
ЭЛЕКТРОННАЯ ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА БЭСМ

ПОД ОБЩЕЙ РЕДАКЦИЕЙ
акад. С. А. ЛЕБЕДЕВА

I

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1959

С. А. ЛЕБЕДЕВ и В. А. МЕЛЬНИКОВ

ОБЩЕЕ ОПИСАНИЕ БЭСМ И МЕТОДИКА ВЫПОЛНЕНИЯ ОПЕРАЦИЙ

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

МОСКВА 1959

АННОТАЦИЯ

В книге дается описание быстродействующей электронной счетной машины (БЭСМ), разработанной в Институте точной механики и вычислительной техники АН СССР. Даны основные параметры машины, принципы и математические основы ее работы. Рассмотрена структурная схема и взаимосвязь основных устройств машины при выполнении различных операций. Приведены принципы работы и общее описание арифметического устройства, устройства управления, запоминающих устройств, а также вводных и выводных устройств. Описываются принятые в машине операции, арифметические основы, методика и последовательность их выполнения.

Книга рассчитана на работников вычислительных центров, студентов, аспирантов и научных работников в области вычислительной математики.

Сергей Алексеевич Лебедев и Владимир Андреевич Мельников
Общее описание БЭСМ и методика выполнения операций

Редактор Ю. М. Безбородов.

Техн. редактор С. Н. Ахламов.

Корректор С. Н. Емельянова.

Сдано в набор 30/VI 1959 г. Подписано к печати 31/X 1959 г. Бумага 84×108^{1/32}.
Физ. печ. л. 6,5+1 вкл. Условн. печ. л. 10,76. Уч.-изд. л. 10,04. Тираж 15 000 экз.
Т-11033. Цена книги 7 р. Заказ 552.

Государственное издательство физико-математической литературы.
Москва, В-71, Ленинский проспект, 15

Типография № 2 им. Евг. Соколовой УПП Ленсовнархоза.
Ленинград, Измайловский пр., 29.

ОГЛАВЛЕНИЕ

Предисловие	7
Введение	9
Основные характеристики машины БЭСМ	22

ЧАСТЬ I

ОБЩЕЕ ОПИСАНИЕ БЭСМ

Глава 1. Представление чисел	24
§ 1. Кодирование чисел	24
§ 2. Представление чисел с учетом порядков	26
§ 3. Представление отрицательных чисел и отрицательных порядков	28
§ 4. Условно бесконечно большие числа и нуль	30
§ 5. Сводка данных по представлению чисел на машине БЭСМ	32
Глава 2. Система команд	34
§ 1. Операции БЭСМ	34
§ 2. Дополнительные сведения об операциях	35
Глава 3. Скелетная схема БЭСМ	56
§ 1. Основные устройства	56
§ 2. Типовая последовательность работы	64
§ 3. Последовательность работы при операциях передачи управления	67
§ 4. Последовательность работы при операциях обращения к МЗУ	71
§ 5. Контроль работы машины	74
Глава 4. Принципы построения основных устройств	78
§ 1. Стандартные блоки	78
§ 2. Арифметическое устройство	84
§ 3. Устройство управления	90
§ 4. Оперативное запоминающее устройство	96

§ 5. Внешнее запоминающее устройство	102
§ 6. Ввод исходных данных и печатание результатов вычислений	107

ЧАСТЬ II

МЕТОДИКА ВЫПОЛНЕНИЯ ОПЕРАЦИЙ

Глава 5. Арифметические операции	110
§ 1. Сложение и вычитание	110
§ 2. Умножение	136
§ 3. Деление	149
§ 4. Изменения порядков у чисел	162
§ 5. Умножение с выводом удвоенного количества рядов	165
§ 6. Деление с выводом остатка	168
Глава 6. Операции передачи кодов	174
§ 1. Передачи чисел	174
§ 2. Передача порядка числа	178
§ 3. Обмен кодами между запоминающими устройствами	181
Глава 7. Логические операции	185
§ 1. Сдвиги кодов	185
§ 2. Сложение кодов команд	187
§ 3. Циклическое сложение	189
§ 4. Выделение целой части числа	190
§ 5. Логическое умножение	194
Глава 8. Операции условных и безусловных переходов и передачи управления	198
§ 1. Сравнение чисел	198
§ 2. Операции передачи управления и безусловных переходов	202
§ 3. Операции останова машины	204
<i>Приложение 1. Список сокращенных обозначений</i>	<i>205</i>
<i>Приложение 2. Функциональная схема АУЧ</i>	<i>209</i>
<i>Приложение 3. Функциональная схема АУП</i>	<i>210</i>

ПРЕДИСЛОВИЕ

В настоящей серии книг (три выпуска) описывается универсальная цифровая электронная вычислительная машина БЭСМ и отдельные ее устройства.

Эта машина, разработанная Институтом точной механики и вычислительной техники Академии наук СССР, эксплуатируется с осени 1952 г. За время эксплуатации в машину внесен ряд усовершенствований и изменений. В настоящее время конструкция машины переработана применительно к условиям серийного производства (БЭСМ-2).

В первом выпуске описания дается общее описание машины и методика выполнения операций.

Во втором выпуске детально рассмотрены арифметическое устройство, устройство управления и типовые элементы, а в третьем выпуске — оперативное запоминающее устройство на ферритовых сердечниках и внешние запоминающие устройства на магнитных барабанах и магнитных лентах.

Книги предназначены для лиц, работающих в области вычислительной техники.

С. А. Лебедев

ВВЕДЕНИЕ

Электронные цифровые вычислительные машины являются мощным средством исследования современных задач науки и техники.

Сложные расчеты, связанные с запуском советских спутников Земли, космической ракеты, исследования в области ядерной физики немыслимы без широкого использования современных вычислительных машин.

В Советском Союзе созданы и эксплуатируются машины БЭСМ, «Стрела», М-2, М-3, «Урал», а также ряд специализированных машин; разрабатываются новые, более совершенные машины.

На этих машинах решено большое число задач в самых различных областях науки и техники. В результате их применения существенно сократились сроки проведения научных исследований и разработки образцов новой техники, снижены затраты на проведение дорогостоящих экспериментов, которые во многих случаях заменяются расчетами.

В своем выступлении на XXI съезде КПСС первый заместитель Председателя Совета Министров СССР Ф. Р. Козлов сообщил, что только за 1958 г. экономия, полученная от автоматизации вычислительных работ с помощью электронных вычислительных машин, составила около одного миллиарда рублей.

Характерными свойствами электронных цифровых вычислительных машин, обеспечившими их быстрое развитие и широкое использование, являются: универсальность применения, высокая скорость расчетов и большая точность получаемых результатов.

Универсальность, т. е. возможность решения разнообразных задач на одной и той же машине, обусловлена тем, что любая задача, решаемая на машине,

сводится к определенной последовательности простых арифметических и логических действий. В машине предусмотрены схемы для выполнения этих действий, являющихся общими для всех задач. Последовательность же выполнения действий, характерная для каждой индивидуальной задачи, а также числа, с которыми приходится оперировать, задаются программой вычислений, которая подготавливается вне машины и затем вводится в машину. Таким образом, переход от решения одной задачи к другой сводится к вводу в машину новой программы и исходных данных.

Высокая скорость расчетов получается путем полной автоматизации вычислительного процесса с помощью электронных схем автоматики. Электронные схемы, срабатывающие за доли микросекунды, позволяют выполнять на современных машинах тысячи и десятки тысяч арифметических и логических действий в секунду, что в сотни тысяч раз превосходит скорость ручного счета. Ведутся работы по дальнейшему увеличению скорости вычислений до нескольких сотен тысяч операций в секунду.

Точность вычислений при численных методах решения задач, если не затрагивать вопрос о точности, даваемой самим методом, в основном определяется количеством разрядов чисел, с которыми производятся арифметические действия. Повышение точности вычислений на машине означает увеличение числа разрядов, т. е. приводит лишь к увеличению оборудования. В этом отношении цифровые электронные вычислительные машины выгодно отличаются от электро моделирующих устройств непрерывного действия, в которых точность вычислений определяется качеством изготовления узлов, и существенное повышение ее встречает непреодолимые технологические и эксплуатационные трудности. Современные электронные цифровые вычислительные машины оперируют с 8÷12-разрядными числами, что обеспечивает необходимую точность для подавляющего типа задач. В тех случаях, когда задача требует большей точности (например, некоторые астрономические задачи), вычисления на машине могут про-

изводиться с удвоенным или утроенным числом разрядов за счет снижения скорости расчета.

Области применения электронных вычислительных машин очень разнообразны. Для математических задач они применяются в тех случаях, когда требуется провести большой объем вычислений. На современных машинах с успехом решаются задачи, требующие выполнения сотен миллионов арифметических действий. Возможность быстро выполнять расчеты в ряде случаев по новому поставила саму методику исследования.

Если раньше математические методы было невозможно применять для решения той или иной задачи ввиду их громоздкости, то при появлении электронных вычислительных машин эти методы стали практически реальными. Это вызвало, с одной стороны, развитие самих методов математических расчетов и, с другой стороны, возможность более подробного исследования различных физических процессов.

Во многих случаях расчеты необходимо производить настолько быстро, чтобы полученные результаты имели практическую ценность. Это особенно наглядно видно на примере надежного предсказания погоды на следующий день. При ручном счете вычисления, связанные с надежным суточным прогнозом погоды, могут потребовать нескольких суток. Естественно, что при такой скорости расчетов результаты теряют практическую ценность. Применение электронных вычислительных машин для этой цели позволяет полностью и своевременно решить эту задачу.

Появившаяся возможность быстрого решения разнообразных задач на электронных вычислительных машинах позволяет также использовать полученные результаты непосредственно для управления тем или иным технологическим процессом, агрегатом, поточной линией и даже целым заводом. Для этой цели создаются специализированные электронные цифровые управляющие машины, которые все больше и больше находят применение в промышленности и, особенно, в военной технике, где быстрое и точное решение определенных задач является острой необходимостью.

В последнее время, в особенности в США, электронные вычислительные машины получили широкое применение для целей статистического и бухгалтерского учета, а также для экономических расчетов. И в этой области они с успехом вытесняют существовавшие ранее для этих целей перфорационные машины.

Помимо выполнения математических задач, на электронных вычислительных машинах можно решать и логические задачи, например переводить тексты с одного языка на другой. В этом случае в машине должен храниться словарь, состоящий из слов, закодированных числами. Машина путем сравнения слов текста со словами «словаря» переводит отдельные слова. После этого с помощью грамматических и синтаксических правил, записанных в виде программы, она обрабатывает найденные слова, изменяя последние по падежам, числам, временам и т. д. и ставя их на нужное место в предложении. Полученный переведенный текст печатается на бумаге. Для успешного перевода необходима очень большая и кропотливая работа филологов и математиков по формулировке законов языка и составлению программы.

Другим примером логических задач может служить игра в шахматы. Машина путем ряда проб выбирает вариант, дающий через определенное число ходов при любых ответах противника наилучший результат. Следует, однако, заметить, что ввиду огромного количества возможных комбинаций при выборе лучшего хода, машина сейчас вынуждена ограничиваться лишь пробой сравнительно небольшого числа ходов. Ясно также, что такой способ нахождения «наилучших» ходов исключает учет стратегических планов игры.

Машина БЭСМ, описание которой дано в данной серии книг, предназначена для решения математических задач. Однако на ней с успехом решались также и некоторые логические задачи.

Для решения на машине какой-либо задачи последняя предварительно должна быть сформулирована в виде математических или логических соотношений, например, алгебраических формул, дифференциальных или интегральных уравнений и т. п. Должны быть за-

даны начальные и граничные условия, оговорены особые условия, заданы численные значения коэффициентов, входящих в формулы и т. п. Все это может быть сделано, если известна физическая сущность исследуемого процесса и изучены его количественные соотношения.

Широко развитые методы численного анализа позволяют большинство математически сформулированных задач решать посредством определенной последовательности арифметических и логических действий. На машинах каждое такое действие выполняется *командой*. В команде, помимо характера действия (сложение, вычитание и т. п.), необходимо также каким-то образом указать числа, с которыми следует произвести данное действие.

Числа в машине хранятся в запоминающем устройстве. Каждая ячейка его имеет свой номер — *адрес*. Поэтому для того, чтобы задать числа, с которыми следует произвести действие, необходимо в команде указать их адреса, т. е. номера ячеек, в которых они хранятся. Полученный после выполнения действия результат обычно используется в последующих вычислениях, поэтому он направляется в одну из ячеек запоминающего устройства и хранится в ней до тех пор, пока не потребуется в последующих вычислениях. Адрес ячейки, в которую направляется результат, также задается в команде.

Команды и числа в машине изображаются *кодами*, представляющими собой условные числа.

Так, например, иногда код команды разбивается на четыре группы — машины с трехадресной системой команд. Условное число в первой группе указывает, какая операция должна быть произведена с числами. Например, сложению приписывается код 1, вычитанию — 2, умножению — 3 и т. д. Эта группа носит название *кода операции* или *адреса операции*. В следующих двух группах указываются номера ячеек запоминающего устройства, из которых берутся числа. Эти группы называются первым и вторым *адресами команды*. В последней группе — третий адрес команды — задается номер ячейки запоминающего устройства,

в которую направляется полученный результат. Так например, код команды: 3 127 256 745 означает, что надо помножить числа, хранящиеся в 127-й и 256-й ячейках запоминающего устройства, и полученный результат направить в 745-ю ячейку.

Помимо трехадресной, применяются также одноадресные и двухадресные системы команд. Системы команд с большим числом адресов, четырехадресные или пятиадресные, не получили широкого применения.

Так как команды и числа обычно хранятся в одном и том же запоминающем устройстве, то выбор той или иной системы команд должен быть увязан с количеством разрядов чисел, с которыми оперирует машина. Для одноадресных машин обычно в одной ячейке запоминающего устройства хранятся две команды, т. е. количество разрядов кода команды примерно равно половине разрядов кода числа. Для двухадресных и трехадресных машин в одной ячейке запоминающего устройства хранится одна команда. Количество разрядов в коде команды в значительной мере определяется емкостью запоминающего устройства, так как в адресах следует предусмотреть возможность указания последней (максимальной по номеру) ячейки.

Одноадресную систему команд имеет машина «Урал». Машина М-3 выполнена по двухадресной системе. Трехадресная система принята в машинах БЭСМ, «Стрела» и М-2.

Решение какой-либо задачи, как указывалось выше, сводится к определенной последовательности арифметических и логических действий, каждое из которых выполняется одной или несколькими командами. Последовательность таких команд составляет *программу* вычислений.

Если бы для выполнения каждого арифметического действия предусматривалась своя команда, то для сложных задач, требующих сотен миллионов арифметических действий, даже написание такой программы было бы нереальным. Поэтому при программировании стремятся сравнительно небольшим количеством команд обеспечить выполнение большого числа арифметических действий.

Наиболее эффективное сокращение количества команд получается за счет циклического повторения отдельных участков программы. Так, например, при решении обыкновенных дифференциальных уравнений методом конечных разностей вычисления для каждого интервала производятся по одним и тем же формулам. Подставляя при решении в каждом интервале вместо прежних значений вновь найденные и повторяя расчеты, можно одной и той же частью программы обеспечить вычисления для любого числа интервалов.

Команды программы хранятся в оперативном запоминающем устройстве и обычно выполняются в порядке их номеров, т. е. в порядке их расположения в ячейках памяти. Для того чтобы повторить какой-либо участок программы, необходимо изменить такую последовательность выполнения команд и перейти к номеру начальной команды. Такой переход, например, может быть осуществлен специальной командой безусловного перехода, располагаемой в конце данного участка программы. Команда безусловного перехода не выполняет арифметического действия, а лишь определяет, к какому номеру команды следует перейти.

Так, располагая команду безусловного перехода, возвращающую вычисления к начальной команде, в конце программы решения обыкновенных дифференциальных уравнений методом конечных разностей, мы обеспечим повторение вычислений для неограниченного числа интервалов. Если вычисления требуется ограничить пределом, когда какая-либо величина достигнет наперед заданного значения, то вместо команды безусловного перехода в конце программы следует поставить команду условного перехода.

Команда условного перехода производит изменение последовательности выполнения команд лишь при соблюдении определенных условий. В БЭСМ условные переходы осуществляются командами сравнения. Команды сравнения сравнивают два числа, вызываемые по первому и второму адресу, и в зависимости от результата сравнения обеспечивают переход или к команде, номер которой указан в третьем адресе, или же к следующей по номеру команде.

Располагая такую команду сравнения в конце программы рассматриваемого примера, а за ней записывая команду остановки вычислений, мы обеспечим повторение расчета интервалов до тех пор, пока не будет достигнуто заданное значение сравниваемой величины. При достижении этого условия команда сравнения выберет следующую по номеру команду, т. е. остановит вычисления.

Команды условного и безусловного переходов широко используются при программировании.

В некоторых случаях процесс вычислений зависит от результата вычислений. Примером может служить расчет процесса регулирования с учетом зоны нечувствительности регулятора. До тех пор пока регулируемая величина не превысила зоны нечувствительности, вычисления должны производиться по одному закону, а после того, как регулируемая величина выйдет из зоны нечувствительности, — по другому закону. Подобный выбор программы вычислений в зависимости от результата осуществляется также командами сравнения.

При решении сложных задач многие этапы расчета оказываются общими. Примерами таких общих этапов являются: извлечение корней, нахождение значений элементарных или специальных функций, непосредственное определение изменения функций за интервал при решении систем обыкновенных дифференциальных уравнений и т. д. Решение таких общих этапов целесообразно производить по составленным заранее стандартным подпрограммам. Эти универсальные стандартные подпрограммы могут применяться при решении любых задач, в которых требуется произвести вычисления данных этапов расчета. В процессе эксплуатации машины накапливается достаточно большой набор подпрограмм для наиболее часто встречающихся этапов расчета. Каждая такая подпрограмма, включенная в машину, может рассматриваться как новая операция, выполняемая машиной.

Программа вычислений для какой-либо задачи складывается из основной программы, характерной именно для этой задачи, и отдельных стандартных подпрограмм. В соответствующих местах основной программы ма-

шина переходит на работу по подпрограмме. После окончания вычислений по этой подпрограмме необходимо вновь вернуться к вычислениям по основной программе.

Переход от основной программы к подпрограммам и обратно осуществляется на машине БЭСМ с помощью специальных команд безусловного перехода.

Во многих задачах необходимо выполнить однотипные вычисления с различными числами. Примером может служить решение дифференциальных уравнений в частных производных методом сеток. Задача решается последовательными итерациями. В каждой итерации значение функции в какой-либо точке сетки определяется через значения функции в соседних точках. Вычислив значение функции в одной точке, необходимо повторить те же самые вычисления для следующей точки и т. д. Значения функций хранятся в различных ячейках запоминающего устройства. Поэтому при повторении вычислений для следующей точки нужно выполнить те же самые команды, что и для предыдущей точки, но с другими значениями адресов. Если для каждой точки использовать свои самостоятельные команды, то программа получается весьма громоздкой. Для сокращения программы применяется метод автоматического изменения адресов.

Дело в том, что код команды представляет собой некоторое условное число и, следовательно, с ним можно производить арифметические действия. Прибавляя к коду команды наперед заданное число, можно изменять требуемым образом адреса команды. В БЭСМ для этой цели предусмотрены специальные команды.

В рассматриваемом примере после команд, выполняющих вычисления для одной точки, располагаются команды, автоматически изменяющие адреса основных команд, подготавливая их для вычисления следующей точки. Соответствующие команды сравнения обеспечивают необходимое количество повторений расчетов для следующих точек.

Составленная программа вычислений, а также исходные данные кодируются специальными устройствами в виде отверстий, пробитых (отперфорированных) в соответствующих местах на перфорационной ленте или

перфорационных картах. Затем лента или карты устанавливаются на вводное устройство и программа вместе с исходными данными переносится в запоминающее устройство, после чего начинается автоматическое выполнение заданной программы вычислений.

Для обеспечения большой скорости вычислений необходимо достаточно быстро выбирать числа и команды из запоминающего устройства. Для этой цели служит быстродействующее оперативное запоминающее устройство (ОЗУ). Емкость оперативного запоминающего устройства, т. е. количество чисел, которое может в нем храниться, в значительной мере определяет гибкость машины применительно к решению разнообразных задач. Современные машины имеют емкость оперативного запоминающего устройства порядка нескольких тысяч, а время выборки составляет несколько микросекунд.

Многие задачи требуют значительно большей емкости запоминающего устройства. Чрезмерное увеличение объема оперативного запоминающего устройства нецелесообразно, так как это приводит к значительному увеличению аппаратуры. Поэтому в машинах обычно предусматривают (помимо оперативного) дополнительное запоминающее устройство, имеющее меньшую скорость выборки и запоминания, но способное хранить до нескольких десятков и даже сотен тысяч чисел. Такие устройства часто называют внешними запоминающими устройствами.

Из внешних запоминающих устройств необходимые данные (числа и команды) передаются группами в оперативное запоминающее устройство, где с ними и производятся требуемые вычисления. Полученные результаты могут быть переданы группами обратно из оперативного запоминающего устройства во внешние. Таким образом, функция внешних запоминающих устройств сводится лишь к обмену кодами с оперативным запоминающим устройством, что при решении математических задач лишь незначительно снижает скорость вычислений, несмотря на сравнительно медленную работу внешних запоминающих устройств.

Арифметические и логические действия выполняются на арифметическом устройстве (АУ).

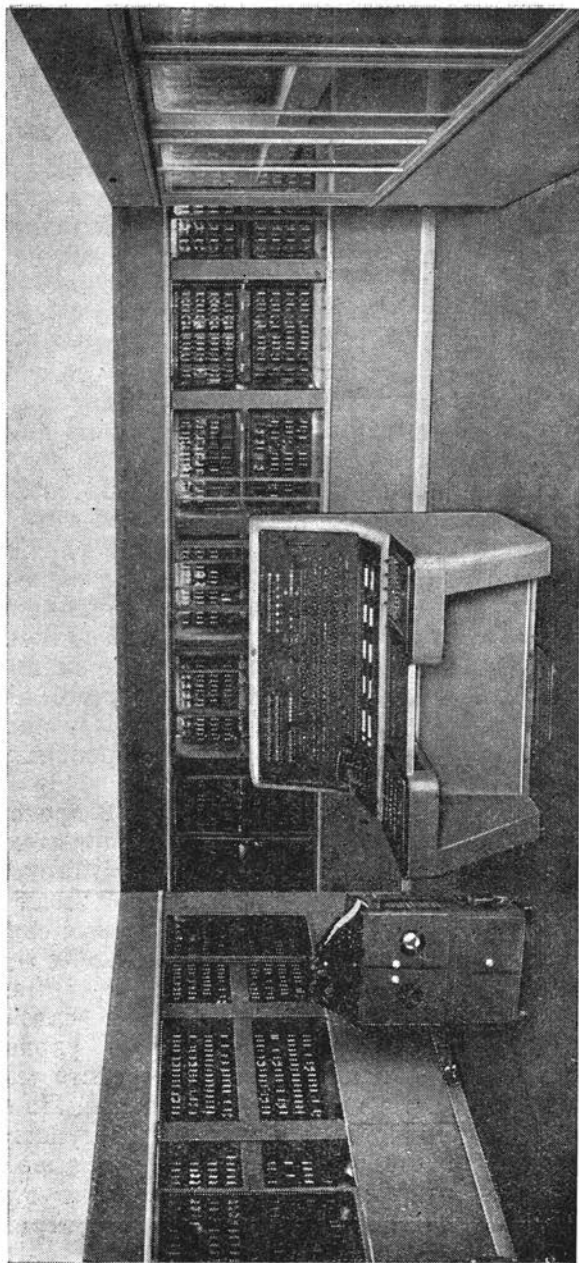


Рис. 1. «Быстродействующая электронная счетная машина» — БЭСМ.

Электронные схемы позволяют производить до нескольких десятков и даже сотен тысяч действий в секунду.

Выборка требуемых чисел из запоминающего устройства, задание действия, которое нужно произвести с этими числами, отсылка результата в запоминающее устройство и выбор следующей операции обеспечиваются на машине устройством управления (УУ). После того как введены исходные данные и программа вычислений, устройство управления обеспечивает полную автоматичность вычислительного процесса.

Окончательные результаты вычислений должны быть отпечатаны в виде таблиц с помощью специального выводного устройства.

Общий вид машины БЭСМ показан на рис. 1. На стойке прямо расположено арифметическое устройство и устройство управления, на стойке справа — оперативное запоминающее устройство, на стойке слева — внешнее запоминающее устройство, в центре — пульт управления вместе с устройством ввода-вывода.

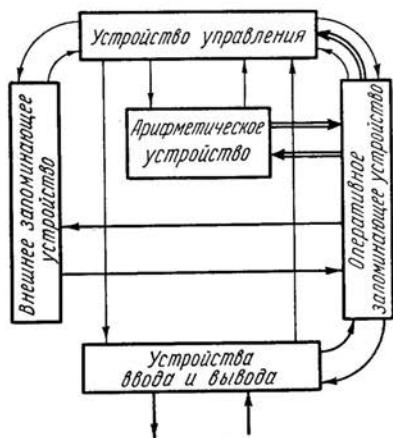


Рис. 2. Блок-схема машины БЭСМ.

По прямым связям из устройства управления поступают сигналы в какое-либо из устройств машины с целью исполнения того или иного действия. По обратным связям в устройство управления поступают сигналы, которые свидетельствуют о том, что выполнение очередного элементарного действия окончено или что очередная команда выполнена и можно переходить к следующей.

Остальные связи отображают обмен кодами между отдельными устройствами машины.

Коды передаются из вводного устройства в оперативное запоминающее устройство, а затем, если это требуется, во внешнее запоминающее устройство. В обратном порядке происходит вывод кодов из машины. Следует отметить, что при работе машины подобные обмены кодами составляют незначительный процент по сравнению с передачей кодов из оперативного запоминающего устройства в арифметическое и обратно. Поэтому связь между арифметическим и оперативным запоминающим устройствами на блок-схеме показана особо. Так же показана и дополнительная связь, идущая из оперативного запоминающего устройства в устройство управления, свидетельствующая о передаче в устройство управления кода очередной команды для ее выполнения.

Электронные вычислительные машины находятся на начальном этапе своего развития. Непрерывно идет совершенствование машин, повышается их быстродействие, увеличивается надежность, уменьшаются габариты. Одновременно расширяются области применения машин, совершенствуются и разрабатываются новые математические методы и методы программирования.

Большие перспективы вычислительной техники открываются в результате применения полупроводниковых приборов. Использование явлений сверхпроводимости, миллимикросекундной техники, сверхвысокой частоты, пленочных материалов открывает новые пути построения электронных вычислительных машин.

Грандиозная программа строительства коммунизма, принятая на XXI съезде КПСС, вновь подчеркивает большое значение вычислительной техники для развития нашего народного хозяйства. В текущем семилетии намечено увеличение выпуска вычислительных машин в 4,5—4,7 раза. Перед учеными, инженерами, конструкторами стоит почетная задача создания новых первоклассных образцов вычислительной техники и применения их в качестве мощного орудия познания человеческого разумом явлений природы и использования их на благо человечества.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МАШИНЫ БЭСМ

1. Система представления чисел — двоичная с учетом порядков. Количество разрядов для кодов чисел — 39. Цифровая часть числа — 32 разряда; знак числа — 1 разряд; порядок числа — 5 разрядов; знак порядка — 1 разряд.

Диапазон чисел, с которыми оперирует машина, примерно от 10^{-9} до 10^{+9} . Точность вычислений примерно 9 десятичных знаков (возможны вычисления с удвоенной точностью).

2. Система команд — трехадресная. Количество разрядов для кодов команд — 39. Код операции — 6 разрядов; коды адресов — 3 адреса по 11 разрядов каждый.

3. В систему операций машины входит: 9 арифметических операций, 8 операций передач кодов, 6 логических операций, 9 операций управления — всего 32 операции *).

Операции могут производиться как с нормализованными, так и с ненормализованными числами.

4. Скорость работы — $8 \div 10$ тысяч операций в секунду.

5. Система обмена кодами между арифметическим устройством и оперативным запоминающим устройством — параллельная.

6. Арифметическое устройство — одно, универсальное.

7. Управление выполнением операций — комбинированное: центральное для стандартных операций и местное для операций, время выполнения которых не укладывается в стандартный цикл центрального управления.

8. Вид оперативного запоминающего устройства — на ферритовых сердечниках. Емкость оперативного запоминающего устройства — 2047 39-разрядных чисел. Время выборки одного числа из оперативного запоминающего устройства составляет 10 мсек.

9. Внешние запоминающие устройства — магнитная запись на барабанах и лентах. Система ввода и вывода кодов в оперативное запоминающее устройство — последовательная.

10. Количество барабанов — два. Емкость запоминающего устройства на одном барабане 5120 чисел. Скорость считывания или записи с барабанов — 800 чисел в секунду. Частота импульсов магнитного барабана — около 35 кГц. Максимальное время ожидания первого числа — 80 мсек, среднее — 40 мсек.

11. Количество лент — четыре. Запись на магнитную ленту производится группами. Максимальное количество чисел в одной

*) Полный список операций, выполняемых БЭСМ, дан в главе 2.

группе — 2047. Емкость каждой ленты — 30 000 чисел (ленты можно менять). Скорость считывания или запись с лент — 400 чисел в секунду. Частота следования импульсов с магнитной ленты около 16 кГц.

Магнитные ленты имеют как прямой, так и обратный ход. Рабочим является только прямой ход.

12. Первоначальный ввод программы и исходных данных производится с перфоленты со скоростью 20 кодов в секунду.

13. Печатающие окончательных результатов производится непосредственно на бумагу со скоростью до 20 чисел в секунду.

14. Контроль выхода числа из располагаемого количества разрядов — аварийный останов машины.

Определение неисправностей — системой тестов и переводом на ручную работу.

15. Количество электронных ламп — 4000. Количество полупроводниковых диодов — 5000. Количество ферритовых сердечников около 200 000.

16. Потребляемая мощность (без вентиляции) около 35 кВт.

17. Машина состоит из следующих основных частей:

1) основной стойки, на которой располагается арифметическое устройство и устройство управления машиной;

2) стойки оперативного запоминающего устройства на ферритовых сердечниках;

3) стойки внешнего запоминающего устройства со схемой управления и магнитными лентами;

4) двух магнитных барабанов;

5) пульта управления, на котором располагаются кнопки и ключи управления, а также мнемоническая схема с сигнализацией от основных элементов машины;

6) вводного устройства с перфоленты и печатающего выводного устройства;

7) источников питания в виде мотор-генераторов и буферной аккумуляторной батареи (располагаются в самостоятельном помещении);

8) вспомогательной аппаратуры: перфораторы, контрольные, стенды для проверки элементов и т. п.

18. Электронные схемы отдельных устройств БЭСМ выполнены по мелкоблочному принципу и собираются из стандартных блоков. Стандартные блоки смонтированы в двухламповых или в четырехламповых каркасах и заканчиваются тридцатиконтактным разъемом. Блоки вставляются в стандартные панели, на обратной стороне которых монтируется схема соединений. Панели располагаются на стойках. Для облегчения условий охлаждения по высоте стойки располагаются лишь две панели. Соединение цепей между отдельными панелями и между панелями и стойкой, как правило, осуществляется через разъемы. Это облегчает наладку и сборку машины при серийном изготовлении.

ЧАСТЬ I

ОБЩЕЕ ОПИСАНИЕ БЭСМ

ГЛАВА I

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

§ 1. Кодирование чисел

БЭСМ оперирует с числами, представленными не в общепринятой десятичной системе, а в двоичной системе счисления.

В десятичной системе для написания чисел используются 10 цифр; в каждом разряде числа цифра может принимать десять значений: от 0 до 9. Число, представляемое цифрой некоторого разряда, в десять раз больше числа, представляемого аналогичной цифрой предыдущего (младшего) разряда.

В двоичной системе любое число может быть изображено с помощью всего лишь двух цифр: 0 и 1; цифра каждого следующего разряда представляет число, в два раза большее соответствующего числа для предыдущего разряда.

Последовательность целых чисел в двоичной и десятичной системах счисления запишется следующим образом:

Двоичная система

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, и т. д.

Десятичная система

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, и т. д.

В общем виде целое число в двоичной системе может быть записано так:

$$x = k_0 \cdot 2^0 + k_1 \cdot 2^1 + k_2 \cdot 2^2 + \dots + k_n \cdot 2^n,$$

где $k_0, k_1, \dots, k_n$ — цифры разрядов, принимающие значения 0 или 1. Такое изображение чисел называется *двоичным кодом* числа. Расположение разрядов в двоичном изображении такое же, как и в десятичном: старшие разряды пишутся слева от младших. Например:

Число 13, изображенное

двоичным кодом 1 1 0 1

Двоичные цифры как

множители при степени

двойки $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Сумма степеней $8 + 4 + 0 + 1 = 13$

Арифметические действия над числами, представленными в двоичной системе, производятся так же, как и в десятичной системе, надо лишь помнить, что сложение двух единиц в каком-либо разряде дает нуль в данном разряде и перенос в следующий разряд. Так, например,

$$\begin{array}{r} + 1010 \\ + 111 \\ \hline 10001 \end{array}$$

Умножение и деление в двоичной системе проще, чем в десятичной системе, так как таблица умножения ограничивается лишь умножением на 0 или 1. Например,

$$\begin{array}{r} \times 1010 \\ 101 \\ \hline 1010 \\ 0000 \\ 1010 \\ \hline 110010 \end{array}$$

Выбор двоичной системы счисления для большинства электронных счетных машин обусловлен тем обстоятельством, что при этом значительно упрощается арифметическое устройство (главным образом за счет выполнения операций умножения и деления). Кроме того, цифру каждого разряда удобно представить, например, в виде включенного или отключенного реле, наличия или отсутствия сигнала в какой-либо цепи и т. п.

Отметим здесь же, что исходные данные для расчетов, а также результаты вычислений должны быть представлены в машине в десятичной системе. Десятичные числа кодируются для представления в машине поразрядно — каждая цифра изображается в двоичной системе (так называемая двоично-десятичная система); для каждой десятичной цифры необходимо иметь четыре двоичных разряда. Например, число 978 изобразится так: 1001 0111 1000. Перевод чисел из двоично-десятичной системы счисления в двоичную и наоборот производится на машине по специальным программам.

§ 2. Представление чисел с учетом порядков

В электронной вычислительной машине числа представляются определенным количеством разрядов. Положение запятой, отделяющей целую часть от дробной, может задаваться различными способами.

В ряде машин положение запятой для всех чисел закрепляется перед некоторым определенным разрядом — это машины с фиксированной запятой. Обычно запятая фиксируется перед первым старшим разрядом, т. е. все числа при расчете должны быть меньше единицы. Последнее обеспечивается выбором соответствующих масштабов. При сложных расчетах заранее трудно определить диапазон получаемых результатов, и поэтому приходится выбирать масштабы с запасом, что ведет к снижению точности, или же предусматривать в программе вычислений автоматическое изменение масштабов, что усложняет программирование.

В некоторых машинах положение запятой указывается для каждого числа — это машины с учетом порядков, или, как их еще называют, машины с плавающей запятой. Указание положения запятой эквивалентно представлению числа в виде его цифровой части и его порядка:

$$x = 2^p A; \quad A = 0, a_1 a_2 a_3 \dots a_n,$$

где целое число p , указывающее положение запятой, — порядок, а правильная дробь A — цифровая часть числа,

Если первая цифра величины A равна 1 ($\frac{1}{2} \leq A < 1$), то представление числа x в том случае называется *нормализованным*.

В результате арифметических действий с нормализованными числами может появиться число в ненормализованной форме, т. е. в виде $2^p \cdot A_1$, причем $A_1 = 0, a_1 \dots a_k a_{k+1} \dots a_n$, где $a_1 = \dots = a_k = 0$ и $a_{k+1} = 1$. Тогда машина производит нормализацию числа, т. е. сдвигает разряды A_1 на k единиц влево и уменьшает порядок p на k , $x = 2^{p-k} \cdot A$, где $A = 0, a_{k+1} a_{k+2} \dots a_n$.

Нормализованное представление чисел с порядками повышает точность вычислений и в большинстве случаев позволяет обойтись без выбора масштабов. Эти

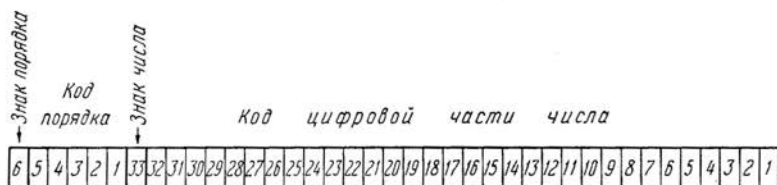


Рис. 3. Распределение разрядов кода числа.

преимущества покупаются ценой усложнения арифметического устройства.

Дело в том, что при выполнении арифметических действий с учетом порядков необходимо предусмотреть дополнительные операции с порядками. Например, при сложении и вычитании порядки необходимо предварительно выравнивать, сдвинув соответственным образом цифровые части чисел, участвующих в операции. При умножении и делении необходимо дополнительно предусмотреть сложение и вычитание порядков.

В некоторых случаях, например при действиях над командами, приходится иметь дело с числами, заданными в ненормализованном виде. Для этого в машинах предусматриваются операции с блокировкой нормализации результатов.

Числовой код БЭСМ имеет 39 двоичных разрядов. Для цифровой части числа отведено 32 двоичных

разряда и для порядков — 5 двоичных разрядов. По одному разряду выделяется для знака числа и для знака порядка (рис. 3). Запятая выбрана перед старшим разрядом цифровой части числа.

Таким образом, диапазон чисел, с которыми может оперировать БЭСМ, лежит в пределах от 2^{-32} до 2^{+31} — примерно от 10^{-9} до 10^{+9} . Каждое число может иметь 32 двоичные цифры — примерно 9 десятичных цифр. (При желании на машине можно производить вычисления с удвоенной значностью чисел за счет снижения скорости вычислений.)

§ 3. Представление отрицательных чисел и отрицательных порядков

Для учета знака чисел, представляемых в машине, необходимо иметь еще один дополнительный разряд в каждом числе. Пусть цифры 0 и 1 знакового разряда означают соответственно знаки «+» и «-». Тогда изображение положительного числа состоит из цифры 0 (изображение знака «+») и из абсолютной величины этого числа.

Для изображения отрицательных чисел употребляются различные коды.

1. Прямой код изображения отрицательного числа состоит из цифры 1 в знаковом разряде, за которой следуют цифры, отражающие абсолютную величину числа.

Прямой код отрицательного числа

$$-0, a_1 a_2 \dots a_n$$

имеет вид

$$1, a_1 a_2 \dots a_n.$$

2. Дополнительный код отрицательного числа состоит из цифры 1 в разряде знака, за которой следуют цифры, изображающие дополнение абсолютной величины числа до единицы старшего разряда. Дополнительный код отрицательного числа

$$-0, a_1 a_2 \dots a_n$$

имеет вид

$$1, b_1 b_2 \dots b_n,$$

где $0, b_1 b_2 \dots b_n = 1 - 0, a_1 a_2 \dots a_n$.

При использовании дополнительного кода сложение чисел сводится к сложению их изображений. При этом существует двоичный перенос в старший разряд — разряд знака, а из разряда знака переноса не существует. Отрицательная сумма получается всегда своим дополнительным кодом.

Существует также вариант дополнительного кода, так называемый обратный код, при котором в разряде знака отрицательного числа стоит по-прежнему единица, а в остальных разрядах находятся цифры, дополнительные к цифрам соответствующих разрядов прямого кода. Изображение числа

$$-0, a_1 a_2 \dots a_n$$

в обратном коде будет $1, \bar{a}_1 \bar{a}_2 \dots \bar{a}_n$,

где $\bar{a}_i = 1$, если $a_i = 0$, и $\bar{a}_i = 0$, если $a_i = 1$.

Дополнительный код отрицательного числа получается из обратного добавлением единицы в младший разряд.

Преобразование чисел из прямого кода в обратный несколько проще, чем в дополнительный. Однако детали производства арифметических операций (связанные, например, с округлением результатов, операцией выделения целой части и т. д.) проще реализуются в дополнительном коде.

Удобно представлять отрицательные числа дополнительным кодом для операций сложения и вычитания. Это обусловлено тем обстоятельством, что сумматор арифметического устройства производит лишь операцию сложения. Операция вычитания поэтому производится посредством прибавления дополнительного кода числа. Кроме того, если при операции сложения или вычитания результат получается отрицательным, то он будет

представлен на сумматоре в виде дополнительного кода. При представлении отрицательных чисел дополнительным кодом отрицательный результат сложения или вычитания может быть непосредственно передан в запоминающее устройство.

При представлении же отрицательных чисел прямым кодом полученный на сумматоре в виде дополнительного кода отрицательный результат должен быть предварительно преобразован в прямой код и лишь после этого передан в запоминающее устройство.

Однако более сложные операции — операции умножения и деления — удобнее производить тогда, когда отрицательные числа заданы их прямыми кодами. В этом случае операции производятся обычным путем, а знак результата получается посредством самостоятельных операций с кодами знаков.

Преобразования прямых кодов в дополнительные и обратно при операциях сложения и вычитания производить проще, чем аналогичные преобразования при умножении и делении. Поэтому для БЭСМ принято представлять цифровые части чисел прямыми кодами.

С порядками чисел производятся лишь операции сложения или вычитания. Поэтому для БЭСМ отрицательные порядки представляются дополнительными кодами.

§ 4. Условно бесконечно большие числа и нуль

При производстве арифметических действий над числами может получиться, что результат вычислений не укладывается в диапазон чисел, представляемых на машине. Если результат вычислений по абсолютной величине больше, чем максимальное число, которое может быть представлено на машине, то условно будем считать, что результат равен бесконечности. Если же результат вычислений меньше, чем минимальное число, которое может быть представлено на машине, то можно полагать, что результат равен нулю.

При составлении программ следует предусматривать, чтобы числа, получаемые в результате вычислений, не стали «условно бесконечными», т. е. не превысили по абсолютной величине максимального числа, изображае-

мого на машине. Образование условно бесконечно большого числа может поэтому возникнуть лишь в результате неправильного программирования поставленной задачи или же оно указывает на нецелесообразность производства дальнейших операций. В БЭСМ появление таких чисел вызывает остановку машины.

В результате действий могут возникать также «условно бесконечно малые числа», т. е. числа, меньшие по абсолютной величине, чем минимальное число, которое может быть представлено на машине. В случае появления такого числа машина оперирует с ним, как с нулем, что соответствует методике приближенных вычислений с ограниченным числом знаков.

Возможны два способа восприятия нуля машиной.

1. Число принимается равным нулю, если его порядок имеет минимальное значение, возможное на машине вне зависимости от кода самого числа, т. е. код порядка 1.00000 (равный —32) показывает, что число равно нулю.

2. Число принимается равным нулю, если его порядок имеет минимальное значение, возможное на машине, и код самого числа равен нулю, т. е. число равно нулю, когда код порядка будет 1.00000 и код числа будет 0.000 . . . 0.

Второй способ расширяет диапазон в сторону низших порядков на 32 разряда. С другой стороны, при наличии наинизшего порядка у числа (код порядка 1.00000) код самого числа в общем случае уже не будет нормализованным. Это усложняет производство арифметических операций, в первую очередь деления, а также схему блокировки. По существу, выигрыш в отношении увеличения диапазона порядков может быть получен и при первом способе индикации нуля путем добавления одного разряда в коде порядка; при этом расширение диапазона происходит на 32 разряда как в сторону низших, так и высших порядков.

Дополнение одного разряда в коде порядка, если это необходимо, лучше, чем производство операций с ненормализованными числами, имеющими наинизший порядок, по следующим причинам:

1) увеличивается рабочий диапазон машины как в сторону низших, так и высших разрядов;

2) операции с малыми числами производятся с полным объемом значащих цифр;

3) упрощается производство операций и блокировка.

В соответствии с этим выбран первый способ восприятия нуля, т. е. число принимается равным нулю, если вне зависимости от кода самого числа его порядок имеет минимальное значение, возможное на машине, т. е. значение 1.00000. В дальнейшем это значение порядка именуется как «код нуля» (или код «0»). Не следует смешивать термин «код нуля», например, с термином «нуль порядка», когда порядок равен нулю (0.00000).

