

Для служебного пользования

Экз. \_\_\_\_\_

ПРЕДПРИЯТИЕ П/Я А-ЗІ62

"УТВЕРЖДАЮ"

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА

БЭСМ-10

ЭСКИЗНЫЙ ПРОЕКТ

ЧАСТЬ I. АППАРАТУРА

-1973-

Ответственные исполнители

Зав.лабораторией № I	<i>Мельников</i>	(В.А.Мельников)
Зав.лабораторией № 5	<i>Королев</i>	(Л.Н.Королев)
Зав.лабораторией № I-18	<i>Рябов</i>	(Г.Г.Рябов)
Зав.лабораторией № I-I	<i>Соколов</i>	(А.А.Соколов)
Зав.лабораторией № I-2	<i>Титкин</i>	(М.В.Титкин)
Зав.лабораторией № I-3	<i>Смирнов</i>	(В.И.Смирнов)
Начальник сектора	<i>Зак</i>	(Л.А.Зак)
Начальник сектора	<i>Иванников</i>	(В.П.Иванников)
Начальник сектора	<i>Томилин</i>	(А.Н.Томилин)
Начальник сектора	<i>Шербаков</i>	(О.К.Шербаков)

Исполнители

От предприятия п/я А-3162		
лаборатории № I, I-I, I-2, I-3		
Ст.научн.сотрудник	<i>Митропольский</i>	(Ю.И.Митропольский)
Ведущий конструктор	<i>Знаменский</i>	(Ю.Н.Знаменский)
Ведущий конструктор	<i>Чехлов</i>	(В.С.Чехлов)
Ведущий конструктор	<i>Ли</i>	(В.Л.Ли)
Ст.научн.сотрудник	<i>Жирков</i>	(В.Ф.Жирков)
Ведущий конструктор	<i>Жуковский</i>	(В.А.Жуковский)
Ведущий конструктор	<i>Большаков</i>	(О.А.Большаков)
Ведущий конструктор	<i>Новизенцев</i>	(В.С.Новизенцев)
Ведущий конструктор	<i>Чайковский</i>	(Л.Ф.Чайковский)
Ведущий конструктор	<i>Головина</i>	(М.А.Головина)
Ведущий конструктор	<i>Аваев</i>	(А.В.Аваев)
Ведущий конструктор	<i>Визун</i>	(И.Д.Визун)
Ведущий конструктор	<i>Зубрилин</i>	(Н.П.Зубрилин)
Ст.научн.сотрудник	<i>Кузьмичев</i>	(Д.Л.Кузьмичев)
Ведущий конструктор	<i>Лавренченко</i>	(В.Е.Лавренченко)
Ведущий инженер	<i>Ульянов</i>	(В.В.Ульянов)
Ведущий инженер	<i>Калиниченко</i>	(Е.П.Калиниченко)
Ведущий инженер	<i>Матисова</i>	(Л.Г.Матисова)
Старший инженер	<i>Горюшев</i>	(Л.С.Горюшев)

Инженер	<i>В.Дубинин</i>	(В.А.Отто)
Старший инженер	<i>М.Сургудзе</i>	(М.Ш.Сургуладзе)
Техник	<i>Богомолов</i>	(Н.А.Богомолов)
Техник	<i>Калитина</i>	(Н.С.Калитина)
Ведущий инженер	<i>М.Терехин</i>	(М.А.Терехин)
Ведущий инженер	<i>Е.Ильинков</i>	(Е.И.Ильинков)
Ведущий инженер	<i>Б.Барабашкин</i>	(В.М.Барабашкин)
Старший инженер	<i>Г.Шабашов</i>	(Г.А.Шабашов)
Старший инженер	<i>М.Феклисов</i>	(М.Е.Феклисов)
Старший инженер	<i>Ю.Гаврин</i>	(Ю.С.Гаврин)
Старший инженер	<i>Л.В.Сидорова</i>	(Л.В.Сидорова)
Старший инженер	<i>Л.К.Крупская</i>	(Л.К.Крупская)
Старший инженер	<i>В.С.Зольникова</i>	(В.С.Зольникова)
Конструктор II к.	<i>Н.Ф.Абасов</i>	(Н.Ф.Абасов)
Старший инженер	<i>В.П.Сиротин</i>	(В.П.Сиротин)
Инженер	<i>В.Т.Дугельная</i>	(В.Т.Дугельная)
Инженер	<i>А.Н.Ершова</i>	(А.Н.Ершова)
Инженер	<i>О.Н.Иванишина</i>	(О.Н.Иванишина)
Инженер	<i>О.П.Лащенко</i>	(О.П.Лащенко)
Старший инженер	<i>Б.Н.Вишневецкая</i>	(Б.Н.Вишневецкая)
Конструктор III к.	<i>Л.К.Хайлова</i>	(Л.К.Хайлова)
Конструктор III к.	<i>Л.А.Самородская</i>	(Л.А.Самородская)
Конструктор II к.	<i>В.В.Солищев</i>	(В.В.Солищев)
Конструктор II к.	<i>В.Р.Холмаков</i>	(В.Р.Холмаков)
Техник	<i>Л.Н.Ананьев</i>	(Л.Н.Ананьев)
Инженер	<i>Н.П.Бурмистров</i>	(Н.П.Бурмистров)
Ведущий инженер	<i>А.А.Гостева</i>	(А.А.Гостева)
Ведущий инженер	<i>А.П.Дудушкин</i>	(А.П.Дудушкин)
Ведущий инженер	<i>В.В.Похлебкин</i>	(В.В.Похлебкин)
Конструктор III к.	<i>Н.В.Коржинский</i>	(Н.В.Коржинский)
Инженер	<i>А.В.Егоров</i>	(А.В.Егоров)
Инженер	<i>Н.Д.Соломатина</i>	(Н.Д.Соломатина)
Инженер	<i>В.С.Морозова</i>	(В.С.Морозова)
Старший инженер	<i>В.Г.Дробышев</i>	(В.Г.Дробышев)
Конструктор III к.	<i>Н.К.Власова</i>	(Н.К.Власова)

