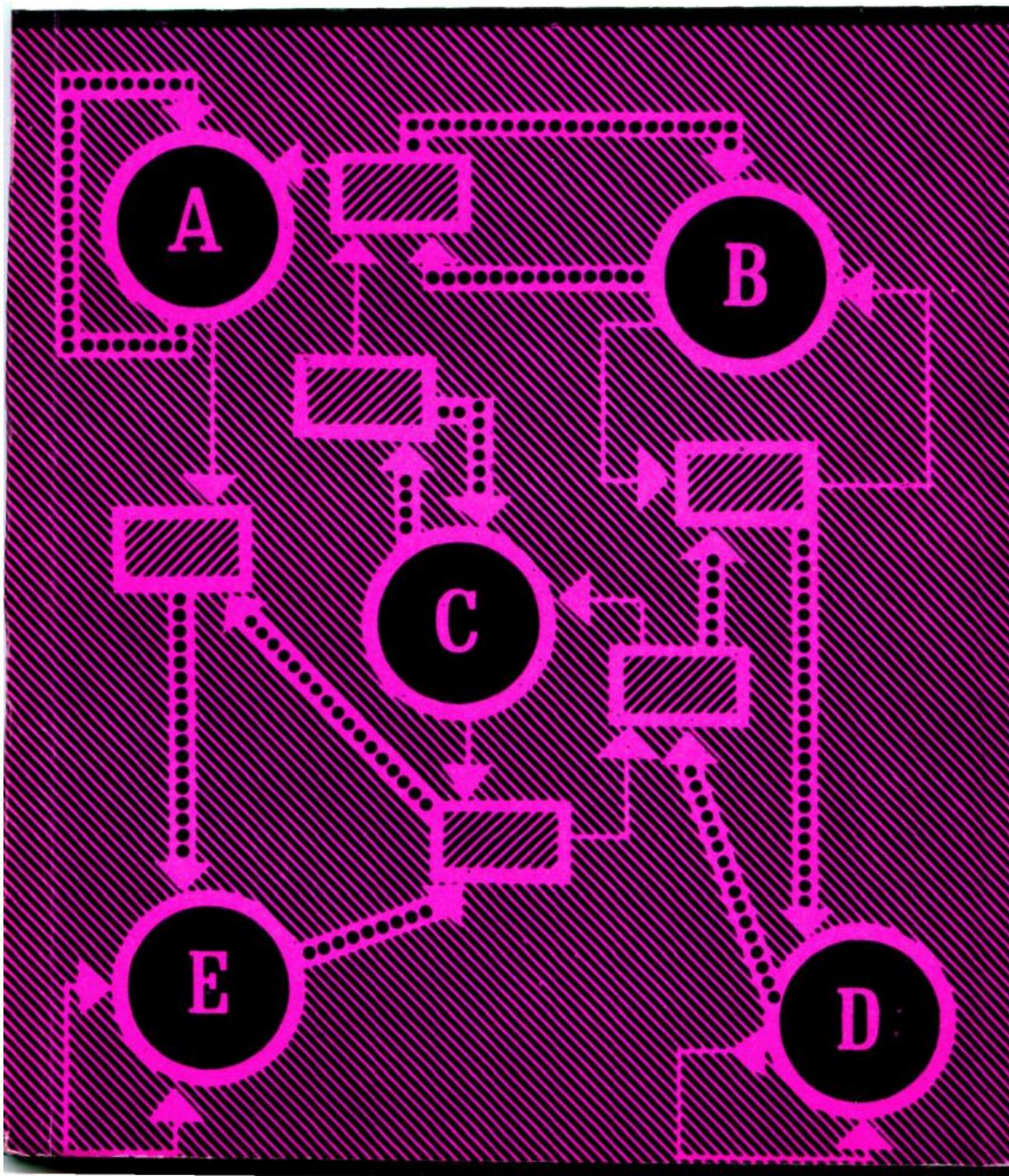


*В. В. Пржиялковский,
Г. Д. Смирнов,
В. Я. Пыхтин*

ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА „МИНСК-32“



ПОСОБИЯ
ДЛЯ РАБОТНИКОВ
ВЫЧИСЛИТЕЛЬНЫХ
ЦЕНТРОВ

В. В. ПРЖИЯЛКОВСКИЙ
Г. Д. СМИРНОВ
В. Я. ПЫХТИН

ЭЛЕКТРОННАЯ
ВЫЧИСЛИТЕЛЬНАЯ
МАШИНА
„МИНСК-32“



«СТАТИСТИКА»
МОСКВА 1972

В книге приводятся основные технические параметры, алгоритмы работы, общая структура и возможности электронной вычислительной машины «Минск-32». Подробно характеризуются ее основные особенности. Рассматриваются также вопросы обеспечения совместимости с ЭВМ «Минск-2», «Минск-22».

Книга рассчитана на специалистов, работающих на машинах этой серии.

*Пржиялковский Виктор Владимирович,
Смирнов Геннадий Дмитриевич,
Пыхтин Вадим Яковлевич.*

**ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ
МАШИНА «МИНСК-32»**

Редактор *И. Г. Дмитриева*

Техн. редактор *В. А. Чуракова*

Корректор *А. Т. Сидорова*

Худ. редактор *Т. В. Стихно*

Обложка художника *Л. С. Эрмана*

Сдано в набор 10/VIII 1971 г.
Подписано к печати 15/II 1971 г.
Формат бумаги 60 × 90¹/₁₆.

Бумага № 1.

Объем 10,0 печ. л. Уч.-изд. л. 10,48.

Тираж 46000 экз. А-10162.

(Тематич. план 1972 г. № 56).

Заказ № 712. Цена 56 коп.

Издательство «Статистика»,
Москва, ул. Кирова, 39.

Типография им. Котлякова
издательства «Финансы» Комитета
по печати при Совете
Министров СССР.
Ленинград, Садовая, 21.

СОДЕРЖАНИЕ

Введение	7
Глава I. Общие данные Эвм « Минск-32»	8
1.1. Анализ семейства машин второго поколения «Минск».....	8
1.2. Назначение и технические данные «Минск 32»	10
Технические данные процессора	10
Технические данные внешнего оборудования.....	11
1.3. Состав машины	12
1.4. Основные особенности эвм «Минск-32»	14
Представление информации.....	14
Структура команды	15
Адресация.....	16
1.5. Основные режимы работы процессора.....	17
Характеристика тактов.....	18
Выборка команды.....	19
Особенности выполнения команд.....	20
1.6. Система команд	21
1.7. Основные режимы обработки цифровой и алфавитно-цифровой информации	22
1.8. Система контроля работы процессора.....	24
1.9. Взаимодействие устройств процессора	26
Глава II. Многофункциональность аппаратуры процессора	27
2.1. Арифметическое устройство	27
2.2. Организация посимвольного обмена в ЭВМ, имеющих слово фиксированной длины	32
Соотношение символа и слова	32
Схема формирования контрольного разряда	33
Контроль сдвига массива информации	33
Глава III Организация многопрограммности однопроцессорной ЭВМ	38
3.1. Анализ загрузки оборудования машины «Минск-22».....	38
Ввод	38
Счет.....	38
Обмен МОЗУ—НМЛ	39
Вывод.....	39
3.2. Организация многопрограммности.....	39
Блок-схема процессора	40
Система прерывания	40
3.3. Средства обеспечения многопрограммности.....	42
Инвариантность программ.....	42
Защита программ	43
Электронные часы (ЭДВ)	44
Управляющая информация многопрограммной ЭВМ	45
Оптимизация размещения управляющей информации многопрограммной ЭВМ.....	46
3.4. Некоторые временные параметры многопрограммной ЭВМ.....	47
Загрузка устройств и системное время.....	47
Коэффициент многопрограммности.....	49
Глава IV Система универсальной связи с внешними устройствами (СУС ВнУ)	51
4.1. Каналы связи	51
Частотные характеристики СК и МК	53
4.2. Система подключения ВнУ	54
4.3. Массивы обмена с внешними устройствами.....	54
4.4. Взаимодействие процессора с внешними устройствами	55
4.5. Приостановка	56
4.6. Выполнение операции ввода-вывода.....	57
Команды ввода-вывода	57
Управляющие слова (УС).....	57
4.7. Программное обслуживание работы с внешними устройствами.....	59
Пуск ВнУ.....	60
Конец работы ВнУ	60
Реакция на сбой ВнУ и каналов связи	60
4.8. Особенности работы устройств мультиплексного и селекторного каналов	60
Выполнение команд связи с ВнУ.....	60
Медленная приостановка.....	62
Организация селекторного канала	64
Окончание обмена	65
4.9. Особенности работы с активными ВнУ.....	65
Активные ВнУ мультиплексного канала	65
Активные ВнУ селекторного канала	66

4.10. Требования к внешним устройствам.....	66
Структура подключения ВнУ.....	66
Требования к согласованию связей ВнУ.....	67
Требования к параметрам сигналов внешних устройств.....	69
Требования к помещению и размещению ВнУ.....	70
Глава V Особенности организации многомашиных комплексов на базе ЭВМ «МИНСК-32»	71
5.1. Некоторые проблемы пользователей ЭВМ широкого применения	71
5.2. Анализ системы «МИНСК-222».....	72
5.3. Особенности ЭВМ широкого применения, влияющие на структуру системной связи ЭВМ.....	72
5.4. Основные особенности создания системы на базе ЭВМ «МИНСК-32».....	74
Основные требования	74
Особенности системной связи	74
5.5. Формат сообщений	75
5.6. Подключение ЭВМ.....	76
Выбор структуры связи.....	76
Связь ЭВМ «Минск-32» в систему	77
5.7. Различные режимы обмена информацией.....	78
Попарный обмен информацией.....	78
Одновременный обмен с несколькими машинами.....	79
5.8. Некоторые потенциальные системные возможности.....	80
Глава VI Вопросы обеспечения совместимости с ЭВМ «МИНСК-2», «МИНСК-22», «МИНСК-22М»	82
6.1. Возможность программного моделирования.....	82
Различные способы совмещения	82
Совмещение на уровне алгоритмического языка.....	82
Трансляция.....	83
Моделирование.....	83
Эмуляция.....	83
Моделирование двух различных по структуре ЭВМ.....	83
6.2. Эмуляция машины «МИНСК-22М» на ЭВМ «МИНСК-32»	86
Некоторые аспекты реализации совместимости	86
Совместимость структур команд.....	87
Совместимость индексирования.....	88
Совместимость команд внешних устройств и управление программой совместимости.....	88
Глава VII Система математического обеспечения ЭВМ «МИНСК-32».....	90
Приложение 1.....	95
Приложение 2.....	98
Приложение 3.....	101
Литература	104

ВВЕДЕНИЕ

Проектом Директив XXIV съезда КПСС в новом пятилетии намечено значительное увеличение выпуска электронных вычислительных машин, нашедших самое широкое применение в народном хозяйстве. Одной из распространенных электронных машин является ЭВМ «Минск-32», рассчитанная на крупносерийное производство. Для того чтобы обеспечить эффективную работу у широкого круга пользователей, ее структура, элементная база, конструкция, состав математического обеспечения должны отвечать целому ряду требований, важнейшими из которых являются универсальность и высокое отношение производительности к стоимости.

Сферой применения ЭВМ «Минск-32» являются научные, научно-технические расчеты и решение информационно-логических задач, связанных с вводом, хранением, обработкой и выводом большого количества информации.

Универсальность ЭВМ для такого применения может быть достигнута соответствующим набором команд, оперирующих как со словами, так и с отдельными символами, гибкой структурой, позволяющей подключение большого количества внешних устройств и наличием в производстве широкого набора внешних устройств различного назначения.

Отношение производительности к стоимости, как известно, падает с ростом производительности ЭВМ. Однако для машины широкого применения, предназначенной для крупносерийного производства, решающее значение имеет стоимость ЭВМ, которая не должна быть высокой.

При проектировании серии ЭВМ типа «Минск» производительность каждой новой модели определялась условием ее стоимости, на которую значительное влияние оказывает выбор структуры, элементной базы и конструкции. Пользователя ЭВМ интересует не только покупная стоимость ЭВМ, но и эксплуатационные расходы, отнесенные к единице времени. На снижение последних большое влияние оказывает повышение надежности ЭВМ, простота конструкции, технической документации и максимальная преемственность с предыдущей моделью. Наличие программной совместимости с предыдущей моделью настолько сокращает затраты на программирование, что создание совместимых моделей в настоящее время стало важнейшей необходимостью.

Так, например, объем прикладных программ пользователей ЭВМ «Минск-22» и «Минск-22М» оценивается десятками миллионов команд общей стоимостью несколько сотен миллионов рублей. Задача обеспечения технического прогресса средств вычислительной техники при условии сохранения имеющегося задела прикладных программ является в настоящее время одной из наиболее сложных проблем проектирования.

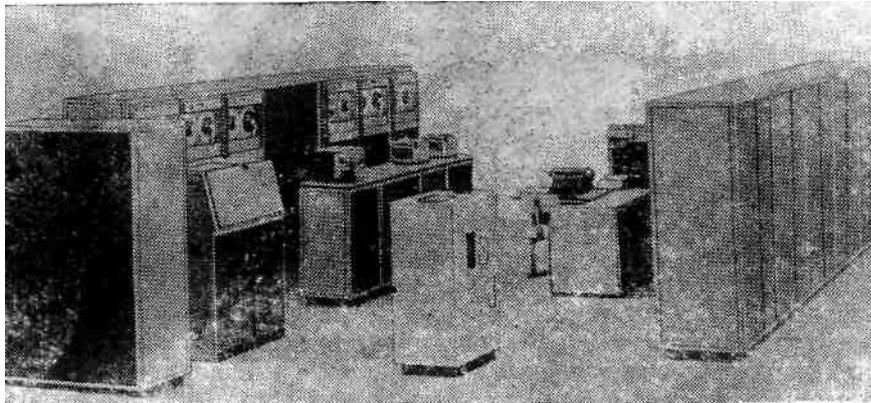
В настоящей книге рассматриваются особенности организации ЭВМ широкого применения «Минск-32», при проектировании которой были предприняты попытки наиболее оптимальным образом решить целый ряд противоречивых проблем, затронутых выше.

Книга не является подробным описанием устройств и схем машины. В ней нет последовательного изложения функционирования ЭВМ при выполнении вычислительного процесса. Авторы попытались сосредоточить внимание на наиболее интересных, по их мнению, вопросах. Тем не менее, она поможет широкому кругу специалистов ознакомиться с конкретной организацией машины и оценить степень эффективности технических решений, принятых в ходе проектирования.

ГЛАВА I
ОБЩИЕ ДАННЫЕ ЭВМ «МИНСК-32»

1.1. АНАЛИЗ СЕМЕЙСТВА МАШИН ВТОРОГО ПОКОЛЕНИЯ «МИНСК»

Многопрограммная универсальная вычислительная машина «Минск-32» является продолжением машин второго поколения семейства «Минск». Она программно совместима с моделями «Минск-2», «Минск-22», «Минск-22М». Программы, написанные в соответствии с



Общий вид ЭВМ «Минск-32».

инструкциями по программированию для этих машин, могут быть без дополнительного перепрограммирования выполнены на ЭВМ «Минск-32».

ЭВМ «Минск-32» является логическим завершением семейства машин «Минск» и наиболее полно отвечает всем требованиям, предъявляемым к серийным ЭВМ широкого применения. Машин второго поколения серии «Минск» можно разбить на 2 группы. К первой группе относятся машины «Минск-2», «Минск-22», «Минск-22М» с базовой машиной «Минск-2». Ко второй группе относятся машины «Минск-23», «Минск-32».

Машины первой группы создавались для решения широкого круга задач научно-технического и плано-экономического характера. Причем, если при разработке первой ЭВМ «Минск-2» в стране имелся достаточный опыт создания ЭВМ для научно-технических расчетов, то структура ЭВМ для экономических расчетов прогнозировалась интуитивно. В связи с этим ЭВМ «Минск-2» состояла из развитого процессора с двоичной арифметикой в режиме с фиксированной и плавающей запятой и минимального набора необходимых устройств ввода-вывода.

Первые же попытки поставить задачи экономического характера показали неэффективность применения для этих целей ЭВМ такой структуры. Выявилась низкая эффективность предложенного комплекта внешних устройств и недостаточная емкость оперативной памяти.

Следующая модель «Минск-22» была выполнена на базе «Минск-2» и частично устранила недостатки предыдущей модели путем расширения емкости МОЗУ до 8192 машинных слов и подключения ряда дополнительных устройств широкой печати, перфокарточного ввода-вывода, магнитных лент и т. д.

Ряд недостатков структуры и логики этих машин был устранен в следующей модели «Минск-22М»: были подключены более производительные механизмы ПЛ-80, МП-1-16, уменьшены габариты машины. Все это позволило на 25—30% повысить эффективность этой машины. Характерными особенностями машин этой группы семейства являлись: простота в эксплуатации и производстве, относительная надежность, дешевизна. Общей чертой машин этой группы является их однотипность структуры, конструкции, комплекса элементов. Все они выполнены на базе процессора одной структуры. Эти качества позволили семейству машин «Минск» стать самыми распространенными у нас в стране среди машин своего класса. Основное их применение — это вычислительные центры НИИ, заводов, отраслей, предприятий.

Вторая половина 60-х годов характерна тем, что в это время стали создаваться различные сложные информационно-логические системы. Был создан ряд систем и на базе ЭВМ первой группы системы по обработке телеметрической информации, системы обработки информации, системы АСУП («Львов», «Опыт») и т. д. Работы в этом направлении показали, что машины

первой группы в дальнейшем не смогут эффективно применяться в народном хозяйстве в силу ряда серьезных недостатков. Разработанные как универсальные процессоры, они не смогли выполнять сложные функции, возлагаемые на базовые ЭВМ системы. Наиболее существенными недостатками ЭВМ первой группы серии «Минск» являются:

- отсутствие возможности увеличения емкости оперативной памяти свыше 8192 машинных слов;
- отсутствие универсальной системы связи с внешними устройствами;
- наличие в качестве алфавитно-цифровой единицы 6-битного символа, отсутствие десятичной арифметики и возможности работы с переменной длиной слова;
- отсутствие операционной системы управления работой ЭВМ (системы типа «Диспетчер»).

Машины второй группы «Минск-23» и «Минск-32» должны были выполнить те функции, которые оказались не под силу машинам первой группы. Анализ экономических задач, поставленных на машинах «Минск-22», показал низкую производительность каналов связи с внешними устройствами. Характерной особенностью этих задач является их высокая информационность в трех основных трактах: ввода, вывода, обмена память-лента.

В ЭВМ «Минск-22» все время ввода и обмена память-лента процессор простаивает, а во время вывода за счет прерывания частично производится совмещение работы процессора и ВНУ. Кроме того, очень большой процент времени счета в машинах первой группы в связи с отсутствием десятичной арифметики тратился на перевод из двоичной системы в десятичную и наоборот.

ЭВМ «Минск-23» явилась первой машиной этой группы семейства, предназначенной специально для обработки экономической информации. Машина обладала следующими особенностями:

- десятичной системой счисления;
- возможностью обработки полей переменной длины;
- наличием СУС ВнУ, позволяющей подключать до 64 различных внешних устройств;
- развитой логикой процессора, ориентированной на обработку экономической информации (сортировка, обмен, редактирование и т. д.).

Все эти качества позволили создать на базе этой машины достаточно производительные системы. В частности, система Аэрофлота по резервированию билетов уже в первый год эксплуатации чела значительный экономический эффект. Был создан ряд отраслевых и производственных систем, базовой ЭВМ в которых была машина «Минск-23».

Однако ЭВМ «Минск-23» не смогла полностью заменить машины первой группы в силу того, что последние были к этому моменту очень широко распространены у нас в стране, а «Минск-23» была ориентирована только для решения экономических задач. В связи с этим машина «Минск-22М», несмотря на свои очевидные недостатки, продолжала выпускаться нашей промышленностью.

Машина «Минск-32» предназначалась для замены морально устаревшей «Минск-22М» в серийном производстве, и в настоящее время она выпускается вместо «Минск-22М». Она лишена недостатков всех своих предшественников и имеет ряд особенностей, выгодно отличающих ее от других моделей семейства «Минск»:

- емкость оперативной памяти до 65 536 37-разрядных машинных слов;
- многопрограммность — машина может одновременно решать до четырех программ;
- наличие СУС ВнУ, в основе которой лежит двухканальная система связи, позволяющая подключать в общей сложности до 136 различных внешних устройств;
- программно-аппаратурная система совместимости, позволяющая решать без дополнительного перепрограммирования все задачи машин «Минск-2», «Минск-22», «Минск-22М»;
- наличие двоичной и десятичной систем счисления, а также возможности адресации как машинного слова, так и алфавитно-цифрового символа, позволяющего решать широкий круг задач от научно-технического до планово-экономического характера;
- возможность объединения нескольких машин «Минск-32 в единый вычислительный комплекс — систему однородных ЭВМ, позволяющую гибко изменять производительность вычислительного комплекса;
- наличие системы «Диспетчер», управляющей работой всего программно-аппаратурного комплекса.

Эта модель наиболее полно отвечает требованиям, предъявляемым к крупносерийным ЭВМ широкого применения.

Машины второй группы семейства «Минск» имеют общую технологическую основу и, в частности, сборно-разборную конструкцию, однотипные внешние устройства. Для обеих групп семейства машин второго поколения «Минск» общим является использование одного и того же комплекса импульсно-потенциальной систем элементов 250 кгц. Специализированный комплекс 600 кгц для процессора ЭВМ «Минск-32» выполнен также на той же конструктивной базе и с использованием почти тех же комплектующих, что и в комплексе 250 кгц. Такая логически-конструктивная преемственность между отдельными моделями групп, а в некоторых вопросах и между группами семейства позволила поддерживать серийное производство этих машин на высоком количественном качественном уровнях.

1.2. НАЗНАЧЕНИЕ И ТЕХНИЧЕСКИЕ ДАННЫЕ «МИНСК 32»

Многопрограммная вычислительная машина «Минск-32» предназначена для решения широкого круга научно-технических планово-экономических задач. В области науки и техники машин может решать следующие задачи: матричные операции, решение системы дифференциальных и алгебраических уравнений, расчет экономических и магнитных полей, расчет энергетических сетей; расчет электроприборов, исследование процессов ядерной физики, расчет сложных механизмов, конструкций и некоторые другие задачи, требующие больших объемов оперативной памяти арифметики с фиксированной и плавающей запятой.

В области экономики, планирования, статистики, управления промышленными предприятиями машина может решать следующие задачи: начисление заработной платы и получение платежной ведомости, составление бухгалтерских сводов, учет материалов, оперативно-производственное планирование, составление сводных спецификаций, планирование перевозок и грузооборота, статистический анализ, информационно-логические задачи, требующие адресации с точностью до символа, десятичной арифметики и больших объемов внешних накопителей.

Технические данные процессора

Процессор предназначен для организации автоматической работы по вводу информации с различных носителей, выполнения арифметических и логических операций, вывода информации на устройства различных носителей, а также синхронизации работы полного комплекса ЭВМ в течение вычислительного процесса.

Представление информации — в двоичном коде с фиксированной и плавающей запятой, алфавитно-цифровое согласно ГОСТу 10859—64 (приложение 3).

Информационной единицей является 37-разрядное двоичное слово или 7-разрядный информационный символ. Разрядность двоичных чисел с фиксированной запятой — 36 разрядов и 1 разряд знака. Разрядность двоичных чисел с плавающей запятой — 28 разрядов мантииссы и 1 разряд знака, 6 разрядов порядка и 1 разряд знака. Разрядность десятичных чисел — 9 десятичных разрядов и 1 разряд знака.

Диапазон двоичных чисел:

— с фиксированной запятой от -1 до $+1$;

— с плавающей запятой от $\pm 2^{+63}$ до $\pm 2^{-64}$ или от $\pm 10^{+19}$ до $\pm 10^{-19}$.

Диапазон десятичных чисел от $\pm 10^{+9}$ до $+10^{-9}$.

Система команд — одно-двухадресная.

Обеспечивается многопрограммная работа четырех рабочих и восьми обслуживающих программ.

Количество индекс-регистров для каждой программы — произвольное.

Количество базисов на каждую программу — 4.

Точность установления границ защиты памяти — 512 машинных слов.

Время выполнения основных операций:

а) операции над двоичными 37-разрядными числами:

— сложение двух чисел $15 \div 40$ мксек, — обмен двух чисел $35 \div 40$ мксек,

— пересылка числа $25 \div 30$ мксек,

— умножение двух чисел $15 \div 130$ мксек,

— передача управления $10 \div 25$ мксек;

б) операции десятичной арифметики над двумя 10 разрядными числами:

- сложение (вычитание) $20 \div 35$ мксек,
- умножение $20 \div 620$ мксек.

Обмен информацией между процессором и внешними устройствами производится 8-разрядным символом (7 разрядов информационных и 1 контрольный для проверки на четность) Имеются два канала связи с внешними устройствами: один мультиплексный и один селекторный.

Работа процессора совмещается с работой внешних устройств (ВнУ) по принципу приостановок. Одновременно могут работать все устройства мультиплексного канала основного комплекта и одно устройство селекторного канала. Общее количество одновременно работающих внешних устройств определяется исходя из пропускной способности каналов. Максимальная пропускная частота мультиплексного канала 66 кГц. Максимальная частота селекторного канала 200 кГц (без работы устройств мультиплексного канала) и 100 кГц при одновременной работе с устройствами мультиплексного канала.

К мультиплексному каналу может быть подсоединено непосредственно 11 внешних устройств (7 устройств основного комплекта и 4 дополнительных) или 104 устройства с помощью трех специальных канальных коммутаторов (7 устройств основного комплекта, 1—непосредственно к группе основного комплекта 96 с помощью коммутаторов, по 32 устройства к каждому коммутатору).

К селекторному каналу может быть подсоединено до 32 устройств с помощью четырех групповых коммутаторов, по 8 устройств к каждому.

Магнитное оперативное запоминающее устройство имеет следующие параметры:

- время обращения 5 мксек;
- разрядность 38 двоичных разрядов (37 информационных, 1 контрольный);
- емкость 16 384 машинных слова;
- возможность наращивания емкости до 65 536 машинных слов блоками емкостью по 16 384 машинных слова (16К); где (К = 1024 машинных слова).

Электронный датчик времени (ЭДВ) работает с частотой 50 Гц

Технические данные внешнего оборудования

Устройство запоминающее на магнитной ленте ЗУМЛ-67 используется в качестве внешнего накопителя больших объемов двоичной информации. Оно имеет следующие параметры:

- емкость одного лентопротяжного механизма $11 \cdot \pm 10^6$ символов;
- количество механизмов основного комплекта 5;
- максимальное количество механизмов на одно устройство управления 8;
- скорость передачи 64 000 симв/сек;
- длина зоны переменная;
- тип лентопротяжного механизма НМЛ-67.

Устройство ввода перфокарточное УВвК-600М предназначено для восприятия информации, нанесенной на перфокарты, преобразования ее в электрические сигналы и передачи их в вычислительную машину.

Его параметры:

- техническая скорость $10 \pm 10\%$ карт/сек;
- режим работы старт-стопный или непрерывный;
- носитель информации 80- или 45-колонные перфокарты по ГОСТу 6198—64;
- емкость приемного и подающего кармана 1000 карт; представление информации по ГОСТу 10859—64 по колонно или любое другое с программным перекодированием.

Устройство вывода перфокарточное УВК-23 предназначено для преобразования информации, выводимой из вычислительной машины, в пробивки на 80-колонные перфокарты.

Его параметры:

- техническая скорость $1,6 \div 2$ карт/сек;
- носитель информации 80-колонные перфокарты;
- представление информации на носителе (ГОСТ 10859—64) или любое другое с программным перекодированием.

Устройство ввода перфоленточное УВВЛ-23 предназначено для восприятия информации, нанесенной на многоканальные перфоленты согласно ГОСТу 10860—64, преобразования ее в электрические сигналы и передачи этих сигналов в вычислительную машину.

Его параметры:

- техническая скорость 1500 строк/сек;
- носитель информации 5÷8-дорожковая перфолента;
- представление информации по ГОСТу 10859—64 и любое другое с программным перекодированием.

Устройство вывода перфоленточное УВЛ-23 предназначено для преобразования выводимой из ЭВМ информации в пробивки на многоканальные перфоленты.

Его параметры:

- максимальная скорость перфорации 80 строк/сек;
- носитель информации 5÷8-дорожковая перфолента;
- представление информации по ГОСТу 10859—64 и любое другое с программным перекодированием.

Устройство вывода печатающее УПч-23 предназначено для преобразования выводимой из вычислительной машины информации в систему печатных знаков, наносимых на бумагу. Печать осуществляется в виде цифр, букв русского и латинского алфавитов и специальных знаков.

Его параметры:

- техническая скорость печати 400 строк/мин;
- количество разрядов в строке 128;
- представление информации первые 78 символов по ГОСТу 10859—64.

Устройство ввода-вывода на базе пишущей машинки УМП-23К предназначено для восприятия информации, вводимой оператором с клавиатуры, преобразования ее в электрические сигналы и передачи их в вычислительную машину, а также для преобразования выводимой из вычислительной машины информации в систему знаков, наносимых на бумагу.

Его параметры:

- техническая скорость печати 10 зн/сек;
- представление информации: первые 92 символа по ГОСТу 10859—64 — специальные символы.

Устройство подготовки данных на перфокартах УПДК предназначено для нанесения алфавитно-цифровой информации на 80-колонные перфокарты по ГОСТу 6198—64, в десятичном коде по ГОСТу 10859—64, а также для контроля ранее отперфорированных перфокарт.

Его параметры:

- носитель информации 80-колонные перфокарты;
- представление информации 78 первых символов по ГОСТу 10859—64.

Устройство подготовки данных на перфоленте КСУ-2 предназначено для нанесения алфавитно-цифровой информации на перфоленту в соответствии с разделом II ГОСТа 10859—64 в 7- и 4-разрядном кодах.

Его параметры:

- носитель информации 5, 6, 7, 8-дорожковая перфолента
- представление информации: первые 92 символа по ГОСТу 10859-64.

1.3. СОСТАВ МАШИНЫ

В состав основного комплекта машины «Минск-32» входят следующие устройства:

- а) процессор, содержащий:
- | | |
|--|--------|
| центральный пульт управления ЦПУ | —1 шт. |
| устройство центрального управления ЦУ | —1 » |
| арифметическое устройство АУ | —1 » |
| устройство обмена УО | —1 » |
| магнитное оперативное запоминающее устройство МОЗУ | —1 » |
- б) устройство УМП-23К с пишущей машинкой «Копсул-254»
- | | |
|--------------|-------|
| «Копсул-254» | — 1 » |
|--------------|-------|
- в) устройство ввода перфокарточное УВВК-600М
- | | |
|-----------|-------|
| УВВК-600М | — 1 » |
|-----------|-------|

- г) устройство вывода перфокарточное УВК-23 с механизмом ПЭМ-80 — 1 »
 - д) устройство ввода перфоленточное УВВЛ-23 с механизмом FS — 1500 — 1 »
 - е) устройство вывода перфоленточное УВЛ с перфоратором ПЛ-80/8 — 2 шт.
 - ж) устройство вывода печатающее УПч-23 с механизмом АЦПУ-128-2 — 1 »
 - з) устройство запоминающее на магнитной ленте ЗУМЛ-67 с пятью лентопротяжными механизмами НМЛ-67 — 1 »
- Кроме того, в состав машины входят:
- устройство подготовки данных на перфокартах УПДК — 1 »
 - контрольно-считывающее устройство КСУ-2 — 2 »
 - комплект ЗИПа — 1 компл.
 - комплект эксплуатационных документов ЭД — 1 »
 - монтажный комплект — 1 »
 - комплект тары — 1 »
- комплект приборов для проверки полупроводниковых ячеек — 1 »
- система математического обеспечения — 1 »

Машина может быть расширена заказчиком произвольным набором внешних устройств, алгоритм связи которых соответствует универсальному принципу связи внешних устройств ЭВМ «Минск-32».

Имеется конструктивная и логическая возможность увеличения оперативной памяти основного комплекта машины до емкости 65 536 машинных слов.

На рис. 1 приведена структурная схема ЭВМ «Минск-32». В машине циркулируют два информационных потока, объединяющих работу устройств процессора и двух каналов, связывающих внешние устройства с процессором. Такое количество информационных потоков определяется наличием в машине двух основных информационных единиц: 37-разрядного машинного слова и 7-разрядного символа. 37-разрядный тракт передает рабочую информацию из МОЗУ в рабочие регистры устройства АУ (P1, P2 и CM) и принимает из этих же регистров

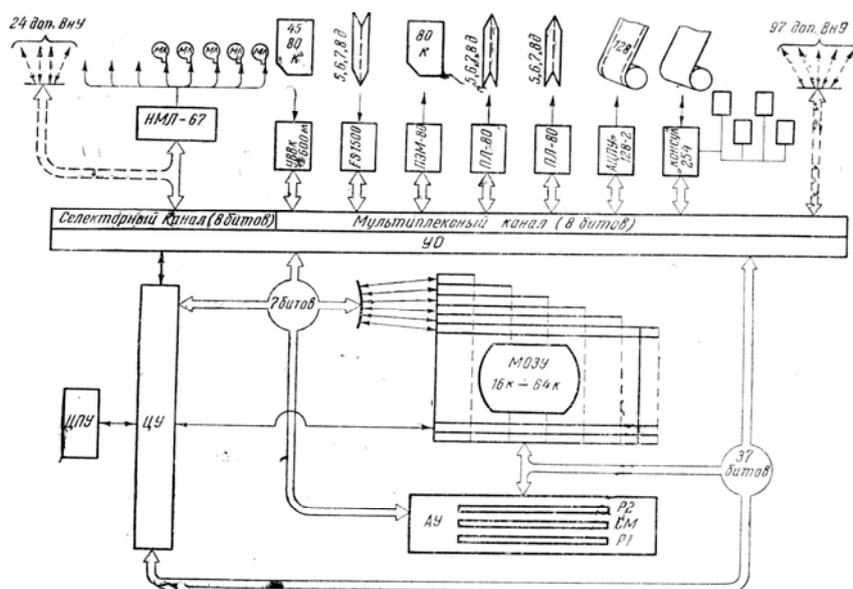


Рис.1 Структурная схема ЭВМ «Минск-32».

информацию для записи в МОЗУ. Этот же 37-разрядный тракт связывает между собой устройства ЦУ, АУ, УО, МОЗУ для передачи информации управления. 7-разрядный тракт служит для

передачи символьной информации в различные устройства процессора. Этот же тракт с помощью специальной маски связан с оперативной памятью. Маска организует посимвольную работу с оперативной памятью: любой символ памяти может быть занесен в тракт, и в любую часть памяти, кратную 7 битам, можно записать символ, принятый из тракта. Устройство ЦУ с помощью специальных цепей управления синхронизирует работу устройств ЦПУ, АУ, УО, МОЗУ.

С помощью мультиплексного канала устройства обмена процессор, связанный с внешними устройствами основных носителей машины, можно подключить еще к 97 внешним устройствам различного назначения.

Селекторный канал в основном комплекте организует работу с пятью лентопротяжными механизмами устройства магнитной ленты. Однако структура этого канала позволяет дополнительно подключать еще до 27 устройств типа НМЛ и НМБ.

В основе работы каналов связи лежит 8-битный символ (7 информационных, 1 контрольный разряды). При обмене информацией 8-битный тракт обмена двух каналов устройства УО подключается к 7-битному информационному тракту процессора.

1.4. ОСНОВНЫЕ ОСОБЕННОСТИ ЭВМ «МИНСК-32»

Представление информации

Как уже упоминалось, в ЭВМ «Минск-32» в качестве информационных единиц используются 37-разрядное машинное слово и 7-разрядный символ. Машинное слово предназначено для хранения командной информации (команд, индексных ячеек, базисных ячеек, информационных ячеек программного уровня, управляющих слов) и рабочей информации (операндов для работы в двоичной системе с фиксированной и плавающей запятой и десятичных операндов). Символ используется для работы с полями переменной длины и при обмене информацией с внешними устройствами.

Основной информационной единицей в ЭВМ является 37-разрядное машинное слово. Рабочие регистры АУ и ячейки памяти имеют 37 двоичных разрядов. За одно обращение к памяти записывается или считывается 37-разрядное машинное слово.



Рис. 2. Представление информации с фиксированной запятой.

На рис. 2 показано распределение весов двоичных разрядов и их назначение при обработке двоичной информации с фиксированной запятой. Крайний левый разряд хранит значение знака. Запятая фиксирована после знакового разряда перед первым, который является старшим по значению. Младшим является крайний, справа, 36-й разряд.

Для изображения двоичных чисел с плавающей запятой машинное слово разбивается на 2 части (рис. 3).

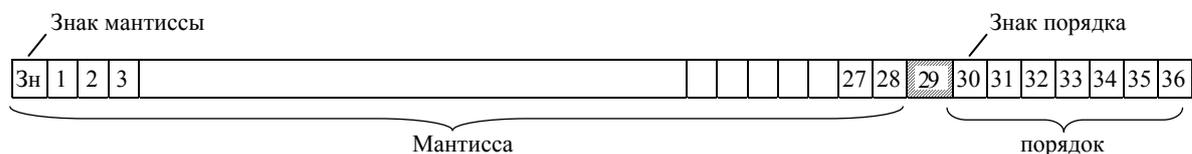


Рис. 3. Представление информации с плавающей запятой.

Значение мантиссы записывается в первых 28 разрядах. Знак мантиссы хранится в знаковом разряде машинного слова. Порядок записывается в разрядах 31÷36. Знак порядка хранится в 30-м разряде машинного слова. 29-й разряд машинного слова является вспомогательным и в процессе выполнения операции используется для сохранения точности результата.

При работе над числами в десятичной системе для изображения одной десятичной цифры в коде 8421 используется 4 двоичных разряда. Знаковый разряд машинного слова предназначен для хранения знака десятичного числа, 36 разрядов — для изображения девяти десятичных цифр (36 разрядов разбиваются на 9 тетрад). Старшая тетрада (1÷4-й разряды) изображает старшую десятичную цифру, младшая девятая тетрада (33÷36-й разряды) изображает младшую десятичную цифру. Десятичные числа рассматриваются как целые, и запятая фиксируется программно.

На рис. 8 изображена одноадресная команда. Отличием ее от двухадресной является наличие определителя на месте первого адреса A1. Значение определителя используется в основном для различия разновидностей команд одной группы. К таким группам можно отнести команды индексной арифметики, работы с управляющей областью памяти, команды работы с внешними устройствами и т. д.

Адресация

Несмотря на наличие в машине двух видов представления информации: машинного слова и символа, при обращении к памяти адресация сводится к заданию адреса машинного слова всегда и дополнительно номера символа в машинном слове во втором случае.

Адресация МОЗУ двоичная. Для задания максимального двоичного адреса (65 536) необходимо иметь 16 битов. Поэтому все регистры машины, обрабатывающие или запоминающие адресную информацию, имеют 16 разрядов (счетчик адреса команд, регистр адреса обращения к МОЗУ и т. д.) или имеют несколько 16-разрядных частей (сумматор АУ при обработке адресной части команды разбивается на две независимые 16-разрядные части, регистр управляющего слова УО состоит также из двух 16-разрядных частей).

Декремент адресной части команды имеет всего 11 битов и позволяет осуществлять непрерывную адресацию внутри массива в 2048 машинных слов. Для получения полного 16-битного адреса обращения к памяти адрес модифицируется содержанием базисной ячейки всегда и индексной ячейки – при необходимости.

Базисные константы всегда содержат 16 битов. На рис. 9 изображены 4 базиса B0, B1, B2, B3, расположенные в двух машинных словах любого программного уровня управляющей области памяти. В каждой программе используются базисы, хранящиеся в ячейках уровня, или же в любой ячейке памяти при использовании команд со специальным базированием. Адрес базисной ячейки складывается из номера уровня данной программы и номера базисной ячейки, указанной в разрядах b_1 , b_2 адресов A1, A2 команды. Любой из двух декрементов a_1 и a_2 команды может модифицироваться любым из четырех базисов.

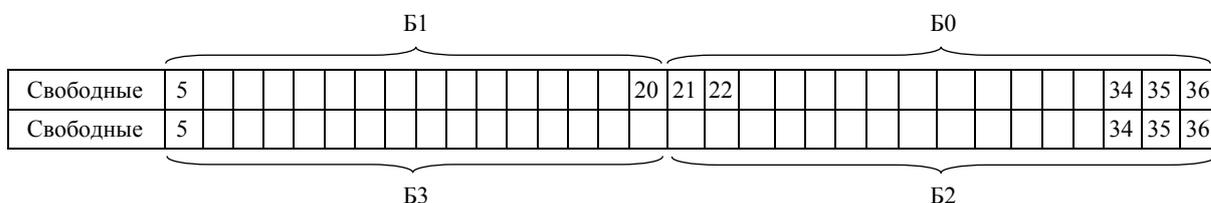


Рис. 9. Расположение базисных констант в ячейках уровня.

Процесс базирования сводится к сложению декремента a_1 или a_2 с содержимым любой базисной ячейки с номером b_1 или b_2 .

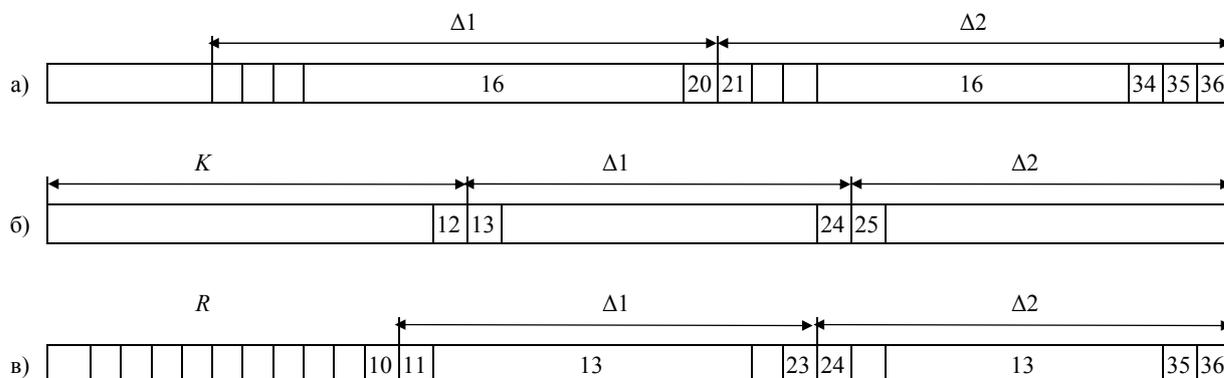


Рис. 10. Структура индексных ячеек.

Структура индексных ячеек приведена на рис. 10. В связи с наличием в ЭВМ трех режимов: режима работы «Минск-32» и режимов совместимости «Минск-22М» и «Минск-22» имеются 3 типа индексных ячеек. Индексная ячейка для режима «Минск 32» (рис. 10, а) состоит из двух 16-разрядных приращений. Приращение Δ_1 модифицирует адрес A1, Δ_2 модифицирует адрес A2.

Разряды $3n \div 4$ в индексной ячейке свободны.

В режиме совместимости для программ «Минск-22» (рис. 10,б) используется второй тип индексной ячейки с 12-битными приращениями Δ_1 и Δ_2 , позволяющими модифицировать адрес внутри массивов в 4096 слов. Старшие 13 разрядов индексной ячейки служат для задания константы повторения циклов K . Максимальное число циклов, задаваемое K , равно 8192.

В режиме совместимости с программами «Минск-22М» (рис 10, в) используется третий тип индексных ячеек с 13-битными приращениями Δ_1 и Δ_2 и 11-битной константой K .

Модификация адресов с помощью индексной ячейки производится только в режиме сложения.

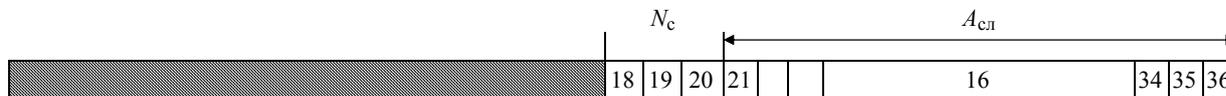


Рис. 11. структура адреса символа.

Адрес для выборки любого 7-битного символа (рис. 11) состоит из двух частей: адреса машинного слова $A_{сл}$, 16 битов которого позволяют указать любое слово внутри массива в 65 536 машинных слов, и номера символа внутри машинного слова N_c , который указывает интересующий символ внутри слова $A_{сл}$. Номер символа N_c принимает значение от 0 до 4 (рис. 5). Для модификации символьных полей используется индексная ячейка, приведенная на рис. 12, где $\Delta_{см}$ — шаг наращивания символьного

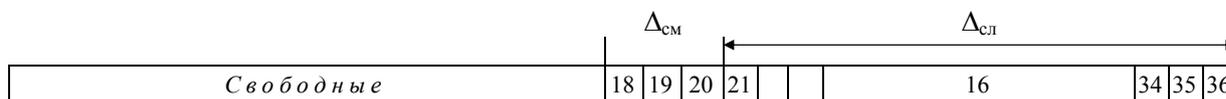


Рис. 12. Индексная ячейка для модификации адресов символьных полей.

поля, $\Delta_{сл}$ — приращение адреса слова. Если в результате индексирования символьной части индекс-ячейкой значение N_c окажется ≥ 5 , происходит автоматическая коррекция N_c и прибавление «+1» к $A_{сл}$. Модификация полей символов производится в сторону увеличения.

1.5. ОСНОВНЫЕ РЕЖИМЫ РАБОТЫ ПРОЦЕССОРА

Выполнение команды представляет собой некоторую последовательность действий над командными и информационными словами. Результатом каждого такого действия является новое состояние регистров и управляющих элементов процессора. Как правило, все действия по обработке информации разбиваются на ряд однотипных микродействий. Каждое такое микродействие выполняется по импульсу задающего генератора машины.

За время одного цикла работы МОЗУ вырабатываются 3 импульса задающего генератора (ГИ1, ГИЗ, ГИ5). Интервал времени, в течение которого вырабатываются эти импульсы, составляет один машинный такт. Выполнение команды — это выполнение определенного количества машинных тактов, следующих в последовательности, характерной для каждой команды. Название такта согласуется с его основным назначением. В каждом такте выполняются типовые действия (формирование и выдача адреса, прием информации в регистры и т. д.) и специфичные действия для каждой команды.