§ 5. Сводка данных по представлению чисел на машине БЭСМ

1. Числа на машине изображаются:

а) кодом, представляющим цифровую часть числа — 32 разряда;

б) кодом, характеризующим знак числа — один разряд;

в) кодом, представляющим порядок числа — 5 разрядов;

г) кодом, характеризующим знак порядка — один разряд.

2. Цифровая часть числа представляется прямым кодом, а знак числа характеризуется кодом знака (код знака 0 соответствует «+», код знака 1 соответствует «—»).

3. Порядок числа представляется дополнительным кодом и кодом знака порядка (аналогично знаку цифровой части).

4. Положение запятой в цифровой части числа принимается перед высшим разрядом, т. е. все цифровые части чисел меньше единицы.

5. Коды чисел, как правило, представляются в нормализованном виде, т. е. первая значащая цифра занимает высший разряд (цифровая часть больше или равна $1/2$). При производстве арифметических действий результат нормализуется.

6. Предусматривается возможность производства операций с ненормализованными числами и исключения нормализации результата.

7. При получении в результате арифметических действий числа, большего максимально представимого на машине (2^{31}), машина автоматически останавливается («аварийный останов»).

8. При получении числа, меньшего минимально представимого числа на машине (2^{-32}), это число принимается за нуль, выражаемый как код порядка 1.00000.

ГЛАВА 2

СИСТЕМА КОМАНД

§ 1. Операции БЭСМ

В БЭСМ принята трехадресная система команд (рис. 4). Каждый адрес (*A1*, *A2*, *A3*) имеет 11 двоичных разрядов, что соответствует емкости оперативного запоминающего устройства в 2047 ячеек (нулевой адрес не используется и при наличии его в адресе команды из оперативного запоминающего устройства поступает машинный ноль).

Для кода операции выделено 6 двоичных разрядов (*AOп*), что дает возможность иметь до 63 типов команд.

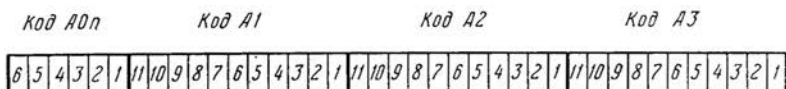


Рис. 4. Распределение разрядов кода команды.

Из этих 6 разрядов, как правило, непосредственно для кодов команд используется 5 разрядов. Шестой разряд кода операций служит для блокировки нормализации результата в операциях с № 1 по № 16 (см. ниже). В других операциях он используется для получения дополнительных операций или вообще не используется.

Код операции определяет, какое действие следует произвести. В первом и втором адресах в большинстве команд указываются номера ячеек оперативного запоминающего устройства, в которых хранятся два числа, с которыми производится заданное действие. В третьем адресе задается номер ячейки запоминающего устройства, в которую посылается полученный результат.

Набор операций, принятый в машине, приведен в таблице 1. В таблице, помимо характеристики данной операции, указано также время, необходимое для ее выполнения. Это время включает выборку команды и чисел из запоминающего устройства, выполнение заданного действия и посылку результата в запоминающее устройство.

Операции, предусмотренные в машине, можно разбить на следующие группы.

Арифметические операции — сложение, вычитание, умножение, деление, умножение с выводом удвоенного количества разрядов, деление с выводом остатка, сложение порядков, вычитание порядков, изменение порядка по адресу.

Операции передачи кодов — передача числа нормальная, передача числа на печать, передача числа с регистров пульта управления, передача числа с изменением знака, передача числа по абсолютной величине, передача числа с изменением знака в зависимости от знака другого числа, передача порядка числа, обмен кодами между оперативным и внешними запоминающими устройствами.

Логические операции — сдвиг с блокировкой порядков, сдвиг по всем разрядам, сложение кодов команд, циклическое сложение, выделение целой части, логическое умножение.

Операции управления — сравнение двух чисел с учетом их знаков, сравнение двух чисел на равенство, сравнение двух чисел по абсолютной величине, передача на местное управление командами без гашения, передача на центральное управление командами без гашения, изменение номера команды на местном управлении командами, изменение номера команды на центральном управлении командами, останов условный, останов.

§ 2. Дополнительные сведения об операциях

Для большинства операций в таблице 1 приведены необходимые данные для определения их конкретного смысла и назначения. Но для некоторых операций требуются дополнительные пояснения.

Таблица 1

Операции, принятые на машине БЭСМ

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мксек
1	0	00001	+	Сложение	Число, номер ячейки которого указан в первом адресе команды, складывается с числом, номер ячейки которого указан во втором адресе. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.	65 ÷ 225
	1	00001	+,*)	Сложение с блокировкой нормализации		
2	0	00010	—	Вычитание	Из числа, номер ячейки которого указан в первом адресе команды, вычитается число, номер ячейки которого указан во втором адресе команды. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.	65 ÷ 225
	1		—	Вычитание с блокировкой нормализации		
					Перед вычитанием чисел выравниваются их порядки. После вычитания производится нормализация и округление результата.	

3	0	00011	×	Умножение	Число, номер ячейки которого указан в первом адресе команды, умножается на число, номер ячейки которого указан во втором адресе команды. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. При умножении порядки чисел складываются, а коды чисел умножаются. После умножения производится нормализация результата и его округление. В запоминающее устройство выводятся старшие 32 разряда произведения.	240
	1		×	Умножение с блокировкой нормализации		
4	0	00100	:	Деление	Число, номер ячейки которого указан в первом адресе команды, делится на число, номер ячейки которого указан во втором адресе команды. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. При делении порядки чисел вычитаются, а коды чисел делятся. Результат всегда получается нормализованным вне зависимости от блокировки нормализации. После деления производится округление результата.	250

*) В случае блокировки нормализации результата перед условным обозначением операции ставится запятая. Характеристика операции дается для случая отсутствия блокировки.

Продолжение

№ операции			6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мксек
5	0	00101			+	Сложение порядков	К порядку числа, номер ячейки которого указан в первом адресе команды, прибавляется порядок числа, номер ячейки которого указан во втором адресе команды. Первое число с измененным порядком нормализуется и направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.	65
	1				+, П	Сложение порядков с блокировкой нормализации		
6	0	00110			— П	Вычитание порядков	Из порядка числа, номер ячейки которого указан в первом адресе команды, вычитается порядок числа, номер ячейки которого указан во втором адресе команды. Первое число с измененным порядком нормализуется и направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.	65
	1				—, П	Вычитание порядков с блокировкой нормализации		
7	0	00111			ИПА	Изменение порядка по адресу	К порядку числа, номер ячейки которого указан в первом адресе команды, прибавляется код второго адреса команды. Первое число с измененным порядком нормализуется и передается	65
	1				,ИПА	Изменение порядка по адресу с блокировкой нормализации		

8	0	01000	$\times(a)$	Умножение с выводом удвоенного количества разрядов	в ячейку запоминающего устройства, номер которой указан в третьем ад- ресе команды.
	1		$\times(a)$	Команда умножения — a Команда умножения — a с блокировкой норма- лизации	Операция производится в две команды.
240					Первая команда: число, номер ячейки которого указан в первом ад- ресе команды, умножается на число, номер ячейки которого указан во вто- ром адресе команды. Производится нормализация результата и первые 32 разряда произведения с их порядком направляются в ячейку запоминаю- щего устройства, номер которой ука- зан в третьем адресе команды.
9	0	01001	$\times(b)$	Команда умножения — b Команда умножения — b с блокировкой норма- лизации	Округление блокируется.
	1		$\times(b)$		Вторая команда: последние 32 разряда произведения с порядком, оставшимся от команды $\times(a)$, нор- мализуются и направляются в ячейку запоминающего устройства, номер ко- торой указан в третьем адресе.
65					Адреса первый и второй в данной коман- де не используются. Фактический по- рядок результата будет на 32 меньше имеющегося порядка. Округление бло- кируется.

Продолжение

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мсек
10	0	01010	:(a)	Деление с выводом остатка Команда деления — a	Операция производится в две команды. Первая команда: число, номер ячейки которого указан в первом адресе команды, делится на число, номер ячейки которого указан во втором адресе команды. Частное направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. После деления округление частного не производится. Результат всегда получается нормализованным.	250
11	0	01011	:(б) :(б)	Команда деления — б Команда деления — б с блокировкой нормализации	Вторая команда: остаток, имеющий порядок и знак делимого, направить в ячейку запоминающего устройства, номер которой указан в третьем адресе второй команды. Во втором адресе этой команды указывается номер ячейки запоминающего устройства, в которой хранится делимое. Если необходимо, то может быть произведена нормализация остатка. Первый адрес не используется.	65

12	0	01100	ПЧ	Передача числа нормальная	Число, номер ячейки которого указан в первом адресе команды, нормализуется и передается в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. Второй адрес не используется.	65
	1		ПЧ	Передача числа нормальная с блокировкой нормализации		
12а	1	01100	ПЧТ	Передача числа на печать	Число, номер ячейки которого указан в первом адресе команды, передается на печатающее устройство и печатается. Настоящая команда отличается от предыдущей наличием кода 1 в 10-м разряде второго адреса и обязательной блокировкой нормализации.	65
126	0	01100	ПЧР	Передача числа с регистров пульта управления	Команда «Передача числа с регистров пульта управления» отличается от операции «Передача числа нормальной» наличием кода 1 в 1-м, 2-м или 3-м разрядах второго адреса.	
	1		ПЧР	Передача числа с регистров пульта управления с блокировкой нормализации	Выбор регистра задается кодом второго адреса: 3-й разряд — триггерный контрольный регистр 2-й разряд — 2-й диодный регистр 1-й разряд — 1-й диодный регистр	
13	0	01101	ПЧ —	Передача числа с изменением знака	Число, номер ячейки которого указан в первом адресе команды, нормализуется и передается с изменением его знака в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.	65
	1		ПЧ —	Передача числа с изменением знака и с блокировкой нормализации	Второй адрес не используется.	

Время зависит от действия печатающего устройства.

Продолжение

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мксек
14	0	01110	ПЧ	Передача числа по абсолютной величине	Число, номер ячейки которого указан в первом адресе команды, нормализуется и передается по абсолютной величине в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. Второй адрес не используется.	65
	1	, ПЧ	Передача числа по абсолютной величине с блоковой нормализацией			
15	0	01111	ПЧ ±	Передача числа с изменением знака в зависимости от знака другого числа	Число, номер ячейки которого указан в первом адресе команды, нормализуется и передается в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. Знак передаваемого числа изменяется на обратный, если знак числа, номер ячейки которого указан во втором адресе, отрицательный, и не изменяется, если знак этого числа положительный.	65
	1	, ПЧ ±	Передача числа с изменением знака в зависимости от знака другого числа с блоковой нормализацией			
16	0	10000	↓	Передача порядка числа	Порядок числа, номер ячейки которого указан в первом адресе команды, представляется в виде нормализованного числа со своим порядком и направляется в ячейку запоминающего	65 + 225
	1	, ↓	Передача порядка числа с блоковой нормализацией			

17	0	10001	←	Сдвиг с блокировкой по- рядков	устройства, номер ячейки которой указан в третьем адресе команды. Второй адрес не используется. Код числа, номер ячейки которого указан в первом адресе команды, сдвигается на количество разрядов, указанных во втором адресе команды. Результат направляется в ячейку за- поминающего устройства, номер кото- рой указан в третьем адресе команды. Сдвиг производится как влево, так и вправо в зависимости от кода 7-го разряда второго адреса (код 1 соот- ветствует сдвигу вправо, код 0 — сдвигу влево). При сдвиге кода числа код его порядка не сдвигается и в запоминающее устройство не передается. Знак числа сохраняется и передается в запоминающее устройство вместе с резуль- татом.	65 ÷ 225
17a	1	10001	←, ←	Сдвиг по всем разрядам	Операция «Сдвиг по всем разрядам» от- личается от операции «Сдвиг с блоки- ровкой порядков» тем, что сдвигается и число и порядок. В зависимости от направления сдвига порядок может быть сдвинут на разряды числа, а число — на разряды порядка. Напра- вление сдвига определяется 7-м раз- рядом второго адреса, как и в преды- дущей команде.	65 ÷ 225

Продолжение

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполне- ний операции в мксек
18	0	10010	СК	Сложение кодов команд	Число, номер ячейки которого указан в первом адресе команды, складывается с числом, номер ячейки которого указан во втором адресе команды. Результат с порядком первого числа направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. Складываются только коды цифровых частей чисел и коды знаков; порядок второго числа не учитывается, т. е. при сложении порядки чисел не выравниваются. Нормализация результата не производится.	65
18a	1	10010	СК	Циклическое сложение	Число, номер ячейки которого указан в первом адресе команды, складывается с числом, номер ячейки которого указан во втором адресе команды. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. При сложении числа, их порядки и знаки рассматриваются как одно целое, т. е. перенос из разряда знака числа производится в 1-й разряд	65

19*) 0	10011	ЦЧ	Выделение целой части	У числа, номер ячейки которого указан в первом адресе команды, выделяется целая часть. Целая часть в виде числа с фиксированной запятой после младшего разряда и со своим знаком направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды. Дробная часть числа, приведенная к нулевому порядку и имеющая всегда положительный знак, направляется в ячейку запоминающего устройства, номер которой указан во втором адресе команды.	65 ÷ 225
20	0	10100	<	Сравнение двух чисел с учетом их знаков	65
				Сравнивается число, номер ячейки которого указан в первом адресе команды, с числом, номер ячейки которого указан во втором адресе команды. Если первое число меньше второго числа, то следующей выполняется команда, номер которой указан в третьем адресе. Если первое число больше или равно второму числу, то выполняется следующая по номеру команда.	
				Сравнение чисел производится с учетом их знаков.	

*) В операциях 19 и 21 ÷ 31 наличие кода операции на выполнение команды не влияет.

Продолжение

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мксек
20a	1	10100	$<$	Сравнение двух чисел на равенство	Сравнивается число, номер ячейки которого указан в первом адресе команды, с числом, номер ячейки которого указан во втором адресе команды. Если числа равны, то выполняется следующая по номеру команда. Если числа не равны, то следующей выполняется команда, номер ячейки которой указан в третьем адресе.	65
21	0	10101	$ < $	Сравнение двух чисел по абсолютной величине	Сравнивается по абсолютной величине число, номер ячейки которого указан в первом адресе команды, и число, номер ячейки которого указан во втором адресе команды. Если первое число меньше второго числа, то следующей выполняется команда, номер ячейки которой указан в третьем адресе. Если первое число больше или равно второму числу, то выполняется следующая по номеру команда.	65
				Операция обращения к внешним запоминающим устройствам *)	Операция выполняется в две команды.	

22	0	10110	M3 (a)	Обращение к внешнему запоминающему устройству («Магнитная запись») — а	Первая команда. В первом адресе указывается характер операции: а) запись, считывание или перемотка, б) барабан или лента, в) номер группы магнитного барабана или номер магнитной ленты (или перфоленты). Во втором адресе указывается номер ячейки на барабане или номер группы на ленте, с которых начинается операция. В третьем адресе указывается номер ячейки в оперативном запоминающем устройстве, с которой начинается операция.	65
23	0	10111	M3 (б)	Обращение к внешнему запоминающему устройству («Магнитная запись») — б	Вторая команда. Во втором адресе указывается номер ячейки на барабане, на которой кончается операция, или число кодов в группе при обращении к ленте. В третьем адресе указывается номер ячейки во внутреннем запоминающем устройстве, куда записывается содержимое «контрольного счетчика» (см. ниже). Первый адрес команды не используется. При операции перемотки магнитной ленты все три адреса свободны.	Время за- висит от количе- ства пе- редавае- мых чи- сел.

*) Обращение к вводному устройству на перфоленте осуществляется таким же образом, как и к магнитной ленте при считывании.

Продолжение

№ операции	6-й разряд кода операции	Код операции	Обозначение операции	Название операции	Характеристика операции	Время выполнения операции в мксек
24	0	11000	ПМУК	Передача на местное управление командами без гашения	Передается управление командами с центрального на местное. Местное управление при этом не гасится и выполняются команды, начиная с номера ячейки, стоящего на местном управлении.	65
25	0	11001	ПЦУК	Передача на центральное управление командами без гашения	Передается управление командами с местного на центральное. Центральное управление при этом не гасится, и выполняются команды, начиная с номера ячейки, стоящего на центральном управлении.	65
26	0	11010	ИМУК	Изменение номера команды на местном управлении командами	Если работа производилась на центральном управлении, то она переходит на местное управление и выполняются команды, начиная с номера ячейки, который указан в третьем адресе.	65
27	0	11011	ИЦУК	Изменение номера команды на центральном управлении командами	Если работа производилась на местном управлении, то она продолжается, но начиная с команды, номер которой указан в третьем адресе. Если работа производилась на местном управлении, то она переходит на центральное управление и выполняются	65

28	0	11100	<i>Ост Усд</i>	Останов условный	команды, начиная с номера ячейки, который указан в третьем адресе. Если работа производилась на центральном управлении, то она продолжается, но начиная с команды, номер которой указан в третьем адресе. По второму адресу в запоминающее устройство засылается команда, имеющая код операции ИЦУК, а в третьем адресе — увеличенный на единицу номер ячейки текущей команды.	65
29	0	11101	Λ	Логическое умножение	Производится останов машины в случае, если включен специальный тумблер на пульте управления.	65
30	0	11110			Производится поразрядное умножение двух чисел, номера ячеек которых указаны в первом и втором адресах команды. Результат направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе. Умножение распространяется на все 39 разрядов.	65
31	0	11111	<i>Ост</i>	Останов	Данный код операции в машине не используется.	65
32	0	00000			Производится останов машины с соответствующей сигнализацией Данный код операции используется для передачи числа с блокировкой нормализации.	65

Операции с порядками. В машине предусмотрены три разновидности команд при операциях над порядками: «Сложение порядков», «Вычитание порядков» и «Изменение порядка по адресу».

Рассмотрим более детально выполнение последней команды, так как при ее осуществлении имеются некоторые особенности.

В первом адресе этой команды задается адрес числа, у которого необходимо изменить порядок. Во втором адресе команды младшие шесть разрядов отводятся под код той величины, на которую изменяется порядок (остальные разряды на выполнение операции не влияют). При выполнении операции «Изменение порядка по адресу» происходит прибавление величины, стоящей во втором адресе, к порядку первого числа.

Если необходимо уменьшить порядок, то величина во втором адресе указывается дополнительным кодом (с кодом 1 в 6-м разряде адреса).

Пусть, например, необходимо у числа

порядок	цифровая часть
0.00111	0.1010 ... 00

уменьшить его порядок на 5. Дополнительный двоичный код числа 5 указывается во втором адресе команды «Изменение порядка по адресу», который в этом случае будет иметь вид 00000111011.

После выполнения операции результат, отсылаемый в запоминающее устройство, будет иметь следующий вид:

0.00010	0.1010 ... 00.
---------	----------------

Передача числа нормальная. Одна из модификаций команды «Передача числа нормальная» (12a) используется для передачи результата на печатающее устройство. При печати результата работа машины останавливается до завершения печати.

Другая модификация команды передачи (126) предусмотрена для передачи чисел с триггерного или диодных регистров. Коды на этих регистрах устанавливаются кнопками с пульта управления. Этой операцией

удобно пользоваться для ввода в машину новых величин в ходе вычислительного процесса, даже не останавливая машину.

Передача порядка. В ряде случаев необходимо иметь порядок числа, представленный в виде обычного числа с его порядком, с тем, чтобы с ним можно было производить дальнейшие арифметические действия.

Максимальное число, которое может быть записано в разрядах порядка, равно 0.11111, т. е. 31. А порядок для данного максимального кода будет 5 (0.00101). Поэтому при выполнении команды «Передача порядка» в разрядах порядка устанавливается код 0.00101, а в старшие пять разрядов числа засылается код порядка.

Например, порядок 0.01101 представится в виде числа таким образом:

0.00101 0.0110100 ... 00.

Если в команде предусмотрена блокировка нормализации, то число в таком же виде запишется в запоминающее устройство.

В случае, если блокировка нормализации отсутствует, то порядок отсылается в запоминающее устройство лишь после того, как результат операции приведет к нормализованному виду. В данном примере мы получим:

0.00100 0.110100 ... 00.

Сдвиг с блокировкой порядка. При этой операции сдвигается только цифровая часть числа. Порядок со своим знаком, а также знак числа в операции сдвига не участвуют. Знак отсылается в запоминающее устройство в прежнем виде, а порядок числа вообще не передается.

Количество разрядов, на которое надо сдвинуть цифровую часть, задается прямым кодом во втором адресе команды сдвига. Направление сдвига определяется 7-м разрядом. Код 0 в этом разряде соответствует сдвигу влево, а код 1 — сдвигу вправо.

Рассмотрим пример. Допустим, что цифровую часть числа

0.01011 1.11010000 ... 00

необходимо сдвинуть вправо на три разряда, т. е. во втором адресе (A2) стоит код 00001000011.

После выполнения операции число отсылается в запоминающее устройство в следующем виде:

0.00000 1.00011010 ... 00.

В случае сдвига влево 7-й разряд второго адреса имеет код 0, а в запоминающее устройство в нашем случае отсылается результат следующего вида:

0.00000 1.10000000 ... 00.

Сдвиг по всем разрядам. При этой операции сдвигаются все разряды числа, включая порядок, знак порядка и знак числа.

Пусть, например, требуется сдвинуть все разряды числа

0.01011 0.11010000 ... 00

влево на два разряда, т. е. второй адрес (A2) имеет код 00000000010.

Результат, отсылаемый в запоминающее устройство, имеет вид

1.01101 1.01000000 ... 00.

При сдвиге вправо на два разряда имеем:

0.00010 1.10110100 ... 00.

Сложение команд. Операция «Сложение кодов команд» введена для изменения адресов команд. В этой операции к адресам команды, вызываемой по первому адресу, прибавляется код, вызываемый по второму адресу. Если рассматривать команду, вызванную по первому адресу, как число, то в операции сложения участвуют все разряды цифровой части числа и его знак, а порядок и знак порядка (соответствуют коду операции) отсылаются в запоминающее устройство в прежнем виде. Например:

Код числа, выбранный		
по адресу A1	0.01100	0.11010000 ... 00
Код числа, выбранный		
по адресу A2	0.00011	0.01000000 ... 00.

Полученный результат имеет следующий вид:

0.01100 1.00010000 ... 00.

Порядок второго числа в операции не участвует. Выравнивания порядков, нормализации результата и округления не производится.

Циклическое сложение. В этой операции происходит равноправное сложение кодов всех 39 разрядов с циклическим переносом из старшего разряда в младший. ~

Если имеется перенос из разряда знака числа, то он поступает в первый разряд порядка, а перенос из разряда знака порядка поступает в первый разряд числа. Например:

1-й код	1.11011	1.11010000 ... 00
2-й код	0.01001	1.11100000 ... 01
Результат	0.00101	1.10110000 ... 10

Данная операция используется преимущественно для целей контроля. Так, например, посредством циклического сложения можно получить контрольную сумму результатов какого-либо этапа расчетов. Произведя повторный расчет с подсчетом новой контрольной суммы, можно убедиться в правильности расчета путем сравнения полученных контрольных сумм.

Выделение целой части. Особенностью данной операции является то, что выделенная целая часть числа располагается в младших разрядах цифровой части числа, отосланного на запоминание по третьему адресу. Дробная часть, запоминаемая по второму адресу, всегда положительна и приведена к нулевому порядку, т. е. может быть и в ненормализованном виде.

Например, после выделения целой части числа 26,25 имеющего вид

0.00101 0.110100100 ... 00,

мы получим целую часть (26)

0,00000 0.00 ... 011010

и дробную часть (0,25)

0.00000

0.010 ... 00.

Целая часть отрицательного числа — 26,25 будет иметь вид

0.00000

1.00 ... 011011

(т. е. — 27) и дробная часть (0,75) —

0.00000

0.110 ... 00.

Обращение к внешним запоминающим устройствам. Для выполнения операции обмена кодами между оперативными и внешними запоминающими устройствами используются две команды $MЗ(a)$ и $MЗ(b)$. Команда $MЗ(a)$ является подготовительной, а $MЗ(b)$ — исполнительной.

Команду $MЗ(a)$ определяют следующие коды:

1. Код операции $MЗ(a)$ — 10110 в разрядах с 34-го по 38-й кода команды.

2. Код адреса $A1$. В коде этого адреса указывается характер операции, номер барабана, номер группы на барабане или номер ленты. Разряды адреса распределены следующим образом:

1-й, 2-й и 3-й разряды отведены для указания номера группы на барабане или номера ленты.

4-й разряд определяет, к какому из барабанов производится обращение. Код 0 в этом разряде соответствует одному барабану, а код 1 — другому.

5-й разряд — наличие кода 1 в этом разряде определяет перемотку ленты со считыванием определенной группы.

6-й разряд не используется.

7-й разряд — наличие кода 1 определяет перемотку ленты без считывания.

8-й разряд — наличие кода 1 определяет обращение к магнитным лентам.

9-й разряд — наличие кода 1 определяет обращение к барабанам.

10-й разряд определяет характер передачи кодов; считывание (код 0) или запись (код 1),

3. Код адреса *A2*. При обращении к одному из барабанов в разрядах с 1-го по 10-й указывается номер ячейки на барабане, с которой следует начать считывание или запись. При обращении к одной из лент в разрядах с 1-го по 6-й указывается номер группы, которую следует считать или записать, или номер группы, до которой следует произвести перемотку.

4. Код адреса *A3*. В разрядах этого адреса указывается номер ячейки в оперативном запоминающем устройстве, с которой следует начать запись или считывание.

Команды *M3(б)* определяют следующие коды:

1. Код операции *M3(б)* — 10111.

2. Код адреса *A1* — разряды этого адреса в данной операции обмена кодами не используются. Они могут быть использованы для других целей.

3. Код адреса *A2*. Указывается номер ячейки магнитного барабана, на которой следует закончить передачу кодов, или количество кодов, которое следует считать с магнитной ленты или записать на нее.

4. Код адреса *A3*. В разрядах этого адреса располагается номер ячейки оперативного запоминающего устройства, в которой следует записать показания «контрольного счетчика». Счетчик подсчитывает количество разрядов в передаваемой информации, имеющих код 1. Он состоит из пяти разрядов, и подсчет ведется по модулю 32. Код из счетчика располагается в младших пяти разрядах цифровой части числа.

Более подробные сведения о всех операциях даны во второй части настоящего выпуска (главы 5—8), где рассматривается методика производства операций на машине.

ГЛАВА 3

СКЕЛЕТНАЯ СХЕМА БЭСМ

§ 1. Основные устройства

В состав электронных вычислительных машин, как правило, входят следующие основные устройства:

А. Арифметические устройства, на которых производятся арифметические действия над кодами чисел, а также другие операции с кодами чисел и кодами команд.

Б. Запоминающие устройства, в которых хранятся коды чисел и команд. Эти коды в заданные моменты времени могут быть переданы из запоминающих устройств в другие устройства машины и обратно.

В. Устройства управления, обеспечивающие автоматичность вычислительного процесса. Эти устройства производят выборку требуемых чисел из запоминающих устройств, задают действие, которое нужно произвести с этими числами, осуществляют отсылку результата в запоминающие устройства, выбирают следующую команду.

Г. Вводные и выводные устройства, служащие для ввода исходных данных и программы вычислений, а также для печатания полученных результатов вычислений.

На машине БЭСМ принято одно универсальное арифметическое устройство (АУ), выполненное на триггерных ячейках (рис. 5). На этом арифметическом устройстве производятся все действия над кодами чисел и кодами команд.

Так как представление чисел производится с учетом порядков, а действия с цифровыми частями чисел и с их

порядками отличны друг от друга, то арифметическое устройство разделяется на две части: арифметическое устройство чисел (АУЧ) и арифметическое устройство порядков (АУП). Каждое из этих арифметических устройств имеет по сумматору (СмЧ и СмП), на которых производятся арифметические действия с кодами

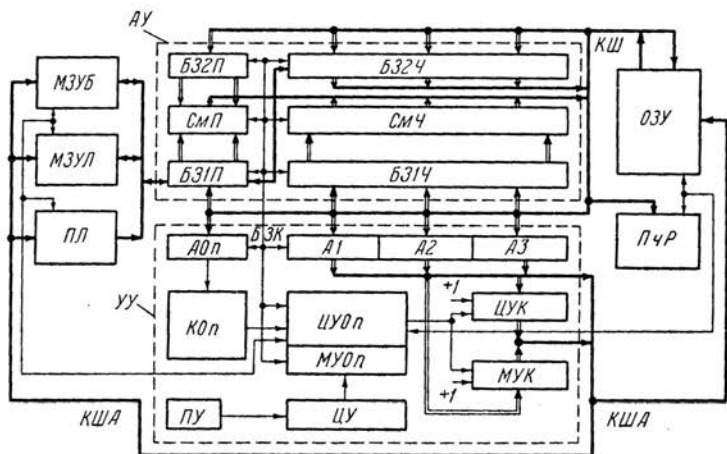


Рис. 5. Скелетная схема БЭСМ.

чисел, и по два приемных регистра (*БЗ1Ч, БЗ1П* и *БЗ2Ч, БЗ2П*) для приема кодов двух чисел, с которыми производится заданное действие. Выполнение того или иного действия с кодами чисел обуславливается работой устройства управления.

Запоминающее устройство электронных счетных машин характеризуется его емкостью — количеством хранимых кодов — и скоростью выборки кодов. Из известных видов запоминающих устройств наиболее быстродействующими являются запоминающие устройства на электронно-лучевых трубках и на магнитных элементах, имеющих прямоугольную петлю гистерезиса. Эти виды запоминающих устройств позволяют производить выборку кода для всех разрядов одновременно (параллельная выборка кода) и не требуют ожидания выборки требуемого кода, как это имеет место в запоминающих

устройствах с электроакустическими трубками или магнитным барабаном.

На машине БЭСМ до последнего времени в качестве оперативного запоминающего устройства (ОЗУ) использовалось запоминающее устройство на электронно-лучевых трубках. Это устройство позволяет обеспечить скорость выборки до 100 000 кодов в 1 секунду и имело емкость 1023 кода.

В 1956 г. в качестве основного вида оперативного запоминающего устройства было введено в действие устройство на магнитных элементах (МОЗУ).

Этот вид запоминающего устройства, хотя в данном, конкретном случае обладает такими же параметрами по скорости, как и запоминающее устройство на электронно-лучевых трубках, имеет значительно меньшие размеры, большую емкость (2047 кодов), проще и надежнее в эксплуатации.

Оперативное запоминающее устройство участвует непосредственно в процессе вычислений; из этого устройства коды чисел или команд поступают в арифметическое устройство и в другие блоки машины. Выборка кодов из запоминающего устройства и посылка в него результатов вычислений обуславливаются работой устройства управления.

Имеющаяся емкость оперативного запоминающего устройства (2047 кода) является недостаточной для решения широкого круга задач. Увеличение емкости запоминающего устройства путем дублирования ОЗУ нерационально, так как это приводит к существенному увеличению аппаратуры и тем самым снижает надежность работы машины в целом. Для увеличения количества хранимых кодов чисел или команд применен другой вид запоминающего устройства с использованием магнитной записи на барабане и на лентах (МЗУБ и МЗУЛ).

Магнитный барабан конструктивно представляет собой полый цилиндр, выполненный из немагнитного материала. В движение он приводится мотором. Поверхность этого цилиндра покрыта тонким слоем ферромагнитного лака. Вдоль образующей магнитного барабана расположены головки, которые служат для записи и считывания кода с поверхности барабана,

Недостатком запоминающего устройства с магнитным барабаном является то обстоятельство, что при выборке требуемого кода необходимо ожидать, пока этот код подойдет под считывающую головку. При однорядных головках в худшем случае это время равно времени одного оборота барабана. Указанное обстоятельство исключает применение магнитного барабана в качестве оперативного запоминающего устройства.

Однако при решении задач, требующих хранения большого количества кодов чисел, как правило, одновременно приходится оперировать лишь с небольшой группой чисел и, закончив вычисления с данной группой, переходить к следующей. Коды чисел, с которыми приходится оперировать одновременно, конечно, должны храниться в оперативном запоминающем устройстве. Но после того как вычисления с данной группой закончены, эти числа могут быть заменены новыми. Подобная замена может быть произведена путем передачи кодов новой группы чисел с магнитного барабана в оперативное запоминающее устройство. Предусматривается также и обратная передача из оперативного запоминающего устройства на магнитный барабан группы кодов чисел. Подобная передача требуется для хранения промежуточных результатов, которые используются в дальнейших вычислениях.

Кроме кодов чисел, на магнитном барабане могут храниться также коды подпрограмм, которые используются на отдельных этапах вычислений. В том случае, когда коды с магнитного барабана или на него передаются группами, одно число за другим, время ожидания момента, когда подойдет требуемый код, имеет значение лишь для первого числа. В этом случае такие обращения к барабану не сильно сказываются на общем времени решения задачи.

В запоминающем устройстве на магнитных лентах используется стандартная магнитная лента, широко применяемая для звукозаписи. Конструкция лентопротяжек несколько отличается от обычных магнитофонов, так как в данном случае требуются большие скорости движения ленты. Лента при движении проходит под вдвоенной головкой, одна половина которой предназначена для

записи и считывания кодов, а другая — для записи и считывания синхронизирующих сигналов.

Магнитные ленты менее оперативны, чем магнитный барабан. Для барабана время ожидания прихода первого кода для любой группы чисел в худшем случае равно времени одного оборота барабана. Для магнитной ленты в худшем случае, когда лента стоит под считывающими головками началом, а требуемая группа находится в конце, необходимо перемотать всю ленту. С другой стороны, количество хранимых чисел на магнитных лентах несравненно больше, чем на магнитных барабанах.

Запоминающее устройство на магнитных лентах используется в основном так же, как и магнитный барабан. Передача и прием кодов осуществляются группами.

Непосредственный обмен кодами между магнитными лентами и магнитным барабаном требует точной синхронизации скоростей обоих этих устройств. Указанное обстоятельство приводит к чрезмерному усложнению устройств, и потому от подобной передачи пришлось отказаться. При необходимости передать коды с магнитной ленты на магнитный барабан или обратно эти коды предварительно должны быть переданы в оперативное запоминающее устройство.

Устройство управления (УУ) машиной должно обеспечить полную автоматичность вычислительного процесса. Выполнение тех или иных действий над числами и выборка этих чисел из оперативного запоминающего устройства, а также отсылка результата в соответствующую ячейку оперативного запоминающего устройства определяются кодом команды. Для выполнения текущей команды необходимо ее код передать во временное запоминающее устройство. Для этой цели в машине предусмотрен блок запоминания команд (БЗК), выполненный на триггерных ячейках.

Выборка кодов команд из оперативного запоминающего устройства на блок запоминания команд обычно производится в порядке следования их номеров. Иногда такой порядок следования команд нарушается. Например, команды условного или безусловного перехода задают номер следующей команды в своем третьем адресе.

Кроме того, должен быть обеспечен переход от основной программы к подпрограмме и после ее завершения возвращение к основной программе. Для последнего случая удобно иметь два блока, ведающих выборкой кодов из оперативного запоминающего устройства, так как при переходе к подпрограмме необходимо сохранить номер команды основной программы до момента обратного возвращения к основной программе.

В соответствии с этим требованием в машине предусмотрены два блока управления командами: блок центрального управления командами (ЦУК) и блок местного управления командами (МУК). (Следует отметить, что эти блоки используются также и для осуществления обмена кодами между оперативным и внешним запоминающим устройствами.)

Блоки центрального и местного управления командами представляют собой электронные счетчики с возможностью гашения имеющегося кода и приема кода из блока запоминания команды. Прибавление единицы к имеющемуся в ЦУК или МУК коду обеспечивает выборку команд в порядке следования их номеров. Прием нового кода с блока запоминания команд дает возможность осуществить условный или безусловный переход и переход к подпрограммам с последующим возвратом к основной программе.

Характер работы отдельных блоков машины зависит от выполняемой операции. Операция задается кодом, находящимся в группе *АОп* на блоке запоминания команд. Для того чтобы подготовить соответствующие цепи на других блоках для выполнения требуемой операции, необходимо код операции превратить в управляющее напряжение. Это производится коммутатором операций (*КОн*). На вход коммутатора операций подается заданный код; на выходной цепи, соответствующей производимой операции, получается управляющее напряжение. Это управляющее напряжение подготавливает требуемые цепи в других блоках машины.

Для выполнения заданной операции принята комбинированная система центрального и местного управления операциями. Те операции, которые являются общими для большинства команд и которые не вызывают

существенного увеличения времени выполнения других команд, производятся от центрального управления операциями (*ЦУOn*).

Те же операции, которые являются индивидуальными для тех или иных команд и которые при осуществлении их от центрального управления вызвали бы существенное замедление выполнения других команд, производятся от местного управления (*МУOn*). Переключение с центрального управления на местное и с местного на центральное осуществляется автоматически.

Работа блока центрального управления операциями заключается в подаче управляющих импульсов или потенциалов в отдельные устройства машины. Эти импульсы или потенциалы вырабатываются блоком центрального управления (*ЦУ*) и поступают в те или иные устройства машины от блока *ЦУOn* в зависимости от кода выполняемой операции, т. е. в зависимости от того, на какой из выходных цепей коммутатора операций имеется управляющее напряжение.

Блок центрального управления (*ЦУ*) циклически повторяет вырабатываемые им импульсы и потенциалы. Цикл работы центрального управления определяет цикл работы машины и включает в себя выборку двух чисел из запоминающего устройства, выполнение заданной операции с этими числами, посылку результата в запоминающее устройство и прием следующей команды. Если какая-либо из операций требует местного управления операциями, то на время работы машины от местного управления центральное управление останавливается и продолжает свой цикл лишь после завершения работы местного управления.

Запуск машины и контроль за ее работой осуществляются с пульта управления (*ПУ*). На пульте управления предусматривается возможность переключения на ручную и автоматическую работу машины, а также работу по циклам.

При ручной работе импульсы, определяющие темп работы машины, задаются нажатием ключа (при каждом нажатии выдается одиночный импульс). При этом можно по сигнальным лампам, расположенным на мне-

монической схеме пульта, проследить за правильностью работы отдельных элементов машины.

При автоматической и циклической работе темп работы машины задается частотой главного генератора импульсов. При циклической работе автоматически выполняется лишь одна операция. После окончания операции машина останавливается и для выполнения следующей операции требуется снова нажать ключ пуска. Циклическая работа позволяет видеть результат после выполнения каждой операции, что особенно существенно при отладке программы.

При автоматической работе все операции по заданной программе выполняются без вмешательства человека. Машина работает до тех пор, пока не закончится весь процесс вычислений, после чего машина останавливается. Возможен также «аварийный» останов машины с соответствующей сигнализацией на пульте управления в тех случаях, когда происходит, например, выход числа из количества располагаемых разрядов.

Первоначальный ввод чисел и команд в машину осуществляется с перфоленты (ПЛ) последовательным кодом. В качестве преобразователя кодов из последовательного в параллельный и обратно служит арифметическое устройство.

Окончательный вывод результатов вычислений производится с помощью печатающего устройства (ПчР).

Таким образом, скелетная схема БЭСМ (рис. 5) состоит из следующих основных устройств:

1. Арифметическое устройство (АУ) с его блоками:
 - а) 1-й блок запоминания кода числа (БЗ1Ч);
 - б) 1-й блок запоминания кода порядка (БЗ1П);
 - в) 2-й блок запоминания кода числа (БЗ2Ч);
 - г) 2-й блок запоминания кода порядка (БЗ2П);
 - д) сумматор кодов чисел (СмЧ);
 - е) сумматор кодов порядков (СмП).
2. Устройство управления (УУ) с его блоками:

- а) блок центрального управления машиной (ЦУ);
- б) блок управления командами (УК), состоящий из блока запоминания команд (АОн, А1, А2, А3), счетчиков центрального и местного управления командами (ЦУК, МУК) и коммутатора операций (КОн);

- в) блок центрального управления операциями (ЦУОп);
- г) блок местного управления операциями (МУОп);
- д) пульт управления (ПУ).
- 3. Оперативное запоминающее устройство (ОЗУ).
- 4. Внешнее запоминающее устройство на магнитных барабанах (МЗУБ) и магнитных лентах (МЗУЛ).
- 5. Устройство ввода на перфоленте (ПЛ).
- 6. Устройство печатания результатов вычислений (ПчР).

§ 2. Типовая последовательность работы

Рассмотрение работы машины удобнее всего начать с момента поступления кода команды на блок запоминания команд (БЗК). Код команды в большинстве случаев состоит из следующих групп:

1. Код операции, указывающий, какая операция должна быть произведена (находится в разрядах АОп).

2. Код номера ячейки запоминающего устройства, в которой хранится код первого числа (находится в разрядах А1).

3. Код номера ячейки запоминающего устройства, в которой хранится код второго числа (находится в разрядах А2).

4. Код номера ячейки запоминающего устройства, в которую должен быть послан результат (находится в разрядах А3).

Так, например, код команды

АОп	А1	А2	А3
000001	00000000001	00000000010	00000000011

означает:

«Сложить (код в АОп) число, находящееся в первой ячейке запоминающего устройства (код в А1), с числом, находящимся во второй ячейке запоминающего устройства (код в А2), и их сумму направить в третью ячейку запоминающего устройства (код в А3)».

При поступлении кода команды на блок БЗК код операции с АОп воздействует на коммутатор операций

(*КОн*) и возбуждает одну из его выходных цепей, соответствующую заданной операции. Управляющее напряжение на этой выходной цепи подготавливает цепи в блоке управления операциями (*ЦУОн*), необходимые для выполнения заданной операции (в рассматриваемом примере — цепи, необходимые для производства операции сложения). Затем соответствующий импульс от устройства центрального управления (*ЦУ*) через подготовленные цепи на блоке *ЦУОн* передает код номера первого адреса с *А1БЗК* по коммутационным шинам адреса (*КША*) в оперативное запоминающее устройство (*ОЗУ*), после чего происходит выборка кода числа (в рассматриваемом примере — из первой ячейки запоминающего устройства).

Выбранный код числа по коммутационным шинам (*КШ*) передается в первый блок запоминания арифметического устройства (*БЗ1Ч* и *БЗ1П*). Таким образом, в *БЗ1Ч* будет находиться цифровая часть первого числа, а в *БЗ1П* — код его порядка. В зависимости от заданной операции с этими кодами производятся те или иные действия. При операции сложения код числа передается с *БЗ1Ч* на сумматор числа (*СмЧ*), а код его порядка — с *БЗ1П* на *СмП*.

По окончании приема первого числа на арифметическое устройство импульс от *ЦУ* через блок *ЦУОн* передает код номера второго адреса с *А2БЗК* по *КША* и происходит выборка кода числа из ячейки *ОЗУ*, соответствующей переданному коду номера с *А2БЗК* (в рассматриваемом примере — из второй ячейки запоминающего устройства).

Выбранный код числа по *КШ* передается во второй блок запоминания арифметического устройства (*БЗ2Ч* и *БЗ2П*). Таким образом, в *БЗ2Ч* будет находиться код второго числа, а в *БЗ2П* — код его порядка. В зависимости от заданной операции с этими кодами производятся те или иные действия.

При операции сложения первоначально производится выравнивание порядков обоих чисел, так как складывать можно лишь числа одного порядка. Выравнивание порядков производится путем сдвига кода числа с меньшим порядком вправо. Так как количество сдвигов

зависит от разности порядков обоих чисел, то эта операция может занимать различное количество времени. Поэтому при выравнивании порядков центральное управление останавливается и сдвиг осуществляется от местного управления операциями (*МУОп*). По окончании выравнивания порядков от *ЦУОп* производится сложение кодов чисел.

После производства заданного действия над числами в арифметическом устройстве получается результат вычисления. Однако, прежде чем передать результат в запоминающее устройство, необходимо его нормализовать (если только в коде команды нет блокировки нормализации), т. е. представить в таком виде, чтобы первая значащая цифра занимала старший разряд. Так как время, необходимое для нормализации, может оказаться самым различным, то эта операция производится при остановленном центральном управлении операциями от *МУОп*.

После нормализации результата соответствующий импульс *ЦУ* через блок *ЦУОп* передает код адреса с *АЗБЗК* по *КША* и выбирает номер ячейки *ОЗУ*, в которую должен быть послан результат. Одновременно код числа с *АУ* поступает в *ОЗУ* и запоминается в выбранной ячейке (в рассматриваемом примере — в третьей ячейке).

На этом заканчивается выполнение заданной команды, и машина переходит к вызову следующей команды.