Лаборатория № 5

Ведущий инженер *М.Г.Чайковский* (М.Г.Чайковский)  
Ведущий инженер *В.И.Максаков* (В.И.Максаков)

Лаборатория № 18

Ведущий конструктор *Л.А.Ричков* (Л.А.Ричков)  
Ведущий инженер *В.Е.Вулихман* (В.Е.Вулихман)  
Ведущий инженер *Л.А.Коростелева* (Л.А.Коростелева)  
Старший инженер *Е.П.Дербакова* (Е.П.Дербакова)  
Инженер *Л.А.Блинова* (Л.А.Блинова)  
Инженер *Г.Г.Василькова* (Г.Г.Василькова)  
Инженер *Н.П.Дроздова* (Н.П.Дроздова)  
Инженер *М.С.Константинов* (М.С.Константинов)  
Инженер *Ю.С.Коротаев* (Ю.С.Коротаев)  
Инженер *Л.М.Эльберт* (Л.М.Эльберт)

Конструкторское бюро

Начальник КБ-2 *Г.И.Шлосберг* (Г.И.Шлосберг)  
Ведущий инженер *Г.А.Градус* (Г.А.Градус)  
Начальник патентного *В.В.Колик* отдела (В.В.Колик)

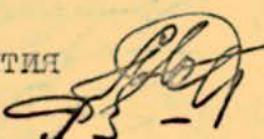
ИТЭФ Комитета по использованию  
атомной энергии СССР

Зав.лабораторией *М.Н.Богомолов* (М.Н.Богомолов)  
Начальник службы ЭВМ *В.Е.Чусов* (В.Е.Чусов)  
Ведущий инженер *Е.С.Ржевский* (Е.С.Ржевский)

ИПМ АН СССР

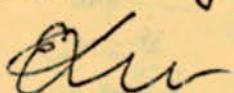
Старший инженер *С.С.Андреев* (С.С.Андреев)

От предприятия п/я Г-4085

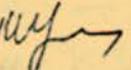
Руководитель предприятия  (Г.П. Морозов)

Гл. инженер  (Р.А. Роздова)

Начальник отдела  (А.П. Едовиченко)

Вед. инженер  (Е.И. Хакалев)

От предприятия п/я Х-5594

Начальник лаборатории  (М.Б. Цибульников)

Вед. инженер  (И. Сенников)

Московский завод "САМ"

Гл.инженер завода		(В.Л.Иванов)
Гл.инженер СКБ		(В.В.Прошаков)
Нач.отдела СКБ		(Л.А.Теплиашкий)
Нач.лаборатории СКБ		(А.А.Усан)
Вед.инженер ОИМ		(Е.И.Бородин)
Вед.инженер СКБ		(Е.А.Сегаль)
Вед.инженер ОИМ		(В.В.Голубев)
Вед.инженер ОИМ		(И.И.Бабитский)
Вед.инженер СКБ		(П.М.Моисеев)
Ст.инженер СКБ		(В.З.Шнитман)
Вед.инженер ОИМ		(А.П.Павлова)
Вед.инженер ОИМ		(Г.С.Пономарева)
Ст.инженер СКБ		(Л.М.Дорофеева)
Вед.инженер СКБ		(С.Л.Смирнов)
Вед.инженер СКБ		(О.Н.Сердокова)
Ст.инженер СКБ		(Л.Н.Иванова)
Ст.инженер СКБ		(В.Б.Боев)
Ст.инженер ОИМ		(В.А.Баканова)
Ст.инженер ОИМ		(И.В.Волкова)
Вед.инженер СКБ		(А.В.Князева)
Вед.инженер СКБ		(Ю.Г.Мовшович)
Вед.инженер СКБ		(А.Ф.Завельский)
Инженер СКБ		(А.П.Теребененков)