Из серии импульсов задающего генератора в каждом такте формируются тактовые импульсы. Количество этих импульсов колеблется от 3 до 6. Исключение составляет такт выполнения арифметических действий, где количество импульсов может быть больше шести. Название тактовых импульсов состоит из сокращенного названия такта и номера импульса. Например, название такта чтения команды — такт K . В нем вырабатываются импульсы: КИ1, КИ2 и т. д., в такте записи информации в управляющую область памяти УЗП вырабатываются импульсы: УЗПИ1, УЗПИ2 и т. д. По тактовому импульсу, соответствующему по времени импульсу ГИ5, происходит переключение тактов.

Следует различать такты ЦУ, АУ и такты приостановок.

Такты ЦУ обеспечивают выполнение чтения кода команды, формирование действительных адресов операндов, чтение операндов, различные действия над операндами и адресами и запись результатов, Такт АУ обеспечивает выполнение арифметических действий над числами с

фиксированной и плавающей запятой и над десятичными числами.

Такты приостановок обеспечивают обмен информацией между процессором и внешними устройствами по их запросам.

Последовательность выполняемых тактов ЦУ может быть прервана по запросу внешнего устройства для обмена информацией с МОЗУ. Такт арифметический не имеет обращения к памяти и может выполняться параллельно с тактами приостановок. Такты приостановок имеют самый высокий приоритет. Очередной такт ЦУ находится в этом случае в состоянии ожидания и выполняется по окончании приоритетных тактов.

Характеристика тактов

Такт К предназначен для чтения кода команды и размещения его в рабочих регистрах процессора. В такте производится также анализ результата предыдущей операции, анализ сбоев предыдущей операции, выдача адресов переходов в счетчик адреса команд (СчАК) и расшифровка нового кода операции.

Такт ИНД предназначен только для индексирования адресов команды. В такте совмещается чтение содержимого индексной ячейки и сложение его с относительными адресами команды.

Такт ИНД не выполняется, если номер индекса не указан (равен нулю) или если выполняемая команда относится к группе команд, где индексная ячейка является одним из операндов.

Такты базирования адресов Б1 и Б2 присутствуют всегда и являются заключительным этапом формирования действительных адресов операндов. В зависимости от адресности выполняется один из тактов Б1 или Б2 или оба такта. В тактах Б1 и Б2 совмещается чтение базисов и сложение их с относительными адресами. Возможно выполнение тактов Б1 и Б2 одновременно, если базисы, указанные для адресов, расположены в одной ячейке МОЗУ. При этом действия над адресами А1 и А2 производятся независимо друг от друга. В тактах Б1 и Б2 может производиться операция, если этими тактами закапчивается операция и ряд подготовительных действий к выполнению операции.

Основное назначение тактов 1ЧТ и 2ЧТ — чтение операндов по адресам А1 и А2 команды. Но часто в этих тактах выполняются подготовительные действия к арифметическому такту и действия по выполнению операции.

Для команд со специальным базированием такт 1ЧТ заменяет такт Б2, в котором выполняются действия, аналогичные действиям в такте Б2.

Такт А предназначен для выполнения арифметических действий. Длительность такта А переменная и определяется как типом операции, так и состоянием операндов. Такт А не выполняется, если один из операндов равен нулю.

Такт 23П предназначен для записи результатов по адресу А2 (в некоторых случаях и по адресу А1). Кроме того, для многих команд в такте 23П выполняются действия над операндами, т. е. с записью результата частично совмещается выполнение операции.

Такты УЧТ и УЗП предназначены для обращения к ячейкам управляющей области памяти. В тактах формируется адрес, состоящий из номера уровня, номера ячейки уровня и фиксированной части адреса, аппаратно реализованной в процессоре и позволяющей обращаться только к ячейкам УОП. При обращении к МОЗУ в этих тактах адрес не проверяется на соответствие границам защиты и, таким образом, только в этих тактах разрешено обращение к УОП в рабочих программах. В тактах УЧТ и УЗП, кроме чтения и записи информации, могут производиться различные действия по выполнению операции.

Такты ИЧТ и ИЗП предназначены для обращения к ячейкам МОЗУ, где хранятся индексные приращения, над которыми производятся арифметические и логические действия. В этих тактах адрес индексной ячейки формируется из содержимого рабочего регистра базиса индекса, хранящегося в этом регистре в течение времени выполнения программы или части ее, и номера индекса, указанного в команде.

Такты ОП1 и ОП2 предназначены для выполнения отдельных нестандартных действий при выполнении операций, когда, например, производится длительная обработка операндов или используются кодовые шины передачи числа (КШЧ) для внутренних передач из регистра в регистр. Такты ОП1 и ОП2 не имеют обращения к памяти.

Такты приостановок 1П, 2П, 3П, 2ПБ предназначены для осуществления обмена между памятью процессора и любым ВнУ. Для выполнения тактов приостановок по запросам медленных устройств выполняется такт чтения управляющего слова — такт 1П, затем такт непосредственного

обмена информацией МОЗУ и внешнего устройства по адресу, указанному в управляющем слове, — такт 2П и запись модифицированного УС в МОЗУ — такт 3П. Для обмена информацией МОЗУ и ВнУ селекторного канала выполняется такт быстрой приостановки — 2ПБ.

Выборка команды

Выборка команды — это чтение кода команды и формирование действительных адресов операндов. Во время выборки команды производятся следующие типовые действия:

- чтение кода команды,
- индексирование адресной части команды,
- базирование адресов команды.

Чтение кода команды производится по текущему адресу программы, хранящемуся в СчАКЕ, или по адресу перехода, хранящемуся в сумматоре (См) или в регистре адреса ЦУ.

Прочитанная информация с кодовых шин принимается по частям в различные регистры ЦУ и АУ (рис. 13).

Код операции принимается в регистр дешифратора операций и хранится там до окончания выполнения операции. Номер индексного поля принимается в регистр номера индекса и хранится там до окончания индексирования адресной части команды. В командах обработки индексных полей номер индекса является частью адреса операнда и хранится в регистре до конца выполнения операции.

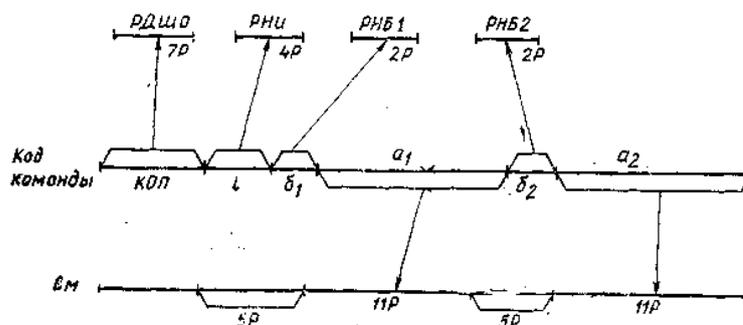


Рис. 13. Схема выборки команды.

Номера базисов адресов принимаются в 2-разрядные регистры базисов и хранятся там до окончания операции. Для команд с определителем в регистр базиса первого адреса принимаются различные признаки (например, признак перехода по нулю или единице какого-то указателя), записанные в команде в разрядах первого базиса.

Относительные 11-разрядные адреса команды принимаются в См и располагаются таким образом, чтобы обеспечить формирование полноразрядных действительных адресов операндов (16 битов). Разряды второго адреса принимаются в соответствующие разряды См, а разряды первого адреса принимаются в См со сдвигом на 3 разряда.

Индексирование адресной части команды заключается в сложении ее с содержимым ячейки индекса, номер которой указан в команде.

При этом складываются одинаково адреса команды и определитель команды, занимающий разряды первого адреса (исключение составляют команды работы с ВнУ, где определитель не индексируется). Адрес индексного поля формируется из базиса индекса (12 старших разрядов адреса) и номера индекса, указанного в команде (4 младших разряда адреса). Базис индекса является величиной постоянной для каждой текущей части программы и хранится в рабочем регистре ЦУ—12-разрядном регистре базиса индекса. Прочитанная по указанному адресу информация принимается в предварительно сброшенный регистр Р1 и на счетные входы триггеров См, в результате чего в См образуется поразрядная сумма кода относительных адресов и индексных приращений. По сигналу образования переносов заканчивается индексирование.

Прием информации в См с кодовых шин числа по счетному входу позволяет совместить в одном машинном такте чтение индексного поля и сложение его с адресной частью команды. Для программ, написанных на языке машины «Минск-22М», индексная ячейка имеет иную структуру — 13-разрядные индексные приращения. Для сложения с разрядами первого адреса разряды кодовых шин приращения первого адреса принимаются со сдвигом на 3 разряда (рис. 14). Для

имитации вычитания индексных приращений из адресов в программах «Минск-22М» предусматривается игнорирование переносов из тринадцатых разрядов сумм. Индексирование для программ «Минск-22» не рассматривается в силу аналогичности с «Минск-22М».

Базирование адресов команды заключается в сложении проиндексированных (или неиндексированных) относительных адресов с 16-разрядными базисными адресами, номера которых указаны в команде.

Для двухадресных команд, как правило, базируются оба адреса. Базирование не производится в том случае, когда адрес не участвует в операции, например, не базируется второй адрес в арифметических и логических операциях 3-й модификации или не базируется один из адресов в командах перехода по знаку и нулю, когда известно, по какому из адресов произойдет переход. В командах с определителем базируется один адрес. В безадресных командах, таких, как «Ничего не делать», «Переключить уровень» (некоторые модификации), базирование одного из адресов выполняется автоматически ввиду того, что к моменту переключения машины на такт базирования отсутствует код операции.

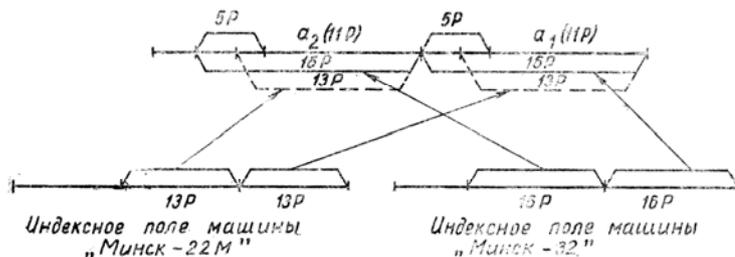


Рис. 14. Схема индексации адресов.

Если базисы в коде команды для обоих адресов расположены в одной ячейке, производится базирование двух адресов в одном машинном такте.

Для каждой программы базисы хранятся в ячейках 0 и 1 уровня этой программы в виде 16-разрядных чисел, по два числа в каждой ячейке. Адрес базиса отыскивается по номеру уровня выполняемой программы, номеру ячейки уровня (0 или 1) и номеру базиса в ячейке. Номер уровня хранится в рабочем регистре ЦУ — регистре номера уровня. Старший разряд номера базиса определяет номер ячейки в уровне, младший разряд номера базиса указывает, какой из двух базисов, находящихся в этой ячейке, используется.

Прочитанный базис принимается с КШЧ на счетные входы разрядов См для образования поразрядной суммы и в регистр Р1. Согласно номеру базиса принимаются либо старшие 16 разрядов, либо младшие 16 разрядов прочитанного слова (рис. 15). По сигналу образования переносов заканчивается базирование.

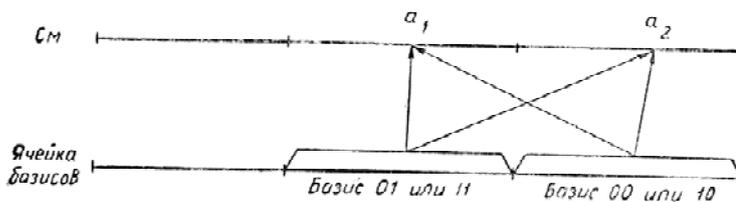


Рис. 15. Схема базирования адресов.

В результате сложения относительных адресов команды с содержимым индексной и базисных ячеек формируются действительные 16-разрядные адреса операндов или переходов.

Во время базирования выполняются некоторые подготовительные действия к выполнению операции — содержимое разрядов См, где хранится определитель, передается в регистр определителя, подготавливается к работе схема защиты, модифицируется адрес команды и т. д. Для ряда команд (переходы по нулю, знаку) тактами базирования завершается выполнение команды.

Особенности выполнения команд

Основная масса команд машины «Минск-32» — это двухадресные команды. Большую часть двухадресных команд составляют команды, выполняемые стандартно. К ним относятся команды

арифметические с фиксированной и плавающей запятой, команды десятичной арифметики и большинство логических команд.

Выполнение стандартной команды состоит из следующих этапов:

- выборки команды;
- чтения операндов;
- действия над операндами;
- записи результата.

В зависимости от особенностей команды отдельные этапы могут отсутствовать. В командах, оперирующих с результатом предыдущего действия, отсутствует чтение одного из операндов, в командах, не требующих записи результата в память, отсутствует такт записи результата. В ряде команд возможно совмещение чтения операнда с выполнением операции. Например, в команде, выполняющей контрольное суммирование, в такте чтения очередного слова суммируемого массива производится сложение этого числа с частичной суммой. Все логические действия производятся также в тактах чтения операндов. Действия над операндами команд стандартного типа выполняются в такте арифметическом (А) или в тактах операции (ОП1 и ОП2). Если один из прочитанных операндов в арифметических командах равен нулю, такт А не выполняется.

Рассмотрим выполнение типовой команды на примере выполнения команды сложения с фиксированной запятой нулевой модификации. Процесс выборки команды описан выше. По окончании выборки команды выполняется чтение операндов. Чтение операнда по адресу А1 команды осуществляется путем выдачи на КША содержимого 5÷20-го разрядов См и пуска МОЗУ на чтение. Прочитанная по этому адресу информация принимается в предварительно «сброшенный» регистр Р1.

Параллельно с приемом в регистр Р1 прочитанная информация принимается в регистр числа МОЗУ и контролируется на соответствие контрольному разряду. Информация с КША принимается в регистр адреса ЦУ и контролируется на соответствие границам защиты программы, в которой выполняется данная команда.

Чтение операнда по адресу А2 команды осуществляется путем выдачи на КША содержимого 21 ÷36-го разрядов См и пуска МОЗУ на чтение. Прочитанная по этому адресу информация принимается в предварительно «сброшенный» См и в регистр числа МОЗУ для контроля на «нечет». С кодовых шин адреса, параллельно с приемом в регистр адреса МОЗУ, адрес А2 принимается в регистр адреса ЦУ (РА), где хранится до выполнения записи результата и контролируется на соответствие границам защиты.

Принятые операнды подготавливаются к выполнению операции сложения, производится инвертирование отрицательного операнда. Местное управление операциями обрабатывает такт арифметического действия, производится сложение содержимого См и регистра Р1.

Результат формируется в См и записывается в МОЗУ по адресу А2 команды путем выдачи на КША из регистра РА и пуска МОЗУ на запись. Записываемая информация выдается из См на КШЧ и принимается в регистр числа МОЗУ, откуда поступает в цепи записи МОЗУ. Перед записью информации в регистре числа формируется контрольный разряд, записываемый вместе со словом. Адрес записи с КША снова принимается в РА и контролируется на соответствие границам защиты.

По окончании операции происходит автоматический переход к выполнению следующей команды, в начале которой результат выполненной команды передается из См в регистр. Р2.

1.6. СИСТЕМА КОМАНД

Система команд машины позволяет обрабатывать информацию с фиксированной длиной, используя двоичную арифметику с фиксированной и плавающей запятой и над целыми числами. Возможна посимвольная обработка алфавитно-цифровой информации, представленной в 7-разрядном коде, в котором для изображения цифр, букв и специальных знаков отводится 7 разрядов. Имеются операции индексной арифметики, операции связи с управляющей областью памяти, операции условных и безусловных переходов.

По характеру применения команды машины можно разделить на три различных класса:

- а) команды, используемые в рабочих программах. Под рабочими программами подразумеваются программы, составленные для решения конкретных задач;
- б) команды, используемые только в программах системы математического обеспечения

машины;

в) команды, используемые только для обеспечения совместимости с ЭВМ «Минск-22М» и «Минск-22». Этот класс команд при программировании не используется, а соответствует определенным командам машины «Минск-22М».

Нет необходимости подробно рассматривать все команды. Перечень их, назначение, диаграмма выполнения и основные данные приведены в приложении 1.

При этом приняты следующие обозначения и сокращения:

- 1) (См) — содержимое сумматора;
- 2) a_1 и a_2 — значения первого и, соответственно, второго адресов кода команды;
- 3) b_1 и b_2 — номера базисов для первого и, соответственно, для второго адресов кода команды;
- 4) (b_1) и (b_2) — содержимое базисов или базисные адреса;
- 5) (bi) — базисный адрес индексов;
- 6) i — номер индекса;
- 7) $A_i = (bi) + i$ — истинный адрес ячейки индекса с номером i ;
- 8) $\Delta_1(i)$ и $\Delta_2(i)$ — константы индексирования для первого и, соответственно, второго адресов из ячейки индекса с номером i ;
- 9) $A_1 = a_1 + (b_1) + \Delta_1(i)$ — истинное значение первого адреса команды;
- 10) $A_2 = a_2 + (b_2) + \Delta_2(i)$ — истинное значение второго адреса команды;
- 11) (A_1) и (A_2) — содержимое ячеек, соответственно, с адресами A_1 и A_2 ;
- 12) $\rightarrow A_2$ — запись результата в ячейку с адресом A_2 .

Время выполнения команд указывается везде с учетом времени выполнения индексирования, которое равняется 5 мксек.

1.7. ОСНОВНЫЕ РЕЖИМЫ ОБРАБОТКИ ЦИФРОВОЙ И АЛФАВИТНО-ЦИФРОВОЙ ИНФОРМАЦИИ

При работе с фиксированной запятой процессор оперирует над значащими цифрами в пределах $1 \cdot 2^{-36} \leq |X| \leq 1 - 2^{-36}$. Все операции, за исключением умножения, выполняются в модифицированном обратном коде. Операция умножения выполняется в дополнительном коде.

Для сохранения истинного знака в разрядную сетку сумматора вводится второй знаковый разряд (3н2 См), фиксирующий верный знак результата. При различных состояниях знаковых разрядов результата («01» или 10) 3н2 См М содержимое 3н2 См М в конце операции передается в 3н1 См М и формируется переполнение разрядной сетки машины (в результате число по модулю больше величины $1 - 2^{-36}$).

Переполнение разрядной сетки машины может произойти при выполнении операций сложения, вычитания и деления. В этом случае устанавливается в «1» указатель «Переполнение», что приводит к прерыванию программы, если при помощи специальной команды указатель «Переполнение» не будет установлен в «0».

В результате выполнения арифметических операций с фиксированной запятой может получиться нулевой результат — число $\pm 000\ 000\ 000\ 000$, которое носит название кода нуля (число по модулю меньше величины $1 \cdot 2^{-36}$), по которому устанавливается в «1» индикатор «Нуль». Для операций сложения, вычитания и вычитания модулей код нуля в машине получается только со знаком «плюс». При любом отрицательном результате устанавливается в единицу индикатор «Минус».

Операции умножения и деления могут выполняться как с округлением, так и без округления, которое устанавливается или блокируется с помощью специальных команд. Если округление разрешено, то при наличии единицы в дополнительном 37-м разряде результата добавляется единица к 36-му разряду См.

В операциях арифметики с плавающей запятой производятся вычисления как над нормализованными числами, так и над ненормализованными.

Диапазон представления чисел с плавающей запятой лежит в пределах $2^{-64} \leq |X| \leq (1 - 1 \cdot 2^{-28}) \cdot 2^{+63}$ в случае нормализованных чисел и $2^{-91} \leq |X| \leq (1 - 1 \cdot 2^{-28}) \cdot 2^{+63}$ в случае ненормализованных чисел.

Все операции над числами с плавающей запятой, за исключением умножения, выполняются в модифицированном обратном коде. Операции умножения выполняются в дополнительном коде. Так же как и при работе над числами с фиксированной запятой, для сохранения истинного знака результата вводится второй знаковый разряд для мантииссы 3н2 См М и второй знаковый разряд

для порядка Зн2 См П.

Действия над порядками и мантиссами производятся отдельно. Поэтому для операций арифметики с плавающей запятой имеется разрыв цепи циклического переноса из 30-го в 29-й разряд, сигнал циклического переноса из Зн2 См М поступает в 29-й разряд, а из Зн2 См П — в 36-й разряд

В операциях арифметики с плавающей запятой могут быть получены следующие результаты:

а. Переполнение положительного порядка.

Порядок результата превышает допустимый диапазон порядка, больше +63. Этот результат выходит за пределы допустимого диапазона представления чисел с плавающей запятой и может считаться положительной или отрицательной бесконечностью, причем ее знак совпадает со знаком мантиссы. В этом случае устанавливается в «1» указатель «Переполение», который вызывает прерывание программ, если не будет установлен в «0» специальной командой.

б. Переполение отрицательного порядка.

Порядок отрицательный и его величина выходит за пределы допустимого диапазона порядка, меньше —63. Такой результат выходит за пределы (меньше минимальной величины) допустимого представления чисел и имеет тот же знак, что и мантисса.

В этом случае машина фиксирует нулевой результат +000 000 000 000 (код нуля).

в. Нулевая мантисса.

Такой результат может получиться только после вычитания равных операндов, умножения на нулевой операнд и деления при нулевом делимом. В этом случае порядок результата гасится и фиксируется код нуля.

При нулевом результате устанавливается в «1» индикатор «Нуль». При выполнении операций сложения, вычитания и вычитания модулей код нуля в машине представляется только со знаком плюс. При выполнении операций сложения порядков и вычитания порядков отрицательное переполение порядков фиксируется так же, как и положительное, индикатором «Переполение». В режиме «Минск-22М» отрицательное переполение порядков в этих операциях фиксируется как код нуля и индикатор «Нуль» устанавливается в «1».

Отрицательный результат всех операций с плавающей запятой фиксируется индикатором «Минус».

Операции сложения, вычитания, умножения и деления над числами с плавающей запятой могут выполняться как с округлением результата, так и без него. При разрешении округления добавляется единица к 29-му разряду, который участвует в реализации арифметических операций с плавающей запятой. При блокировке округления или при сложении чисел разных знаков прибавление единицы к 29-му разряду не производится. После выполнения любой операции, кроме операций сложения и вычитания порядков, в 29-й разряд результата записывается нуль.

Результат арифметики с плавающей запятой может нормализоваться либо оставаться ненормализованным.

Разрешение или блокировка нормализации задается с помощью специальных команд. Блокирование предусматривается только для нормализации влево.

Десятичные числа могут изображать целые числа и числа с фиксированной запятой. В случае чисел с фиксированной запятой машина оперирует над числами, удовлетворяющими условию

$$0 < |X| < 1 - 1 \cdot 10^{-9}.$$

Сложение (вычитание) производится по правилам двоичной арифметики в прямом коде с последующей коррекцией. В этом случае возможно переполение разрядной сетки машины, которое фиксируется указателем «Переполение».

При коде +000 000 000 000 фиксируется нулевой результат и индикатор «Нуль» устанавливается в «1», причем при выполнении операций сложения (вычитания) код нуля представляется в машине только со знаком плюс.

При отрицательном результате десятичной арифметики устанавливается в единицу индикатор «Минус». Если участвующие в операциях числа имеют тетрады со значениями «1010», «1011», «1100», «1101», «1110», «1111», то устанавливается в «1» указатель «Не цифра».

Операция десятичного умножения может выполняться с округлением и без него. При разрешении округления специальной командой к результату произведения добавляется единица в 36-й разряд См, если дополнительный десятичный разряд, которым является старшая тетрада

регистра P2, имеет значение 5 или больше

Операции над символами в машине производят действия над 7-разрядными символами. Причем в операциях участвуют только символы с номерами $0 \div 4$, размещенными соответственно в разрядах машинного слова с 1 по 34.

Чтение и запись символов производится в соответствии с их номером, записанным в команде. Индикаторы «Нуль» и «Минуса устанавливаются в «1», когда результат операции соответственно нулевой и отрицательный.

1.8. СИСТЕМА КОНТРОЛЯ РАБОТЫ ПРОЦЕССОРА

Правильность выполнения программы в процессоре проверяется встроенной аппаратурой контроля. С помощью цепей контроля проверяется частично работа аппаратуры и частично ошибки программы. При возникновении сбоя в работе процессора устанавливается указатель этого сбоя. Всего имеется 10 указателей сбоя семь указателей фиксируют сбой при выполнении программы (при выполнении команд, когда работают ЦУ и АУ), а три указателя фиксируют сбой в тактах обмена информацией с ВнУ. Последние указатели называются указателями сбоев устройства обмена.

В результате возникновения сбоев устройств ЦУ и АУ происходит переключение на программу нулевого уровня (см. гл. 3). Результатом возникновения сбоев процессора в тактах обмена с ВнУ является переключение на программу первого уровня.

Рассмотрим подробнее причины, условия возникновения сбоев и особенности реакции процессора на каждый сбой. Сбой «Подпрограмма» возникает при обращении к занятой подпрограмме. Для проверки занятости подпрограммы, к которой производится обращение, анализируется содержимое первой ячейки подпрограммы. Если подпрограмма свободна, содержимое ее первой ячейки равно нулю. При неравенстве нулю содержимого ячейки выполнение операции не продолжается и производится переключение на нулевой уровень. Переключение на нулевой уровень по этому сбою производится всегда ввиду того, что нет команды перехода по этому указателю.

Сбой «Переполнение» может возникнуть при выполнении арифметических действий в операциях с фиксированной и плавающей запятой и в операциях над десятичными числами. «Переполнение может быть предусмотрено программой. В таких случаях после арифметической команды следует команда перехода по переполнению. В связи с тем, что переполнение может быть нормальным результатом, запись результата при возникновении переполнения не запрещается.

Если переполнение не предусмотрено программой, при его обнаружении производится прерывание на нулевой уровень с запоминанием:

а) в 5-й ячейке прерванного уровня — единичного состояния указателя сбоя «Переполнение», адреса сбойной команды плюс $2(k + 2)$;

б) в 7-й ячейке прерывающего уровня — номера уровня прерванной программы.

Сбой «Недействительный код» возникает при чтении незадействованного кода операции (команды переключения уровней, модифицирования индексных ячеек и модифицирования ячеек уровня с незадействованными комбинациями определителя), при попытке изменить границы защиты программы из рабочей программы и при использовании в рабочей программе команд модифицирования шеек управляющей области памяти, предназначенных только для служебных программ. Сбой «Недействительный код» возникает также при попытке заблокировать прерывание в рабочих программах. При возникновении описываемого сбоя после формирования адресов выполнение операции заканчивается, выбирается и анализируется следующая команда. Если эта команда не является командой перехода по описываемому сбою, происходит прерывание на уровень 0. При этом в ячейке прерываемого уровня запоминается единичное состояние указателя «Недействительный код», адрес сбойной команды плюс 2. В 7-й ячейке уровня прерывающей программы записывается номер прерванного уровня. Сбой «Не цифра» возникает в случае, если тетрады десятичных чисел, над которыми производятся арифметические действия, имеют нецифровые комбинации кодов. Результат операции на нецифровую комбинацию не проверяется.

При возникновении описываемого сбоя операция выполняется до конца, затем выбирается и анализируется следующая команда. Если эта команда не является командой перехода по сбою «Не цифра», происходит переключение на программу нулевого уровня с записью на прерываемый уровень единичного состояния указателя сбоя «Не цифра» и адреса сбойной команды плюс 2. На

уровень 0 в 7-ю ячейку записывается номер прерванного уровня. Сбой «Защита» возникает при несоответствии адреса граничным значениям, заданным для программы, в которой производится обращение по этому адресу.

На соответствие границам «защиты» проверяются адреса команд, адреса чтения операндов, адреса записи результатов и адреса переходов. В тактах УЗП и УЧТ при обращении к ячейкам управляющей области памяти проверка не производится. При возникновении сбоя «Защита» операция выполняется до конца, но результат операции в память не записывается. Если произошел сбой по адресу перехода, этот адрес в счетчик адреса команд не передается. Анализируется так же, как при возникновении других сбоев следующая команда. При сбое адреса перехода читается для анализа команда по адресу перехода. Если анализируемая команда не является командой перехода по описываемому сбою, происходит переключение на программу уровня 0 с записью на прерываемый уровень единичного состояния указателя сбоя «Защита» и адреса сбойной команды плюс 2 (в том числе и команды перехода, в которой неверен адрес перехода). На уровень 0 в 7-ю ячейку записывается номер прерванного уровня.

Сбой «МОЗУ» возникает в случае, если в прочитанной из МОЗУ информации число единиц (включая контрольный разряд) является четным. Если сбой «МОЗУ» возникает в тактах ЦУ, устанавливается причина прерывания на уровень 0 (указатель НЧ). Если сбой «МОЗУ» возникает при чтении управляющего слова (указатель НУС) или при чтении информационного слова при вписывании в него символа, или при чтении из него символа (указатель НИС), то устанавливается причина прерывания на уровень 1. При возникновении сбоя МОЗУ операция выполняется до конца, но запись результата операции запрещается. Исключения составляет команда «Писать символ», где сбой МОЗУ прочитанного слова, в которое вписывается символ, возникает после того, как произойдет запись сбойного слова с вписанным символом. При установлении в единицу указателя НУС или НИС после окончания очередной операции происходит переключение на программу уровня 1. При установлении в единицу указателя НЧ происходит переключение на уровень 0 с записью в 5-ю ячейку прерванного уровня единичного состояния этого указателя и адреса сбойной команды плюс 2. На уровень 0 в 7-ю ячейку записывается номер прерванного уровня.

Указатель сбоя ПУР устанавливается при возникновении любого из вышеперечисленных сбоев процессора во время выполнения программного или автоматического переключения программ. При возникновении сбоя ПУР программное или автоматическое переключение выполняется до конца и затем анализируется следующая команда. Если это не команда перехода по сбою ПУР, то происходит переключение на уровень 0. В этом случае работа производится на трех различных уровнях. Для обнаружения же места сбойной команды необходимо запомнить номер уровня и адрес этой команды. Например, производилось командное переключения с уровня 3 на уровень 5 и в команде произошел сбой.

Производится переключение на уровень 0 с запоминанием в 7-й ячейке номера уровня сбойной команды — уровня 3. Кроме того, в модификациях команд переключения с записью восстанавливаемого начального адреса программы в случае сбоя не допускается восстановления этого адреса. Для обнаружения сбойной команды в начале выполнения команд переключения на уровень 0 в 7-ю ячейку записывается номер уровня выполняемой команды.

Если в команде возникает сбой, то при переключении по этому сбою на уровень 0 запрещается изменять содержимое 7-й ячейки нулевого уровня. Таким образом, сохраняется до анализа сбоя номер уровня сбойной команды. В ячейке 5 уровня сбойной команды запоминается адрес сбойной команды плюс 1. Указатели сбоев на этом уровне не запоминаются. Указатели сбоев запоминаются на том уровне, с которого произошло переключение на уровень 0.

Сбои ПУР в командах переключения и в автоматических переключениях отличить трудно. Единственным признаком является отсутствие каких-либо указателей в 5-й ячейке уровня, с которого произошло переключение на нулевой уровень. Для отыскания места сбоя в этом случае необходимо на нулевом уровне прочитать номер уровня, на этом уровне прочитать адрес команды, проверить две команды (с указанным адресом минус 1 и минус 2). Если эти команды не являются командами переключения, сбой относится к автоматическому переключению. Если же команда с адресом $K-1$ является командой переключения, следует прочитать содержимое 5-й ячейки того уровня, на который переключает эта команда. Наличие хотя бы одной «1» в разрядах указателей сбоя в этой ячейке говорит о том, что сбой произошел в команде переключения,

отсутствие — свидетельствует о принадлежности сбоя к автоматическому переключению.

Особо следует отметить реакцию на сбой в команде «Переключить, выполнить команду и возвратиться». Сбои в этой команде могут возникнуть на этапе переключения на уровень выполняемой команды, во время выполнения команды и при возврате на прежний уровень. При возникновении сбоя на этапах переключения устанавливается указатель сбоя ПУР. При возникновении сбоя внутри выполняемой команды указатель сбоя ПУР не устанавливается.

Если сбой возник на этапе переключения на уровень выполняемой команды или в самой выполняемой команде, возврата на прежний уровень не происходит, анализируется следующая за выполняемой команда. Если это команда перехода по возникшему сбою, то происходит после ее выполнения возврат на прежний уровень. В противном случае происходит переключение на уровень 0 с записью в 7-ю ячейку этого уровня номера уровня команды «Переключить, выполнить команду и возвратиться».

При обмене информацией процессора с ВнУ адрес обмена контролируется на соответствие границам области, в которую разрешено обращение. Границы защиты указаны в управляющем слове каждого устройства. При несоответствии адреса обмена границам защиты устанавливается указатель ЗУС. По этому указателю после окончания текущей операции происходит прерывание на программу уровня 1.

Контроль связи с внешним устройством описан подробно в главе IV, а контроль сдвига информации — в главе II.

1.9. ВЗАИМОДЕЙСТВИЕ УСТРОЙСТВ ПРОЦЕССОРА

Основной информационной магистралью, связывающей все устройства процессора между собой, является тракт 37-кодовых шин (КШЧ). По этому тракту машинные слова передаются в различных направлениях:

- а) из МОЗУ в АУ, ЦУ, УО;
- б) из АУ, ЦУ, УО в МОЗУ;
- в) из одних регистров АУ в другие.

Например, при чтении команды из любой ячейки памяти машинное слово команды выдается на шины КШЧ. Одновременно открываются приемные вентили регистров, фиксирующих соответствующие части команды: код операции попадает в регистр дешифратора операции, номер индекса — в регистр номера индекса, b_1 и b_2 — в регистр номеров базиса и адреса a_1 и a_2 — в соответствующие части См.

Аналогично при чтении любого из операндов код числа выдается из МОЗУ на КШЧ, которые направляют его в тот из регистров АУ, вентили которого в данный момент открыты. Запись в ячейки МОЗУ возможна в принципе из любого регистра процессора, имеющего связь с КШЧ.

Вторым информационным трактом являются внутренние кодовые шины символа (КШС). Они служат для обмена символьной информацией между различными регистрами процессора. Например, передача символа из АУ в МОЗУ производится путем выдачи соответствующих разрядов регистра Р2 на шины КШС, с которых код символа принимается в регистр символа РС устройства УО. Потенциалы регистра поступают вместе с сигналами маски в цепи размещения символа устройства МОЗУ. Производится запись символа в нужную часть выбранного машинного слова.

ГЛАВА II МНОГОФУНКЦИОНАЛЬНОСТЬ АППАРАТУРЫ ПРОЦЕССОРА

Одним из наиболее эффективных способов снижения стоимости машины является организация многофункциональности аппаратуры процессора, совмещение большого количества разнотипных операций на аппаратуре одного устройства.

В настоящей главе рассматривается организация многофункциональности арифметического устройства АУ и блока связи с оперативной памятью МОЗУ.

2.1. АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО

Известно, что арифметическое устройство — это комплекс аппаратуры, предназначенный для выполнения ряда арифметических и логических операций, выраженных в принятой системе счисления. Обычно арифметическое устройство состоит из отдельных узлов, основу которых составляет сумматор [8], [9].

Сложность и конкретное структурное решение каждого типа арифметического устройства (АУ) определяется назначением той ЭВМ, в состав которой входит АУ. В то же самое время арифметическое устройство определяет возможности ЭВМ.

Можно сформулировать обязательные функции, которые выполняются арифметическими устройствами современных ЭВМ широкого применения, это: двоичная арифметика с фиксированной запятой, двоичная арифметика с плавающей запятой, адресная арифметика, десятичная арифметика, спецарифметика.

По характеру использования элементов и узлов, по своей структуре арифметические устройства можно разделить на комбинированные (многофункциональные) и блочные (специализированные).

В специализированных АУ основные арифметические операции выполняются блоками, функционально независимыми друг от друга [8], [9], [10]. Специализация может быть между отдельными блоками одного АУ или же между целыми устройствами. Такая реализация эффективна в специализированных системах, ориентированных на решение определенного класса задач. Примером могут служить специализированные множительные устройства, устройства датчиков случайных чисел, устройства для решения дифференциальных уравнений с помощью алгоритма быстрого преобразования Фурье, ускоряющие в некоторых случаях вычисления в несколько десятков и даже сотен раз [11], [12], [13]. Такие устройства узкоспециализированы. Они нужны не всем заказчикам, поэтому должны быть выполнены в виде отдельного специализированного блока, подключаемого к ЭВМ в случае необходимости.

Однако для современных универсальных вычислительных машин широкого применения можно назвать круг обязательных функций, возлагаемых на процессор. В частности, для машин малого и среднего класса типа «Минск-32», предназначенных для решения широкого круга задач научно-технического и плано-экономического характера обязательными функциями процессора можно считать:

- обработку двоичной информации в режиме с фиксированной и плавающей запятой;
- обработку адресной информации;
- обработку десятичной информации.

Арифметическое устройство, выполняющее все эти функции, экономичнее реализовать в комбинированном варианте. Блочные, специализированные устройства, независимо от своего назначения, занимают приблизительно одно и то же оборудование. Определяется это прежде всего тем, что каждое из устройств должно содержать в себе громоздкое многоразрядное регистровое оборудование для приема, хранения и выдачи информации. Гораздо экономичнее в процессоре ЭВМ широкого применения совместить функции различных АУ. Следует иметь в виду, что независимо от варианта организации АУ существует общий обязательный набор микроопераций.

В ЭВМ широкого применения, предназначенных для решения как научно-технических, так и плано-экономических задач, наиболее оптимальным с точки зрения оборудования является структура комбинированного АУ, совмещающего в себе функции: арифметики с фиксированной запятой; арифметики с плавающей запятой, десятичной арифметики, адресной арифметики, специальной арифметики.

В основе организации многофункционального АУ лежит идея использования ряда

микросумматоров, коммутируемых в нужных последовательностях в зависимости от требуемого режима работы (9). Блок-схема такого АУ, используемого в ЭВМ «Минск-32», приведена на рис. 16. Комбинированное АУ включает в себя: 37-разрядный регистр Р2, блок 10 микросумматоров, 37-разрядный регистр Р1, блок местного управления.

Машинное слово имеет 37 битов, включая знаковый разряд. При представлении чисел с фиксированной запятой запятая фиксируется перед старшим (нулевым) разрядом, т. е. все числа должны быть по модулю меньше 1. Для цифровой части числа фиксированной запятой отведено 36 битов регистров Р1 и Р2 и 9 микросумматоров 1Т ÷ 9Т. При этом клапаны К₁ ÷ К₉ объединяют 9 микросумматоров в единый 36-разрядный сумматор. Нулевой разряд хранит знак числа. Устройство использует модифицированный обратный код, поэтому микросумматор знаков Зн Т содержит 2 разряда.

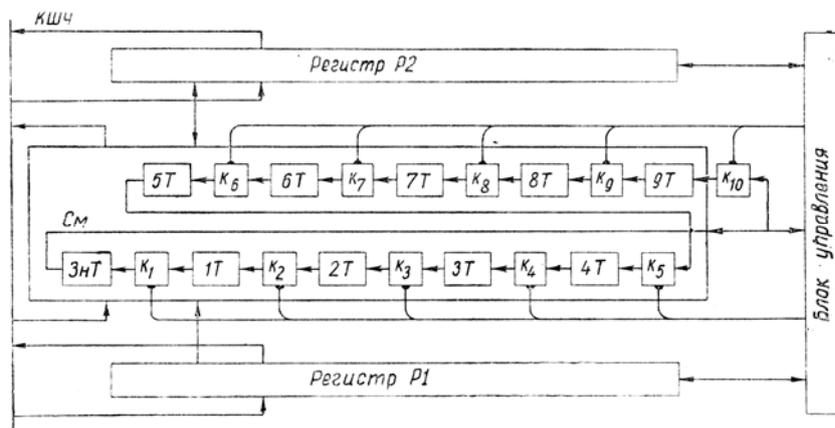


Рис. 16. Блок-схема арифметического устройства.

Как известно (раздел 1.8), при работе с плавающей запятой для мантиссы числа отводится 28 разрядов (с 1 по 28), для порядка — 6 (с 31 по 36). По два знаковых разряда выделяются для мантиссы и порядка. Мантисса представляется числом с запятой, фиксированной перед старшим разрядом, порядок — целым числом.

Числа, участвующие в операциях с плавающей запятой, могут быть как нормализованными, так и ненормализованными. Микросумматоры с помощью клапанов разбиваются на 2 независимые группы. С помощью клапанов К₁ ÷ К₇ объединяются микросумматоры 3н Т, 1Т ÷ 7Т в непрерывный сумматор мантиссы. Клапан К₈ отключает от мантиссы 2 микросумматора 8Т, 9Т, которые с помощью клапана К₉ объединены в сумматор порядка.

При обработке адресной информации все микросумматоры объединяются в 2 независимые группы для обработки двух 16-битных адресов: А1 и А2. Первый адрес обрабатывается микросумматорами 2Т, 3Т, 4Т, 5Т, которые объединены между собой клапанами К₃, К₄, К₅. Второй адрес обрабатывается микросумматорами 6Т, 7Т, 8Т, 9Т, которые объединяются между собой клапанами К₇, К₈, К₉. Клапан К₆ разрывает связи между микросумматорами 5Т и 6Т. Используя такую группировку микросумматоров, можно выполнять еще ряд команд специализированной арифметики, в частности, команд для обработки индексных констант.

При посимвольной обработке алфавитно-цифровой информации в работе АУ участвуют первых 2 микросумматора 1Т, 2Т, объединенные между собой открытым клапаном К₄. Клапан К₃ отделяет сумматор символа от остальной части АУ. К операциям над символами можно отнести операции приема, выдачи, замещения, обмена, сложения и т. д.

К операциям десятичной арифметики можно отнести операции сложения, вычитания, умножения. Целесообразность этих операций в машинах широкого применения обосновывается длительностью выполнения подпрограмм перевода из двоичной системы в десятичную и наоборот: от 2 до 9 мксек для ЭВМ «Минск-22». Это значительная цифра, если учесть объемы задач обработки данных. Скажем, для такой типовой задачи, как расчет заработной платы для завода средней мощности, объем информации только одного вывода составляет около 120 000 строк печати на механизм АЦПУ-128-2. Если взять, что в каждой строке содержится только 20% цифровой информации, то для перевода из двоичной системы в десятичную понадобится машинного времени (для ЭВМ «Минск-22»).

$$\frac{120000 \cdot 20 \cdot 128}{100} \cdot 2 \cdot 10^{-3} \text{сек.} = 6120 \text{сек}$$

Применение десятичной арифметики на несколько порядков повышает эффективность работы процессора в задачах обработки данных (отсутствует перевод $2 \rightarrow 10$, $10 \rightarrow 2$).

Очень важным вопросом при разработке десятичного АУ для ЭВМ широкого применения является выбор кода десятичных цифр [8], [9], [14]. Наиболее приемлемым кодом является код 8421, учитывая наглядность, определяющей простоту эксплуатации, оптимальность выполнения схем с точки зрения затраты оборудования, удобство с точки зрения совместимости двоичной и двоично-десятичной арифметики. Этот код позволяет совместить двоичный параллельный сумматор на n битов с двоично-десятичным сумматором на $n/4$ десятичных разрядов с использованием отдельного знакового сумматора.

Рассмотрим возможные варианты совмещения двоичного и десятичного сумматоров.

Арифметические действия над десятичными числами в кода 8421 производятся в двоичной системе счисления с введением необходимых поправок на десятичный перенос в старшие разряды коррекции. Однако десятичный код 8421 создает некоторые трудности и неудобства при образовании обратного кода отрицательных десятичных чисел. Получение обратного кода 8421 (дополнение каждого разряда до девяти) в существующих схемах связано с введением некоторых дополнительных операций: необходимо из девяти вычесть число, стоящее в данном разряде.

$D_{\text{обр}} = 9 - D$, где D — одноразрядное десятичное число.

Операция вычитания является трудоемкой. Обычно ее заменяют операцией сложения с шестью и последующим инвертированием, т. е. дополнением полученного числа до 15.

$15 - (D + 6) = 9 - D$, а это и есть дополнение до 9.

Дополнение до 9 получают также первоначальным инвертированием двоичного числа с последующим прибавлением десяти по модулю 16 (игнорируется перенос в старший разряд). Известно также, что для дополнения до 9 кода 8421, если разряды кода представлены параллельно, используются схемы с более простым алгоритмом, чем сложение и инвертирование. Для получения дополнения до 9 разряд с весом «1» всегда инвертируется, разряд с весом «2» — остается, разряд с весом «4» в дополнении содержит «1», если разряд «2» или «4» в исходном состоянии находится в единице. Разряд с весом «8» в дополнении содержит «0», если разряд «2» или «4», или «8» находится в исходном состоянии в единице, т. е. имеем следующие соотношения:

$$\begin{aligned} D_1 \text{ обр} &= \overline{D_1}, \\ D_2 \text{ обр} &= D_2, \\ D_4 \text{ обр} &= \overline{D_2} \overline{D_4} + \overline{D_2} D_4, \\ D_8 \text{ обр} &= \overline{D_2} + \overline{D_4} + \overline{D_8} \end{aligned}$$

В этом случае требуется дополнительная схема анализа и дешифрации исходного состояния двоичных разрядов тетрады.

В ЭВМ «Минск-32» реализован вариант устройства, который позволяет довольно просто и удобно с наименьшей затратой оборудования и времени решить проблему получения обратного кода десятичного числа и очень легко совместить на одной аппаратуре работу двоичного и двоично-десятичного сумматоров.

В этом устройстве перед выполнением операции десятичного сложения (вычитания) двух чисел исключается непосредственный перевод отрицательного десятичного числа в обратный код. Вместо обратного кода отрицательного числа берется его инверсия (поразрядное дополнение до 15).

$$-D = 15 - D = (9 - D) + 6,$$

т. е. число будет участвовать в операции с избытком 6. В этом случае десятичный перенос легко осуществляется с помощью переноса из самого старшего разряда двоичного представления десятичного разряда, т. е. операция сложения над числами производится целиком по правилам двоичной арифметики.

Коррекция положительного результата (прямой код) в десятичных разрядах, где возник перенос, осуществляется автоматически (вместо 10 переносится 16, т. е. исключаются 6 избыточных единиц) из десятичных разрядов, где не возник перенос, необходимо вычесть избыточные шесть

единиц. Вычитание шести (0110) заменяется прибавлением десяти (1010) по модулю 16 (игнорируется единица переноса в соседний старший разряд).