Если управление производится блоком *ЦУК*, то после выполнения заданной команды к коду, имеющемуся в блоке *ЦУК*, прибавляется единица (цепь «+1»). Блок *ЦУК* представляет собой накапливающий счетчик, и добавление к нему единицы увеличивает на 1 имевшийся в нем ранее код. Соответствующим импульсом *ЦУ* этот код передается с *ЦУК* по *КША* и происходит выборка кода следующей по порядку команды из ячейки *ОЗУ*, соответствующей переданному коду номера команды. Выбранный код команды по *КШ* передается в блок запоминания команд, и машина приступает к выполнению следующей команды.

Если управление выборкой команд из запоминающего устройства производится блоком *МУК*, то операция

происходит точно так же, как и при управлении блоком ЦУК, за исключением того, что единица добавляется к коду на блоке МУК и передается с него на ОЗУ. Блок МУК совершенно идентичен блоку ЦУК.

Большинство команд выполняется на машине в рассмотренной только что последовательности, а именно:

1) возбуждение цепей, определяющих характер выполнения заданной команды (передача кода операции с АОп на КОп и выдача управляющего напряжения с КОп на ЦУОн);

2) выборка кода первого числа из ОЗУ в соответствии с кодом его адреса в А1БЗК и прием кода первого числа на АУ (передача кода с А1БЗК на КША, выборка кода из ОЗУ на КШ и передача его на АУ);

3) выборка кода второго числа из ОЗУ в соответствии с кодом его адреса в А2БЗК и прием кода второго числа на АУ (передача кода с А2БЗК на КША, выборка кода из ОЗУ и передача его на АУ);

4) производство на АУ тех или иных действий с кодами чисел, в зависимости от заданной операции, и получение на АУ результата;

5) передача кода результата из АУ в ОЗУ в соответствии с кодом адреса ячейки в АЗБЗК (передача кода с АЗБЗК на КША, выборка ячейки в ОЗУ и передача в нее кода результата из АУ);

6) выборка кода следующей по номеру команды из ОЗУ и передача его в БЗК (изменение на единицу кода в ЦУК или МУК, передача его на КША, выборка кода команды с ОЗУ и передача его на БЗК).

§ 3. Последовательность работы при операциях передачи управления

Для ряда операций, приведенных выше, последовательность работы машины нарушается. В основном это относится к операциям, связанным с изменением обычной последовательности вызова очередной команды.

Изменения порядка выполнения команд в зависимости от результата вычисления осуществляются на машине командами сравнения. Код команды сравнения состоит из следующих групп.

1. Код операции (в *АОп*), указывающий, что должна быть произведена операция сравнения двух чисел и в зависимости от результата сравнения выбрана для исполнения команда из той или иной ячейки.

2. Код номера ячейки запоминающего устройства, в которой хранится код первого сравниваемого числа (в *А1*).

3. Код номера ячейки запоминающего устройства, в которой хранится код второго сравниваемого числа (в *А2*).

4. Код номера ячейки запоминающего устройства, в которой хранится код команды, к исполнению которой следует перейти, если выполняется некоторое условие.

Если же это условие не выполняется, то исполняется следующая по порядку команда.

Так, например, код команды

<i>АОп</i>	<i>А1</i>	<i>А2</i>	<i>А3</i>
010100	000000000001	000000000010	000010000000

означает: сравнить (код в *АОп*) число, находящееся в первой ячейке запоминающего устройства (код в *А1*), с числом, находящимся во второй ячейке запоминающего устройства (код в *А2*). Если первое число меньше второго числа, то следующей должна выполняться команда, находящаяся в 64-й ячейке запоминающего устройства; если же первое число больше или равно второму числу, то должна быть взята следующая по номеру команда.

При поступлении кода команды сравнения на блок *БЗК* код операции с *АОп* воздействует на коммутатор операций (*КОп*) и возбуждает его выходную цепь, соответствующую операции сравнения. Управляющее напряжение на этой выходной цепи подготавливает цепи в блоке *ЦУОп*, необходимые для выполнения заданной операции сравнения.

Затем соответствующий импульс от *ЦУ* через подготовленные цепи на блоке *ЦУОп* передает код адреса первого числа с *А1БЗК* по *КША* и происходит выборка кода числа из ячейки *ОЗУ*, соответствующей переданному коду адреса (в рассматриваемом примере — из первой ячейки *ОЗУ*). Выбранный код числа по *КШ* передается в *Б31Ч* и *Б31П* и затем в *СмЧ* и *СмП*.

По окончании приема первого числа на *АУ* импульс от *ЦУ* через блок *ЦУОн* передает код адреса второго числа с *А2БЗК* по *КША* и происходит выборка кода числа из ячейки *ОЗУ*, соответствующей переданному коду адреса (в рассматриваемом примере — из второй ячейки *ОЗУ*). Выбранный код числа по *КШ* передается в *БЗ2Ч* и *БЗ2П*.

Арифметическое устройство производит сравнение двух принятых чисел, и если первое число меньше второго, то возбуждаются цепи, которые от импульсов *ЦУ* через блок *ЦУОн* гасят код, имевшийся на блоке *ЦУК*, и передают на него по *КША* код с *АЗБЗК*. Если же первое число больше второго или равно ему, то возбуждаются другие цепи, гашения кода на *ЦУК* не происходит, а к имеющемуся коду на *ЦУК* прибавляется единица.

Полученный код на блоке *ЦУК* соответствующим импульсом *ЦУ* передается с *ЦУК* по *КША* и происходит выборка кода следующей команды из ячейки *ОЗУ*, соответствующей переданному коду адреса команды. Выбранный код команды по *КШ* передается в *БЗК*, и машина приступает к выполнению следующей команды (если управление выборкой команд производилось блоком *МУК*, то все указанные выше операции производятся на блоке *МУК*, а не *ЦУК*).

В случае необходимости перехода от основной программы к какой-либо подпрограмме, в основную программу включается команда «Изменение номера команды на местном управлении командами». Работа на местном управлении при этом начинается с команды, адрес которой указан в третьем адресе выполняемой команды (в первом и втором адресах коды отсутствуют).

Так, например, код команды

<i>АОн</i>	<i>А1</i>	<i>А2</i>	<i>А3</i>
011010	000000000000	000000000000	00001000000

означает: изменить номер команды на местном управлении командами (код *АОн*), если работа производилась на центральном управлении, то перейти на местное управление и работу начать с команды, номер которой указан в *А3* (00001000000).

При поступлении на блок *БЗК* кода такой команды код операции с *АОп* воздействует на коммутатор операций (*КОн*) и возбуждает соответствующую выходную цепь. При этой команде выборка кодов чисел с *ОЗУ* не производится и *АУ* в операции не участвует.

Соответствующие импульсы *ЦУ* через блок *ЦУОн* переключают управление командами с блока *ЦУК* на блок *МУК*, гасят код, имевшийся на блоке *МУК*, и передают на него по *КША* код с *АЗБЗК*. Затем код, полученный на блоке *МУК*, соответствующим импульсом *ЦУ* передается с *МУК* по *КША* и происходит выборка кода первой команды подпрограммы из ячейки *ОЗУ*, соответствующей переданному коду номера команды. Выбранный код команды по *КШ* передается в *БЗК*, и машина приступает к выполнению заданной подпрограммы.

При переключении управления командами с центрального на местное код, имевшийся в блоке *ЦУК*, остается неизменным на все время выполнения заданной подпрограммы.

В конце каждой подпрограммы ставится команда: «Передача на центральное управление командами без гашения». Код такой команды имеет лишь код операции в разрядах *АОп*. Во всех остальных адресах коды отсутствуют, т. е. код команды будет:

<i>АОп</i>	<i>А1</i>	<i>А2</i>	<i>А3</i>
011001	000000000000	000000000000	000000000000

При поступлении кода такой команды на блок *БЗК* код операции с *АОп* воздействует на *КОн* и возбуждает соответствующую выходную цепь. При этой команде выборка кодов чисел с *ОЗУ* не производится и *АУ* в операциях не участвует.

Соответствующие импульсы *ЦУ* через блок *ЦУОн* переключают управление командами с блока *МУК* на блок *ЦУК* и прибавляют единицу к коду, имевшемуся в блоке *ЦУК*. Затем код, полученный на блоке *ЦУК*, соответствующим импульсом *ЦУ* передается с *ЦУК* на *КША*, и происходит выборка кода следующей по порядку команды основной программы из ячейки *ОЗУ*, соответствующей переданному коду номера команды. Выбранный код команды по *КШ* передается в *БЗК*, и машина возвращается к вычислениям по основной программе.

Кроме этих двух команд: «Изменение номера команды на местном управлении командами» и «Передача на центральное управление без гашения», имеются еще две команды, обратные по своему характеру: «Изменение номера команды на центральном управлении командами» и «Передача на местное управление без гашения».

Таким образом, введение центрального и местного управления командами позволяет в любом месте основной программы перейти к вычислениям по любой подпрограмме и после завершения вычислений по заданной подпрограмме вновь вернуться к прерванному месту основной программы. Введение такого управления значительно упрощает программирование решения задач.

Необходимо отметить, что команда «Изменение номера команды на центральном управлении командами» (ИЦУК) в последнее время была модернизирована. При выполнении этой команды, помимо основного назначения, формируется так называемая *команда возврата*, которая записывается в ОЗУ по второму адресу А2 (ранее он не использовался). Команда возврата представляет собой обычную команду ИЦУК, в третьем адресе которой указан номер команды, на которую следует возвратиться после выполнения подпрограммы (он равен номеру исполняемой команды, увеличенному на единицу). Наличие такой возможности позволяет выполнять подпрограммы с автоматическим возвратом к основной программе и без блока МУК.

§ 4. Последовательность работы при операциях обращения к МЗУ

Существенно отличается от стандартной последовательность работы машины при выполнении операций, связанных с обменом кодов между оперативным запоминающим устройством и магнитными запоминающими устройствами (и перфолентой). На машине предусмотрены следующие передачи кодов:

1. Передача кодов с магнитной ленты в оперативное запоминающее устройство.
2. Передача кодов из оперативного запоминающего устройства на магнитную ленту.
3. Перемотка ленты до заданного номера группы.

4. Передача кодов с перфоленты (вводное устройство) в оперативное запоминающее устройство.

5. Передача кодов из оперативного запоминающего устройства на магнитный барабан.

6. Передача кодов с магнитного барабана в оперативное запоминающее устройство.

Все перечисленные операции, как уже упоминалось, выполняются в две команды *МЗ(а)* и *МЗ(б)*.

При поступлении на *БЗК* кода первой команды *МЗ(а)* код операции с *АОп* воздействует на *КОп* и возбуждает необходимую выходную цепь. Соответствующий импульс *ЦУ* передает код с *А1БЗК* по *КША* на внешнее запоминающее устройство. Этот код подготавливает цепи выборки: перфолента, магнитная лента или барабан; считывание или запись; номер группы барабана или номер ленты.

Следующий импульс *ЦУ* передает код с *А2БЗК* на *КША* и подготавливает цепи выборки требуемого номера ячейки на барабане или номера группы чисел на ленте. Затем импульсы *ЦУ* гасят код, имевшийся на *МУК*, передают по *КША* код из *А3БЗК* на *МУК*, прибавляют единицу к коду, имеющемуся на *ЦУК*, и передают получившийся код с *ЦУК* в *ОЗУ*. Происходит выборка кода следующей команды из соответствующей ячейки *ОЗУ*, который по *КШ* передается на *БЗК*.

При поступлении на *БЗК* второй команды «Магнитная запись — б» происходит останов *ЦУ* и управление машиной передается на внешнее запоминающее устройство.

Когда подойдет номер числа, с которого надо начать, например, считывание (в случае магнитного барабана), открываются цепи, связывающие магнитный барабан с оперативным запоминающим устройством, и коды чисел последовательно поступают на один из сдвиговых регистров *АУ*. Когда код одного числа полностью окажется на регистре, происходит передача его в оперативное запоминающее устройство по адресу, имеющемуся на *МУК*. Затем прибавляется единица к коду на *МУК*, что обеспечивает выборку следующей ячейки.

В случае записи процесс происходит в обратном порядке. Код параллельно поступает на регистр *АУ* и, по-

следовательно сдвигаясь, разряд за разрядом записывается на барабан.

В начале передачи кодов чисел одновременно передается также код адреса с *А2БЗК* на внешнее запоминающее устройство, которое контролирует окончание магнитной записи. Когда подойдет номер числа, на котором следует окончить магнитную запись, запускается *ЦУ* и машина продолжает работать по основной программе.

В случае считывания с магнитной ленты передача кодов начинается тогда, когда подойдет группа чисел, номер которой задан в *А2* команды *МЗ(а)*. После этого коды чисел с магнитной ленты поступают последовательно на сдвиговый регистр *АУ*. Операция продолжается до тех пор, пока не считается заданное количество кодов данной группы чисел, после чего выключается из работы *МЗУЛ*, запускается *ЦУ* и машина продолжает работать по основной программе.

В случае записи на магнитную ленту операция начинается, как только лента достигнет нормальной скорости. Предварительно записывается код номера группы, который был передан в *МЗУ* с адреса *А2* в подготовительной команде. Затем код с *МУК* передается по *КША* в *ОЗУ*, выбирается код числа из соответствующей ячейки *ОЗУ* и поступает на регистр *АУ*. С регистра код числа последовательно поступает на *МЗУЛ* и записывается на соответствующей ленте. После окончания приема одного числа прибавляется единица к коду на *МУК*; новый код адреса передается с *МУК* в *ОЗУ*, выбирается код числа из соответствующей ячейки и передается на регистр. Код с *А2БЗК* второй команды магнитной записи, посланный во внешнее запоминающее устройство, контролирует количество записанных кодов чисел. Когда количество записанных кодов будет равно заданному, операция магнитной записи заканчивается, запускается *ЦУ* и машина продолжает работать по основной программе.

Обращение к вводному устройству на перфоленте (*ПЛ*) осуществляется таким же образом, как и к магнитной ленте при операции «Считывание», в две команды.

§ 5. Контроль работы машины

При решении задач необходимо иметь уверенность в правильности работы машины. Большое число элементов и, особенно, электронных ламп требует повышенного внимания к вопросам контроля. Отказ в работе хотя бы одного элемента, хотя бы одной лампы может привести к неправильным результатам вычислений.

Контроль правильности работы машины осуществляется на БЭСМ путем задания серии тестов. Эти тесты сводятся к решению по специальным программам ряда элементарных задач с заранее известными ответами. При расхождении полученного и заданного ответов машина автоматически останавливается. Тесты составляются таким образом, чтобы охватить при работе все элементы машины, причем в наиболее тяжелых для них режимах.

Особое внимание при тестовых испытаниях уделяется электронным лампам. Для этой цели работа по тестам проводится при ухудшенном режиме работы электронных ламп. Благодаря этому выявляются лампы, находящиеся вблизи границы надежной работы, и обеспечивается гарантия правильной работы машины на определенный срок. Ухудшение режима электронных ламп осуществляется изменением накала или смещения.

В случае неисправности какого-либо элемента его необходимо достаточно быстро обнаружить и исправить. Для быстрого обнаружения неисправного элемента система тестов построена таким образом, что расхождение ответов на отдельных этапах проверки обычно дает указание, в каком устройстве имеется неисправность. Сигнализация на пульте управления помогает уточнить местоположение неисправного блока. Окончательное установление места повреждения производится или по осциллографу, или путем переключения машины на одиночную работу. При проверке по осциллографу устанавливается операция, дающая ошибку, и машина запускается в режиме многократного повторения этой операции. Осциллографом просматривается работа блоков, могущих привести к данной ошибке, и устанавливается неисправный или малонадежный блок. При переключе-

нии на одиночную работу каждое нажатие пускового ключа дает лишь один управляющий сигнал, выполняющий на машине какую-либо элементарную операцию. За выполнением этих элементарных операций можно наблюдать по сигнальным лампам на пульте управления и во многих случаях определить неисправный элемент.

Такими методами удается достаточно быстро определить неисправный, или малонадежный блок и заменить его резервным. (Определение причин неисправности в блоке и ее устранение производятся вне машины на контрольных стендах.)

Наибольшие неприятности доставляют неисправности, связанные с плохими контактами в ламповых панелях, разъемах и местах плохой пайки. Недостаточно надежный контакт приводит к случайным сбоям в работе машины, которые не всегда проявляются при тестовых проверках. Для выявления таких мест при проведении тестов создается вибрация отдельных плат и блоков. При этом ухудшаются условия работы контактов и удается установить малонадежные места.

Профилактические испытания машины по тестовым программам при ухудшенном режиме работы ламп проводятся периодически. Чем сильнее ухудшается режим работы ламп, тем на больший срок может быть гарантирована правильная работа машины, но зато и большее число ламп будет выходить из строя. На БЭСМ профилактическая проверка машины производится, как правило, один раз в сутки.

Тестовые испытания машины исключают систематические ошибки при производстве вычислений. Однако ввиду огромного количества операций, производимых машиной, полностью не исключена возможность случайной ошибки (например, вследствие кратковременного скачка питающего напряжения). Поэтому результаты вычислений обязательно должны быть проверены вне зависимости от проверки правильности работы отдельных элементов машины.

Возможны методы аппаратного контроля правильности вычислений. Однако эти методы усложняют схему машины и увеличивают количество аппаратуры.

Дополнительные схемы контроля сами могут явиться причиной неправильной работы машины и, кроме того, обычно они не охватывают работы всех элементов машины. Поэтому для БЭСМ принят метод логического контроля правильности расчетов, выполняемый непосредственно самой программой вычислений.

Простейшим примером метода логического контроля является применяемый при ручных вычислениях «расчет в две руки», т. е. производство двух расчетов и сопоставление их результатов. Так как систематическая ошибка исключается проверочными тестами, то совпадение результатов двух расчетов является достаточной гарантией правильности произведенных вычислений.

Помимо «расчета в две руки», в зависимости от типа задачи могут применяться более сложные методы логического контроля.

Так, например, при решении системы линейных уравнений проверкой вычислений может служить подстановка найденных неизвестных в исходное уравнение. Если после такой подстановки результат не превышает заданной точности вычислений, то это является достаточной гарантией правильности вычислений. При численном интегрировании дифференциальных уравнений с учетом, например, третьих разностей проверкой правильности вычислений может служить величина следующей четвертой разности. Если величина четвертой разности не превышает заданной точности вычислений, то это указывает на правильность вычислений. Может быть применен также следующий метод логического контроля: дополнительно к требуемым величинам вычисляются также вспомогательные величины, являющиеся следствием первых. Так, например, при решениях уравнений баллистики могут быть составлены дифференциальные уравнения не только для составляющих скорости (V_x и V_y), но также и для полной скорости V . Проверкой вычислений будет являться сравнение вычисленного значения V^2 с величиной $V_x^2 + V_y^2$. Точно так же проверкой вычислений может служить сравнение значений какой-либо величины, найденной различными

методами вычислений. Например, при решении дифференциальных уравнений сопоставление двух расчетов, проведенных при различном шаге, является достаточной гарантией правильности вычислений.

При вычислении таблиц по рекуррентным формулам можно во многих случаях вычислить ряд опорных значений другими методами. Верный выход на опорные значения является достаточной гарантией правильности всех подсчитанных промежуточных значений.

ГЛАВА 4

ПРИНЦИПЫ ПОСТРОЕНИЯ ОСНОВНЫХ УСТРОЙСТВ

Машина БЭСМ сконструирована в основной своей части из отдельных стандартных блоков, схемы которых смонтированы в двухламповых или четырехламповых каркасах. Эти блоки расположены на стойках, по которым разведены межблочные соединения. Устройство управления и арифметическое устройство занимают одну отдельную стойку. Оперативное запоминающее устройство и внешнее запоминающее устройство на магнитных барабанах и магнитных лентах также расположены на отдельных стойках. Такое отдельное расположение различных узлов значительно упрощает техническую эксплуатацию машины и делает возможным проводить профилактическую подготовку и проверку машины автономно по стойкам.

§ 1. Стандартные блоки

В машине имеется несколько типов стандартных блоков: триггерный блок, вентильный блок, блок формирования, блок катодных повторителей, блок инверторов, блок цепочки переносов, диодные блоки и блоки линий задержек. Эти блоки были специально разработаны применительно к условиям работы машины и тщательно исследованы.

На этих основных типах блоков построены схемы устройства управления и арифметического устройства, а также автомата управления запоминающих устройств. Поэтому надежность работы всей машины в значительной степени определяется надежностью

работы стандартных блоков. Работоспособность стандартных блоков БЭСМ перед установкой их на машину определяется на специальном стенде для испытания блоков. Все стандартные блоки машины БЭСМ в комплексе допускают изменения постоянных питающих напряжений на $\pm 2\%$, а напряжение накала на $\pm 10\%$.

Каждый каркас стандартного блока имеет один разъем, с помощью которого блок соединяется со стойкой. Разъем имеет три ряда ножевых контактов, по 10 штук в каждом ряду. К этим контактам подводятся цепи, необходимые для работы схемы: питающие напряжения, входные и выходные сигналы.

Триггерный блок. Одним из основных элементов, входящим во все устройства машины, является триггерная ячейка (блок *T*). Триггерная ячейка имеет два устойчивых состояния, что весьма удобно для представления кодов чисел в двоичной системе; одному состоянию (положению) приписывается значение кода 0, другому — значение кода 1. Состоянию «1» соответствует высокий потенциал на выходной шине *Вых*«1», а состоянию «0» — высокий потенциал на выходной шине *Вых*«0». В схеме блока *T* предусмотрены два входа на установку триггера в положение «1» (*У*«1») и два — на установку в положение «0» (*У*«0»).

При действии импульса на обе сетки триггерной ячейки последняя меняет одно свое устойчивое состояние на другое. Подобный режим работы позволяет использовать триггер в качестве счетной ячейки для сложения кодов чисел. Счетный вход (*СчВх*) получается путем соединения входов *У*«1» и *У*«0».

Весьма просто получается также на триггере импульс двоичного переноса в высший разряд в случае перехода триггера из положения «1» в положение «0». Для этой цели нулевой перепад анодного напряжения триггера дифференцируется и полученный импульс двоичного переноса подается на вход триггера следующего разряда.

Триггерные ячейки используются в БЭСМ не только для сложения кодов, но также и для промежуточного запоминания кодов. В этом случае импульсы кода подаются лишь на одну из сеток триггера. На другую

сетку подается «гасящий» импульс (устанавливающий триггер в положение «0»). Кроме того, триггерные ячейки используются в БЭСМ для сдвига кодов вправо или влево.

Со стороны сеток триггер защищен диодами так, чтобы паразитные импульсы не вызывали его срабатывания.

В электронных счетных машинах режим работы триггеров несравненно сложнее, чем у обычных счетчиков. Поэтому к триггерам в электронных счетных машинах предъявляются значительно более строгие требования в отношении надежности их работы, помехоустойчивости, быстроты перехода из одного устойчивого состояния в другое и т. п.

Блоки ламповых вентилях. Для управления отдельными цепями электронной счетной машины широко применяются вентильные схемы. Электронные вентили являются устройствами совпадения и работают по логическому закону «и», т. е. на выходе вентиля сигнал возникает лишь в том случае, когда имеются управляющие напряжения на всех его входах. Если на одном из входов не будет управляющего напряжения, то не будет сигнала и на выходе вентиля.

В одном вентильном блоке (*В*) смонтировано два отдельных вентиля. Каждый вентиль имеет свой импульсный трансформатор. Концы обмоток выходного трансформатора выведены на самостоятельные контакты, так же как и аноды вентильных ламп. Это сделано с целью иметь возможность включать оба вентиля на один какой-либо трансформатор для осуществления собирательной схемы. Вентили можно использовать в качестве усилителей. В обычной схеме включения на один из входов вентиля подается перепад напряжения с блока *Т*.

Разновидностью блока *В* является блок вентилях с линией задержки (*ВЗ*). Сочетание вентиля с линией задержки широко используется в схеме арифметического устройства для осуществления сдвигов и переноса. В блоке *ВЗ* имеются две линии задержки, начало и конец которых выведены на отдельные контакты разъема. Время задержки одной линии составляет $0,3 \text{ мксек} \pm 20\%$.

Блоки формирования импульсов. В отдельных цепях БЭСМ необходимо формировать импульсы. Формирование импульсов требуется в тех случаях, когда на пути прохождения импульса имеются большие емкости и малые сопротивления. При этом импульсы искажаются по форме и меняют свою амплитуду. Между тем надежность работы отдельных устройств БЭСМ значительно повышается, если импульсы имеют стандартную форму и амплитуду. Кроме того, в некоторых устройствах приходится маломощным импульсом управлять большим количеством элементов, а также получать импульсы путем дифференцирования фронта перепада напряжения.

Для этих целей в машине применены специальные блоки формирователей. Таких блоков имеется два типа. Один из них, блок Φ , с более мощным выходным сигналом и более строгой его формой применяется в цепях сдвига и для стробирования импульсов сквозного переноса на арифметическом устройстве.

Второй тип формирователя используется в основном в цепях передачи кодов из одного устройства в другое. По схеме этот формирователь объединен с вентилем и называется блоком ВФ. Такая комбинация широко используется в выходных цепях машины.

Блоки катодных повторителей. В тех случаях, когда необходимо маломощные потенциалы подавать на большие нагрузки, в БЭСМ предусмотрены катодные повторители, которые, слабо нагружая питающие цепи, позволяют значительно повысить мощность выходного сигнала.

Установка катодных повторителей между отдельными устройствами развязывает работу этих устройств и тем самым повышает надежность. В машине использованы два типа катодных повторителей (применительно к различной величине нагрузки).

Блоки инверторов. Для инвертирования перепадов напряжения используются блоки инверторов, которые широко применяются в схемах устройства управления.

Первый тип блока инверторов является усилителем мощности и применяется как переходной элемент при

работе триггерных регистров на электронные коммутаторы. (В одном таком блоке расположены два инвертора.)

Второй тип блока инверторов предназначен для инвертирования перепадов напряжения и получения схем совпадения для потенциалов низкого уровня. (В таком блоке расположены четыре инвертора.)

Блок цепочки переносов. Для осуществления сквозного переноса при сложении двух чисел на сумматоре арифметического устройства предусмотрен блок Ц, называемый цепочкой переносов. Схема блока Ц смонтирована в четырехламповом каркасе.

Этот блок имеет три входа (см. рис. 7), один из которых подключен к единичному выходу триггера (*Вых «1» Т*), на другой вход подается стробирующий сигнал, обеспечивающий получение переноса в следующий разряд и, наконец, третий вход предназначен для приема импульса переноса из предыдущего разряда (*Вх П*). Перенос в следующий разряд выдается по цепи *Вых П*. Второй выход блока Ц, обеспечивающий установку на нуль триггера данного разряда при прохождении импульса переноса через цепочку, на рисунке не показан.

Блоки дешифраторов и диодных сборок. В дешифраторах, в схемах цепей объединения, а также в схемах совпадения используются блоки, выполненные на полупроводниковых диодах. Один такой блок можно использовать для дешифратора на четыре или на восемь выходов.

Схемы объединения выполняются на блоках диодных сборок. Комбинацией из этих блоков можно получать схемы дешифраторов на 16, 32 и более выходов.

Блок линий задержек. В отдельных блоках БЭСМ требуется задержать на некоторое время поступающий импульс, чтобы за это время успел сработать тот или другой элемент. Для этой цели в машине применены электромагнитные линии задержки, которые смонтированы в блоке ЛЗ.

В этом блоке размещены четыре линии задержки, со временем задержки 0,6 мксек каждая. Входы и выходы линий задержек, а также отводы через 0,2 мксек выве-

дены на отдельные контакты с целью иметь возможность путем последовательного включения получать время задержки до 2,4 мксек.

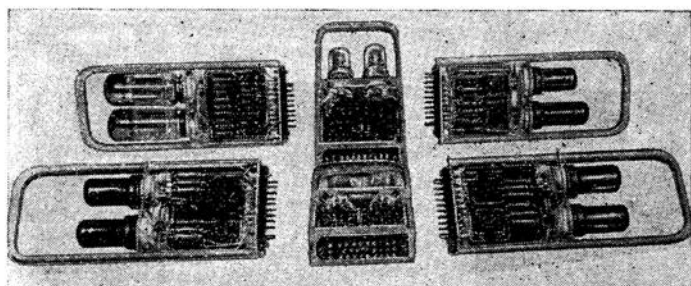


Рис. 6. Стандартные блоки.

Все перечисленные блоки относятся к группе стандартных блоков, они применяются во всех устройствах машины. Внешний вид некоторых из них показан на рис. 6.

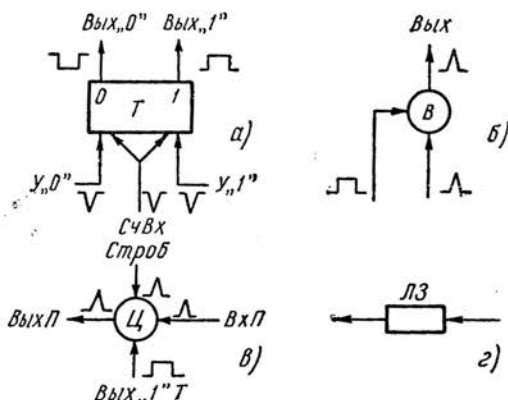


Рис. 7. Условные обозначения основных элементов: а — триггерная ячейка; б — вентиль; в — цепочка переносов; г — линия задержки

Другая группа блоков — нестандартные блоки — обладают некоторыми специфическими особенностями и используются только на каком-либо одном устройстве.

К таким блокам, например, можно отнести усилители записи и считывания на магнитных барабанах и магнитных лентах. Описание этих блоков будет дано при рассмотрении соответствующих устройств.

На рис. 7 даны условные обозначения основных элементов БЭСМ, которые применяются на функциональных схемах настоящего выпуска. Условные обозначения других элементов и подробный разбор их устройства даются во втором выпуске настоящего издания.

§ 2. Арифметическое устройство

Арифметическое устройство имеет следующие блоки:

- 1) первый блок запоминания кода числа — БЗ1Ч,
- 2) первый блок запоминания кода порядка числа — БЗ1П,

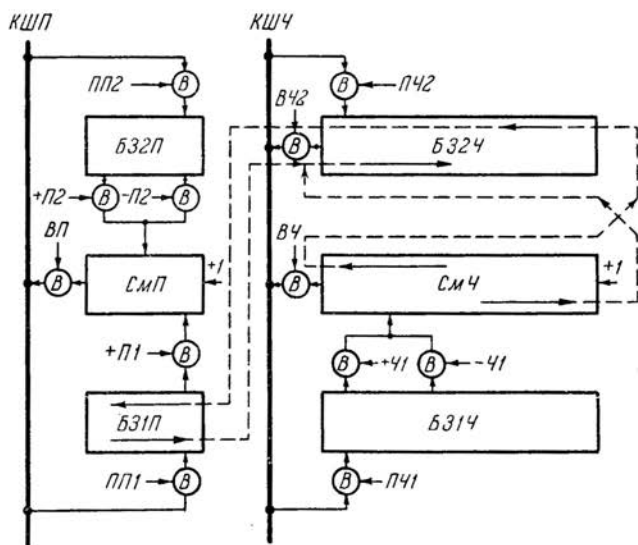


Рис. 8. Блок-схема арифметического устройства.

- 3) второй блок запоминания кода числа — БЗ2Ч,
- 4) второй блок запоминания кода порядка числа — БЗ2П,

- 5) блок суммирования кодов чисел — *СмЧ*,
- 6) блок суммирования кодов порядков — *СмП*.

Между отдельными блоками арифметического устройства должны быть предусмотрены связи, обеспечивающие следующие возможности:

- 1) прием кодов чисел и порядков с кодовых шин (*КШ*) на оба приемных регистра;
- 2) передача прямого или дополнительного кода с приемного регистра на сумматор;
- 3) выдача результата с сумматора прямым кодом на кодовые шины *КШ*.

Осуществление этих связей показано на рис. 8. Прием кодов осуществляется приемными вентилями *ПЧ1* и *ПП1*, *ПЧ2* и *ПП2*. На один вход этих вентилях подаются импульсы с кодовых шин *КШП* и *КШЧ*, а на другой — сигналы от центрального управления операциями (см. главу 3).

С целью сокращения количества аппаратуры предусмотрены лишь следующие передачи с регистров на сумматор: передача прямым и дополнительным кодами с первого регистра числа (*+Ч1* и *—Ч1*); передача прямым кодом с первого регистра порядков (*+П1*); передача прямым и дополнительным кодами со второго регистра порядков (*+П2* и *—П2*).

Второй регистр числа (*БЗ2Ч*) непосредственно с сумматором не связан. Передача с *БЗ2Ч* на *СмЧ* требуется лишь для операции сложения и производится после выравнивания порядков обоих слагаемых. Так как первое слагаемое находится на *СмЧ*, то *БЗ1Ч* свободно и передача второго слагаемого с *БЗ2Ч* на *СмЧ* производится путем выдачи кода с *БЗ2Ч* на кодовые шины (вентили *ВЧ2*) и приема его на *БЗ1Ч* (вентили *ПЧ1*), а затем с *БЗ1Ч* на *СмЧ* прямым или дополнительным кодом.

На арифметическом устройстве для порядков отсутствует передача дополнительным кодом с *БЗ1П* на *СмП*, так как для всех операций, предусмотренных в машине, подобного действия не требуется.

Выдача результата с сумматора на кодовые шины прямым кодом выполняется вентилями *ВЧ* и *ВП*. Кроме того, как уже упоминалось, предусмотрена выдача кода на кодовые шины с *БЗ2Ч* (*ВЧ2*).

Для сложения двух чисел необходимо предварительно выравнивать их порядки, что производится путем сдвига вправо кода числа, имеющего меньший порядок, на соответствующее количество разрядов. Так как заранее не известно, какое число имеет меньший порядок, то возможность сдвига кода числа вправо должна быть предусмотрена для обоих чисел, на блоках *СмЧ* и *БЗ2Ч*. (Сдвиги в блоках на рис. 8 условно обозначены стрелками.)

Для нормализации результата на блоке *СмЧ* должен быть предусмотрен также сдвиг влево.

При умножении множимое принимается на *БЗ1Ч*, а множитель — на *БЗ2Ч*. Операция умножения осуществляется следующим образом: код множителя на *БЗ2Ч* сдвигается вправо на один разряд. Если при этом на правом выходе *БЗ2Ч* получается импульс, соответствующий коду 1, то множимое прибавляется с *БЗ1Ч* на *СмЧ*. Затем частичное произведение на *СмЧ* сдвигается вправо на один разряд и поступает на старший разряд *БЗ2Ч* (см. пунктир на рис. 8). Таким образом, для осуществления операции умножения необходимо обеспечить возможность сдвига кодов чисел вправо как на *БЗ2Ч*, так и на *СмЧ*.

При делении делимое устанавливается на *СмЧ*, а делитель — на *БЗ1Ч*. Операция деления осуществляется путем вычитания или прибавления кода делителя к содержимому сумматора числа и последующего сдвига этой величины влево. Коды, выходящие со старшего разряда *СмЧ*, определяют коды частного, которые поступают на младшие разряды *БЗ2Ч* (см. пунктир на рис. 8). Чтобы старшие разряды частного заняли старшие разряды *БЗ2Ч*, необходимо обеспечить возможность сдвига кода на *БЗ2Ч* влево. Таким образом, операция деления требует осуществления сдвига кодов влево на *БЗ2Ч* и *СмЧ*.

Для осуществления операции сдвига кода числа по всем разрядам (39 разрядов), а также для преобразования кода из последовательного в параллельный и обратно при обращении к внешнему запоминающему устройству на магнитных барабанах и лентах предусмотрен сдвиг вправо и влево на блоке *БЗ1П*. Причем

при сдвиге влево код со старших разрядов $B324$ поступает на младшие разряды $B31П$, а при сдвиге вправо код с младших разрядов $B31П$ поступает на старшие разряды $B324$.

Данная схема связей блоков $AУ$ позволяет за счет несколько более сложной передачи кодов уменьшить количество требуемой аппаратуры.

Значительное внимание было уделено ускорению операции сложения кодов чисел, являющейся основной операцией. При сложении кодов чисел необходимо учитывать двоичный перенос из разряда в разряд. Наиболее простое решение получается по схеме, изображенной на рис. 9.

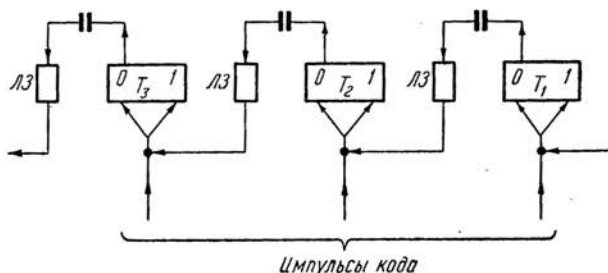


Рис. 9. Элементарная схема сумматора.

При подаче на счетный вход триггерной ячейки импульса кода она переходит из одного состояния равновесия в другое. Это эквивалентно прибавлению единицы к коду, имевшемуся в триггерной ячейке. При переходе триггерной ячейки из положения «1» в положение «0» на соответствующем аноде триггера возникает импульс переноса $ИП$, который подается на вход следующей триггерной ячейки. Для четкой работы триггера необходимо, чтобы импульс переноса поступал на триггер, после того как тот займет свое новое положение после прибавления импульса кода. Для этой цели в цепи импульса переноса необходимо установить задержку $ЛЗ$.

Несмотря на свою простоту, эта схема имеет весьма существенный недостаток. В наиболее неблагоприятном случае, когда после подачи импульсов кода все триггерные

ячейки, кроме крайней правой, встанут в положение «1», а в крайней правой (в первой) ячейке произойдет переход из положения «1» в положение «0», возникает импульс переноса из первой ячейки во вторую. Этот импульс переноса с задержкой переведет вторую ячейку из положения «1» в положение «0», что вызовет импульс переноса из второй ячейки в третью. Этот импульс переноса с задержкой переведет третью ячейку из положения «1» в положение «0». Процесс будет продолжаться до тех пор, пока последняя триггерная ячейка не перейдет из положения «1» в положение «0».

Таким образом, описанная схема дает каскадный перенос из разряда в разряд. Для рассматриваемого наиболее неблагоприятного случая суммарное время переносов будет равно сумме времен задержек в цепях переноса.

Каскадный характер переносов сильно задерживает операцию сложения. Поэтому была разработана схема,

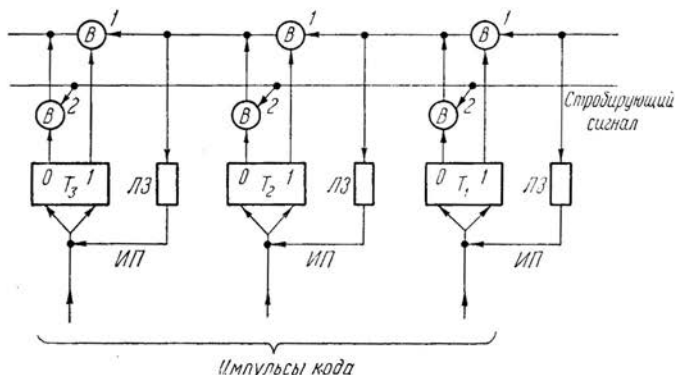


Рис. 10. Схема сумматора со сквозным переносом.

исключающая каскадный характер двоичных переносов (рис. 10). В основу разработки этой схемы был положен следующий принцип: импульс переноса распространяется вдоль ячеек, стоящих в положении «1»; если на пути находится ячейка, стоящая в положении «0», то импульс переноса переключает ее в положение «1», но в следующую ячейку не поступает.

Этим условием можно воспользоваться для исключения каскадного характера импульсов переноса. Каждый возникший импульс переноса направляется не только в соседнюю ячейку, но и вдоль всех остальных разрядов по цепочке переноса, ответвляясь по пути во все прилегающие ячейки. Распространение импульса переноса вдоль цепочки контролируется вентилями 1, управляемыми анодными напряжениями триггеров. Ячейка, стоящая в положении «0», запирает вентиль и не дает возможности импульсу переноса пройти по цепочке дальше. Таким образом, импульс переноса направляется сразу во все требуемые разряды.

Импульс переноса должен поступать на цепочку переносов лишь после окончания переходного процесса в триггерах, который возник в результате подачи кодов второго числа. С другой стороны, вторичные импульсы переноса, возникающие при переходе триггерной ячейки из положения «1» в положение «0» от импульса переноса, не должны поступать на цепочку переносов, так как это исказит результат. Для обеспечения этого импульс переноса подается на цепочку переносов через вентиль 2, управляемый сдвинутым импульсом, который принято называть стробирующим сигналом, а в цепи поступления импульса переноса в триггерную ячейку создается дополнительная задержка.

Подобная схема обеспечивает отсутствие каскадного характера переноса. Время сложения двух кодов составляет около 3 *мксек*.

Для уменьшения времени, требуемого на сдвиг кодов, принята схема, изображенная на рис. 11. В данном случае импульс сдвига подается одновременно на вентили, управляемые триггерами, и на входы установки триггеров на «0». Импульс сдвига успевает пройти через вентиль, так как задний фронт триггера запаздывает по отношению к переднему фронту импульса U_{00} на время порядка $0,1 \div 0,12$ *мксек*. Пройдя через электромагнитную линию задержки, импульс сдвига ставит триггер в положение «1». Тот факт, что триггер, имея рабочую частоту 1 *мгц*, ставится в положение «1», спустя всего 0,3 *мксек* после установки на «0»,

не мешает его устойчивой работе, так как импульс $У«1»$ имеет несколько большую амплитуду и длительность.

В результате применения этой схемы время, требуемое для сдвига кода на один разряд, составляет около 1,2 мксек. Это позволило создать запас как для сложения, так и для сдвига кодов. (На сложение кодов отводится 3,5 мксек и на сдвиг кодов 1,5 мксек).

Разработанное арифметическое устройство обладает достаточной гибкостью, позволяя весьма просто осуществлять самые разнообразные операции.

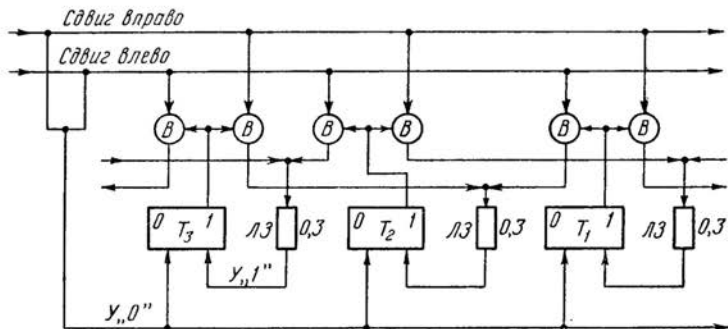


Рис. 11. Схема осуществления сдвига.

В приложении приведены функциональные схемы арифметических устройств для чисел и для порядков. Эти схемы являются упрощенными и даны для облегчения разбора последовательности выполнения операций, принятых на машине БЭСМ (см. часть II). Некоторые цепи на схемах изображены условно. Например, в цепях сдвига и цепочки переносов не показаны вентили и линии задержки в каждом разряде.

§ 3. Устройство управления

Устройство управления должно обеспечить следующие этапы работы машины:

а) автоматический ввод программы вычислений и исходных данных в машину;

б) выборку кодов команд из оперативного запоминающего устройства;

в) выборку кодов чисел из оперативного запоминающего устройства;

г) непосредственное выполнение операции;

д) выдачу результатов вычислений в оперативное запоминающее устройство;

е) выдачу окончательных результатов из машины.

Общая блок-схема устройства управления БЭСМ представлена на рис. 12. Она состоит из следующих основных блоков:

1) блок центрального управления машиной (ЦУ),

2) блок управления командами (УК),

3) блок центрального управления операциями (ЦУОп),

4) блок местного управления операциями (МУОп),

5) пульт управления (ПУ),

6) задающий генератор (ЗГ).

Эти блоки обеспечивают связь и взаимодействие между арифметическим устройством машины, оперативным запоминающим устройством, вводным и выводным устройствами и внешним запоминающим устройством.

Центральное управление (ЦУ). Назначением блока ЦУ является обеспечение цикла работы машины. За время такого цикла должна быть выбрана очередная команда, выбраны числа, с которыми производится та или иная операция, получен результат, который запоминается в ОЗУ.

