmetic // IEEE Trans. On Computers. — 1983. — C-32, № 4. — P. 388–396.

[167] *Jenkins W. K., Lao S. F.* The design of an integrated RNS digital filter module based on serial-by-modulus arithmetic // IEEE Int. Cont. Comput. Design: VLSI Comput. and Process Rye Brook. NY. — 1987. — P. 634–637.

[168] *Katti S.* A new residue arithmetic error correction scheme // IEEE Trans. On Computers. — 1996. — Vol. 45, № 1. — P. 13–19.

[169] *Kawamura S., Koike M., Sano F. and Shimbo A.* Cox-Rower Architecture for Fast Parallel Montgomery Multiplication // Proceedings EUROCRYPT 2000, LNCS 1807. — Springer Verlag, 2000. — P. 523–538.

[170] *Kim Moon Soo, Tomalechi Noluhiro.* Fault tolerant digital filters using pulse-train residue arithmetic circuits // Trans. Inst. Electron. Inform. and Commun. Eng., 1987. — E 70, № 10. — P. 1009–1017.

[171] *Krishna K., Sun J.* On theori and fast algoritms for error correction in residue number system product codes // IEEE Trans. On Computers. — 1993. — Vol. 42, July. — P. 840–852.

[172] *Mandelbaum D. M.* Error correction in residue arithmetic // IEEE Trans. Comput. — 1972. — Vol. 21, № 6. — P. 538–545.

[173] *Mandelbaum D. M.* On a class of arithmetic codes and a decoding algorithm // IEEE Trans. On Information Theory. — 1976. — № 21. — P. 85–88.

[174] *Mandelbaum D. M.* Further results on decoding arithmetic residue codes // IEEE Trans. On Information Theory. — 1978. — № 24. — P. 643–644.

[175] *Mirsalehi Mir M., Shmir Joseph, Caulfield H. John.* Residue arithmetic processing utiliring optical Fredkin gate arrays // Appl. Opt. — 1987. — Vol. 26, № 19. — P. 3940–3946.

[176] *Orton G. A., Peppard L. E., Tavares S. E.* New fault tolerant techniques for residue number systems // IEEE Trans. On Computers. — 1992. — Vol. 41, № 11. — P. 1453–1464.

[177] *Paillier P.* Low-cost double-size modular exponentiation or how to stretch your cryptoprocessor. In H. Imai and Y. Zheng, editors, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC'99, LNCS-1560, — Springer Verlag, 1999. — P. 223–234.

[178] *Posch K. C., Posch R.* Residue number systems, a key to parallelism in public key cryptography // Proc. of Fourth IEEE Symp. on Parallel and Distributed Processing. — 1992. — P. 432–435.

[179] *Posch K. C., Posch R.* RNS-Modulo Reduction in Residue Number Systems // IEEE Trans. on Parallel and Distributed Systems. — 1995. — Vol. 6, № 5. — P. 449—454.

[180] *Smith J. C., Taylor F. J.* A fault tolerant GEQRNS processing element for linear systolic array DSP applications // IEEE Trans. On Computers. — 1995. — Vol. 44, № 9. — P. 1121–1130.

[181] *Soderstrand M. A., Jenkins W. K., Jullien G. A. and Tailor F. J.* Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. — N. Y.: IEEE Press, 1986.

[182] *Su C. C., Lo H. Y.* An algorithm for scaling and single residue error correction in the residue number systems // IEEE Trans. On Computers. — 1990. — Vol. 39, № 8. — P. 1053–1064.

[183] *Swartzlander E. E.* Fault tolerant arithmetic via time-shared TMR // Proc. of the SPIE, the international society for optical engineering. — 1999. — Vol. 3807. — P. 84–92.

[184] *Szabo N. S., Tanaka R. I.* Residue arithmetic and its applications to computer technology. — N. Y.: McGraw-Hill, 1967.

[185] *Tahir J. M., Dayland S. S., Gorgui-Naguib, Hinton O. R.* Fault tolerant arithmetic unit using duplication and residue codes // Integration, The VLSI Jurnal. — 1995. — Vol. 18, № 2–3. — P. 187–200.