Если результат получился отрицательный, то его необходимо инвертировать. Перед инвертированием результата в тех разрядах, где возник перенос, необходимо произвести коррекцию — прибавить шесть (0110), так как перенос в этом случае равносителен вычитанию 16 вместо 10. Последующее же инвертирование автоматически исключает те избыточные шесть единиц, которые были введены в начале операции. Так, например, складываем $A+(-B)$, причем результат отрицательный, получаем

$$A + (-B) = A + B \text{ инв.} = A + (15 - B) - [A + (15 - B)] \text{ инв.};$$

получили инверсный результат, который требуется инвертировать:

$$[A + (15 - B)] \text{ инв.} = 15 - [A + (15 - B)] = -A + B = -[A + (-B)].$$

Таким образом, в этом алгоритме совмещается перевод числа в обратный код с коррекцией результата двоичного сложения. Для десятичных отрицательных чисел с одинаковыми знаками операция сложения (вычитания) производится также в прямом коде с тем, чтобы избежать введения дополнительного десятичного разряда в случае переполнения с целью сохранения верного знака результата и с целью применения однотипной коррекции.

Коррекция в этом случае такая же, как и при операциях над положительными числами, т. е. прибавляется шесть (—0110) в тех разрядах, где возник перенос или где числа больше 9 (скрытый десятичный перенос). В этом случае перед операцией знаки операндов гасятся, знак результата запоминается. Переполнение фиксируется знаковыми разрядами сумматора, т. е. здесь также используются цепи двоичной арифметики.

Примеры:

1. $19 - 88 = -69$ (11.0110.1001)

	11.0110.1001	
	+00.0001.1001	
	<u>11.1000.1000</u>	

 - а) инвертирование (-88) 11.0111.0111;
 - б) операция

	+00.0001.1001	
	11.0111.0111	
	<u>11.1001.0000</u>	
	0110	— результат отрицательный;
		— коррекция в том разряде,
		где возник перенос;
	<u>11.1001.0110</u>	
 - в) 11.0110.1001 — инвертирование.
2. $-19 + 88 = 69$ (00.011.1001)

	11.0001.1001	
	<u>00.1000.1000</u>	

 - а) инвертирование (-19) 11.1110.0110;
 - б) операция

	11.1110.0110	
	00.1000.1000	
	<u>00.0110.1110</u>	
	1	← циклический перенос
	00.0110.111	
	1010	— коррекция в разрядах,
		где нет переноса
		перенос игнорируется.
00.0110.1001		
3. $-19 - 88 = -107$ (11.0001.0000.0111)
 - а) перед операцией знаки операндов гасятся, знак результата запоминается;
 - б) операция

	+00.0001.1001		
	<u>00.1000.1000</u>		
	00.1010.0001	} коррекция в тех разрядах, где	
	<u>0110.0110</u>		} возник перенос и где тетрада
	01.0000.0111		

При этом знаковые разряды фиксируют переполнение.

На рис. 17 изображена блок-схема алгоритма выполнения операции устройством для сложения (вычитания) десятичных чисел. Реализация десятичного сумматора на 10 десятичных разрядов

(вместе со знаком) потребовала дополнительно оборудования около 4% от всего оборудования АУ.

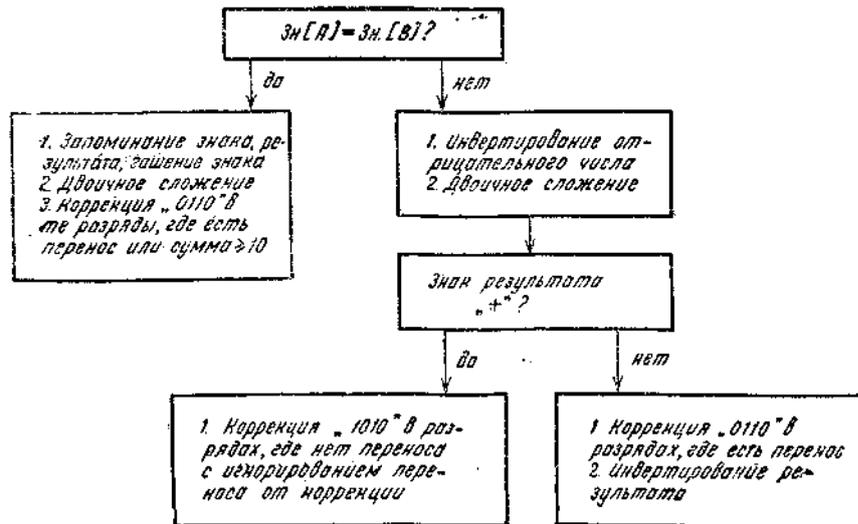


Рис. 17. Алгоритм операции сложения (вычитания) десятичных чисел.

Введение аппаратуры десятичной арифметики с использованием кода 8421 позволило очень удобно разбить 36 двоичных разрядов машинного слова на 9 тетрад. То же разбиение на 9 тетрад используется и для организации группового переноса при работе двоичной арифметики. Роль этих тетрад выполняют как раз микросумматоры $1T \div 9T$ (см. рис. 16). В нужные моменты (для учета или игнорирования переноса) включаются или отключаются клапаны $K_1 \div K_9$, разрывая или отключая тетрады. Реализация вышеописанного алгоритма на комплексе элемента 600 кгц позволила получить следующее время выполнения операций:

а) непосредственное сложение (вычитание) двух 10-разрядных чисел 10 мксек;

б) умножение методом последовательного суммирования $10 \div 620$ мксек. В таком АУ десятичное число содержит знак и девять десятичных цифр. Десятичные числа могут изображать целые числа и числа с фиксированной запятой. Для чисел с фиксированной запятой машина оперирует в диапазоне.

$$0 \leq |X| \leq 1 - 1 \cdot 10^{-9}$$

Для целых десятичных чисел можно условно зафиксировать запятую после 36-го разряда. В этом случае диапазон чисел будет

$$0 \leq |X| \leq 1 \cdot 10^{+9} - 1.$$

При комбинированном АУ доля аппаратуры распределяется следующим образом:

а) основная регистровая часть для режимов с фиксированной и плавающей запятыми 275 стандартных ячеек;

б) адресная часть требует дополнение 8 стандартных ячеек;

в) десятичная арифметика требует дополнение 21 ячейку.

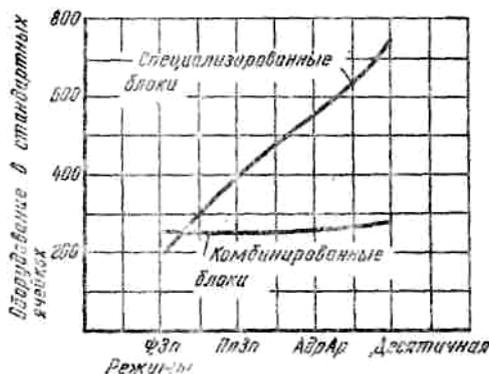


Рис. 18. Зависимость оборудования АУ в зависимости от типа реализации.

На рис. 18 приведены зависимости оборудования для реализации арифметического устройства при наращивании специализированными блоками и комбинированным исполнением АУ. Из этих зависимостей видно, что в комбинированном варианте (кривая 2) большая затрата только стартового оборудования 275 ячеек. Это больше, чем в каждом отдельном случае для режимов с фиксированной и плавающей запятыми при специализированных блоках, но гораздо меньше их суммы (около 400 ячеек). Прирост каждого режима в кривой 2 гораздо меньше, чем в кривой 1. Чтобы реализовать комбинированным вариантом все 4 режима, необходимо около 300 ячеек, против 750 в случае использования

специализированных блоков.

Соотношение символа и слова

В современных ЭВМ для облегчения процедуры обработки алфавитно-цифровой информации должна быть аппаратура адресации (выборки) не только машинного слова, но и каждого алфавитно-цифрового символа.

Машинное слово используется для представления командной и рабочей информации (команды, индексные ячейки, управляющие слова, базисные ячейки, операнды для работы в двоичной системе с фиксированной и плавающей запятыми и десятичные операнды). Символ используется для работы с полями переменной длины и для обработки массивов внешних устройств.

Известны различные варианты совмещения работы по словам и символам [10]. Выбранная в ЭВМ «Минск-32» схема совмещения имеет прежде всего ту особенность, что в одном машинном

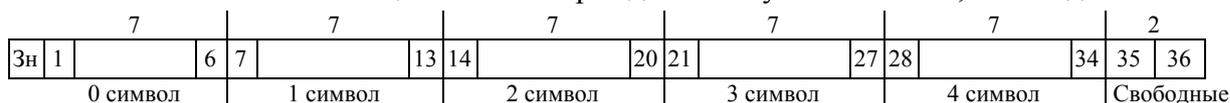


Рис. 19. Расположение символьной информации в машинном слове.

слове укладывается не целое число символов, так как наряду с 37-битным машинным словом в качестве информационной единицы используется 7-битный алфавитно-цифровой символ.

Все ВнУ, независимо от типа, используют для обмена с процессором 8-битный символ. При обработке полей переменной длины в одном машинном слове располагается 5 символов, слева направо (рис. 19). При посимвольной работе как бы не существует машинных слов. В программах ведется адресация независимо от машинных слов переменного поля символов. При этом читаются символы из машинного слова или записываются в машинное слово слева направо. При увеличении адреса очередного символа, если исчерпана емкость машинного слова, происходит переход на следующее слово и т. д. Таким образом, последовательное увеличение адреса символа соответствует продвижению слева направо по словам, снизу вверх по адресам.

При работе с внешними устройствами существуют 2 режима компоновки 7-битным символом 37-разрядного машинного слова:

- а) по 5 символов в одно слово;
- б) по 6 символов в одно слово.

На рис. 20 показано размещение алфавитно-цифрового массива в 37-разрядной памяти. В одном машинном слове укладывается пять 7-битных символов, каждый шестой символ начинает новое машинное слово. Тридцать пятый и тридцать шестой разряды машинного слова в этом случае в работе не участвуют.

На рис. 21 показано размещение массивов двоичной информации (команды, операнды). В одну ячейку памяти записывается, читается 6 символов. Каждый шестой символ имеет только 2 значащих разряда и 5 фиктивных, всегда нулевых. Каждый седьмой символ начинает новое машинное слово.

При работе с символами адрес состоит из двух частей: адреса слова и номера символа в слове. Несмотря на наличие в машине двух видов представления информации: слова и символа, при

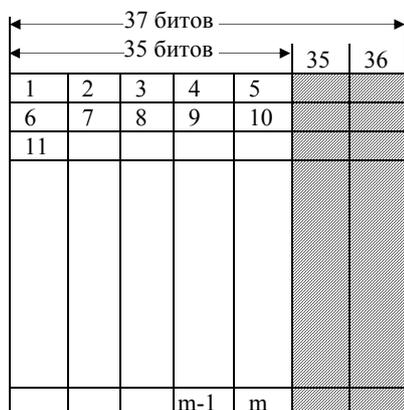


Рис.20. Расположение алфавитно-цифрового массива в 37-разрядной памяти

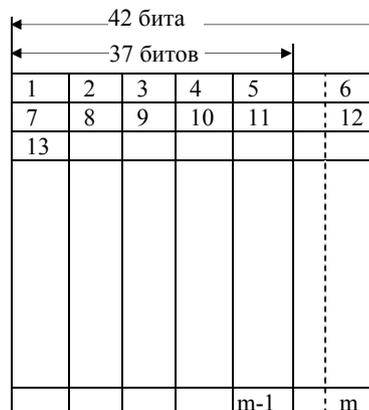


Рис.21. Расположение двоичной информации в 37-разрядной памяти

обращении к памяти адресация всегда сводится к заданию адреса машинного слова и дополнительно номера символа в машинном слове во втором случае.

Такая многофункциональность работы МОЗУ, когда возможно, одновременное обращение и словами и символами, позволяет организовать две эффективные схемы контроля:

- схему формирования контрольного разряда,
- схему контроля сдвига массива информации.

Схема формирования контрольного разряда

Известны устройства, позволяющие осуществлять контроль единичной ошибки с помощью контрольного разряда, содержимое которого дополняет количество единиц контролируемого слова до четного или нечетного. Однако очень часто эти устройства не позволяют оперативно формировать контрольный разряд слова при изменении его части. На рис. 22 приведен алгоритм работы схемы формирования контрольного разряда, примененный в ЭВМ «Минск-32». Эта схема ускоряет работу МОЗУ при формировании

контрольного разряда слова с новым вписанным символом. В машине применен контроль на «нечет» (контрольный 38-и разряд «К.» дополняет до нечетного количество единиц в 37-разрядном машинном слове). При записи слова в МОЗУ контрольный разряд формируется и отсылается в память вместе со своим машинным словом. При чтении проверяется соответствие машинного слова своему контрольному разряду.

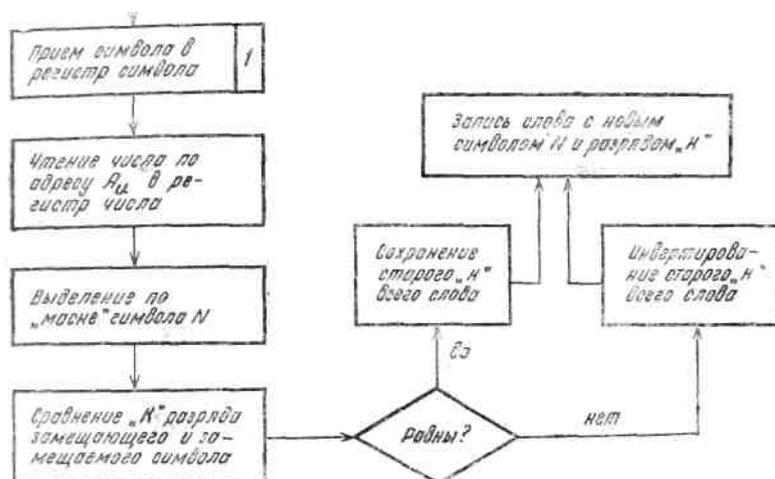


Рис. 22. Алгоритм формирования контрольного разряда.

Формирование контрольного разряда информационного слова при изменении части этого слова производится при чтении слова, для чего образуются контрольные разряды частей слова (символов), сравниваются контрольные разряды заменяемого и записываемого символов, и по результату сравнения только этих двух символов формируется контрольный разряд всего нового слова. Предложенная схема формирования контрольного разряда имеет существенный экономический эффект. Она сокращает на 10% время цикла обращения к МОЗУ.

Контроль сдвига массива информации

В настоящее время вопросу контроля правильности работы ЭВМ уделяется большое внимание. Потеря информации часто приводит к значительным экономическим убыткам. Однако любой контроль, простейшая схема проверки правильности работы — это избыточность аппаратуры. Выбор того или иного алгоритма контроля это прежде всего нахождение оптимального решения конфликтного соотношения эффективность-стоимость. Современные ЭВМ снабжены различными системами контроля, от простейшего контроля по паритету и до полного дублирования аппаратуры [15]. Можно сказать, что проблема контроля единицы информации решена на сегодняшний день более или менее полно. Существуют различные способы не только обнаружения, но и исправления ошибок в символе, слове. Несколько сложнее решается вопрос контроля массивов информации. Актуальность этой проблемы объясняется естественной тенденцией увеличения информационности решаемых задач.

Например, в таких двух типовых для промышленных предприятий задачах, как «Расчет заработной платы» и «Учет материалов» для одного из заводов страны объем информации при

решении на ЭВМ «Минск-22» следующий:

Тип задачи	Количество символов при обмене МОЗУ—НМЛ	Количество символов на УПЧ	Время ввода
Учет материалов.....	$139 \cdot 10^6$	$980 \cdot 10^3$	52 мин.
Заработная плата.....	$56 \cdot 10^6$	$5 \cdot 10^6$	1 час. 39 мин.

Можно легко представить себе объем такой задачи, как «Перепись населения», если она должна решаться годами на нескольких ЭВМ «Минск-32». Обычно при решении таких задач массивы контролируются специальными константами — контрольными суммами. Однако применение контрольного суммирования требует дополнительной аппаратуры и машинного времени. Кроме того, контрольное суммирование не всегда обнаруживает такое нарушение массивов, как перетасовка перфокарт, «переворот массива».

В ЭВМ «Минск-32» применен аппаратный метод обнаружения потери единицы информации, появления информации, перетасовки и т. д., т. е. все те нарушения в последовательности чередования символов, которые приводят к сдвигу массива обмена.

Эта схема предназначена для контролирования массивов обмена в тракте память — внешние устройства, т. е. информации, которая циркулирует в каналах связи. В основе работы этой схемы лежит вышеописанная схема формирования контрольного разряда всего машинного слова по контрольным разрядам символов.

Необходимые условия работы схемы контроля сдвига информации:

- квант обмена (символ) должен иметь контроль по паритету;
- какому-то определенному количеству полных символов должен соответствовать один неполный;
- выбранному определенному количеству полных и относящемуся к ним неполному символу по их контрольным разрядам должен формироваться общий контрольный разряд по паритету;
- проверка правильности формирования общего контрольного разряда производится не сравнением контрольных разрядов символов, а пересчетом «1» в каждом небольшом массиве с неполным символом.

На рис. 23 приведена схема компоновки массива с неполным символом, которая применена в ЭВМ «Минск-32». После каждых пяти полноразрядных символов, имеющих свой контрольный разряд, следует шестой неполный, имеющий всего два значащих информационных бита и соответствующий их информации контрольный разряд. Информация пяти полных и одного неполного символа записывается посимвольно в одно 37-разрядное машинное слово, после

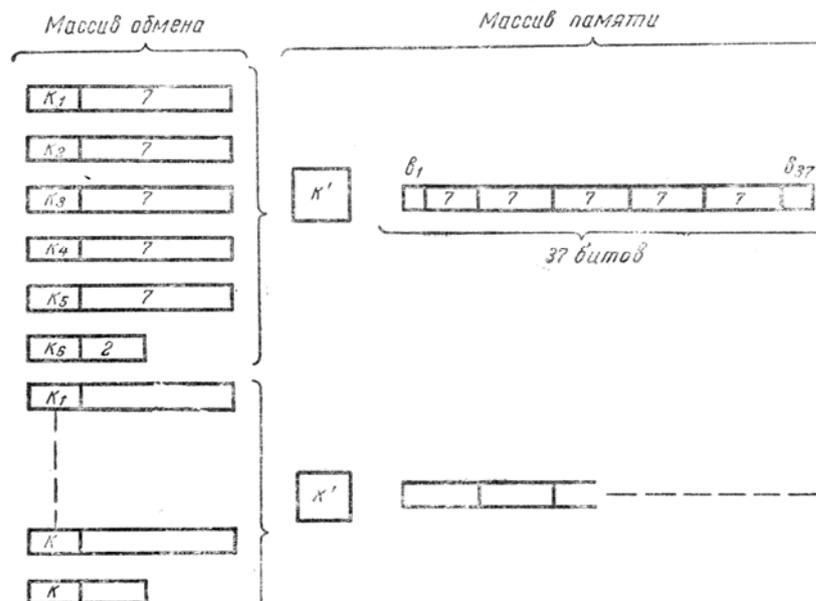


Рис. 23. Схема формирования массива с неполным символом.

записи последнего неполного символа общий контрольный разряд K' всего 37-разрядного машинного слова должен выражаться формулой

$$K' = \sum_{i=1}^6 K_i \quad (2.1)$$

Выражение (2.1) определяет формируемый контрольный разряд каждого слова по контрольным разрядам массива обмена. Причем, информация K' должна соответствовать количеству единиц во всем машинном слове

$$K' = \sum_{i=1}^{37} b_i \quad (2.2)$$

Выражение (2.2) определяет истинный контрольный разряд каждого машинного слова в массиве памяти. Поэтому при верно скомпонованном массиве должно соблюдаться следующее равенство

$$\sum_{i=1}^6 K_i = \sum_{i=1}^{37} b_i \quad (2.3)$$

В выражениях 2.1÷2.3 суммирование производится по модулю 2.

Если в процессе обмена произошел сдвиг в массиве обмена (неважно в какую сторону), то, начиная с места сдвига, произойдет нарушение правой составляющей выражения (2.3), так как на место всех неполных символов будут записываться в машинных словах полные символы. Происходит потеря информации, которая обнаруживается схемой, реализующей выражение (2.3). Реализация вышеописанного алгоритма контроля в ЭВМ «Минск-32» имеет значительный эффект. Ошибка в компоновке массива обнаруживается сразу же в момент возникновения ошибки (при обмене массивом), а не в процессе вычислений. Применение такого контроля позволяет обнаруживать очень частые ошибки чисто процедурного характера: заправка перфоленты не той стороной, перетасовка карт, ввод другого массива на место заготовленного. В этом случае установка специального указателя в ЭВМ является сигналом для оператора о неверной компоновке рабочего массива.

Этот контроль обнаруживает не только процедурные ошибки, ошибки оператора, но и сбои аппаратуры, которые приводят к сдвигам информационных массивов. Вопрос о целесообразности применения такого контроля становится актуальным в последнее время в связи с увеличением объема информации современных задач, тем более, что его использование не требует предварительного подсчета констант сумм, что очень валео при использовании аппаратуры АПД и межмашинных связях.

Все ли полные символы, попадая на место неполных, приводят к обнаружению сдвига?

Естественно, что в различных по характеру массивах существует различная способность к обнаружению пропажи символов. Другими словами, в массивах различного характера существует неодинаковый процент обнаруживающих полных символов, которые, попадая на место неполных, приводят к срабатыванию схемы контроля. Скажем, поля двоичной информации, поля алфавитно-цифровой информации имеют различную вероятность обнаружения. В частности, было исследовано поле двоичной информации в коде ГОСТ 10859-64. Подсчет показал, что 50% из 128 символов ГОСТ 10859-64, попадая на место неполных символов в выполненной схеме контроля, обнаруживают сдвиг информации. Если обозначить вероятность обнаружения через p , то для поля двоичной информации в коде ГОСТ 10859-64 $p = 0,5$. Кроме того анализ таблицы символов показал, что на величину p не влияет разрядность, избыточность неполного символа. Это очень существенно, так как сводится к минимуму избыточность контрольной части. В принципе, можно брать однокбитовый неполный символ. В машине «Минск-32» используется 2-битовый неполный символ, так как он получается в результате естественного размещения 7-битового символа в 37-разрядном машинном слове ($\frac{37}{7} = 5 + 2/7$). Применение предлагаемой схемы алгоритма

обнаружения сдвига массива информации целесообразно в ЭВМ, у которых $\frac{B_{\text{слова}}}{B_{\text{символа}}} = K_{\text{ц}} + B_{\text{ост}}$,

где $B_{\text{слова}}$ — разрядность слова, $B_{\text{символа}}$ — разрядность символа, $K_{\text{ц}}$ — целое число, $B_{\text{ост}}$ — остаток разрядностью $B_{\text{ост}}$

В тех же ЭВМ, где $\frac{B_{\text{слова}}}{B_{\text{символа}}} = K_{\text{ц}}$, схема, реализующая алгоритм обнаружения, должна быть

видоизменена. В этом случае вводить специально избыточность нецелесообразно. На место неполного символа следует в массивах обмена писать всегда нулевой символ, а в каналах обмена ввести аппаратуру проверки попадания полного символа на место неполного. Эта аппаратура дешевле избыточности в памяти, к тому же величина p для равновероятностного кода поднимается до значения 0,99 [33]. На основании выше изложенного можно сделать следующие выводы.

Многофункциональность аппаратуры является эффективным способом снижения стоимости оборудования процессора, так как существенно влияет на стоимость. Для машин широкого применения при использовании принципа микросумматоров, изложенного в данной работе, оптимальным способом совмещаются на одном оборудовании режимы: двоичной арифметики с фиксированной и плавающей запятыми, адресной арифметики, спецарифметики, десятичной арифметики. При этом почти в 2,5 раза выигрывается оборудование в сравнении с применением специализированных блоков. В табл. 2.1 ÷ 2.3 приведены данные распределения стоимости и оборудования ЭВМ «Минск-32». Рассмотренный алгоритм десятичного сложения позволяет эффективно использовать код 8421 для совмещения функций двоичного и двоично-десятичного сумматоров. Этот алгоритм позволяет реализовать простую и достаточно быстродействующую схему коррекции. При незначительных затратах оборудования на реализацию предложенного алгоритма (около 2,2% от оборудования процессора) операции десятичной арифметики выполняются со скоростью, соизмеримой со скоростью операций двоичной арифметики.

При обращении к памяти как по словам, так и по символам целесообразно совмещать схемы контроля по словам и символам. При этом рассмотренная схема формирования контрольного разряда слова на 10% сокращает цикл обращения к памяти, что имеет существенный экономический эффект.

Применение рассмотренного алгоритма формирования контрольного разряда позволяет организовать оригинальную схему контроля сдвига массивов информации.

Эффективность работы предложенной схемы контроля достаточно высока уже при массивах величиной в 16 машинных слов. Это обстоятельство позволяет применять предлагаемый способ контроля взамен известного контроля с использованием контрольных сумм. Опыт работы на ЭВМ «Минск-32» показал его достаточно высокую эффективность.

Таблица 2.1

Распределение стоимости и оборудования ЭВМ «Минск-32»

Наименование устройства	% от общей стоимости ЭВМ	% от общего электронного оборудования ЭВМ
1. Устройство питания.....	1,5	2,2
2. Арифметическое устройство.....	5,8	16,6
3. Центральное устройство управления.....	5,45	15,5
4. Устройство обмена	5,15	12,5
5. Центральный пульт.....	1,5	1,2
6. Оперативная память на 16К.....	17	9,85
7. Вывод на перфоленту.....	1,2	1,9
8. Ввод с перфолент.....	2	1,85
9. Ввод с перфокарт.....	2,3	4,16
10. Вывод на перфокарты.....	4,4	7,7
11. Вывод на печать.....	7,35	5,86
12. Накопитель на магнитной ленте с пятью НМЛ... ..	23,6	11,1
13. Устройство подготовки на перфоленте.....	3,95	1,3
14. Устройство подготовки на перфокартах.....	2	3
15. Устройство ввода-вывода с пишущей машинки..	2,1	4,1
16. Запасные части, сервис, документация.....	15,3	1,21

Таблица 2.2

Распределение площади размещения ЭВМ

Общая площадь	Доля общей площади		Вспомогательная
	Процессор	ВнУ	
80 м ²	40%	60%	40 м ²

Распределение оборудования процессора (МОЗУ емкостью 16К)

Блок процессора	% от общего электронного оборудования
1 Оперативная память.....	17,7
2 Арифметическое устройство.....	29,4
в том числе:	
регистровая часть	7,2
фиксированная запятая	16,4
плавающая запятая.....	3
адресная часть	0,3
десятичная арифметика	2,2
блок совместимости	0,3
3 Центральное устройство управления	27,8
в том числе:	
многопрограммность	6,5
совместимость с М-22М	0,9
совместимость с М-2, М-22	0,5
управляющая часть	20,4
4 Устройство обмена	22,5
в том числе:	
каналы связи	21,6
системная связь	0,9
5 Пульт управления	2,4

В настоящее время многопрограммность стала одной из основных особенностей современных ЭВМ.

Известны различные варианты организации многопрограммности у нас в стране и за рубежом. Это такие машины, как БЭСМ-6, «Днепр-22», «Урал-14», М-222, «Минск-23», ТХ2, Стретч, МК-4В, NEAC-2203, системы машин третьего поколения IBM/360, Spectra-70, Siemens-4004, Sistem-4, РДР-8 и т. д. [10], [16].

В данной главе рассматривается организация многопрограммности ЭВМ «Минск-32» с точки зрения равномерности загрузки оборудования ЭВМ при минимальных затратах оборудования на реализацию самой многопрограммности.

3.1. АНАЛИЗ ЗАГРУЗКИ ОБОРУДОВАНИЯ МАШИНЫ «МИНСК-22»

Большой интерес для проектировщиков представляет проблема эффективности оборудования, равномерности загрузки оборудования ЭВМ. В этом смысле очень наглядной иллюстрацией может служить наиболее распространенная однопрограммная универсальная машина «Минск-22» [34]. Работа всех внешних устройств этой машины строго детерминирована характером, объемом текущей задачи и производительностью механизмов, входящих в эти устройства.

В состав процессора ЭВМ «Минск-22» входят: арифметическое устройство (АУ), устройство центрального управления (ЦУ), оперативное запоминающее устройство (МОЗУ).

Структурно для ЭВМ «Минск-22» любую задачу можно представить, как комбинацию следующих четырех этапов: ввода, счета, обмена МОЗУ — НМЛ, вывода. Некоторые этапы в специфичных задачах могут отсутствовать, в частности третий и четвертый. В некоторых задачах этапы могут выполняться многократно. Существенный интерес представляет ответ на вопрос о распределении загрузки устройств процессора по времени и оборудованию для каждого этапа.

Ввод

Под вводом понимается процесс перенесения информации с носителей внешних устройств в ячейки памяти МОЗУ. Ввод осуществляется в машине «Минск-22» с перфоленты и перфокарт. Это наиболее распространенные носители для современных ЭВМ. Для конкретности определим, что ввод осуществляется с перфоленты со скоростью 1000 строк/сек.

Арифметическое устройство (АУ) производит на своих регистрах разворот машинного слова, при этом на каждые 1 *млсек* считывания символа информации тратится 4 *мксек* работы АУ. Кроме того, на каждые 6 символов, т. е. 6 *млсек*, АУ отрабатывает 24 *мксек* такта контрольного суммирования и записи готового слова в память МОЗУ. Таким образом, время занятости АУ ($t_{\text{АУ}}$) определяется

$$t_{\text{АУ}} = \frac{4}{1000} + \frac{24}{6000} = 0,008$$

Из всего времени ввода АУ работает только 0,8%. При этом в работе участвуют только блоки двух регистров Р1 и СМ, что составляет 55% оборудования всего устройства. Центральное устройство управления (ЦУ) обслуживает работу всего ввода, вырабатывая тактовые импульсы. Можно считать, что ЦУ работает все 100% времени ввода, но при этом занято около 60% оборудования всего устройства.

Счет

Под счетом понимается непрерывный автоматический процесс выполнения последовательности арифметических и логических операций, требующих обмена информацией между устройствами АУ, МОЗУ, ЦУ, а также автономного функционирования основных узлов этих устройств.

Во время выполнения операций счета все 100% оборудования АУ занято все 100% времени. В ЦУ 75% оборудования занято 100% времени. Около 25% оборудования блока связи с пультом ЦПУ и с внешними устройствами в работе не участвуют. Для оценки загрузки оперативной памяти МОЗУ необходимо проанализировать наиболее часто встречаемую операцию счета. По данным [17], [18] свыше 70% операций счета являются команды логики (пересылки, передачи

управления) и элементарные операции с фиксированной запятой. Такой усредненной операцией в ЭВМ «Минск-22» является 5-тактная команда, 3—4 такта которой являются тактами обращения к МОЗУ, т. е. МОЗУ работает в этих операциях в среднем 70%. В длинных операциях с фиксированной запятой (типа деления, умножения) и в операциях с плавающей запятой доля времени МОЗУ составляет от 10 до 50% (в среднем 30%). Если считать, что такие операции будут составлять оставшиеся 30% операций счета, то загрузка МОЗУ (P) составляет

$$P = 0,7 \cdot 70\% + 0,3 \cdot 30\% = 58\%.$$

При этом занято все 100% оборудования МОЗУ.

Обмен МОЗУ—НМЛ

Под обменом МОЗУ — НМЛ понимается процесс передачи информации из оперативной памяти на носители внешних накопителей (НМЛ) и наоборот. Во время обмена с магнитной лентой в арифметическом устройстве производится разворот машинного слова, при этом регистры АУ работают 4 мксек в каждые 40 мксек, а каждые 240 мксек в течение 24 мксек в АУ отрабатывается такт записи информации в МОЗУ и контрольного суммирования. Таким образом, АУ работает 20% времени обмена и при этом занято 55% оборудования АУ.

В ЦУ занято 60% оборудования в течение 60% времени обмена. Устройство МОЗУ на каждые 240 мксек тратит 24 мксек, т. е. участвует в работе только 10% всего времени обмена.

Вывод

Под выводом понимается процесс перенесения информации из памяти на носители внешних устройств. В [17] показано, что при самых благоприятных ситуациях (даже при использовании прерывания) в ЭВМ «Минск-22» во время вывода 50% времени процессор находится в режиме статического останова. При этом в момент выполнения самих команд вывода устройства МОЗУ и АУ работают 80% времени. Таким образом, можно считать, что занятость устройств АУ и МОЗУ составляет не выше 40% общего времени вывода. Загрузка оборудования так же, как и при вводе, распределяется в следующей пропорции: АУ — 50%, ЦУ — 60%, МОЗУ — 100%.

Результаты анализа четырех этапов решения задачи на устройствах процессора сведены в диаграммы, приведенные на рис. 24.

Анализируя диаграммы, можно сделать следующие выводы.

1. Счет полностью загружает по времени и оборудованию устройства АУ и ЦУ и на 65% память МОЗУ. Это наиболее полный производительный этап задачи.

2. Во время ввода-вывода и обмена МОЗУ — НМЛ недозагружены устройства АУ и МОЗУ. А в моменты нагрузки используется только половина оборудования устройств АУ и ЦУ.

Недоиспользование оборудования процессора можно проиллюстрировать конкретными цифрами реальных задач ЭВМ «Минск-22», приведенными в табл. 3.1. Из этой таблицы видно, что за все время решения задачи только около 66% времени используется АУ и 40% времени занято работой такое дорогостоящее устройство, как МОЗУ. Эти устройства требуют дозагрузки.

Таблица 3.1

Тип задачи	Общее время задачи	Время вводов	Время счета	Время обмена МОЗУ—НМЛ	Время вывода	Занятость устройств от общего времени задачи (в %)		
						АУ	ЦУ	МОЗУ
Расчет заработной платы	22 час. 24 мин.	1 час. 4 мин.	13 час. 28 мин.	1 час. 15 мин.	4 час.	66,5	100	40
Учет материалов	20 час. 58 мин.	52 мин.	11 час. 29 мин.	3 час. 5 мин.	55 мин.	62	100	39

3.2. ОРГАНИЗАЦИЯ МНОГОПРОГРАММНОСТИ

В предыдущем разделе показано, что задача, состоящая из классических этапов, не в состоянии обеспечить равномерной загрузки устройств процессора однопрограммной ЭВМ. Существенно недозагружены такие дорогие устройства, как АУ и МОЗУ. При этом следует иметь в виду, что в машине

«Минск-22» устройство АУ выполняет и функции адресной арифметики, что по времени счета занимает 20 ÷ 30% времени.

Из данных диаграммы (см. рис. 24) видно, что МОЗУ может одновременно обслуживать все четыре этапа решения задачи: ввода, счета, вывода, обмена. Арифметическое устройство одновременно может вести только одну программу счета вместе с функциями адресной арифметики. Следовательно, исходя из всего вышеизложенного, на однопроцессорных машинах широкого применения есть ресурсы для одновременного выполнения четырех различных этапов: ввода, обмена, счета, вывода. Такую загрузку одного процессора можно осуществить только выполнением одновременно не- I скольких программ.

Блок-схема многопрограммной ЭВМ «Минск-32» приведена на рис. 25. Такая ЭВМ может обслуживать одновременно 4 этапа решения задачи, загружая устройства процессора в соотношении, приведенном в диаграммах (см. рис. 24). Однако ту часть оборудования АУ, которая участвовала в этапах ввода-вывода и обмена и приводила к занятости АУ до 20 ÷ 40% времени, следует выделить в отдельное устройство обмена УО.

Блок-схема процессора

Процессор машины «Минск-32» (см. рис. 25) можно рассматривать как комбинацию двух микропроцессоров, работающих на одну оперативную память. Операционный микропроцессор состоит из арифметического устройства и центрального устройства управления. Канальный микропроцессор представляет собой устройство обмена УО, управляющее работой каналов ввода-вывода. К оперативной памяти в каждый момент времени может обращаться один из этих двух микропроцессоров. Центральное устройство управления синхронизирует очередность их работы в зависимости от ситуации. Такая структура процессора позволяет обслуживать одновременно несколько программ, переключая работу каждого микропроцессора на одну или несколько нужных программ.

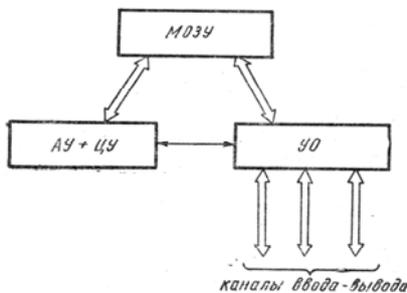


Рис. 25. Блок-схема многопрограммной ЭВМ

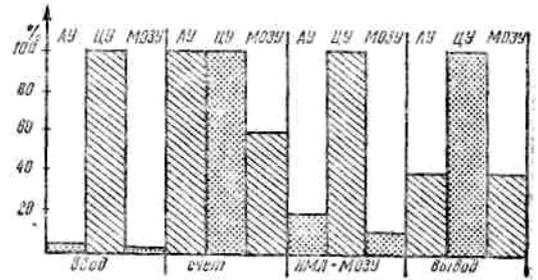
Чтобы поддерживать равновесие между потребностью и возможностью использования вычислительных средств и обеспечить быструю реакцию системы на изменение внешних условий, необходимо спроектировать систему таким образом, чтобы вычислительные устройства, микропроцессоры можно было перераспределять между программами. Такой системой может быть прерывание.

Система прерывания

Однопроцессорная машина «Минск-32» является многопрограммной. Возможность перехода машины от одной программы к другой оптимально обеспечивается системой прерывания, которая собирает и анализирует возникающие причины прерывания, и, в случае необходимости, переключает машину на нужную программу, прервав при этом работу менее приоритетной программы. Всего в машине 12 направлений прерывания для обслуживания четырех рабочих программ и восьми программ обслуживающей системы «Диспетчер».

На рис. 26 изображена блок-схема организации системы прерывания ЭВМ «Минск-32». Прерывание используется только для переключения работы процессора с одной из 12 программ другую. Число направлений прерывания 12, хотя причин прерывания может быть больше.

Занятость во времени устройств процессора на различных этапах решения задачи для «Минск-22»



Загрузка оборудования процессора на различных этапах решения задачи для «Минск-22»

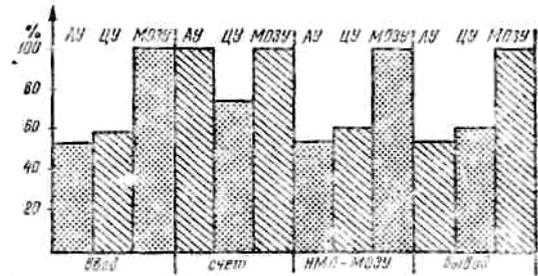


Рис. 24. Результаты анализа этапов решения задачи на устройствах процессора.

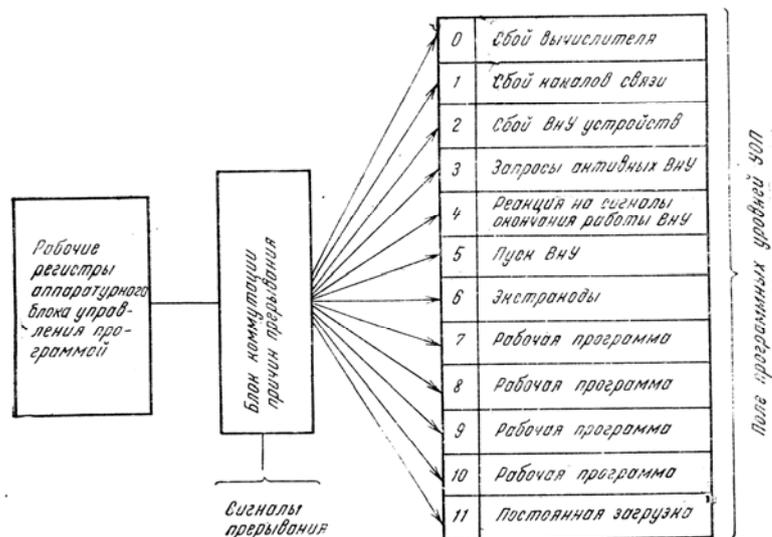


Рис. 26. Блок-схема организации прерывания.

Процедуру прерывания осуществляет операционный микропроцессор, он же может в любой момент времени выполнять только одну из 12 программ и в то же время загружает работой каналный микропроцессор по одновременному обслуживанию ВЧУ нескольких программ.

Процесс прерывания сводится к обмену информацией между рабочими регистрами блока управления программой (сосредоточенной в различных устройствах операционного микропроцессора) и ячейками соответствующие уровней: прерываемого и прерывающего, хранящихся в отведенном для них поле программных уровней памяти МОЗУ. В специальном регистре номера уровня хранится номер текущей программы, который меняется при прерывании.

Нужно различать прерывание двух типов: автоматическое и программное.

Автоматическое прерывание осуществляется при появлении извне сигнала-причины, например, сигнала сбоя или окончания работы какого-либо внешнего устройства.

Программное прерывание осуществляется с помощью специальных команд переключения с одного уровня на другой и осуществляется любой из программ системы «Диспетчер». Программное прерывание осуществляет возврат к прерванной программе или переключение с одной из четырех рабочих программ на другую в случае необходимости.

При выборе количества уровней прерывания для системы «Диспетчер» можно руководствоваться различными критериями. В частности, количество уровней ЭВМ «Минск-32» в основном было определено требованиями по обслуживанию внешних устройств и сбойных ситуаций.

Программы «Диспетчер» занимают 7 первых наиболее приоритетных уровней и один наименее приоритетный— 12.

Назначение уровней прерывания следующее:

Нулевой уровень программы системы «Диспетчер», анализирующей сбой процессора;

1-й — уровень программы системы «Диспетчер», анализирующей сбой каналов связи;

2-й — уровень программы системы «Диспетчер», анализирующей сбой внешних устройств;

3-й — уровень активных внешних устройств, устанавливающих самостоятельно, независимо от программ, связь с машиной. Это уровень таких устройств, как пишущая машинка, ЭВМ, аппаратура; передачи данных по телефонным и телеграфным линиям и т. д.

4-й — уровень программы системы «Диспетчер», реагирующей на сигналы окончания работы внешних устройств;

5-й — уровень программы системы «Диспетчер», осуществляющей пуск внешних устройств;

6-й — уровень программы системы «Диспетчер», расшифровывающей экстракоды и включающей соответствующие подпрограммы;

7, 8, 9, 10-й — уровни рабочих программ;

11-й — уровень системы «Диспетчер», хранящей данные программы постоянной загрузки машины.

Автоматическое прерывание вызывает переключение на 0 ÷ 6-й уровни программ системы

«Диспетчер».

При появлении какого-либо сигнала прерывания блок коммутации причин прерывания фиксирует и определяет его приоритетность. Наиболее приоритетной является программа нулевого уровня. При появлении причины этого уровня прерывается работ, любой программы системы «Диспетчер» или работа рабочей программы и происходит переключение работы процессора на программу нулевого уровня. После окончания работы программы нулевого уровня система «Диспетчер» возвращает процессор на продолжение работы над прерванной программой или же может переключить на любую другую программу. При одновременном возникновении нескольких причин прерывания уровней $0 \div 6$ приоритет отдается уровню с меньшим номером. Если процессор находится в работе над программой одного из уровней $1 \div 6$, то автоматического прерывания не может произойти ни на какой другой уровень (вверх или вниз), прервать эти уровни может только причина нулевого уровня. Возврат с уровней $0 \div 6$ на прерванную или любую другую программу осуществляется по решению программы системы «Диспетчер». Это программное прерывание осуществляется как сверху вниз, так и снизу вверх.

3.3 СРЕДСТВА ОБЕСПЕЧЕНИЯ МНОГОПРОГРАММНОСТИ

Распределение оборудования машины для каждой задачи, установление очередности решения, распределение оперативной памяти и внешних накопителей, закрепление внешних устройств за каждой программой, реакция на сбойные ситуации, защита различных программ от взаимного влияния друг на друга, обеспечение инвариантности программ — вот основные проблемы, которые возникают при решении комплекса задач на однопроцессорной машине. Все эти проблемы должны решаться программно-аппаратурным комплексом.

Инвариантность программ

Соблюдение требования инвариантности программ является обязательным для многопрограммной ЭВМ, так как последовательность расположения программ в оперативной памяти зависит от многих факторов. Как известно, инвариантность достигается в машинах путем процедуры базирования адресов, которая заключается в следующем. Вся программа или какие-то ее независимые части пишутся в относительных адресах, позволяющих непрерывно адресовать ячейки внутри абстрактного массива определенной величины. Для абсолютной привязки программы к какой-то области памяти используется базисная константа. Абсолютный адрес получается после модификации относительного адреса базисной константой. Изменяя базисную константу, можно перемещать всю программу или какие-то ее части в памяти.

Структура двухадресной команды (см. рис. 7) является типовой для машины «Минск-32». Однако операция базирования не зависит от адресности команды. При использовании базирования значение абсолютного адреса операнда или адреса перехода определяется следующим выражением

$$A_{1,2}^{16} = a_{1,2}^{11} + (\bar{b}_{1,2})^6 + (i)^{16},$$

где $A_{1,2}^{16}$ — 16-разрядное значение адреса A1 или A2;

$a_{1,2}^{11}$ — 11-разрядный относительный адрес a_1 или a_2 ;

$(\bar{b}_{1,2})^6$ — 16-разрядное содержимое базисной ячейки с адресом b_1 или b_2 адреса A1 или A2;

$(i)^{16}$ — 16-разрядное значение содержимого i -й ячейки.

Процесс индексирования давно известен и используется в этой машине в обычном своем назначении. Следует только отметить адресацию полей индексных ячеек. Обычно в однопрограммных машинах поля i — ячеек располагались в одном определенном месте: либо в быстрых регистрах, либо в начальной (конечной) части МОЗУ. В многопрограммных ЭВМ каждая программа должна иметь свое независимое индексное поле. Так как расположение программ в МОЗУ привязывается с помощью специальных базисных констант, то и поля индексных ячеек должны привязываться абсолютно с помощью констант базисов индекса. В этом случае адрес индексной ячейки определяется из выражения

$$A_i^{16} = i^4 + B_i^{16}$$

где i^4 — 4-разрядный номер индексной ячейки в команде;

B_i^{16} — 16-разрядная константа базиса индекса.

Изменяя базис B_i , можно не только перемещать поле i ячеек каждой программы, но и задавать для одной программы произвольное число индексных полей.