При выполнении какой-либо операции ЦУ выдает серию сигналов, которые, проходя через схему ЦУОп,

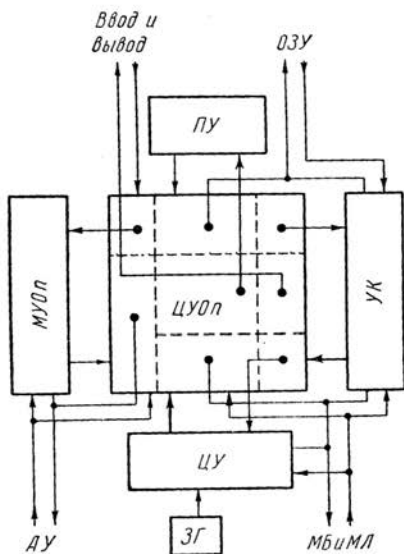


Рис. 12. Блок-схема устройства управления.

воздействуют на различные устройства машины. Причем в каждый момент времени выдается единственный сигнал, действующий по своей самостоятельной цепи. Число цепей, связывающих блок *ЦУ* с блоком *ЦУOn*, соответствует числу сигналов, выдаваемых за один цикл. Количество сигналов, выдаваемых блоком *ЦУ*, определяется числом элементарных стандартных действий, которые необходимо осуществить при выполнении операций, принятых на машине. К таким элементарным стандартным действиям можно отнести: установку на «0» различных схем машины; выдачу адресов; прием кодов чисел и команд; передачу кодов из одного регистра в другой и т. д.

Другим весьма важным фактором, характеризующим работу блока *ЦУ*, является продолжительность цикла. Это время в значительной степени определяет скорость машины. Для БЭСМ оно составляет 65 мксек. Большинство операций укладывается в данное время.

Управление командами. Блок управления командами (*УК*) определяет номер (адрес) очередной команды, которую должна выполнить машина; осуществляет ее прием; сохраняет код операции в течение всего цикла *ЦУ*.

Управление командами состоит из блока запоминания команд (*БЗК*), двух счетчиков центрального и местного управления командами (*ЦУК* и *МУК*) и коммутатора операций (*КOn*).

Коммутатор операций (*КOn*) связывает устройство управления командами с устройством управления операциями. Он преобразует код операции в управляющее напряжение на одной из 32 выходных цепей. Таким образом, каждому номеру операции будет соответствовать своя возбужденная цепь на выходе *КOn*.

Управление операциями. Как уже упоминалось, в машине имеются два блока управления операциями: центральное управление операциями (*ЦУOn*) и местное управление операциями (*МУOn*). Для команд, не требующих местного управления операциями, время выполнения определяется блоком центрального управления машиной (*ЦУ*). На время работы местного управления операциями работа *ЦУ* приостанавливается. Таким образом, время, затрачиваемое на выполнение

любой команды, определяется суммой времен одного цикла центрального управления и работой местного управления, если оно участвует в данной операции.

В машине на местное управление выделены следующие элементарные действия:

- 1) нормализация чисел влево,
- 2) выравнивание порядков при сложении и вычитании,
- 3) выполнение собственно операции умножения,
- 4) выполнение собственно операции деления,
- 5) выполнение собственно операции сдвига,
- 6) выполнение собственно операции выделения целой части.

Элементарные действия, необходимые для выполнения той или иной команды, идут в определенной последовательности во времени, а их характер зависит от кода операции. В ряде случаев характер элементарных действий зависит также от результата, получающегося на том или ином устройстве машины. Приведем несколько примеров: если при команде сложения принятое число имеет положительный знак, то его надо прибавить, если же знак — отрицательный, то его следует вычесть; выравнивание порядков при сложении чисел зависит от разности порядков; при командах сравнения переход к номеру команды, указанному в третьем адресе, зависит от результата сравнения двух чисел; непосредственный обмен кодами между оперативным запоминающим устройством и магнитным барабаном начинается при совпадении текущего номера числа с заданным; и т. д.

Таким образом, в общем случае выработка сигнала для того или иного элементарного действия определяется:

- а) сигналом, идущим в определенный момент времени,
- б) сигналом кода операции,
- в) сигналами от арифметического и других устройств машины.

Для центрального управления операциями (ЦУОп) сигналы, определяющие момент выполнения той или иной элементарной операции, создаются блоком ЦУ. За

один цикл работы *ЦУ* вырабатывается серия сигналов, идущих с промежутками 2,5 мксек (цикл работы *ЦУ* составляет 65 мксек). При переходе на местное управление операциями *ЦУ* останавливается на соответствующем сигнале и вновь запускается при окончании операций на местном управлении, продолжая при этом прерванную последовательность временных сигналов.

Сигналы, определяющие код операции, поступают на блок *ЦУОн* от коммутатора операций (*KОн*), а сигналы, характеризующие то или иное состояние других устройств машины, подаются от соответствующих триггерных ячеек этих устройств. Сам блок центрального управления операциями в основном состоит из вентилей, диодных сборок и усилителей.

В зависимости от назначения выходных цепей блока *ЦУОн* его можно разбить на ряд схем:

1. Схема управления работой оперативного запоминающего устройства.
2. Схема управления внешними запоминающими устройствами на магнитных лентах и магнитных барабанах.
3. Схема управления работой арифметического устройства.
4. Схема управления вводными и выводными устройствами.
5. Схема управления блоком центрального управления.
6. Схема управления блоком управления командами.
7. Схема управления блоком местного управления операциями (пуск схем местного управления операциями).
8. Схема, осуществляющая связь с пультом управления.

Блок местного управления операциями (*МУОн*) принципиально отличается от блока центрального управления операциями тем, что моменты времени выполнения тех или иных элементарных операций уже не определяются работой блока *ЦУ*. Как правило, работа местного управления сводится к последовательному повторению той или иной комбинации чередования элементарных операций. Количество таких повторений или за-

дается (например, при умножении), или же определяется результатом вычислений (например, при нормализации чисел). Выбор той или иной комбинации чередования элементарных операций производится блоком *ЦУОн*, который при переходе на местное управление возбуждает одну из цепей блока *МУОн*. По окончании выполнения операций на блоке *МУОн* автоматически запускается *ЦУ* и управление операциями переключается на *ЦУОн*.

В управление операциями входит также схема блокировки (*СхБл*). Назначением этой схемы является прекращение операции и изменение чередования элементарных операций в случае возникновения на *АУ* определенных результатов. Например, при сложении и вычитании исключается выравнивание порядков при расхождении их больше чем на 32 и сразу производится передача в запоминающее устройство большего числа.

Схема блокировки обеспечивает также автоматический аварийный останов машины при выходе получаемого числа из располагаемого количества разрядов. Конструктивно схема блокировки размещается в блоке *МУОн*.

Пульт управления. Управление машиной производится с пульта управления. Помимо автоматической работы машины, предусмотрена работа по циклам и от одиночных сигналов. Работа по циклам заключается в том, что блок центрального управления (*ЦУ*) останавливается в конце каждого цикла. Каждое нажатие пускового ключа дает один цикл работы *ЦУ*, т. е. выполнение одной команды. При работе от одиночных сигналов каждое нажатие ключа обеспечивает выдачу лишь одного сигнала от блока *ЦУ*. Таким образом, имеется возможность проследить работу машины по отдельным тактам. За время одного такта производится одно из элементарных действий выполняемой операции.

Переключатель режимов работы машины, а также ключи пуска и останова расположены на пульте управления. Там же расположены кнопки управления, позволяющие устанавливать любые коды на различных регистрах машины, а также гасить их.

На пульт же выведена сигнализация (на неоновых лампах) для визуального наблюдения за работой

наиболее важных устройств машины. К таким устройствам в первую очередь относятся все числовые или знаковые разряды арифметического устройства, регистры команд и счетчики операций, отдельные элементы, находящиеся в устройстве управления и в запоминающих устройствах. Все это представляет своеобразную mnemonicкую схему вычислительной машины. Естественно,

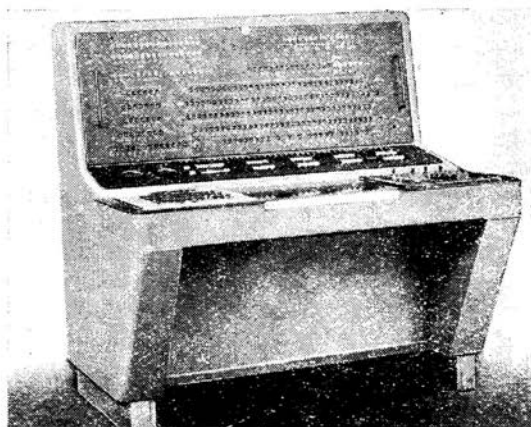


Рис. 13. Пульт управления.

что при работе машины, учитывая большую скорость вычислений, оператор не в состоянии визуально контролировать ход решения задачи. Но автоматический останов, который произойдет в результате какой-либо ошибки в работе машины, позволит оператору с помощью mnemonicческой схемы обнаружить причину появления ошибки. Пульт управления машины БЭСМ представлен на рис. 13.

§ 4. Оперативное запоминающее устройство

За последнее время получили широкое применение запоминающие устройства на ферритовых сердечниках с прямоугольной петлей гистерезиса, которые имеют ряд существенных преимуществ по сравнению с запоминающими устройствами на электронно-лучевых трубках.

Основными недостатками последних являются следующие.

1. Выбор ячейки запоминающего устройства производится путем установки луча в заданную точку экрана. Для установки луча требуется подать на отклоняющие пластины трубки напряжения вполне определенной величины, причем с большой точностью и быстрым установлением. Таким образом, выбор ячейки запоминающего устройства производится по количественному признаку (по величине отклоняющего напряжения), в то время как все остальные устройства машины работают по качественному признаку (наличие или отсутствие сигнала). Качественный признак обеспечивает значительно большую надежность работы, чем количественный. Кроме того, в связи с использованием количественного признака приходится предъявлять более высокие требования к источникам питания.

2. Несмотря на принятые меры, длительность хранения записанных кодов в работающей трубке сравнительно мала, что требует частого восстановления записи и создает неудобства при эксплуатации.

Использование ферритов для построения оперативных запоминающих устройств позволяет освободиться от указанных недостатков и существенно повысить надежность работы вычислительных машин, уменьшив их размеры и сократив расходы, связанные с эксплуатацией.

Принцип действия запоминающего устройства на ферритах основан на способности магнитного материала намагничиваться до состояния насыщения в положительном или отрицательном направлении в зависимости от того, какой сигнал (положительный или отрицательный) подается через обмотку сердечника.

Запись двоичного кода можно осуществить, условно приняв одно из магнитных состояний сердечника за код 1, а противоположное — за код 0. Условившись, что коду 1 соответствует положительное направление, для считывания информации будем подавать размагничивающие сигналы отрицательной полярности. Если при этом сердечник хранил код 1, то произойдет его перемагничивание и в выходной обмотке наведется импульс напряжения. Если же сердечник хранил код 0, то изменения его

состояния не произойдет и в выходной обмотке сигнала не возникнет. Естественно, что после выборки кода необходимо восстановить первоначальное состояние сердечника, осуществляемое специальной схемой.

Запоминающее устройство на ферритовых сердечниках может хранить записанную информацию в течение неограниченного времени. Энергии на поддержание информации при этом не требуется.

На машине БЭСМ разработано оперативное запоминающее устройство на ферритах. Время одного обращения к ОЗУ составляет около 10 мксек, а емкость его — 2047 двоичных 39-разрядных чисел. Из известных

схем — матричной и схемы Z — выбрана схема Z.

В матричной схеме (рис. 14) перемагничивание сердечника производится от воздействия двух токов (J_x и J_y), идущих по двум взаимно-перпендикулярным шинам. На сердечник a , находящийся на пересечении возбужденных шин, действует суммарное поле от двух токов и он перемагничивается. На остальные сердечники

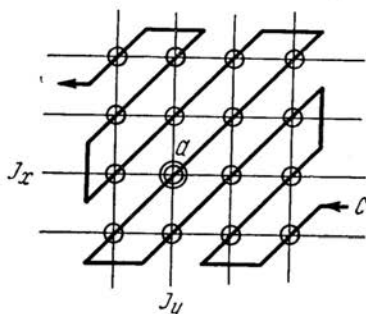


Рис. 14. Матричная схема для запоминания одного разряда.

расположенные на возбужденных шинах, действует поле лишь от одного тока, величина которого недостаточна для перемагничивания сердечника, поэтому эти сердечники останутся в неизменном состоянии.

Считывающая обмотка (C) пронизывает все сердечники одного разряда. На этом проводе, помимо полезного сигнала, возникают импульсы помехи от полувывбренных сердечников, так как реальная форма петли гистерезиса отличается от идеально прямоугольной. Поэтому приходится принимать ряд мер для снижения уровня помех. При применении матричных схем накладываются определенные требования с точки зрения стабильности амплитуды и длительности тока, что делает невозможным проведение работы в форсированном режиме, т. е. в режиме с большими амплитудами токов.

Кроме того, матричная схема требует тщательного отбора сердечников с точки зрения однородности их характеристик.

В схеме Z считывание производится одним током J_z , идущим по проводу, пронизывающему числовую линейку — сердечники всех разрядов выбранного числа (рис. 15). Амплитуда тока считывания может быть любой

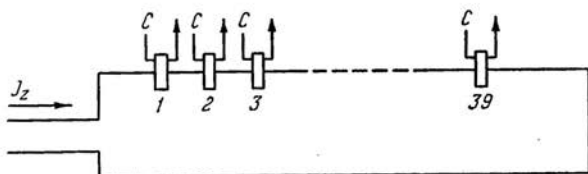


Рис. 15. Числовая линейка схемы Z .

величины. Это позволяет форсированно перемагничивать сердечники, что повышает быстродействие и увеличивает амплитуду выходного сигнала. При таком принципе считывания не возникает полувывбранных сердечков, как это имеет место в матричной схеме. Поэтому на считывающем проводе C , который пронизывает сердечники одного разряда всех чисел, отсутствует помеха, что позволяет упростить цепи считывания и повышает надежность работы.

Недостатком запоминающего устройства типа Z является необходимость иметь вентиль на каждое число.

Для этой цели удобно применять магнитные вентили или, как их часто называют, координатные трансформаторы, работающие по принципу совпадения токов.

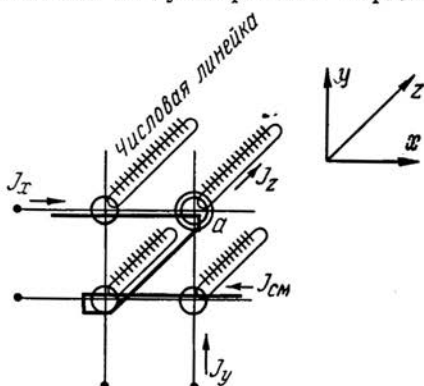


Рис. 16. Схема выборки числовой линейки.

В схеме Z , помимо двух токов J_x и J_y , идущих по двум взаимно-перпендикулярным проводам (как в матричной схеме), по отдельной обмотке подается еще постоянный ток смещения $J_{см}$ (рис. 16). За счет тока смещения рабочая точка трансформатора выбирается таким образом, что при возбуждении только по одному проводу трансформатор не перемагничивается. При возбуждении по двум взаимно-перпендикулярным проводам трансформатор, находящийся на их пересечении, перемагнитится и на выходной его обмотке возникнет сигнал J_z , осуществляющий считывание кода с выбранной числовой линейки (обмотки считывания с разрядов на рисунке не показаны). По окончании возбуждающих импульсов трансформатор возвращается снова в рабочую точку.

Запись кода или регенерация его после считывания в схеме Z производится следующим образом. Все сердечники одного разряда, помимо обмотки считывания, пронизаны также обмоткой записи. После окончания сигнала считывания все сердечники выбранного числа будут намагничены в отрицательном направлении, соответствующем коду 0. После считывания производится запись кода. Для этого от координатного трансформатора в сердечники выбранного числа поступает импульс тока положительной полярности (противоположной импульсу считывания). В те разряды, в которые хотят записать код 1, по обмотке записи подается сигнал положительной полярности, а в те разряды, в которых следует записать код 0, сигнал по обмотке записи идет отрицательной полярности. Положительный сигнал по обмотке записи, складываясь с сигналом от координатного трансформатора, перемагнитит сердечник в положительном направлении (код 1). Отрицательный сигнал записи вычитается из сигнала координатного трансформатора и сердечник останется намагниченным в отрицательном направлении (код 0). Амплитуда сигнала в обмотке записи не должна превышать определенной величины, с тем чтобы не разрушать записанную информацию в сердечниках не выбранных чисел (обмотка записи пронизывает все сердечники своего разряда). В соответствии с амплитудой сигнала записи выбирается и амплитуда

положительного сигнала от координатного трансформатора.

Блок-схема оперативного запоминающего устройства на ферритах показана на рис. 17. Она состоит из: регистра адреса $РА$, на который принимается код адреса, поступающий из машины по шинам $КША$; регистра числа

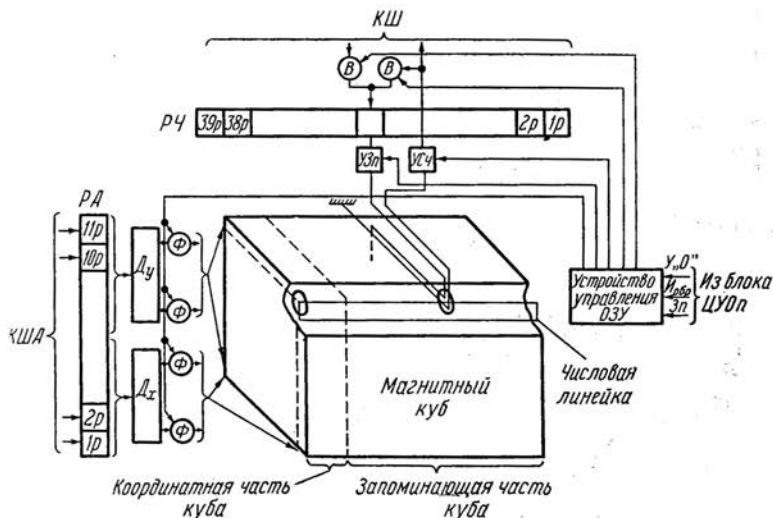


Рис. 17. Блок-схема запоминающего устройства на ферритах.

$РЧ$, который служит для приема кода числа с шин $КШ$ при записи и для осуществления регенерации при считывании; магнитного куба; устройства управления, обеспечивающего получение определенной последовательности импульсов. (Сигналы установки нуля ($У«0»$), записи ($Зп$) и обращения к оперативному запоминающему устройству ($И_{обр}$) поступают из устройства управления машины.)

Код, принятый на регистр $РА$, преобразуется дешифраторами $Д_x$ и $Д_y$, один из которых управляет формирователями (Φ) для осуществления выборки числа по оси x , а другой — по оси y .

Для производства записи и считывания кодов имеются специальные усилители ($УЗп$ и $УСч$).

Магнитный куб собирается из отдельных кассет, конструктивно выполненных в виде рамок, на которых

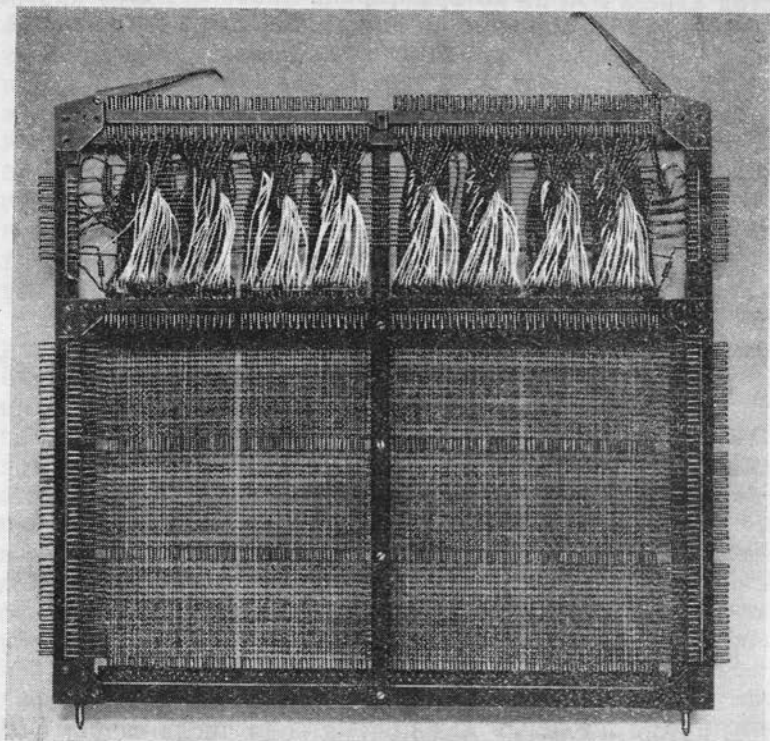


Рис. 18. Кассета магнитного куба.

производится расшивка ферритовых сердечников. На одной кассете (рис. 18) размещается 128 39-разрядных двоичных чисел.

§ 5. Внешнее запоминающее устройство

Для расширения круга решаемых задач, требующих большого объема памяти, в машине предусмотрено дополнительное запоминающееся устройство (внешнее),

выполненное на магнитных барабанах и магнитных лентах.

Магнитный барабан предназначен для хранения достаточно большого количества кодов.

Емкость магнитного барабана ограничивается площадью его боковой поверхности и допускаемой плотностью записи. На машине БЭСМ приняты два магнитных барабана, каждый из которых имеет емкость в 5120 чисел (пять групп по 1024 числа).

На магнитном барабане установлены 84 считывающих, записывающих головок, из них 80 головок для кодов, одна головка для синхронизирующих импульсов, одна — для фиксации начала отсчета и две резервные. Между головками и барабаном имеется воздушный зазор порядка 35 микрон. На одной дорожке барабана записывается 64 числа. Плотность записи составляет около трех импульсов на миллиметр. Барабан вращается со скоростью 750 об/мин. Среднее время ожидания подхода требуемого кода под магнитную головку составляет 40 мсек, а последующая выборка или запись происходит со скоростью 800 чисел в секунду. Частота импульсов магнитного барабана получается около 35 кГц. Амплитуда импульса считывания на магнитной головке составляет 50—60 мВ. Усилители считывания и записи предусмотрены на каждую группу. Переключение их с дорожки на дорожку в пределах группы осуществляется автоматически специальной лампово-диодной схемой.

Для большинства решаемых на БЭСМ типовых задач время обращения к барабанам составляет в среднем 8—10% от общего времени (в случае принятой последовательной системы обмена кодами). При параллельном способе это время могло бы быть около 1%. Однако при этом значительно увеличилось бы количество электронных ламп, так как на каждый разряд был бы нужен отдельный комплект усилителей считывания и записи, а для связи с машиной число входных и выходных элементов вместо одного возросло бы до 39. Поэтому с целью сокращения количества аппаратуры система считывания и записи на магнитном барабане принята последовательная. Преобразование же из последовательной системы в параллельную и обратно, необходимое при

обмене кодами между магнитным барабаном и оперативной памятью, осуществляется на арифметическом устройстве. На рис. 19 показаны магнитные барабаны, установленные на машине БЭСМ.

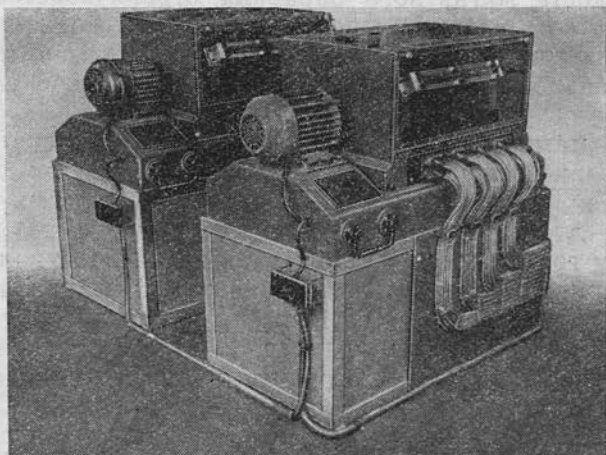


Рис. 19. Магнитные барабаны.

Запоминающее устройство на магнитных лентах осуществляется в виде четырех магнитофонов последовательного действия с шириной ленты 6,5 мм. На ленте имеются две дорожки. Одна дорожка служит для записи синхронизирующих импульсов, а другая — для записи кода. На кодовой дорожке записывается также номер данной группы кодов.

На магнитофонах предусмотрен как прямой, так и обратный ход магнитной ленты. Рабочим является лишь прямой ход. При прямом ходе осуществляется считывание любой ранее записанной группы по заданному ее номеру или производится запись. Обратный ход служит для автоматического подвода прошедшей группы к магнитной головке. Это позволяет оперативно использовать магнитные ленты в режиме считывание — запись в пределах одной или нескольких групп.

Длина магнитной ленты на каждом магнитофоне составляет около 200 м. На одном миллиметре записыв-

вается восемь импульсов. На одной ленте может храниться до 30 000 чисел (с учетом промежутков между группами), а всего на четырех магнитофонах около 120 000 чисел.

Бобины с лентами можно быстро менять, что обычно производится без остановки машины. Установка головок на магнитофонах выполнена с точностью, обеспечивающей возможность перестановки запасных лент

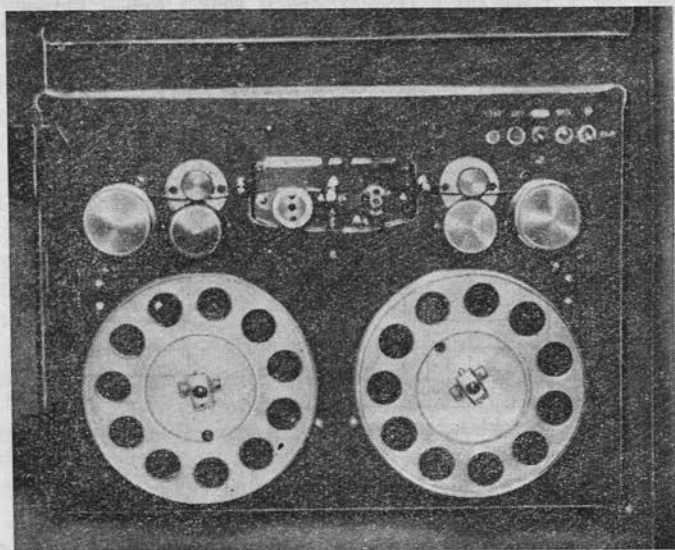


Рис. 20. Вид магнитофона.

с одного магнитофона на другой. Скорость ленты составляет 2 м/сек; частота импульсов — 16 кГц и скорость выборки или записи — 400 чисел в секунду. Амплитуда импульса считывания на магнитной головке составляет 15—20 мВ. На четырех магнитофонах предусмотрен один комплект усилителей записи и считывания для кодовых и синхронизирующих импульсов. Коммутация усилителей осуществляется релейной схемой.

На рис. 20 показан магнитофон запоминающего устройства на магнитных лентах.

На рис. 21 приведена блок-схема запоминающего устройства на магнитных барабанах и лентах. Она включает в себя: два барабана *МБ1* и *МБ2* со своими приводами *П*; четыре ленты *МЛ1*, *МЛ2*, *МЛ3*, *МЛ4* и перфорационную ленту *ПЛ* (приводы лент управляются); блоки усилителей считывания *УСч* и записи

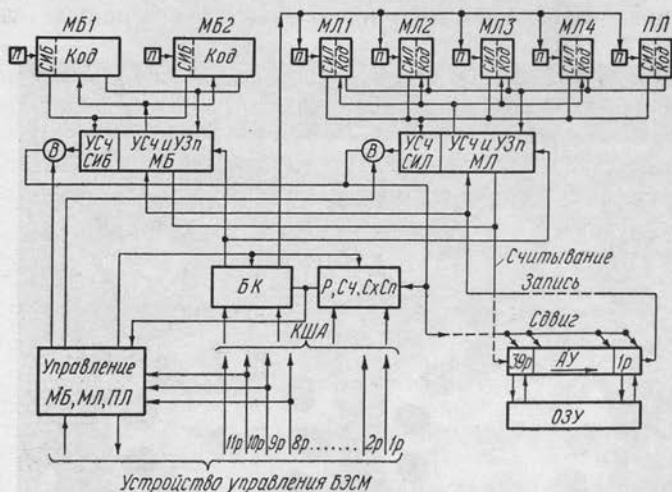


Рис. 21. Блок-схема МЗУ.

УЗп для кодовых и синхронизирующих импульсов (*СИБ* — при обращении к барабану, и *СИЛ* — к ленте); блока коммутации *БК*; блока регистров — счетчиков — схем совпадения (*Р*, *Сч*, *СхСп*) и, наконец, блока управления *МБ*, *МЛ*, *ПЛ*.

Блок *БК* предназначен для выбора того или другого барабана и заданной на нем дорожки. В случае обращения к магнитной ленте *БК* выбирает номер магнитофона.

Блоки регистров (*Р*) служат для промежуточного хранения адресов кодов на *МБ* или *МЛ* при обращении к ним. Счетчики (*Сч*), на которые поступают синхронизирующие импульсы с *МБ* и *МЛ*, регистрируют количество считываемых или записываемых кодов. Сигналы о том,

что на запись или считывание прошло необходимое количество кодов, обеспечивают схемы совпадения ($CxSn$).

Блок управления задает определенную последовательность управляющих сигналов, которая осуществляет операцию обмена кодами.

При обращении к внешнему запоминающему устройству центральное управление машиной останавливается и такт работы задается синхронизирующими импульсами *СИБ* или *СИЛ*. В случае операции считывания код с *УСч МБ* или *УСч МЛ* последовательно поступает на старшие разряды арифметического устройства *АУ*. Синхронизирующие импульсы *СИБ* и *СИЛ* обеспечивают сдвиг в сторону младших разрядов. После того как число будет полностью считано, оно параллельно отсылается в *ОЗУ* и начинается считывание другого числа. Когда заданная группа чисел будет передана в *ОЗУ*, происходит пуск центрального управления и машина переходит к выполнению следующей команды; запись производится аналогичным образом в обратном порядке.

Автоматика управления магнитных барабанов и магнитных лент в основной своей части являются общими.

Считывание с перфоленты происходит таким же образом, как и с магнитных лент.

§ 6. Ввод исходных данных и печатание результатов вычислений

Первоначальный ввод чисел и команд в машину осуществляется с перфорированной бумажной ленты в виде последовательного кода. На перфоленте имеются две дорожки: одна для синхронизирующих импульсов, другая для импульсов кода. Считывание отперфорированных кодов производится посредством фотоэлектрического датчика. Скорость считывания составляет 20 чисел в секунду.

Подготовка перфолент производится вне машины на специальных перфораторах. Для сверки двух перфолент применяется фотоэлектрический контрольный.

После первоначального ввода программы в машину она может быть в дальнейшем переписана на магнитную

ленту. Хранение программ на магнитных лентах имеет большой смысл, так как ввод с них осуществляется быстрее, чем с перфолент.

При создании быстродействующих электронных счетных машин большое внимание уделяется выводным устройствам. До последнего времени на машине БЭСМ основным устройством вывода окончательных результатов являлось фотопечатающее устройство. Вывод результатов вычислений производился путем записи их на магнитную ленту с последующей переписью их на киноплёнку вне машины на специальном устройстве. Последовательный код с магнитной ленты поступал на сдвиговый регистр, на котором преобразовывался в параллельный. Соответствующие дешифрирующие устройства для каждой цифры управляли точечными источниками света, которые проектировали на киноплёнку изображение цифры. Скорость работы фотопечатающего устройства составляла 200 чисел в секунду. Проявление киноплёнки производилось на проявочной машине. Размножение копий на бумагу также осуществлялось на специальной машине.

Однако эта система при многих положительных качествах являлась громоздкой и имела высокую стоимость; а необходимые при этом фотопроцессы задерживали получение результатов, что создавало неудобства при эксплуатации машины.

В качестве основного выводного устройства в настоящий момент на машине БЭСМ принято электромеханическое печатающее устройство, которое, обладая хорошими графическими качествами и сравнительно малой стоимостью, позволяет получать результаты вычислений непосредственно на бумагу. Сочетание механической печати с электронным управлением дает возможность повысить скорость вывода такого типа печатающего устройства до 20 чисел в 1 сек.

В данном устройстве используется непрерывно вращающийся валик, на котором выгравированы цифры (количество цифр устанавливается применительно к машине; для БЭСМ оно равно 16). На одной оси с валиком расположен цилиндр с отверстиями, соответствующими цифрам (кодовые отверстия). Кроме кодовых,

имеются синхронизирующие и управляющие отверстия. Установленные фотоэлектрические датчики посылают управляющие сигналы в специальную электронную схему и к молоточкам в момент, когда подойдет нужная цифра на шрифтовом валике. Молоточки через красящую среду (копировальная бумага) печатают цифры на бумагу.

Набор цифр от 0 до 15 сделан с целью иметь возможность печатать как десятичные, так и шестнадцатеричные числа, которые поступают на печатающее устройство в двоичном изображении. Печатание шестнадцатеричных чисел необходимо при выводе команд. В случае печатания только десятичных чисел (цифры $0 \div 9$) можно, уменьшив диаметр шрифтового валика, увеличить скорость вывода до 30 чисел в 1 секунду.

Конструктивно вводное (перфорационная лента) и выводное (печать) устройства выполнены на одной стойке (рис. 22).

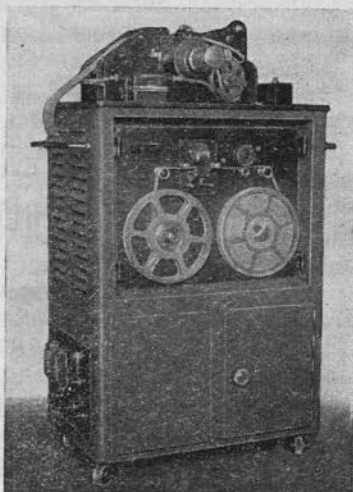


Рис. 22. Перфолента и печатающее устройство.

ЧАСТЬ II

МЕТОДИКА ВЫПОЛНЕНИЯ ОПЕРАЦИЙ

ГЛАВА 5

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

§ 1. Сложение и вычитание

При представлении на машине чисел с учетом их порядков необходимо до производства сложения или вычитания двух чисел выравнивать их порядки. Для этой цели следует число, имеющее меньший порядок, сдвинуть вправо на количество разрядов, равное разности порядков этих двух чисел.

Пусть первое число будет $2^p \cdot A$, а второе $2^q \cdot B$. Если прибавить к сумматору порядков порядок p первого числа (подать прямым кодом) и затем вычесть порядок q второго числа (подать дополнительным кодом), то на сумматоре порядков получаем величину $p - q$.

Если $p > q$, то код знака на сумматоре порядков будет 0 (разность положительна). Это указывает на то, что порядок первого числа больше порядка второго, и следовательно, второе число надо сдвинуть вправо на количество разрядов, равное этой разности.

Сдвигаем второе число вправо и при каждом сдвиге на один разряд вычитаем единицу из сумматора порядков до тех пор, пока на сумматоре порядков не получится нуль (0.00000), т. е. второе число сдвигается на $p - q$ разрядов вправо. Каждый сдвиг числа на один разряд вправо соответствует увеличению его порядка на единицу. Следовательно, после указанного сдвига второго числа его порядок будет $q + (p - q) = p$. Таким образом, порядки обоих чисел будут выравнены и можно производить с ними операцию сложения или вычитания.

Если после вычитания порядков код знака на сумматоре порядков будет 1 (разность отрицательна), т. е. $p < q$, это указывает на то, что порядок первого числа меньше порядка второго и, следовательно, первое число надо сдвинуть вправо на количество разрядов, равное этой разности, т. е. на $q - p$.

Сдвигаем первое число вправо и при каждом сдвиге на один разряд прибавляем по единице к сумматору порядка до тех пор, пока на сумматоре порядка не получится нуль (0.00000), т. е. первое число сдвигается на $q - p$ разрядов вправо. После этого сдвига порядок первого числа будет $p + (q - p) = q$. Таким образом, порядки обоих чисел будут выравнены и с ними можно производить операцию сложения или вычитания.

При сравнении порядков обоих чисел может получиться, что разность порядков превысит количество разрядов, предназначенное для учета порядков на машине. При этом высший разряд кода займет ячейку знака (*Зн1СмП*) и, следовательно, исказит знак порядка. Для обеспечения индикации правильного знака на сумматоре порядков следует предусмотреть две ячейки знака. Вторая ячейка знака *Зн2СмП* дает всегда правильное значение знака порядка.

Следует отметить, что после выравнивания порядков оба числа будут иметь порядок большего числа, т. е. в случае кода положительного знака на сумматоре порядков после вычитания порядков это будет порядок первого числа (p); в случае же кода отрицательного знака на *СмП* это будет порядок второго числа (q). Непосредственно после производства операции сложения или вычитания порядок полученного результата не изменится и будет равен порядку наибольшего числа.

Однако результат, полученный на сумматоре чисел (*СмЧ*), может оказаться ненормализованным. При сложении двух чисел одинаковых знаков возможно получение на *СмЧ* числа, большего единицы (в этом случае первая значащая цифра займет разряд знака *Зн1СмЧ*). Чтобы не потерять правильное значение знака, необходимо предусмотреть две ячейки для кода знака числа (максимально возможное увеличение количества разрядов при сложении двух чисел составляет один разряд).

Таким образом, вторая ячейка знака ($3n2Cm4$) всегда сохранит правильный код знака. Если оба числа были положительны (код знака 0), то при увеличении количества разрядов суммы на один разряд в двух ячейках знака сумматора чисел получится код 01. Для нормализации результат необходимо сдвинуть вправо на один разряд и одновременно увеличить на единицу порядок суммы (т. е. прибавить единицу к порядку наибольшего числа). Если оба числа были отрицательными (код знака 1), то при увеличении количества разрядов суммы на один разряд в двух ячейках знака основного сумматора получится код 10. Для нормализации результат необходимо сдвинуть вправо на один разряд и одновременно увеличить на единицу порядок суммы.

Таким образом, если при операции сложения или вычитания в двух ячейках знака $Cm4$ образуются разные коды (01 или 10), это указывает на необходимость сдвига результата вправо на один разряд и прибавления единицы к порядку суммы.

При выравнивании порядков правые разряды сдвигаемого числа пропадают и не участвуют в операциях сложения и вычитания.

Таким образом образуется систематическая ошибка. Для устранения систематической ошибки целесообразно ввести округление сдвигаемого числа. С этой целью в арифметическом устройстве предусмотрен дополнительный разряд. Округление чисел возможно производить лишь на сумматоре, так как приемные регистры ($B314$ и $B324$) не имеют счетных цепей.

При операциях сложения и вычитания первое число поступает на сумматор прямым кодом, если оно положительно, и дополнительным кодом, если оно отрицательно. Таким образом, если при выравнивании порядков приходится сдвигать первое число, то ошибка от отбрасывания последних разрядов всегда положительна, т. е. результат получается с недостатком. В этом случае необходимо произвести округление путем прибавления единицы к дополнительному разряду сумматора. Второе число поступает на второй блок запоминания ($B324$) всегда прямым кодом. Поэтому при сдвиге вто-

рого числа знак ошибки при отбрасывании последних разрядов совпадает со знаком числа.

Если сдвигаемое второе число положительно, то результат получается с недостачей и необходимо произвести округление путем прибавления единицы к дополнительному разряду сумматора. Если же сдвигаемое второе число отрицательно, то результат получается с избытком и необходимо отбросить цифру в дополнительном разряде сумматора, т. е. не производить округления.

Таким образом, при выравнивании порядков округление результата путем прибавления единицы к дополнительному разряду сумматора следует производить всегда, за исключением случая, когда сдвигается второе число (на *Б324*) и это число отрицательно.

Помимо систематической ошибки, возникающей при выравнивании порядков, возможно также появление ошибки в тех случаях, когда при сложении или вычитании результат выходит из числа располагаемых разрядов и для нормализации его необходимо сдвинуть вправо на один разряд. При этом последний разряд результата теряется. Ошибка от отбрасывания последнего разряда результата всегда положительна, так как положительный результат на сумматоре представляется прямым кодом, а отрицательный — дополнительным кодом. Таким образом, в этом случае будет иметь место систематическая ошибка. Для устранения систематической ошибки, в случае необходимости сдвига результата вправо, требуется произвести округление путем прибавления единицы к дополнительному разряду сумматора. Нетрудно убедиться, что в этих случаях округление, вызываемое выравниванием порядков, производить не следует, так как двойное округление приведет к систематической ошибке.

На основании вышеизложенного можно установить следующие правила округления результата при операциях сложения и вычитания.

Если результат сложения или вычитания двух чисел выходит из количества располагаемых разрядов на сумматоре, то полученный результат необходимо сдвинуть вправо на один разряд и произвести округление путем

прибавления единицы к дополнительному разряду сумматора. Если же сдвига результата вправо не требуется, то округление результата производится во всех случаях, за исключением того, когда при выравнивании порядков сдвигалось вправо второе число (на *БЗ2С*) и знак этого числа был отрицательный.

При округлении возможен, хотя и редкий, случай, когда после округления результат выйдет из числа располагаемых разрядов (когда все разряды, включая дополнительный, до округления имели коды 1). Поэтому необходимо предусмотреть проверку выхода из располагаемых разрядов (коды 01 или 10 на разрядах знака *СмС*) и, в случае необходимости, повторить нормализацию результата (сдвиг вправо на один разряд и увеличение порядка суммы на единицу).

Если результат сложения или вычитания получается отрицательный, т. е. в дополнительном коде, то производится преобразование его в прямой код. Для этой цели результат с сумматора передается в первый блок запоминания (*БЗ1С*) и затем вновь поступает на сумматор дополнительным кодом (дополнительный код от дополнительного дает прямой код). В отличие от первоначальной подачи отрицательных чисел на сумматор передача дополнительным кодом производится без знака. Знак результата остается в знаковой ячейке первого блока запоминания (*ЗнБЗ1С*). Это вызывается тем, что при сложении на сумматоре может получиться в дополнительном коде результат 11.000...0, что означает число -1 . При переводе этого результата в прямой код получим ненормализованное число 01.000...0. Если же для перевода результата в прямой код передачу на сумматор дополнительным кодом осуществить совместно со знаком, то результат будет неправильный [11.1111...1 (обратный код) $+ 1 = 00.0000...0$].

В рассмотренном случае после преобразования отрицательного результата из дополнительного в прямой код необходимо вновь проверить выход результата из располагаемых разрядов (коды 01 и 10 ячеек знака сумматора чисел), произвести сдвиг числа вправо на один разряд и увеличение порядка результата на единицу. (В остальных случаях при преобразовании допол-

нительного кода в прямой нормализации результата вправо не требуется.)

При сложении двух чисел разных знаков возможно уменьшение количества значащих разрядов. Это уменьшение количества значащих разрядов может быть произвольным, вплоть до полного пропадания значащих разрядов при одинаковых числах. Как указывалось выше, полученный результат необходимо нормализовать, т. е. сдвинуть число на сумматоре чисел на столько разрядов влево, чтобы первая значащая цифра заняла старший разряд. При каждом сдвиге влево на один разряд необходимо уменьшать на единицу порядок суммы (т. е. вычитать единицу из порядка наибольшего числа).

Нормализация результата влево производится всегда в прямом коде, т. е. при отрицательном результате после перевода его в прямой код.

При производстве операций сложения и вычитания возможны особые случаи, требующие специальных блокировок. В первую очередь это относится к случаям, когда одно из чисел равно нулю.

Если одно из чисел равно нулю (т. е. его порядок имеет код 1.00000), то нет смысла производить операцию сложения или вычитания, а следует просто передать на запоминание другое число с соответствующим знаком.

При сравнении порядков обоих чисел может получиться, что разность порядков превышает количество разрядов машины, и, следовательно, при выравнивании порядков одно из чисел выйдет из числа располагаемых разрядов и не окажет влияния на результат вычислений. Таким образом, производить выравнивание порядков в этом случае нецелесообразно, а следует непосредственно передать большее число с соответствующим знаком на запоминание.

В итоге операции сложения или вычитания результат может превысить максимально возможное число, которое может быть представлено на машине. Это может обнаружиться при нормализации результата (сдвиг числа вправо и прибавление единицы к его порядку), когда порядок числа превысит число

располагаемых разрядов. В этом случае в двух ячейках знака сумматора порядков будет код 01. Как указывалось выше, в этом случае должен быть произведен автоматический останов машины.