[186] *Thornton M. A., Dreschler R., Miller D. M.* Spectral Techniques in VLSI CAD. Kluwer Academic Publishers, 2002.

[187] *Thornton M. A., Dreschler R., Miller D. M.* Spectral Techniques in VLSI CAD. Kluwer Academic Publishers, 2002.

[188] *Tkachenko A. V., Finko O. A.* Transformation and Syntesis of Complex Structures of Excessive Calcules // Atuomation and Remote Control. — 1995. — Vol. 56, № 5. — P. 765–770.

[189] *Valach M.* Vznik kodu a ciselne zbytkovych tzid Stroje na zpracovani informaci // Sbornik III. Nak. I CSAV. — 1955.

[190] *Wigley N. M., Jullien G. A.* On modulus replication for residue arithmetic computations of complex inner products. IEEE Trans. on Computers. — 1990. — Vol. 39, № 8. — P. 1065–1076.

[191] *Wigley N. M., Jullien G. A., Reaume D.* Large Dynamic Range Computations over Small Finite Rings // IEEE Trans. on Computers. — 1994. — Vol. 43, № 1. — P. 78–86.

[192] *Yanushkevich S.* Arithmetical Canonical Expansions of Boolean and MVL Functions as Generalized Reed-Muller Series // Proc. IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansions in Circuit Design. — Japan. — 1995. — P. 300–307.

[193] *Yanushkevich S., Shmerko V., Dziurzanski P.* Word Level Decision Diagrams Upon the Conditions of Linearity // IEEE Trans. Comput. Design of Integrated Syst. — 2001. — Vol. XX, month. — P. 1–40.

[194] *Yanushkevich S., Dziurzanski P., Shmerko V.* Word-Level Models for Efficient Computation of Multiple-Valued Functions, Part 1: LAR // IEEE 32th Int'l Symp. on Multiple-Valued Logic. Boston. USA. — 2002. — P. 202–208.

[195] *Zhang C. N., Shirazi B., Jun D., Y.* Residue number conversion // Explor. Technol.: Today and tomorrow. Fall Joint Comput. Conf., Dallas. Tex., 25—29 Oct., 1987. — Washington: D. C., 1987. — P. 390–396.

[196] *Zhang D., Jullien G. A., Miller W. C.* Recursive reduction in finite ring computations // 23rd Asilomar Conf. Signals, Syst. and Comput., Pasific Grove, Calif., Oct. 30 — Nov. 1, 1989: Conf. Rec., vol. 2, San Jose (Calif.). — 1989. — P. 854–857.

[197] *Zhang D., Jullien G. A., Miller W. C.* A neural-like approach to finit ring computation // IEEE Trans. Circuits and Syst. — 1990. Vol. 37, № 8. — P. 1048–1052.

[198] *Zhang D., Jullien G. A., Miller W. C.* VLSI implementation of neural-like networks for finite ring computations // Proc. 32nd Midwest Symp. Circuits and SYst., Champaign, III, Aug. 14–16, 1989. Vol 1, N. Y., 1990. — P. 485–488.

[199] *Zhang D.* Parallel VLSI neural system designs. // Springer, 1998. — P. 257.

Предметный указатель

З

И

К

Л

М

О

П

Р

С

Т

У

Ф

Ш

Ю

Научное издание

ФИНЬКО Олег Анатольевич

МОДУЛЯРНАЯ АРИФМЕТИКА ПАРАЛЛЕЛЬНЫХ ЛОГИЧЕСКИХ ВЫЧИСЛЕНИЙ: МОНОГРАФИЯ

Под редакцией *доктора технических наук, профессора В. Д. МАЛЮГИНА*

Редактор С. В. Логвиненко
Технический редактор С. С. Савчук
Оригинал-макет О. А. Финько
Корректор М. В. Ионов

Подписано в печать . .2003. Формат 60×90 1/16.
Усл. печ. л. 17,5. Уч.-изд. л. 17,7. Тираж 500 экз.
Изд № 62. Зак. № 2/62.