Защита программ

Одним из основных условий обеспечения многопрограммности является соблюдение независимости отдельных программ, информационных массивов, т. е. обеспечение защиты любой из находящихся в памяти программ от нежелательного вмешательства со стороны всех остальных программ. В основном существует тенденция, что защита должна быть организована таким образом, чтобы исключать только возможность записи в программу при неправильной адресации. Однако введение защиты и на чтение информации во многом облегчает процесс отладки программ. Опыт работы на ЭВМ «Минск-32» доказал справедливость этого предположения.

В ЭВМ «Минск-32» используются следующие виды защиты [19].

1. Постоянная защита, обеспечивающая сохранность управляющей области памяти (УОП). Границы и критерий защиты для этой области жесткие и задействованы в схемах процессора.

2. Защита программных и информационных массивов рабочих программ. Границы защиты могут быть заданы для программных или информационных массивов. защите подлежат адреса чтения, записи и переходов.

3. Защита массивов обмена с внешними устройствами. При обмене с внешними устройствами массивами информации каждому устройству отводится рабочее поле, в управляющем слове указываются границы этих полей, защищающие соседние программные и информационные поля.

Рассмотрим подробнее принципы работы каждого вида защиты.

Постоянная защита. Постоянной защитой охватываются области памяти УОП. Характерной особенностью этой области является то, что обращение к ее ячейкам для записи, чтения или перехода разрешается только по специальным командам и признакам.

Ячейки УОП запрещенной области памяти защищены не только от рабочих программ, но и от обслуживающих программ. Содержимое ячеек этой области может изменяться только по специальным командам обращения к ней или в моменты программных и автоматических переключений уровней, а обращение к этим ячейкам производится только с помощью специальных тактов. Содержимое ячеек УОП, которые хранят границы защиты, может быть изменено только с помощью специальных команд системы «Диспетчер».

Защита программных информационных массивов по переменным границам. Так как в любой момент времени в оперативной памяти многопрограммной ЭВМ может храниться информация нескольких программ, то для предупреждения нежелательного вмешательства одной программы в работу другой каждой программе отводятся участки памяти, к которым разрешено обращение внутри этой программы. При обращении к ячейкам памяти, находящимся вне заданных границ, вырабатывается сигнал сбоя по защите.

В машине «Минск-32» выбрана защита, когда все адреса чтения, записи и переходов проверяются на соответствие выражению

$$Z_{\text{низ}} \leq A_T \leq Z_{\text{верх}} \quad (3.1)$$

где A_T — 16-разрядный текущий адрес обращения к МОЗУ;

$Z_{\text{низ}}$ — 7-разрядная константа нижней границы защиты;

$Z_{\text{верх}}$ — 7-разрядная константа верхней границы защиты. Константы $Z_{\text{низ}}$ и $Z_{\text{верх}}$ хранятся в специальной ячейке управляющей области памяти. Обращаться к этой ячейке для замены содержимого могут только специальные команды системы «Диспетчер». Следят за этим схемы постоянной защиты. Работа схемы для реализации выражения (3.1) понятна из рассмотрения блок-схемы, приведенной на рис. 27. Каждый адрес обращения принимается в 16-разрядный регистр адреса R_T^{16} , старшие 7 разрядов которого R_T^7 сравниваются с содержимым двух регистров $R_{\text{защ.н.}}^7$ и

$R_{\text{защ.в.}}^7$

В случае нарушения выражения (3.1) вырабатывается сигнал «Сбой по защите». Эта схема позволяет контролировать непрерывные участки, отведенные для программы. Задание границ защиты осуществляется с точностью до листа, величиной в 512 ячеек, так как в адресе сравниваются только 7 старших разрядов, а 9 младших битов адреса адресуют ячейки внутри заданного листа. Фактически константы *Защ. Н* и *Защ. В* являются номерами младшего и старшего разрешенного листа для данной программы. На рис. 28 изображен участок памяти объемом в $(m-n)$ листов, в котором разрешено работать текущей программе. Объем программы $V_T = (m-n) \cdot 512$. Точность задания листов зависит от разрядности констант m и n .

Можно уменьшить объем листов, увеличивая разрядность констант *Защ. Н.*, *Защ. В.*, что приведет к уменьшению потери памяти на стыке программ.

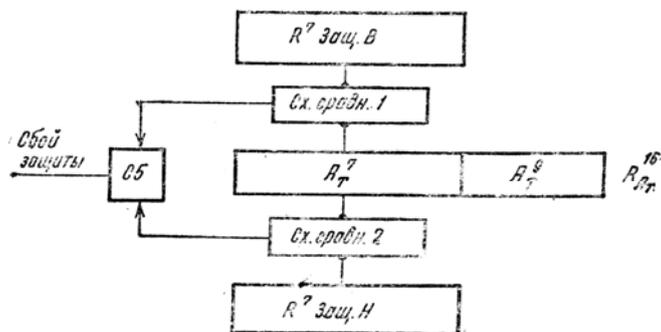


Рис. 27. Схема защиты.

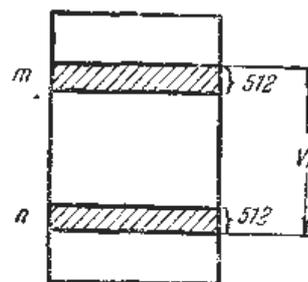


Рис. 28. Защищаемый участок памяти объемом V_m .

Защита массивов обмена с внешними устройствами. Как упоминалось выше, каналный микропроцессор организует одновременную работу нескольких ВнУ из различных программ. Наиболее прогрессивным способом организации одновременной работы нескольких внешних устройств является принцип приостановок с использованием специальных управляющих слов (УС). Более подробно сам принцип приостановок и структура управляющих слов будут рассмотрены в главе IV. В аспекте защиты программ следует только отметить, что так как информацию о массиве каждого внешнего устройства несет УС, то в нем необходимо указывать и информацию о защите. Если работа ведется по управляющему слову, в котором задаются нижний и верхний адреса обмена, то они одновременно являются нижней и верхней границами защиты. Если работа ведется с полями переменной длины, величина которых не может быть определена (когда обмен ведется до какого-то граничного отличительного символа), то в управляющем слове должна указываться граница максимально возможного разрешенного листа обмена. Если обмен с внешними устройствами производится вне заданных границ, вырабатывается сигнал «Сбой по защите». Такое использование управляющих слов позволяет обеспечить работу многопрограммной ЭВМ и при сопряжении с ней аппаратуры передачи данных по телеграфным и телефонным линиям, и с терминальными пультами, и с другими ЭВМ, т. е. с любыми активными ВнУ. Пуск всех ВнУ, загрузка и контроль управляющих слов, распределение памяти при работе с большим количеством ВнУ производятся программами системы «Диспетчер».

Электронные часы (ЭДВ)

Применение электронного датчика времени (ЭДВ) позволяет эффективнее организовать многопрограммность: подсчитывать время решения каждой программы, проверять правильность работы внешних устройств и остальных блоков программы, пускать в нужное время нужные устройства и программы, контролировать правильность хранения произвольной информации и сбойность аппаратуры.

В ЭВМ «Минск-32» в качестве счетчика ЭДВ используется одна из ячеек МОЗУ. Каждые 20 мксек (50 гц) по сигналу из блока ЭДВ производится вызов содержимого этой ячейки МОЗУ на регистровую часть процессора, где производится добавление «+1» к содержимому, после чего промодифицированная константа вновь отсылается в ячейку МОЗУ. Машинное слово позволяет хранить 2 константы: одну — текущего значения времени, вторую — граничного времени. При каждой модификации на «+1» производится сравнение этих констант на равенство. В случае

равенства этих констант вырабатывается специальный сигнал прерывания с установкой в «1» состояние специального индикатора ЭДВ. Меняя эти константы, можно задавать нужные интервалы прерывания.

В ЭВМ «Минск-32» в качестве ячейки ЭДВ используется ячейка управляющих слов для ВнУ, а процедура модификации аналогична процессу приостановок при работе с ВнУ. В этом случае датчик ЭДВ можно рассматривать как внешнее устройство. Такой подход существенно экономит оборудование, необходимое на громоздкие регистровые счетчики. При этом затраты машинного времени невелики. Так если ЭДВ включен, то процессор тратит $t_{\text{эдв}} = \frac{15}{20\,000} = 0,075\%$ времени счета. В сутки это может составить $t_{\text{эдв} \cdot \text{max}} = 1,1$ мин.

Управляющая информация многопрограммной ЭВМ

Для управления ходом выполнения нескольких программ требуется некоторая управляющая информация. Большое количество ВнУ и программ требует внушительных объемов управляющей информации, не вмещающейся в обычную регистровую аппаратуру. Известны различные варианты размещения управляющей информации [20]. Введение отдельной специализированной памяти не рентабельно, так как это снижает надежность процессора ЭВМ и удорожает систему. Достаточно большие объемы и высокая скорость работы оперативной памяти современных машин позволяют выделить специальную область (УОП) для хранения управляющей информации.

Возможность хранения в оперативном накопителе массивов служебной управляющей информации позволяет значительно экономить оборудование, стройнее и проще строить логику переключения работы процессора с одной программы на другую. Наиболее просто и естественно разбить оперативную память многопрограммной ЭВМ на следующие области:

- управляющую область памяти (УОП);
- служебную область памяти (СОП);
- рабочую область памяти (РП).

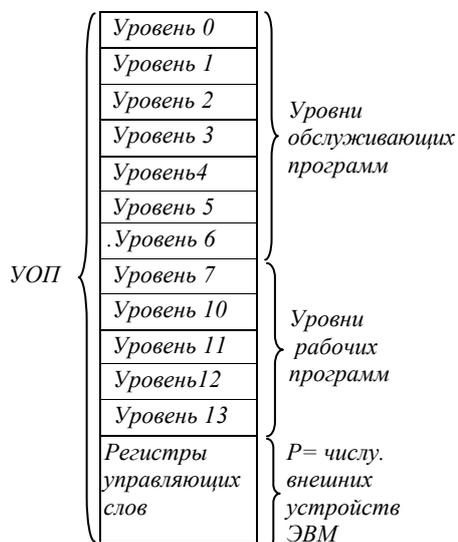


Рис. 29. Структура управляющей области памяти.

СОП служит для хранения программ обслуживающей системы «Диспетчер». В ЭВМ «Минск-32» объем этой памяти достигает 6000 ячеек МОЗУ. Эта область защищена обычной схемой защиты с использованием верхней и нижней границ защиты.

РП — поле рабочих программ. В машине «Минск-32» это поле занимает объем от 10 до 58 тыс. ячеек МОЗУ и может разместить одновременно до четырех независимых программ. Различные программные поля защищены друг от друга границами, установленными для них системой «Диспетчер».

УОП состоит из нескольких программных уровней с нужным числом ячеек в каждом и регистров управляющих слов для работы с внешними устройствами (рис. 29).

Ячейки каждого уровня содержат необходимую информацию для начального пуска программы или ее продолжения, т. е. состояние рабочих регистров операционного микропроцессора и основные данные самой программы.

В регистрах управляющих слов хранится информация, задающая для внешнего устройства наряду с командой ввода-вывода режим его работы, начальный адрес обмениваемого массива в МОЗУ и признак конца обмена, всю необходимую информацию для управления канальным микропроцессором работой нескольких ВнУ.

В ЭВМ «Минск-32» управляющая область памяти УОП имеет объем в 256 ячеек. 96 ячеек используется для хранения управляющей информации 12 программных уровней (8 уровней системы «Диспетчер» и 4 уровня рабочих программ). 160 ячеек используется для хранения значений управляющих слов 136 внешних устройств и ЭДВ. Все 12 программных уровней имеют одинаковую структуру, независимо от

функций программы (рис. 30). Нулевая и первая ячейки служат для хранения четырех текущих значений базисов данной программы: Б0, Б1, Б2, Б3. Третья ячейка хранит значение верхней и нижней границ защиты. Четвертая ячейка программного уровня в разрядах 21 ÷ 36 хранит значения базиса индексных ячеек данной программы. Четыре младших разряда базиса всегда содержат нулевую информацию.



Рис. 30. Структура программного уровня.

Пятая ячейка уровня служит для хранения при прерывании текущих значений индикаторов, указателей и регистра СчАК данной программы. В ячейку с адресом 6 записывается результат операции, который содержался в устройстве АУ в момент прерывания. Седьмая ячейка в разрядах 14 ÷ 17 хранит номер уровня, с которого произошло прерывание на данный уровень для возможности осуществления цепочки многократного прерывания. Вторая ячейка уровня не используется, так как она является дополняющей число ячеек уровня до восьми. Это облегчает двоичную адресацию ячеек уровня.

Оптимизация размещения управляющей информации многопрограммной ЭВМ

В современных ЭВМ можно отметить следующие основные этапы выполнения типовой команды: выборка команды и модификация команды;

- выборка операндов;
- выполнение операции;
- запись результата.

В зависимости от типа команды может меняться комбинация этапов, часть из них может отсутствовать. Часто некоторые из этих этапов совмещаются. Наиболее интересным и постоянно присутствующим этапом любой команды является выборка и модификация команды. Этому этапу необходимо уделять особое внимание, так как он занимает существенную долю времени выполнения команды. В ЭВМ «Минск-22» доля этого времени для типовой 5-тактной команды колеблется в диапазоне 20 ÷ 40% от времени выполнения всей команды. В многопрограммных ЭВМ в связи с увеличением дополнительных процедур обработки команды (базирование, защита) доля этого времени должна увеличиться (в ЭВМ «Минск-32» она составляет около 60%). Базирование, защита оперируют с управляющей информацией, хранящейся для каждой программы в специальных ячейках УОП. Большой интерес для разработчиков представляет проблема оптимизации параметров оборудования для хранения управляющей информации и времени выполнения команды. Должен быть оптимально решен вопрос о регистровом дублировании информации, расположенной в ячейках уровня: что из информации текущей программы должно быть расположено в ячейках уровня,— что в рабочих регистрах процесса и что должно дублироваться.

Базисные константы. Как уже упоминалось выше, процесс базирования сводится к суммированию относительного адреса с константой базиса. Поэтому на этап базирования необходимо отводить время $t_{сл}$. Если $t_{сл}$ соизмеримо с временем цикла оперативной памяти $t_{в}$, то нет необходимости тратить большое регистровое оборудование для хранения этих констант, а

достаточно их хранить только в ячейках УОП оперативной памяти. При этом цикл суммирования совмещается с циклом чтения из МОЗУ базисных констант.

Если у ЭВМ соблюдается зависимость:

$$t_{ц} - t_{ож} \geq t_{д} + t_{сл}, \quad (3.2)$$

где $t_{ц}$ — время цикла МОЗУ;

$t_{ож}$ — время ожидания информации из МОЗУ;

$t_{д}$ — время дешифрации информации команды;

$t_{сл}$ — время базирования,

то целесообразно базисные константы хранить на специальных регистрах, так как процесс базирования можно совместить с тактом чтения команды.

В ЭВМ «Минск-32» нарушается выражение (3.2), поэтому нет смысла хранить базисы в регистрах, так как $t_{ц} = 5$ мксек; $t_{ож} = 1,7 \div 2$ мксек; $t_{д} = 1,2$ мксек; $t_{сл} = 3,0$ мксек. При этом процесс базирования совмещается с процессом чтения констант из МОЗУ

Базис индекса. При выборке команды может быть индексирование адресов. В многопрограммной ЭВМ адрес индексной ячейки определяется по формуле

$$A_i^{16} = i^4 + B_n^{16}.$$

Если B_n будет иметь все 16 битов значащими, то необходимо тратить время на получение истинного адреса A_i^{16} путем суммирования. В этом случае значение B_n^{16} можно хранить только в ячейках памяти, так как необходимое время суммирования можно совмещать с тактом обращения к МОЗУ. Однако в этом случае индексирование будет состоять из двух этапов: формирования адреса и индексирования. Так как процесс индексирования — довольно часто встречающаяся процедура, то это существенно увеличивает время выполнения операции. Необходимо избавиться от этапа получения адреса индексной ячейки. В ЭВМ «Минск-32» для этого применяют 12-разрядный регистр B_n^{16} , который хранит старшие 12 разрядов базиса B_n^{16} . Получение же полного 16-битного адреса A_i^{16} осуществляется путем «пристегивания» к 12 старшим разрядам B_n 4-разрядного номера i^4 (рис. 31).

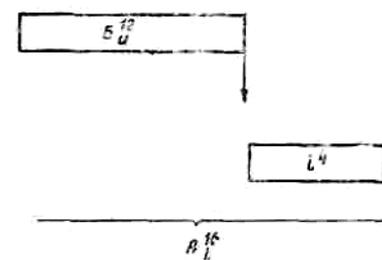


Рис. 31. Формирование адреса базисной ячейки.

Осуществляется эта процедура уже на кодовых шинах адреса путем одновременной выдачи содержимого B_n^{12} на 12 старших кодовых шин адреса и содержимого i^4 на 4 младшие кодовые шины адреса. Благодаря такой процедуре получения адреса сокращается время индексирования в 2 раза.

Константы защиты. Работа схемы защиты по вышеописанному алгоритму требует наличия в каждом такте обращения к МОЗУ констант Защ. Н, Защ. В. Выше уже упоминалось, что в современных ЭВМ для средней команды выше 50% времени отнимает время обращения к МОЗУ. Если процесс анализа защиты осуществлять, каждый раз читая константы Защ. Н. и Защ. В, то это почти вдвое увеличит время выполнения команды. Поэтому целесообразно хранить константы Защ. В и Защ. Н в специальных регистрах защиты. Остальные ячейки уровня как раз и предназначены для хранения информации регистровой части процессора в моменты прерывания, поэтому нет смысла рассматривать вопрос о регистровом дублировании этой информации.

3.4. НЕКОТОРЫЕ ВРЕМЕННЫЕ ПАРАМЕТРЫ МНОГОПРОГРАММНОЙ ЭВМ

Загрузка устройств и системное время

ЭВМ «Минск-32» является многопрограммной машиной, способной одновременно вести решение четырех рабочих программ. Прерывание, режим приостановок позволяют на оборудовании одного процессора организовать эффективный многопрограммный режим работы. На первых серийных образцах этой модели были измерены временные параметры загрузки устройств МОЗУ, АУ и системное время (время, необходимое на выполнение программ системы «Диспетчер»).

Для исследования были выбраны следующие задачи:

«Обращение матрицы» — большой удельный вес работы с МЛ («ОБРМ»).

«Решение дифференциальных уравнений методом Рунге-Кут-та» — большая доля операций с плавающей запятой («РДУ»).

«Трансляция» — большая доля логических операций и работы с МЛ.

Измерение производилось с помощью счетчиков в соответствующих точках процессора «Минск-32».

В табл. 3.2 приводится загрузка устройств АУ и МОЗУ. В графе «Время работы АУ» приведено время работы АУ только над операцией, исключая время модификации команды. Все замеры производились в 2-программном режиме, когда рассматриваемые задачи решались совместно с программой счета «Постоянная загрузка» («ПЗ»).

Таблица 3.2

Загрузка устройства АУ и МОЗУ во время решения задачи на ЭВМ «Минск-32»

Тип задачи	Полное время решения задачи	Время работы МОЗУ	Время работы АУ*	% занятости МОЗУ	Общий % занятости АУ**
„ОБРМ“	4 час. 45 мин.	4 час. 36 мин.	8 мин.	97	80
„РДУ“	53 мин. 50 сек.	27 мин. 34 сек.	26 мин. 16 сек.	51	95
„Трансляция“	12 мин. 29 сек.	12 мин. 24 сек.	5 сек.	99	80

* Без учета выборки команды.

** С учетом выборки и обработки команды.

В задаче «ОБРМ» при большом количестве обмена с магнитной лентой (в обмене участвует $3,5 \cdot 10^8$ битов) много тактов приостановки и тактов обращения к МОЗУ во время выполнения логических операций. Доля занятости МОЗУ составляет 97%. Это соответствует почти непрерывному потоку информации. Вторая задача «РДУ» имеет большую долю длинных операций с плавающей запятой. За счет этого возрастает загрузка арифметического устройства и снижается загрузка МОЗУ. При 3-программном режиме решения задач «ПЗ», «ОБРМ» и «РДУ» последняя успевает решиться 4 раза без увеличения времени решения задачи «ОБРМ». Системное время подсчитывалось путем суммирования времени работы процессора на уровнях $0 \div 5$. На задачах чистого счета, где очень мало связи с внешними устройствами, доля системного времени очень мала (около 0,25%). На задачах связи с ВнУ («Трансляция») доля системного времени увеличивается до 5,5%.

Большие величины системного времени могут быть при обслуживании быстродействующих ВнУ. При каждом обращении к внешнему устройству происходят два выхода на программы системы «Диспетчер»: при пуске ВнУ и анализе окончания обмена (рис. 32).

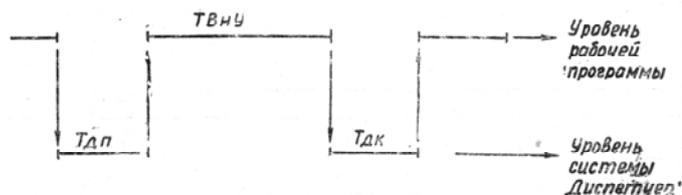


Рис. 32. Взаимодействие рабочей программы и программ системы «Диспетчер» при работе с внешними устройствами.

Время системного взаимодействия определяется коэффициентом системности η_c из следующего выражения

$$\eta_c = \frac{T_{дп} + T_{дк}}{T_{ВнУ}} \cdot 100\%$$

где $T_{дп}$ — время программы системы «Диспетчер» по пуску ВнУ; $T_{дк}$ — время программы системы «Диспетчер» при окончании работы ВнУ; $T_{ВнУ}$ — время работы ВнУ.

Коэффициент системности η_c определяет долю системного времени. Величины $T_{дп}$ и $T_{дк}$ постоянны для ЭВМ «Минск-32». Следовательно, η_c зависит от $T_{ВнУ}$, которое различно для различных внешних устройств. Наибольшее значение η_c принимает при работе с магнитной лентой.

Так, в ЭВМ «Минск-32» для средней длины зоны обмена в 2048 символов для накопителя НМЛ-67:

$$T_{\text{ВНУ}} = 45 \div 50 \text{ мсек}, \quad T_{\text{дп}} \approx 3 \text{ мсек}, \quad T_{\text{дк}} \approx 4,2 \text{ мсек},$$

$$\eta_c = \frac{3 + 4,2}{50} \approx 0,15$$

Для машины «Минск-32» эту цифру можно считать пределом η_c , т. е. если вся задача будет состоять только из работы с магнитной лентой, то 15% работы процессора будет потрачено на обслуживание этой работы.

Из результатов измерений видно, что η_c для средней задачи с магнитной лентой не превышает 5,5%.

Коэффициент многопрограммности

Для определения коэффициента многопрограммности рассмотрим диаграмму многопрограммного и однопрограммного режимов работы, приведенную на рис. 33.

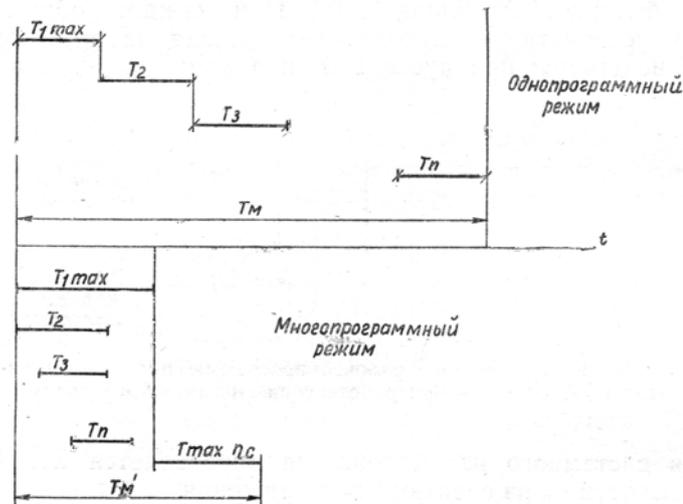


Рис. 33. Диаграмма однопрограммного и многопрограммного режимов.

Пусть имеется n задач с временами решений $T_1, T_2, T_3, \dots, T_n$. Задачи решаются в однопрограммном режиме путем последовательного пуска. Общее время решения в однопрограммном режиме T_m будет равно

$$T_m = \sum_{i=1}^n T_i$$

Пусть T_i будет иметь максимальное время решения задачи. В многопрограммном режиме за период решения задачи с максимальным временем будут поставлены остальные задачи, при этом $T_{i \text{ max}}$ увеличится на время $(T_{i \text{ max}} \cdot \eta_c)$ за счет потерь на системное взаимодействие и на величину $\Delta t_{\text{ож}}$ за счет ожидания освобождения каналов и накопителей. Таким образом

$$T'_m = T_{i \text{ max}} + T_{i \text{ max}} \cdot \eta_c + \Delta t_{\text{ож}} = T_{i \text{ max}} \cdot (1 + \eta_c) + \Delta t_{\text{ож}}.$$

Коэффициент многопрограммности $K_{\text{мп}}$ характеризует увеличение производительности ЭВМ за счет режима многопрограммности и определяется выражением

$$K_{\text{мп}} = \frac{T_m}{T'_{1m}} = \frac{\sum_{i=1}^n T_i}{T_{i \text{ max}} \cdot (1 + \eta_c) + \Delta t_{\text{ож}}}$$

Для ЭВМ «Минск-32» максимальное значение $K_{\text{мп}}$ получается при $n = 4$ и $T_1 = T_2 = T_3 = T_4 = T_{\text{max}}$, $\Delta t_{\text{ож}} \rightarrow 0$.

$$K_{\text{мп}} = \frac{4}{1 + 0,15} = 3,50$$

Как видно из вышеизложенного в однопрограммных ЭВМ недоиспользуется оборудование процессора в процессе решения задачи. Так в ЭВМ «Минск-22» АУ используется в среднем 66%, а МОЗУ — 40% времени решения задачи. Применение многопрограммности увеличивает загрузку

устройств. В ЭВМ «Минск-32» благодаря многопрограммности загрузка МОЗУ повышается до 97%, а АУ до 95%.

Наиболее эффективным методом повышения загрузки и производительности однопроцессорной ЭВМ является многопрограммность. При этом многопрограммность однопроцессорной ЭВМ наиболее эффективно решается системой прерывания. Для экономии оборудования, необходимого для организации прерывания, управляющую информацию можно располагать в отдельной части МОЗУ (УОП). Производительность ЭВМ не уменьшается, если управляющая информация текущей программы частично располагается в регистровом оборудовании процессора, частично в ячейках управляющей области МОЗУ, что значительно экономит оборудование процессора. В табл. 3.3 и 3.4 приведены экспериментальные данные, характеризующие режим многопрограммности ЭВМ «Минск-32».

Для осуществления многопрограммного режима однопроцессорная ЭВМ должна иметь аппаратуру защиты программ от взаимного влияния:

- жесткую защиту по постоянным признакам;
- защиту по переменным признакам для рабочих программ;
- защиту переменных массивов обмена с ВЛУ.

В рассмотренном варианте многопрограммности для ЭВМ «Минск-32» оптимальной схемой защиты с точки зрения оборудования, не замедляющей быстродействие процессора, является схема защиты по листам с использованием констант верхней и нижней границ в совокупности со схемой переполнения при базировании.

Как известно, для режима многопрограммности обязательным является обеспечение инвариантности программ, что достигается с помощью специальных базисных констант для рабочих полей и полей индексных ячеек. Рассмотренные схемы базирования сокращают этап базирования при незначительных затратах оборудования, а схема организации ЭВД существенно экономит оборудование.

Реализованный в ЭВМ «Минск-32» способ позволяет оптимально организовать режим многопрограммности однопроцессорной ЭВМ широкого применения. Блок организации многопрограммности, занимающий около 3,3% всего электронного оборудования машины, позволил в некоторых случаях увеличить производительность машины в 3,5 раза.

Таблица 3.3

Загрузка устройств АУ и МОЗУ во время решения задачи на ЭВМ «Минск-32»

Тип задачи	Полное время решения задачи	Время работы МОЗУ	Время работы АУ	%занятости МОЗУ
„ОБРМ“	4час.45мин.	4час.36мин.	8мин.	99
„РДУ“	53мин.50сек.	27мин.34сек.	26мин.16сек.	51
„Трансляция“	12мин.29сек.	12мин.24сек.	5сек.	99

Таблица 3.4

Распределение времени процессора между программами в двухпрограммном режиме

Тип задачи	Рабочая программа (в %)	Вторая задача чистого счета (в %)	Системное время (в %)	Характеристика рабочей программы
„РДУ“	87,75	12	0,25	Преобладание счета с выводом на печать
Трансляция „ПИР“	3,4	91,4	5,1	Преобладание обмена память—НМЛ
Трансляция „Печать“	3,8	90,3	5,9	То же
Трансляция „ВТИПЛ“	4	91,5	4,5	»

СИСТЕМА УНИВЕРСАЛЬНОЙ СВЯЗИ С ВНЕШНИМИ УСТРОЙСТВАМИ (СУС ВнУ)

Под системой СУС ВнУ понимается универсальная система связи, характеризующаяся наличием определенного набора сигналов, идентичностью разъемов сопряжения и электрических сигналов и одной и той же временной диаграммой взаимодействия для всех внешних устройств независимо от их типа.

Учитывая необходимость получения максимальной производительности ЭВМ на определенном комплексе оборудования и совместимости различных конфигураций, можно сформулировать следующие основные требования, предъявляемые к организации связи процессора с внешними устройствами:

- совмещение во времени работы процессора над программой и обмена информации с внешними устройствами;
- возможность одновременного ведения обмена с большим количеством внешних устройств;
- максимальное использование возможностей и быстродействия процессора при организации потока обменной информации;
- возможность гибкого наращивания большого количества внешних устройств;
- универсальность способов пуска внешних устройств, анализа их состояний, реакций на сигналы этих устройств, процесса обмена информацией.

Наиболее полно вышеперечисленные требования могут быть удовлетворены, если принять следующий принцип организации связи процессора и внешних устройств:

- обмен информацией между процессором и внешними устройствами осуществляется в режиме приостановки программы;
- пуск внешних устройств на отработку конкретной операции и опрос его состояния производится специальной универсальной командой;
- осуществляется программная реакция процессора на сигналы конца работы внешнего устройства, его сбоя.

Исходными данными при проектировании логических схем связи с внешними устройствами должны быть:

- быстродействие процессора $f_{оп}$, определяющееся тактом обращения к активной памяти;
- быстродействие внешних устройств $F_{ВнУ}$, определяющееся периодом следования запросов на обмен.

Наиболее оптимальным вариантом реализации вышеизложенных требований является организация двух каналов связи: селекторного (СК) и мультиплексного (МК).

Если $K = \frac{F_{ВнУ}}{f_{оп}}$, то медленными внешними устройствами называются устройства, у которых

K выше первого порядка, а быстрыми — у которых K — величина первого порядка. Медленные устройства подключаются к мультиплексному каналу, быстрые — к селекторному.

Все медленные устройства должны работать одновременно, и их количество определяется суммарной частотой всех запущенных устройств. Общая частота работающих внешних устройств не должна превышать пропускной способности мультиплексного канала.

В селекторном канале может работать только одно быстрое устройство. Определяется это в основном оборудованием этого канала и тем, что частота таких внешних устройств уже соизмерима с циклом оперативного накопителя машины.

4.1. КАНАЛЫ СВЯЗИ

В процессоре многопрограммной ЭВМ «Минск-32» (см. рис. 25) работой двух каналов управляет каналный микропроцессор, представляющий устройство обмена (УО). На рис. 34 приведена схема, подсоединения внешних устройств к двум каналам УО.

Все устройства мультиплексного канала могут работать одновременно, и количество одновременно работающих устройств определяется их суммарной частотой, которая не должна превышать пропускной частоты этого канала.

В селекторном канале может работать по обмену информацией одновременно с процессором только одно быстрое устройство. Определяется это в основном оборудованием этого канала и тем, что частота быстрых ВнУ уже соизмерима с циклом оперативной памяти ЭВМ.

Структура связи мультиплексного канала такова, что, кроме семи, внешних устройств основного комплекта, подключенных к нулевой группе, можно к этой же группе подключить дополнительно еще одно устройство. Три дополнительные группы мультиплексного канала позволяют дополнительно подсоединить либо непосредственно 3 устройства (по 1 устройству на каждую группу), либо 3 группы коммутаторов, каждая из которых может подключить до 32 ВнУ.

Таким образом, общее количество ВнУ мультиплексного канала определяется

$$N_M = 7_{\text{осн}} + 1_{\text{доп}} + n_1 + n_2 + n_3,$$

где N_M — общее количество ВнУ канала МК;

n_1, n_2, n_3 — количества ВнУ дополнительных групп МК, принимают значения от 0 до 32.

Максимальное количество ВнУ мультиплексного канала равно 104.

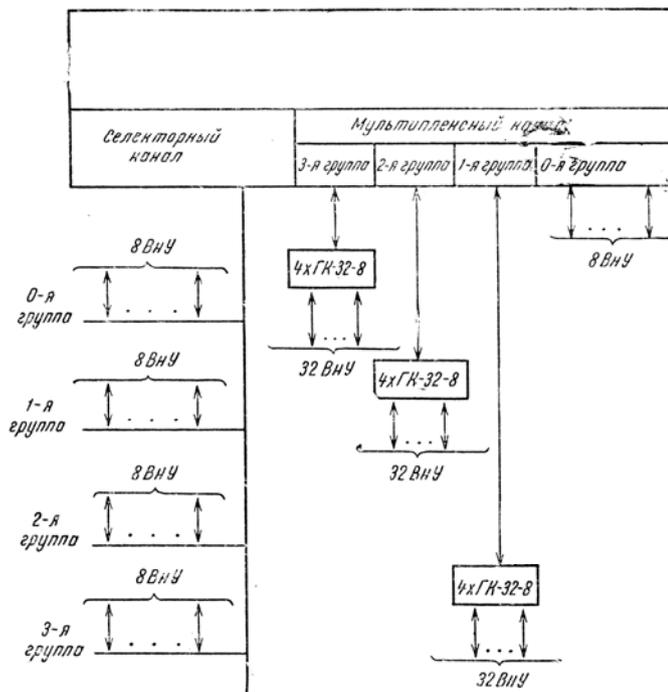


Рис. 34. Схема подключения внешних устройств к каналам ЭВМ.

К нулевой группе СК можно подключить в дополнение еще 3 ЛПМ. В каждой из трех дополнительных групп селекторного канала можно подключать устройство, управляющее работой до восьми накопителей на магнитной ленте, магнитном барабане и т. п.

Таким образом, общее количество устройств селекторного канала определяется

$$N_c = 5 + n_{\text{осн}} + m_1 + m_2 + m_3,$$

N_c — общее количество ВнУ СК ($N_{c \text{ max}} = 32$);

$n_{\text{осн}}$ — количество дополнительных ВнУ основной группы;

$n_{\text{осн max}} = 3$;

m_1, m_2, m_3 — количества ВнУ дополнительных групп селекторного канала принимают значения от 0 до 8.

Все внешние устройства ЭВМ «Минск-32» являются устройствами ввода-вывода информации в оперативную память и из оперативной памяти. К устройствам ввода-вывода относятся накопители аппарата передачи данных, считывающие и перфорирующие механизмы, пишущая машинка, печатающее устройство и т. д.

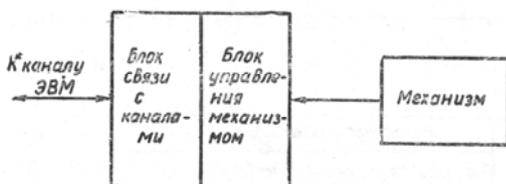


Рис. 35. Блок-схема внешнего устройства.

По характеру работы все внешние устройства ЭВМ «Минск-32» можно разбить на пассивные и активные. Пассивные ВнУ начинают обмен информацией только по команде из ЭВМ. Активные ВнУ могут быть сами инициаторами обмена, а также управляться командами из ЭВМ. По своей структуре все ВнУ идентичны

По своей структуре все ВнУ идентичны

и в основном должны удовлетворять блок-схеме, приведенной на рис. 35.

Блок связи с каналом у всех внешних устройств идентичен и должен соответствовать стандартному алгоритму работы соответствующего канала. Благодаря наличию стандартного сопряжения последовательность сигналов одинакова для всех устройств одного канала. Блок управления механизмом специфичен для каждого типа ВнУ и отражает особенность каждого механизма. В частном случае это может быть аппаратура сопряжения с каким-то генератором информации (датчики, каналы).

Частотные характеристики СК и МК

Так, если при цикле $t_{вч}$ оперативной памяти ЭВМ на один символ тратится время $t_{уд}$, то коэффициент взаимодействия

$K_{вз} = \frac{t_{вч}}{t_{уд}}$ показывает, что за одно обращение к оперативной памяти пишется-читается $K_{вз}$ единиц

информации обмена. При этом, загрузка обоих каналов выражается формулой

$$\frac{f_{вубк}}{K_{вз1}} + \frac{f_{вусмк}}{K_{вз2}} \leq f_{оп},$$

где $K_{вз1}$ — коэффициент взаимодействия селекторного канала;
 $K_{вз2}$ — коэффициент взаимодействия мультиплексного канала;
 $f_{оп}$ — быстродействие оперативной памяти;
 $f_{вубк}$ — быстродействие устройства селекторного канала;
 $f_{вусмк}$ — суммарное быстродействие ВнУ мультиплексного канала.

$$f_{вусмк} = f_{вУ1} + f_{вУ2} \dots + f_{вУm},$$

где $f_{вУ1}, f_{вУ2}, f_{вУm}$ — быстродействие первого, второго, m -го внешних устройств мультиплексного канала.

Быстродействие устройств мультиплексного канала, обладающего старт-стопными устройствами обмена информацией, не принимается в расчет. Количество таких устройств, работающих параллельно с расчетными, ограничивается лишь возможностями подсоединения к процессору.

В машине «Минск-32» при $f_{оп}$, равном 200 кГц, $K_{вз1}$, равном 1, $K_{вз2}$, равном $\frac{1}{3}$, максимально возможное быстродействие внешнего устройства селекторного канала равно 200 кГц, а мультиплексного — 66 кГц.

Допустимое быстродействие внешнего устройства СК определяется из формулы

$$V_{внУСК} \leq \frac{200}{1+K} \text{ кГц},$$

где K — коэффициент инерционности взаимодействия канала с оперативной памятью ЭВМ из-за занятости последней. Обычно $K = 1$, так как всегда возможны случаи занятости оперативной памяти в процессе выполнения команд обращения к ВнУ или операций над символами. При создании специализированного комплекса на базе ЭВМ «Минск-32», требующего специализированной программы-диспетчера, организующей монопольные режимы обмена информацией с внешними устройствами, коэффициент K приобретает значение 0. Допустимое быстродействие каждого ВнУ МК зависит от уровня его приоритета, наличия одновременного обмена с ВнУ СК, инерционности взаимодействия МК и определяется по формуле

$$V_{внУМК} \leq \frac{600}{10 \left(K + \frac{V_{внУСК}}{200 - V_{внУСК}} \right) + 3(3m + 3 - K)},$$

где m — уровень приоритета данного ВнУ МК среди одновременно работающих устройств на

канале. Если ВнУ — самое приоритетное, $m = 0$, если перед ним есть одно более приоритетное устройство, $m = 1$ и т. д.

В табл. 4.1 приведены параметры быстроедействий ВнУ СК и МК для нескольких вариантов конкретных условий, влияющих на них.

Таблица 4.1

ВнУ СК, кГц	ВнУ МК, кГц	Мультиплексный канал	
		K	m
200	0	0	
0	60	0	0
0	40	1	0
100	30	0	0
100	20	0	1
100	18	1	0
100	15	1	1

Вышеприведенные параметры должны учитываться при подключении только устройств, не обладающих старт-стопными свойствами.

4.2. СИСТЕМА ПОДКЛЮЧЕНИЯ ВнУ

Как уже упоминалось, связь процессора с ВнУ как МК, так и СК выполнена по универсальному принципу, заключающемуся в наличии определенного набора сигналов и одной и той же временной диаграммы взаимодействия для всех внешних устройств независимо от их типа. Всем внешним устройствам присваивается номер, являющийся адресом этих внешних устройств. Номер ВнУ зависит от номера разъема УО, к которому подключено данное ВнУ.

Подключаемые ВнУ МК делятся на 4 группы: основную и 3 дополнительные. Основная группа состоит из восьми ВнУ, подключаемых непосредственно к процессору и имеющих номера $40_8 + 47_8$ (индекс 8 означает нумерацию в восьмеричной системе счисления). Семь номеров этой группы используются для подключения семи ВнУ основного комплекта ЭВМ. Каждая из трех дополнительных групп может работать с 32 ВнУ, подключение которых производится через специальные групповые коммутаторы.

ВнУ первой дополнительной группы имеют номера $100 \div 137$, второй — $140 \div 177$ и третьей — $200 \div 237$. Вместо каждого из трех групповых коммутаторов можно непосредственно подключить по одному ВнУ с номерами 100, 140, 200. Приоритет с точки зрения удовлетворения запросов падает по мере увеличения его номера (ВнУ с номером 40 самое приоритетное на МК, а с номером 237 — имеет самый малый приоритет), что необходимо учитывать при подключении с МК сравнительно быстрых ВнУ.

Подключение ВнУ к СК блочное, т. е. имеется возможность подключения четырех блоков, каждый из которых имеет 8 номеров (ВнУ) и единое управление ими. ВнУ первого блока имеет номера $00 \div 07$, второго — $10 \div 17$, третьего — $20 \div 27$ и четвертого — $30 \div 37$.

4.3. МАССИВЫ ОБМЕНА С ВНЕШНИМИ УСТРОЙСТВАМИ

По каждому адресу в ячейке МОЗУ содержится 38 двоичных разрядов, из которых 37 являются информационными и 1 контрольным, дополняющим число единиц в машинном слове до нечетного. Обмен с МОЗУ можно производить не только 37-разрядными словами, но и 7-разрядными символами. В основном обмен символами осуществляется при работе с ВнУ. На рис. 36 приведены 37 информационных разрядов ячейки МОЗУ с разбивкой на 6 символов.

Символы пронумерованы слева направо и имеют номера 0, 1, 2, 3, 4, 5. Символы $0 \div 4$ являются полными, и каждый содержит 7 разрядов. Символ с номером 5 включает только 35-й и 36-й разряды машинного слова. Все внешние устройства, независимо от типа, используют для обмена с процессором 8-разрядный символ, который является основой организации

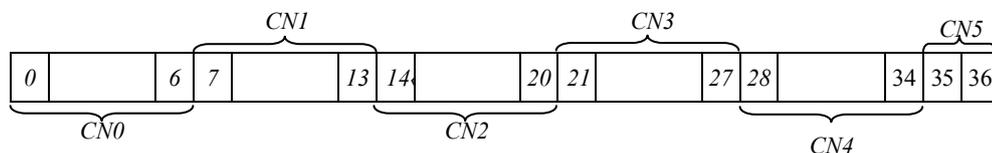


Рис. 36. Расположение информации 6 символов в 37-разрядном слове.

универсальной связи с ВнУ (структура символа приведена на рис. 6).

Символы считываются или записываются из машинного слова слева направо. При увеличении адреса очередного символа, если исчерпана емкость машинного слова, происходит переход на следующее машинное слово с продвижением по нему слева направо. Таким образом, последовательное увеличение адреса символа соответствует продвижению слева направо по словам, снизу вверх — по адресам.

При работе с внешними устройствами существуют два следующих режима компоновки 7-битным символом 37-разрядного машинного слова: по 5 символов в одном слове; по 6 символов в одном слове. На рис. 29 показано размещение алфавитно-цифровой информации в 37-разрядной памяти. В одном машинном слове укладывается 5 семиразрядных символов, каждый 6-й символ начинает новое машинное слово. 35-й и 36-й разряды в этом случае в работе не участвуют.

На рис. 21 показано размещение двоичной информации (команды, операнды) в массиве памяти для обмена с внешними устройствами. В одно машинное слово записывается или считывается 6 символов. Каждый шестой символ имеет только два значащих разряда и 5 фиктивных, всегда нулевых. Каждый седьмой символ начинает новое машинное слово. Работа в этом случае происходит как бы с 42-разрядным словом, имеющим всегда в разрядах $37 \div 41$ нулевую информацию.

4.4 ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОРА С ВНЕШНИМИ УСТРОЙСТВАМИ

Взаимодействие процессора с ВнУ можно разбить на 2 независимых процесса: командное взаимодействие; информационное.

Командное взаимодействие — это работа программно-аппаратурного комплекса, в составе которой лежит процесс выполнения команд связи с ВнУ с использованием прерывания. Результатом командного взаимодействия является определение состояния ВнУ (в частности, возможности последующей организации обмена информацией с нужным ВнУ), осуществление пуска выбранного ВнУ на конкретный вид обмена. При командном взаимодействии (см. рис. 25) из АУ и ЦУ служебная информация передается в устройство обмена, которое выбирает канал и нужное ВнУ, задает режим работы выбранного устройства. В результате командного взаимодействия из информационных трактов работает только 37-разрядный тракт обращения АУ + ЦУ к МОЗУ и тракты соответствующих каналов по передаче служебной информации во ВнУ.

Имеются два вида командного взаимодействия.

Один из видов командного взаимодействия — определение возможности организации обмена информацией с нужным ВнУ — заключается в опросе состояния данного ВнУ, для чего каждое ВнУ имеет набор указателей, определяющих его состояние. Обязательными для всех ВнУ должны быть следующие указатели: «Готово», «Занято», «Сбой». Кроме них, каждое конкретное ВнУ может иметь свои специфичные указатели (нестандартные), дополняющие сведения о его состоянии. Например, для магнитной ленты такие указатели, как: «Не вся зона считана», «Конец ленты»; для устройства ввода с перфокарт — «Пустой карман» и т. д.

Общее количество указателей не должно превышать шести. Указатели ВнУ делятся на активные и пассивные. К активным указателям следует относить те, которые указывают на состояние ВнУ, явившееся результатом только данной конкретной операции, например, из стандартных — «Сбой». Активные указатели должны сбрасываться при первом же опросе их из процессора. Пассивные указатели определяют состояния ВнУ, связанные с длительными временными интервалами. Эти указатели должны оставаться в единичном состоянии до устранения данной ситуации и не сбрасываться при опросе. Из стандартных указателей к ним относятся «Готово» и «Занято» (устройство остается занятым работой с процессором на время отработки полученной им команды, или устройство не готово до устранения причины неготовности).