При вычитании чисел результат может получиться меньше минимально возможного числа на машине. Это может обнаружиться при нормализации результата (сдвиг числа влево и вычитание единицы из его порядка при каждом сдвиге на один разряд).

В случае получения минимально возможного кода порядка (код 1.00000) дальнейшая нормализация числа должна быть прекращена и на запоминание передается «код 0» (код порядка 1.00000).

Ниже приводятся примеры, поясняющие операцию сложения *).

Пример 1. Случай с нормализацией результата вправо.

Знак порядка	Порядок	Знак числа	Число	
0.	011	0.	11100100	Первое число
0.	001	0.	10010101	Второе число

Определяем разность порядков:

$$\begin{array}{rcl}
 00.011 & \text{Прямой код порядка первого числа (прибавление к сумматору порядков)} & \\
 + & & \\
 11.111 & \text{Дополнительный код порядка второго числа (вычитание из сумматора порядков)} & \\
 \hline
 00.010 & &
 \end{array}$$

Так как код знака разности порядков получился положительным, то это означает, что порядок первого числа больше порядка второго числа и, следовательно, перед операцией сложения необходимо произвести сдвиг второго числа вправо на количество разрядов, равное 0.010 (т. е. на два разряда). При каждом сдвиге

*) Примеры, поясняющие принципиальное осуществление операций, даны с сокращенным количеством разрядов чисел и их порядков. В дальнейшем везде принято: количество разрядов чисел — 8 и соответственно порядков — 3. Это соотношение количества разрядов числа и порядка эквивалентно соотношению 32 разрядов числа и 5 разрядов порядка.

вправо на один разряд следует вычитать единицу из разности порядков, а именно:

		Дополнительный разряд	
+ 00.010	00.10010101	0	Исходное число
+ 11.111	00.01001010	1	Первый сдвиг
+ 00.001			
+ 11.111	00.00100101	0	Второй сдвиг
00.000	Получение нуля порядка и прекращение выравнивания порядков.		

После выравнивания порядков производится сложение чисел на сумматоре чисел, а на сумматор порядков передается код порядка большего числа.

Порядок	00.11100100	0	Прямой код первого
большого	+		числа
числа			Сдвинутый прямой
	00.00100101	0	код второго числа
00.011	01.00001001	0	Результат.

Так как в ячейках знака получился код 01, то это означает, что следует произвести нормализацию результата путем сдвига числа вправо на один разряд и прибавления единицы к его порядку:

+ 00.011	01.00001001	0	Исходное число
+ 1	00.10000100	1	Сдвиг вправо
00.100			

В данном случае производится округление результата:

+ 00.10000100	1
	1
00.10000101	0

Передача на запоминание:

0.100 0.10000101

Пример 2. Случай с нормализацией результата влево.

1.110	1.11101101	Первое число
1.111	0.10010101	Второе число

*) При сдвиге дополнительного кода в сумматоре вправо содержимое первого знакового разряда сумматора переходит в его старший разряд.

Результат сложения получился положительный и выражен прямым кодом. Ячейка высшего разряда сумматора имеет код 0, и поэтому необходимо произвести нормализацию числа путем сдвига влево. При каждом сдвиге следует вычитать единицу из порядка.

+ 11.111	00.00011111	0	Исходное число
+ 11.111	00.00111110	0	Первый сдвиг
+ 11.110			
+ 11.111	00.01111100	0	Второй сдвиг
+ 11.101			
+ 11.111	00.11111000	0	Третий сдвиг
11.100	Прекращение нормализации, так как ячейка высшего разряда сумматора имеет код 1.		

Передача на запоминание:

1.100 0.11111000

Пример 3. Случай получения «кода нуля» при нормализации результата влево.

1.001	1.11101101	Первое число
1.010	0.10010101	Второе число

Определяем разность порядков:

+ 11.001	Прямой код порядка первого числа
+ 00.110	Дополнительный код порядка второго числа
11.111	

Производим сдвиг дополнительного кода первого числа вправо на количество разрядов, равное 0.001 (т. е. на один разряд), с одновременным прибавлением единицы к разности порядков при каждом сдвиге на один разряд:

+ 11.111	11.00010011	0	Дополнительный код первого числа
+ 1	11.10001001	1	Первый сдвиг
00.000	Получение нуля порядка и прекращение выравнивания порядков.		

После выравнивания порядков производим суммирование:

11.10001001	1	Дополнительный код первого числа
00.10010101	0	Прямой код второго числа
<u>00.00011110</u>	1	

Производим округление результата:

11.010	+	00.00011110	1
			1
<u>11.010</u>		<u>00.00011111</u>	0

Ячейка высшего разряда сумматора имеет код 0 и, следовательно, необходимо произвести нормализацию числа путем сдвига влево.

При каждом сдвиге следует вычитать единицу из порядка:

+	11.010	00.00011111	0	Исходное число
	11.111	00.00111110	0	Первый сдвиг
	<u>11.001</u>			
+	11.111	00.01111100	0	Второй сдвиг
	<u>11.000</u>			Блокировка дальнейшей нормализации, так как получился порядок «кода нуля».

Передача на запоминание «кода нуля»:

1.000	0.00000000
-------	------------

Пример 4. Случай получения результата большего, чем максимально возможное число на машине.

0.111	0.11100010	Первое число
0.101	0.10011011	Второе число

Определяем разность порядков:

+	00.111	Прямой код порядка первого числа
	11.011	Дополнительный код порядка второго числа
	<u>00.010</u>	

Производим сдвиг второго числа вправо на количество разрядов 0.010 (т. е. на два разряда):

+ 00.010	00.10011011	0	Исходное число
+ 11.111	00.01001101	1	Первый сдвиг
00.001			
+ 11.111	00.00100110	1	Второй сдвиг
00.000			Код нуля порядка и прекращение выравнивания порядков.

Производим суммирование:

	+ 00.11100010	0	Прямой код первого числа
	+ 00.00100110	1	Прямой код сдвинутого второго числа
00.111	01.00001000	1	

Производим нормализацию путем сдвига числа вправо на один разряд с одновременным прибавлением единицы к порядку числа:

+ 00.111	01.00001000	1	Исходное число
+ 1	00.10000100	0	Сдвиг вправо
01.000			

Величина порядка вышла из числа располагаемых разрядов и, следовательно, полученное число больше максимально возможного числа, которое может быть представлено на машине. Производится автоматический останов машины.

Пример 5. Случай выхода второго числа из числа располагаемых разрядов при выравнивании порядков.

0.110	0.10110011	Первое число
1.011	0.11011010	Второе число

Определяем разность порядков:

+ 00.110	Прямой код порядка первого числа
+ 00.101	Дополнительный код порядка второго числа
01.011	

Так как код знака разности порядков получился положительным (вторая ячейка знака порядков), то порядок

первого числа больше порядка второго числа, поэтому второе число необходимо сдвинуть вправо на количество разрядов 1011 (т. е. на 11 разрядов). Однако всего имеется 8 разрядов и при сдвиге второго числа на 11 разрядов вправо оно выйдет из располагаемого количества разрядов и не скажется на результате. Таким образом, производить выравнивание порядков и операцию сложения нецелесообразно и следует просто передать первое число на запоминание.

Передача на запоминание:

0.110 0.10110011

Пример 6. Случай получения кода 11.00...0 с превращением его в прямой код.

0.110	1.10110111	Первое число
0.101	1.10010010	Второе число

Определяем разность порядков:

$\begin{array}{r} + 00.110 \\ 11.011 \\ \hline 00.001 \end{array}$	Прямой код порядка первого числа Дополнительный код порядка второго числа
--	--

Производим сдвиг второго числа вправо на количество разрядов, равное 0.001 (т. е. на один разряд), с одновременным вычитанием единицы из разности порядков при каждом сдвиге на один разряд:

$\begin{array}{r} + 00.001 \\ 11.111 \\ \hline 00.000 \end{array}$	1.10010010 0 Исходное число 1.01001001 0 Первый сдвиг
	Получение нуля порядка и прекращение выравнивания порядков.

Производим суммирование:

$\begin{array}{r} 11.01001001 \quad 0 \\ + 11.10110111 \quad 0 \\ \hline 00.110 \quad 11.00000000 \quad 0 \end{array}$	Дополнительный код первого числа Дополнительный код сдвинутого второго числа
--	---

Результат сложения получился отрицательный и выраженный дополнительным кодом.

Код числа с сумматора передается на блок запоминания и потом снова на сумматор дополнительным кодом (без знака).

00.110 01.00000000 0

Производим нормализацию путем сдвига вправо на один разряд с одновременным прибавлением единицы к порядку:

+	00.110	01.00000000	0	Исходное число
	00.001	00.10000000	0	Сдвиг вправо
	00.111			

Передача на запоминание:

00.111 1.10000000

Осуществление операции «Сложение» (операция 1)

В соответствии с принятой методикой операция сложения разбивается на следующие этапы:

1) прием чисел из запоминающего устройства на арифметическое устройство;

2) выравнивание порядков;

3) сложение чисел;

4) округление, преобразование в прямой код и нормализация полученного результата;

5) передача результата в запоминающее устройство.

1. Перед приемом чисел на арифметическое устройство необходимо погасить имеющиеся в нем коды, оставшиеся от выполнения предыдущей операции. Иными словами, необходимо установить в положение «0» триггерные ячейки блоков запоминания чисел и порядков (БЗ1Ч, БЗ2Ч, БЗ1П и БЗ2П) и триггерные ячейки сумматора числа и сумматора порядков (СмЧ и СмП). Для этой цели должна быть предусмотрена цепь, по которой подается импульс, устанавливающий все триггерные ячейки арифметического устройства в положение «0» — цепи $У«0»АУЧ$ и $У«0»АУП$ (см. приложения 2 и 3).

Оба слагаемых должны быть приняты на блоки арифметического устройства. На *СмЧ* принимается первое число, на *БЗ2Ч* — второе число.

Для того чтобы обеспечить подачу первого числа на сумматор прямым или дополнительным кодом, это число необходимо предварительно принять на первый блок запоминания (*БЗ1Ч*). Для этой цели служит цепь *ПЧБЗ1Ч*. Наличие управляющего потенциала на этой цепи открывает приемные вентили первого блока запоминания (*БЗ1Ч*), и код числа от кодовых шин (*КШЧ*) поступает в *БЗ1Ч*. Для приема знака первого числа должен быть подан управляющий потенциал на цепь *ПЗнБЗ1Ч* (разделение цепей для приема числа и приема знака требуется для операций умножения и деления). Помимо цифровой части числа, необходимо также принять и его порядок. Порядок первого числа принимается на блок *БЗ1П*. Для этой цели подается управляющий потенциал на цепь *ППБЗ1П*, открывающий приемные вентили первого блока запоминания порядка (*БЗ1П*), и код порядка от кодовых шин (*КШП*) поступает в *БЗ1П* (непосредственная подача кода порядка на сумматор порядков недопустима, так как после выравнивания порядков обоих слагаемых код на *СмП* изменяется, а для получения порядка суммы необходимо иметь порядок большего числа).

После приема первого числа и его порядка на блоки *БЗ1Ч* и *БЗ1П* необходимо их передать на *СмЧ* и *СмП*. Передача числа на *СмЧ* должна производиться прямым кодом, если знак числа положителен, или дополнительным кодом, если знак отрицателен. Таким образом выбор вида передачи определяется кодом, находящимся в ячейке знака (*ЗнБЗ1Ч*), и в схему управления должны быть выведены потенциалы от этой триггерной ячейки (цепи «1» *ЗнБЗ1Ч* и «0» *ЗнБЗ1Ч*). При передаче числа с *БЗ1Ч* на *СмЧ* прямым кодом необходимо подать управляющий импульс на вентили, соединяющие *БЗ1Ч* и *СмЧ*, т. е. импульс на цепь *+ЧБЗ1Ч*. При передаче дополнительным кодом управляющий импульс подается по цепи — *ЧБЗ1Ч* на другие соединительные вентили, дающие передачу числа с *БЗ1Ч* на *СмЧ* обратным кодом. Этот же импульс через линию задержки поступает

на дополнительный разряд сумматора (*ДРСмЧ*) и переводит обратный код в дополнительный. Одновременно с подачей управляющего импульса по цепи — *ЧБЗ1Ч* необходимо подать импульс на входы знаковых ячеек сумматора (*Зн1СмЧ* и *Зн2СмЧ*), что осуществляется по цепи «+1»*ЗнСмЧ* (разделение цепей — *ЧБЗ1Ч* и «+1»*ЗнСмЧ* вызвано требованием обеспечить передачу дополнительным кодом без знака при преобразовании отрицательного результата в прямой код).

Передача порядка с *БЗ1П* на *СмП* производится всегда прямым кодом, так как отрицательные порядки хранятся в запоминающем устройстве в виде дополнительного кода. Эта передача осуществляется подачей управляющего импульса по цепи +*ПБЗ1П* на вентили, соединяющие блок *БЗ1П* и *СмП*.

Второе число принимается на блок *БЗ2Ч*. Знак второго числа принимается на знаковую ячейку блока *БЗ1Ч*, которая предварительно должна быть погашена. Одновременно с гашением кода в ячейке *ЗнБЗ1Ч* гасится также код в блоке *БЗ1Ч*, так как в дальнейшем этот блок потребуется для приема кода. Гашение кода в ячейке *ЗнБЗ1Ч* и в блоке *БЗ1Ч* осуществляется подачей импульсов в цепи *У«0»ЗнБЗ1Ч* и *У«0»БЗ1Ч*. Прием кода второго числа на *БЗ2Ч*, его знака на ячейку *ЗнБЗ1Ч* и его порядка на *БЗ2П* осуществляется подачей управляющих потенциалов на цепи *ПЧБЗ2Ч*, *ПЗнБЗ1Ч* и *ППБЗ2П*. Эти потенциалы открывают вентили, через которые импульсы кода от кодовых шин поступают на соответствующие блоки.

Таким образом, после приема чисел код первого числа находится на *СмЧ* вместе со знаком, а его порядок — на *СмП*. Код второго числа находится на *БЗ2Ч*, его знак — в ячейке *ЗнБЗ1Ч*, а его порядок — на *БЗ2П*.

2. Для выравнивания порядков обоих чисел необходимо определить разность порядков, что производится вычитанием порядка второго числа из порядка первого числа. Для этой цели необходимо передать порядок второго числа с *БЗ2П* на *СмП* дополнительным кодом, что осуществляется подачей импульса в цепь — *ПБЗ2П*. Этот импульс поступает на вентили, соединяющие блоки *БЗ2П* и *СмП*, и число с *БЗ2П* передается на *СмП* обратным

кодом. Для преобразования обратного кода в дополнительный этот же импульс через линию задержки подается на вход первого разряда *СмП*.

Если разность порядков положительна (порядок первого числа больше, чем порядок второго числа), то для выравнивания порядков необходимо сдвигать вправо второе число (на *БЗ2Ч*). Если же разность порядков отрицательна, то необходимо сдвигать вправо первое число (на *СмЧ*). Для того чтобы определить, какое число необходимо сдвигать, в схему управления должны быть поданы потенциалы от знаковой ячейки сумматора порядков (цепи «1»*Зн2СмП* и «0»*Зн2СмП*).

Сдвиг кодов на *БЗ2Ч* и *СмЧ* производится путем подачи импульсов сдвига в цепи *ПрБЗ2Ч* и *ПрСмЧ* соответственно.

При каждом сдвиге к коду на *СмП* необходимо прибавлять единицу, если разность порядков отрицательна. Если разность порядков положительна, необходимо произвести сначала обращение кода разности. Это производится с целью ускорения операции сдвига при выравнивании порядков, так как прибавление единицы («+1»*СмП*) происходит непосредственно в цепь сквозного переноса сумматора порядков и тем самым ускоряет процесс сложения. Для определения необходимости обращения кода в схему управления должны быть поданы потенциалы «0»*Зн2СмП* и «0»*СмП*. Импульс *ОбрСмП* (см. приложение 3) выдает на *СмП* одновременно сигналы *+ПБЗ2П* и *-ПБЗ2П*, но блокирует выдачу стробирующего сигнала. Сквозного переноса на *СмП* в этом случае не происходит, а на сумматоре получаем обращенный кол разности порядков.

Сдвиг кодов следует производить до тех пор, пока на *СмП* не образуется код 00.00000. Для того чтобы определить момент окончания выравнивания порядков, в схему управления подается сигнал сквозного переноса со второй ячейки знака *СмП*, сигнал *ПерЗн2СмП*.

3. После выравнивания порядков соответственно сдвинутые коды находятся на *БЗ2Ч* и *СмЧ* и можно приступать к их суммированию. Предварительно необходимо код числа с *БЗ2Ч* передать на *БЗ1Ч*, так как *БЗ2Ч* не имеет непосредственной связи с *СмЧ*. Для этой цели по-

дается управляющий потенциал *ПЧБЗ1Ч*, открывающий вентили, соединяющие *БЗ1Ч* с кодовыми шинами, а в цепь *ЗЧБЗ2Ч* подается импульс, который выдает код с *БЗ2Ч* на кодовые шины.

Передача второго числа с *БЗ1Ч* на *СмЧ* производится точно так же, как и первого числа, т. е. подаются импульсы на цепь $+ЧБЗ1Ч$, если в ячейке *ЗнБЗ1Ч* имеется код 0, или на цепи $-ЧБЗ1Ч$ и $+1»ЗнСмЧ$, если в ячейке *ЗнБЗ1Ч* имеется код 1. Порядок суммы равен порядку наибольшего числа. Этот порядок необходимо передать на *СмП*. Гашения *СмП* не требуется, так как после окончания выравнивания порядков на *СмП* всегда находится код 00.00000 («0»*СмП*). Порядки обоих слагаемых находятся на *БЗ1П* и *БЗ2П*. Выбор порядка, который следует подать на *СмП*, определяется тем, какое из чисел сдвигалось при выравнивании порядков. Если сдвигалось второе число (на *БЗ2Ч*), то следует передать порядок первого числа (с *БЗ1П*), если же сдвигалось первое число (на *СмЧ*), то следует передать порядок второго числа (с *БЗ2П*). Передача порядка суммы на *СмП* осуществляется подачей импульсов в цепь $+ПБЗ1П$, или в цепь $+ПБЗ2П$. Выбор цепи, в которую следует подать импульс, определяется устройством управления, в котором предусмотрена специальная триггерная ячейка *ТП* (триггер порядков). Код 0 в этой ячейке («0»*ТП*) указывает, что сдвигалось второе число и следует подать импульс в цепь $+ПБЗ1П$. Код 1 в этой ячейке («1»*ТП*) указывает, что сдвигалось первое число и следует подать импульс в цепь $+ПБЗ2П$.

4. Теперь необходимо проверить выход суммы из числа располагаемых разрядов. Выход суммы из числа располагаемых разрядов характеризуется разными кодами в знаковых ячейках сумматора, т. е. кодами 01 или 10. При возникновении в знаковых ячейках таких кодов в устройство управления должен быть подан управляющий потенциал, указывающий на необходимость произвести нормализацию результата вправо. Для этой цели служит выходная цепь «01—10»*ЗнСмЧ*. При необходимости произвести нормализацию результата вправо требуется сдвинуть число на *СмЧ* вправо на один разряд и прибавить единицу на *СмП*. Это

осуществляется подачей импульса в цепи *ПрСмЧ* и «+1»*СмП*. Одновременно подается импульс, гасящий ячейку *ЗнБЗІЧ* (импульс в цепи *У«0»ЗнБЗІЧ*). Как будет видно ниже, это необходимо для того, чтобы деблокировать схему округления результата (после нормализации результата вправо округление производится во всех случаях).

После проверки и, в случае необходимости, осуществления нормализации результата вправо производится округление, для чего прибавляется единица к дополнительному разряду *СмЧ*. Для осуществления этого подается импульс в цепь «+1»*ДРСмЧ*.

Округление результата не должно производиться лишь в том случае, если не было нормализации результата вправо и при выравнивании порядков сдвигалось вправо второе число (на *БЗ2Ч*), причем знак этого числа был отрицателен. Разрешение произвести округление определяется наличием управляющего потенциала в цепи «0»*ЗнБЗІЧ* или в цепи «1»*ТП* (управляющий потенциал в цепи «1»*ТП* указывает, что при выравнивании порядков сдвигалось вправо первое число на *СмЧ*). В случае нормализации результата вправо знаковая ячейка (*ЗнБЗІЧ*), как указывалось выше, всегда гасится, и следовательно, в этих случаях всегда будет производиться округление.

После округления возможен выход результата из числа располагаемых разрядов, и в этом случае требуется произвести нормализацию вправо. Это осуществляется точно так же, как это было описано ранее (необходимость произвести нормализацию вправо определяется наличием управляющего потенциала в цепи «01—10»*ЗнСмЧ*; сама нормализация осуществляется подачей импульса в цепи *ПрСмЧ* и «+1»*СмП*).

Если результат получился отрицательный и, следовательно, представлен на *СмЧ* в виде дополнительного кода, то необходимо преобразовать его в прямой код. Для этой цели результат с *СмЧ* должен быть передан на *БЗІЧ* и *ЗнБЗІЧ*, а затем подан на *СмЧ* дополнительным кодом (без знака). Дополнительный код от дополнительного дает прямой код. Для стандартности операций целесообразно осуществить такую передачу вне зависимости

от знака результата. В этом случае передача с *БЗ1Ч* на *СмЧ* осуществляется прямым кодом, если число положительно, и дополнительным кодом (без знака), если число отрицательно.

Перед передачей результата с *СмЧ* на *БЗ1Ч* и *ЗнБЗ1Ч* необходимо предварительно погасить ячейки *ЗнБЗ1Ч* и *БЗ1Ч*, что осуществляется подачей импульса в цепи *У«0»ЗнБЗ1Ч* и *У«0»БЗ1Ч*. Сама передача осуществляется подачей управляющего потенциала в цепи *ПЗнБЗ1Ч* и *ПЧБЗ1Ч*. Этот потенциал открывает вентили, соединяющие входные цепи ячеек *ЗнБЗ1Ч* и *БЗ1Ч* с кодовыми шинами (*КШЧ*). Подача кода с *СмЧ* на *КШЧ* производится импульсом в цепях *ЗЧСмЧ* и *ЗЗнСмЧ*. Таким образом, код с *СмЧ* будет передан на *БЗ1Ч* и *ЗнБЗ1Ч*.

Передав код с *СмЧ* на *БЗ1Ч*, необходимо погасить *СмЧ*, так как затем на него будет принят код с *БЗ1Ч*. Для этой цели подается импульс в цепь *У«0»СмЧ*. Если затем в ячейке *ЗнБЗ1Ч* имеется код 0 (управляющий потенциал на цепи «0»*ЗнБЗ1Ч*), то это указывает, что результат положительный и его следует передать на *СмЧ* прямым кодом. Если же в ячейке *ЗнБЗ1Ч* имеется код 1 (управляющий потенциал на цепи «1»*ЗнБЗ1Ч*), то передачу следует осуществить дополнительным кодом. В отличие от подачи числа дополнительным кодом при сложении чисел в этом случае передается дополнительным кодом лишь само число без знака. Таким образом, для осуществления передачи результата с *БЗ1Ч* на *СмЧ* необходимо: подать импульс в цепь *+ЧБЗ1Ч*, если имеется управляющий потенциал на цепи «0»*ЗнБЗ1Ч*, или в цепь *-ЧБЗ1Ч*, если управляющий потенциал имеется на цепи «1»*ЗнБЗ1Ч* (импульс «+1»*ЗнСмЧ* не подается).

После передачи результата на *СмЧ* необходимо проверить выход числа из располагаемых разрядов (управляющее напряжение на цепи «01—10»*ЗнСмЧ*) и в случае необходимости нормализовать результат вправо (подать импульс в цепи *ПрСмЧ* и «+1»*СмП*).

После преобразования кода на *СмЧ* в прямой код проверяется необходимость нормализации кода на сумматоре.

Если в результате сложения чисел разных знаков в сумме уничтожились старшие разряды, то необходимо произвести нормализацию суммы влево. Так как после преобразования кода результат на *СмЧ* всегда представлен прямым кодом, то критерием необходимости нормализации влево служит наличие кода 0 в старшем разряде *СмЧ*. Для этой цели с 32-й ячейки выведена цепь «0»32*СмЧ*. Наличие управляющего потенциала на этой цепи указывает на необходимость нормализации результата влево. Сама нормализация влево производится путем подачи импульсов сдвига в цепь *ЛСмЧ*. При каждом сдвиге необходимо вычитать единицу из кода порядка, но с целью ускорения операции (так же как и при выравнивании порядков) происходит предварительное обращение кода на *СмП* и производится прибавление единицы путем подачи импульса в цепь «+1»*СмП*. Нормализация влево продолжается до тех пор, пока в старшем разряде *СмЧ* не получится код 1, т. е. пока не пропадет управляющий потенциал на цепи «0»32*СмЧ*. Нормализация результата влево может быть заблокирована, если это предусмотрено в коде команды.

После нормализации результат находится на *СмЧ* в виде прямого кода, знак его хранится в ячейке *ЗнБЗ1Ч*, а порядок — в *СмП*, код которого обращается вторично и принимает правильное значение.

5. Для передачи результата на запоминание следует подать импульсы в цепи *ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*.

При осуществлении операции сложения предусмотрены блокировки:

1) если одно из чисел равно нулю (код порядка 1.00000), то операция сложения не производится и в запоминающее устройство передается непосредственно другое число;

2) если порядки чисел расходятся на величину большую, чем количество располагаемых разрядов на сумматоре, то сложение чисел не производится и в запоминающее устройство передается большее число;

3) если результат превышает наибольшее число, которое может быть представлено на машине, то происходит автоматический останов;

4) если результат меньше минимального числа, которое может быть представлено на машине, то в запоминающее устройство передается «код нуля» (код порядка 1.00000).

1. Если первое число равно нулю, то операция сложения должна быть заблокирована и в запоминающее устройство передан код второго числа. Показателем того, что число равно нулю, служит код порядка 1.00000. Для того чтобы иметь возможность определить, что первое число равно нулю, с ячеек *БЗ1П* выведена соответствующая цепь — код «0» *БЗ1П*. При коде 1.00000 на *БЗ1П* на этой цепи возникает управляющий потенциал, который блокирует выравнивание порядков, гасит код на *СмЧ* и *СмП* (импульс в цепях *У«0»СмЧ* и *У«0»СмП*), а также устанавливает триггер порядков в положение «1» (управляющий потенциал на цепи *У«1»ТП*). В соответствии с этим на *СмП* будет передан порядок второго числа, а на *СмЧ* поступит код второго числа.

Аналогично, если второе число равно нулю, то возникает управляющий потенциал на цепи код «0» *БЗ2П*, который блокирует выравнивание порядков, гасит код на *БЗ2Ч* и *СмП* (импульс в цепях *У«0»БЗ2Ч* и *У«0»СмП*), а также устанавливает триггер порядков в положение «0» (управляющий потенциал на цепи *У«0»ТП*). В соответствии с этим на *СмП* будет передан порядок первого числа, а на *СмЧ* останется код первого числа.

2. Если порядок второго числа больше порядка первого числа на величину, равную или превышающую количество располагаемых разрядов (32), то операция сложения должна быть заблокирована и в запоминающее устройство передан код второго числа. В этом случае при сравнении порядков (при вычитании порядка второго числа из порядка первого числа) на *СмП* образуется код 11.00000, или в его ячейках знака — код 10.

Для того чтобы иметь возможность определить возникновение подобных кодов, с ячеек *СмП* выведены соответствующие цепи: код «0» *СмП* и «10» *ЗнСмП*. При наличии управляющего потенциала на одной из этих цепей

блокируется выравнивание порядков, гасится код на *СмЧ* и *СмП* и триггер порядков устанавливается в положение «1», т. е. производятся те же действия, что и при возникновении управляющего потенциала на цепи код«0»БЗ1П. В результате на запоминание будет передан код второго числа.

Если порядок первого числа больше порядка второго числа на величину, равную или превышающую количество располагаемых разрядов на *СмЧ*, то операция сложения должна быть заблокирована и в запоминающее устройство передан код первого числа. В этом случае в ячейках знака *СмП* образуется код 01. Соответствующая выходная цепь («01»ЗнСмП) выведена с *СмП*. При возникновении управляющего потенциала на этой цепи необходимо произвести те же действия, что и при наличии управляющего потенциала на цепи код«0»БЗ2П (блокировка выравнивания порядков, гашение БЗ1Ч и *СмП* и установка «0»ТП).

3. Если при нормализации результата вправо порядок превысит максимально возможную величину (код 01 в ячейках знака *СмП*), то это указывает на выход результата из располагаемого числа разрядов и должен быть произведен останов машины. В этом случае образуется управляющий потенциал на цепи «01»ЗнСмП, который и обеспечивает останов машины. Действие этого потенциала должно осуществляться лишь после определения необходимости производства нормализации вправо, так как на этой цепи может получиться управляющее напряжение и при сравнении порядков.

4. Если при нормализации результата влево порядок получится равный минимально возможному порядку (11.00000), то процесс нормализации влево прекращается и на запоминание передается код порядка 1.00000 (код самого числа при этом не имеет значения). Для осуществления этого окончание процесса нормализации влево определяется не только исчезновением управляющего потенциала в цепи «0»З2СмЧ, но также и возникновением управляющего потенциала в цепи код«0»СмП.

Элементарные операции

1°. Установка на «0» всех элементов АУ ($У«0»АУ$).

2°. Прием первого числа и его знака на $БЗ1Ч$ и $ЗнБЗ1Ч$, а его порядка на $БЗ1П$ ($ПЧБЗ1Ч$, $ПЗнБЗ1Ч$ и $ППБЗ1П$).

3°. Передача числа прямым или дополнительным кодом (в зависимости от кода знака числа) с $БЗ1Ч$ на $СмЧ$ ($+ЧБЗ1Ч$, если «0» $ЗнБЗ1Ч$, или $-ЧБЗ1Ч$ и «+1» $ЗнСмЧ$, если «1» $ЗнБЗ1Ч$). Передача порядка прямым кодом с $БЗ1П$ на $СмП$ ($+ПБЗ1П$).

4°. Установка на «0» $ЗнБЗ1Ч$ и $БЗ1Ч$ ($У«0»ЗнБЗ1Ч$ и $У«0»БЗ1Ч$).

5°. Прием второго числа на $БЗ2Ч$ и его знака на $ЗнБЗ1Ч$, а его порядка на $БЗ2П$ ($ПЧБЗ2Ч$, $ПЗнБЗ1Ч$ и $ППБЗ2П$).

6°. Передача порядка дополнительным кодом с $БЗ2П$ на $СмП$ ($-ПБЗ2П$) для сравнения порядков.

7°. Если «0» $Зн2СмП$, то обращение кода на $СмП$ и сдвиги числа вправо на $БЗ2Ч$ ($ПрБЗ2Ч$) с одновременными прибавлениями единицы к $СмП$ («+1» $СмП$). Если «1» $Зн2СмП$, то сдвиги числа вправо на $СмЧ$ ($ПрСмЧ$) с одновременными прибавлениями единицы к $СмП$ («+1» $СмП$). Сдвиги продолжаютсЯ до тех пор, пока на $СмП$ не будет код 00.00000.

8°. Передача второго числа с $БЗ2Ч$ на $БЗ1Ч$ прямым кодом ($ПЧБЗ1Ч$ и $ЗЧБЗ2Ч$).

9°. Передача числа прямым или дополнительным кодом (в зависимости от кода на ячейке $ЗнБЗ1Ч$) с $БЗ1Ч$ на $СмЧ$ ($+ЧБЗ1Ч$, если «0» $ЗнБЗ1Ч$, или $-ЧБЗ1Ч$ и «+1» $ЗнСмЧ$, если «1» $ЗнБЗ1Ч$). Передача на $СмП$ прямым кодом порядка числа с $БЗ1П$, если ранее был код 0 в ячейке $Зн2СмП$, или с $БЗ2П$, если ранее был код 1 в ячейке $Зн2СмП$ ($+ПБЗ1П$, если «0» $ТП$, или $+ПБЗ2П$, если «1» $ТП$).

10°. Производится проверка выхода результата из разрядов (разные коды в ячейках знака $СмЧ$) и в случае необходимости результат сдвигается на один разряд вправо с соответствующим прибавлением единицы к $СмП$ и установкой на «0» $ЗнБЗ1Ч$ (если «01—10» $ЗнСмЧ$, то $ПрСмЧ$, «+1» $СмП$, $У«0»ЗнБЗ1Ч$).

11°. Если ячейка знака *БЗ1Ч* находится в положении «0» или *ТП* находится в положении «1», то производится округление результата путем прибавления единицы к дополнительному разряду *СмЧ* (если «0»*ЗнБЗ1Ч* или «1»*ТП*, то «+1»*ДРСмЧ*).

12°. Производится проверка выхода результата из разрядов (разные коды в ячейках знака *СмЧ*) и в случае необходимости результат сдвигается на один разряд вправо с соответствующим прибавлением единицы к *СмП* (если «01—10»*ЗнСмЧ*, то *ПрСмЧ* и «+1»*СмП*).

13°. Установка на «0» ячеек *БЗ1Ч* и *ЗнБЗ1Ч* (*У«0»БЗ1Ч* и *У«0»ЗнБЗ1Ч*).

14°. Код с *СмЧ* передается на *БЗ1Ч* и ячейку *ЗнБЗ1Ч* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч*, *ЗЧСмЧ* и *ЗЗнСмЧ*).

15°. Производится установка на «0» *СмЧ* (*У«0»СмЧ*).

16°. Передается число с *БЗ1Ч* на *СмЧ*. Если «1»*ЗнБЗ1Ч* — дополнительным кодом (без знака); если «0»*ЗнБЗ1Ч* — прямым кодом. (Если «0»*ЗнБЗ1Ч*, то + *ЧБЗ1Ч*, если «1»*ЗнБЗ1Ч*, то — *ЧБЗ1Ч*.)

17°. Производится проверка выхода результата из разрядов (код 01 в ячейках *ЗнСмЧ*) и в случае необходимости результат сдвигается вправо на один разряд с соответствующим прибавлением единицы к *СмЧ* (если «01—10»*ЗнСмЧ*, то *ПрСмЧ* и «+1»*СмП*).

18°. Если нормализация при составлении программы предусмотрена, то производится проверка наличия значащей единицы в старшем разряде и в случае необходимости производится нормализация числа путем его сдвига влево до тех пор, пока не удовлетворится указанное условие. Перед началом нормализации обращается код на *СмП* и при каждом сдвиге на один разряд производится прибавление единицы к порядку (если «0»*З2СмЧ*, то *ЛСмЧ* и «+1»*СмП*).

После окончания нормализации производится второе обращение кода порядка (*Обр СмП*).

19°. Передача числа прямым кодом с *СмЧ*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП* на запоминание (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

1°. При появлении кода «0» на БЗ1П (код 1.00000 после операции 2°) операция 7° блокируется, а вместо этого производится:

- а) установка на «0» СмЧ (У«0»СмЧ);
- б) установка на «0» СмП (У«0»СмП);
- в) установка на «1» ТП (У«1»ТП).

Остальные операции сохраняются.

Таким образом, на внутреннее запоминающее устройство будет передано второе число с его знаком и порядком.

2°. При появлении кода «0» на БЗ2П (код 1.00000 после операции 5°) операция 7° блокируется, а вместо этого производится:

- а) установка на «0» БЗ2Ч (У«0»БЗ2Ч);
- б) установка на «0» СмП (У«0»СмП);
- в) установка на «0» ТП (У«0»ТП).

Остальные операции сохраняются.

На запоминание будет передано первое число с его знаком и порядком.

3°. При появлении на СмП кода «0» (11.00000) или кода 10 на ячейках его знака при сравнении порядков (после операции 6°) производятся те же операции, что и в п. 1° блокировок, т. е. передается в оперативное запоминающее устройство второе число со своим знаком и порядком.

4°. При появлении на ячейках знака СмП кода 01 при сравнении порядков (после операции 6°) производятся те же операции, что и в п. 2° блокировок, т. е. на запоминание передается первое число со своим знаком и порядком.

5°. При появлении на ячейках знака СмП кода 01 при нормализации результата вправо (после операций 10°, 12°, а также 17°) происходит автоматический останов машины (аварийный останов).

6°. При появлении на СмП кода «0» в процессе нормализации результата влево (в процессе операции 18°) операция нормализации прекращается и на запоминание передается код «0».

Осуществление операции «Вычитание» (операция 2)

Для осуществления операции вычитания необходимо изменить знак у второго числа. В остальном же эта операция ничем не отличается от операции сложения. Изменение знака второго числа проще всего осуществить путем установки ячейки *ЗнБЗ1Ч* в положение «1» перед приемом второго числа. Для этой цели предусмотрена цепь $У«1»ЗнБЗ1Ч$. Так как прием кода знака производится на счетный вход, то при приеме кода знака 0 на ячейке *ЗнБЗ1Ч* останется код 1, при приеме же кода знака 1 на ячейке *ЗнБЗ1Ч* получится код 0.

Таким образом обеспечивается изменение знака второго числа.

В соответствии с этим все элементарные операции для выполнения вычитания, а также блокировки остаются теми же, что и для сложения, за исключением элементарной операции 4°, которая заменяется следующей:

Установка на «0» *БЗ1Ч* и установка на «1» *ЗнБЗ1Ч* ($У«0»БЗ1Ч$ и $У«1»ЗнБЗ1Ч$).

§ 2. Умножение

При умножении двух чисел цифровая часть произведения получается от перемножения прямых кодов цифровых частей этих чисел, а порядок произведения равен сумме кодов порядков обоих чисел. Таким образом, в результате умножения двух чисел $2^p \cdot A$ и $2^q \cdot B$ будем иметь:

$$2^p \cdot A \times 2^q \cdot B = 2^{p+q} (A \times B).$$

Знак произведения определяется как сумма кодов знака. Действительно, произведение двух положительных чисел дает положительное произведение — коды знаков $0 + 0 = 0$. Произведение двух отрицательных чисел дает положительное произведение — коды знаков $1 + 1 = 0$. Произведение положительного числа на отрицательное дает отрицательное произведение — коды знаков $0 + 1 = 1 + 0 = 1$.

Выполнение операции умножения производится следующим образом. Коды разрядов множителя последовательно поступают на управляющие цепи, передающие код множимого на сумматор. Всякий раз, когда разряд множителя, на который производится умножение, имеет код 1, множимое прибавляется на сумматор. После прохождения одного кода множителя величина на сумматоре сдвигается на один разряд. Такая последовательность действий повторяется до тех пор, пока не пройдут все разряды множителя.

Множитель можно подавать или старшими или младшими разрядами вперед. При подаче множителя старшими разрядами вперед число на сумматоре надо сдвигать влево. Сумматор при этом должен иметь двойное количество счетных ячеек, так как возможны переносы из младших разрядов в старшие. При подаче множителя младшими разрядами вперед число на сумматоре надо сдвигать вправо. Сумматор при этом может иметь количество счетных ячеек, равное числу разрядов.

Для иллюстрации процесса умножения рассмотрим пример:

$$\begin{aligned}\text{Множимое } A &= .101011 \\ \text{Множитель } B &= .110101.\end{aligned}$$

Код множи- теля	Сумматор	
1	$\begin{array}{r} + .000000 \\ + .101011 \\ \hline .101011 \end{array}$	Передача множимого на сумматор
0	$\begin{array}{r} .010101 \ 1 \\ .010101 \ 1 \end{array}$	Сдвиг вправо на 1 разряд Множимое на сумматор не передается
1	$\begin{array}{r} + .001010 \ 11 \\ + .101011 \\ \hline .110101 \ 11 \end{array}$	Сдвиг вправо на 1 разряд Передача множимого на сумматор
0	$\begin{array}{r} .011010 \ 111 \\ .011010 \ 111 \end{array}$	Сдвиг вправо на 1 разряд Множимое на сумматор не передается

1	$\begin{array}{r} + .001101\ 0111 \\ .101011 \\ \hline .111000\ 0111 \end{array}$	Сдвиг вправо на 1 разряд Передача множимого на сумматор
---	---	--

1	$\begin{array}{r} + .011100\ 00111 \\ .101011 \\ \hline 1.000111\ 00111 \\ 1.00011\ 100111 \end{array}$	Сдвиг вправо на 1 разряд Передача множимого на сумматор Сдвиг вправо на 1 разряд Результат
---	---	---

Как видно из рассмотренного примера, цифры, вышедшие вправо от основных разрядов, в дальнейшем суммировании не участвуют. Если требуется иметь удвоенное количество разрядов произведения, эти цифры могут быть запомнены в обычных ячейках статического запоминания.

При умножении двух нормализованных чисел, из которых каждое меньше 1 и больше или равно $0,1 \left(\frac{1}{2}\right)$, произведение будет всегда меньше 1 и больше или равно $0,01 \left(\frac{1}{4}\right)$. Поэтому нормализации произведения вправо не требуется; нормализация же произведения влево может потребоваться максимум на один разряд. При умножении ненормализованных чисел может потребоваться нормализация и на большее число разрядов.

Критерием необходимости нормализации произведения влево может явиться тот же критерий, что и при операции сложения, а именно, наличие кода 0 в ячейке высшего разряда сумматора. В случае необходимости нормализации произведения число на сумматоре сдвигается на один разряд влево и одновременно вычитается единица из порядка произведения. Сдвиг влево происходит до тех пор, пока в старшем разряде сумматора не появится код 1.

После нормализации произведения необходимо произвести его округление, для чего следует прибавить единицу к дополнительному разряду сумматора.

Сложение порядков сомножителей производится обычным способом на сумматоре порядков и в особых

пояснениях не нуждается. То же относится к сложению кодов знаков сомножителей.

При производстве операции умножения возможны особые случаи, требующие специальных блокировок.

Если код одного из сомножителей равен нулю (т. е. его порядок имеет код 1.00000), то умножение не производится и передается на запоминание код «0».

При сложении порядков обоих сомножителей на сумматоре порядков может получиться порядок произведения меньший, чем минимально возможный порядок на машине (т. е. код порядка 11.00000 или код 10 в ячейках знака сумматора порядков). В этом случае операция умножения не производится и на запоминание передается код «0» (код порядка 1.00000).

При сложении порядков обоих сомножителей на сумматоре порядков может получиться порядок произведения больший, чем максимально возможный порядок на машине (т. е. в ячейках знака сумматора порядков будет код 01). В этом случае машина автоматически останавливается.

При нормализации результата возможно образование на сумматоре порядка кода «0» (т. е. порядка 11.00000). В этом случае на запоминание должен быть передан код «0». Однако специальных действий для этого не требуется, так как и нормальная передача результата на запоминание даст правильный код числа (код порядка 1.00000, т. е. код «0»).

Ниже приводятся примеры, поясняющие операцию умножения.

Пример 1. Случай, не требующий нормализации результата.

0.011	0.11011011	Множимое
1.111	1.11001101	Множитель

Складываем отдельно коды порядков чисел и получаем код порядка произведения до нормализации:

$$\begin{array}{r}
 + 00.011 \\
 11.111 \\
 \hline
 00.010
 \end{array}$$

Складываем отдельно коды знаков чисел и получаем код знака произведения:

$$\begin{array}{r} +0 \\ 1 \\ \hline 1 \end{array}$$

Умножаем коды чисел:

Код множителя	Сумматор	
1	$\begin{array}{r} +00.00000000 \\ 00.11011011 \\ \hline 00.11011011 \end{array}$	
0	$\begin{array}{r} 00.01101101 \ 1 \\ 00.01101101 \ 1 \end{array}$	Сдвиг
1	$\begin{array}{r} +00.00110110 \ 11 \\ 00.11011011 \\ \hline 01.00010001 \ 11 \end{array}$	Сдвиг
1	$\begin{array}{r} +00.10001000 \ 111 \\ 00.11011011 \\ \hline 01.01100011 \ 111 \end{array}$	Сдвиг
	$\begin{array}{r} 00.10110001 \ 1111 \\ 00.10110001 \ 1111 \end{array}$	Сдвиг
0	$\begin{array}{r} 00.01011000 \ 11111 \\ 00.01011000 \ 11111 \end{array}$	Сдвиг
0	$\begin{array}{r} 00.01011000 \ 11111 \\ 00.01011000 \ 11111 \end{array}$	
1	$\begin{array}{r} +00.00101100 \ 011111 \\ 00.11011011 \\ \hline 01.00000111 \ 011111 \end{array}$	Сдвиг
1	$\begin{array}{r} +00.10000011 \ 1011111 \\ 00.11011011 \\ \hline 01.01011110 \ 1011111 \\ 00.10101111 \ 01011111 \end{array}$	Сдвиг
	$\begin{array}{r} +1 \\ \hline 00.10101111 \ 11011111 \end{array}$	Сдвиг. Нормализация не требуется. Округление

Передача на запоминание:

0.010 1.10101111

Пример 2. Случай, требующий нормализации результата.