Совместное издательство: ин-т проблем управ. им. В. А. Трапезникова РАН и Краснодарский воен. ин-т

Множительный участок КВИ

ОПЕЧАТКИ

Стр.	Строки	Напечатано	Должно быть
20	Табл. 1.1	$x_2\ x_1$	$x_1\ x_2$
27	8-я св.	$\mathbf{Y} = [\mathbf{Y}_d\|\mathbf{Y}_{d-1}\|\dots\|\mathbf{Y}_1]^T = \dots$	$\mathbf{Y} = [\mathbf{Y}_d\|\mathbf{Y}_{d-1}\|\dots\|\mathbf{Y}_1] = \dots$
29	2-я св.	S_3-блока	S_2-блока
30	3-я сн.	$\frac{df(X)}{dx_i} = -f(x_1, \dots, \overline{x}_i, \dots, x_n)\dots$	$\frac{df(X)}{dx_i} = f(x_1, \dots, x_i, \dots, x_n)\dots$
31	3-я св. ф-ла (1.13)	$\frac{\widetilde{df}(X)}{\widetilde{dx}_i} = -f(x_1, \dots, \overline{x}_i, \dots, x_n)\dots$	$\frac{\widetilde{df}(X)}{\widetilde{dx}_i} = -f(x_1, \dots, x_i, \dots, x_n)\dots$
32	5-я св.	$f(X) = x_2\overline{x}_2 \vee \overline{x}_3$	$f(X) = x_1\overline{x}_2 \vee \overline{x}_3$
33	3-я сн.	... (1.16), в (1.18), ...	(1.16), в (1.17), ...
33	1-я сн.	$\mathbf{G} = \mathbf{A}_{2^n} * \mathbf{A}_{2^n}\mathbf{R}$	$\mathbf{G} = \mathbf{A}_{2^n} * \left(\mathbf{A}_{2^n}^{-1}\mathbf{R}\right)$
34	2-я св.	$\mathbf{R} = \mathbf{A}_{2^n}\mathbf{A}_{2^n} * \mathbf{G}$	$\mathbf{R} = \mathbf{A}_{2^n}\left(\mathbf{A}_{2^n} * \mathbf{G}\right)$
50	17-я сн.	блока S_3	блока S_2
52	10-я св. ф-ла (2.12)	$\dots\|(m-1)f(x_1, \dots, \overline{x}_i, \dots, x_n)\dots$	$\|(m-1)f(x_1, \dots, x_i, \dots, x_n)\dots$
53	8-я св.	$f(X) = x_2\overline{x}_2 \vee \overline{x}_3$	$f(X) = x_1\overline{x}_2 \vee \overline{x}_3$
60	4-я св.	$(\omega_{j,1}, \dots, \omega_{j,v}) = \{\omega_j\} = \{\omega_j\} =$	$(\omega_{j,1}, \dots, \omega_{j,v}) = \{\omega_j\} =$
104	5-я сн.	... теоремы 2.2 ...	... теоремы 5.5 ...
110	10-я св. ф-ла (5.10)	$\sum \omega_{j,\,i}^{(k)} \dots$	$\sum \omega_i^{(k)} \dots$
110	11-я св.	$\omega_{j,\,i}^{(k)} = \lvert c_i^k \rvert_m$	$\omega_i^{(k)} = \lvert c_i^{(k)} \rvert_m$
111	9-я сн.	$\omega_i = \sum \omega_{j,\,i}^{(k)} \dots$	$\omega_i^{(k)} = \sum \omega_i^{(k)} \dots$
111	8-я сн.	$\dots \sum \omega_{j,\,i}^{*(k)} \dots$	$\dots \sum \rho_{j,\,i}^{*(k)} \dots$
114	10-я сн.	Далее, (6.1)	Далее, (6.2)

Уважаемый читатель!
Приношу свои извинения за многочисленные опечатки. Но смысл, всё-же, везде – верен.
Редактор монографии – проф. Малючин В.Д. и уважаемые рецензенты – проф. Амербаев В.М. и проф. Лебедев Е.К. – все принимали активное участие в подготовке книги на протяжении многих лет. Каждый из них был самым ярким представителем своего научного направления, на которых и основывается книга.
Если книга окажется полезной, – благодарю за ссылки.

ofinko@yandex.ru;
ofinko.ru