Для опроса состояний указателей ВнУ используется команда ИУВУ («Идти по указателю внешнего устройства»). При обращении к ВнУ СК предварительно опрашивается состояние самого канала: занят он или свободен. Операции связи с ВнУ СК возможны только при условии, что СК не занят очередным обменом. При соответствии состояния ВнУ условию пуска производится программный пуск ВнУ.

Вторым видом командного взаимодействия является процесс пуска ВнУ на отработку конкретной разновидности обмена, который осуществляется командой ВВИ («Ввести — вывести

информацию»). Под пуском ВнУ подразумевается сообщение ему нужного кода команды обмена (определителя) и возможности начала организации непосредственного обмена.

4.5. ПРИОСТАНОВКА

В основе информационного взаимодействия лежит непосредственный обмен информацией в режиме приостановки. После выполнения команды связи с внешним устройством, производящей пуск механизма внешнего устройства и подготовку массива памяти МОЗУ к обмену (засылка управляющего слова в соответствующую ячейку управляющей области памяти), процессор приступает к следующей команде. Запущенное ВнУ посылает в нужные моменты (например, при вводе перфоленты при каждом прохождении очередной дорожки с пробивками под головкой считывающих фотодиодов) запросы о необходимости получения или выдачи очередного символа.

Приостанавливается работа в конце очередного такта АУ или ЦУ при выполнении любой текущей команды, и начинают выполняться такты приостановки УО: выбирается информация управляющего слова, определяется режим обработки появившегося запроса ВнУ, в случае необходимости пишется в память или считывается из нее символ обмена и, если необходимо, записывается в память промодифицированное управляющее слово. После тактов приостановки, выполнение которых производило УО, управление передается АУ + ЦУ для продолжения выполнения тактов приостановленной команды. С частотой запросов внешнего устройства вышеописанная ситуация повторяется (в случае фотоввода — с частотой порядка 1 кГц). При работе процессора над программой одновременно в память разрешается доступ нескольких ВнУ в режиме приостановки. Приостанавливается любая команда любой программы, в том числе и команды связи с ВнУ.

Таким образом, инициаторами ведения информационного обмена являются сами ВнУ. Обмен информацией в режиме приостановки селекторного и мультиплексного каналов происходит поразному. Непосредственный обмен информацией между ВнУ МК и МОЗУ производится в режиме приостановки с ожиданием доработки очередного такта, в момент которого пришел запрос. При возникновении запроса на обмен информации ВнУ МК (одного или нескольких) ЦУ + АУ приостанавливают выполнение очередных тактов команд программы на время, необходимое для непосредственного обмена информацией с этим ВнУ (или несколькими ВнУ).

Непосредственный обмен информацией между ВнУ СК и МОЗУ также организуется по запросам ВнУ, но в этом случае обмен происходит без ожидания, т. е. устройство выдает символ в процессор вместе с запросом при вводе и получает символ из процессора в ответ на запрос при выводе. Эта разница определяется тем, что СК имеет небольшую буферную память (на 3 символа) с возможностью непосредственного доступа из ВнУ, асинхронного с работой процессора. Позволяя ВнУ вести безинерционный обмен информацией со своей буферной памятью (БН), СК сам организует запрос на обмен с МОЗУ (как и ВнУ МК, только с более высоким приоритетом удовлетворения). Этот обмен между МОЗУ и БП СК продолжается до полной загрузки БП информацией при выводе и полной разгрузки при вводе.

Обмен информацией (символом) между ВнУ МК и МОЗУ называется медленной приостановкой. Обмен информацией между БП БК и МОЗУ (символом) называется быстрой приостановкой.

Как уже упоминалось, компоновка массивов обмена ведется по управляющим словам (УС). Каждому ВнУ соответствует свое УС, хранящееся в соответствующей ячейке УОП (дежурном регистре). При информационном взаимодействии между ВнУ МК и процессором УС из соответствующей ВнУ ячейки УОП, в которой УС хранится в промежутках между непосредственным обменом, извлекается в специальный оперативный регистр УС, согласно состоянию которого производится управление процессом обмена. После обмена информацией модифицированное УС (модифицируется адрес Ан) отсылается в свою ячейку УОП. Так как УОП расположена в памяти МОЗУ машины, то по вышеописанному принципу приостановки процесс обмена информацией между процессором и ВнУ МК определяют 3 основные операции:

- чтение УС в оперативный регистр из УОП МОЗУ;
- обмен информацией, анализ и модификация адреса обмена Ан;
- запись модифицированного УС в УОП.

Таким образом, при непосредственном обмене единицей информации требуется 3 обращения к МОЗУ. Медленная приостановка «Минск-32» длится 15 мксек. Это определяет пропускную

способность, быстродействие МК — 66 кци. При полной загрузке по быстродействию МК не остается времени для удовлетворения запросов на обмен от устройств СК в случае их параллельной работы.

Обмен информацией СК организован таким образом, что одна быстрая приостановка требует одного обращения к оперативной памяти и тратится на это 5 мксек. Достигается это тем, что в СК имеется оперативный триггерный регистр УС. А так как на СК работает только одно устройство, то УС хранится в триггерном регистре УС в течение обмена. Максимальная частота СК равна 200 кци, при этом не может работать ни одно устройство МК.

Окончание обмена может определяться как процессором так и ВнУ (в зависимости от специфики режима обмена). Оно характеризуется тем, что ВнУ перестает вырабатывать запросы и переходит на отработку внутреннего цикла, т. е. подготовку к очередному обращению из процессора. После отработки внутреннего цикла ВнУ устанавливает в нулевое состояние указатель «Занято» и вырабатывает сигнал «Конец работы», который поступает в процессор и является причиной автоматического переключения его на 4-й уровень (уровень окончания работы ВнУ). Активное ВнУ МК при окончании режима ввода информации вырабатывает сигнал «Запрос пр.» — запрос прерывания на специальный уровень программы «Диспетчер», где расположен блок работы с активными ВнУ (3-й уровень).

Программа «Диспетчер» (в случае необходимости, совместно с рабочей программой) определяет по состоянию ВнУ правильность выполнения операции обмена. После этого обмен считается завершенным.

4.6. ВЫПОЛНЕНИЕ ОПЕРАЦИИ ВВОДА-ВЫВОДА

Команды ввода-вывода

Командное взаимодействие в ЭВМ «Минск-32» со всеми 136 ВнУ осуществляется с помощью трех команд:

ИУВУ (идти по указателю ВнУ, код «—65»);

НВУИ (найти внешнее устройство по признаку и идти, код «— 66»);

ВВП (ввести-вывести информацию, код «—67»).

Все команды взаимодействия процессора с ВнУ имеют универсальную структуру представления в машинном слове, как показано на рис. 37.

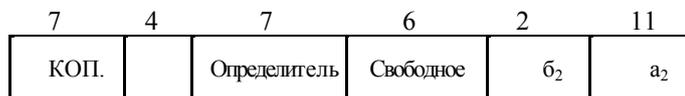


Рис. 37. Структура команд связи с ВнУ.

КОП — код операции;

i — номер индексной ячейки;

Определить — определитель команды;

Свободные — свободные разряды;

б₂ — номер базиса для адреса а₂;

а₂ — относительный адрес. В команде ВВП он указывает адрес управляющего слова (УС), в командах ИУВУ и ИВУИ — адрес программного перехода.

Для реализации универсального принципа обращения к ВнУ номер ВнУ, к которому производится обращение, указывается в 13 ÷ 20-м разрядах индексной ячейки (рис. 38) и в результате предварительной обработки команды (индексирования) получается на сумматоре (в разрядах 13 ÷ 20 См. соответственно). Поэтому индексирование во всех командах обращения к ВнУ обязательно, т. е. содержимое разрядов i не должно быть равно «0000».

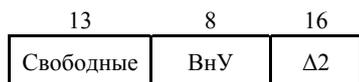


Рис. 38. Структура i -й ячейки для команд связи с ВнУ.

Управляющие слова (УС)

Для формирования массивов и управления процессом обмена между процессором и ВнУ в ЭВМ «Минск-32» используются два типа УС. Они отличаются признаком окончания обмена. В первом типе УС этим признаком является рабочий информационный символ, код которого

задается в информации УС (рис. 39).

1	1	3	7	1	1	7	16
r	0	N_c	A_3	P	0	K	A_n

Рис. 39. УС с признаком конца до символа.

где r — указатель количества символов в каждом обменном машинном слове: при $r = 0$ каждое обменное слово состоит из пяти символов; при $r = 1$ каждое обменное слово состоит из шести символов (6-й неполный);

N_c — номер очередного символа обменного слова ($0 \div 4$ или $0 \div 5$, в зависимости от значения r);

A_n — адрес очередного обменного слова в МОЗУ;

A_3 — адрес защиты сверху по листам (возможен обмен по адресу A_n со значением его старших семи разрядов, не превышающих значения A_3);

K — символ — признак конца обмена (определяется в этом случае совпадением кодов заданного признака и текущего обменного символа);

P — указатель, определяющий участие в обмене символа, совпадающего с K : при $P = 0$ этот символ не участвует в обмене, т. е. не записывается в память и не наносится на носитель ВнУ; при $P = 1$ символ подлежит обмену.

В процессе обмена компоненты N_c и A_n подвергаются модификации. Таким образом, если перед началом обмена были заданы начальные значения ($A_{но}$ и $N_{со}$), то после обмена эти компоненты имеют конечное значение ($A_{нк}$ и $N_{ск}$), что является адресом очередного символа, следующего за последним символом, участвующим в обмене. При окончании обмена информацией по причине сбоя $A_{нк}$ и $N_{ск}$ соответствуют адресу символа, в процессе обмена которых произошел сбой.

Во втором типе УС признаком конца обмена является полноразрядный адрес A_k (16 двоичных разрядов). Этот адрес одновременно является и полноразрядной границей защиты сверху. Структура УС приведена на рис. 40.

1	1	3	16	16
r	1	C	A_k	A_n

Рис. 40. УС с признаком конца до адреса.

Обмен оканчивается, если текущее значение A_n оказалось равным A_k , т. е. A_k является адресом первого необменного слова ($A_{нк} = A_k, N_c = 0$).

В процессе выполнения команды ВВИ (пуска ВнУ), если устройство оказалось незанятым, УС из ячейки с адресом A_2 (адрес указан в команде) переписывается в специальный дежурный регистр хранения УС, расположенный в УОП с адресом, соответствующим номеру пускаемого ВнУ (рис. 41).

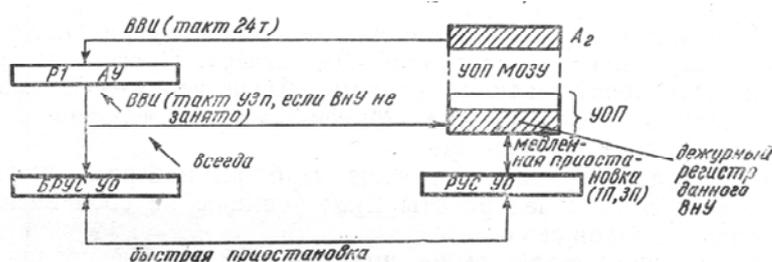


Рис. 41. Диаграмма пуска внешнего устройства.

При работе с ВнУ СК УС помещается еще и в специальный регистр хранения УС СК (БРУС).

В процессе информационного обмена в режиме приостановок с ВнУ МК происходит чтение УС из соответствующих дежурных регистров для организации непосредственного обмена, модификация их адресной части и последующая запись УС (возврат) в дежурный регистр.

В процессе обмена с ВнУ СК УС читается не из дежурного регистра, а из БРУСа и затем отсылается в него (после соответствующей модификации адресной части). После окончания обмена с ВнУ СК его УС из БРУСа переписывается в дежурный регистр УОП. Таким образом, в дежурном регистре оказывается начальное значение УС при пуске ВнУ и конечное после обмена, а промежуточные значения хранятся в БРУСе.

Для хранения УС и информации уровней в МОЗУ выделен ряд ячеек. Это поле условно называется управляющей областью памяти (УОП). Весь объем УОП 256 машинных слов. Для хранения значений УС 136 ВнУ и электронного датчика времени (ЭДВ) используется часть УОП, содержащая 160 ячеек.

Расположение УС в дежурных регистрах УОП согласно номерам ВнУ приведено в табл. 4.2.

Таблица 4.2

ВнУ	Адреса ячеек УОП (восьмеричные)	Номера ВнУ (восьмеричные)
ВнУ СК.....	140 ÷ 177	0 ÷ 37
Основная группа МК.....	200 ÷ 207	40 ÷ 47
ЭДВ.....	210	50 (ЭДВ)
Свободные.....	211 ÷ 237	—
1-я группа МК.....	240 ÷ 277	103 ÷ 137
2-я группа МК.....	300 ÷ 337	140 ÷ 177
3-я группа МК.....	340 ÷ 377	200 ÷ 237

4.7. ПРОГРАММНОЕ ОБСЛУЖИВАНИЕ РАБОТЫ С ВНЕШНИМИ УСТРОЙСТВАМИ

Выполнение программ на машине «Минск-32» управляется системой обслуживающих программ «Диспетчер». Программы ЭВМ «Минск-22М» обслуживаются блоком «Диспетчера» — «Совместимость». При работе в режиме «Минск-32» диспетчеризацию осуществляет блок «Координатор».

Следует рассматривать три основных блока программ по связи с ВнУ: пуск ВнУ; конец работы ВнУ; реакцию на сбой внешних устройств и каналов связи.

Как уже упоминалось выше, имеются специальные уровни, обслуживающие соответствующие программы системы «Диспетчер» по связи с ВнУ. Это уровни сбоев (1 и 2), уровни пуска внешних устройств (5, 3), уровни анализа окончания работы ВнУ (4, 3). В основе включения нужных программ системы «Диспетчер» по связи с ВнУ лежит вышеописанный процесс прерывания. Все команды связи с ВнУ на уровнях 7 ÷ 1 3 являются экстракодовыми и служат причиной прерывания на соответствующие уровни (в режиме «М-32» — на уровень 5, в режиме «М-22М» и «М-22» — на уровень 6). На уровнях 0 ÷ 6 команды связи с ВнУ являются исполнительными.

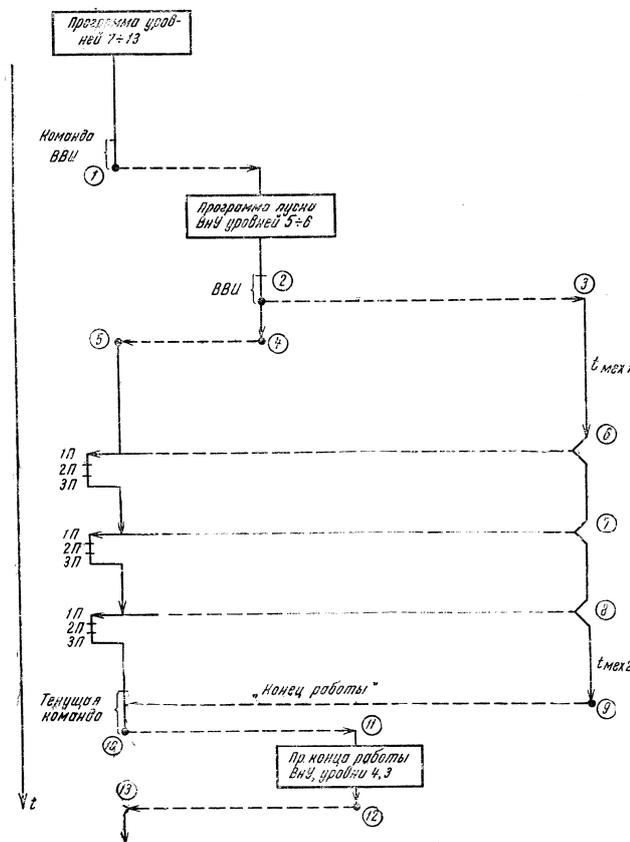


Рис. 42. Взаимодействие программ при работе с внешними устройствами.

Кроме того, сигналом прерывания при необходимости организации связи с ВнУ может служить

специальный сигнал «Запрос на прерывание» активных ВнУ (при этом происходит прерывание на уровень 3). В конце работы ВнУ вырабатывают сигнал «Конец работы», который является причиной автоматического прерывания на программы 4-го уровня (в случае активных ВнУ — на 3-й уровень по тому же сигналу «Запрос на прерывание»).

Взаимодействие программ при работе с ВнУ изображено на рис. 42. Сплошным вектором изображаются последовательности команд, представляющие собой любую программу. Пунктиром показан процесс прерывания. Приостановки изображены для иллюстрации независимости командного и информационного взаимодействия.

Пуск ВнУ

Рассмотрим последовательность команд (программу) уровней $7 \div 13$. После выполнения любой из команд связи с ВнУ («—65», «—66», «—67») в точке 1 (рис. 42) происходит прерывание работы программ уровней $7 \div 13$. Аппаратура процессора настраивается на выполнение программ уровня 5 или 6. В блоке программ 5-го уровня производится выполнение операций опроса состояния указателей и пуска ВнУ. Программы уровня 6 производят действия по определению состояния запрашиваемого ВнУ, выдаче УС, освобождению ВнУ, закреплению ВнУ за конкретной программой и т. д.

Если в точке 2 программы пуска ВнУ уровня 5 появилась команда ВВИ, происходит механический пуск выбранного внешнего устройства (точка 3). После прохождения программы уровней 5 и 6 до точки 4 происходит возврат (прерывание) в точку 5 прерванной программы уровней $7 \div 13$.

Запущенное ВнУ в точке 3, отработав время механического пуска $t_{\text{мех. 1}}$ до точки 6, начинает выставлять запросы на приостановки для организации информационного взаимодействия (точки 6, 7, 8). Процесс приостановки (такты 1П, 2П, 3П) вклинивается и раздвигает выполнение команд текущей программы, в нашем случае— это команды программ уровней $7 \div 13$.

Конец работы ВнУ

В точке 9 ВнУ, выполнив нужное количество приостановок и отработав время успокоения $t_{\text{мех. 2}}$, вырабатывает сигнал «Конец работы». В точке 10 после выполнения текущей команды и при отсутствии блокировки прерывания происходит прерывание на программу уровня 4 (точка 11). Для активных ВнУ происходит прерывание на уровень 3 (также точка 11). Программа конца работы ВнУ по сброшенному в «0» указателю «Занято» определяет закончившее работу ВнУ, причем проверяются только запущенные ранее ВнУ. Для окончивших работу ВнУ опрашиваются и записываются состояния всех указателей. После опроса состояния указателей определяется тип ВнУ и номер программы пуска этого ВнУ. В точке 12 управление передается одной из программ, в зависимости от состояния аппаратурно-программного блока прерывания. После возврата в точку 13 продолжается выполнение рабочей программы уровней $7 \div 13$.

Реакция на сбой ВнУ и каналов связи

Программы, обслуживающие сбойные ситуации ВнУ и каналов, выполняются после того, как определяется, что окончание работы связано со сбойной ситуацией. При неготовности ВнУ или при единичном состоянии указателя «Сбой» определяется программа, для которой осуществляется обмен со сбойным ВнУ. При возможности устранения последствий сбоя без вмешательства оператора управление передается на выполнение определенных действий по сбоям. При невозможности устранения последствий сбоя без вмешательства оператора на пишущую машинку оператора выводится соответствующее указание. После получения ответа оператора о выполненных им действиях управление передается программе, обслуживающей данную директиву.

4.8. ОСОБЕННОСТИ РАБОТЫ УСТРОЙСТВ МУЛЬТИПЛЕКСНОГО И СЕЛЕКТОРНОГО КАНАЛОВ

Выполнение команд связи с ВнУ

Команда ИУВУ (код «—65») является командой условного перехода по состоянию указателя ВнУ. Условие перехода и конкретный указатель (или группа указателей) указываются в

определителе команды (разряды 11 ÷ 17):

- 11 разряд — в единичном состоянии определяет переход по единичному состоянию указателя ВнУ, в нулевом — по нулевому состоянию указателя ВнУ;
- 12 разряд — в единичном состоянии определяет, что должен быть опрошен указатель «Готово» ВнУ;
- 14 разряд — определяет опрос указателя «Сбой»;
- 17 разряд — определяет опрос указателя «Занято».

Единичные состояния разрядов 13, 15, 16 соответствуют опросу нестандартных указателей.

Опрос указателей производится по шинам КШС ВнУ (кодовые шины символа) путем выдачи на них определителя в соответствии с табл. 4.3.

Если хотя бы один из опрашиваемых согласно определителю команды указатель ВнУ находится в единичном состоянии, последнее должно выработать ответный сигнал «Указатель есть». Задержка ответного сигнала при этом на выходе ВнУ относительно сигнала опроса на входе ВнУ (КШС ВнУ) не должна превышать 0,6 мксек. Указатель «Сбой» при опросе сбрасывается, если он находился в единичном состоянии.

Опрос указателей ВнУ МК допускается во всех случаях, за исключением тактов обмена информацией. Опрос указателей ВнУ СК возможен лишь в случае, если ВнУ не находится в состоянии обмена (указатель состояния обмена процессора с ВнУ СК находится в УО и называется «Указатель занятости СК»).

Команда реализуется последовательным выполнением следующих тактов:

→К→Инд→Б2→ОП1→ОП2→К.

При выполнении тактов Б2, ОП1, ОП2 запрещена организация приостановок, что позволяет достичь неразрывного и необходимого времени взаимодействия между ВнУ и процессором.

Временная диаграмма алгоритма взаимодействия приведена на рис. 43. Сигнал КП («Конец передачи») является заключительным сигналом взаимодействия Вч с ВнУ. В случае, если ВнУ в момент опроса было готово к работе и не занято, оно должно по сигналу КП сбросить регистр команд, так как определитель в этот регистр заносится на случай, если данное взаимодействие организовано по команде ВВИ, и в этом случае может произойти наложение последующего кода определителя на данный при последующем обращении к ВнУ уже с командой ВВИ, что приведет к обработке ложной команды (или просто невыполнению).

Команда НВУИ (код «—66») является, как и ИУВУ, командой условного перехода по состоянию указателей ВнУ, в процессе выполнения которой производится опрос заданных в

Таблица 4.3

Разряды команды	КШС ВнУ
12	Б
13	А
14	8
15	4
16	2
17	1

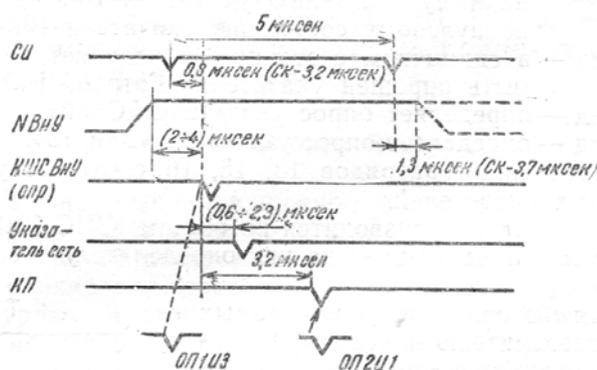


Рис. 43. Временная диаграмма выполнения команды опроса указателей устройства.

определителе команды указателей ВнУ. Временная диаграмма взаимодействия при выполнении этой команды ничем не отличается от приведенной на рис. 43, так как назначением команды является поиск ВнУ с заданным состоянием конкретного указателя, то при несоответствии состояния опрашиваемого указателя ВнУ, заданного в команде, производится автоматическая модификация номера ВнУ на «+1» и повторяются такты ОП1, ОП2 (рис. 44, точка а). Таким образом, производится последовательный опрос всех ВнУ, начиная с начального (заданного в индексной ячейке команды). В случае нахождения ВнУ с заданным указателем происходит переход по адресу А2, а в сумматоре АУ в разрядах 13 ÷ 20 оказывается номер найденного ВнУ

(рис. 44, точка б). В случае если среди существующих ВнУ нет заданного состояния указателя, то после опроса последнего номера ВнУ (237₈) происходит переход на выполнение следующей (по СчАК) команды, а на сумматоре АУ оказывается номер ВнУ на единицу больше последнего (240₈). Это является спецификой, отличающей данную команду от ИУВУ. Команда реализуется последовательностью выполнения следующих тактов:

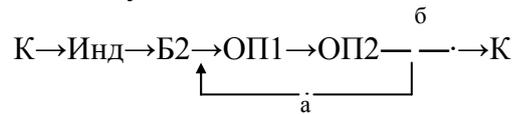


Рис. 44. Выполнение команды НВУИ.

Команда ВВИ (код «—67») является командой пуска ВнУ на отработку команды, заданной определителем. Выполняемые при этом команды могут иметь служебное назначение и не быть командами обмена информацией. В процессе пуска ВнУ по команде выполняются следующие ее вспомогательные функции:

- определяется состояние ВнУ,
- производится (или нет) запись УС в ячейку УОП, соответствующую номеру пускаемого ВнУ.

ВнУ может находиться в трех состояниях: свободно, занято, не готово. Свободным или занятым ВнУ может быть лишь при условии готовности работы его с процессором.

Процессор засылает в выбранное согласно номеру ВнУ по КШС определитель команды. Через некоторое время из Вч во ВнУ посылается сигнал «Пуск 1 ВнУ», назначением которого является:

- пуск ВнУ на отработку команды, заданной определителем, если ВнУ готово и не занято;
- опрос состояния ВнУ.

ВнУ в ответ на «Пуск 1 ВнУ» вырабатывает в Вч сигналы:

- «ВнУ свободно», если оно готово к работе и не занято выполнением предыдущей команды;
- «ВнУ занято», если оно готово к работе, но занято.

Если ВнУ не готово к работе, оно не вырабатывает ответных сигналов. Сигнал КП (конец передачи) является заключительным сигналом обращения.

Временная диаграмма выполнения команды ВВИ приведена на рис. 45.

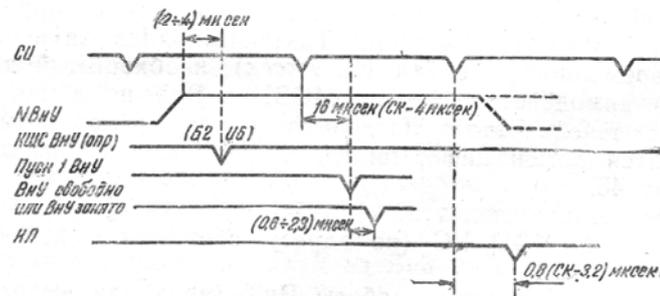
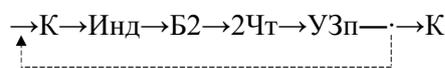


Рис. 45. Временная диаграмма выполнения команды пуска устройства.

Команда реализуется выполнением следующих тактов:



Такты Б2, 2Чт и УЗп выполняются неразрывно. В такте 2Чт производится чтение УС в регистр Р1 АУ из ячейки с адресом А2. В случае занятости ВнУ команда закликивается до его освобождения, при этом блокируется запись УС в ячейку УОП.

В случае неготовности ВнУ в Вч вырабатывается сигнал прерывания на 2-й уровень (уровень сбоя ВнУ и неготовности). Если ВнУ готово и свободно, в такте УЗп производится запись УС из регистра Р1 АУ в ячейку УОП (11-й разряд определителя команды находится в нулевом состоянии). Единичное состояние 11-го разряда указывает на необходимость блокировки записи УС в ячейку УОП, что соответствует режиму пуска ВнУ для работы с прежним УС.

Медленная приостановка

Медленная приостановка является операцией непосредственного обмена информацией между процессором и ВнУ МК. Организация этой операции производится при наличии следующих

условий: есть запрос МК, нет запроса СК, нет запрета приостановок.

Операции предшествует процесс подготовки, который заключается в следующем:

- определении номера ВнУ, выставившего запрос;
- осуществлении выборки ВнУ согласно его номеру;
- запрещении выполнения (переключения) очередного такта выполняющейся команды;
- установлении режима начала медленной приостановки (первого такта — 1П).

К началу операции приостановок ВнУ уже выбрано (т. е. потенциал «Номер ВнУ» на входе устройства высокий), а его номер занесен в РА ВнУ.

Операция состоит из последовательного выполнения следующих тактов: 1П, 2П, 3П. Все они являются тактами обращения к оперативной памяти процессора. Такты выполняются неразрывно, образуя временной интервал (15 мксек), необходимый для организации взаимодействия между МОЗУ и ВнУ по обмену информацией. В течение этого интервала между процессором и ВнУ производится обмен информацией и служебными сигналами согласно рис. 46.

В такте 2П производятся:

- чтение из УОП УС (по адресу, соответствующему номеру ВнУ);
- определение режима работы ВнУ (ввод или вывод);
- прием из ВнУ при вводе информационного символа;
- анализ обмена.

Для определения режима работы ВнУ процессор выдает сигнал «Пуск 2 ВнУ», на который ВнУ, если оно готово к работе, вырабатывает ответный сигнал «Ввод» или «Вывод», в зависимости от режима его работы. Сигнал «Пуск 2 ВнУ» является также для ВнУ определителем операции обмена информацией. По нему ВнУ должно сбросить свой запрос.

Выдача информации из ВнУ в процессор производится сигналом «Обмен». Анализ обмена заключается в определении возможности организации обмена и сводится к операции сравнения A_k или A_3 с A_n УС. В случае, если $A_k < A_n$ или $A_3 < A_n$, ВнУ останавливается сигналом «Сбой Вч»,

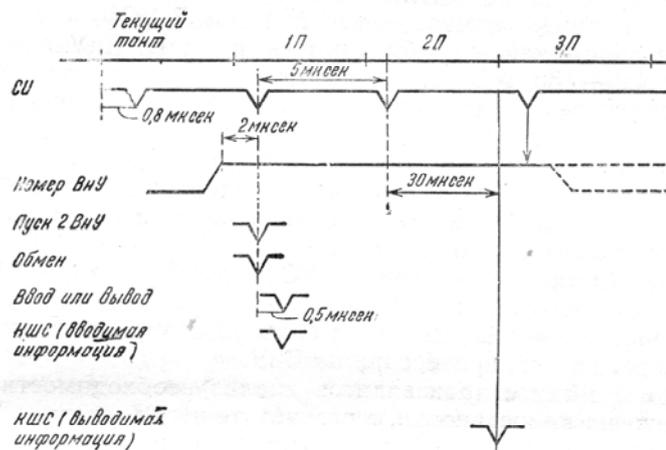


Рис. 46. Временная диаграмма медленной приостановки.

обмен символом блокируется, производится прерывание выполняемой процессором программы, переключение на 1-й программный уровень и установка в «1» указателя «ЗУС» (сбой по защите УС). Если $A_k = A_n$, производится останов ВнУ сигналом «КМ» и блокировка обмена символом.

В такте 2П производятся:

- анализ обмена;
- чтение или запись символа (в зависимости от режима обмена) по адресу, соответствующему текущим значениям N_c и A_n ;
- модификация адреса A_2 на «+1» (N_c или A_n);
- выдача символа в выводное устройство.

Анализ обмена во втором такте приостановки состоит из следующих операций:

- сравнения вводимого в МОЗУ из ВнУ символа с признаком K УС в начале такта (перед его записью в ОП);
- сравнения выводимого из МОЗУ во ВнУ символа с признаком K УС в конце такта (после его чтения из ОП);
- сравнения A_n с A_k при вводе информации в конце такта (после модификации A_n УС).

В случае если вводимый символ оказался равным K УС и 12р. УС (P) находится в «0», то

производится останов ВнУ сигналом «КМ» и блокировка ввода символа в ОП.

Если вводимый символ равен K УС и P УС находится в «1», то производится останов ВнУ сигналом «КМ» и разрешается запись символа в ОП.

Если выводимый символ равен признаку K УС и P УС — в «0», то производится останов ВнУ (переданный в него символ в этом случае не наносится на носитель).

Если выводимый символ равен K УС и P УС — в «1», то производится запоминание необходимости останова ВнУ при следующем запросе—13р. УС устанавливается в «1», символ выводится. При последующем запросе, при анализе обмена в конце такта 2П, произойдет останов ВнУ сигналом «КМ» по единичному состоянию 13р. УС.

Если $A_n = A_k$, вводное ВнУ останавливается сигналом «КМ» (символ, поступивший во ВнУ, не наносится на носитель).

В такте 3П производится:

—запись модифицированного УС в УОП по адресу, соответствующему номеру ВнУ;

—выработка заключительного сигнала взаимодействия — «КП» (конец передачи) из процессора во ВнУ.

В этом же такте производится анализ необходимости продолжения режима приостановки, в результате чего может:

—повториться процесс медленной приостановки с данным или другим запрашивающимся ВнУ;

—начаться процесс быстрой приостановки;

—начаться процесс записи УС СК в УОП после окончания обмена с ВнУ СК;

—начаться продолжение выполнения тактов приостановленной команды (если нет ни одного из типов приостановок).

Организация селекторного канала

В настоящее время ЭВМ комплектуется целым рядом «быстрых» ВнУ, скорость работы которых приближается к быстродействию процессора. Примером таких «быстрых» ВнУ могут служить накопители на магнитной основе — магнитные диски, барабаны, ленты, карты и т. д.

Проектируемые селекторные каналы для связи процессора ЭВМ с ее быстрыми внешними устройствами, как правило, являются результатом поиска оптимального сочетания стремлений минимизировать затраты на оборудование и обеспечить максимально возможное быстродействие канала.

Одной из специфических особенностей этих «быстрых» устройств является непрерывность движения носителя в процессе массового обмена. Обмен информацией между МОЗУ и ВнУ СК в ЭВМ «Минск-32» производится через буферную память СК (БП): МОЗУ→БП СК→ВнУ при выводе, ВнУ→БП СК→МОЗУ при вводе.

Взаимодействия МОЗУ с БП СК и ВнУ с БП СК являются асинхронными процессами по отношению друг к другу. Взаимодействие БП СК с МОЗУ называется быстрой приостановкой. Оно производится в один такт (обмен символом) по запросу СК, который является приоритетным по отношению к запросам ВнУ МК.

Этот запрос организуется при наличии неполной загрузки БП СК информацией процессора при выводе и неполной загрузки этой памяти от информации ВнУ при вводе. Запрос сохраняется до устранения дефицита разгрузки-загрузки БП, что соответствует повторению тактов быстрой приостановки. Быстрая приостановка производится в один такт, так как УС ВнУ СК хранится в специальном регистре УО (БРСУе), что не требует дополнительных тактов его чтения и записи в УОП. В процессе одного такта БП производится обмен одним символом между МОЗУ и БП СК. Процесс быстрой приостановки синхронен с работой процессора.

Процесс обмена информацией между БП СК и ВнУ СК является асинхронным по отношению к процессору и производится следующим образом. При возникновении во ВнУ СК необходимости получить символ из процессора или ввести символ в МОЗУ ВнУ СК вырабатывает сигнал «Запрос импульсный». При выводе символ поступает во ВнУ в ответ на «Запрос импульсный» с незначительной задержкой. При вводе ВнУ отправляет в Вч символ вместе с сигналом «Запрос импульсный».

Окончание обмена

Окончание обмена между процессором и ВнУ характеризуется прекращением выработки запросов на обмен ВнУ и переходом на отработку внутреннего цикла ВнУ. ВнУ оканчивает обмен либо по специальным сигналам из процессора, либо после отработки полученной им команды, либо при возникновении во ВнУ сбойной ситуации.

Одним из сигналов окончания обмена из процессора является «КМ» (конец массива), который вырабатывается процессором в следующих случаях:

— в информации, подлежащей обмену, встретился символ, совпавший с символом признака конца обмена;

— произошло сравнение адресов $A_n = A_k$.

Выводные устройства, получившие символ вместе с сигналом «КМ», не выводят символ на носитель. По сигналу «КМ» ВнУ должно последовательно выполнить следующие операции:

- прекратить выработку запросов;
- отработать внутренний цикл — перейти в исходное состояние;
- сбросить указатель «Занято»;
- установить необходимые активные указатели;
- послать в процессор сигнал «Конец работы».

Аналогичную последовательность операций ВнУ должно выполнить и в случае окончания отработки во ВнУ им полученной команды. Сброс указателя «Занято» и последующие операции должны выполняться за короткий промежуток времени (не более $10 \div \text{мксек}$).

Обмен оканчивается также и по причине возникновения сбойных ситуаций.

При возникновении сбоя в процессоре, при обмене информацией между ним и ВнУ процессор вырабатывает сигнал «Сбой Вч», по которому ВнУ прекращает обмен. Причинами выработки этого сигнала может быть: $A_k < A_n$ (сбой по защите УС—ЗУС); $A_3 < A_n$ (сбой по защите УС—ЗУС); сбой МОЗУ в тактах 1П (сбой по «нечету» УС—НУС) и 2П (сбой по «нечету» информационного слова — НИС); сбой по «нечету» вводимого в процессор из ВнУ символа, сбой из-за неготовности ВнУ при обмене (нет ответного сигнала «Ввод» или «Вывод» или есть оба).

В случае сбойного окончания обмена по «Сбою Вч» ВнУ должно последовательно выполнить следующие операции: прекратить выработку запросов; установить в «1» указатель «Сбой»; отработать внутренний цикл; сбросить указатель «Занято».

Установка в «1» указателя «Сбой» должна производиться не позднее чем через $10 \div 20 \text{ мксек}$ после «Сбоя Вч». Выводные устройства не реагируют на символ, полученный вместе с сигналом «Сбой Вч».

В случае сбойного окончания обмена по внутренней причине ВнУ, последнее должно последовательно выполнить следующие операции: прекратить выработку запросов; отработать внутренний цикл; сбросить указатель «Занято»; установить в «1» указатель «Сбой» и другие нужные активные указатели; отослать в процессор сигнал «Сбой ВнУ».

Сброс указателя «Занято» и последующие операции не должны занимать более $10 \div 20 \text{ мксек}$.

Установка всех активных указателей во ВнУ должна производиться по сигналам «СИ» (синхронизирующие импульсы) из процессора. Допускается асинхронная по отношению к работе процессора установка и сброс пассивных указателей.

4.9. ОСОБЕННОСТИ РАБОТЫ С АКТИВНЫМИ ВнУ

Отличительной особенностью активных ВнУ является возможность установления связи между процессором и ВнУ по инициативе самого ВнУ. К таким ВнУ относятся пишущая машинка, аппаратура передачи данных, электронная вычислительная машина при работе в системе из нескольких машин. Для работы с такими ВнУ выделен специальный уровень N_3 , который обслуживают программы системы «Диспетчер» по установлению связи и управлению работой активных устройств. В целом алгоритм командного и информационного взаимодействия активных устройств идентичен алгоритму работы пассивных устройств селекторного и мультиплексного каналов.

Активные ВнУ мультиплексного канала

Отличает эти устройства от пассивных ВнУ наличие ряда специфичных сигналов и указателей.

Основным сигналом таких ВнУ является сигнал «Запрос на прерывание», который посылается устройством в процессор, если устройство должно вводить информацию в машину, и в случае окончания ввода массива информации. По этому сигналу производится прерывание работы процессора на 3-й уровень, программа которого производит поиск внешнего устройства, выработавшего, запрос. Признаком того, что данное устройство запрашивается в ЭВМ на ввод информации, является «1» состояние указателя «Начало ввода» (этому указателю соответствует разряд 2 символа в команде опроса). В «0» состоянии этот указатель сбрасывается в следующих случаях:

- по команде опроса этого указателя;
- по команде ввода из машины;
- сигналом сбоя.

Активные ВнУ селекторного канала

Быстрый канал процессора ЭВМ «Минск-32» позволяет подключаться к аналогичной ЭВМ «Минск-32» для организации вычислительного комплекса — системы машин. Достигается это за счет того, что основным признаком связи машин между собой выбран алгоритм взаимодействия процессора и внешнего устройства, т. е. машина рассматривается по отношению к другой, как ВнУ.

Взаимодействие машин в системе сводится к обмену информацией между устройствами МОЗУ машин с помощью команд связи с ВнУ. Одна машина вводит информацию, другая — выводит.

Более подробно вопрос связи ЭВМ в систему освещен в главе V.

4.10. ТРЕБОВАНИЯ К ВНЕШНИМ УСТРОЙСТВАМ

Структура подключения ВнУ

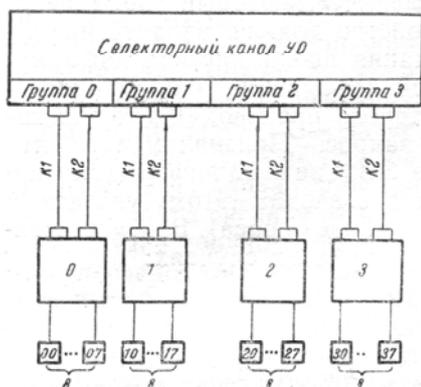


Рис. 47. Подключение устройств к селекторному каналу.

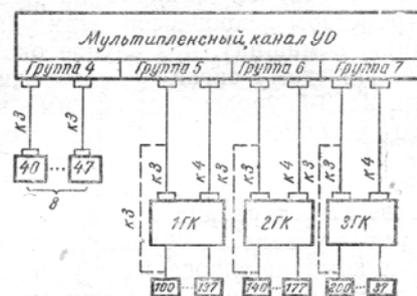


Рис. 48. Подключение устройств к мультиплексному каналу.

СК УО обеспечивает подключение четырех блоков, по 8 устройств в каждом, как показано на рис. 47. Подключение каждого из таких блоков производится двумя сигнальными кабелями К1 и К2. В случае отключения (отсоединения) блока на соответствующие ему разъемы канала ставятся специальные заглушки. По питанию местные устройства управления каждого блока подключаются к устройству питания (УПВ) основного комплекта.

МК УО обеспечивает подключение непосредственно восьми устройств (основной комплект ВнУ) и трех групповых коммутаторов (ГК) для подключения дополнительных ВнУ. Каждый ГК может обеспечить подключение до 32 устройств, как показано на рис. 48. Подключение ВнУ непосредственно к МК УО производится кабелем К3.

Подключение ГК к МК УО производится двумя кабелями К3 и К4. Вместо любого из ГК к МК УО возможно непосредственное подключение одного ВнУ кабелем К3 (рис. 48, штриховая линия). Кабели К1, К2, К3, К4 идентичны и имеют одинаковую схему распайки. При непосредственном подключении дополнительных ВнУ к МК УО питание этих устройств производится так же, как и ВнУ основного комплекта, от УПВ процессора. При подключении дополнительных ВнУ и ГК питание этих ВнУ производится от ГК, который по первичной сети должен быть подключен к источнику 380/220 в 50 гц. Разводка кабелей питания в случае

подключения к УПВ должна соответствовать цоколевке дополнительных разъемов УПВ.

Связь процессора с ВнУ как СК, так и МК выполнена по универсальному принципу, заключающемуся в выборке ВнУ (коммутация канала взаимодействия) и организации конкретного взаимодействия. В процессе взаимодействия процессора с ВнУ по кабелям К1÷К4 передаются стандартные сигналы, соответствующие конкретному виду взаимодействия и приведенные на рис. 49÷52. Всем ВнУ присваиваются номера, являющиеся адресами. Номер ВнУ зависит от номера разъема УО, к которому подключено данное ВнУ.

Назначение, функции и параметры сигналов разъемов связи приведены в приложении 2.

Требования к согласованию связей ВнУ

Как видно из таблицы приложения 2, все сигналы связи между процессором и ВнУ требуют применения импульсно-потенциального комплекса элементов. Все сигналы процессора, за исключением сигналов «СИ» и «Сброс общий», должны приниматься во ВнУ при его комплексном режиме работы, при условии его выборки, т. е. при наличии на входе устройства высокого потенциала «Номер ВнУ».

Прием сигналов «СИ» и «Сброс общий» должен запрещаться лишь при автономном режиме работы устройства. Сброс указателей «Сбой», «Начало ввода», «Ввод», «Конец ленты», «Не вся

Контакты разъема	Наименование сигнала	Тип связи
a1	Пуск 1 ВнУ	
a2	Конец передачи	»
a3	Си	»
a4		
a5		
a6	КМ	имп.
a7	Сбой Вч	»
a8	Общий сброс	»
a9		
a0	Номер группы	пот.
в1	1 КШС ВнУ	имп.
в2	2 КШС ВнУ	»
в3	4 КШС ВнУ	»
в4	8 КШС ВнУ	»
в5	АКШС ВнУ	»
в6	БКШС ВнУ	»
в7	МКШС ВнУ	»
в8	ККШС ВнУ	»
в9		
в0		
с1	ВнУ свободно	»
с2	ВнУ занято	
с3		
с4		
с5		
с6	Сбой ВнУ	имп.
с7	Конец работы ВнУ	»
с8	Указатель есть	»
с9		
с0		

Рис 49. Набор сигналов кабеля К1

Контакты разъема	Наименование сигнала	Тип связи
a1	Номер "0" у-ва	пот
a2	Номер "1" у-ва	»
a3	Номер "2" у-ва	»
a4	Номер "3" у-ва	»
a5	Номер "4" у-ва	»
a6	Номер "5" у-ва	»
a7	Номер "6" у-ва	»
a8	Номер "7" у-ва	»
a9		
a0		
в1		
в2		
в3		
в4		
в5		
в6		
в7	Запрос имп	имп
в8		
в9		
в0		
с1		
с2	Конец обмена	имп
с3		
с4		
с5		
с6		
с7		
с8		
с9		
с0		

Рис 50. Набор сигналов кабеля К2

зона считана» должен производиться после их опроса с задержкой, не превышающей 10÷15 мксек (а также при очередном пуске ВнУ),

В ЭВМ «Минск-32» для связи с внешними устройствами применены:

— для передачи во ВнУ импульсных и потенциальных сигналов—соответственно ячейки формирователей ЗФП1 и 4ИМ1;

— для приема из ВнУ импульсных и потенциальных сигналов — соответственно ячейки обратных формирователей ЗФП2 и инверторов 4И1.

Нагрузочные способности ячеек на каждую из шин связи с ВнУ (с учетом тока, потребляемого элементами согласования) составляют: для ячеек 4ИМ1 —20 ма; для ячеек ЗФП1

— 10 ма. Потребление по входам (с учетом того, что сигнал может поступать на несколько элементов) составляет: для ячеек ЗФП2 — 70 ма; для ячеек 4И1 — 9 ма.