1.110	1.10110101	Множимое
1.100	1.10101011	Множитель

Складываем отдельно коды порядков чисел и получаем код порядка произведения до его нормализации:

$$\begin{array}{r} + 11.110 \\ + 11.100 \\ \hline 11.010 \end{array}$$

Складываем отдельно коды знаков чисел и получаем код знака произведения:

$$\begin{array}{r} + 1 \\ + 1 \\ \hline 0 \end{array}$$

Умножаем коды чисел:

Код множителя	Сумматор	
1	$\begin{array}{r} + 00.00000000 \\ + 00.10110101 \\ \hline 00.10110101 \end{array}$	
1	$\begin{array}{r} + 00.01011010 \ 1 \\ + 00.10110101 \\ \hline 01.00001111 \ 1 \end{array}$	Сдвиг
0	$\begin{array}{r} 00.10000111 \ 11 \\ 00.10000111 \ 11 \end{array}$	Сдвиг
1	$\begin{array}{r} + 00.01000011 \ 111 \\ + 00.10110101 \\ \hline 00.11111000 \ 111 \end{array}$	Сдвиг
0	$\begin{array}{r} 00.01111100 \ 0111 \\ 00.01111100 \ 0111 \end{array}$	Сдвиг
1	$\begin{array}{r} + 00.00111110 \ 00111 \\ + 00.10110101 \\ \hline 00.11110011 \ 00111 \end{array}$	Сдвиг

0	00.01111001	100111	Сдвиг
	00.01111001	100111	
1	+ 00.00111100	1100111	Сдвиг
	+ 00.10110101		
	00.11110001	1100111	
	00.01111000	11100111	Сдвиг

Ячейка высшего разряда сумматора имеет код 0, поэтому необходимо произвести нормализацию влево, с вычитанием единицы из порядка произведения.

+ 11.010	00.01111000	11100111	Полученное число
+ 11.111	00.11110001	11001110	Сдвиг
11.001			

Производится округление:

$$\begin{array}{r}
 00.11110001 \ 1100111 \\
 \quad \quad \quad +1 \\
 \hline
 00.11110010 \ 0100111
 \end{array}$$

Передача на запоминание:

1.001 0.11110010

Пример 3. Случай, при котором результат умножения получается меньшим, чем наименьшее число, возможное на машине.

1.011	0.11010110	Множимое
1.010	1.10101101	Множитель

Складываем коды порядков:

$$\begin{array}{r}
 + 11.011 \\
 + 11.010 \\
 \hline
 10.101
 \end{array}$$

Так как порядок получился меньшим, чем порядок наименьшего числа, возможного на машине, то операция непосредственного умножения не производится и на запоминание передается *код нуля*, т. е.

1.000 0.00000000

Пример 4. Случай, в котором результат умножения получается большим, чем наибольшее число, возможное на машине.

0.110	0.11101011	Множимое
0.101	1.10100111	Множитель

Складываем коды порядков:

$$\begin{array}{r} + 00.110 \\ + 00.101 \\ \hline 01.011 \end{array}$$

Так как порядок получился большим, чем наибольшее число, возможное на машине, то операция непосредственного умножения не производится и машина автоматически останавливается (аварийный останов).

Осуществление операции «Умножение» (операция 3)

Операция умножения разбивается на следующие этапы:

- 1) прием чисел из запоминающего устройства на арифметическое устройство;
- 2) умножение чисел;
- 3) нормализация и округление результата;
- 4) передача результата в запоминающее устройство.

1. Прием первого сомножителя на арифметическое устройство мало чем отличается от приема первого слагаемого при операции сложения. Первоначально гасятся коды, оставшиеся на арифметическом устройстве от предыдущей операции (импульс в цепи $У«0»АУ$). Затем код первого числа принимается на $БЗ1Ч$, его знак на ячейку $ЗнБЗ1Ч$ и его порядок на $БЗ1П$ аналогично операции сложения (управляющие потенциалы на цепях $ПЧБЗ1Ч$, $ПЗнБЗ1Ч$ и $ППБЗ1П$).

В отличие от операции сложения передача первого числа на $СмЧ$ не производится (множимое должно остаться на блоке $БЗ1Ч$). Передается на $СмП$ прямым кодом лишь порядок множимого для того, чтобы в дальнейшем к нему можно было прибавить порядок

множителя. Передача порядка с *БЗ1П* на *СмП*, как и при операции сложения, осуществляется посредством подачи импульса в цепь $+ПБЗ1П$.

Множитель (второе число) принимается на *БЗ2Ч*, его знак на *ЗнБЗ1Ч*, а порядок его на *БЗ2П* точно так же, как и при операции сложения (управляющие потенциалы на цепях *ПЧБЗ2Ч*, *ПЗнБЗ1Ч* и *ППБЗ2П*). В отличие от операции сложения гашение *ЗнБЗ1Ч* и *БЗ1Ч* перед приемом второго числа не производится, так как множимое должно остаться на *БЗ1Ч*, в ячейке же *ЗнБЗ1Ч* образуется знак произведения. Знаковая ячейка является счетной ячейкой, и сложение знаков обоих сомножителей дает знак произведения.

Таким образом, после приема чисел код множимого находится на *БЗ1Ч*, а его порядок — на *СмП*. Код множителя находится на *БЗ2Ч*, а его порядок — на *БЗ2П*. В ячейке *ЗнБЗ1Ч* находится знак произведения.

2. При умножении чисел в первую очередь производится суммирование порядков сомножителей. Для этой цели к порядку первого числа, находящегося на *СмП*, необходимо прибавить порядок второго числа, находящегося на *БЗ2П*, что осуществляется подачей импульса в цепь $+ПБЗ2П$. Импульс по этой цепи поступает на вентили, соединяющие *БЗ2П* с *СмП*, и передает порядок с *БЗ2П* на *СмП* прямым кодом.

Для непосредственного умножения цифровых частей сомножителей необходимо обеспечить последовательное поступление кодов разрядов множителя на управляющую цепь, которая осуществляет передачу кода множимого с *БЗ1Ч* на *СмЧ*. Кроме того, после прохождения каждого разряда множителя требуется сдвигать вправо на один разряд сумму частичных произведений, образующихся на *СмЧ*. Эти условия могут быть выполнены путем подачи импульсов сдвига вправо на *СмЧ* и *БЗ2Ч* и посылкой выходного импульса с *БЗ2Ч* через линию задержки в цепь $+ЧБЗ1Ч$ (см. приложение 2).

Импульсы сдвига, подаваемые на *БЗ2Ч* (по цепи *ПрБЗ2Ч*), сдвигают вправо код множителя каждый раз на один разряд. Если текущий младший разряд множителя (*1БЗ2Ч*) имеет код 1, то при сдвиге на правом вы-

ходе *БЗЗЧ* образуется импульс, который через линию задержки поступает на цепь $+ЧБЗЗЧ$ и обеспечивает передачу множимого с *БЗЗЧ* на *СмЧ*. Для того чтобы этот импульс поступал лишь при операции умножения, цепь подачи импульса пропускается через вентиль, который открыт лишь при этой операции (управляющий потенциал на цепи «Умн»). Если текущий младший разряд множителя имеет код 0, то выходного импульса с *БЗЗЧ* не образуется и множимое с *БЗЗЧ* на *СмЧ* не подается. Одновременно импульсы сдвига подаются также на *СмЧ* (по цепи *ПрСмЧ*), что обеспечивает сдвиг суммы частичных произведений каждый раз на один разряд вправо.

При сдвиге кода в *БЗЗЧ* старшие разряды освобождаются от имевшегося в них кода множителя при каждом импульсе на один разряд. Освобождающиеся старшие разряды *БЗЗЧ* можно использовать для приема кодов младших разрядов произведения, что необходимо для операции умножения с выводом удвоенного количества разрядов. Подобный прием младших разрядов на *БЗЗЧ* предусмотрен также и для обычной операции умножения. Для этого выходной импульс с младшего разряда *СмЧ* (с *ДРСмЧ*) подается на вход старшего разряда *БЗЗЧ* (*ЗЗБЗЗЧ*).

Разрешение подачи этого импульса осуществляется вентилем при наличии управляющего потенциала в цепи *Умн*, кода 1 в *ДРСмЧ* и импульса в цепи сдвига *ПрСмЧ*.

Первый импульс сдвига не вызывает изменений на *СмЧ*, так как первоначально на *СмЧ* все разряды имели код 0, а импульс $+ЧБЗЗЧ$, который при этом может возникнуть (если в первом разряде *БЗЗЧ* имеется код 1), приходит уже после сдвига на *СмЧ*. Последующие импульсы сдвига вызывают сдвиг суммы частичных произведений вправо. Таких импульсов сдвига должно быть 32. Таким образом, суммарное количество импульсов сдвига для осуществления непосредственного умножения цифровых частей сомножителей должно быть $32 + 1 = 33$. Подача заданного количества импульсов сдвига определяется специальным счетчиком в устройстве управления машиной. После того как этот счетчик

отсчитает 33 импульса, подача импульсов сдвига прекращается.

3. Полученное на *СмЧ* произведение может иметь ненормализованную форму. В этом случае результат следует сдвигать влево на один разряд и вычитать единицу из кода порядка произведения. Вычитание единицы осуществляется прибавлением ее к обращенному коду суммы порядков, имеющемуся на *СмП* с последующим повторным обращением.

При нормализации требуется сдвигать влево не только код, имеющийся на *СмЧ*, но также и код, имеющийся на *БЗЧ*. При этом выходные импульсы со старшего разряда *БЗЧ* (*32БЗЧ*) должны приниматься на дополнительный разряд *СмЧ* (*ДРСмЧ*). Для этой цели выходной импульс ячейки *32БЗЧ* через ventиль, открытый лишь при операции умножения, поступает на вход ячейки *ДРСмЧ*.

Критерием необходимости нормализовать результат влево, как и при операции сложения, является наличие управляющего потенциала, в цепи «0»*32СмЧ*. Сама нормализация влево осуществляется путем подачи импульсов в цепи *ЛСмЧ*, *ЛБЗЧ* и «+1» *СмП* (вначале производится обращение кода на *СмП*). Нормализация заканчивается тогда, когда в старшем разряде *СмЧ* появляется код 1, т. е. тогда, когда пропадает управляющее напряжение в цепи «0»*32СмЧ*. После окончания нормализации происходит повторное обращение кода на *СмП* для восстановления его правильного значения. Нормализация результата влево может быть заблокирована, если имеется соответствующее указание в коде команды.

Округление результата производится путем прибавления единицы к дополнительному разряду сумматора (импульс в цепи «+1»*ДРСмЧ*).

4. Подача результата в запоминающее устройство осуществляется, как и при операции сложения, путем подачи импульса в цепи *ЗЧСмЧ*, *ЗЗнБЗЧ* и *ЗПСмП*.

При осуществлении операции умножения предусмотрены следующие блокировки:

1) если один из сомножителей равен нулю (код порядка 1.00000), то операция умножения не производится

и в запоминающее устройство передается код «0» (код порядка 1.00000);

2) если при сложении порядков сомножителей порядков произведения получается меньше, чем минимальное число, которое может быть представлено на машине, то операция умножения не производится и в запоминающее устройство передается код «0»;

3) если в процессе нормализации произведения влево порядок получается равный минимально возможному порядку (11.00000), то процесс нормализации влево прекращается и на запоминание передается код «0»;

4) если в процессе умножения произведение превышает максимальное число, которое может быть представлено на машине, то происходит автоматический останов.

1. Если первое число равно нулю, то операция умножения должна быть заблокирована и в запоминающее устройство передан код «0» (код порядка 1.00000). Как и при операции сложения, показателем того, что первое число равно нулю, служит наличие управляющего потенциала на цепи код«0»БЗ1П. Этот потенциал блокирует операцию умножения цифровых частей сомножителей, гасит *СмП* (импульс в цепи *У«0»СмП*) и затем в знаковых ячейках *СмП* устанавливает код 1 (импульс в цепи «+1»Зн*СмП*). Таким образом, на *СмП* образуется код «0» (код 11.00000), который и передается в запоминающее устройство.

Аналогичная блокировка происходит и при возникновении управляющего потенциала на цепи код«0»БЗ2П (второе число равно нулю).

2. Если при сложении порядков сомножителей возникает управляющий потенциал на цепях код«0»*СмП* или «10»Зн*СмП*, то производится блокировка, аналогичная п. 1.

3. При нормализации произведения влево на *СмП* может возникнуть код «0» (код 11.00000). В этом случае нормализация влево должна быть прекращена (имеет большое значение при умножении ненормализованных чисел с нормализацией результата). Указание о прекращении нормализации влево задается возникновением управляющего потенциала на цепи (код«0»*СмП*).

отсчитает 33 импульса, подача импульсов сдвига прекращается.

3. Полученное на *СмЧ* произведение может иметь ненормализованную форму. В этом случае результат следует сдвигать влево на один разряд и вычитать единицу из кода порядка произведения. Вычитание единицы осуществляется прибавлением ее к обращенному коду суммы порядков, имеющемуся на *СмП* с последующим повторным обращением.

При нормализации требуется сдвигать влево не только код, имеющийся на *СмЧ*, но также и код, имеющийся на *БЗЗЧ*. При этом выходные импульсы со старшего разряда *БЗЗЧ* (*ЗЗБЗЗЧ*) должны приниматься на дополнительный разряд *СмЧ* (*ДРСмЧ*). Для этой цели выходной импульс ячейки *ЗЗБЗЗЧ* через вентиль, открытый лишь при операции умножения, поступает на вход ячейки *ДРСмЧ*.

Критерием необходимости нормализовать результат влево, как и при операции сложения, является наличие управляющего потенциала, в цепи «0»*ЗЗСмЧ*. Сама нормализация влево осуществляется путем подачи импульсов в цепи *ЛСмЧ*, *ЛБЗЗЧ* и «+1» *СмП* (вначале производится обращение кода на *СмП*). Нормализация заканчивается тогда, когда в старшем разряде *СмЧ* появляется код 1, т. е. тогда, когда пропадает управляющее напряжение в цепи «0»*ЗЗСмЧ*. После окончания нормализации происходит повторное обращение кода на *СмП* для восстановления его правильного значения. Нормализация результата влево может быть заблокирована, если имеется соответствующее указание в коде команды.

Округление результата производится путем прибавления единицы к дополнительному разряду сумматора (импульс в цепи «+1»*ДРСмЧ*).

4. Подача результата в запоминающее устройство осуществляется, как и при операции сложения, путем подачи импульса в цепи *ЗЧСмЧ*, *ЗЗнБЗЗЧ* и *ЗПСмП*.

При осуществлении операции умножения предусмотрены следующие блокировки:

1) если один из сомножителей равен нулю (код порядка 1.00000), то операция умножения не производится

и в запоминающее устройство передается код «0» (код порядка 1.00000);

2) если при сложении порядков сомножителей порядок произведения получается меньше, чем минимальное число, которое может быть представлено на машине, то операция умножения не производится и в запоминающее устройство передается код «0»;

3) если в процессе нормализации произведения влево порядок получается равный минимально возможному порядку (11.00000), то процесс нормализации влево прекращается и на запоминание передается код «0»;

4) если в процессе умножения произведение превысит максимальное число, которое может быть представлено на машине, то происходит автоматический останов.

1. Если первое число равно нулю, то операция умножения должна быть заблокирована и в запоминающее устройство передан код «0» (код порядка 1.00000). Как и при операции сложения, показателем того, что первое число равно нулю, служит наличие управляющего потенциала на цепи код«0»БЗ1П. Этот потенциал блокирует операцию умножения цифровых частей сомножителей, гасит *СмП* (импульс в цепи У«0»СмП) и затем в знаковых ячейках *СмП* устанавливает код 1 (импульс в цепи «+1»ЗнСмП). Таким образом, на *СмП* образуется код «0» (код 11.00000), который и передается в запоминающее устройство.

Аналогичная блокировка происходит и при возникновении управляющего потенциала на цепи код«0»БЗ2П (второе число равно нулю).

2. Если при сложении порядков сомножителей возникает управляющий потенциал на цепях код«0»СмП или «10»ЗнСмП, то производится блокировка, аналогичная п. 1.

3. При нормализации произведения влево на *СмП* может возникнуть код «0» (код 11.00000). В этом случае нормализация влево должна быть прекращена (имеет большое значение при умножении ненормализованных чисел с нормализацией результата). Указание о прекращении нормализации влево задается возникновением управляющего потенциала на цепи (код«0»СмП).

4. Если при сложении порядков возникнет управляющий потенциал на цепи «01»ЗнСмП (порядок больше максимально возможного), то умножение цифровых частей чисел не блокируется, так как в случае необходимости нормализации произведения влево его порядок уменьшается и может вновь стать меньше максимально допустимого. Поэтому проверка выхода произведения из располагаемых разрядов (управляющий потенциал на цепи «01»ЗнСмП) должна производиться лишь после окончания операции умножения (перед выдачей результата в запоминающее устройство).

Элементарные операции

(На все время операции умножения выход с 1БЗ2Ч соединяется с цепью +ЧБЗ1Ч, выход ДРСмЧ соединяется с входом 32БЗ2Ч, а выход 32БЗ2Ч соединяется с входом ДРСмЧ.)

1°. Установка на «0» всех элементов АУ (У«0»АУ).

2°. Прием первого числа и его знака на БЗ1Ч и ЗнБЗ1Ч, а его порядка на БЗ1П (ПЧБЗ1Ч, ПЗнБЗ1Ч и ППБЗ1П).

3°. Передача порядка с БЗ1П на СмП(+ПБЗ1П).

4°. Прием второго числа на БЗ2Ч и его знака на ЗнБЗ1Ч, а его порядка на БЗ2П (ПЧБЗ2Ч, ПЗнБЗ1Ч и ППБЗ2П).

5°. Передача порядка прямым кодом с БЗ2П на СмП(+ПБЗ2П).

6°. Производится одновременно сдвиг вправо на один разряд на СмЧ и БЗ2Ч (ПрСмЧ и ПрБЗ2Ч) и прибавление единицы к счетчику количества сдвигов. Счетчик отсчитывает 33 сдвига.

7°. Если в коде операции предусмотрено, то производится нормализация влево (если «0»32СмЧ, то ЛСмЧ, ЛБЗ2Ч и «+1» СмП). В этом случае вначале производится обращение сумматора порядков (ОбрСмП), так же, как в операции сложения.

8°. Округление результата путем прибавления единицы на дополнительный разряд СмЧ («+1»ДРСмЧ).

9°. Передача числа с СмЧ и его знака с ЗнБЗ1Ч, а также его порядка с СмП на запоминание (ЗЧСмЧ, ЗЗнБЗ1Ч и ЗПСмП).

Блокировки

1[∞] При появлении на БЗ1П кода «0» (код 1.000000) после операции 2° умножение не производится и на запоминание передается код «0». А именно:

- а) блокируется операция 6°,
- б) устанавливается «0» на СМП (У«0»СМП),
- в) устанавливается «1» в ячейках знака СМП («+1»ЗнСМП).

2[∞]. При появлении на БЗ2П кода «0» (код 1.000000) после операции 4° умножение, так же как и в п. 1[∞], не производится, а на запоминание передается код «0».

3[∞]. При появлении на СМП кода «0» (код 11.000000) или кода 10 на ячейках его знака в результате сложения порядков (после операции 5°) умножение не производится, а на запоминание передается код «0».

4[∞]. При появлении на СМП кода «0» при нормализации результата влево (в процессе операции 7°) нормализация прекращается и на запоминание передается код «0».

5[∞]. При появлении на ячейках знака СМП кода 01, если таковой будет обнаружен после окончания умножения (п. 7°), происходит аварийный останов машины.

§ 3. Деление

При делении одного числа на другое с их порядками частное получается как результат деления кодов цифровых частей этих чисел, а порядок частного равен разности кодов порядков этих чисел. Таким образом, будем иметь:

$$2^p \cdot A : 2^q \cdot B = 2^{p-q} (A : B),$$

знак частного определяется как сумма кодов знаков делимого и делителя.

Если делимое и делитель являются нормализованными числами, то их первые значащие цифры занимают высший разряд. Вычитая из делимого делитель, определяют знак остатка. Если остаток положительный (делимое больше делителя), то первая цифра частного будет 1. Если же остаток отрицательный (делимое

меньше делителя), то первая цифра частного будет 0; попытка вычесть делитель оказалась неудачной. В этом случае можно поступать двумя разными способами.

По первому способу, прибавляя вновь делитель к остатку, возвращаемся к исходному значению делимого. Сдвигаем делимое на один разряд влево и вновь повторяем попытку вычесть делитель. При положительном остатке сдвигаем его влево на один разряд и вновь производим вычитание делителя. Таким образом, после каждой попытки вычесть делитель из остатка, получаем следующую цифру частного: при положительном остатке это будет 1, при отрицательном — 0. При отрицательном остатке необходимо вернуться к первоначальному значению остатка путем обратного прибавления делителя. После того как попытка вычесть остаток будет произведена и в случае необходимости остаток будет возвращен к первоначальному значению, производится сдвиг остатка на один разряд влево, после чего вновь производится попытка вычесть делитель. Эти операции повторяются столько раз, сколько знаков хотят иметь в частном.

Второй способ отличается от первого тем, что не производится восстановление первоначального значения остатка в случае получения отрицательного числа после вычитания делителя. При получении отрицательного числа оно сдвигается на один разряд влево и после этого делитель уже не вычитается, а прибавляется. Положительный знак остатка после вычитания делителя означает, что частное имеет в данном разряде цифру 1; отрицательный знак остатка после вычитания делителя означает, что частное имеет в данном разряде цифру 0.

Действительно, пусть из остатка y_1 вычитается делитель z . Новый остаток будет $y_1 - z$.

После сдвига на один разряд влево новый остаток примет значение (10 означает число 2):

$$y_2 = 10(y_1 - z).$$

Если новый остаток получился отрицательным, то к нему прибавляется (а не вычитается) делитель, т. е. следующий остаток будет:

$$y_2 + z = 10(y_1 - z) + z = 10y_1 - z.$$

Таким образом, получается автоматическое восстановление первоначального остатка, сдвинутого на один разряд влево, из которого вычитается делитель. Если вновь полученный остаток опять будет отрицательный, то производится та же операция, т. е. остаток сдвигается на один разряд влево

$$y_3 = 10(10y_1 - z)$$

и к нему прибавляется z . Тогда получим:

$$10(10y_1 - z) + z = 100y_1 - z.$$

Операции повторяются до тех пор, пока вновь образуемый остаток не будет положительным, после чего производится вычитание делителя. Этот метод дает стандартность операций и отнимает меньше времени, поэтому он и принят для выполнения операции деления на БЭСМ.

Операции, выполняемые при делении, могут быть сформулированы следующим образом.

Из остатка (в частном случае делимого) вычитается или прибавляется делитель, после чего новый остаток сдвигается на один разряд влево и вновь повторяется та же операция. Вычитание делителя производится в том случае, если остаток положителен. Если же остаток отрицателен, то делитель прибавляется к сдвинутому остатку. Если после операции остаток положителен, то в частном получается единица; если же остаток отрицательный, то в частном получается нуль.

При делении двух нормализованных чисел, из которых каждое меньше 1 и больше или равно $0,1\left(\frac{1}{2}\right)$, частное будет всегда меньше 10(2) и больше $0,1\left(\frac{1}{2}\right)$. Если частное получается больше 1, то при представлении его в нормализованном виде порядок частного должен быть увеличен на единицу, т. е. при делении одного числа с порядком p на другое число с порядком q порядок частного будет $p - q + 1$. В этом случае при первом вычитании делителя из делимого остаток получается положительным и для получения 33 цифр частного

(один разряд нужен для округления) операция сложения или вычитания должна повториться 33 раза (см. ниже пример 2). В том же случае, когда при первом вычитании делителя из делимого результат получается отрицательным, число операций для получения 33 цифр частного должно быть равно 34 (см. пример 1).

После производства деления необходимо произвести округление частного, для чего следует прибавить единицу к дополнительному разряду. Так как статические запоминающие блоки не имеют счетных ячеек, то для округления частного оно должно быть передано на сумматор.

Вычитание порядков делимого и делителя производится обычным способом на сумматоре порядков и в особых пояснениях не нуждается. То же относится к сложению кодов знаков делимого и делителя.

При операции деления возможны особые случаи, требующие специальных блокировок.

Если код делимого равен нулю (т. е. его порядок имеет код 1.00000), а делитель — конечное число, то деление не производится и передается на запоминание код «0» (код порядка 1.00000).

Если код делителя равен нулю, то частное будет или бесконечность, или неопределенность. И в том, и в другом случае машина автоматически останавливается.

При вычитании порядков делимого и делителя на сумматоре порядков может получиться порядок частного меньший, чем минимально возможный порядок на машине (т. е. код порядка 11.00000 или код в ячейках знака сумматора порядков 10). В этом случае операция деления не производится и на запоминание передается код «0».

При вычитании порядков делимого и делителя на сумматоре порядков может также получиться порядок частного больший, чем максимально возможный порядок на машине (т. е. в ячейках знака сумматора порядка будет код 01). В этом случае машина автоматически останавливается.

Ниже приводятся примеры, поясняющие операцию деления,

Пример 1. Случай получения отрицательного остатка при первом вычитании.

0.110	0.10110000	Делимое
0.011	1.11011011	Делитель

Производим вычитание кода порядка делителя из кода порядка делимого и получаем код порядка частного:

+ 00.110	Прямой код порядка делимого
+ 11.101	Дополнительный код порядка делителя

00.011	

Складываем отдельно коды знаков делимого и делителя и получаем код знака частного:

+ 0
+ 1

1

Производим деление:

Частное Сумматор

+ 00.10110000	1) Прямой код делимого
+ 11.00100101	Дополнительный код делителя

11.11010101	
0 + 11.10101010	2) Сдвиг
+ 00.11011011	Прямой код делителя

0 00.10000101	
01 + 01.00001010	3) Сдвиг
+ 11.00100101	Дополнительный код делителя

01 00.00101111	
011 + 00.01011110	4) Сдвиг
+ 11.00100101	Дополнительный код делителя

011 11.10000011	
0110 + 11.00000110	5) Сдвиг
+ 00.11011011	Прямой код делителя

0110 11.11100001	

01100	+ 11.11000010	6) Сдвиг
	+ 00.11011011	Прямой код делителя
01100	00.10011101	
011001	+ 01.00111010	7) Сдвиг
	+ 11.00100101	Дополнительный код де-
011001	00.01011111	лителя
0110011	+ 00.10111110	8) Сдвиг
	+ 11.00100101	Дополнительный код де-
		лителя
0110011	11.11100011	
01100110	+ 11.11000110	9) Сдвиг
	+ 00.11011011	Прямой код делителя
01100110	00.10100011	
011001101	+ 01.01000010	10) Сдвиг
	+ 11.00100101	Дополнительный код де-
011001101	00.01100111	лителя
0110011011	00.11001110	Сдвиг

Полученное частное передается на сумматор:

Порядок	Число	
00.011	+ 00.11001101	1
		1
	00.11001110	0

Округление

Передача на запоминание:

0.011 1.11001110

Пример 2. Случай получения положительного остатка при первом вычитании.

0.010	0.11110010	Делимое
1.110	0.10101011	Делитель

Производим вычитание порядков:

+ 00.010	Прямой код порядка делимого
+ 00.010	Дополнительный код порядка делителя
00.100	

Складываем коды знаков чисел:

$$\begin{array}{r} +0 \\ 0 \\ \hline 0 \end{array}$$

Производим деление:

Частное	Сумматор	
	$\begin{array}{r} + 00.11110010 \\ 11.01010101 \\ \hline 00.01000111 \end{array}$	1) Прямой код делимого Дополнительный код делителя
1	$\begin{array}{r} + 00.10001110 \\ 11.01010101 \\ \hline 11.11100111 \end{array}$	2) Сдвиг Дополнительный код делителя
1	$\begin{array}{r} + 11.11000110 \\ 00.10101011 \\ \hline 00.01110001 \end{array}$	3) Сдвиг Прямой код делителя
10	$\begin{array}{r} + 00.11100010 \\ 11.01010101 \\ \hline 00.00110111 \end{array}$	4) Сдвиг Дополнительный код делителя
10	$\begin{array}{r} + 00.01101110 \\ 11.01010101 \\ \hline 11.10000110 \end{array}$	5) Сдвиг Дополнительный код делителя
101	$\begin{array}{r} + 00.01101110 \\ 11.01010101 \\ \hline 11.10000110 \end{array}$	6) Сдвиг Прямой код делителя
101	$\begin{array}{r} + 00.11000110 \\ 00.10101011 \\ \hline 00.00110001 \end{array}$	7) Сдвиг Дополнительный код делителя
1011	$\begin{array}{r} + 00.01100010 \\ 11.01010101 \\ \hline 11.10110111 \end{array}$	8) Сдвиг Прямой код делителя
1011	$\begin{array}{r} + 11.01101110 \\ 00.10101011 \\ \hline 00.00011001 \end{array}$	9) Сдвиг Дополнительный код делителя
10110	$\begin{array}{r} + 00.00011001 \\ 11.10000111 \\ \hline 11.00001110 \end{array}$	Сдвиг
10110	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
101101	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
101101	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
1011010	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
1011010	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
10110101	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
10110101	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
10110101	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	
101101010	$\begin{array}{r} + 00.00011001 \\ 11.01010101 \\ \hline 11.10000111 \end{array}$	

Деление прекращается, так как первая значащая цифра частного достигла ячейки высшего разряда. Порядок частного увеличивается на единицу.

Полученное частное передается на сумматор:

$$\begin{array}{r}
 + \begin{array}{r} 00.100 \\ 1 \end{array} \\
 \hline
 00.101
 \end{array}
 \quad
 \begin{array}{r}
 + \begin{array}{r} 00.10110101 \quad 0 \\ 1 \end{array} \\
 \hline
 00.10110101 \quad 1
 \end{array}
 \quad \text{Округление}$$

Передача на запоминание:

0.101 0.10110101

Пример 3. Случай получения *кода нуля* при вычитании порядков.

1.010	0.10110011	Делимое
0.100	1.11011100	Делитель

Производим вычитание порядков:

$$\begin{array}{r}
 + \begin{array}{r} 11.010 \\ 11.100 \end{array} \\
 \hline
 10.110
 \end{array}
 \quad
 \begin{array}{l}
 \text{Прямой код порядка делимого} \\
 \text{Дополнительный код порядка делителя}
 \end{array}$$

Операция деления не производится, а на запоминание передается *код нуля*, т. е.

1.000 0.00000000

Пример 4. Случай получения порядка частного большего, чем максимально возможный:

0.101	1.11100101	Делимое
1.011	0.10110110	Делитель

Производим вычитание порядков:

$$\begin{array}{r}
 + \begin{array}{r} 00.101 \\ 00.101 \end{array} \\
 \hline
 01.010
 \end{array}
 \quad
 \begin{array}{l}
 \text{Прямой код порядка делимого} \\
 \text{Дополнительный код порядка делителя}
 \end{array}$$

Производится автоматический останов машины.

Осуществление операции «Деление» (операция 4)

Операция деления разбивается на следующие этапы:

1) прием чисел из запоминающего устройства на арифметическое устройство;

2) деление чисел;

3) округление результата и нормализация;

4) передача результата в запоминающее устройство.

1. Прием делимого на арифметическое устройство аналогичен приему первого числа при операции сложения, за исключением того, что передача с *БЗ1Ч* на *СмЧ* всегда производится прямым кодом. Делитель принимается на *БЗ1Ч*. Перед приемом делителя имевшийся на *БЗ1Ч* код делимого должен быть погашен (импульс в цепи *У«0»БЗ1Ч*). Код в ячейке *ЗнБЗ1Ч* (знак делимого) остается, и следовательно, при приеме знака делителя в этой ячейке образуется знак частного (аналогично операции умножения). Порядок делителя принимается на *БЗ2П*. Таким образом, после приема чисел на *СмЧ* находится код делимого, на *СмП* — порядок делимого, на *БЗ1Ч* — код делителя, на *БЗ2П* — порядок делителя и в ячейке *ЗнБЗ1Ч* — код знака частного.

2. При делении чисел первоначально из порядка делимого вычитается порядок делителя, что осуществляется передачей дополнительным кодом порядка делителя с *БЗ2П* на *СмП* (импульс в цепи — *ПБЗ2П*).

В соответствии с принятой методикой деление цифровых частей чисел производится следующим образом. Делитель с *БЗ1Ч* передается на *СмЧ* прямым или дополнительным кодом в зависимости от знака остатка, находящегося на *СмЧ*. Если знак остатка (в частном случае делимого) положительный, то делитель передается дополнительным кодом (вычитается). Если же знак остатка на *СмЧ* отрицательный, то делитель передается прямым кодом (прибавляется). После передачи делителя с *БЗ1Ч* на *СмЧ* полученный остаток на *СмЧ* сдвигается на один разряд влево и операции повторяются. Знак остатка после его сдвига характеризуется кодом в ячейке *Зн2СмЧ*. Если в этой ячейке имеется код 0, то знак остатка положительный и управляющий

потенциал появляется на цепи «0»Зн2СмЧ; если же имеется код 1, то знак остатка отрицательный и управляющий потенциал будет на цепи «1»Зн2СмЧ. Управляющие потенциалы на этих цепях определяют, следует ли передавать делитель с БЗ1Ч на СмЧ прямым или дополнительным кодом. Если управляющий потенциал имеется на цепи «0»Зн2СмЧ, то эту передачу следует произвести дополнительным кодом, если же на цепи «1»Зн2СмЧ — прямым кодом. Передача прямым кодом осуществляется подачей импульса в цепь +ЧБЗ1Ч, а передача дополнительным кодом — подачей импульса в цепь —ЧБЗ1Ч и «+1»ЗнСмЧ. Сдвиг вновь полученного остатка производится подачей импульса в цепь ЛСмЧ.

Цифра частного также определяется знаком остатка. Если остаток положителен (управляющий потенциал на цепи «0»Зн2СмЧ), то цифра частного будет 1, если же остаток отрицателен (управляющий потенциал на цепи «1»Зн2СмЧ), то цифра частного будет 0. Цифры частного запоминаются в БЗ2Ч. Для этой цели они направляются на дополнительный разряд БЗ2Ч (ДРБЗ2Ч). Для того чтобы можно было принять следующую цифру частного, необходимо имеющиеся уже в БЗ2Ч цифры сдвинуть на один разряд влево. Прием цифр частного на дополнительный разряд БЗ2Ч производится посредством подачи импульсов в цепь У«1»ДРБЗ2Ч. Если знак остатка положительный (управляющий потенциал на цепи «0»Зн2СмЧ), то в цепь У«1»ДРБЗ2Ч подается импульс и в ячейке ДРБЗ2Ч устанавливается код 1. Если же знак остатка отрицательный (управляющий потенциал на цепи «1»Зн2СмЧ), импульса в цепь У«1»ДРБЗ2Ч не поступает и в ячейке ДРБЗ2Ч остается код 0. Сдвиг имеющихся на БЗ2Ч цифр частного осуществляется путем подачи импульса в цепь ЛБЗ2Ч. Импульсы сдвига влево на БЗ2Ч поступают одновременно с импульсом сдвига на СмЧ.

Сдвиг кодов на СмЧ и БЗ2Ч, а также передача делителя с БЗ1Ч на СмЧ производятся до тех пор, пока в старшем разряде БЗ2Ч не появится код 1 или пока не будет получено 33 цифры частного.

Во-первых, для определения момента окончания деления со старшего разряда *БЗ2Ч* выведена цепь управления «1»*З2БЗ2Ч*. На этой цепи возникает управляющий потенциал тогда, когда в ячейке *З2БЗ2Ч* получится код 1. Этот управляющий потенциал прекращает операцию деления и к порядку частного добавляется единица (импульс в цепи «+1»*СмП*). Если окончание деления на 33-м отсчете еще не произошло, то к *СмП* ничего не прибавляется, а производится дополнительный, 34-й, сдвиг. Сдвиги отсчитываются счетчиком на 33 отсчета, применяемым и в умножении; 34-й сдвиг осуществляется специальной схемой.

3. Для округления частного полученный на *БЗ2Ч* результат должен быть передан на *СмЧ*. Передача кода с *БЗ2Ч* на *СмЧ* осуществляется через *БЗ1Ч*, так как непосредственной связи *БЗ2Ч* и *СмЧ* не предусмотрено. Предварительно необходимо погасить имеющиеся коды на *БЗ1Ч* и *СмЧ* (импульс в цепях *У«0»БЗ1Ч* и *У«0»СмЧ*). Для передачи кода с *БЗ2Ч* на *БЗ1Ч* подается управляющий потенциал на цепь *ПЧБЗ1Ч* и импульс в цепь *ЗЧБЗ2Ч*. Передача с *БЗ1Ч* на *СмЧ* осуществляется прямым кодом, подачей импульса в цепь «+»*ЧБЗ1Ч*. Округление частного на *СмЧ* производится импульсом в цепи «+1»*ДРСмЧ*.

4. Передача результата в запоминающее устройство осуществляется точно так же, как и при операции сложения (импульс в цепях *ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

При осуществлении операции деления предусмотрены блокировки:

1) если делимое равно нулю (код порядка 1.00000), то операция деления не производится и в запоминающее устройство передается код «0» (код порядка 1.00000);

2) если делитель равен нулю (код порядка 1.00000), то происходит автоматический останов;

3) если порядок частного получается меньше, чем минимальное число, которое может быть представлено на машине, то операция деления не производится и в запоминающее устройство передается код «0»;

4) если порядок частного получается больше, чем максимальное число, которое может быть представлено на машине, то происходит автоматический останов.

1. Если делимое равно нулю (управляющий потенциал на цепи код «0»БЗ1П), то операция деления блокируется и в запоминающее устройство передается код «0». Это осуществляется точно так же, как и при операции умножения в случае, когда один из сомножителей равен нулю.

2. Если делитель равен нулю (управляющий потенциал на цепи код «0»БЗ2П), то происходит автоматический останов.

3. Если при вычитании порядков в ячейках знака Смп возникает код 10 (управляющий потенциал на цепи «10»ЗнСмп), то это означает, что результат меньше минимального числа, которое может быть представлено на машине. В этом случае деление не производится и в запоминающее устройство передается код «0», точно так же как это осуществляется при операции умножения.

4. Если при вычитании порядков или при нормализации результата вправо в ячейках знака Смп образуется код 01 (управляющий потенциал на цепи «01»ЗнСмп), то это означает, что результат больше максимального числа, которое может быть представлено на машине, и необходимо произвести автоматический останов.

Элементарные операции

1°. Установка на «0» всех элементов АУ (У«0»АУ).

2°. Прием первого числа на БЗ1Ч, его знака на ЗнБЗ1Ч, а его порядка, на БЗ1П (ПЧБЗ1Ч, ПЗнБЗ1Ч и ППБЗ1П).

3°. Передача числа с БЗ1Ч на СмЧ прямым кодом (+ЧБЗ1Ч). Передача порядка с БЗ1П на Смп прямым кодом (+ПБЗ1П).

4°. Установка на «0» БЗ1Ч (У«0»БЗ1Ч).

5°. Прием второго числа и его знака на БЗ1Ч и ЗнБЗ1Ч, а его порядка на БЗ2П (ПЧБЗ1Ч, ПЗнБЗ1Ч и ППБЗ2П).

6°. Передача порядка дополнительным кодом с БЗ2П на Смп (—ПБЗ2П).

7°. Прибавляется или вычитается на *СмЧ* число, стоящее на *БЗ1Ч*, в зависимости от кода *Зн2СмЧ* (при «1»*Зн2СмЧ* число прибавляется, при «0»*Зн2СмЧ* — вычитается). (Если «0»*Зн2СмЧ*, то — *ЧБЗ1Ч* и «+1»*ЗнСмЧ*; если «1»*Зн2СмЧ*, то +*ЧБЗ1Ч*.)

После каждого сложения или вычитания число на *СмЧ* и *БЗ2Ч* сдвигается влево, а на ячейке *ДРБЗ2Ч* устанавливается код 1, если имеется «0»*Зн2СмЧ* (*ЛСмЧ*, *ЛБЗ2Ч* и *У«1»ДРБЗ2Ч*, если «0»*Зн2СмЧ*).

8°. При каждом сдвиге прибавляется единица к счетчику на 33 отсчета.

Если при 33-м отсчете счетчика в старшем разряде *БЗ2Ч* появится код 1 («1»*З2БЗ2Ч*), то операция 7° прекращается и затем прибавляется единица к *СмП* («+1»*СмП*). Если же код 1 в *З2БЗ2Ч* при 33-м отсчете счетчика не появится, то производится 34-й сдвиг и к *СмП* ничего не прибавляется.

9°. Установка на «0» *СмЧ* и *БЗ1Ч* (*У«0»СмЧ* и *У«0»БЗ1Ч*).

10°. Передача числа с *БЗ2Ч* на *БЗ1Ч* (*ПЧБЗ1Ч* и *ЗЧБЗ2Ч*).

11°. Передача числа с *БЗ1Ч* на *СмЧ* прямым кодом (+*ЧБЗ1Ч*).

12°. Прибавление единицы к *ДРСмЧ* («+1»*ДРСмЧ*).

13°. Передача числа с *СмЧ* и его знака с *ЗнБЗ1Ч*, а также его порядка с *СмП* на запоминание (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

1°. При появлении кода «0» на *БЗ1П* (код 1.00000) после операции 2° деление не производится и передается код «0» на запоминание, аналогично п. 1° блокировок при умножении.

2°. При появлении кода «0» на *БЗ2П* (код 1.00000) после операции 5° происходит автоматический останов.

3°. При появлении в ячейках знака *СмП* кода 10 при вычитании порядков (после операции 6°) деление не производится и передается код «0» на запоминание, аналогично п. 1° блокировок при умножении.

4°. При появлении на ячейках знака *СмП* кода 01 после операций 6° и 8° происходит автоматический останов.

§ 4. Изменения порядков у чисел

Изменение порядков у чисел эквивалентно умножению или делению числа на 2^s , где показатель s может быть как положительным, так и отрицательным. Величина s может быть задана во втором адресе команды изменения порядков или же находится в разрядах порядка в соответствующей ячейке запоминающего устройства, номер которой указан во втором адресе команды изменения порядков.

В машине предусмотрены следующие операции изменения порядков у чисел:

- А. Сложение порядков (операция 5).
- Б. Вычитание порядков (операция 6).
- В. Изменение порядка по адресу (операция 7).

Осуществление операций изменения порядков чисел (операции 5, 6, 7)

Все операции изменения порядков разбиваются на следующие этапы:

- 1) прием первого числа,
- 2) прием порядка второго числа или порядка из адреса команды,
- 3) изменение порядка,

- 4) передача результата в запоминающее устройство.

1. Прием первого числа для всех операций изменения порядков производится на *БЗ1Ч* и его порядок на *БЗ1П* (управляющие потенциалы в цепях *ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*). Предварительно все элементы арифметического устройства должны быть погашены (импульс в цепи *У«0»АУ*). Принятое число передается с *БЗ1Ч* на *СмЧ*, а его порядок — с *БЗ1П* на *СмП* прямым кодом.

2. Прием порядка второго числа производится на *БЗ2П*. Для сложения и вычитания порядков прием порядка второго числа ничем не отличается от обычного — подаются управляющие потенциалы на цепи *ППБЗ2П*.

Для операции «Изменение порядка по адресу» код порядка должен быть передан со второго адреса команды (от блока запоминания команд). Код адресов команд может выдаваться с блока запоминания команд (*БЗК*) лишь на кодовые шины адреса (*КША*). Для того чтобы можно было код адреса принять на *БЗ2П*, необходимо соответствующие кодовые шины адреса подать на арифметическое устройство (6 шин). Эти шины через соответствующие вентили должны приключаться к входам *БЗ2П* (приложение 3). При подаче управляющего потенциала на цепь *ППБЗ2ПА* эти вентили открываются и код адреса по *КША* поступает на *БЗ2П*.