Контакты разъема	Наименование сигнала	Тип связи
a1	Пуск 1 ВнУ	
a2	Конец передачи	»
a3	Си	»
a4	Пуск 2 ВнУ	»
a5	Обмен	»
a6	КМ	»
a7	Сбой Вч	»
a8	Общий сброс	»
a9		
a0	Номер ВнУ	пот.
в1	1 КШС ВнУ	имп.
в2	2 КШС ВнУ	»
в3	4 КШС ВнУ	»
в4	8 КШС ВнУ	»
в5	АКШС ВнУ	»
в6	БКШС ВнУ	»
в7	МКШС ВнУ	»
в8	ККШС ВнУ	»
в9	Запрос пр.	»
в0		
с1	ВнУ свободно	»
с2	ВнУ занято	»
с3	Запрос ВнУ	пот
с4	ввод	имп
с5	вывод	»
с6	Сбой ВнУ	»
с7	Конец работы ВнУ	»
с8	Указат.есть	»
с9	Мен. границы	»
с0		

Рис 51. Набор сигналов кабеля К3

Контакты разъема	Наименование сигнала	Тип связи
a1	Номер "0" у-ва	пот
a2	Номер "1" у-ва	»
a3	Номер "2" у-ва	»
a4	Номер "3" у-ва	»
a5	Номер "4" у-ва	»
a6	Номер "5" у-ва	»
a7	Номер "6" у-ва	»
a8	Номер "7" у-ва	»
a9	Номер "8" у-ва	»
a0	Номер "8" у-ва	»
в1	Номер "16" у-ва	»
в2	Номер "16" у-ва	»
в3	Запрос "8" у-ва	»
в4	Запрос "8" у-ва	»
в5	Запрос "16" у-ва	»
в6	Запрос "16" у-ва	»
в7		»
в8		»
в9	Запрос ВнУ0	»
в0		
с1	Есть ГК	»
с2		
с3	Запрос ВнУ1	»
с4	Запрос ВнУ2	»
с5	Запрос ВнУ3	»
с6	Запрос ВнУ4	»
с7	Запрос ВнУ5	»
с8	Запрос ВнУ6	»
с9	Запрос ВнУ7	»
с0	"-15в"	»

Рис 52. Набор сигналов кабеля К4

Форма сигнала на выходе ячейки ЗФП1 приведена на рис. 53, форма сигнала на выходе ячейки 4ИМ1 — на рис. 54. Формы импульсных и потенциальных сигналов, поступающих от

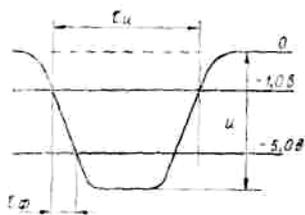


Рис. 53. Импульсный сигнал процессора.

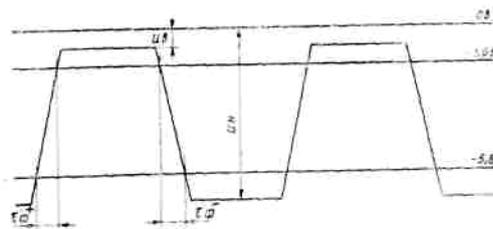


Рис. 54. Потенциальный сигнал процессора.

внешних устройств, должны быть аналогичны формам сигналов, приведенным соответственно на рис. 55 и 56.

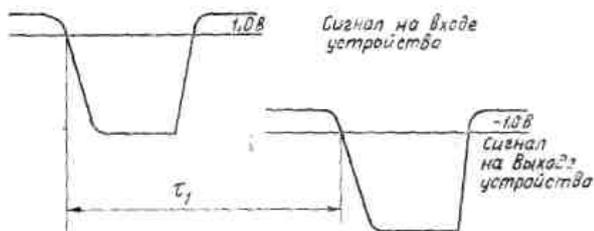


Рис. 55. Согласование импульсных сигналов в устройстве.

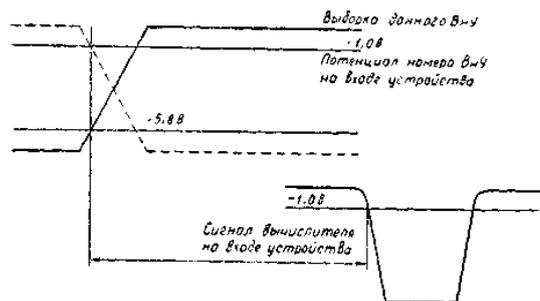


Рис. 56. Согласование импульсного и потенциального сигналов в устройстве.

В табл. 4.5 показаны ориентировочные входные и выходные параметры ячеек, применяемых для связи с ВнУ. Выходные параметры приведены для выходов ячеек процессора при связи с ВнУ через кабель типа ИКМ-6 длиной не более 10 м.

Требования к параметрам сигналов внешних устройств

Внешние устройства должны передавать символ (при вводе), а также сигналы «Свободно», «Занято», «Ввод», «Вывод», «Есть указатель», «Менять границу» с задержкой $\tau_1 \leq 0,6$ мксек относительно соответствующего сигнала процессора, поступающего на вход ВнУ (см. рис. 55).

Таблица 4.5

Параметры ячеек, применяемых для связи с ВнУ			
Ячейки	Необходимые параметры входных сигналов	Параметры выходных сигналов	Помехоустойчивость
ЗФП1		$\tau_u = 0,6 \div 1,8$ мксек $\tau_\phi < 0,1$ мксек $U = 7,0 \div 9,4$ в	
4И1	$0,025$ мксек $< \tau_\phi^+ \leq 0,4$ мксек $0,025$ мксек $< \tau_\phi^- \leq 1,0$ мксек $U_B = 0 \div 1,0$ в $U_H = - (5,8 \div 10)$ в		- 1,9 в (от уровня 0 в), + 1,5 в (от уровня — 6,3 в)
4ИМ1		$U_B = - (0,1 \div 0,5)$ в $U_H = - (8,0 \div 10,0)$ в $\tau_\phi^+ \leq 0,4$ мксек $\tau_\phi^- \leq 1,0$ мксек	
ЗФП2	$U \leq 4$ в $\tau_u = 0,7 \div 1,5$ мксек $\tau_\phi \leq 0,45$ мксек		1,4 в

Сигнал «Пуск 2 ВнУ», «Обмен» или определитель на входе внешнего устройства появляется с задержкой τ (4 мксек $> \tau_2 \geq 2$ мксек) по отношению к потенциалу номера ВнУ (см. рис.46).

Прием сигналов «Сбой Вч» и «Конец массива» во внешнем устройстве МК должен разрешаться не позднее 7 мксек после сигнала «Конец передачи» такта обмена с данным устройством.

Прием информации с КШС ВнУ должен разрешаться не позднее 5 мксек после сигнала «Конец передачи» такта обмена с данным устройством.

ВнУ должны выдавать сигнал «Запрос ВнУ» с задержкой τ_3 ($1,0$ мксек $< \tau_3 < 3,5$ мксек) относительно синхроимпульса процессора на входе устройства (рис. 57)

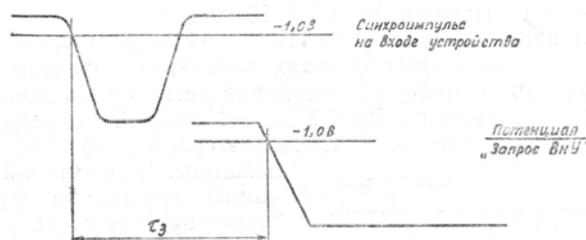


Рис. 57. Синхронизация сигналов в устройстве.

Внешние устройства должны иметь схему подрезки поступающих в них импульсных сигналов. Схема подрезки должна уменьшать амплитуду приходящих импульсных сигналов с тем, чтобы последние при отсутствии разрешения не вызывали срабатывания соответствующих элементов ВнУ. Схема подрезки не должна нагружать сигналы, приходящие во ВнУ при выключенном

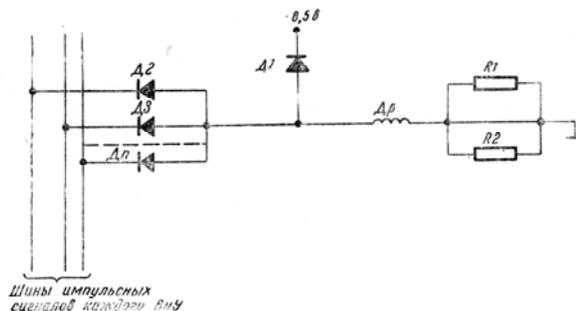


Рис. 58. Схема согласования сигналов по амплитуде.

Д1 — стабилитрон Д814Б, Д2, Д3..., Дn — диоды Д220, Др — трансформатор ТИ 101 (1Ф4.120.048Сн), R1, R2 — 100 ом (2 Вт).

питании данного устройства. На рис. 58 приведена схема подрезки сигналов, рекомендуемая для внешних устройств, использующих элементы «Минск-2».

Требования к помещению и размещению ВнУ

Для подключения ВнУ к ЭВМ «Минск-32» используется коаксиальный кабель ИКМ-6, позволяющий подключать ВнУ на расстоянии до 10 м. Подключение на большие расстояния с использованием кабеля ИКМ-6 невозможно, так как последний имеет распределенное омическое сопротивление около 4 ом/м.

Каждый заказчик может самостоятельно увеличить расстояние до подключаемых ВнУ, используя коаксиальные кабели, обладающие следующими волновыми параметрами:

- распределенная емкость — не более 45 пф/м;
- распределенное сопротивление — не более 0,3 ом/м;
- волновое сопротивление — 100 ± 10 ом.

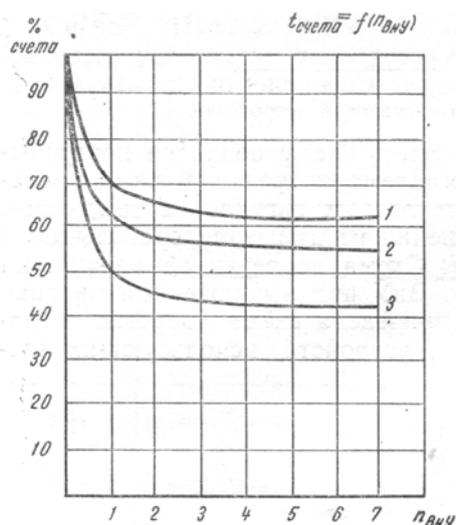


Рис. 59 Диаграмма занятости процессора при обслуживании работы внешних устройств.

Значения точек $n_{внУ}$ 1 — магнитная лента, барабан. ЭВМ; 2 — ввод карт; 3 — ввод с перфоленты; 4 — вывод на печать; 5 — вывод карт; 6 — вывод на перфоленту 1; 7 — вывод на перфоленту 2.

Так, при применении кабеля ИКМ-2 возможно удлинение связей до 30 м. В некоторых случаях заказчику придется дополнительно подбирать по месту согласующие резисторы в линиях связи,

Условия эксплуатации ВнУ ЭВМ «Минск-32» определяются в основном требованиями по обеспечению работоспособности самой машины. Все устройства машины нормально функционируют при температуре окружающего воздуха $+10 \div +30^\circ \text{C}$. Наиболее благоприятные условия эксплуатации при температуре $+18 \div +22^\circ \text{C}$ и относительной влажности $65 \div 15\%$. В связи с этим желательна установка системы кондиционирования воздуха. Вся машина питается от промышленной сети трехфазного переменного тока напряжением 380/220 в с отклонением от $+10$ до -15% и частотой 50 ± 1 гц. В помещении с машиной рекомендуется освещенность $190 \div 260$ лк (на высоте 1 м от пола, вблизи устройств).

Реализованная в ЭВМ «Минск-32» СУС ВнУ позволила разрешить следующие проблемы:

- организовать различные совместимые комплексы;
- организовать системное взаимодействие;
- повысить производительность за счет одновременной

работы нескольких внешних устройств и процессора.

На рис. 59 приведена зависимость занятости процессора на обслуживание работы внешних устройств основного комплекта машины для трех случаев, когда работающим устройством селекторного канала являются: 1 — магнитная лента, 2 — магнитный барабан, 3 — ЭВМ. По оси ординат отложена величина освобожденного времени процессора для счета (в %).

Из графиков видно, что при одновременном обмене со всеми внешними устройствами основного комплекта на чистый счет остается времени:

- в ленточном варианте — 64%;
- в барабанном варианте — 56,5%;
- при системном обмене двух ЭВМ — 44%.

ГЛАВА V
**ОСОБЕННОСТИ ОРГАНИЗАЦИИ МНОГОМАШИННЫХ
КОМПЛЕКСОВ НА БАЗЕ ЭВМ «МИНСК-32»**

На протяжении всей истории развития средств вычислительной техники конструкторы занимались проблемой повышения производительности ЭВМ. Одной из основных целей, которую преследуют работы по созданию многомашинных комплексов, как раз и является повышение производительности вычислительных средств. При существующих временных параметрах логических элементов и запоминающих устройств, которые практически подошли к теоретическому пределу, такой структурный метод повышения производительности, как создание системы нескольких ЭВМ, является наиболее эффективным и возможно пока единственным путем в решении проблемы сверхвысоких производительностей. Во многих литературных источниках убедительно показана необходимость создания средств вычислительной техники с производительностью то 10^9 опер/сек и выше [21], [22]. Очевидно, что в качестве базового модуля таких систем сверхвысокой производительности, должны быть ЭВМ высокого класса или же специализированные процессоры. Создание систем на базе ЭВМ широкого применения преследует несколько другие цели.

Задачей настоящей главы является отражение тех отличительных особенностей, которые характерны для систем на базе ЭВМ широкого применения, а также доказательство в связи с этим возможности использования в качестве каналов системных связей обычного селекторного канала СУС ВнУ, что реализовано в ЭВМ «Минск-32».

5.1. НЕКОТОРЫЕ ПРОБЛЕМЫ ПОЛЬЗОВАТЕЛЕЙ ЭВМ ШИРОКОГО ПРИМЕНЕНИЯ

Существует ряд проблем для пользователя ЭВМ широкого применения, которые наиболее полно и оптимально могут быть решены путем создания систем на базе крупносерийных ЭВМ. К таким проблемам можно отнести следующие.

1. Возможность гибкого наращивания мощности вычислительных средств ВЦ. По истечении определенного времени может оказаться, что скорость работы ЭВМ вычислительного центра не удовлетворяет потребителя. Всегда окажется экономически более выгодным приобретение дополнительно однотипной, а не совершенно новой более производительной ЭВМ.

В частности, в настоящее время существуют задачи, которые не могут быть практически поставлены на существующих машинах в связи с недостаточным максимальным объемом оперативной памяти. Использование так называемого «эффекта памяти» позволяет применять совершенно новые алгоритмы, увеличивающие в некоторых случаях производительность процессора в несколько сот и тысяч раз.

2. В связи с интенсивным внедрением вычислительных машин в экономику в некоторые отрасли науки и техники в последнее время возросла потребность в создании систем дублирования, т. е. систем, состоящих из двух однотипных машин, работающих по совершенно идентичным программам. К таким системам можно отнести банковские системы, транспортные агентства, некоторые космические и военные объекты. Ошибка всего в 1 бит в расчетах и в точности хранения данных может привести к значительным экономическим потерям. Примером может служить используемая в оперативных центрах ПВО США система SAGE, состоящая из двух ЭВМ.

3. В последнее время складывается тенденция к организации различных уровней совместимости. Объединение различного количества ЭВМ в единый комплекс позволяет создавать различные по производительности гибкие совместимые комплексы.

Каждая из вышеперечисленных проблем может быть решена своим частным способом. Но одним из общих способов, решающим сразу все проблемы, как раз и является обеспечение возможности комплексирования ЭВМ в единый вычислительный комплекс-систему.

Использование в качестве базового модуля системы универсальной ЭВМ малого класса накладывает некоторую специфику на постановку задачи, на сам принцип объединения машин в систему. Следует учитывать некоторые особенности ЭВМ малого класса, которые влияют на выбор алгоритма связи и максимально возможное количество ЭВМ, объединенных в систему. К таким особенностям следует отнести уже упоминавшиеся — универсальность, серийность, габаритность и доступность.

Из всего вышеизложенного следуют два основных вывода:

машины малого класса должны иметь возможность комплексирования их в вычислительные системы с максимальным использованием универсальной логики и математического обеспечения. Доля специальных системных затрат в таких ЭВМ должна быть сведена до минимума;

критерий выбора количества одновременно подключенных машин малого класса в единую систему должен учитывать возможности современных ВЦ и задачи, которые стоят перед такими системами. При этом должна быть обеспечена гибкость постепенного наращивания мощности ВЦ, изменения количества объединяемых ЭВМ.

5.2. АНАЛИЗ СИСТЕМЫ «МИНСК-222»

Известны различные системы вычислительных машин [21], [22], о частности, система «Минск-222» на базе серийно выпускаемых машин малого класса «Минск-2», «Минск-22». Опытный образец этой системы состоял из двух ЭВМ «Минск-22» и одной ЭВМ «Минск-2». В этой работе не ставится целью подробно анализировать все особенности этой системы. Система «Минск-222» подробно описана во многих источниках [23], [24], [25]. Пуск в работу этой системы в апреле 1966 г. и последующая опытная эксплуатация подтвердили правильность многих основных положений создания однородных вычислительных систем [21] и доказали эффективность систем на базе серийных ЭВМ малого класса. Следует отметить ряд существенных особенностей работы системы «Минск-222», которые были учтены в последующих работах и, в частности, при проектировании системных связей в ЭВМ «Минск-32» [26], [35].

Организация связи ЭВМ «Минск-22» в систему осуществлялась с помощью специальных пяти системных команд, дополнительно введенных в систему команд машины при доработке. Так как эти дополнительные команды не входили в систему математического обеспечения этих машин, то это затрудняло программирование задач системного взаимодействия. Следовательно, стартовое матобеспечение и логика машины каждой серийной ЭВМ должны учитывать особенности работы в системе.

Машины «Минск-2(22)» не проектировались для работы в сложных системах. Поэтому реакция машины на сбойные ситуации, пуск, останов решена очень упрощенно и рассчитана на ручное вмешательство оператора с инженерного пульта управления. В сложных информационных системах должна быть предусмотрена возможность программной реакции на сбойные ситуации, возможность анализа, поиска причин неисправности и, что самое важное, программного исключения последствий сбоя и ошибки для возможности продолжения функционирования системы.

ЭВМ «Минск-2(22)», будучи не приспособленными для работы в системе, имеют пульт оператора, неудобный для работы в системе. Оператор, находясь за пультом одной из машин, не знает, что происходит на других машинах системы. Следовательно, пульт каждой машины системы должен иметь информацию о состоянии всех остальных ЭВМ системы или должен быть специальный общий пульт управления всей системой. Допускается вывод информации о состоянии всей системы на пишущие машинки каждой ЭВМ системы.

Система сопряжения «Минск-222» построена таким образом, что информация не только принимается в каждую машину системы, но и ретранслируется через них. При выходе из строя одной из машин разрывается непрерывная цепочка машин на месте вышедшей из строя машины. Следовательно, соединение машин в систему должно быть независимым друг от друга, необходима избыточность связи.

Система «Минск-222» рассчитана на подключение большого количества машин (до 16).

Создание системы *n* однородных серийных машин широкого применения имеет смысл в каком-то уникальном случае и имеет скорее ценность научного эксперимента. Маловероятно предположить, чтобы кто-то создал систему из 16 ЭВМ «Минск-22», установленных в одном машинном зале. Это была бы очень громоздкая, трудноуправляемая система. Схемное же решение устройства сопряжения в таком плане накладывает отпечаток громоздкости. Скорее всего наибольшее распространение у широкого потребителя получают системы двух-четырёх ЭВМ. Инженерное же решение создания таких систем требует гораздо меньше оборудования, а логика объединения упрощается.

5.3. ОСОБЕННОСТИ ЭВМ ШИРОКОГО ПРИМЕНЕНИЯ, ВЛИЯЮЩИЕ НА СТРУКТУРУ СИСТЕМНОЙ СВЯЗИ ЭВМ

Как уже упоминалось выше, современные ЭВМ широкого применения обладают рядом

существенных и в некотором роде обязательных особенностей, которые накладывают свой отпечаток на процедуру связи ЭВМ в систему. Использование в качестве модуля вычислительной системы не специализированных машин, а обычных крупносерийных ЭВМ требует учитывать следующие их особенности.

1. Многопрограммность. Эта особенность ЭВМ позволяет организовывать такую процедуру связи ЭВМ в систему, когда программы системной связи могут располагаться на одном из рабочих уровней. Используя многопрограммность, можно пойти на упрощение межмашинных связей, так как потери машинной производительности при синхронизации связываемых ЭВМ (при ожиданиях) компенсируются выполнением любой другой программы из пакета обслуживаемых задач. В частности, такими задачами могут быть другие независимые куски единой общей системной задачи.

2. Наличие управляющей программной системы «Диспетчер». Кроме общепринятых функций, система «Диспетчер» может участвовать и в обслуживании межмашинных связей, анализируя конфликтные ситуации, устраняя последствия сбойных ситуаций, устанавливая очередность ЭВМ при обмене информацией.

3. Наличие СУС ВнУ. Как упоминалось в главе III, процессор машины «Минск-32» можно рассматривать как комбинацию двух микропроцессоров, работающих на одну оперативную память.

Основной особенностью канального микропроцессора является наличие СУС ВнУ, содержащей 2 канала связи: один селекторный и один мультиплексный. Более подробно СУС ВнУ описана в главе IV настоящей работы и в [20].

Основная особенность работы этих каналов заключается в том, что процесс связи с ВнУ осуществляется в режиме прерывания (пуск ВнУ, окончание работы ВнУ), а непосредственный информационный обмен ведется в режиме приостановок с использованием управляющих слов. Прерывание осуществляется дважды на весь массив обмена для каждого ВнУ: при пуске, когда задается его режим работы, и в конце массива обмена, после конца работы ВнУ. Приостановка выполняется на каждую единицу обмена каждого ВнУ (на каждый символ).

Используя вышеописанную СУС ВнУ, можно организовать обмен системной информацией между машинами вместе с обслуживанием остальных внешних устройств и программ каждой машины. В основу системного взаимодействия положены вышеупомянутые процедуры прерывания и приостановок. Однако следует отметить, что ЭВМ — это необычное внешнее устройство. Машину следует относить к группе активных ВнУ (особенности активных ВнУ описаны в главе IV).

Следует иметь в виду, что связь ЭВМ друг с другом отличается в нескольких важных аспектах от связи таких активных внешних устройств, как телетайпы, пишущие машинки и другие дистанционные оконечные устройства, у которых присутствует человек-оператор. Эти отличия объясняются тем, что в процессе связи и общения не участвует взаимодействующий человек-оператор. Когда телетайп является приемником некоторых сообщений, то обслуживающий его оператор действует как чрезвычайно сложный и тонкий интерпретатор, а также корректор ошибок. Если действия над получаемым сообщением должны осуществлять ЭВМ, то указанные функции возлагаются на ЭВМ. Вычислительная машина должна сама осуществлять тщательный контроль за наличием ошибок.

Другое отличие между межмашинной связью и оконечными устройствами, рядом с которыми присутствует человек-оператор, состоит в том, что человек редко может извлечь преимущество из наличия весьма скоростных информационных каналов. В противоположность этому ЭВМ весьма хорошо зарекомендовала себя при быстром принятии решений, так что она обычно может эффективно взаимодействовать по быстродействующим каналам обмена. В то же время машинная связь отличается также от связи между вычислительной машиной и такими устройствами, как магнитная лента, магнитные диски и различные выводные устройства, так как ЭВМ располагает возможностями по обработке данных, которыми устройства хранения данных не обладают. Это позволяет организовывать связь не просто по принципу «сообщение за сообщением», а более сложно и трудоемко.

Основные требования

При проработке алгоритма блока системной связи «Минск-32» соблюдались следующие требования:

1. Все ЭВМ системы должны быть равноправны, т. е. каждая ЭВМ системы должна иметь возможность писать или читать информацию любой ЭВМ системы. Причем при обмене должна иметься возможность передачи как рабочей, так и служебной информации.

2. Должна быть соблюдена непосредственная связь каждой ЭВМ друг с другом по возможно наикратчайшему пути.

3. Число специальных системных команд должно быть сведено к минимуму при условии максимального использования универсальной логики и СУС ВнУ. При этом система «Диспетчер» каждой ЭВМ должна учитывать все дополнительные особенности системной связи.

4. В качестве основного режима взаимодействия машин системы должен быть применен алгоритм взаимодействия процессора с внешним устройством: одна машина рассматривается по отношению к другой как ВнУ.

Так как ЭВМ «Минск 32» — это программно-аппаратурный комплекс, т. е. управление автоматическим ходом решения программ ведется с помощью системы программ «Диспетчер» и логических цепей процессора, то и организация систем нескольких ЭВМ «Минск-32» основывается на программно-аппаратурном алгоритме. Опыт работ по созданию системы «Минск-222» показал, что жесткий алгоритм связи очень затрудняет отработку связи машин в системе [27], [35]. Вопрос объединения ЭВМ в систему — относительно новый, в процессе его отработки возможно появление различных алгоритмов, программ. В этом смысле применение мягкой логики с использованием микропрограммной структуры или программно-аппаратурного алгоритма является более выгодным и удобным. Использование этого алгоритма позволяет реализовать вышеперечисленные требования, а также ряд основных положений однородных систем [21]: однородность структуры; возможность программного изменения конфигурации систем и подсистем; принцип параллельности работы. Подключение машин в систему производится с помощью одной из групп селекторного канала. Взаимодействие машин в системе сводится к обмену информацией между оперативными накопителями машин с помощью команд связи с ВнУ. В основу обмена информацией между машинами системы положена работа машин с командами ввода-вывода. Одна машина обрабатывает команду ввода, другая — вывода. При этом непосредственное информационное взаимодействие производится также в режиме приостановок. Так как все машины подключены к одной и той же группе СК, то для их различия им присваиваются различные номера.

СУС ВнУ машины позволяет каждому внешнему устройству, в том числе ЭВМ, иметь свои специфичные указатели. Для взаимодействия с другими ЭВМ системы машина имеет следующие системные указатели: «Готова М», «Начало обмена М», «Сбой М». Указатель «Готова М» определяет готовность машины к работе в системе. Может управляться программно специальной командой, т. е. устанавливаться в «0» или «1», включая или исключая данную ЭВМ из системы. Кроме того, он реагирует на работоспособность самой аппаратуры и сигнализирует о том, пущена или остановлена машина. Этот указатель может быть опрошен всеми машинами, подключенными к системе.

Указатель «Начало обмена М» определяет, что данная ЭВМ получила право на обмен с другой ЭВМ системы и находится в состоянии обмена, который соответствует выводу информации из нее. Указатель опрашивается другими ЭВМ системы. Указатель «Сбой М» определяет, что при участии данной ЭВМ в системном обмене информацией произошел сбой. Указатель является внутренним, т. е. опрашивается только собственной машиной. Работа машин в системе управляется с помощью стандартных команд СУС ВнУ: ввода-вывода; опроса состояния указателей.

Особенности системной связи

Системное взаимодействие машин «Минск-32» характеризуется следующими особенностями:

— обмен ведется только между парой ЭВМ, при этом может быть несколько одновременных независимых пар;

- машина-инициатор обмена всегда выводная;
- скорость обмена постоянная и равна 100 килобайт/сек.

Обмен информацией между ЭВМ системы состоит из трех этапов: начала обмена; непосредственного обмена; окончания обмена. Все этапы реализуются с помощью стандартных программно-аппаратурных приемов, принятых для взаимодействия процессора с ВнУ: первый и третий этапы обмена сопровождаются обязательным прерыванием; второй этап осуществляется в режиме стандартных приостановок с использованием управляющих слов обеих ЭВМ. Начало обмена осуществляется программным блоком пуска ВнУ. Если ЭВМ, к которой хочет обратиться данная машина (инициатор), готова к работе в системе и если не занят селекторный канал машины-инициатора, выполняется команда ВВИ (пуск ЭВМ, к которой обращается машина-инициатор).

Имеют место два случая, определяющие результаты выполнения команды ВВИ: обмен между данными ЭВМ не разрешен; обмен разрешен. Запрещение обмена между данными ЭВМ возможно, когда любая из этих ЭВМ уже занята в обмене с любой другой ЭВМ системы. В случае разрешения обмена особенностью выполнения команды является выработка сигнала «Запрос М», который является отличительной особенностью активных ВнУ. Он организует (аналогично пишущей машинке) прерывание вводной ЭВМ на 3-й программный уровень для связи с программой системного взаимодействия. Одновременно производится установка в «1» указателя УНОМ (указателя начала обмена машины-инициатора). Указатель является пассивным и опрашивается внешними ЭВМ системы. Он предназначен для перевода в состояние занятости данной пары ЭВМ, для поиска вводной машиной номера машины-инициатора.

Результатом прерывания вводной машины на 3-й программный уровень является начало второго этапа взаимодействия — непосредственного обмена. Этап заключается в выполнении программы отыскания номера машины-инициатора (по указателю УНОМ) и команды ВВИ для найденного номера машины-инициатора. Специфической особенностью выполнения данной команды является включение специального генератора запросов иницируемой машины, согласно которым осуществляется непосредственный ввод информации в вводную машину из инициатора в режиме обычной быстрой приостановки. В качестве обменной информации может передаваться как рабочая информация, так и инструкции из машины-инициатора в иницируемую, которые после окончания обмена расшифровываются и выполняются последней.

Окончание обмена (начало третьего этапа взаимодействия) определяется управляющими словами обменивающихся ЭВМ. В этом случае вырабатывается сигнал «Конец обмена С», который производит прерывание обеих ЭВМ на 4-й программный уровень (аналогично «Концу работы» при обмене с ВнУ) и сброс указателя «Занят БК». В машине, вводящей информацию, этим сигналом производится выключение генератора запросов. В машине-инициаторе этот сигнал сбрасывает в «0» указатель УНОМ.

При возникновении сбойной ситуации машина, в которой она возникла, вырабатывает сигнал «Сбой М», по которому обмен оканчивается, производится прерывание обеих ЭВМ на 1-й программный уровень, установка в «1» указателей «Сбой М» этих ЭВМ и сброс в «0» их указателей «Занят БК».

5.5. ФОРМАТ СООБЩЕНИЙ

Для возможности обработки рабочей информации и инструкций вся передаваемая информация может быть классифицирована на следующие группы:

«сообщение» — последовательность символов, организованных для целей переноса информации от источника к приемнику. Оно включает весь «текст» и необходимый «заголовок»;

«заголовок» — последовательность знаков, которая образует собой вспомогательную информацию, необходимую для расшифровки и определения передаваемого «текста» информации. Такая вспомогательная информация может, например, включать в себя сообщение о ее принадлежности, приоритетности, нумерации сообщения, ответ это или запрос.

«текст» — некоторый массив данных, который должен быть передан как единое целое от отправителя к получателю.

В типичном сообщении каждый знак представляет собой 7-битный символ, сопровождаемый одним битом контроля на нечет. Могут быть бестекстовые сообщения, содержащие только заглавную информацию, как, например, подтверждение, ответ на запрос, сам запрос и т. д.

Заглавная информация используется для того, чтобы помочь программе системной связи передать, получить, упорядочить и направить сообщение по нужному направлению. В качестве единицы обмена используется 7-битный символ. С помощью символа может переноситься как алфавитно-цифровая, так и двоичная информация. При перенесении алфавитно-цифровой информации часть символов из всех 128 можно было бы использовать как служебные, разграничивающие и модифицирующие заглавную информацию. Однако необходимость несения двоичной информации 7-битным символом требует использовать все 128 символов в качестве информационных. Поэтому в сообщениях заголовок необходимо отделять пространственно. Сделать это можно, отведя под заголовок постоянно первые n символов, либо первый символ сообщения считать всегда служебным, указывающим длину заглавного сообщения.

Передаваемые массивы очень удобно контролировать с помощью системы обнаружения сдвига информации, описанной в главе 2. Удобство объясняется идентичностью разрядных сеток и неполных символов ЭВМ системы, состоящей из однородных машин.

5.6. ПОДКЛЮЧЕНИЕ ЭВМ

Выбор структуры связи

При выборе схемы подключения должны быть известны максимальное количество ЭВМ, объединяемых в систему; конфигурация связи ЭВМ в систему. Как уже упоминалось выше, системы на базе машин широкого применения объединяют несколько ЭВМ. Дальнейшее увеличение числа ЭВМ связано с преодолением многих очень противоречивых трудностей. Это мнение подтверждается опытом использования системы «Минск-222». В ЭВМ «Минск-32» очень удобно (с точки зрения принятой структуры разбиения групп каналов) ограничить максимальное количество ЭВМ в системе числом, кратным 8: 8, 16, 24, 32. Это разбиение соответствует группам селекторного канала. Наиболее оптимальным оказывается число до 8.

Следует иметь в виду, что исходя из анализа тех проблем, которые должны решаться системами на базе крупносерийных универсальных ЭВМ, наибольшее распространение у широкого пользователя получают системы из двух ЭВМ. Необходимо учитывать так же то обстоятельство, что очень многие пользователи не будут создавать системы на базе ЭВМ. В связи с этим структура объединения должна быть выбрана такой, чтобы у пользователей одиночных ЭВМ и систем из двух ЭВМ была минимальная избыточность оборудования за счет блока системного сопряжения. В то же самое время должна оставаться возможность объединения m машин, где $2 \leq m \leq 8$.

В системе «Минск-222» была выбрана связь, когда объединялись только 2 соседние ЭВМ справа и

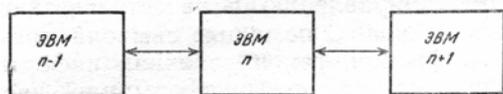


Рис. 60. Схема связей ЭВМ в системе «Минск-222».

слева [23]. (рис. 60). Информация из n -й ЭВМ может непосредственно передаваться только в $(n-1)$ -ю и в $(n+1)$ -ю ЭВМ. В случае, если необходимо связаться $(n-1)$ -й и $(n+1)$ -й ЭВМ, в этой связи должна участвовать и n -я ЭВМ. Такая система связи имеет ряд существенных недостатков.

1. В случае выхода из строя n -й ЭВМ исчезает возможность связи $(n-1)$ -й и $(n+1)$ -й ЭВМ.
2. Отсутствует возможность создания подсистем внутри системы из не рядом стоящих ЭВМ.
3. Все промежуточные ЭВМ должны участвовать в ретрансляции информации, что исключает возможность организации в данные моменты систем с участием ретранслируемых ЭВМ.

Как известно, для повышения надежности, живучести и скорости обмена каждая ЭВМ системы должна иметь число направлений связи более 2. Идеальной структурой является связь, где каждая ЭВМ системы имеет непосредственное соединение с каждой машиной системы. Рассчитаем количество связей и их избыточность в такой системе. Пусть имеется n точек, которые нужно связать друг с другом таким образом, чтобы каждая имела непосредственную связь друг с другом. Общее число связей при этом определяется как $\frac{n(n-1)}{2}$. Для передачи же информации всем n точкам необходима только $(n-1)$ связь. Количество избыточных связей:

$$\frac{n(n-1)}{2} - (n-1) \left(\frac{n}{2} - 1 \right) = \frac{(n-1)(n-2)}{2}. \quad (5.1)$$

На рис. 61 приведена зависимость общего количества связей P и избыточных Q от числа объединяемых точек n . Избыточность резко возрастает с увеличением числа n . Так в системе четырех информационных точек для нарушения связей нужно разорвать более трех связей, а уже в системе восьми точек связь будет нарушена при разрушении более 21-й связи.

Связь ЭВМ «Минск-32» в систему

В структуре связи ЭВМ «Минск-32» с машинами системы учитываются два вышеприведенных требования. Проблема решается следующим образом. Одна из групп селекторного канала имеет возможность обмениваться информацией в одном направлении, т. е. один выход. Это позволяет очень легко создавать систему двух ЭВМ «Минск-32», объединив непосредственно соответствующие группы селекторного канала, как это показано на рис. 62.

0, 1, 2-я группы селекторного канала каждой ЭВМ используются для подключения собственных внешних устройств. Третьи группы каждой ЭВМ объединяются между собой для обмена системной информацией. Объединение производится без доработок селекторного канала каждой ЭВМ и без дополнительных устройств с помощью специального кабеля. Системное оборудование каждой ЭВМ состоит из 15 стандартных ячеек комплекса 600 кгц, что составляет около 0,4% электронного оборудования каждой ЭВМ. В системе «Минск-222» системный блок содержал свыше 80 стандартных ячеек 250 кгц комплекса, что составляло около 4% от электронного оборудования и около 1,2% стоимости машин «Минск-22». Для подключения m машин в диапазоне $2 < m \leq 8$ используется специальный групповой коммутатор систем, который не входит в основной комплект машины. Внутренняя структура коммутатора выполнена по принципу, соответствующему алгоритму (5.1). Схема объединения восьми машин в систему в этом случае приведена на рис. 63. В зависимости от места подключения к коммутатору машинам присваиваются номера. Как видно из этой схемы, каждая ЭВМ имеет непосредственную связь с любой из семи машин системы. В каждый момент времени обмен могут вести любые 4 независимые пары. Например, 1—5, 2—6, 3—7, 4—8 или 1—8, 2—7, 3—6, 4—5. Закон объединения ЭВМ в пары обмена произвольный.

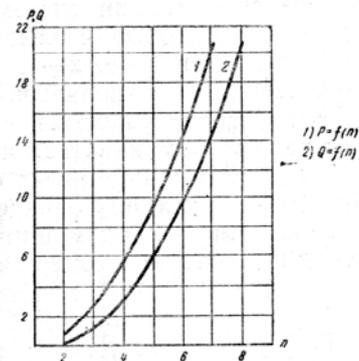


Рис. 61. Зависимость общего количества связей P и избыточных Q от числа объединяемых точек n .

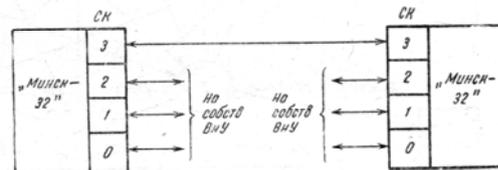


Рис. 62. Система двух ЭВМ «Минск-32».

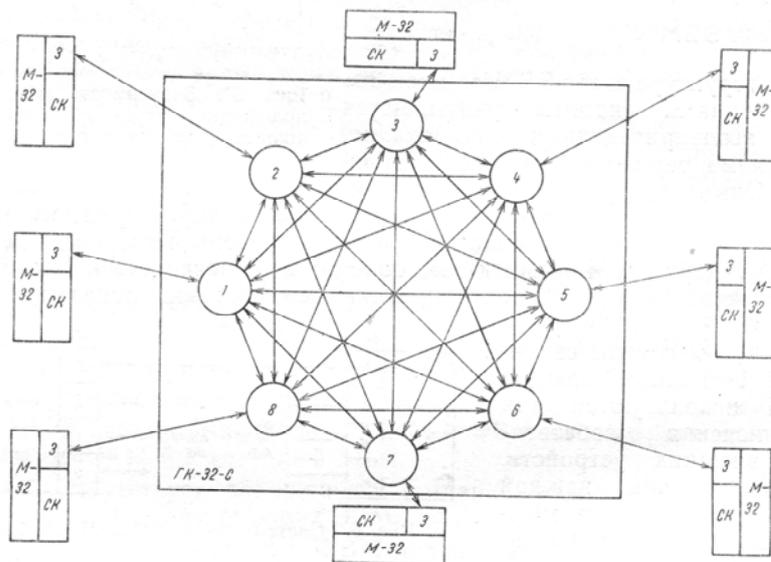


Рис. 63. Система восьми ЭВМ «Минск-32».

В случае конфликтной ситуации, если в один и тот же момент необходимо обмениваться информацией несколькими ЭВМ с одной и той же ЭВМ, приоритет отдается машине с меньшим

номером. Приоритет жесткий, определяемый аппаратурой коммутатора, и зависит для каждой машины от места ее подключения. Оборудование коммутатора содержит около 300 ячеек комплекса 600 кГц, что составляет около 1,7% электронного оборудования всей системы из восьми ЭВМ и около 0,4% от стоимости всей системы.

В системе «Минск-222» для общего системного блока восьми ЭВМ эти цифры соответственно составляли 4% и 1,2%. На рис. 64 приведена зависимость $S = f(n)$ отношения стоимости системного оборудования к общей стоимости электронного оборудования от количества ЭВМ системы. На рис. 65 приведена зависимость $Q = f(n)$ отношения системного оборудования к общему электронному оборудованию также от количества ЭВМ системы.

Из этих характеристик видно, что удельные стоимостные и электронные затраты системы на базе «Минск-32» лежат ниже уровней соответствующих затрат в системе «Минск-222» независимо от числа машин в системе. В системах с использованием «Минск-32» наименее рентабельными оказываются системы из трех ЭВМ.

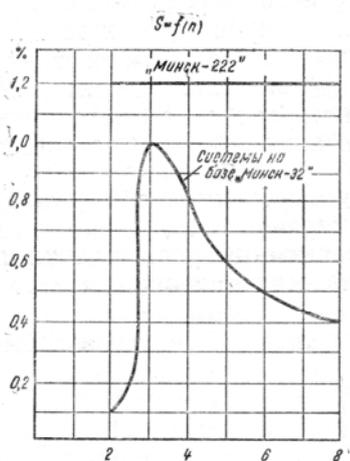


Рис. 64. Зависимость отношения стоимости системного оборудования к общей стоимости электронного оборудования от количества ЭВМ системы

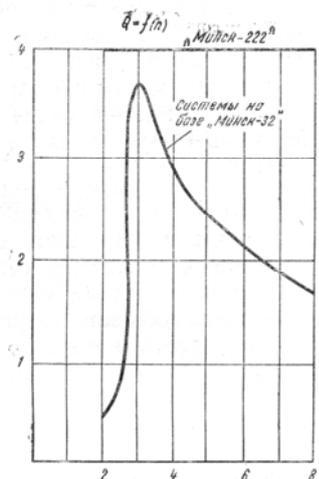


Рис. 65. Зависимость отношения системного оборудования к общему оборудованию от количества ЭВМ системы.

5.7. РАЗЛИЧНЫЕ РЕЖИМЫ ОБМЕНА ИНФОРМАЦИЕЙ

Попарный обмен информацией

В системе на базе «Минск-32» режим обмена между несколькими парами имеет следующие качественные особенности:

независимый обмен можно вести одновременно между несколькими парами;

пары машин не зависят от их места расположения в системе, от их номеров;

во время непосредственного обмена информацией каждая машина может выполнять любую работу по выполнению программ или взаимодействию с другими ВнУ.

Последнее положение является одной из особенностей системного взаимодействия. Объясняется это алгоритмом СУС ВнУ «Минск-32», который позволяет вести обмен информацией в режиме приостановок с совмещением работы процессора и ВнУ. Это положение следует иметь в виду при оценке производительности во всех случаях системного взаимодействия.

На рис. 66 приведена диаграмма взаимодействия пары машин в системе «Минск-222». Связываются 2 машины: (n) и $(n + 1)$. Векторами обозначены текущие программы каждой машины. Особенностью связи в системе «Минск-222» является синхронизация машин (выравнивание перед обменом) и последующее синхронное выполнение команд передачи и приема. При прохождении машины (n) до точки 1, где находится команда системной связи, эта машина должна в течение времени $t_{ож}$ ожидать момента, когда ЭВМ $(n+1)$ дойдет в точку 3 до команды системного взаимодействия. Общее время системного взаимодействия пары машин определяется:

$$T_c = t_{ож} + t_{об}, \quad (5.2)$$

где $t_{ож}$ — время синхронизации, зависит от расхождения программ в машинах и от разности обмена.

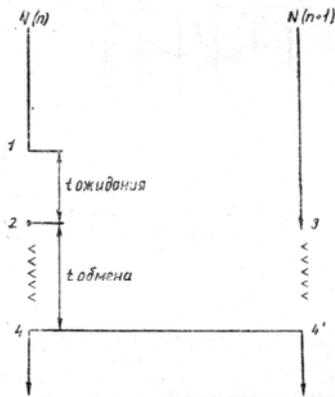


Рис. 66. Диаграмма взаимодействия пары машин системы «Минск-222».

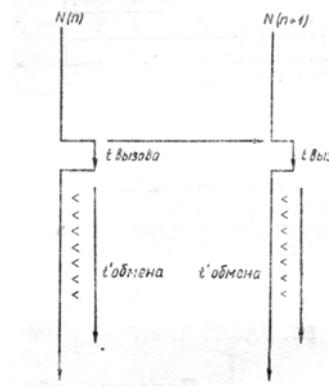


Рис. 67. Диаграмма взаимодействия пары машин в системе на базе «Минск-32».

Период $t_{ож}$ для машины (n) является чистым временем потери, так как она находится весь этот отрезок времени в состоянии статистического останова. Весь период $t_{об}$ машины ведут только обмен с частотой обмена.

Для системы на базе «Минск-32» тот же самый парный обмен приведен на рис. 67. Общее время взаимодействия выражается формулой

$$T'_c = t_{выз} + t'_{обм} \quad (5-3)$$

где $t_{выз}$ — время вызова машин на связь, является постоянной величиной, связанной с объемом программ прерывания по системному взаимодействию, для каждой машины это время асинхронно по отношению к выполняемой программе;

$t'_{обм}$ — время обмена, зависит также от величины обмениваемого массива.

Величина $t_{выз}$ в среднем всегда окажется меньше величины $t_{ож}$, так как тяжело полностью синхронизировать машины, а тем более программы. Параметр $t'_{обм}$ — величина, приблизительно равная величине $t_{об}$, если считать, что обмениваемые массивы равны. Однако следует иметь в виду, что во время $t'_{обм}$ каждая из машин совмещает обмен с выполнением любой программы, в том числе и системных подпрограмм.

Таким образом, в любом среднем случае, для задач общего применения, когда программы специально не подбираются для системы $T'_c < T_c$. Кроме того, следует иметь в виду, что производительность машин за период T'_c всегда выше, чем за период T_c , так как во втором случае машины заняты только системным взаимодействием. Особенностью систем, использующих в качестве модулей многопрограммные ЭВМ, является отсутствие простоев, так как система «Диспетчер» может заполнять интервалы простоев любыми программами из пакета заказов и совмещает саму процедуру обмена со счетом или с работой других ВнУ.

Одновременный обмен с несколькими машинами

Одной из основных особенностей системы «Минск-222» является возможность ведения одновременного обмена с несколькими машинами, когда одна машина передает, а несколько принимают одну и ту же информацию.

На рис. 68 приведена диаграмма взаимодействия, когда ($n + 1$)-я машина является передающей, а остальные — приемными. Как и в случае парного обмена, общее время системного взаимодействия может выражаться формулой (5.2). Однако $t_{ож}$ определяется в этом случае наибольшим из времен ожидания всех участвующих в обмене машин (в нашем случае — $t_{ож3}$). Каждая машина теряет в этом случае свое время $t_{ож1}$, $t_{ож2}$, $t_{ож3}$. Общее время потери в системе будет

$$t_{потери} = t_{ож1} + t_{ож2} + t_{ож3} + \dots + t_{ожn}$$

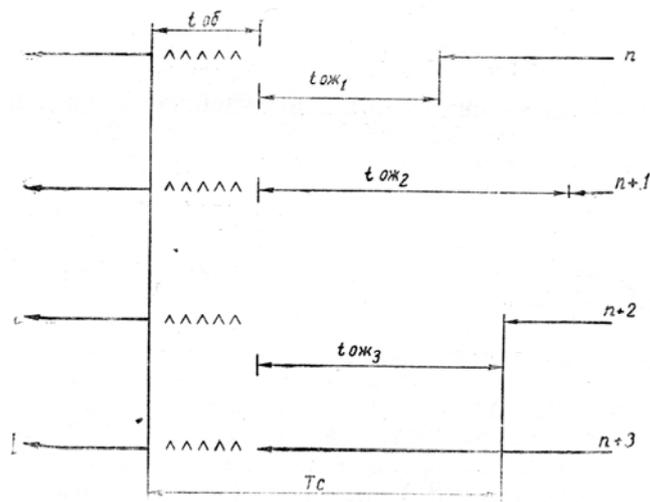


Рис. 68. Диаграмма взаимодействия n машин системы «Минск-222».

Диаграмма взаимодействия четырех машин «Минск-32» приведена на рис. 69.

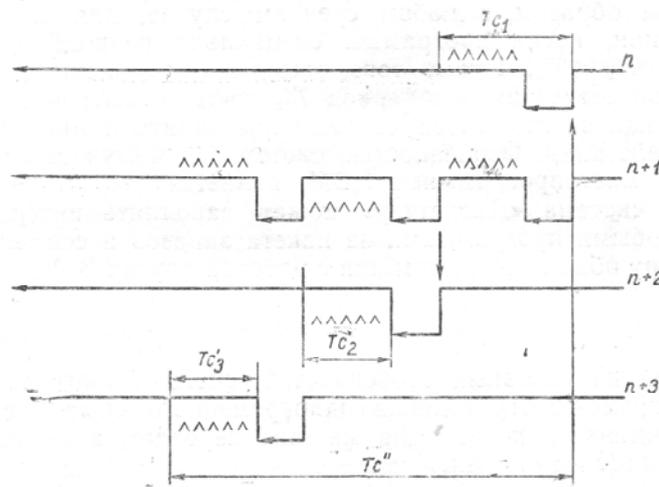


Рис. 69. Диаграмма взаимодействия n машин системы на базе «Минск-32».

Общее время системного взаимодействия выражается формулой

$$T''_c = T'_{c1} + T'_{c2} + T'_{c3} + \dots + T'_{cп},$$

где $T'_{c1} \dots T'_{cп}$ — время системного взаимодействия для n пар, определяемое выражением (5.3). T''_c будет не более T_c , если

$$t_{\text{ож.max}} \geq \sum_{i=1}^{n-1} T'_{ci}$$

При этом следует иметь в виду, что для всей системы в этом случае $t_{\text{потери}}=0$

5.8. НЕКОТОРЫЕ ПОТЕНЦИАЛЬНЫЕ СИСТЕМНЫЕ ВОЗМОЖНОСТИ

Использование СУС ВнУ как основы системного взаимодействия позволяет расширять возможные варианты системных связей машин. В частности, в последнее время появляется необходимость объединять несколько ЭВМ в единую вычислительную сеть с использованием аппаратуры передачи данных (создание сети отраслевых ВЦ, сети вычислительных центров гидрометеослужбы и т. д.). Использование аппаратуры АПД с СУС ВнУ ЭВМ «Минск-32» позволит объединять несколько таких машин, удаленных на большие расстояния. При этом применяется тот же принцип работы с активными внешними устройствами с предварительным установлением связи с помощью прерывания и ведением последующего информационного обмена в режиме приостановок. Аппаратура АПД может подключаться к мультиплексному каналу машины «Минск-32». Объединяться в этом случае может более восьми машин. Функции сводятся

к простому обмену информацией, при этом скорость обмена лимитируется пропускной способностью телеграфных и телефонных линий. Следует иметь в виду, что в этом случае должны появиться специфические указатели такого системного взаимодействия, отличные от вышеописанных и определяемые особенностями аппаратуры АПД.

Использование СУС ВнУ позволяет пользователю подключать любое устройство, отвечающее требованиям СУС ВнУ. Эта возможность позволяет гибко изменять конфигурацию не только вышеописанными способами, но и путем подключения различных ЭВМ, процессоров, блоков памяти. Если есть необходимость создания системы неоднородных ЭВМ, то в этом случае вторая специализированная машина должна быть доработана под алгоритм СУС ВнУ основной машины. Например, можно подключить к «Минск-32» ЭВМ «Минск-22» по вышеописанному способу, если последняя будет доработана под алгоритм системной связи «Минск-32».

Наконец, использование СУС ВнУ как системного канала легко позволяет создавать системы с общим полем памяти, скажем, набор накопителей НМЛ, работающих сразу на две ЭВМ. Аналогично можно создавать общие поля оперативных памяти, накопителей на магнитных барабанах, дисках.

При существующих проблемах пользователей ЭВМ широкого применения необходимо, чтобы такие машины имели возможность объединения в систему нескольких ЭВМ.

Для связи ЭВМ в систему можно использовать не специализированные системные каналы, а каналы связи с внешними устройствами СУС ВнУ, рассматривая одну ЭВМ по отношению к другой как внешнее устройство.

Используя особенности СУС ВнУ, можно свести к минимуму чисто системные особенности каналов связи. Блок системной связи должен быть выполнен с минимальными затратами оборудования. У пользователей одиночных ЭВМ и систем из двух ЭВМ «Минск-32» системные затраты составляют 0,2% стоимости и 0,5% оборудования всего комплекса.

Для подключения свыше двух ЭВМ в систему экономичнее использовать специализированные коммутаторы. В этом случае системные затраты составляют $(0,4 \div 1)$ % стоимости и $(1,5 \div 3,5)$ % оборудования всего комплекса, при этом повышается также надежность системы благодаря избыточности связей.

ГЛАВА VI
ВОПРОСЫ ОБЕСПЕЧЕНИЯ СОВМЕСТИМОСТИ С ЭВМ
«МИНСК-2», «МИНСК-22», «МИНСК-22М»

Совместимость становится одной из характерных особенностей ЭВМ широкого применения по следующим причинам.

Во первых, современное состояние проектирования и особенно использования ЭВМ показывает некоторую тенденцию отставания математического обеспечения (software) от технических средств (hardware). Это определяет низкое стартовое использование новых моделей ЭВМ. Совместимость позволяет в стартовый период использовать на новых моделях математическое обеспечение предыдущих ЭВМ и тем самым выровнять диспропорцию software и hardware.

Во-вторых, обеспечивая преемственность нескольких моделей ЭВМ, совместимость существенно снижает одно из наиболее весомых слагаемых стоимости эксплуатации ЭВМ широкого применения— затраты на программирование. В связи с этим понятие совместимость определяется [28], как один из наиболее употребимых терминов современной вычислительной техники. ЭВМ, удовлетворяющая понятию программной совместимости, должна выполнять ряд требований, наиболее существенными из которых являются:

- использование имеющихся программ непосредственно в новой ЭВМ;
- использование носителей информации и структуры набора данных старой модели;
- использование носителей структуры существующих наборов данных.

Известны 3 вида совместимости:

совместимость с машинами различных изготовителей;

совместимость с машинами различных поколений одного изготовителя;

совместимость от наименьших к наибольшим машинам внутри одного поколения машин, одного изготовителя или от наименьших к расширенным комплектам внутри одного семейства ЭВМ.

Несмотря на некоторые различия в подходе к проблеме совместимости, преемственность различных моделей является главной проблемой, в решении которой заинтересован изготовитель и заказчик. Объясняется это большим экономическим эффектом от реализации совместимости.

При разработке ЭВМ «Минск-32» учитывался тот фактор, что эта машина должна была заменить на рынке широко распространенную машину «Минск-22», не перечеркивая всего, что создано для семейства этих машин. Если принять, что при средней производительности программиста 5 команд в день стоимость одной команды составляет около 5 руб., и к концу 1969г. общий набор команд у заказчиков семейства «Минск-22» составлял около 10^7 команд, то стоимость всего математического задела этих машин превышала на тот период 50 млн. руб. Естественно, что при таком объеме математических программ ЭВМ «Минск-22» вполне необходимым и оправданным являлась реализация в следующей модели семейства ЭВМ «Минск» какого-то комплекса, обеспечивающего совместимость старой модели и новой.

6.1. ВОЗМОЖНОСТЬ ПРОГРАММНОГО МОДЕЛИРОВАНИЯ

Различные способы совмещения

Существует несколько способов совмещения различных моделей ЭВМ [28]. Пусть существует ЭВМ₁ и ЭВМ₂, характеризуемые множествами A_1, B_1, C_1 , и A_2, B_2, C_2 . Причем $A_1 = \{a'_1, a''_1, \dots, a^n_1\}$ и $A_2 = \{a'_2, a''_2, \dots, a^n_2\}$ являются множествами микрокоманд каждой ЭВМ, $B_1 = \{v'_1, v''_1, \dots, v^m_1\}$ и $B_2 = \{v'_2, v''_2, \dots, v^m_2\}$ являются множеством команд внутренней системы команд каждой ЭВМ, а $C_1 = \{c'_1, c''_1, \dots, c^p_1\}$ и $C_2 = \{c'_2, c''_2, \dots, c^p_2\}$ являются внешними множествами трансляторов с машинно-независимого языка D . Предметом рассмотрения являются две программы P_1 и P_2 двух ЭВМ. Множества A_1, A_2, B_1, B_2 являются внутренней характеристикой ЭВМ, а C_1, C_2, D — внешней.

Совмещение на уровне алгоритмического языка

Любая часть программы P_1 , если она соответствует множеству D , может быть включена в программу P_2 или являться независимой программой для ЭВМ₂ при наличии в ЭВМ₂ множества C_2 , т. е. если программа P_1 написана на языке D , то она может быть выполнена на ЭВМ₂, если у нее

есть транслятор C_2 с языка D на внутренний язык B_2 .

Трансляция

Необходимым условием трансляции для упомянутых двух ЭВМ является наличие дополнительного транслятора с языка B_1 на язык B_2 , определяемого множеством $T_2 = \{t'_2, t''_2, \dots, t^a_2\}$, причем $B_1 \subset T_2 \supset B_2$.

Трансляция — это процесс, переводящий каждую команду или небольшую группу команд из машинного языка старого процессора в язык нового. Трансляция выполняется один раз и предшествует времени выполнения программы. После этапа трансляции $P_1 = P_2$. Однако это сопряжено с очень большими трудностями, связанными с отличием полей команд от полей операндов, индексных ячеек от базисных. Требуется большое количество исходных данных. Трудно транслировать команду, которая модифицируется перед своим действием. А в современных ЭВМ процесс выборки и модификации команды очень сложен и отличается от машины к машине. Казалось бы, что трудно осуществить трансляцию программ только сильно отличающихся по структуре ЭВМ, однако ближайшее рассмотрение возможности трансляции программ машины «Минск-22» на язык очень близкой по структуре машины «Минск-22М» показало, что существует много трудностей, усложняющих трансляцию даже близких по структуре ЭВМ.

Моделирование

Моделирование отличается от трансляции тем, что каждая команда интерпретируется во время выполнения программы. В ЭВМ₂ существует интерпретирующее множество $U = \{i'_2, i''_2, \dots, i^n_2\}$. Отображение всего состояния машины будучи промоделированным сохраняется, так как оно должно представлять моделируемую машину бит в бит, символ в символ. Множество A_1 моделируемой машины ЭВМ₁ отображается и изменяется с помощью U в программах P_2 моделирующей ЭВМ₂. В отличие от трансляции моделирование должно интерпретировать команду каждый раз, когда она выполняется, поэтому должно происходить снижение производительности. Благодаря простоте моделирование очень часто используется (особенно при разработках новых моделей).

Эмуляция

Эмулятор — это программно-аппаратурный блок совмещающей ЭВМ₂, определяемый дополнительными множествами микрокоманд A'_2 , команд внутренней системы B'_2 и программ интерпретации U'_2 , причем $B_1 = B_2 + B'_2 + U'_2$. Так как множество A_2 является подмножеством B_2 ($A_2 \subset B_2$), то $B_1 = A_2 + A'_2 + B_2 + B'_2 + U'_2$.

Создание эмулятора требует разрешения нескольких проблем. Первая относится к изображению состояния моделируемой машины в эмулирующей ЭВМ. Так как блок эмулятора содержит как схемную часть A'_2 , так и средства программирования B'_2 , U'_2 , то встает вопрос о пропорции между этими блоками, о близости регистровой части эмулятора к структуре моделируемой ЭВМ и т. д.

Второй проблемой является моделирование системы связи с внешними устройствами особенно, если совмещаемые ЭВМ далеки по структуре друг от друга (и связанная с этим проблема моделирования и перекодирования носителей). Обычно эти функции возлагаются на программную часть эмулятора U'_2 . Следующей проблемой является выбор небольшой группы специальных команд B'_2 , облегчающих процесс эмуляции.

Все эти методы имеют свои преимущества и недостатки. Тщательный анализ всех способов совмещения не является целью данной работы, этому посвящены отдельные работы [28]. В данной главе дается обоснование выбора наиболее оптимального способа совмещения для ЭВМ широкого применения.

Моделирование двух различных по структуре ЭВМ

Анализ кратких особенностей различных способов, совместимости, приведенных в предыдущем разделе, показывает, что наибольший интерес для конструкторов ЭВМ широкого

применения представляют моделирование и эмуляция.

Как упоминалось выше, моделирование существенно увеличивает время решения задачи на новой модели. Требуется определить степень замедления и те дополнительные требования, которые выдвигает моделирование, как способ совместимости. Исходные данные: моделируемая машина— ЭВМ «Минск-22М», моделирующая —ЭВМ серии IBM-360, модель 40. Предлагается структура моделирующей программы (МП), блок-схема которой приведена на рис. 70.

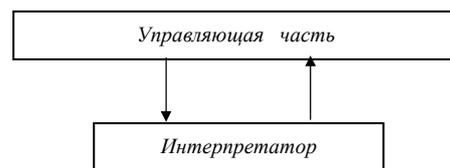


Рис.70.Блок-схема моделирующей программы.

Управляющая часть МП должна обеспечивать:

- подготовку к выполнению очередной программы ЭВМ «Минск-22М» (П22М);
- связь оператора с МП, принимая от оператора директивы и выдавая ему сообщения и указания:
- реакцию на сбои процессора и периферийного оборудования;
- определенные действия по устранению последствий сбоев; — систему автоматического прерывания, принятую в ЭВМ «Минск-22М».

Интерпретатор должен обеспечивать программную реализацию всей системы команд машины «Минск-22М». Всю систему команд ЭВМ «Минск-22М» можно разделить на две большие группы по следующим признакам:

1-я группа — команды ЭВМ «Минск-22М», которые имеют аналог в системе команд ЭВМ серии IBM-360;

2-я группа — команды ЭВМ «Минск-22М», которые не имеют соответствующих команд в системе команд ЭВМ серии IBM-360.

Блок-схема интерпретатора представлена на рис. 71. К командам 1-й группы можно отнести команды типа «Сложить с фиксированной запятой». «Логически сложить», «Переслать» и т. п. К командам 2-й группы можно отнести некоторые команды переходов, модификации индексной ячейки, специальные команды связи с внешними устройствами, работы с пульта, обслуживания прерывания и т. д.

Для того чтобы произвести примерную количественную оценку объема МП, необходимо построить блок-схемы некоторых подпрограмм. При рассмотрении блок-схем моделирующих

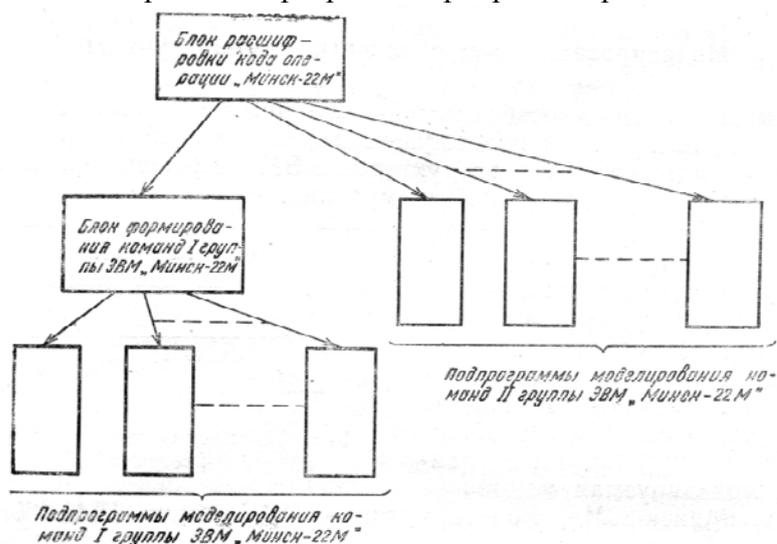


Рис. 71. Блок-схема интерпретатора.

программ, которые не приводятся в данной книге в силу своей громоздкости, были установлены их приблизительные объемы.

В табл. 6.1 приведен примерный объем подпрограмм моделирования различных групп команд машины «Минск-22М» на языке IBM-360. Приблизительно для реализации интерпретатора, исходя из данных таблицы, нужно около $2800 \div 3000$ команд системы IBM-360. Приводить в

данной работе громоздкие блок-схемы управляющей части моделирующей программы нецелесообразно, так как они отражают узкую специфику ЭВМ «Минск-22М».

Так как управляющая часть МП предназначена для выполнения тех же функций, которые выполняют монитор и анализатор сбоев в программной системе совместимости ЭВМ «Минск-32», можно с полным основанием утверждать, что объем управляющей части МП будет не меньше объемов монитора и анализатора сбоев. Принимаем эти объемы равными. Количество констант и рабочих ячеек для всей моделирующей программы принимаем таким же, как и в программной системе совместимости ЭВМ «Минск-32».

Итого, управляющая часть должна занимать около 3000 слов системы IBM-360. Моделирующая программа, состоящая из двух частей: интерпретатора и управляющей части — приблизительно должна занимать объем памяти около 3000 слов IBM-360. Если при этом учесть, что при моделировании вся моделируемая программа «Минск-22М» должна лежать в оперативной памяти (а ее объем составляет 8192 ячейки «Минск-22М»), то в машине со структурой IBM-360 должно быть выделено рабочее поле памяти для моделирования объемом

$$K > 22\,000 \text{ слов.}$$

Если принять процентное распределение команд в программах счета [25]: 70%—операции посылки, логики, переходов; 29% — операции типа сложения; 1% — операции типа умножения, то, исходя из данных табл. 6.1, замедление счета может быть подсчитано по следующему выражению

$$K = \frac{0,7P_{л} + 0,29P_{с} + 0,1P_{у}}{l}$$

где $P_{л}$ — количество команд блока моделирования операций логики, который равен 50, $P_{с}$ — количество команд блока моделирования операций сложения, который равен 400; $P_{у}$ — количество команд блока моделирования операций умножения, который равен 450; l — коэффициент увеличения быстродействия ЭВМ₂.

$$K \approx \frac{0,7 \cdot 50 + 0,29 \cdot 450 + 0,1 \cdot 450}{l} \approx \frac{200}{l}$$

Таблица 6.1

Команды ЭВМ "Минск-22"	Примерный объем подпрограммы моделирования в командах IBM-360
Операции арифметики с фиксированной запятой	450
Операции арифметики с плавающей запятой	400
Логические операции и операции пересылки	50
Операции передачи управления	110
Операции над индексами	250
Вспомогательные операции типа перехода по ключам, блокировки округления и т. д.	50
Операции связи с ВнУ	1000
Блок расшифровки кода операции.....	500

На практике приемлемым снижением быстродействия в результате совместимости считается величина $(1,2 \div 1,8)$. Исходя из этого, выражение для K примет вид

$$K = \frac{200}{(1,2 \div 1,8)l} = \frac{110 \div 165}{l}. \quad (6.1)$$

Из выражения (6.1) видно, что моделирующая ЭВМ₂ должна иметь быстродействие процессора на 2 порядка выше, чем у ЭВМ₁. Так, при моделировании на ЭВМ IBM-360/40, имеющей время выборки одного слова 2,5 мксек, задачи «Минск-22М», имеющей цикл МОЗУ 24 мксек, будут решаться в $10 \div 16$ раз медленнее, а на такой ЭВМ, как «Минск-32» с циклом МОЗУ 5 мксек, — в $20 \div 32$ раза медленнее.

Таким образом, моделирование программ одной ЭВМ на другой машине такого же класса, но отличной структуры нецелесообразно в связи с большой затратой машинного времени на моделирование и оперативной памяти для размещения моделирующей и моделируемой программ.

Основным выводом предыдущего раздела настоящей главы является то, что обеспечить совместимость машины «Минск-22М» путем моделирования на машине «Минск-32», имеющей процессор с производительностью в 5 раз выше, практически нельзя. Более производительным способом является эмуляция. Известны различные способы реализации эмуляторов [28]. Вопросы организации микропрограммного эмулирования уделено достаточно внимания [28], [29], [30], поэтому этот вариант не будет рассматриваться в данной работе. Реализация эмулятора в ЭВМ с жестким управлением имеет некоторую специфику. В данной работе рассматривается один из возможных вариантов эмулятора с жестким управлением, реализованный в ЭВМ «Минск-32» [31]. Рассматривается эмуляция ЭВМ «Минск-22М» в машине «Минск-32». Совместимость с машиной «Минск-22» не анализируется, так как она мало чем отличается от совместимости с «Минск-22М».

Некоторые аспекты реализации совместимости

Вся эмулирующая часть машины «Минск-32» состоит из двух независимых блоков: аппаратного блока совместимости; блока программ совместимости.

Благодаря эмуляции совместимость была обеспечена на достаточно высоком уровне:

— использование тех же носителей информации, что и при выполнении программ «Минск-22М» (П22М) на совмещаемой машине (исключением является магнитная лента с адресной структурой);

— ввод и вывод информации в таком же виде, как на машине «Минск-22М»;

— сохранение точности вычислений;

— сокращение времени выполнения П22М по сравнению с временем выполнения ее на машине «Минск-22М».

Аппаратурная и программная часть эмулятора распределены в следующем соотношении. Аппаратурная часть совместимости обеспечивает эмуляцию команд основной группы системы команд машины «Минск-22М»: арифметической, логической команд переходов. Появление любой из этих команд приводит к ее непосредственному выполнению на регистрах процессора машины «Минск-32». Выполнение этих команд ведется на уровне 1:1, т. е. каждой команде «Минск-22М» соответствует одна команда «Минск-32».

Программная часть эмулятора обеспечивает совместимость в основном «длинных» операций «Минск-22М»: работу с внешними устройствами (ВнУ), работу оператора, систему прерывания и сбоев. Это является целесообразным и оправданным с экономической точки зрения по следующим причинам.

Время выполнения операций работы ВнУ определяют прежде всего временные параметры работы механизмов, которые на несколько порядков ниже скорости работы процессора. Время решения задач, имеющих большие объемы работы ВнУ, определяется временем работы механизмов, участвующих в решении данной задачи. Поэтому любая моделирующая программа полностью укладывается в цикл работы обслуживаемого ею механизма. В этом случае можно ориентировочно считать

t моделируемой ЭВМ = t моделирующей $\approx t$ механизма.

Каналы связи эмулируемой и эмулирующей машины могут существенно отличаться. Значительно отличаются обычно и сами механизмы, коды устройств и носителей. Поэтому идти на аппаратное совмещение просто экономически невыгодно, так как это привело бы или к увеличению объемов аппаратуры, обслуживающей каналы связи с ВнУ, или к дублированию некоторых ВнУ.

Все это справедливо не только для связи с ВнУ, но и для системы прерывания, обслуживания сбойных ситуаций, связи оператора с машиной. Аппаратурная часть эмулятора совмещает 71 % команд «Минск-22М», программная часть — 29%.

В настоящей работе не рассматривается программная система совместимости, так как она подробно освещена в ряде работ [32], [36]. Укажем только ее основные функции:

1. Ввод и подготовка к выполнению П22М по управляющей информации, заданной оператором согласно инструкции выполнения этой программы на машине «Минск-22М».

2. Пуск программы «Минск-22М»,

3. Моделирование команд П22М, не реализованных аппаратно в «Минск-32».

4. Моделирование системы прерывания машины «Минск-22М».
5. Реакция на сбой машины «Минск-32» при выполнении П22М.
6. Реакция на прерывание по концу работы ВнУ.
7. Учет времени выполнения программы.
8. Реализация указаний оператора по управлению выполнением П22М и выдача сообщений оператору.

Для обеспечения программной совместимости машины «Минск-22М» в «Минск-32» предусмотрены следующие аппаратные средства:

- в качестве основной группы системы команд «Минск-32» взята группа арифметических и логических команд «Минск-22М». Некоторые специфичные команды являются к «Минск-32» экстракодовыми командами и моделируются специальными подпрограммами;
- разрядная сетка, форма представления чисел в целом идентичны в обеих машинах;
- специальный индикатор «М-22М», от состояния которого зависит способ выборки команд и формирования адресов команды, а также ход выполнения операций;
- основные носители информации (перфокарты, перфоленты) машины «Минск-22М» могут быть восприняты и воспроизведены на машине «Минск-32». Магнитная лента с машины «Минск-22М» не может быть прочитана на «Минск-32».

Основным указателем, настраивающим машину «Минск-32» на режим работы с программами «Минск-22М», является индикатор режима работы «М-22М». Индикатор режима работы «М-22М» 1 может быть установлен в необходимое состояние либо во время ввода данных программы на уровень этой программы, либо по специальной команде, если внутри программы «Минск-32» потребуется обращение к какой-то программе «Минск-22М». По состоянию индикатора «М-22М» формируются потенциалы команд, различно выполняемых в этих двух режимах и имеющих один код операции. Кроме того, по состоянию индикатора «М-22М» различно выполняется формирование адресов команды.

Совместимость структур команд

Структура типовых команд совмещаемых машин состоит из следующих слагаемых (рис. 72):

- кода операции (КОП);
- номера индексной ячейки (НИ);
- номера базисных ячеек для адресов a_1 и a_2 (b_1, b_2);
- относительных частей адресов обращения к памяти (a_1, a_2).

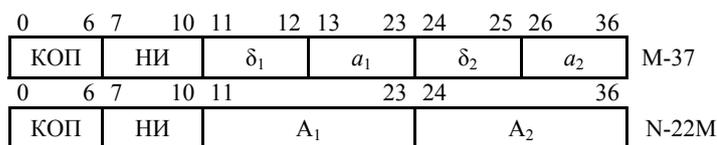


Рис. 72. Структура команд совмещаемых машин.

Совместимость в основном определила структуру команд новой модели «Минск-32». Структура команды «Минск-32» отличается от команды «Минск-22М» представлением адресных частей. Адрес в «Минск-22М» представляется в 13-разрядной форме, тогда как в «Минск-32» он представляется двумя компонентами: относительный адрес и базис.

При интерпретации команд «Минск-22М» на машине «Минск-32» представляется возможным и для команд «Минск-22М» рассматривать 13-разрядный адрес, состоящий из таких же частей, предъявив специальные требования к заданию базисного адреса и сохранив неизменной процедуру базирования (рис. 73). При интерпретации программ «Минск-22М» задаются константы базисного адреса как показано на рис. 74.

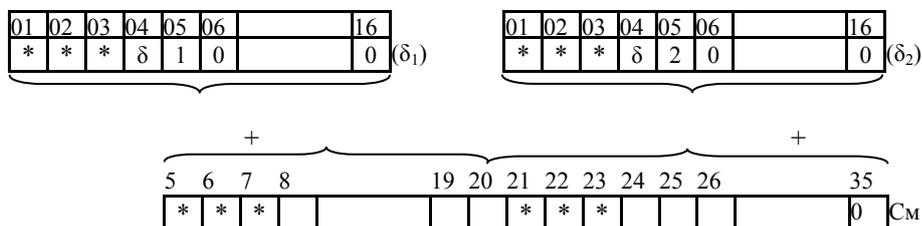


Рис. 73. Базирование при совместимости.

	01	02	03	04	05	06	10
1) для базиса 00	*	*	*	0	0	0	0
2) для базиса 01	*	*	*	0	1	0	0
3) для базиса 10	*	*	*	1	0	0	0
4) для базиса 11	*	*	*	1	1	0	0

Рис. 74. Базисные константы при совместимости.

При этом следует иметь в виду, что два старших бита каждого адреса A_1 и A_2 команды «Минск-22М» автоматически интерпретируются номерами базисов b_1 и b_2 . Таким образом, 13-разрядным адресом может быть адресована любая ячейка в пределах памяти «Минск-22М», оставшиеся три разряда базисного адреса (разряды с *) дают возможность располагать блоки программ «Минск-22М» в любых участках памяти «Минск-32».

Совместимость индексирования

Индексные приращения для программ «Минск-32» представляют собой 16-разрядные числа, записанные в разрядах $5 \div 20$ и $21 \div 36$ индексной ячейки. Для обеспечения совместимости машины сохранена структура индексной ячейки «Минск-22М», представляющая собой два 13-разрядных числа, расположенных в разрядах $11 \div 23$ и $24 \div 36$. Для программ «Минск-22М» прием индексных приращений осуществляется особым образом (см. рис. 14). Индексное приращение для первого адреса принимается со сдвигом в разряды $8 \div 20$ регистра P1 и сумматора, а индексное приращение для второго адреса — в разряды $24 \div 36$ регистра P1 и сумматора

Совместимость команд внешних устройств и управление программой совместимости

Новая организация связи с внешними устройствами (СУС ВнУ) в машине «Минск-32» позволила упростить и сократить количество команд обращения к внешним устройствам. Команды обращения к внешним устройствам в машине «Минск-22М» и в ЭВМ «Минск-32» воспринимаются как экстракоды, которые вызывают переключение на специальные моделирующие программы, в которых соответствующие команды реализуются точно так же, как они выполняются на машине «Минск-22М».

В ЭВМ «Минск-32» при решении любой программы работа аппаратуры процесса управляется информацией, содержащейся в ячейках программного уровня. Каждой программе, в том числе и совмещаемой, соответствует свой программный уровень. При эмуляции программ «Минск-22М» на машине «Минск-32» необходимо в ячейки с номером 0 и 1 уровня совмещаемой программы записать нужные базисные константы. Индикатор «М-22М» должен быть установлен в единичное состояние, и его состояние записано вместе с индикаторами, управляющими работой аппаратуры процессора, в левую часть ячейки с номером 5. В правую часть этой ячейки записывается константа начального адреса программы, в ячейку с номером 3 записываются константы верхней и нижней границ защиты для совмещаемой программы. Индексные ячейки в программах М-22М располагаются в начале программы, поэтому константа базиса индекса, которая записывается в ячейку с номером 4, совпадает с начальным адресом блока МОЗУ, в котором расположена программа. Структура уровня программы совместимости приведена на рис. 75, где приняты обозначения: б — константа базиса; К — начальный адрес программы «Минск-22М»; См — сумматор; № уровня — номер уровня.

Таким образом, обеспечить совместимость двух различных по структуре ЭВМ одного класса путем моделирования практически невозможно в связи с большими затратами объемов оперативной памяти и существенным увеличением времени решения программ моделируемой ЭВМ. Очевидно, что моделирование машин малого и среднего классов возможно на машинах более высокого класса. Задачи будут решаться на моделирующей ЭВМ с той же эффективностью, если ее процессор имеет быстроедействие на два порядка выше, чем у моделируемой.

Для упрощения решения проблемы совместимости в машинах широкого применения можно использовать принцип «жесткой» эмуляции, когда совмещающая машина приближается в некоторых вопросах к совмещаемой ЭВМ (выбор одной и той же разрядной сетки близкого формата команд).

При незначительных затратах на реализацию блока совместимости (1,7% от общего оборудования процессора и около 6 тыс. команд программной части эмулятора вместе с блоком «Диспетчера» совместимости) в ЭВМ «Минск-32» получены следующие результаты:

- эмулируемые программы «счета» выполняются в $3 \div 4$ раза быстрее, чем на «Минск-22М»;
- эмулируемые программы, имеющие большой удельный вес работы с ВнУ, решаются в пропорции 1:1, так как время решения этих задач на новой модели определяется коэффициентом K_c :

$$K_c = \frac{\sum V_{ВнУН}}{\sum V_{ВнУСт}}$$

где $\sum V_{ВнУСт}$ — суммарная скорость работы ВнУ при решении

<i>N</i> ячейки уровня	<i>Содержимое ячейки</i>	
0	$\delta + 04\ 000$	$\delta + 00\ 000$
1	$\delta + 14\ 000$	$\delta + 10\ 000$
2	<i>Свободная</i>	
3	$\delta + 20\ 000$	$\delta + 00\ 000$
4		$\delta + 00\ 000$
5	1	+ индикаторы $\delta + \kappa$
6	<i>Содержимое См</i>	
7	<i>N уровня</i>	

Рис. 75. Структура программного уровня совмещаемой программы.

задачи на совмещаемой ЭВМ, а $\sum V_{ВнУН}$ — суммарная скорость работы ВнУ при решении задач на совмещающей ЭВМ.

Принцип «жесткой» эмуляции не накладывает ограничений на развитие дополнительных эффективных возможностей в новой модели. Приближение по структуре совмещаемой и совмещающих моделей не означает простого копирования старой модели на более производительной, новой. В машине «Минск-32» наряду с совместимостью со старыми моделями «Минск-2», «Минск-22», «Минск-22М» удалось реализовать такие новые логические возможности, как многопрограммность, возможность расширения емкости МОЗУ до 64К, использование десятичной арифметики, организация СУС ВнУ, управление с помощью операционной системы, возможность организации системы нескольких ЭВМ и т. д.

Наряду с совместимостью со старыми моделями в структуре машин широкого применения необходимо закладывать дополнительные возможности для создания перспективных совместимых комплексов. В частности, СУС ВнУ должна быть обязательной принадлежностью машин широкого применения как гибкое средство изменения комплектности ЭВМ.

СИСТЕМА МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ЭВМ «МИНСК-32»

Система обеспечивает мультипрограммную (до четырех рабочих программ) пакетную обработку потока программ, составленных для машин «Минск-32» и «Минск-22».

Операционная система «Диспетчер» обеспечивает: связь оператора с машиной, организацию очереди заданий на выполнение программ, загрузку программ в память и их пуск, автоматический переход выполнения одной программы к другой, последовательное выполнение программ машины «Минск-22», одновременное выполнение до четырех программ машины «Минск-32», распределение внешних устройств и памяти между одновременно выполняющимися программами, учет текущего и израсходованного отдельными программами времени, обработку сбойных ситуаций.

Система программирования базируется на стандартной структуре программного модуля и единой методике сборки-загрузки.

В состав МО входят следующие компоненты:

система тестового контроля; система программирования: библиотечная система, обслуживающие программы, система символического кодирования; операционная система «Диспетчер»: формирователь системной ленты, начальный вызов, резидент, координатор, совместимость, разгрузка ленты вывода, вся система МО ЭВМ «Минск-22». При разработке системы тестового контроля ЭВМ «Минск-32» был учтен опыт наладки и эксплуатации ЭВМ «Минск-22» и «Минск-23». Особенностью системы является четкая реализация принципов блочной структуры и унификации использования тестов. Построенные по блочному принципу тесты состоят из наборов отдельных проверок, каждая из которых проверяет отдельный узел устройства и минимально использует другие узлы и устройства машины. По желанию оператора любая из проверок может быть исключена из выполнения или выполняться многократно. Максимальная стандартизация действия оператора по пуску тестовых программ, управлению их выполнением после обнаружения сбоя в работе машины облегчает использование тестов обслуживающим персоналом.

Система тестового контроля состоит из двух блоков: наладочных и проверочных тестов. Наладочные тесты позволяют осуществлять инженерную отладку отдельных устройств вовремя комплексной наладки машины. Пуск и работа тестовых программ осуществляется с инженерного пульта. Проверочные тесты проверяют правильность функционирования отдельных узлов машины в составе программно-аппаратурной системы «Минск-32». Тесты рассматриваются, как одна из рабочих программ, пуск их и работа управляются операционной системой «Диспетчер».

Система программирования ЭВМ «Минск-32» представляет собой совокупность средств, с помощью которых осуществляется подготовка программ к выполнению.

В базовый состав системы программирования входят:

— библиотечная система с комплектом управляющих и обслуживающих программ «Библиотекарь»;

— программы, обслуживающие процесс подготовки программ («Корректор», «Сборщик», «Загрузчик»);

— система символического кодирования.

Система программирования применяет модулярную технологию и открыта для пополнения другими языками программирования и трансляторами.

Система программирования предполагает стандартную внешнюю структуру любой программы, используемой для работы на машине. Стандартная структура программ позволяет «Библиотекарю» легко включать в библиотеку новые программы, осуществлять поиск и выдачу нужных программ и исключение устаревших программ из библиотеки. В минимальный состав библиотеки входят первоочередные программы обработки данных (программы перекодировок и переводов, ввода-вывода, редактирования, сортировки, отладки и др).

Любая программа может состоять из одной или нескольких составных частей. Каждая из таких частей может быть написана как отдельная программа на каком-либо входном языке, имеющемся в системе программирования. При необходимости (для исправления или модификации) в программы на входных языках могут вноситься изменения с помощью программы «Корректор». Части программы, записанные на различных языках, транслируются соответствующими **трансляторами**. Каждый транслятор переводит программы на единый внутренний язык системы программирования — язык загрузки. Такое оформление результатов трансляции делает их

независимыми от входных языков и позволяет записывать отдельные части программ на различных входных языках. Программа на языке загрузки не зависит от места выполнения ее в оперативной памяти и содержит в себе, кроме кодов команд, констант и т. п. дополнительную информацию, которая используется при размещении ее объектов в оперативной памяти, а также сведения о других программах, используемых в ней.

Перед выполнением части программы на языке загрузки собираются программой «Сборщик» в единое целое. Результат сборки рассматривается как самостоятельная программа на языке загрузки. Собранная программа может быть вновь присоединена к любой составляемой программе наравне с другими программами.

Перед выполнением программа на языке загрузки должна быть загружена в выделенное для нее место памяти. Под загрузкой программы понимается размещение объектов программы в памяти машины в соответствии с принятым в системе программирования способом распределения памяти и соответствующая их настройка. Результатом загрузки является машинная программа, настроенная по месту в памяти, куда она загружена и на котором она будет выполняться. Загрузку программ осуществляет программа «Загрузчик».

Системные средства программирования вместе с набором часто используемых программ обработки составляют библиотеку программ общего пользования, хранящуюся на системной магнитной ленте. Наряду с этим предусмотрена возможность организации на магнитных лентах пользователей личных библиотек пользователей.

Система символического кодирования, разработанная для ЭВМ «Минск-32», имеет назначение и организацию, аналогичные системам ССК для ЭВМ «Минск-22» и «Минск-23». Результаты трансляции представляются в едином внутреннем языке загрузки. С помощью этой системы разработаны, документированы и эксплуатируются все компоненты МО ЭВМ «Минск-32».

Операционная система «Диспетчер»— набор специальных программ, управляющих выполнением на машине «Минск-32» потока программ, составленных для машин «Минск-32» и «Минск-22». Система «Диспетчер» обеспечивает:

- связь оператора с машиной через пишущую машинку пульта оператора;
- организацию очереди заданий на выполнение программ;
- загрузку программ в оперативную память машины и их пуск;
- автоматический переход от выполнения одной программы к другой;
- последовательное выполнение программ машины «Минск-22»;
- одновременное выполнение до четырех программ машины «Минск-32»;
- распределение внешних устройств и памяти между одновременно выполняющимися программами;
- автоматическое накопление выводимой информации на магнитную ленту вывода в тех случаях, если в данный момент отсутствует свободное физическое устройство для вывода;
- работу внешних устройств одновременно с работой вычислителя;
- учет текущего времени и времени выполнения вычислителем отдельных программ;
- сигнализацию об ошибках в программах и работе оборудования и частичное исправление последствий этих ошибок.

Указанные функции реализуются программами «Диспетчера»:

формирование ленты системы;

начальный вызов

резидент;

координатор;

совместимость;

разгрузка ленты вывода.

Программа «**Формирование ленты системы**» настраивает «Диспетчер» в соответствии со списком оборудования, входящего в состав конкретной конфигурации машины. Этот список содержит сведения об объеме оперативной и внешней памяти, о наличии определенных типов внешних устройств и номеров связей, используемых ими при подключении к машине. По этим данным программа формирует таблицу оборудования и размещает ее на магнитную ленту системы. На нее же записываются программы «Координатор», «Совместимость» и «Резидент», а в

дальнейшем записывается и библиотека программ. Корректировка таблицы производится при изменении состава оборудования. Наличие таблицы оборудования позволяет составителю программ указывать лишь условные обозначения внешних устройств. А распределение внешних устройств между программами осуществляется «Диспетчером» автоматически.

Программа «**Начальный вызов**» считывает с магнитной ленты системы в оперативную память часть «Диспетчера» («Резидент», Координатор») и таблицу оборудования и настраивает «Диспетчер» к выполнению программ машины «Минск-32».

Программа «**Координатор**» обеспечивает выполнение одновременно до четырех программ машины «Минск-32» и автоматический переход от выполнения одной программы к другой.

Программа «**Совместимость**» управляет непрерывным процессом выполнения программ машины «Минск-22».

При своем функционировании программы «Координатор» и «Совместимость» сменяют друг друга на одном и том же месте оперативной памяти.

Программа «**Резидент**» постоянно находится в оперативной памяти и производит автоматическую смену в оперативной памяти

и контроль правильности смены программ «Совместимость» и «Координатор» при необходимости выполнения программ той или другой машины.

Программа «**Разгрузка ленты вывода**» обеспечивает выдачу на перфокарты или печать результатов, записанных на магнитную ленту вывода из-за отсутствия в нужный момент соответствующего оборудования.

Структура и функции «Координатора». В состав координатора входят следующие основные части:

- Монитор;
- Супервайзер;
- Администратор;
- Анализатор сбоев вычислителя.

Монитор:

- обеспечивает двухстороннюю связь оператора с «Координатором»;
- принимает задания на выполнение программ и формирует из них очередь;
- выбирает для выполнения задания из очереди в соответствии с приоритетом и требуемым объемом оперативной и внешней памяти;
- загружает программы в оперативную память и осуществляет их пуск;
- заканчивает выполнение одних программ и переходит к выполнению других.

Для связи оператора с «Координатором» разработан язык директив и запросов, поступающих от оператора, и язык указаний и сообщений, выдаваемых «Координатором».

Язык директив и запросов оператора позволяет:

- поставить в очередь задание на выполнение программ как машины «Минск-32», так и машины «Минск-22»;
- исключить из очереди любое задание;
- запретить на некоторое время выполнение любой программы и затем продолжить ее выполнение;
- изменять приоритет выполняющихся программ;
- исключить из работы временно вышедшее из строя внешнее устройство и включить его в работу после ремонта;
- передать «Координатору» календарные данные (дату, время) ;
- запросить состояние очереди заданий, ресурс памяти и внешних устройств;
- сообщать о принятом решении и выполненных действиях, затребованных «Координатором» или программами;
- повторять выполнение программы с заданных ею адресов
- переходить к выполнению программ машины «Минск-22».

Язык сообщений и указаний позволяет «Координатору» выдавать на пишущую машинку пульта оператора:

- ответы на директивы и запросы оператора;
- сообщения о ходе выполнения программ;
- сообщения о неисправной работе устройств;

- текущее время через определенные интервалы;
- сообщения о занятости и освобождении внешних устройств программы;
- указания оператору от «Координатора» и программ о необходимости принятия решений и выполнения определенных действий.

Супервайзер:

распределяет внешние устройства между одновременно выполняющимися программами и управляет работой этих устройств. При возникновении сбоя в работе внешних устройств «Супервайзер» информирует об этом оператора и устраняет последствия сбоя либо самостоятельно, либо с участием оператора.

Администратор:

- следит за выполнением различных блоков «Координатора»;
- представляет вычислитель одной из одновременно выполняющихся программ в соответствии с ее приоритетом;
- ведет учет текущего времени и времени выполнения вычислителем отдельных программ;
- передает программе календарные данные;
- включает работу программы через заданные интервалы времени и по запросам внешних устройств.

Анализатор сбоев вычислителя:

- реагирует на сбои вычислителя (переполнение разрядной сетки, нарушение защиты, недействительный код команды и информации и т. д.);
- информирует оператора о сбое в работе вычислителя;
- организует повторное выполнение некоторой части этой программы с заданного программой адреса;
- прекращает выполнение программы при повторных сбоях и отсутствии адреса для повторения.

«Диспетчер» может функционировать при любом составе машины за счет наличия специальной таблицы оборудования, формируемой для каждого конкретного комплекта машины и хранимой вместе с программами «Диспетчер» (резидент, совместимость, координатор и разгрузка ленты вывода) на специальной магнитной ленте системы.

«Диспетчер» является неотъемлемой частью логики ЭВМ «Минск-32». Он не ориентирован на какую-то специальную систему программирования и обеспечивает эффективное использование оборудования машины и возможность вмешательства оператора в процессе прохождения задач на машине.

При работе ЭВМ над программой «Диспетчер» использует около 6К ячеек оперативной памяти, центральный пульт с пишущей машинкой, два НМЛ для магнитных лент системы и вывода.

Программа «Совместимость» вместе с аппаратными средствами, заложенными в структуру ЭВМ «Минск-32», обеспечивает выполнение на этой машине всех машинных программ ЭВМ «Минск-22». Возможность использования математического обеспечения машины «Минск-22» позволяет ускорить этап внедрения новой модели в народное хозяйство.

В состав «Совместимости» входят следующие основные части:

Монитор;

Интерпретатор;

Анализатор сбоев.

Интерпретатор осуществляет программную имитацию логических возможностей машины «Минск-22», не реализованных аппаратам в машине «Минск-32». **Монитор** и **Анализатор** сбоев выполняют функции, сходные с функциями аналогичных частей «Координатора». Отличия в их структуре и работе вызваны спецификой режима совместимости и необходимостью имитации связи оператора с ЭВМ «Минск-22».