3. Непосредственное изменение порядка производится путем передачи кода порядка с *БЗ2П* на *СмП*. Для операции «Сложение порядков» и «Изменение порядка по адресу» передача осуществляется прямым кодом (импульс в цепи $+ПБЗ2П$) и происходит сложение порядков. Для операции «Вычитание порядков» передача производится дополнительным кодом (импульс в цепи $-ПБЗ2П$) и происходит вычитание порядков.

4. Передача результата в запоминающее устройство осуществляется обычным способом, путем подачи импульса в цепи *ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*.

При осуществлении операций изменения порядков у чисел предусмотрены блокировки:

1) если при операции вычитания порядков второе число равно коду «0», то операция блокируется и производится аварийный останов. В остальных случаях, если первое или второе числа равны нулю (коды порядков 1.00000), изменения порядков не производится и в запоминающее устройство передается код «0» (код порядка 1.00000);

2) если при изменении порядков получается число меньшее, чем максимально воспроизводимое на машине, то в запоминающее устройство передается код «0» (код порядка 1.00000);

3) если при изменении порядков получается число большее, чем максимально воспроизводимое на машине, то происходит автоматический останов.

Выполнение блокировок ничем не отличается от таковых для операции умножения и не требует пояснений.

Элементарные операции

Сложение порядков (операция 5)

1°. Установка на «0» всех элементов $AУ(У«0»AУ)$.

2°. Прием первого числа на $B31Ч$, его знака на $ЗнB31Ч$ и его порядка на $B31П$ ($ПЧB31Ч$, $ПЗнB31Ч$ и $ППB31П$).

3°. Передача числа с $B31Ч$ на $СмЧ$ прямым кодом и его порядка с $B31П$ на $СмП$ прямым кодом ($+ЧB31Ч$ и $+ПB31П$).

4°. Прием порядка второго числа на $B32П$ ($ППB32П$).

5°. Передача порядка с $B32П$ на $СмП$ прямым кодом ($+ПB32П$).

6°. Когда предусмотрена нормализация результата, то производится сдвиг на $СмЧ$ влево и прибавление единицы к $СмП$ с предварительным обращением его кода. (Если «0» $32СмЧ$, то $ОбрСмП$ и затем производится «+1» $СмП$ и $ЛСмЧ$ до тех пор, пока не станет «1» $32СмЧ$, после чего $ОбрСмП$.)

7°. Передача на запоминание числа с $СмЧ$, его знака с $ЗнB31Ч$ и его порядка с $СмП$ ($3ЧСмЧ$, $3ЗнB31Ч$ и $3ПСмП$).

Вычитание порядков (операция 6)

Элементарные операции те же, что и для «сложения порядков», за исключением операции 5°.

5°. Передача порядка с $B32П$ на $СмП$ дополнительным кодом ($-ПB32П$).

Изменение порядка по адресу (операция 7)

1°. Установка на «0» всех элементов $AУ(У«0»AУ)$.

2°. Прием первого числа на $B31Ч$, его знака на $ЗнB31Ч$ и его порядка на $B31П$ ($ПЧB31Ч$, $ПЗнB31Ч$ и $ППB31П$).

3°. Передача числа с $B31Ч$ на $СмЧ$ прямым кодом и его порядка с $B31П$ на $СмП$ прямым кодом ($+ЧB31Ч$ и $+ПB31П$).

4°. Прием кода со второго адреса $B3К$ на $B32П$ ($ППB32ПА$).

5°. Передача порядка с *БЗ2П* на *СмП* прямым кодом (+*ПБЗ2П*).

6°. Когда предусмотрена нормализация результата, то при «0» *З2СмЧ* производится сдвиг *СмЧ* влево с прибавлением единицы к обращенному предварительно *СмП*. После окончания нормализации производится повторное обращение *СмП*. (Если «0»*З2СмЧ*, то *ОбрСмП* и затем *ЛСмЧ* и «+1»*СмП* до тех пор, пока не станет «1»*З2СмЧ*, после чего снова *ОбрСмП*.)

7°. Передача на запоминание числа с *СмЧ*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП*(*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

1°. При наличии кода «0» на *БЗ2П* в операции вычитания порядков производится аварийный останов. В остальных случаях при возникновении кода «0» на *БЗ1П* (код 1.00000), кода «0» на *БЗ2П*, кода «0» на *СмП* (код 11.00000) и кода 10 на ячейках знака *СмП* на запоминание передается код «0» аналогично п. 1° блокировок при умножении.

2°. При возникновении в ячейках знака *СмП* кода 01 происходит «аварийный» останов машины.

§ 5. Умножение с выводом удвоенного количества разрядов

Операция «Умножение с выводом удвоенного количества разрядов» (необходимая для производства действий с удвоенной точностью) ничем не отличается от обычной операции умножения, за исключением того, что округление результата не производится, а вывод результата в запоминающее устройство производится в две команды.

После производства операции умножения и нормализации результата порядок произведения находится в ячейках сумматора порядков, знак произведения — в ячейке знака *БЗ1Ч*, старшие 33 разряда произведения — в ячейках сумматора, а остальные 31 разряд — в ячейках *БЗ2Ч*. Округление результата при данной операции не производится. Передача порядка, знака и кода старших 32 разрядов произведения выполняется

нормальным путем точно так же, как и при обычной операции умножения с округлением результата.

Передача остальных 32 разрядов произведения происходит при поступлении следующей по номеру команды. (В одной команде нет возможности указать две ячейки запоминающего устройства, куда следует направить полученный результат.) В этой команде в третьем адресе указывается номер ячейки запоминающего устройства, в которую следует направить оставшиеся 32 разряда произведения. Оставшиеся на *ДРСмЧ* и *БЗ2Ч* 32 разряда произведения передаются на *СмЧ* (через *БЗ1Ч*). Порядок оставшегося числа будет на 32 меньше величины, имеющейся на *СмП*, что необходимо иметь в виду при дальнейших операциях с этим числом. Можно оставшиеся 32 разряда произведения нормализовать, что производят обычным путем, сдвигая число на сумматоре влево до тех пор, пока в ячейке старшего разряда не будет код 1. После нормализации передача числа в запоминающее устройство производится обычным образом. Производство или блокировка нормализации второй части произведения определяется цифрой 0 или 1 в шестом разряде кода операции второй команды умножения.

Осуществление операции
«Умножение с выводом
удвоенного количества разрядов»
(операции 8, 9)

При поступлении второй команды (операция 9) блокируется цепь гашения арифметического устройства (цепь *У«0»АУ*). Младшие 32 разряда произведения находятся на *БЗ2Ч* и *ДРСмЧ*, т. е. смещены на один разряд влево по отношению к разрядам *БЗ2Ч*. Предварительно следует все младшие 32 разряда произведения перевести на *БЗ2Ч*, для чего их необходимо сдвинуть на один разряд вправо. Для этой цели требуется подать импульс в цепи *ПрСмЧ* и *ПрБЗ2Ч*. Правый выход *СмЧ* (с *ДРСмЧ*) следует приключить на левый вход *БЗ2Ч* (*32БЗ2Ч*), т. е. соответствующий вентиль должен быть открыт (управляющий потенциал на цепи *Умн(б)*),

Передачу младших 32 разрядов произведения в запоминающее устройство можно было бы осуществить непосредственно с *БЗ2Ч*. Однако в целях стандартности операций целесообразно перевести код с *БЗ2Ч* на *СмЧ*, что производится аналогично переводу частного при операции деления (гашение *БЗ1Ч* и *СмЧ*, передача с *БЗ2Ч* на *БЗ1Ч* и передача с *БЗ1Ч* на *СмЧ* прямым кодом). Младшие 32 разряда произведения могут быть нормализованы на *СмЧ*, если дано соответствующее указание в коде команды. Нормализация осуществляется обычным путем (импульсы в цепях *ЛСмЧ* и «+1»*СмП*, если «0»*32СмЧ* с выполнением *ОбрСмП* до и после нормализации).

Блокировки при операции умножения с выводом удвоенного количества разрядов остаются те же, что и для обычной операции умножения с округлением.

Элементарные операции

Операции 1°—7° те же, что и для умножения с округлением результата.

8°. Передача на запоминание числа с *СмЧ* (первые 32 разряда), его знака с *ЗнБЗ1Ч*, его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Затем поступает следующая команда, в которой блокируется цепь установки на «0» элементов *АУ*.

1°. Сдвиг числа на *СмЧ* и *БЗ2Ч* вправо на один разряд (*ПрСмЧ* и *ПрБЗ2Ч*).

2°. Установка на «0» *БЗ1Ч* и *СмЧ* (*У«0»БЗ1Ч* и *У«0»СмЧ*).

3°. Передача числа с *БЗ2Ч* на *БЗ1Ч* (*ПЧБЗ1Ч* и *ЗЧБЗ2Ч*).

4°. Передача числа с *БЗ1Ч* на *СмЧ* прямым кодом (+*ЧБЗ1Ч*).

5°. Если в команде предусмотрено, то производится нормализация числа на *СмЧ* влево. (Если «0»*32СмЧ*, то *ОбрСмП* и затем *ЛСмЧ* и «+1»*СмП*, пока не станет «1»*32СмЧ*, после чего снова *ОбрСмП*).

6°. Передача на запоминание числа с *СмЧ*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

Блокировки остаются теми же, что и для операции «Умножение».

§ 6. Деление с выводом остатка

При производстве деления в отдельных случаях (при операциях с удвоенным числом знаков) требуется определить не только частное, но и остаток.

Как указывалось выше, деление на БЭСМ осуществляется путем последовательных вычитаний или прибавлений делителя и сдвига получающегося остатка. Если после первого вычитания число на сумматоре останется положительным, то это означает, что делимое больше делителя, и следовательно, прямой код этого числа на сумматоре и будет представлять остаток. Если после вычитания делителя из прямого кода делимого на сумматоре получится отрицательное число, то делимое меньше делителя, и, следовательно, остаток после первой попытки деления будет равен делимому. Для того чтобы получить этот остаток, следует к полученному числу на сумматоре прибавить прямой код делителя. Тем самым вернемся к первоначальному значению числа на сумматоре. Прямой код этого числа будет представлять остаток после первой попытки деления. Аналогичные рассуждения можно применить и к последующим этапам операции деления. Таким образом, получим общее правило для определения абсолютной величины остатка.

Если после прибавления прямого или дополнительного кода делителя к числу, находящемуся на сумматоре, на сумматоре получится положительное число, то его прямой код и будет представлять остаток. Если же величина на сумматоре получится отрицательной, то к ней необходимо прибавить прямой код делителя. Прямой код вновь полученного числа и будет представлять остаток.

Однако принятая методика производства деления с округлением частного, которая применяется чаще, чем деление с выводом остатка, вносит некоторые схемные

осложнения. При делении с округлением частное поступает на вход дополнительного разряда второго блока запоминания ($B32Ч$), и деление производится до тех пор, пока в частном не получится 33 цифры (одна из них в дополнительном разряде).

Уменьшение количества цифр на одну требует изменения схемы управления операцией деления и дополнительных индикаторных цепей. Примерно такие же изменения потребуются, если частное подавать на вход не дополнительного разряда второго блока запоминания ($ДРБ32Ч$), а на вход его основного младшего разряда ($1Б32Ч$), т. е. если предусмотреть дополнительное коммутирующее устройство с его схемой управления и с изменением схемы управления операцией деления.

Другим осложнением является то, что элементарная операция деления заканчивается всегда сдвигом на $B32Ч$. Это необходимо для того, чтобы освободить место в дополнительном разряде $B32Ч$ для последней цифры частного, которая и поступает в него. Так как цепи сдвига влево $B32Ч$ и $СмЧ$ общие, то для определения остатка необходимо величину, полученную на сумматоре после деления, сдвинуть вправо на один разряд и лишь после этого применить вышеизложенное правило определения остатка.

Для упрощения устройств управления операцией деления с выводом остатка целесообразно сохранить полностью принятую схему производства деления с округлением и только лишь блокировать округление частного.

Следует иметь в виду, что после нахождения 33-й цифры частного на сумматоре будем иметь не тот остаток, который ищем, а некоторое другое число, связанное с ним следующими соотношениями.

Обозначим остаток, который ищется, — положительный остаток после получения 32 цифр частного — через T'_n , а фактическое число, которое было на сумматоре после получения 32-й цифры частного, но до сдвига, через T_n . Тогда будем иметь:

1. В случае, если число на сумматоре было положительное ($\langle 0 \rangle 3n2СмЧ$),

$$T'_n = T_n.$$

2. В случае, если число на сумматоре было отрицательное («1»3н2СмЧ).

$$T'_n = T_n + Z \quad \text{или} \quad T_n = T'_n - Z.$$

После сдвига T_n влево на сумматоре чисел будем иметь число $2T_n$. Произведя следующее деление, получим (до сдвига):

1. В том случае, если число на сумматоре (T_n) было положительное:

$$T_{n+1} = 2T_n - Z = 2T'_n - Z,$$

следовательно,

$$T'_n = \frac{T_{n+1} + Z}{2}.$$

2. В том случае, если число на сумматоре (T_n) было отрицательное:

$$T_{n+1} = 2T_n + Z = 2T'_n - Z,$$

следовательно,

$$T'_n = \frac{T_{n+1} + Z}{2}.$$

Таким образом, вне зависимости от знака числа T_{n+1} , стоящего на сумматоре (до сдвига), получаем общую формулу для определения 32-го остатка:

$$T'_n = \frac{T_{n+1} + Z}{2}.$$

Иными словами, для получения 32-го остатка после завершения деления (33 цифры в частном) необходимо произвести следующие операции.

Сдвинуть число на сумматоре вправо на один разряд, прибавить к нему прямой код делителя и полученный результат сдвинуть на один разряд вправо. Тогда прямой код величины на сумматоре будет представлять реальный остаток после 32 цифр частного.

Помимо абсолютной величины остатка, необходимо определить его порядок. Каждый сдвиг влево числа на сумматоре в процессе производства деления уменьшает

порядок остатка на единицу (первоначальный порядок равен порядку делимого). Количество сдвигов на один меньше, чем число необходимых элементов операций деления, так как остаток получается до сдвига результата на сумматоре. Как указывалось выше, число элементарных операций деления для получения 32 цифр частного (первая цифра 1) будет равно 32 или 33. Следовательно, порядок остатка будет или $p - 31$ или $p - 32$, где p — порядок делимого. Соответственно порядок частного будет или $p - q + 1$, или $p - q$, где q — порядок делителя.

Таким образом, для определения порядка остатка надо к порядку частного прибавить порядок делителя. Полученная величина порядка будет на 32 больше действительного порядка остатка, что необходимо иметь в виду при дальнейших операциях с остатками. Полученный остаток можно нормализовать, сдвигая число на сумматоре влево до тех пор, пока в ячейке старшего разряда не будет код 1. Нормализацию можно и блокировать.

Знак остатка соответствует знаку делимого. Поэтому для вывода остатка со знаком необходимо погасить знак частного и принять вместо него знак делимого.

Следует отметить, что операция деления с выводом на запоминание частного и остатка выполняется в две команды. Во второй команде должен быть указан номер ячейки запоминающего устройства, в которую следует направить код остатка (в третьем адресе команды). Кроме того, во втором адресе второй команды необходимо дать номер ячейки запоминающего устройства, в которой хранится код делимого, для того, чтобы по его знаку можно было установить знак остатка.

Осуществление операции «Деление с выводом остатка (операции 10, 11)

При поступлении второй команды следует заблокировать гашение элементов арифметического устройства (блокировка цепи $У \ll 0 \gg АУ$), так как коды, имеющиеся на арифметическом устройстве, необходимы для определения остатка.

Для определения цифровой части остатка необходимо сдвинуть число на *СмЧ* вправо на один разряд, что осуществляется подачей импульса в цепь *ПрСмЧ*. Затем к этому числу следует прибавить делитель, т. е. передать прямым кодом число с *БЗ1Ч* (на *БЗ1Ч* находится код делителя). Сдвинув полученный результат вправо на один разряд (импульс в цепи *ПрСмЧ*), определим цифровую часть остатка. Полученный код остатка может быть нормализован, если это предусмотрено в команде. Нормализация осуществляется обычным путем.

Для определения знака остатка следует погасить знак частного, находящийся в ячейке *ЗнБЗ1Ч* (импульс в цепи *У«0»ЗнБЗ1Ч*), и принять на эту ячейку знак делимого (управляющий потенциал на цепи *ПЗнБЗ1Ч*).

Для определения порядка остатка следует к порядку частного (находится на *СмП*) прибавить порядок делителя (находится на *БЗ2П*). Это осуществляется путем передачи прямым кодом числа с *БЗ2П* на *СмЧ* (импульс в цепи *+ПБЗ2П*).

Передача остатка в запоминающее устройство производится обычным путем (импульсы в цепях *ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки для операции деления с выводом остатка остаются теми же, что и для обычной операции деления с округлением. Дополнительно следует предусмотреть блокировку при нормализации влево остатка (прекращение нормализации влево в случае получения на *СмП* кода порядка 11.00000 и появления управляющего потенциала на цепи *код«0»СмП*).

Элементарные операции

Операции 1°—8° те же, что и при делении с округлением результата.

9°. Передача частного на запоминание с *БЗ2Ч*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП* (*ЗЧБЗ2Ч*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Далее, поступает следующая команда вывода остатка, причем блокируются цепи установки на «0» *АУ* (блокировка цепи *У«0»АУ*):

1°. Число на *СмЧ* сдвигается вправо на один разряд (*ПрСмЧ*),

2°. Передача с *БЗ1Ч* на *СмЧ* прямым кодом (+*ЧБЗ1Ч*).

3°. Установка на «0» *ЗнБЗ1Ч* (*У«0» ЗнБЗ1Ч*).

4°. Прием из запоминающего устройства знака делимого на ячейку *ЗнБЗ1Ч* (*ПЗнБЗ1Ч*).

5°. Передача прямым кодом с *БЗ2П* на *СмП* (+*ПБЗ2П*).

6°. Сдвиг числа на *СмЧ* на один разряд вправо (*ПрСмЧ*).

7°. Если предусмотрено, то производится нормализация влево (если «0»*З2СмЧ*, то *ОбрСмП* и затем *ЛСмЧ* и «+1»*СмП*, до тех пор пока не станет «1»*З2СмЧ*, после чего снова *ОбрСмП*).

8°. Передача на запоминание остатка с *СмЧ* его знака с *ЗнБЗ1Ч* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

При делении (элементарные операции 1°—8°) остаются те же блокировки, что и при операции «Деление с округлением». Для дальнейших операций следует предусмотреть лишь блокировку при нормализации остатка (элементарная операция 7°), которая может быть такой же, как и при операции сложения и вычитания.

ГЛАВА 6

ОПЕРАЦИИ ПЕРЕДАЧИ КОДОВ

§ 1. Передачи чисел

В машине предусмотрены следующие операции передачи числа:

- 1) передача числа нормальная (операция 12),
- 2) передача числа с изменением знака (операция 13),
- 3) передача числа по абсолютной величине (операция 14),
- 4) передача числа с изменением знака в зависимости от знака другого числа (операция 15).

Во всех операциях передачи числа, кроме последней, в первом адресе команды указывается номер числа, которое следует передать, а в третьем адресе команды указывается номер ячейки запоминающего устройства, в которую следует передать данное число. Во втором адресе код отсутствует. Для операции передачи числа с изменением знака в зависимости от знака другого числа во втором адресе дается номер ячейки числа, знак которого следует учесть в операции.

В последнее время функции операции «Нормальная передача числа», были несколько расширены. Дело в том, что при решении задач на машине удобно, не меняя программы с исходными данными и даже не останавливая машины, вводить в ходе вычислительного процесса новые величины, а также выводить контрольные значения. Для этой цели на пульте управления имеется кнопчатая клавиатура, с помощью которой производится управление тремя регистрами по 39 разрядов каждый. Два регистра выполнены на диодах, а один — на триг-

герах. Диодные и триггерный регистры расположены на стойке арифметического устройства. Сигнальные лампы с триггерного регистра выведены на пульт управления. Триггерный контрольный регистр (*ТКЗУ*) предназначен для вывода контрольных значений и результатов вычислений при производстве операции «Печать». На двух диодных регистрах (*Д1КЗУ* и *Д2КЗУ*) оператор может набирать различные коды. Так как данные регистры не входят в систему адресов оперативного запоминающего устройства, то для считывания установленного на них кода используется операция передачи числа, во втором адресе которой могут быть указаны адреса регистров: 3-й разряд — *ТКЗУ*, 2-й разряд — *Д2КЗУ* и 1-й разряд — *Д1КЗУ*. Признаком того, что необходимо обращаться к регистрам пульта управления и блокировать обращение по первому адресу к оперативному запоминающему устройству, является наличие кода 1 в 1-м, 2-м или 3-м разрядах второго адреса. Для операции «Печать» признаком служит 10-й разряд второго адреса (в первом адресе указывается номер ячейки печатаемого числа).

Осуществление операций передач чисел (операции 12—15)

Для всех операций передач чисел прием передаваемого числа производится на *БЗ1Ч*, его знака на ячейку *ЗнБЗ1Ч*, а его порядка на *БЗ1П* (управляющие потенциалы в цепях *ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*). Предварительно все элементы арифметического устройства должны быть погашены (импульс в цепи *У«0»АУ*). Принятое число передается прямым кодом с *БЗ1Ч* на *СмЧ* (импульс в цепи *+ЧБЗ1Ч*), а его порядок с *БЗ1П* на *СмП* (импульс в цепи *+ПБЗ1П*). Передача в запоминающее устройство осуществляется обычным путем (импульс в цепях *ЗЧСмЧ*, *ЗЗнБЗ1Ч*, *ЗПСмП*).

Отдельные виды передач чисел отличаются друг от друга лишь изменением знака у передаваемого числа. При нормальной передаче числа изменения знака не происходит и операция выполняется так, как это описано выше. Для обеспечения передачи числа с изменением знака необходимо изменить знак у передаваемого

числа на обратный. Проще всего это осуществить, устанавливая перед приемом числа ячейку *ЗнБЗІЧ* в положение «1», (импульс в цепи *У«1»ЗнБЗІЧ*). Передача числа по абсолютной величине может быть выполнена посредством гашения кода знака в ячейке *ЗнБЗІЧ* после приема числа (импульс в цепи *У«0»ЗнБЗІЧ*). Передача числа с изменением знака в зависимости от знака другого числа осуществляется путем приема кода знака второго числа на ячейку *ЗнБЗІЧ*. При этом коды знаков складываются, что и обеспечивает соответствующее изменение знака передаваемого числа.

Операции передачи чисел могут производиться как с нормализованными, так и с ненормализованными числами. Кроме того, эти операции могут быть использованы для перевода ненормализованного числа в нормализованное. Для этой цели в коде команды должно быть указание о необходимости нормализации результата. Нормализация производится обычным путем (импульсы в цепях *ЛСмЧ* и «+1»*СмП* с обращениями кода на *СмП*).

Никаких блокировок при передаче чисел не требуется, за исключением блокировки нормализации, если получается число меньшее, чем минимально возможное на машине. В этом случае нормализация влево прекращается и в запоминающее устройство передается код «0» (код порядка 1.00000).

Элементарные операции

Передача числа нормальная (операция 12)

- 1°. Установка на «0» всех элементов *АУ*.
- 2°. Прием первого числа на *БЗІЧ*, его знака на *ЗнБЗІЧ* и его порядка на *БЗІП* (*ПЧБЗІЧ*, *ПЗнБЗІЧ* и *ППБЗІП*).
- 3°. Передача числа с *БЗІЧ* на *СмЧ* прямым кодом и его порядка с *БЗІП* на *СмП* прямым кодом (+*ЧБЗІЧ* и +*ПБЗІП*).
- 4°. Если предусмотрено в команде, то производится нормализация влево. (При операции «Печать», помимо

кода 1 в 10-м разряде второго адреса, имеется всегда код 1 в 6-м разряде кода операции, поэтому в данном случае нормализации не происходит.)

5°. Передача на запоминание числа с *СмЧ*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Передача числа с изменением знака (операция 13)

Элементарные операции те же, что и для операции «Передача числа нормальная», добавляется лишь предварительная элементарная операция установки «1» в ячейке *ЗнБЗ1Ч* (*У«1»ЗнБЗ1Ч*).

Передача числа по абсолютной величине (операция 14)

Элементарные операции те же, что и для операции «Передача числа нормальная». Лишь между третьей и четвертой элементарными операциями добавляется элементарная операция установки на «0» ячейки *ЗнБЗ1Ч* (*У«0»ЗнБЗ1Ч*).

Передача числа с изменением знака в зависимости от знака другого числа (операция 15)

1°. Установка на «0» всех элементов *АУ* (*У«0»АУ*).

2°. Прием первого числа на *БЗ1Ч*, его знака на *ЗнБЗ1Ч* и его порядка на *БЗ1П* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*).

3°. Передача числа с *БЗ1Ч* на *СмЧ* прямым кодом и его порядка с *БЗ1П* на *СмП* прямым кодом (*+ЧБЗ1Ч* и *+ПБЗ1П*).

4°. Прием знака второго числа на *ЗнБЗ1Ч* (*ПЗнБЗ1Ч*).

5°. Если предусмотрено в команде, то производится нормализация влево.

6°. Передача на запоминание числа с *СмЧ*, его знака с *ЗнБЗ1Ч* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗ1Ч* и *ЗПСмП*).

Блокировки

Блокировка предусматривается лишь для окончания нормализации влево в случае появления на *СмП* кода порядка 11.00000 (управляющий потенциал на цепи код «0» *СмП*).

§ 2. Передача порядка числа

В ряде случаев, например при логарифмировании, удобно иметь порядок числа, представленный в нормализованном виде с тем, чтобы с ним можно было в дальнейшем производить все необходимые операции. Иными словами, порядок числа должен быть представлен в виде нормализованного числа и его порядка.

Максимально возможная абсолютная величина порядка может быть равна 31, или в двоичной системе 0.11111. При представлении порядка в нормализованном виде максимальный порядок этого числа может быть не больше 5, или в двоичной системе 0.00101. Преобразование порядка в нормализованное число может быть осуществлено на машине следующим образом.

Как указывалось выше, отрицательный порядок хранится в запоминающем устройстве в виде дополнительного кода и кода знака. Отрицательные числа хранятся в виде прямого кода и кода знака.

Если порядок имеет отрицательный знак, то для преобразования его из дополнительного кода в прямой используется возможность передачи чисел с *БЗ1Ч* на *СмЧ* как прямым, так и дополнительным кодом. Для этого порядок вместе со знаком принимается на пять старших разрядов *БЗ1Ч* и на *ЗнБЗ1Ч*. Для нормализации код порядка передается на *СмЧ*, а на сумматоре порядков устанавливается код 0.00101(5), после чего производят нормализацию. В результате порядок будет представлен в виде нормализованного числа на *СмЧ* с его порядком на *СмП* и его знаком в ячейке *ЗнБЗ1Ч*, после чего можно передать результат в запоминающее устройство обычным путем.

Если порядок числа имеет код 1.00000, т. е. число меньше минимально представляемого на машине (код «0»), то это соответствует отрицательному порядку, бесконечно большому по абсолютной величине, и необходимо произвести автоматический останов машины.

Осуществление операции

«Передача порядка числа» (операция 16)

Для представления порядка в виде нормализованного числа и его порядка необходимо код порядка принять на *БЗ1Ч*. Для этой цели кодовые шины порядка (*КШП*) приключаются через соответствующие вентили на входы ячеек *БЗ1Ч*, включая и ячейку *ЗнБЗ1Ч* (см. приложение 3). При наличии управляющего потенциала на цепи *ППБЗ1Ч* код порядка поступает в старшие разряды *БЗ1Ч*, а знак порядка — в ячейку *ЗнБЗ1Ч*. Предварительно перед приемом порядка все элементы арифметического устройства гасятся (импульс в цепи *У«0»АУ*). Одновременно с приемом порядка на *БЗ1Ч* порядок принимается также на *БЗ2П* (управляющий потенциал на цепи *ППБЗ2П*), что необходимо для целей блокировки. Кроме того, на *БЗ1П* устанавливается код 0.00101(5). Для этой цели на входы первой и третьей ячеек *БЗ1П* по цепи *У«5»БЗ1П* подается импульс. Таким образом, код порядка в ненормализованном виде будет находиться на *БЗ1Ч*, а его порядок на *БЗ1П*.

Для нормализации порядок передается с *БЗ1Ч* на *СмЧ*. Если порядок положительный, то передача осуществляется прямым кодом. Если же порядок отрицательный (представлен дополнительным кодом), то передача производится дополнительным кодом (дополнительный код от дополнительного дает прямой код). При передаче дополнительным кодом знак порядка не передается. Передача осуществляется точно так же, как и при преобразовании кода в операции сложения, т. е. если «0»*ЗнБЗ1Ч*, то +*ЧБЗ1Ч*, если «1»*ЗнБЗ1Ч*, то —*ЧБЗ1Ч*. Одновременно код с *БЗ1П* передается на *СмП*.

Нормализация числа на *СмЧ* производится обычным способом.

Передача результата в запоминающее устройство также производится нормальным образом (импульс в цепях *ЗЧСмЧ*, *ЗЗнБЗІЧ* и *ЗПСмП*).

Блокировка передачи порядка осуществляется в случае, если порядок имеет код 1.00000, т. е. число меньше минимально возможного на машине. В этом случае должен произойти автоматический останов. Для этой блокировки используется цепь *код«0»БЗ2П*. На *БЗ2П* принимается передаваемый порядок и возникновение управляющего потенциала на цепи *код«0»БЗ2П* при операции передачи порядка укажет на необходимость произвести автоматический останов машины.

Кроме того, следует предусмотреть блокировку при нормализации влево в случае возникновения на *СмП* кода 11.00000 (управляющий потенциал на цепи *код«0»СмП*). При появлении управляющего потенциала на цепи *код«0»СмП* нормализация влево прекращается и в запоминающее устройство передается *код «0»* (код порядка 1.00000).

Элементарные операции

1°. Установка на «0» всех элементов *АУ* (*У«0»АУ*).

2°. Установка кода 0.00101(5) на *БЗІП*, прием порядка на *БЗІЧ*, а его знака на *ЗнБЗІЧ* и одновременный прием порядка на *БЗ2П* (*У«5»БЗІП*, *ППБЗІЧ* и *ППБЗ2П*).

3°. Передача числа прямым или дополнительным кодом (в зависимости от кода знака в ячейке *ЗнБЗІЧ*) с *БЗІЧ* на *СмЧ*, а также передача прямым кодом с *БЗІП* на *СмП* (*+ЧБЗІЧ*, если «0»*ЗнБЗІЧ*, или *—ЧБЗІЧ*, если «1»*ЗнБЗІЧ*, а также *+ПБЗІП*).

4°. Если предусмотрено в команде, производится нормализация числа.

5°. Передача в запоминающее устройство числа с *СмЧ*, его знака с *ЗнБЗІЧ* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнБЗІЧ* и *ЗПСмП*).

Блокировки

1°. При *коде «0»* на *БЗ2П* (код порядка 1.00000) после элементарной операции 2° (управляющий потенциал на цепи *код«0»БЗ2П*) происходит автоматический останов.

2°. При появлении *кода «0»* на *СмП* (управляющий потенциал на цепи *код«0»СмП*) при нормализации влево (элементарная операция 4°) нормализация прекращается и в запоминающее устройство передается *код «0»*.

§ 3. Обмен кодами между запоминающими устройствами

В машине предусматривается два вида внешнего запоминающего устройства, или, как оно еще называется, «Магнитного запоминающего устройства» (*МЗУ*): запоминающее устройство в виде магнитной ленты (*МЗУЛ*) с возможностью запуска и останова ее и запоминающее устройство в виде магнитного барабана (*МЗУБ*), все время вращающегося.

С узлом *МЗУ* непосредственно связано устройство ввода на перфоленте (*ПЛ*), использующее автоматику магнитной ленты и предназначенное только для ввода программы задачи в оперативное запоминающее устройство.

Возможны следующие варианты операции обмена кодов с *МЗУ*:

1. Передача кода с *ОЗУ* на *МЗУБ* (запись на барабан).

2. Передача кода с *МЗУБ* на *ОЗУ* (считывание с барабана).

3. Передача кода с *ОЗУ* на *МЗУЛ* (запись на ленту).

4. Передача кода с *МЗУЛ* на *ОЗУ* (считывание с ленты).

5. Ввод кода с *ПЛ* на *ОЗУ* (работа устройства ввода на схеме автоматики *МЗУЛ*).

Управление непосредственно обменом кодами с оперативным запоминающим устройством производится

схемой управления магнитного запоминающего устройства. От центрального управления БЭСМ производится лишь подготовка операции с МЗУ.

Осуществление операций обмена кодами
между запоминающими устройствами
(операции 22—23)

Обмен кодами с МЗУ производится в две команды: МЗ(а) и МЗ(б).

Команду МЗ(а) составляют:

- а) код операции — МЗ(а) (10110);
- б) код адреса А1 — код характера операции «Запись» или «Считывание», «Барабан» или «Лента», «Прямой ход» или «Перемотка» (для ленты), а также код номера группы на барабане или номера ленты;
- в) код адреса А2 — код номера числа и дорожки на барабане, с которых следует начать запись или считывание, или код номера группы, которую следует считать или записать на ленту;
- г) код адреса А3 — код номера ячейки в ОЗУ, с которой следует начать обмен кодов.

Сначала производится установка на нуль всех узлов устройства МЗУ. Затем из блока ЦУОп выдается в МЗУ сигнал, открывающий на приемном регистре входные вентили. На этот регистр поступает код с А1 из блока запоминания команд. После этого на второй приемный регистр передается код с А2. В дальнейшем код с А3, соответствующий номеру ячейки оперативного запоминающего устройства, направляется на МУК.

На этом подготовка обращения к МЗУ заканчивается. Команда МЗ(а) на БЗК гасится и выбирается следующая команда МЗ(б). С приемом команды МЗ(б) блокируется цепь У«0»МЗУ.

Высоким потенциалом из ЦУ по цепи «ПЧ» производится подготовка входных цепей в БЗ1П и БЗ2Ч; она необходима для приема кода из ОЗУ или МЗУ и соответственно передачи в МЗУ или ОЗУ.

Так как выдача и прием кода в МЗУ ведется последовательным способом, а прием и выдача кода

в ОЗУ — параллельным способом, необходимо соответственно преобразовать код. Для преобразования кода используются регистры БЗ1П и БЗ2Ч (в АУ), которые при обращении к МЗУ работают в режиме сдвига.

Сдвиг кода осуществляется синхронизирующими импульсами МЗУ в сторону младших разрядов.

Команду МЗ(б) составляют:

а) код операции МЗ(б) — (010111);

б) код адреса А1 — нулевой код.

Если при проверке МЗУ появляется необходимость в записи одного и того же кода, в 6-м разряде адреса А1 устанавливается код 1.

В этом случае в управлении БЭСМ (ЦУОп) происходит блокировка цепи «+1»УК;

в) код адреса А2 — код номера числа и дорожки на барабанах, на которых следует закончить запись или считывание; количество кодов, которое следует записать или считать с ленты;

г) код адреса А3 — номер ячейки, куда записывается содержимое контрольного счетчика.

Через некоторое время после приема команды МЗ(б) ЦУОп останавливает центральное управление и посылает в МЗУ пусковой импульс (импульс разрешения совпадения). С этого момента управление БЭСМ передается на МЗУ и начинается непосредственно операция обмена кодами.

После окончания обмена кодами в схеме МЗУ вырабатывается сигнал «Импульс завершения операции», которым устанавливаются в положение «0» все элементы внешнего запоминающего устройства. Этот же сигнал поступает в цепь пуска ЦУ, и управление машиной возвращается к ЦУ.

Элементарные операции МЗ(а)

1°. Установка на «0» всех элементов МЗУ.

2°. Выдача в МЗУ А1.

3°. Выдача в МЗУ А2.

4°. Выдача на МУК кода с А3.

Элементарные операции МЗ(б)

1°. Блокировка гашения МЗУ.

2°. Подготовка входных цепей АУ для приема кода из ОЗУ.

3°. Останов ЦУ и работа от МЗУ.

4°. Сдвиг кода на БЗ2Ч и БЗ1П и запись его на МЗУ.

Момент окончания записи производится по коду, указанному во втором адресе команды.

5°. Окончание записи, приход из МЗУ импульса завершения операции, пуск ЦУ.

6°. Выдача АЗ для записи контрольного счетчика.

Элементарные операции обмена кодов даны в сокращенном виде, так как полное их рассмотрение требует более подробного ознакомления со схемой МЗУ.

ГЛАВА 7

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

§ 1. Сдвиги кодов

В ряде случаев требуется произвести сдвиг кодов числа или команды.

Величина необходимого сдвига разрядов и его знак указываются во втором адресе команды. Код сдвига во втором адресе занимает шесть разрядов. Седьмой разряд дает знак сдвига. При положительном коде (код 0) происходит сдвиг числа влево, при отрицательном коде (код 1) — сдвиг числа вправо. Сдвинутое число направляется в ячейку запоминающего устройства, номер которой указан в третьем адресе команды.

При операции сдвига с блокировкой порядков рядки чисел, а следовательно, и коды операций не принимаются во внимание и сдвиги разрядов с ними не производятся. При передаче результата в запоминающее устройство порядок числа (код операции) не передается. Код в ячейке знака числа сдвигу не подвергается и передается в запоминающее устройство без изменений.

При операции сдвига по всем разрядам в сдвиге равноправно участвуют разряды числа, порядка, знака числа и знака порядка.

Осуществление операции «Сдвиг с блокировкой порядков» (операция 17)

Сдвигаемое число принимается на *БЗ2Ч* и *ЗнБЗ2Ч*. Порядок числа принимается на блок *БЗ1П*. Перед приемом числа необходимо погасить все элементы арифметического устройства (импульс в цепи *У«0»АУ*). Прием

числа осуществляется подачей управляющих потенциалов на цепи *ПЧБЗ2Ч*, *ПЗнБЗ2Ч* и *ППБЗ1П*. Сдвиг происходит на блоке *БЗ2Ч*.

Указание, в какую сторону и на сколько разрядов необходимо сдвинуть принятое число, задается в семи младших разрядах второго адреса команды. Код сдвига необходимо принять на арифметическое устройство. От второго адреса блока запоминания команд (*БЗК*) этот код выдается по кодовым шинам адреса (*КША*). Соответствующие кодовые шины адреса приключены через вентили к входам шести ячеек *БЗ2П* (приложение 3). Для того чтобы принять код сдвига на *БЗ2П*, следует подать управляющий потенциал на цепь *ППБЗ2ПА*. При этом указанные вентили открываются и код сдвига поступает на *БЗ2П*. Приняв код сдвига на *БЗ2П*, его передают на *СмП*.

Задание количества разрядов, на которое необходимо сдвинуть число, производится всегда в прямом коде, поэтому передачу кода с *БЗ2П* на *СмП* производят в дополнительном коде по цепи —*ПБЗ2П*. Это дает возможность при сдвиге прибавлять, а не вычитать единицы из *СмП*.

Окончанием операции сдвига служит сигнал переноса с первой ячейки знака *СмП*, сигнал *ПерЗн1СмП*.

В запоминающее устройство передается лишь цифровая часть непосредственно с *БЗ2Ч* и *ЗнБЗ2Ч* (импульс в цепи *ЗЧБЗ2Ч*). Порядок в запоминающее устройство не передается. Никаких блокировок не предусматривается.

Элементарные операции

- 1°. Установка на «0» всех элементов *АУ(У«0»АУ)*.
- 2°. Прием первого числа и его порядка на блоки *БЗ2Ч*, *ЗнБЗ2Ч* и *БЗ1П* (*ПЧБЗ2Ч*, *ПЗнБЗ2Ч* и *ППБЗ1П*).
- 3°. Прием кода со второго адреса *БЗК* на *БЗ2П* (*ППБЗ2ПА*).
- 4°. Передача порядка с *БЗ2П* на *СмП* дополнительным кодом (—*ПБЗ2П*).
- 5°. Сдвиг числа влево на *БЗ2Ч* и прибавление единицы к *СмП* (если код 0 в 7-м разряде *А2*) или сдвиг

на *БЗ2Ч* вправо и прибавление единицы к *СмП* (если код 1 в 7-м разряде *А2*) до тех пор, пока не будет сигнала на цепи *ПерЗн1СмП* (*ЛБЗ2Ч* и «+1»*СмП*, если код 0 в 7-м разряде *А2* или *ПрБЗ2Ч* и «+1»*СмП*, если код 1 в 7-м разряде *А2*).

6°. Передача в запоминающее устройство числа с *БЗ2Ч* и знака с *ЗнБЗ2Ч* (*ЗЧБЗ2Ч* и *ЗЗнБЗ2Ч*).

Осуществление операции «Сдвиг по всем разрядам» (операция, 17)

При данной операции потенциал «, ← » из устройства управления открывает вентили, связывающие блоки *БЗ1П* и *БЗ2Ч* таким образом, что получается один общий регистр сдвига. Код числа и порядка, принятый на этот регистр, может быть сдвинут по всем разрядам. Для этого при операции сдвига одновременно подаются сигналы *ЛБЗ1П* и *ЛБЗ2Ч* или *ПрБЗ1П* и *ПрБЗ2Ч*.

Элементарные операции в команде отличаются от предыдущих в пунктах 5° и 6°.

5°. Сдвиг влево на *БЗ1П* и *БЗ2Ч* и прибавление единицы к *СмП* (если код 0 в 7-м разряде *А2*) или сдвиг на *БЗ1П* на *БЗ2Ч* вправо и прибавление единицы к *СмП* (если код 1 в 7-м разряде *А2*) до тех пор, пока не будет сигнала на цепи *ПерЗн1СмП* (*ЛБЗ2Ч*, *ЛБЗ1П* и «+1»*СмП*, если код 0 в 7-м разряде *А2*, или *ПрБЗ2Ч*, *ПрБЗ1П* и «+1»*СмП*, если код 1 в 7-м разряде *А2*).

6°. Передача в запоминающее устройство числа с *БЗ2Ч*, знака с *ЗнБЗ2Ч* и порядка с *СмП*, куда он предварительно засылается импульсом +*ЛБЗ1П* (сигналы *ЗЧБЗ2Ч* и *ЗПСмП*).

§ 2. Сложение кодов команд

В ряде случаев необходимо производить операции сложения и вычитания цифровых частей чисел без учета кодов, соответствующих порядкам чисел. Это необходимо, например, для производства операций с командами, когда требуется изменить какой-либо адрес команды. При этом в результате операции необходимо сохранить порядок первого числа, соответствующий коду операции изменяемой команды при производстве дейст-

вий с командами. Нормализация полученного результата не должна производиться. Данная операция отличается от обычной операции сложения только тем, что в ней исключаются элементарные операции выравнивания порядков, нормализации результата и округления.

Вычитание с блокировкой порядков в качестве самостоятельной операции не предусматривается, так как оно всегда может быть осуществлено путем прибавления дополнительного кода.

Осуществление операции «Сложение кодов команд» (операция 18).

При сложении кодов команд, как и при большинстве других операций, предварительно гасятся все элементы арифметического устройства (импульс в цепи $У«0»АУ$). Прием первого числа производится обычным образом на $БЗ1Ч$, $ЗнБЗ1Ч$ и $БЗ1П$ (управляющие потенциалы на цепях $ПЧБЗ1Ч$, $ПЗнБЗ1Ч$ и $ППБЗ1П$). Затем число передается на сумматор прямым кодом. В отличие от обычной передачи на $СмЧ$ передается прямым кодом не только цифровая часть числа с $БЗ1Ч$, но также и код, находящийся в ячейке $ЗнБЗ1Ч$. Это вызвано тем обстоятельством, что в коде команды адреса, с которыми производится операция сложения, занимают 33 разряда, т. е. включают и ячейку знака. Передача порядка с $БЗ1П$ на $СмП$ осуществляется нормальным образом. Таким образом, для выполнения передачи числа на сумматор следует подать импульс в цепи $+ЧБЗ1Ч$, $+ЗнБЗ1Ч$ и $+ПБЗ1П$.

Прием второго числа отличается от обычного тем, что принимается лишь цифровая часть числа. Цифровая часть числа поступает на $БЗ1Ч$ и $ЗнБЗ1Ч$ (управляющие потенциалы на цепях $ПЧБЗ1Ч$ и $ПЗнБЗ1Ч$). Предварительно необходимо погасить коды, находящиеся в ячейках $БЗ1Ч$ и $ЗнБЗ1Ч$ (импульс в цепях $У«0»БЗ1Ч$ и $У«0»ЗнБЗ1Ч$).