Совместимость позволяет:

- уменьшить в 4 раза время выполнения программ машины «Минск-22» вычислительного характера (с небольшим объемом использования магнитной ленты);
- облегчить работу оператора, так как манипуляции оператора с многочисленными кнопками и клавишами на машине «Минск-22» заменены работой на клавиатуре пишущей машинки;
- автоматизировать процесс последовательного выполнения программ;
- помочь оператору и инженеру принять правильное решение при сбоях в работе

оборудования. Для сбоя внешних устройств в зависимости от вида сбоя во многих случаях осуществляется автоматическое исправление ошибки путем повторного выполнения операции (например, повторный обмен с магнитной лентой, повторный вывод карты);

— получить протокол работы на машине с указанием времени выполнения каждой отдельной программы.

Совместимость не требует никаких изменений первичных носителей программ и исходных данных (перфоленты и перфокарты) и сохраняет точность вычислений и форму выдаваемых результатов.

Система команд

Группа	Название команды	Код команды	Вид команды	Название СКМ	Последовательность тактов		Время в мксек.	Примечания
					Структура кода команды	Структура индексной ячейки		
Операции арифметики с фиксированной запятой	Сложить	+10	L 01a1 02a2	СФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 40		
		+11	L 01a1 02a2	СФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 35		
		+12	L 01a1 02a2	СФ	K-инд → 01 → 02 → 14т → А → 23П	20 + 25		
	Вычесть	+13	L 01a1 00	СФ	K-инд → 01 → 14т → А → 23П	35 + 40		
		+21	L 01a1 02a2	ВФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 35		
		+22	L 01a1 02a2	ВФ	K-инд → 01 → 02 → 14т → А → 23П	20 + 25		
	Вычесть модули	+30	L 01a1 02a2	ВМФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 40		
		+51	L 01a1 02a2	ВМФ	K-инд → 01 → 02 → 14т → А → 23П	30 + 35		
		+52	L 01a1 02a2	ВМФ	K-инд → 01 → 02 → 14т → А → 23П	30 + 55		
	Умножить	+31	L 01a1 02a2	УФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 130		
+32		L 01a1 02a2	УФ	K-инд → 01 → 02 → 14т → А → 23П	30 + 125			
+33		L 01a1 00	УФ	K-инд → 01 → 14т → А → 23П	30 + 126			
Разделить	+40	L 01a1 02a2	УФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 115			
	+41	L 01a1 02a2	ДФ	K-инд → 01 → 02 → 14т → 24т → А → 23П	20 + 125			
	+42	L 01a1 02a2	ДФ	K-инд → 01 → 02 → 14т → А → 23П	30 + 210			
Операции арифметики с плавающей запятой	Сложить	+70	L 01a1 02a2	УФП	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 330		
		+71	L 01a1 02a2	УФП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 235		
		+72	L 01a1 02a2	УФП	K-инд → 01 → 02 → 14т → А → 23П	30 + 230		
	Вычесть	+73	L 01a1 00	УФП	K-инд → 01 → 14т → А → 23П	30 + 276		
		+74	L 01a1 02a2	ДФП	K-инд → 01 → 02 → 14т → 24т → А → 23П	40 + 250	п-число операндов по А1	
		+75	L 01a1 02a2	ДФП	K-инд → 01 → 02 → 14т → А → 23П	30 + 85		
	Арифметически сдвинуть	+84	L 01a1 02a2	АСД	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 100		
		+85	L 01a1 02a2	АСД	K-инд → 01 → 02 → 14т → А → 23П	30 + 95		
		+86	L 01a1 02a2	АСД	K-инд → 01 → 02 → 14т → А → 23П	30 + 85		
		+87	L 01a1 00	АСД	K-инд → 01 → 14т → А → 23П	20 + 85		
Операции арифметики с плавающей запятой	Сложить	+14	L 01a1 02a2	СП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 165		
		+15	L 01a1 02a2	СП	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 150		
		+16	L 01a1 02a2	СП	K-инд → 01 → 02 → 14т → А → 23П	30 + 150		
	Вычесть	+24	L 01a1 02a2	ВП	K-инд → 01 → 02 → 14т → 24т → А → 23П	20 + 145		
		+25	L 01a1 02a2	ВП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 155		
		+26	L 01a1 02a2	ВП	K-инд → 01 → 02 → 14т → А → 23П	30 + 150		
	Вычесть модули	+27	L 01a1 00	ВП	K-инд → 01 → 14т → А → 23П	20 + 145		
		+54	L 01a1 02a2	ВМП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 155		
		+55	L 01a1 02a2	ВМП	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 150		
	Умножить	+56	L 01a1 02a2	УП	K-инд → 01 → 02 → 14т → А → 23П	30 + 150		
+57		L 01a1 00	УП	K-инд → 01 → 14т → А → 23П	20 + 145			
+34		L 01a1 02a2	УП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 180			
Разделить	+35	L 01a1 02a2	УП	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 175			
	+36	L 01a1 02a2	УП	K-инд → 01 → 02 → 14т → А → 23П	30 + 175			
	+37	L 01a1 00	УП	K-инд → 01 → 14т → А → 23П	20 + 165			
Служить	+44	L 01a1 02a2	ДП	K-инд → 01 → 02 → 14т → 24т → А → 23П	35 + 240			
	+45	L 01a1 02a2	ДП	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 235			
	+46	L 01a1 02a2	ДП	K-инд → 01 → 02 → 14т → А → 23П	30 + 235			
Вычесть	+47	L 01a1 00	ДП	K-инд → 01 → 14т → А → 23П	20 + 225			
	+72	L 01a1 02a2	СПП	K-инд → 01 → 02 → 14т → 24т → А → 23П	40			
Вычесть	+73	L 01a1 02a2	СПП	K-инд → 01 → 02 → 14т → 24т → А → 23П	40			

Продолжение

Группа	Название команды	Код команды	Вид команды	Название СКМ	Последовательность тактов	Время в мксек.	Примечания	
Операции арифметики с фиксированной запятой	Сложить	+01	L 01a1 00	СД	K-инд → 01 → 14т → 0П1 → А → 23П	25 + 95		
	Вычесть	+02	L 01a1 00	СД	K-инд → 01 → 14т → 0П1 → А → 23П	25 + 95		
	Умножить	+03	L 01a1 00	УД	K-инд → 01 → 14т → 0П1 → А → 23П	25 + 620		
	Умножить целые числа	+04	L 01a1 02a2	УДЦ	K-инд → 01 → 02 → 14т → 24т → 23П	35		
	Логические операции	Поразрядно сложить	+05	L 01a1 02a2	ЛСЯ	K-инд → 01 → 02 → 14т → 24т → 23П	30	
		Логически сложить	+06	L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 23П	30	
			+07	L 01a1 00	ЛС	K-инд → 01 → 14т → А → 23П	20	
			+74	L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 24т → 23П	35	
		Логически умножить	+75	L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 24т → 23П	30	
	+76		L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 23П	30		
+77	L 01a1 00		ЛС	K-инд → 01 → 14т → А → 23П	20			
+70	L 01a1 02a2		ЛС	K-инд → 01 → 02 → 14т → 24т → 23П	35			
Логически сдвинуть	+71	L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 24т → 23П	30			
	+72	L 01a1 02a2	ЛС	K-инд → 01 → 02 → 14т → 23П	20			
	+73	L 01a1 00	ЛС	K-инд → 01 → 14т → А → 23П	35 + 100			
	+80	L 01a1 02a2	ЛСД	K-инд → 01 → 02 → 14т → 24т → А → 23П	30 + 95			
Операции пересылки	Переслать	+10	L 01a1 02a2	ПСА	K-инд → 01 → 02 → 14т → 23П	30 + 200	п-число побитовых пересылок в случае группового режима	
	Переслать с обратным знаком	+11	L 01a1 02a2	ПСАИ	K-инд → 01 → 02 → 14т → 23П	30		
	Переслать модуль	+12	L 01a1 02a2	ПСАМ	K-инд → 01 → 02 → 14т → 23П	30		
	Переслать знак	+14	L 01a1 02a2	ПСАИЗ	K-инд → 01 → 02 → 14т → 24т → 23П	35		
	Обменять	+15	L 01a1 02a2	ПСАМ	K-инд → 01 → 02 → 14т → 24т → 23П → 23П	40 + 120 + 200		
	Переслать по порядку	+16	L 01a1 02a2	ПСАП	K-инд → 01 → 02 → 14т → 24т → 23П	35		
	Переслать в четный побитовый	+17	L 00 02a2	ПСАЧ	K-инд → 02 → 24т → А → 23П	20		
	Операции управления	Структура команд-01, 02	-01	01c 02a2	ПСС	K-инд → 02 → 0П1 → 24т	25	
Прочитать символ		-02	01c 02a2	ПС	K-инд → 02 → 0П1 → 23П → 24т	30		
Записать символ		-30	01a1 02a2	ПЗП	K-инд → 01 → 02 → 23П	25		
Идти и записать		-36	L 01a1 02a2	ПЗП	K-инд → 01 → 02 → 24т	25		
Идти по знаку		-32	L 01a1 02a2	ПЗП	K-инд → 01 → 02 → 24т	15		
Идти по нулю		-34	L 01a1 02a2	ПЗП	K-инд → 01 → 02 → 24т	15		
Идти по несовпадению		-37	L 01a1 02a2	ПЗП	K-инд → 01 → 02 → 24т	30		
Идти по четности		-22	L 01a1 02a2	ПЗП	K-инд → 01 → 02 → 24т → 0П1 → 23П	35 + 30		
Идти к подпрограмме		-31	L 01a1 02a2	ПЗП	K-инд → 01 → 14т → 24т → 23П	30 + 100	п-число команд, выполняемых между пересылками	
Идти из подпрограммы		-21	L 01a1 02a2	ПЗП	K-инд → 01 → 14т → 24т → 23П → 24т → 23П → 24т → 23П → 24т → 23П	35		
Операции передачи	Идти по указателю "Пепелание"	-33	L 01a1 02a2	ПЗП	K-инд → 01 → 14т → 24т → 23П → 24т → 23П → 24т → 23П → 24т → 23П	15	п-переход по адресу А1 при установленном указателе п-0-переход по адресу А2 при установленном указателе	

Группа	Название команды	Код команды	Вид команды	Название ССК	Последовательность тактов	Время в мксек	Примечание	
Операции над индикаторами	Модифицировать индекс	-20 01р		ММ	$\begin{matrix} 7 & 4 & 1 & 4 & 8 & 2 & 11 \\ \text{кол} & \text{с} & \text{оп} & \text{свободны} & \text{оп} & \text{оп} & \end{matrix}$			
	Формирование адреса символа	-20 0000		ММЛ00	К → 62 → 24т → 44т → А → 43П	30		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0001		ММЛ01	К → 62 → 24т → 44т → А → 43П	30		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0010		ММЛ02	К → 62 → 24т → 44т → А → 43П	30		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0011		ММЛ03	К → 62 → 24т → 44т → А → 43П	25+120		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0100		ММЛ10	К → 62 → 24т → 44т → 43П	25		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0110		ММЛ12	К → 62 → 24т → 44т → 43П	25		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 0111		ММЛ13	К → 62 → 24т → 44т → 43П	25		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 1000	-201 01a 02a2	ММЛ20	К → 62 → 24т → 43П	15		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 1010		ММЛ22	К → 62 → 44т → 23П	20		
	$(0a_1 + 1) + (0a_2 + 0a_1) \rightarrow (0a_1 + 1)$	-20 1011		ММЛ23	К → 62 → 44т → 23П	15		
	$\Delta 1(0a_1 + 0a_2) \rightarrow \Delta 1(0a_1 + 1)$	-20 1100		ММЛ30	К → 62 → 24т → 44т → 43П	25		
	$\Delta 2(0a_1 + 0a_2) \rightarrow \Delta 2(0a_1 + 1)$	-20 1101		ММЛ31	К → 62 → 24т → 44т → 43П	25		
	$\Delta 2(0a_1 + 0a_2) \rightarrow \Delta 2(0a_1 + 1)$	-20 1110		ММЛ32	К → 62 → 24т → 44т → 43П	25		
	$\Delta 2(0a_1 + 0a_2) \rightarrow \Delta 2(0a_1 + 1)$	-20 1111		ММЛ33	К → 62 → 24т → 44т → 43П	25		
	Операции переключения команд	Включить в линию со ступенчатой нагрузкой	-05	-251 01a 02a	УБП	К → инд ¹ → 61 → 14т → 23П	20+25	
		Переключить на ступенчатую нагрузку	-05	-051 01a 02a	УБП	К → инд ¹ → 61 → 14т → 24т	25	
		Выполнить команду	-25	-251 01a 02a	ВЫП	К → инд ¹ → 61 → 14т → (M' → ...)	25+30	Чл - время выполнения команды M'
Идти и изменить базис		-28	-281 01a 02a	УБИ	К → инд ¹ → 61 → 14т → 44т → ОП1 → 43П	30+35		
Операции обмена с УОИ	Модифицировать ячейку уровня	-17 01р		МУ	$\begin{matrix} 7 & 4 & 1 & 4 & 8 & 2 & 11 \\ \text{кол} & \text{с} & \text{оп} & \text{свободны} & \text{оп} & \text{оп} & \end{matrix}$			
	$(N) + (R_1) \rightarrow N$	-17 0001		МУЛ01	К → инд ¹ → 62 → 24т → 44т → А → 43П	30+35		
	$(N) + (R_2) \rightarrow N$	-17 0010		МУЛ02	К → инд ¹ → 62 → 24т → 44т → А → 43П	30+35		
	$(N) + (R_3) \rightarrow N$	-17 0101		МУЛ11	К → инд ¹ → 62 → 24т → 44т → 43П	30		
	$(N) + (R_4) \rightarrow N$	-17 0110		МУЛ12	К → инд ¹ → 62 → 24т → 44т → 43П	30		
	$(N) + (R_5) \rightarrow N$	-17 0111	-171 01a К 02a	МУЛ13	К → инд ¹ → 62 → 24т → 44т → 43П	30		
	$(R_1) \rightarrow N$	-17 1000		МУЛ20	К → инд ¹ → 62 → 24т → 43П	20+25	п - количество переданных слов	
	$(R_2) \rightarrow N$	-17 1001		МУЛ21	К → инд ¹ → 62 → 24т → 43П	20+25		
	$(R_3) \rightarrow N$	-17 1010		МУЛ22	К → инд ¹ → 62 → 44т → 23П	20+25		
	$(R_4) \rightarrow N$	-17 1011		МУЛ23	К → инд ¹ → 62 → 44т → 23П	15+20		
Вспомогательные операции экстракоды	Установить индикаторы	-04 1р	-04 1г Инд 00	У	$\begin{matrix} 7 & 4 & 1 & 12 \\ \text{кол} & \text{с} & \text{оп} & \text{Индикаторы} & \text{свободные} & \end{matrix}$	15		
	Индикатор - вкл оп	-04 13р			К → инд ¹ → 62			
	Индикатор - вкл норм	-04 14р						
	Индикатор - нуль	-04 15р						
	Ничего не делать	* 00	-001 01a 02a	НЕТ	К → инд ¹ → 61	15		
	Экстракоды 7+13УР	-10+47 -50+57 -60+56 -65+67				К → инд ¹ → 61 → 14т → абт. прерывание на 6 уровень К → инд ¹ → 61 → 14т → абт. прерывание на 5 уровень	10+15 20	

Группа	название команды	Код команды	Вид команды	Название ССК	Последовательность тактов	Время в мксек	Примечание
Операции установки и определения неготовности составной машины	Идти по указателю вычислителя	-30 1р	-33 с 01р 02a2	МУЦУ	$\begin{matrix} 7 & 4 & 1 & 8 & 2 & 11 \\ \text{кол} & \text{с} & \text{оп} & \text{свободны} & \text{оп} & \text{оп} & \end{matrix}$	15	
	Вид перехода	-33 1р					
	Указатель "переполнение"	-33 13р					
	Указатель "неиспольз. код"	-33 14р					
	Указатель "нецифра"	-33 15р					
	Указатель "свой по защите"	-33 16р					
	Указатель "свой по нечету"	-33 17р					
	Указатель "свой ПЗР (свой при переключении уровня)"	-33 18р					
	Идти по указателю устройства обмена	-23 1р	-23 с 01р 02a2	МУУО	К → инд ¹ → 62	15	
	Указатели: занят быстрый канал (БК)	12р					
	Свой по нечету УС	13р					
	Свой по защите УС	14р					
	Свой по информационного слова	15р 16р					
	ЗДВ	16р					
	Включить - выключить ЗДВ	-23				$\begin{matrix} 7 & 4 & 1 & 5 & 1 & 6 & 2 & 11 \\ \text{кол} & \text{с} & \text{оп} & \text{00 000} & \text{1 000 000} & \text{02} & \text{02} & \end{matrix}$	
	Разрешить - запретить работу машины в системе	-23				$\begin{matrix} 7 & 4 & 1 & 5 & 1 & 6 & 2 & 11 \\ \text{кол} & \text{с} & \text{оп} & \text{000 000} & \text{1 000 000} & \text{02} & \text{02} & \end{matrix}$	
	Идти по ключу	-35	-35 с 01к 02a2	МКЛУ	К → инд ¹ → 62	15	
	Установить индикаторы	-04 1р	-04 1г Инд 00	У	$\begin{matrix} 7 & 4 & 1 & 12 \\ \text{кол} & \text{с} & \text{оп} & \text{Индикаторы} & \text{свободные} & \end{matrix}$	15	
Баркировка окружения*	-04 13р						
Баркировка нормализаций*	-04 14р						
Нуль*	-04 15р						
Баркировка пристановки*	-04 16р						
Клота*	-04 17р						
Минск - 22М*	-04 18р						

Группа	Название команды	Код команды	Вид команды	Назначение ССК	Информация	Время в мксек	Примечание																											
					структура индексной ячейки Микс-22М* Последовательность тактов																													
Обработка код командной машины "Микс-22М"	Содержимое рабочего уровня.				<table border="1"> <tr> <th colspan="4">Вид зависимой информации</th> </tr> <tr> <td>0</td> <td>0</td> <td>001000</td> <td>0 + 007000</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 + 01000</td> <td>0 + 01000</td> </tr> <tr> <td>2</td> <td></td> <td colspan="2">Нулевые разряды</td> </tr> <tr> <td>3</td> <td>0</td> <td>0 + 02000</td> <td>0 + 00000</td> </tr> <tr> <td>4</td> <td>0</td> <td>00000</td> <td>0 + 00000</td> </tr> <tr> <td>5</td> <td>0</td> <td>00000</td> <td>0 + 0</td> </tr> </table>	Вид зависимой информации				0	0	001000	0 + 007000	1	0	0 + 01000	0 + 01000	2		Нулевые разряды		3	0	0 + 02000	0 + 00000	4	0	00000	0 + 00000	5	0	00000	0 + 0	<p>Операции с кодами +00; +04; +77; -04; -10; -14; -16; -22; -30; -32; -34; -70; -71; -74; -75; -76; Выполняется одинаково при состоянии 0' и 1' индикатора "Микс-22М"</p>
	Вид зависимой информации																																	
	0	0	001000	0 + 007000																														
	1	0	0 + 01000	0 + 01000																														
	2		Нулевые разряды																															
	3	0	0 + 02000	0 + 00000																														
	4	0	00000	0 + 00000																														
	5	0	00000	0 + 0																														
	Выполнение командной машины "Микс-22М"	Модифицировать индекс	-20	-20 L 0, 01, 02, 02	Микс-22	K → 01 → 02 → 2чт → 011 → 011 →	30-35																											
		Ити к подпрограмме	-31	-31 L 0, 01, 02, 02	Микс-22	K → 013 → 01 → 02 → 011 → 012 → 23П →	35																											
Ити по перепалнению		-33	-33 L 0, 01, 02, 02	Микс-22	K → 013 → 01 → 02 →	15																												
Ити по ключу (для М-22 и М-32)		-35 *	-35 L 0K 02, 02	Микс-22	Коп 1 L 230000 K 02 02			K-номер ключа (1,2,4,10,20,40,100) 2-позиция перебора (2-1-переход по включенному ключу, 2-0-по выключенному ключу)																										
Ити по несовпадению		-37	-37 L 0, 01, 02, 02		K → 013 → 01 → 02 → 2чт →	30		Выполняется как ИНЕС																										
а) 0 индикатора "Карта"					K → 013 → 01 → 02 → 2чт →	15		Выполняется как ЭКСТРАКОД																										
б) 1 индикатора "Карта"					K → 013 → 01 → 02 → 2чт → А → 23П	35																												
Сложить порядки		-72	-72 L 0, 01, 02, 02		K → 013 → 01 → 02 → 2чт → А → 23П	35																												
Вычесть порядки		-73	-73 L 0, 01, 02, 02		K → 013 → 01 → 02 → 2чт → А → 23П	35																												
Экстракоды		+01-02																																
	+03-00																																	
	-01-02																																	
	-03-05																																	
	-06-07																																	
	-15-17																																	
	-21																																	
	-23+27																																	
	-36																																	
	-40+42																																	
-50+57																																		
-60+67																																		
а) на 0-6 уровне					K → 013 → 01 → 02 → 011 → 2чт → 01000	30																												
б) на 7-13 уровне					K → 013 → 01 → 02 → 011 → 2чт → 011	20																												

ПРИЛОЖЕНИЕ 2

НАЗНАЧЕНИЕ, ФУНКЦИИ СИГНАЛОВ РАЗЪЕМОВ СВЯЗЕЙ

№ П/П	Название сигнала	Вид сигнала	Источник сигнала	Назначение сигнала	По какому сигналу вырабатывается	Примечание
1	Пуск 1 ВнУ	Импульс	Вычислитель	Анализирует способность ВнУ выполнить полученную команду	2ЧтИ4	Если ВнУ способно выполнить полученную команду, оно отвечает «ВнУ свободно» и устанавливает в «1» указатель «Занято». Если ВнУ работоспособно, но новую команду выполнить не может, оно отвечает «ВнУ занято»
2	Конец передачи	»	»	Является признаком окончания команды ВВИ или ИУВУ, а также обмена данным символом	УЗПИЗ— в команде ВВИ, ОП2И1 — в команде ИУВУ, ПИЗ3 — в тактах обмена информацией с ВнУ МК	Если ВнУ свободно, производит сброс регистра команд устройства. При выводе символа сбрасывает потенциал приема символа во ВнУ и вырабатывает сигнал обработки полученного символа
3	СИ	»	»	Сигнал синхронизации работы вычислителя и внешних устройств	Для МК — ГИ2, для БК — ГИ5 (через каждые 5 мксек)	По этому сигналу из ВнУ выдаются сигналы: «Сбой ВнУ», «Конец работы» (для МК и БК), «Запрос прерывания» (для МК), «Конец обмена» (для БК). ВнУ МК по СИ выдают запросы
4	Пуск 2 ВнУ	»	»	Является признаком обмена информацией	ПИ12	Во ВнУ сбрасывает запрос и анализирует режим работы ВнУ (ввод или вывод). В случае вывода устанавливает потенциал приема символа во ВнУ
5	Обмен	»	»	Является сигналом опроса регистра символа для выдачи информации в вычислитель	ПИ12	По этому сигналу ВнУ МК при вводе выдает символ вместе с контрольным разрядом (с задержкой относительно сигнала «Обмен» не более 0,6 мксек)
6	КМ (конец массива)	Импульс	Вычислитель	Признак окончания обмена	По одному из импульсов: ПИ15, ПИ21, ПИ25, У, 1 "	Символ, с которым получен сигнал КМ, на носитель информации не выводится. По сигналу КМ устройство прекращает запросы и заканчивает внутренний цикл, после чего выдает сигнал «Конец работы» (активные ВнУ по окончании ввода выдают сигнал «Запрос прерывания»)

№ п/п	Название сигнала	Вид сигнала	Источник сигнала	Назначение сигнала	По какому сигналу вырабатывается	Примечание
7	Общий сброс	»	»	Приводятся в исходное состояние внешние устройства	При нажатии на кнопку „СБРОС“	Прием сигнала во ВнУ запрещается только при автономном режиме устройства
8	Передать номер ВнУ	—	—	—	—	В ЭВМ «Минск-32» не используется
9	Номер ВнУ	Потенциал	Вычислитель	Выбирается необходимое ВнУ	Б2И1 — для команд ВВИ и ИУВУ, ГИЗ— перед обменом информацией	Сохраняется высоким на время связи с данным ВнУ
10	Сигналы по кодовым шинам символа (КШС ВнУ)	Импульс	Вычислитель или ВнУ	Передача символа или определителя	Определитель — по ОПИИЗилиБ2И6, символ при выводе— по ПИ25, символ при вводе — по сигналу „Обмен“	Относительно сигнала «Обмен» задержка информации не более 0,6 сек. Символ с КШС принимается в регистр символа ВнУ при обмене. Если ВнУ свободно, определитель принимается в регистр команд ВнУ. При вводе с ВнУ БК символ поступает одновременно с сигналом «Запрос имп.». При выводе на ВнУ БК символ выдается по сигналу «Запрос имп.»
11	Сбой Вч	»	Вычислитель	Сигнал о сбойной ситуации в вычислителе во время обмена с ВнУ	При сбое — по одному из импульсов, ПИ15, ПИ21, ПИ25, У„1“Т2ПБ	ВнУ по этому сигналу прекращает запросы, устанавливает в «1» указатель «Сбой», отрабатывает внутренний цикл. Символ, с которым получен сигнал «Сбой Вч», на носитель информации не выводится. После окончания внутреннего цикла сигнал «Сбой ВнУ» или «Конец работы» не выдается
12	Запрос прерывания	Импульс	ВнУ	Признак работы активных ВнУ на ввод	По синхроимпульсу вычислителя	Активное ВнУ выдает этот сигнал при желании получить разрешение на ввод массива (перед вводом каждого символа не выдается) и после окончания ввода массива. Вычислитель по этому сигналу устанавливает в «1» триггер причины выхода на 3-й уровень
13	ВнУ свободно	»	»	Признак того, что ВнУ способно выполнить полученную команду	„Пуск 1 ВнУ“ (задержка не более 0,6 мксек)	При этом ВнУ устанавливает в «1» указатель «Занято»
14	ВнУ занято	»	»	Признак неспособности ВнУ выполнить полученную команду	„Пуск 1 ВнУ“ (задержка не более 0,6 мксек)	ВнУ работоспособно, но выполняет предыдущую команду (или общее управление устройствами не способно выполнить полученную ранее и сейчас команды). При ответе «ВнУ занято» снова выполняется та же команда ВВП
15	Запрос ВнУ	Потенциал	»	Признак готовности ВнУ ввести или вывести символ	По синхроимпульсу вычислителя (задержка не более 3,5 мксек и не менее 1,0 мксек)	При наличии запроса — низкий потенциал. Снимается по сигналу «Пуск 2 ВнУ» или следующим за этим сигналом синхроимпульсом
16	Запрос группы	»	Коммутатор	Признак того, что хотя бы в одном ВнУ группы есть запрос	По запросам ВнУ	При наличии запроса — низкий потенциал. Снимается при снятии всех запросов ВнУ группы
17	Ввод	Импульс	ВнУ	Признак работы ВнУ на ввод	„Пуск 2 ВнУ“ (задержка не более 0,6 мксек)	
18	Вывод	Импульс	ВнУ	Признак работы ВнУ на вывод	„Пуск 2 ВнУ“ (задержка не более 0,6 мксек)	
19	Сбой ВнУ	»	»	Сигнал нарушения работы ВнУ	По синхроимпульсу вычислителя	Сигнал выдается после окончания внутреннего цикла ВнУ. При выдаче сигнала ставится в «1» указатель «Сбой» и сбрасывается в «0» указатель «Занято». По этому сигналу в вычислителе устанавливается в «1» триггер причины выхода на 2-й уровень

№ П/П	Название сигнала	Вид сигнала	Источник сигнала	Назначение сигнала	По какому сигналу вырабатывается	Примечание
20	Конец работы ВнУ	»	»	Сигнал нормального окончания работы ВнУ	По синхроимпульсу вычислителя	Выдается после окончания внутреннего цикла ВнУ. При выдаче сигнала сбрасывается в «0» указатель «Занято». По этому сигналу в вычислителе устанавливается в «1» триггер причины выхода на 4-й уровень
21	Указатель есть	»	»	Сигнал о том, что опрашиваемый указатель находится в „I“ состоянии	При поступлении определителя из вычислителя (задержка не более 0,6 мксек)	При опросе указателей «Сбой» и «Ввод» указатели сбрасываются в «0» (после ответа «Есть указатель»)
22	Менять границу	»	»	Признак работы с ВнУ, вводящими информацию массивами не менее, чем из n символов	„Пуск 2 ВнУ“ (задержка не более 0,6 мксек)	Выдается теми устройствами, которые при выполнении команды «Ввести массив» после сигнала «КМ» не могут сразу прекратить ввод (например, устройство ввода с перфокарт не может мгновенно остановить движение перфокарты). По этому сигналу может быть расширена область МОЗУ, разрешенная для обмена (после ее заполнения)
23	Номер устройства n	Потенциал	Вычислитель	Для выборки ВнУ внутри подгрупп	Б2И1 — в командах ВВИ и ИУВУ, ГИЗ — перед обменом информацией	Высокий при $m^* = n$
24	Номер ВнУ8	Потенциал	Вычислитель	Для выборки подгруппы	Б2И1 — в командах ВВИ и ИУВУ, ГИЗ — перед обменом информацией	Высокий, если $k^* = 1, 3$
25	Номер ВнУ 8	»	»	»	Б2И1 — в командах ВВИ и ИУВУ, ГИЗ — перед обменом информацией	Низкий, если $k^* = 1, 3$
26	Номер ВнУ 16	»	»	»	»	Высокий, если $k^* = 2, 3$
27	»	»	»	»	»	Низкий, если $k^* = 2, 3$
28	Запрос ВнУ8	»	Коммутатор	Для формирования номера ВнУ	При наличии запроса в соответствующей подгруппе	Высокий, если выставило запрос ВнУ подгруппы 1, 3
29	»	»	»	»	»	Низкий, если выставило запрос ВнУ подгруппы 1, 3
30	Запрос ВнУ16	»	»	»	»	Высокий, если выставило запрос ВнУ подгруппы 2, 3
31	»	»	»	»	»	Низкий, если выставило запрос ВнУ подгруппы 2, 3
32	Запрос ВнУ „ n^* “	Потенциал	Коммутатор	Для формирования номера ВнУ	При наличии запроса от ВнУ $m^* = n$ и отсутствии запросов от более приоритетных ВнУ	Высокий потенциал, если среди выставивших запросы ВнУ n является самым приоритетным
33	Есть ГК	»	»	»	Выдается постоянно „— 15 е“	В вычислителе используется как признак отсутствия коммутатора
34	Есть ГК	»	»	»	Выдается постоянно потенциал "земля"	В вычислителе используется как признак подключения коммутатора
35	Запрос импульсный	Импульс	ВнУ БК	Признак запроса на обмен символом с ВнУ БК	Выдается асинхронно	При вводе информации выдается одновременно с символом. При выводе информации вычислитель по этому сигналу выдает символ
36	Конец обмена	»	»	Признак окончания обмена информацией	По синхроимпульсу вычислителя	Выдается ВнУ БК после прекращения обмена информацией с вычислителем. Вычислитель по этому сигналу сбрасывает в «0» указатель «Занят БК» и производит запись УС в МОЗУ. В ЭВМ «Минск-32» не используется
37	Начало обмена	»	»	»	»	»

Примечание. В каждой группе может быть не более 32 ВнУ. ВнУ внутри группы можно условно разделить на подгруппы, по 8 устройств в каждой. Тогда номер ВнУ в группе можно представить как $N_8 = Km$, где K — номер подгруппы (0, 1, 2, 3), m — номер устройства в подгруппе (0 ÷ 7).

КОДИРОВАНИЕ ИНФОРМАЦИИ СОГЛАСНО ГОСТу 10859-64

№ п/п	Внутренний код машины	Наименование символов	Символы	Код перфокарт	Символы Пч	Символы М, УПДК, СУ—2
1	0000000	Ноль	0	0	0	0
2	0000001	Единица	1	1	1	1
3	0000010	Двойка	2	2	2	2
4	0000011	Тройка	3	3	3	3
5	0000100	Четверка	4	4	4	4
6	0000101	Пятерка	5	5	5	5
7	0000110	Шестерка	6	6	6	6
8	0000111	Семерка	7	7	7	7
9	0001000	Восьмерка	8	8	8	8
10	0001001	Девятка	9	9	9	9
11	0001010	Плюс	+	0-2-8	+	+
12	0001011	Минус	—	0-3-8	—	—
13	0001100	Дробная черта	/	0-4-8	/	/
14	0001101	Запятая	,	0-5-8	,	,
15	0001110	Точка	.	0-6-8	.	.
16	0001111	Пробел	—	0-7-8	Пробел	Пробел
17	0010000	Основание десятичной системы счисления	10	12	10	10
18	0010001	Стрелка вверх	↑	12-0-1	↑	↑
19	0010010	Круглая скобка, левая	(12-0-2	((
20	0010011	Круглая скобка, правая)	12-0-3))
21	0010100	Умножение	×	12-0-4	×	×
22	0010101	Равно	=	12-0-5	=	=
23	0010110	Точка с запятой	;	12-0-6	;	;
24	0010111	Квадратная скобка левая	[12-0-7	[[
25	0011000	Квадратная скобка правая]	12-0-8]]
26	0011001	Звездочка	*	12-0-9	*	*
27	0011010	Открывающая кавычка	‘	12-2-8	‘	‘
28	0011011	Закрывающая кавычка	,	12-3-8	,	,
29	0011100	Не равно	≠	12-4-8	≠	≠
30	0011101	Меньше	<	12-5-8	<	<
31	0011110	Больше	>	12-6-8	>	>
32	0011111	Двоеточие	:	12-7-8	:	:
33	0100000	Прописные буквы русского алфавита	А	11	А	А
34	0100001		Б	11-0-1	Б	А
35	0100010		В	11-0-2	В	В
36	0100011		Г	11-0-3	Г	Г
37	0100100		Д	11-0-4	Д	Д
38	0100101		Е	11-0-5	Е	Е
39	0100110		Ж	11-0-6	Ж	Ж
40	0100111		З	11-0-7	З	З
41	0101000		И	11-0-8	И	И
42	0101001		Й	11-0-9	Й	Й
43	0101010		К	11-2-8	К	К
44	0101011		Л	11-3-8	Л	Л
45	0101100		М	11-4-8	М	М
46	0101101		Н	11-5-8	Н	Н
47	0101110	О	11-6-8	О	О	
48	0101111	П	11-7-8	П	П	
49	0110000	Прописные буквы русского алфавита	Р	12-11-0	Р	Р
50	0110001		С	12-11-1	С	С
51	0110010		Т	12-11-2	Т	Т
52	0110011		У	12-11-3	У	У
53	0110100		Ф	12-11-4	Ф	Ф
54	0110101		Х	12-11-5	Х	Х
55	0110110		Ц	12-11-6	Ц	Ц
56	0110111		Ч	12-11-7	Ч	Ч
57	0111000		Ш	12-11-8	Ш	Ш
58	0111001		Щ	12-11-9	Щ	Щ
59	0111010		Ы	12-11-0-2-8	Ы	Ы
60	0111011		Ь	12-11-0-3-8	Ь	Ь
61	0111100		Э	12-11-0-4-8	Э	Э
62	0111101		Ю	12-11-0-5-8	Ю	Ю
63	0111110	Я	12-11-0-6-8	Я	Я	
64	1111111	Исправление или		12-11-0-3-7-8-9		

№ п/п	Внутренний код машины	Наименование символов	Символы	Код перфокарт	Символы Пч	Символы М, УПДК, СУ—2
		выделение				
65	0111111	Прописные буквы латинского алфавита	D	12-11-0-7-8	D	D
66	1000000		F	0-3-9	F	F
67	1000001		G	1-3-9	G	G
68	1000010		I	2-3-9	I	I
69	1000011		I	0-1-2-3-9	I	I
70	1000100		L	3-4-9	L	L
71	1000101		N	3-5-9	N	N
72	1000110		Q	3-6-9	Q	Q
73	1000111		R	3-7-9	R	R
74	1001000		S	3-8-9	S	S
75	1001001		U	0-1-3-8-9	U	U
76	1001010		V	0-2-3-8-9 -	V	V
77	1001011		W	1-2-3-8-9 0	W	W
78	1001100		Z	3-4-8-9	Z	Z
79	1001101	Надчеркивание	—	0-3-5-8-9		—
80	1001110	Меньше равно	<	0-3-6-8-9		<
81	1001111	Больше равно	>	0-3-7-8-9		>
82	1010000	Логическое „или“	∨	12-3-9		∨
83	1010001	Логическое „и“	∧	12-0-1-3-9		∧
84	1010010	Импликация	⊃	12-0-2-3-9		⊃
85	1010011	Логическое „нет“	-	12-1-2-3-9		-
86	1010100	Знак промежутка	+	12-0-3-4-9		+
87	1010101	Тождество	≡	12-0-3-5-9		≡
88	1010110	Процент	%	12-0-3-6-9		%
89	1010111	Ромбик	◇	12-0-3-7-9		◇
90	1011000	Вертикальная черта		12-0-3-8-9		
91	1011001	Горизонтальная черта	—	12-1-3-8-9		—
92	1011010	Подчеркивание	—	12-2-3-8-9		—
93	1011011	Восклицательный знак	!	12-0-1-2-3-8		!
94	1011100	Кавычки прямые	„	12-3-4-8-9		„
95	1011101	Твердый знак	Ъ	12-3-5-8-9		Ъ
96	1011110	Градус	°	12-3-6-8-9		°
97	1011111	Апостроф	'	12-3-7-8-9		'
98	1100000	Стрелка влево	→	11-3-9		→
99	1100001	Стрелка вправо	←	11-0-1-3-9		←
100	1100010	Вопросительный знак	?	11-0-2-3-9		?
101	1100011	Стрелка вниз	↓	11-1-2-3-9		↓
102	1100100	Знак диаметра	∅			∅
103	1100101	Плюс минус	±	11-0-3-4-9		±
104	1100110	Знак чистоты обработки	∇	11-0-3-5-9		∇
105	1100111			11-0-3-6-9		
106	1101000			11-0-3-7-9		
107	1101001			11-0-3-8-9		
108	1101010			11-1-3-8-9		
109	1101011			11-2-3-8-9		
110	1101100			11-0-1-2-3-8-9		
111	1101101			11-3-4-8-9		
112	1101110			11-3-5-8-9		
113	1101111			11-3-6-8-9		
114	1110000			11-3-7-8-9		
115	1110001			12-11-0-3-9		
116	1110001			12-11-1-3-9		
117	1110010			12-11-2-3-9		
118	1110011			12-11-0-1-2-3-9		
119	1110100			12-11-3-4-9		
120	1110101			12-11-3-5-9		
121	1110110			12-11-3-6-9		
122	1110111			12-11-3-7-9		
123	1111000			12-11-3-8-9		
124	1111001			12-11-0-1-3-8-9		
125	1111010			12-11-0-2-3-8-9		Черный цвет
126	1111011			12-11-1-2-3-8-9		Красный цвет
127	1111100			12-11-0-3-4-8-9		Возврат каретки
128	1111101			12-11-0-3-5-8-9		Перевод строки
128	1111110			12-11-0-3-6-8-9		

Пояснение к таблице

1. С перфокарт и 8-дорожковой перфоленты можно вводить в МОЗУ и выводить на них из МОЗУ все 128 символов, но отперфорировать на этих носителях с помощью УПДК и КСУ-2 можно только символы, указанные в соответствующей графе таблицы. Это связано с тем, что в качестве клавиатуры КСУ-2 и УПДК используется клавиатура пишущей машинки с ограниченным набором символов.

2. В обмене между МОЗУ и магнитной лентой участвуют все 128 символов.

3. При вводе перфокарт в коде ГОСТа 10859-64 пустая колонка (нет пробивок) преобразуется в код 1111110.

4. При выводе информации в коде ГОСТа 10859-64 на перфокарту символ с кодом 1111110 соответствует пустой колонке.

1. «Расчет экономической эффективности внедрения ЭВМ». Перевод с японского. М., «Прогресс», 1968.
2. Захаров В. М. Оптимизация структуры вычислительной системы. Труды I Всесоюзной конференции по вычислительным системам. Вып. 1, Новосибирск, 1968.
3. Knight K. E. Changes in computer performance. *Datamation*, 1966, Sypt, p. 40.
4. Knight K. E. Evolving computer performance 1963—1967. *Datamation*, 1968, Jan. p. 31.
5. Глушков В. М. Два универсальных критерия эффективности вычислительных машин. ДАН УССР, 1960, № 1.
6. Gosch H. K. The Future of computing — *Indus and Eng. Chemistry*. 1958, vol. 50, No 11.
7. Kittle H., Mertens P. Einige quantitative Untersuchungen zur Crossendegression von Datenverarbeitung an lagen. — *Electron Datenverarb.* 1965. Bd. 7, N 6.
8. Карцев М. А. Арифметика цифровых машин. М., «Наука», 1969.
9. Анисимов Б. Е., Четвериков В. И. Основы теории и проектирования цифровых вычислительных машин, М. «Машиностроение», 1965.
10. Вычислительная система IBM-360 под редакцией Штаркмана В. С. М., «Советское радио». 1969.
11. Klahn R., Shively R. FFT — Shortcut to Fburiar analysis. *Electronics*, No. 8, V41. 1968, p. 124—129.
12. Klahn R., Shively R., Gomez E, The time — saver FET hardware. *Electronics*, No. 13, Y41, 1968, p. 92—97.
13. Система ускоренной цифровой фильтрации. — «Электроника», 1969, № 4, стр. 66.
14. Ричардс Р. К. Арифметические операции в цифровых вычислительных машинах. ИЛ, 1957.
15. Уршова Г. И. Аппаратурный контроль и надежность специализированных ЭВМ. «Советское радио». 1969.
16. Канада Х. Система прерываний электронной вычислительной машины «ПЕАС-2203». Материалы объединенного конгресса четырех электротехнических обществ, состоявшегося в 1960 г., № 375.
17. Флеров А. Б. Вопросы оценки производительности универсальных Цифровых вычислительных машин. Сб. «Об использовании вычислительных средств для обработки учетно-плановой информации». Минск. Труды НИИЭМП, 1968.
18. Найт К. Е. Оценка развития рабочих характеристик ЦВМ. — «Зарубежная радиоэлектроника», № 11, «Советское радио», 1968.
19. Пыхтин В. Я., Мальцева В. А. Защита программ от взаимного влияния в мультипрограммной ЭВМ. Сб. «Вопросы радиоэлектроники». Сер. ВТ. Вып. 5, 1969.
20. Пыхтин В. Я., Запольский А. П., Смирнов Г. Д. Способ подключения внешних устройств с частотой запросов, соизмеримой с циклом оперативной памяти. Сб. «Вопросы радиоэлектроники». Серия ЭВТ. Вып. 5, 1969.
21. Евреинов Э. В., Косарев Ю. Г. Однородные универсальные вычислительные системы высокой производительности. «Наука», 1966.
22. Голубев-Новожилов Ю. С. Многомашинные комплексы вычислительных средств. «Советское радио», 1967.
23. Пыхтин В. Я., Лопато Г. П., Василевский А. В., Сидристый Б. А., Хорошевский В. Г. Системное устройство элементарной машины вычислительной системы «Минск-222». В кн.: Вычислительные системы Вып. 23. Новосибирск, «Наука», 1966.
24. Пыхтин В. Я., Косарев Ю. Г., Жуков Е. И., Головяшкина Л. В., Колосова Ю. И. Особенности употребления команд системы «Минск-222». В кн.: Вычислительные системы. Вып. 24. Новосибирск, «Наука», 1967.
25. Вычислительные системы. Сборник трудов ИМ СО АН СССР. Вып. 23. Новосибирск, 1966.
26. Пыхтин В. Я., Лопато Г. П., Смирнов Г. Д., Василевский А. Н. Анализ системы однородных вычислительных машин «Минск-222». Материалы республиканской научно-технической конференции. «Применение математических методов в вычислительной технике в народном хозяйстве». Минск, 1969.
27. Пыхтин В. Я., Прижялковский В. В., Смирнов Г. Д., Запольский А. П. Организация многомашинных комплексов на базе ЭВМ типа «Минск». Материалы республиканской научно-технической конференции. «Применение математических методов и вычислительной техники в народном хозяйстве». Минск, 1968.
28. Pattersn G. V. Understanding compatability in computing systems. «Control Engineering», V13, No. 7, 1966.
29. The feature of compatability 1401 in system 360/30 Communications of the ACM, V8, No. 12, December, 1965, p. 773—776.
30. Emulator Spectra 70/45 for RCA-301 Communicatins of the ACM. 8, No. 12, December 1966.
31. Пыхтин В. Я., Мальцева В. А., Смирнов Г. Д. Аппаратура совместимости «Минск-32» с машиной «Минск-22М». Сб. «Вопросы радиоэлектроники». Сер. ЭВТ. Вып. 5, 1969.
32. Кушнерев Н. Т., Неменман М. Е., Рубан Э. П. Программная система совместимости машины «Минск-32» с машиной «Минск-22М». Сб. «Вопросы радиоэлектроники». Сер. ЭВТ. Вып. 5, 1969.
33. Матлина Р. Х., Пыхтин В. Я., Пшоник А. Г. Контроль сдвига массивов информации в каналах ЭВМ. Сб. «Вопросы радиоэлектроники» Сер. ВТ. Вып. 8, 1970.
34. Марголин М. С., Надененко В. К., Смирнов Г. Д. Электронная вычислительная машина «Минск-22». Минск, «Высшая школа», 1969 152 с.
35. Запольский А. П., Лопато Г. П., Прижялковский В. В., Пыхтин В. Я., Смирнов Г. Д. Организация системы вычислительных машин на базе ЭВМ «Минск-32». Сб. «Вопросы радиоэлектроники». Сер. ВТ. Вып. 8, 1970.
36. Кушнерев Н. Т., Рубан Э. П. Моделирование на ЭВМ «Минск-32» обмена ЭВМ «Минск-22» с магнитной лентой адресной структуры. Сб. «Вопросы радиоэлектроники». Сер. ВТ. Вып. 8, 1970.