Сложение кодов цифровой части чисел производится путем передачи кода второго числа на $СмЧ$. При этом разряд знака рассматривается как старший разряд кода числа, т. е., помимо импульса в цепи $+ЧБЗ1Ч$, подается также импульс в цепь $+ЗнБЗ1Ч$.

Полученная сумма кодов, включая разряд знака вместе с кодом порядка первого числа, передается в запоминающее устройство, т. е. подается импульс в цепи *ЗЧСмЧ*, *ЗЗнСмЧ* и *ЗПСмП*.

Никаких блокировок при сложении команд не предусматривается.

Элементарные операции

1°. Установка на «0» всех элементов АУ (*У«0»АУ*).

2°. Прием первого числа на *БЗ1Ч*, его знака на *ЗнБЗ1Ч* и его порядка на *БЗ1П* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*).

3°. Передача прямым кодом числа с *БЗ1Ч* и знака с *ЗнБЗ1Ч* на *СмЧ* и его порядка с *БЗ1П* на *СмП* (*+ЧБЗ1Ч*, *+ЗнБЗ1Ч* и *+ПБЗ1П*).

4°. Установка на «0» *БЗ1Ч* и *ЗнБЗ1Ч* (*У«0»БЗ1Ч* и *У«0»ЗнБЗ1Ч*).

5°. Прием второго числа на *БЗ1Ч* и его знака на *ЗнБЗ1Ч* (*ПЧБЗ1Ч* и *ПЗнБЗ1Ч*).

6°. Передача числа с *БЗ1Ч* и его знака с *ЗнБЗ1Ч* на *СмЧ* прямым кодом (*+ЧБЗ1Ч* и *+ЗнБЗ1Ч*).

7°. Передача в запоминающее устройство числа с *СмЧ* и с *ЗнСмЧ* и его порядка с *СмП* (*ЗЧСмЧ*, *ЗЗнСмЧ* и *ЗПСмП*).

Блокировок никаких не требуется.

§ 3. Циклическое сложение

Операция «Циклическое сложение» предусмотрена для целей контроля, когда необходимо складывать коды двух чисел, рассматривая разряды числа, порядка и знаков как единое целое. При циклическом сложении число, номер которого указан в первом адресе, складывается с числом, номер которого указан во втором адресе. В случае, если имеется перенос из знакового разряда числа, то он направляется в первый разряд сумматора порядков, а перенос из знакового разряда порядка поступает на первый разряд сумматора чисел. Таким образом получается замкнутый цикл. Полученный результат запоминается по третьему адресу. Код данной операции использует код операции сложения команд с единицей в шестом разряде адреса операции.

Осуществление операции «Циклическое сложение» (операция 18)

При данной операции потенциал («СК») открывает вентили, соединяющие цепи переноса *ПерЗн2СмЧ* с цепью «+1»СмП и *ПерЗн2СмП* с цепью переноса первого разряда *СмЧ* (см. приложение 2). Получается замкнутый сумматор на 39 разрядов из блоков *СмЧ* и *СмП*. На этом сумматоре складываются два числа. При сложении числа и их порядки рассматриваются как единое целое.

Элементарные операции при циклическом сложении отличаются от элементарных операций при сложении команд в пунктах 5° и 6°.

5°. Прием второго числа на *БЗ1Ч*, его знака на *ЗнБЗ1Ч* и порядка на *БЗ2П* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ2П*).

6°. Передача числа с *БЗ1Ч*, его знака с *ЗнБЗ1Ч* на *СмЧ* и передача порядка с *БЗ2П* прямым кодом на *СмП* (*+ЧБЗ1Ч*, *+ЗнБЗ1Ч* и *+ПБЗ2П*).

§ 4. Выделение целой части числа

Принципиально выделение целой части числа производится весьма просто. Для этого сдвигают код числа влево и при каждом сдвиге вычитают единицу из порядка числа. Выходящие цифры числа принимаются на какой-либо блок статического запоминания. Сдвиг продолжается до тех пор, пока код порядка не будет равен нулю. Цифры, принятые на внешний блок, будут представлять целую часть числа, а остаток — дробную.

Для осуществления операции выделения целой части число передается на сумматор прямым кодом, если оно положительно, и дополнительным, если оно отрицательно, а его порядок передается на сумматор порядка всегда прямым кодом.

Далее число на сумматоре сдвигается влево и выходящие коды принимаются на регистр. При каждом сдвиге вычитается единица из кода на сумматоре порядка. Сдвиг производится до тех пор, пока код порядка не станет равным нулю. Если в самом начале имелся

код порядка, равный нулю или меньший нуля, то выделение целой части блокируется. При наличии порядка меньше нуля производится денормализация числа, выполняемая аналогично выравниванию порядков в операциях сложения и вычитания.

По окончании сдвига целая часть расположится на младших разрядах регистра, остаток будет на сумматоре числа, а его порядок — на сумматоре порядка.

Остаток на сумматоре, приведенный к нулевому порядку, всегда положителен. Знак целой части соответствует знаку первоначального числа.

При дальнейших операциях с целой частью следует иметь в виду, что в запоминающем устройстве целая часть хранится в виде ненормализованного числа с фиксированной запятой после последнего разряда (см. главу 2, § 2).

Осуществление операции «Выделение целой части числа» (операция 19)

Гашение элементов арифметического устройства и прием числа на *БЗ1Ч*, *ЗнБЗ1Ч* и *БЗ1П* производится обычным путем (импульс в цепи *У«0»АУ* и затем подача управляющих потенциалов на цепи *ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*).

Принятое число передается на *СмЧ* прямым кодом, если оно положительно (импульс в цепи *+ЧБЗ1Ч*, если имеется управляющий потенциал на цепи *«0»ЗнБЗ1Ч*), или дополнительным кодом, если оно отрицательно (импульс в цепи *—ЧБЗ1Ч*, если имеется управляющий потенциал на цепи *«1»ЗнБЗ1Ч*). Порядок числа передается на *СмП* прямым кодом (импульс в цепи *+ПБЗ1П*).

Целая часть числа принимается на младшие разряды *БЗ2Ч*. Если число отрицательно, то и целая часть его также отрицательна и должна быть представлена на *БЗ2Ч* дополнительным кодом. Так как в первоначальном положении все разряды *БЗ2Ч*, на котором образуется целая часть, находятся в положении *«0»*, то в случае отрицательного числа для получения дополнительного кода их необходимо предварительно установить в положение *«1»*.

Для этой цели в арифметическом устройстве предусмотрена цепь $У«1»БЗ2Ч$ (приложение 2), приключенная на все входы основных разрядов $БЗ2Ч$. При наличии управляющего потенциала на цепи $«1»ЗнБЗ1Ч$ в цепь $У«1»БЗ2Ч$ подается импульс, который устанавливает все ячейки основных разрядов $БЗ2Ч$ (кроме $ДРБЗ2Ч$) в положение $«1»$.

Непосредственное выделение целой части производится посредством сдвига влево числа на $СмЧ$ и приема выходящих кодов с $З2СмЧ$ на младший основной разряд $БЗ2Ч$ (на $1БЗ2Ч$). Замыкание этой цепи осуществляется вентилем (см. приложение 2), который открыт лишь при операции «Выделение целой части» (управляющий потенциал на цепи $ЦЧ$). Сдвиг влево производится лишь в том случае, если порядок числа больше нуля, так как лишь при этом абсолютная величина числа больше единицы. Если же порядок числа отрицательный (управляющий потенциал на цепи $«1»Зн2СмП$) или равен нулю (управляющий потенциал на цепи $«0»СмП$), то сдвиг не производится. При наличии управляющего потенциала на цепи $«1»Зн2СмП$ производится денормализация числа (см. выше).

При каждом сдвиге на $СмЧ$ числа влево на один разряд его порядок уменьшается на единицу. Перед началом сдвигов производится обращение порядка на $СмП$ ($ОбрСмП$) и вычитание единицы заменяется прибавлением ее к $СмП$. Для осуществления этого одновременно с импульсами сдвига кодов (импульсы в цепях $ЛСмЧ$ и $ЛБЗ2Ч$) подаются также импульсы в цепь $«+1»СмП$. Сдвиг кодов влево продолжается до тех пор, пока на $СмП$ не получится код 00.00000, т. е. пока не возникнет управляющий потенциал на цепи $«0»СмП$.

После окончания сдвига кодов влево целая часть числа расположится на младших разрядах $БЗ2Ч$ прямым или дополнительным кодом (в зависимости от знака числа), а дробная часть числа останется на $СмЧ$.

Передача дробной части числа в запоминающее устройство производится подачей импульса в цепи $ЗЧСмЧ$ и $ЗПСмП$ (дробная часть числа всегда положительна).

Для передачи целой части в запоминающее устройство в случае отрицательного знака необходимо предварительно дополнительный код целой части преобразовать в прямой код. Для этого код с *БЗ2Ч* передается на *БЗ1Ч* и затем с *БЗ1Ч* дополнительным кодом на *СмЧ*. В целях стандартности операций подобная передача осуществляется и в случае положительного знака; в этом случае число с *БЗ2Ч* на *БЗ1Ч* передается прямым кодом. (Предварительно необходимо погасить коды на *БЗ1Ч* и *СмЧ*.) Передача с *БЗ2Ч* на *БЗ1Ч* производится подачей управляющего потенциала на цепь *ПЧБЗ1Ч* и импульса в цепь *ЗЧБЗ2Ч*. Если целая часть положительна (управляющий потенциал на цепи «0»*ЗнБЗ1Ч*), то число с *БЗ1Ч* передается на *СмЧ* прямым кодом (импульс в цепи +*ЧБЗ1Ч*). Если же целая часть отрицательна (управляющий потенциал на цепи «1»*ЗнБЗ1Ч*), то число с *БЗ1Ч* передается на *СмЧ* дополнительным кодом (импульс в цепи —*ЧБЗ1Ч*).

После преобразования в прямой код целая часть числа передается в запоминающее устройство (импульс в цепях *ЗЧСмЧ* и *ЗЗнБЗ1Ч*).

Элементарные операции

1°. Установка на «0» всех элементов *АУ* (*У*«0»*АУ*).

2°. Прием числа на *БЗ1Ч*, его знака на *ЗнБЗ1Ч* и его порядка на *БЗ1П* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ1П*).

3°. Передача числа прямым или дополнительным кодом (в зависимости от кода знака числа) с *БЗ1Ч* на *СмЧ* и передача порядка с *БЗ1П* на *СмП*; если «1»*ЗнБЗ1Ч*, то все ячейки *БЗ2Ч*, кроме ячейки *ДРБЗ2Ч*, устанавливаются в положение «1» (+*ЧБЗ1Ч*, если «0»*ЗнБЗ1Ч*, или —*ЧБЗ1Ч* и *У*«1»*БЗ2Ч*, если «1»*ЗнБЗ1Ч*, а также +*ПБЗ1П*).

4°. Сдвиг числа на *СмЧ* и *БЗ2Ч* влево и прием выходящего кода с ячейки *З2СмЧ* на ячейку *1БЗ2Ч*, если отсутствуют коды «1»*Зн2СмП* и «0»*СмП*. При каждом сдвиге производится прибавление единицы к обращенному коду на *СмП*. Сдвиг производится до тех пор, пока на *СмП* не будет нуль порядка (код 00.00000). (*ОбрСмП* и затем *ЛСмЧ*, *ЛБЗ2Ч* и «+1» *СмП*, если нет «1»*Зн2СмП*

и «0»СмП.) Если «1»Зн2СмП, то производится денормализация (см. выше, а также вып. 2, гл. IV, § 5).

5°. Передача на запоминание дробной части числа с СмЧ и ее порядка с СмП (ЗЧСмЧ и ЗПСмП).

6°. Установка на «0» БЗ1Ч и СмЧ (У«0»БЗ1Ч и У«0»СмЧ).

7°. Передача целой части числа с БЗ2Ч на БЗ1Ч (ЗЧБЗ2Ч и ПЧБЗ1Ч).

8°. Передача целой части числа прямым или дополнительным кодом (в зависимости от кода знака числа) с БЗ1Ч на СмЧ (+ЧБЗ1Ч, если «0»ЗнБЗ1Ч, или -ЧБЗ1Ч, если «1»ЗнБЗ1Ч).

9°. Передача на запоминание целой части числа с СмЧ и ее знака с ЗнБЗ1Ч (ЗЧСмЧ и ЗЗнБЗ1Ч).

§ 5. Логическое умножение

Логическое (или поразрядное) умножение двух чисел производится по всем 39 разрядам и служит для выделения части кодов числа или команды.

Данная операция обозначается знаком $\wedge$ и для каждого разряда производится по следующим правилам:

$$0 \wedge 0 = 0$$

$$1 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$1 \wedge 1 = 1$$

Например, $10111001 \wedge 00101101 = 00101001$.

Операцию логического умножения оказалось удобно реализовать с помощью другой поразрядной операции, обозначаемой знаком $\vee$, — логического сложения или наложения кодов, которая задается следующими правилами:

$$0 \vee 0 = 0$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$1 \vee 1 = 1$$

Например, $10111001 \vee 00101101 = 10111101$.

На однозначных числах легко проверить следующую формулу, связывающую операции логического умножения

и сложения с обычными операциями сложения и вычитания:

$$a \wedge b = a + b - (a \vee b).$$

Так как при вычислении по формуле $(a + b) - (a \vee b)$ из однозначных чисел получается снова однозначное число, то при действиях над многозначными числами не будут сказываться переносы из разряда в разряд, т. е. соотношение $a \wedge b = (a + b) - (a \vee b)$ верно и для чисел многозначных.

Например:

$$a = 10111001$$

$$b = 00101101$$

$$a + b = 11100110$$

$$a \vee b = 10111101$$

$$- (a \vee b) = 01000011 \text{ — дополнительный код}$$

$$a \wedge b = (a + b) - (a \vee b) = 00101001.$$

В соответствии с вышеизложенным операция логического умножения производится на БЭСМ согласно приведенной формуле. Сначала на сумматорах числа и порядка получают сумму двух чисел, а затем вычитают из нее код наложения двух чисел, который образуется на втором блоке запоминания числа и порядка. Результат логического умножения по третьему адресу передается в оперативное запоминающее устройство. Никаких блокировок при логическом умножении не предусматривается.

Осуществление операции «Логическое умножение» (операция 29)

Вначале при данной операции производится установка на нуль всех элементов АУ ($У \ll 0 \gg АУ$).

Прием первого числа осуществляется на оба блока запоминания (ПЧБЗ1Ч, ПЗнБЗ1Ч, ППБЗ1П, ПЧБЗ2Ч, ППБЗ2П). Затем происходит передача числа с его зна-

ком и порядком на *СмЧ* и *СмП* ($+ЧБЗ1Ч$, $+ЗнБЗ1Ч$ и $+ПБЗ1П$), после чего первый блок запоминания устанавливается в нуль и на него принимается второе число. Второе число одновременно принимается и на второй блок запоминания, на котором находится первое число. Таким образом, на втором блоке получится наложение кодов двух чисел.

Произведя передачу кода второго числа с *БЗ1Ч* на сумматор ($+ЧБЗ1Ч$, $+ЗнБЗ1Ч$, $+ПБЗ1П$), получаем на сумматоре сумму двух кодов. Теперь осталось только вычесть код наложения двух чисел из полученной суммы. Для этого производим гашение *БЗ1Ч* ($У«0»БЗ1Ч$) и передаем код с *БЗ2Ч* на *БЗ1Ч* ($ПЧБЗ1Ч$ и $ЗЧБЗ2Ч$), после чего производим вычитание ($-ЧБЗ1Ч$, $-ЗнБЗ2Ч$ и $-ПБЗ2П$). Необходимо отметить, что при непосредственном выполнении операции на машине пришлось несколько нарушить данную последовательность выполнения. Это касается сложения порядков. Удобнее было сначала сделать вычитание наложенного кода порядков, а затем уже прибавление порядка второго числа, что и отмечено в элементарных операциях.

Элементарные операции

- 1°. Установка на «0» всех элементов *АУ* ($У«0»АУ$).
- 2°. Прием первого числа и его знака на *БЗ1Ч* и *ЗнБЗ1Ч*, а также на *БЗ2Ч* и *ЗнБЗ2Ч* ($ПЧБЗ1Ч$, $ПЗнБЗ1Ч$, $ПЧБЗ2Ч$), а его порядка на *БЗ1П* и *БЗ2П* ($ППБЗ1П$ и $ППБЗ2П$).
- 3°. Передача числа его знака и порядка прямым кодом на *СмЧ* и *СмП* ($+ЧБЗ1Ч$, $+ЗнБЗ1Ч$, $+ПБЗ1П$).
- 4°. Установка на «0» *БЗ1Ч*, *ЗнБЗ1Ч* и *БЗ1П* ($У«0»БЗ1Ч$, $У«0»ЗнБЗ1Ч$, $У«0»БЗ1П$).
- 5°. Прием второго числа, его знака и порядка на *БЗ1Ч*, *ЗнБЗ1Ч* и *БЗ1П*, а также на *БЗ2Ч*, *ЗнБЗ2Ч* и *БЗ2П* ($ПЧБЗ1Ч$, $ПЗнБЗ1Ч$, $ППБЗ1П$, $ПЧБЗ2Ч$, $ППБЗ2П$).
- 6°. Передача наложенного кода с *БЗ2П* на *СмП* дополнительным кодом ($-ПБЗ2П$).

7°. Передача на *СмЧ* и *СмП* второго числа прямым кодом ($+ЧБЗ1Ч$, $+ЗнБЗ1Ч$, $+ПБЗ1П$).

8°. Гашение *БЗ1Ч* ($У«0»БЗ1Ч$).

9°. Передача кода с *БЗ2Ч* на *БЗ1Ч* ($ПЧБЗ1Ч$ и $ЗЧБЗ2Ч$).

10°. Передача наложенного кода обоих чисел с *БЗ1Ч* и *ЗнБЗ2Ч* на *СмЧ* дополнительным кодом ($-ЧБЗ1Ч$, $-ЗнБЗ2Ч$).

11°. Передача числа с *СмЧ* и *СмП* в запоминающее устройство ($ЗЧСмЧ$, $ЗЗнСмЧ$, $ЗПСмП$).

ГЛАВА 8

ОПЕРАЦИИ УСЛОВНЫХ И БЕЗУСЛОВНЫХ ПЕРЕХОДОВ И ПЕРЕДАЧИ УПРАВЛЕНИЯ

§ 1. Сравнение чисел

Операция сравнения определяет дальнейший ход процесса вычисления. В машине предусмотрены три операции сравнения: сравнение двух чисел по их абсолютной величине ($|<|$), сравнение двух чисел с учетом их знака ($<$) и сравнение двух чисел на равенство ($=$). При операциях сравнения в первом и втором адресах команды даются номера ячеек сравниваемых чисел. Сравнение производится путем вычитания из первого числа второго. В первых двух операциях, если результат больше нуля или равен нулю, т. е. знак результата положительный, идет следующая по порядку номеров команда. Если же результат меньше нуля, т. е. знак результата отрицательный, то следующей идет команда, номер которой указан в третьем адресе.

Различие в этих операциях сравнения заключается лишь в том, что при сравнении по абсолютной величине ($|<|$) вычитание чисел производится без учета их знаков, а при сравнении с учетом знака ($<$) учитываются знаки чисел.

В операции «Сравнение на равенство» выполнение следующей по порядку команды происходит только в случае, если сравниваемые числа одинаковы.

Чтобы не производить выравнивания порядков, как это делается при вычитании, для операции сравнения с целью сокращения времени, приняты следующие положения: если порядок первого числа больше порядка второго числа, то знак разности всегда совпадает со

знаком первого числа. Если же порядок первого числа меньше порядка второго числа, то знак разности определяется знаком второго числа и противоположен ему. Для чисел с одинаковыми порядками знак разности определяется непосредственным вычитанием цифровых частей чисел.

Таким образом, для сравнения двух нормализованных чисел необходимо предварительно получить разность порядков чисел. Если эта разность порядков равна нулю, то для определения знака разности чисел необходимо вычесть цифровые части чисел. При положительной разности порядков знак разности чисел совпадает со знаком первого числа. При отрицательной разности порядков знак разности чисел противоположен знаку второго числа. Для сравнения по абсолютной величине, методика остается такой же с учетом того, что знаки обоих чисел всегда положительны.

Для операции сравнения на равенство признаком для перехода по третьему адресу является наличие кода 1 после выполнения операции, хотя бы в одном из 39 разрядов.

Осуществление операций сравнения (операции 20—21 —, 20)

Установка на нуль AU ($U \ll 0 \gg AU$) и прием чисел осуществляются стандартным образом; кроме того, цифровая часть и знак первого числа еще дополнительно принимаются на блок $B324$ ($ПЧБ314$, $ПЗнБ314$, $ППБ31П$, $ПЧБ324$). Затем производится передача на сумматор первого числа в зависимости от знака прямым или дополнительным кодом, также стандартным образом. Порядок всегда передается на сумматор прямым кодом. В случае операции сравнения по абсолютной величине код числа передается всегда прямым кодом. Далее устанавливается на нуль блок $B314$, на единицу $ЗнБ314$ и принимается второе число на этот блок, а его порядок на $B32П$. (Второе число будет находиться на $B314$ с обратным знаком.) В случае сравнения по абсолютной величине знак числа на $ЗнБ314$ не принимается и таким образом второе число на $B314$ становится отрицательным.

Производится вычитание порядков ($-ПБ32П$). По разности порядков определяется ход дальнейших

действий. Если разность порядков положительна («0»Зн2СмП), то знак результата сразу определяется по знаку первого числа, которое в данный момент находится на сумматоре числа. Если же разность порядков отрицательна («1»Зн2СмЧ), то для определения знака результата производится гашение на нуль сумматора (У«0»СмЧ) и передача на него с БЗ1Ч прямым (+ЧБЗ1Ч) или дополнительным (—ЧБЗ1Ч) кодом второго числа. Знак числа на сумматоре (Зн2СмЧ) при этом будет противоположен первоначальному знаку второго числа и будет определять результат сравнения.

В случае равенства порядков («0»СмП) гашение сумматора не производится, а второе число стандартно передается на СмЧ, так как в этом случае необходимо произвести вычитание цифровых частей.

Переход по третьему адресу или к следующей команде в операциях сравнения с учетом знака и по абсолютной величине производится в зависимости от знака результата. Управление передается следующей команде, если «0»Зн2СмЧ, или по третьему адресу, если «1»Зн2СмЧ. Для последнего случая код на блоке ЦУК или МУК гасится путем предварительной установки всех разрядов на «1» с последующим прибавлением единицы к младшему разряду счетчика (У«1»ЦУК и «+1»ЦУК или У«1»МУК и «+1»МУК). Код АЗ передается на ЦУК или МУК (АЗЦУК или АЗМУК) и по этому адресу вызывается новая команда.

Элементарные операции

Сравнение с учетом знаков (операция 20)

1°. Установка на «0» всех элементов АУ (У«0»АУ).

2°. Прием первого числа на БЗ1Ч, его знака на ЗнБЗ1Ч и его порядка на БЗ1Ч (ПЧБЗ1Ч, ПЗнБЗ1Ч и ППБЗ1П).

Дополнительно первое число принимается на БЗ2Ч и его знак на ЗнБЗ2Ч (ПЧБЗ2Ч). Это делается для того, чтобы сохранить после сравнения код первого числа.

3°. Передача числа, а также и его знака с БЗ1Ч на СмЧ прямым кодом (если «0»ЗнБЗ1Ч) или дополнительным кодом (если «1»ЗнБЗ1Ч) и его порядка пря-

мым кодом с *БЗ1П* на *СмП* (+*ЧБЗ1Ч*, если «0»*ЗнБЗ1Ч*, или —*ЧБЗ1Ч* и «+1»*ЗнСмЧ*, если «1»*ЗнБЗ1Ч*, а также +*ПБЗ1П*).

4°. Установка на «0» *БЗ1Ч* и на «1» *ЗнБЗ1Ч* (*У*«0»*БЗ1Ч* и *У*«1»*ЗнБЗ1Ч*).

5°. Прием второго числа на *БЗ1Ч*, его знака на *ЗнБЗ1Ч* и его порядка на *БЗ2П* (*ПЧБЗ1Ч*, *ПЗнБЗ1Ч* и *ППБЗ2П*).

6°. Передача порядка с *БЗ2П* на *СмП* дополнительным кодом (—*ПБЗ2П*).

7°. Установка на «0» *СмЧ*, если «1»*Зн2СмП* (*У*«0»*СмЧ*, если «1»*Зн2СмП*).

8°. Если имеется «1»*Зн2СмП* или «0»*СмП*, то производится передача числа, а также его знака с *БЗ1Ч* на *СмП* прямым кодом (если «0»*ЗнБЗ1Ч*) или дополнительным кодом (если «1»*ЗнБЗ1Ч*). (Если «1»*Зн2СмП* или «0»*СмП*, то +*ЧБЗ1Ч* при «0»*ЗнБЗ1Ч* или —*ЧБЗ1Ч* и «+1»*ЗнСмЧ* при «1»*ЗнБЗ1Ч*).

9°. Если имеется «0»*Зн2СмЧ*, то производится переход к следующей по номеру команде. Если имеется «1»*Зн2СмЧ*, — переход к команде, номер которой указан в третьем адресе. (Если «0»*Зн2СмЧ*, то «+1»*ЦУК* или «+1»*МУК*. Если «1»*Зн2СмЧ*, то *У*«1»*ЦУК*, «+1»*ЦУК* и *АЗЦУК* или *У*«1»*МУК*, «+1»*МУК* и *АЗМУК*.)

Сравнение по абсолютной величине (операция 21)

Данная операция отличается от предыдущей тем, что в элементарной операции 3° происходит только передача числа с *БЗ1Ч* на *СмЧ* всегда прямым кодом (+*ЧБЗ1Ч*) и в операции 5° отсутствует прием знака на *ЗнБЗ1Ч*.

Сравнение на равенство (операция 20)

Эта операция отличается от операции с учетом знака тем, что в элементарной операции 9° производится переход к команде, номер которой указан в третьем адресе, если нет кода 00.00000 в *СмП* или кода 00.0...0 в *СмЧ* («0»*СмП* и «0»*СмЧ*). В случае наличия таких кодов происходит переход к следующей по номеру команде.

§ 2. Операции передачи управления и безусловных переходов

В машине предусмотрено центральное и местное управление командами.

Передача управления с центрального на местное и обратно может осуществляться четырьмя командами:

1. Передача на *МУК* без гашения счетчика *МУК* (*ПМУК*).

2. Передача на *ЦУК* без гашения счетчика *ЦУК* (*ПЦУК*).

3. Изменение номера команды на *МУК* с гашением счетчика *МУК* (*ИМУК*).

4. Изменение номера команды на *ЦУК* с гашением счетчика *ЦУК* (*ИЦУК*).

Таким образом, имеются два типа команд передачи управления с центрального на местное или наоборот — с гашением и без гашения. Команды передачи управления с гашением — «Изменение *ЦУК*» и «Изменение *МУК*» — предусматривают также переход на другую программу или на другой ее участок без перехода на другое управление.

Кроме того, при выполнении команды «Изменение *ЦУК*» предусмотрено формирование и запоминание в *МОЗУ* команды возврата на прерванную программу. Запоминание производится по адресу *A2*.

Элементарные операции

Операции «Передача на *МУК*» и «Передача на *ЦУК*» (операции 24, 25)

1°. Производится прибавление единицы к счетчику *МУК* или *ЦУК*. Если «1»*ТУК*, то «+1»*МУК*. Если «0»*ТУК*, то «+1»*ЦУК* (*ТУК* — триггер управления командами функции, работы которого будут ясны из дальнейшего.)

2°. Осуществляется переход на другое управление. Если выполняется команда «Передача на *МУК*», то *У*«1»*ТУК*. Если выполняется команда «Передача на *ЦУК*», то *У*«0»*ТУК*,

Операция «Изменение ЦУК» (операция 27)

1°. Производится прибавление единицы к счетчику ЦУК или МУК в зависимости от того, на каком управлении работала перед этим машина (если «0»ТУК, то «+1»ЦУК, или если «1»ТУК, то «+1»МУК).

2°. Производится передача в запоминающее устройство кода команды возврата на прерванную программу. Для этого выдается на кодовые шины код из ячеек блока ЦУК (ЗЧЦУК) и код номера операции «Изменение ЦУК» (У«27»). Элементарная операция У«27» осуществляется путем подачи импульсов на КШП (см. приложение 3) в 1-й, 2-й, 4-й и 5-й разряды. Код со счетчика ЦУК поступает на младшие 11 разрядов сумматора, откуда вместе с образованным на СМП кодом 11011 отсылается в запоминающее устройство по второму адресу АЗБЗК.

3°. Если управление перед этим осуществлялось местное, то происходит переход на центральное управление — ТУК переводится из положения «1» в положение «0» (У«0»ТУК).

4°. Производится установка на «0» ячеек блока ЦУК (У«1»ЦУК и затем «+1»ЦУК).

5°. Производится передача кода из ячеек адреса АЗБЗК в ячейки блока ЦУК (АЗЦУК).

Операция «Изменение МУК» (операция 26)

1°. Производится прибавление единицы к коду в счетчике ЦУК или МУК. Это зависит от того, на каком управлении перед этим работала машина. Если «0»ТУК, то «+1»ЦУК, или если «1»ТУК, то «+1»МУК.

2°. Если управление перед этим осуществлялось центральное, то происходит переход на местное управление — ТУК переводится в положение «1» (У«1»ТУК).

3°. Производится установка на «0» ячеек блока МУК (У«1»МУК и затем «+1»МУК).

4°. Код из ячеек адреса АЗБЗК передается в ячейки блока МУК (АЗМУК).

§ 3. Операции останова машины

В машине принято две операции останова машины: условный и безусловный останов.

Последний включается в программу и по окончании вычислений решаемой задачи производит останов машины.

Операция «Условный останов» также включается в программу, но останов машины происходит только в случае включенного тумблера на пульте управления. Эта операция особенно удобна при отладке программ.

При выходе чисел из количества располагаемых разрядов производится «аварийный» останов машины, но непосредственно без участия какой-либо специально предназначенной для этого команды. При этом машина останавливается с соответствующей сигнализацией на пульте управления.

ПРИЛОЖЕНИЕ I

СПИСОК СОКРАЩЕННЫХ ОБОЗНАЧЕНИЙ

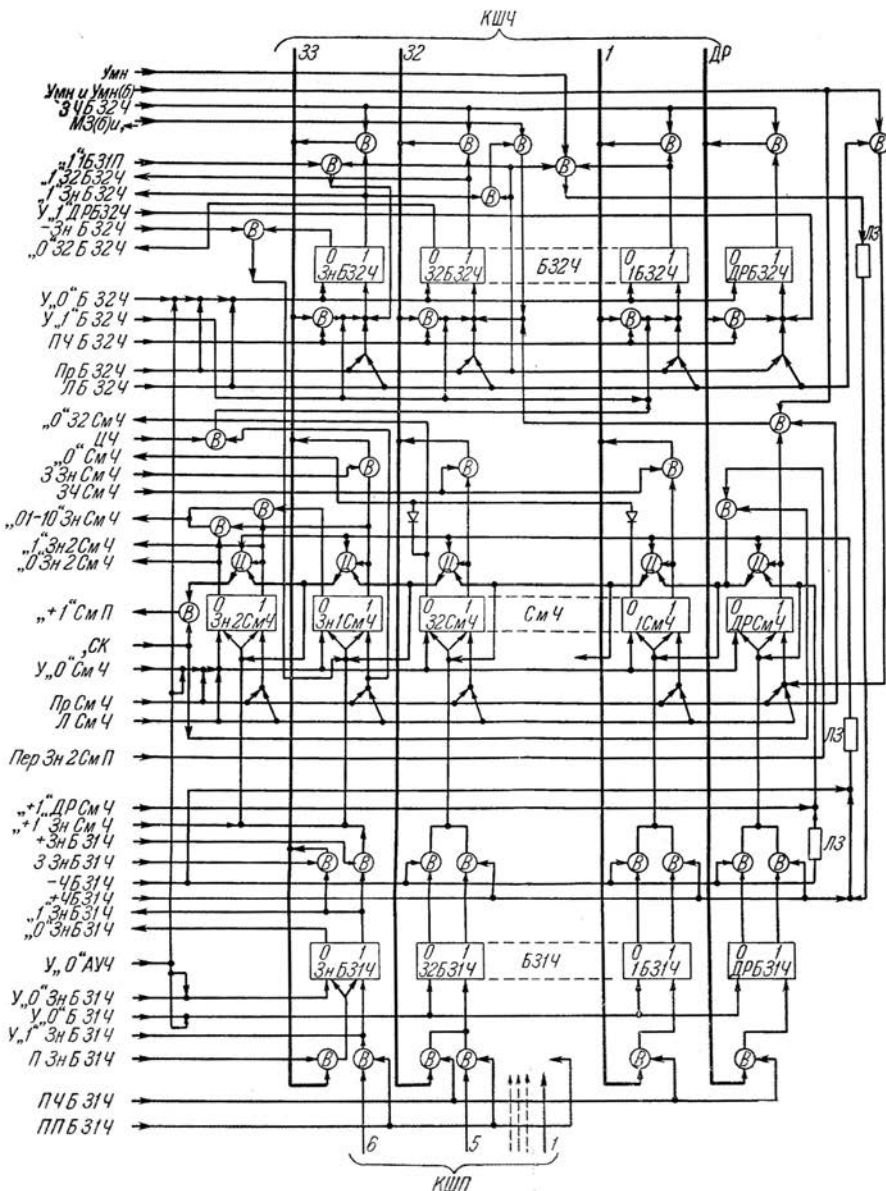
- АОп* — код номера операции (адрес операции команды).
АОпБЗК — ячейки запоминания кода номера операции в *БЗК*.
АУ — арифметическое устройство.
АУП — арифметическое устройство порядка.
АУЧ — арифметическое устройство числа.
А1 — первый адрес кода команды.
А2 — второй адрес кода команды.
А3 — третий адрес кода команды.
А1БЗК — первый адрес блока запоминания команд.
А2БЗК — второй адрес блока запоминания команд.
А3БЗК — третий адрес блока запоминания команд.
АЗМУК — прием кода в ячейки блока *МУК* с ячеек *АЗБЗК*.
АЗЦУК — прием кода в ячейки блока *ЦУК* с ячеек *АЗБЗК*.
БЗК — блок запоминания команд.
БЗ1П — 1-й блок запоминания кода порядка числа.
БЗ1Ч — 1-й блок запоминания кода числа.
БЗ2П — 2-й блок запоминания кода порядка числа.
БЗ2Ч — 2-й блок запоминания кода числа.
В — электронный вентиль (обозначение на схемах).
ДРБЗ2Ч — дополнительный разряд *БЗ2Ч*.
ДРСмЧ — дополнительный разряд *СмЧ*.
Д1КЗУ — 1-е диодное контрольное задающее устройство.
Д2КЗУ — 2-е диодное контрольное задающее устройство.
ЗЗнБЗ1Ч — выдача кода с ячейки знака *БЗ1Ч* на *КШ*.
ЗЗнБЗ2Ч — выдача кода с ячейки знака *БЗ2Ч* на *КШ*.
ЗЗнСмЧ — выдача кода с ячейки знака *СмЧ* на *КШ*.
ЗнБЗ1П — ячейки запоминания знака в *БЗ1П*.
ЗнБЗ1Ч — ячейка знака в *БЗ1Ч*.
ЗнБЗ2П — ячейка знака в *БЗ2П*.
ЗнБЗ2Ч — ячейка знака в *БЗ2Ч*.
Зн1СмП — 1-я ячейка знака *СмП*.
Зн1СмЧ — 1-я ячейка знака *СмЧ*.
Зн2СмП — 2-я ячейка знака *СмП*.
Зн2СмЧ — 2-я ячейка знака *СмЧ*.
ЗПСмП — выдача кода порядка с ячеек *СмП* на *КШ*.
ЗЧБЗ2Ч — выдача кода числа с ячеек *БЗ2Ч* на *КШ*.
ЗЧСмЧ — выдача кода с ячеек *СмЧ* на *КШ*.

- ЗЧЦУК** — выдача кода из ячеек блока **ЦУК**.
КОп — коммутатор операций.
КШ — кодовые шины.
КША — кодовые шины адреса.
КШП — кодовые шины порядков.
КШЧ — кодовые шины чисел.
код «0» — условный ноль на **БЭСМ**, характеризующийся наличием кода 1.00000 в разрядах порядка.
код «0» БЗ1П — наличие кода 1.00000 в ячейках блока **БЗ1П**.
код «0» БЗ2П — наличие кода 1.00000 в ячейках блока **БЗ2П**.
код «0» СМП — наличие кода 1.00000 в ячейках блока **СМП**.
ЛЗ — линия задержки (обозначение на схемах).
ЛБЗ1П — сдвиг кода влево в ячейках **БЗ1П**.
ЛБЗ2Ч — сдвиг кода влево в ячейках **БЗ2Ч**.
ЛСМЧ — сдвиг кода влево в ячейках **СМЧ**.
МЗУ — магнитное запоминающее устройство.
МЗУБ — магнитное запоминающее устройство на барабане.
МЗУЛ — магнитное запоминающее устройство на ленте.
МОЗУ — магнитное оперативное запоминающее устройство.
МУК — местное управление командами.
МУОп — местное управление операциями.
ОбрСМП — обращение кода на **СМП**.
ОЗУ — оперативное запоминающее устройство.
ПерЗн1СМП — перенос из первого разряда знака **СМП**.
ПерЗн2СМП — перенос из второго разряда знака **СМП**.
ПерЗн2СМЧ — перенос из второго разряда знака **СМЧ**.
ПЗнБЗ1Ч — прием кода в ячейку знака **БЗ1Ч**.
ПЗнБЗ2Ч — прием кода в ячейку знака **БЗ2Ч**.
ПЛ — перфолента (вводное устройство).
ППБЗ1П — прием порядка в ячейки блока **БЗ1П**.
ППБЗ1Ч — прием порядка в ячейки блока **БЗ1Ч**.
ППБЗ2П — прием порядка в ячейки блока **БЗ2П**.
ППБЗ2ПА — прием порядка в ячейки блока **БЗ2П** с **КША**.
ПрБЗ1П — сдвиг кода в ячейках блока **БЗ1П** вправо.
ПрБЗ2Ч — сдвиг кода в ячейках блока **БЗ2Ч** вправо.
ПрСмЧ — сдвиг кода в ячейках блока **СмЧ** вправо.
ПУ — пульт управления.
ПЧБЗ1Ч — прием кода числа в ячейки блока **БЗ1Ч**.
ПЧБЗ2Ч — прием кода числа в ячейки блока **БЗ2Ч**.
СМП — сумматор порядков.
СМЧ — сумматор чисел.
СчВх — счетный вход триггера.
Т — триггер (обозначение на схемах).
ТКЗУ — триггерное контрольное запоминающее устройство.
ТП — триггер управления передачей порядка большего числа (триггер порядков).
УК — блок управления командами.
УУ — устройство управления.
У«0» — вход установки триггера в положение «0».
У«0»АУП — установка ячеек **АУП** в положение «0».
У«0»АУЧ — установка ячеек **АУЧ** в положение «0».

- У«0»Б31П — установка ячеек Б31П в положение «0».
 У«0»Б31Ч — установка ячеек Б31Ч в положение «0».
 У«0»Б32П — установка ячеек Б32П в положение «0».
 У«0»Б32Ч — установка ячеек Б32Ч в положение «0».
 У«0»ЗнБ31Ч — установка ячейки знака Б31Ч в положение «0».
 У«0»МЗУ — установка всех элементов МЗУ в положение «0».
 У«0»СмП — установка ячеек СмП в положение «0».
 У«0»СмЧ — установка ячеек СмЧ в положение «0».
 У«0»ТП — установка триггера порядка в положение «0».
 У«0»ТУК — установка триггера управления командами в положение «0».
 У«1» — вход установки триггера в положение «1».
 У«1»Б32Ч — установка ячеек Б32Ч в положение «1».
 У«1»ДРБ32Ч — установка ячейки дополнительного разряда Б32Ч в положение «1».
 У«1»ЗнБ31Ч — установка ячейки знака Б31Ч в положение «1».
 У«1»МУК — установка ячеек блока МУК в положение «1».
 У«1»ТП — установка триггера порядка в положение «1».
 У«1»ТУК — установка триггера управления командами в положение «1».
 У«1»ЦУК — установка ячеек блока ЦУК в положение «1».
 У«5»Б31П — установка кода 0.00101 в ячейках блока Б31П.
 У«27» — установка кода 0.11011 на КШП.
 Ц — цепочка переносов (обозначение на схемах).
 ЦУ — блок центрального управления.
 ЦУК — блок центрального управления командами.
 ЦУОп — блок центрального управления операциями.
 «0»ЗнБ31П — наличие положения «0» в ячейке знака Б31П.
 «0»ЗнБ31Ч — наличие положения «0» в ячейке знака Б31Ч.
 «0»Зн2СмП — наличие положения «0» во второй знаковой ячейке СмП.
 «0»Зн2СмЧ — наличие положения «0» во второй знаковой ячейке СмЧ.
 «0»СмП — наличие кода 0.00000 в ячейках СмП.
 «0»СмЧ — наличие кода 0.00...0 в ячейках СмЧ.
 «0»ТП — наличие положения «0» в триггере порядка.
 «0»ТУК — наличие положения «0» в триггере управления командами.
 «0»32Б32Ч — наличие положения «0» в ячейке 32-го разряда Б32Ч.
 «0»32СмЧ — наличие положения «0» в ячейке 32-го разряда СмЧ.
 «01»ЗнСмП — наличие кода 01 в ячейках знака СмП.
 «01—10»ЗнСмЧ — наличие кода 01 или 10 в ячейках знака СмЧ.
 «1»ДРБ32Ч — наличие положения «1» в ячейке дополнительного разряда блока Б32Ч.
 «1»ЗнБ31П — наличие положения «1» в ячейке знака Б31П.
 «1»ЗнБ31Ч — наличие положения «1» в ячейке знака Б31Ч.
 «1»ЗнБ32Ч — наличие положения «1» в ячейке знака Б32Ч.
 «1»Зн2СмП — наличие положения «1» во второй знаковой ячейке СмП.
 «1»Зн2СмЧ — наличие положения «1» во второй знаковой ячейке СмЧ.

- «1»ТП — наличие положения «1» в триггере порядка.
«1»ТУК — наличие положения «1» в триггере управления командами.
«10»ЗнСмП — наличие кода 10 в ячейках знака СмП.
«1»БЗ1П — наличие положения «1» в ячейке первого разряда БЗ1П.
«1»32БЗ2Ч — наличие положения «1» в ячейке 32-го разряда БЗ2Ч.
+ ЗнБЗ1Ч — выдача прямого кода с ячейки знака БЗ1Ч в ячейки знака СмЧ.
+ ПБЗ1П — передача прямого кода с ячеек БЗ1П на СмП (прибавление кода БЗ1П к СмП).
+ ПБЗ2П — передача прямого кода с ячеек БЗ2П на СмП (прибавление кода на БЗ2П к СмП).
+ ЧБЗ1Ч — передача прямого кода с ячеек БЗ1Ч на СмЧ (прибавление кода на БЗ1Ч к СмЧ).
— ЗнБЗ2Ч — выдача обратного кода с ячейки знака БЗ2Ч в ячейки знака СмЧ.
— ПБЗ2П — передача дополнительного кода с ячеек БЗ2П на СмП (вычитание кода на БЗ2П из СмП).
— ЧБЗ1Ч — передача дополнительного кода с ячеек БЗ1Ч на СмЧ (вычитание кода на БЗ1Ч из СмЧ).
«+1»ЗнСмП — прибавление единицы в ячейки знака СмП.
«+1»ЗнСмЧ — прибавление единицы в ячейки знака СмЧ.
«+1»ДРСмЧ — прибавление единицы в ячейку дополнительного разряда СмЧ.
«+1»МУК — прибавление единицы к коду в ячейках блока МУК.
«+1»СмП — прибавление единицы к коду в ячейках СмП.
«+1»ЦУК — прибавление единицы к коду в ячейках блока ЦУК.
-

ФУНКЦИОНАЛЬНАЯ СХЕМА АУЧ



ФУНКЦИОНАЛЬНАЯ СХЕМА АУП