ОГЛАВЛЕНИЕ

	стр.
ВВЕДЕНИЕ	II
ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР И ОПЕРАТИВНАЯ ПАМЯТЬ	21
Введение	21
Организация памяти и работы процессоров	—
в многопроцессорной системе	23
Управление оперативной памятью	37
Устройство управления ЦП БЭСМ-10	64
Выполнение основных типов команд	—
в УУ БЭСМ-10	66
Организация обращения к СОЗУ	72
Арифметическое устройство	78
Структура арифметического устройства	81
Буфер арифметических команд	85
Блок буферных регистров чисел	91
Регистры первого и второго операндов	93
Блок выполнения логических операций	94
Блок выполнения операций сложения и вычитания	104
Блок выполнения операций умножения	116
Блок выполнения операций деления	129
Блок выполнения операций над последовательностями	132
Обеспечение совместимости с БЭСМ-6	134

стр.

Оценка производительности многопроцессорной системы	I40
КОММУТАТОР КАНАЛА I-ГО УРОВНЯ	I48
ПЕРИФЕРИЙНАЯ СИСТЕМА	I56
Введение	I56
Структура периферийной машины ПМ-IO	I58
Узел обмена	I60
Буфер директив	I69
Загрузка информации в служебные регистры ПМ-IO	I70
Загрузка директив в буфер	I72
Разгрузка очереди директив	I74
Периферийный процессор	I75
Основные принципы реализации	I75
Структура процессора	I79
Блок регистров	I82
Блок обработки	I86
Схема формирования операндов и результата операции	I89
Нанооперации считывания	I91
Нанооперации записи	I91
Операции обработки	I92
Выполнение команд процессора	201
Каналлер	208
Сопряжение с каналом 2-го уровня	213
Управление МОЗУ	218
Адаптер канала I-го уровня	221
Коммутатор канала 2-го уровня БЭСМ-IO	226

	стр.
Адаптер преобразования сопряжения	239
Подключение высокоскоростных внешних запоминающих устройств	242
Средства сопряжения с каналами связи	248
Телеграфный мультиплексор	251
Функции мультиплексора	251
Принцип построения	256
Система приказов	261
Структура мультиплексора	266
Телефонный мультиплексор	269
Функции мультиплексора	269
Принцип реализации	272
Универсальный стык с адаптерами	280
ЭЛЕМЕНТНАЯ И КОНСТРУКТИВНАЯ БАЗА БЭСМ-10	288
Введение	288
Программируемая интегральная диодная матрица	288
Диодная матрица с предварительной настройкой	293
Электрическая схема основного усиительного элемента	294
Управляющие цепи в ДЛТУ-схемах	301
Регистровые запоминающие элементы	305
Основные микросхемы ДЛТУ-схем	308
Основные типы связей между элементами	309
Полуавтомат для настройки и контроля диодных матриц	315
Система электропитания	320
Конструкция	340

СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ АППА-  
РАТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ БЭСМ-10

Назначение системы	350
Использование формул	351
Базовая система программирования	352
Организация внешней памяти	353
Архив функциональных элементов	355
Архив формул	356
Словарь названий сигналов	357
Размещение элементов	358
Алгоритм А1	362
Алгоритм А2	366
Алгоритм А3	368
Итеративный алгоритм (А4)	372
Разнесение соединений по слоям МПП	374
Программа трассировки	377
АНАЛИЗ ПАТЕНТНОЙ СИТУАЦИИ	382

## В В Е Д Е Н И Е

Предприятием п/я А-ЗІ62 были сделаны предложения о разработке для нужд народного хозяйства и обороны страны вычислительной системы БЭСМ-ІО, которая является продолжением линии универсальных ЭВМ типа БЭСМ. (БЭСМ - М-20 - БЭСМ-6 - АС-6 - БЭСМ-ІО).

Обоснование необходимости разработки новой мощной вычислительной системы, определение ее главных параметров, выбор основных направлений проектирования содержатся в аванпроекте вычислительной системы БЭСМ-ІО, выпущенном в марте 1973 года.

Сравнительно небольшой промежуток времени между сроками представления аванпроекта и эскизного проекта, естественно, привел к определенным трудностям. Однако эти обстоятельства были учтены еще при создании аванпроекта, в котором наряду с техническими предложениями и обоснованием выбора направлений проектирования целый ряд вопросов представлен с детализацией и глубиной проработки, отвечающей требованиям к эскизному проектированию. Поскольку основные характеристики системы и ее отдельных модулей были подробно изложены в аванпроекте, темой эскизного проекта явилось уточнение некоторых параметров системы, проверка ряда основных положений аванпроекта и техническое проектирование конкретных узлов и устройств БЭСМ-ІО.

Распределение материала, направленность и стиль его изложения в значительной степени определились характером проектирования - детальной разработкой отдельных, функционально самостоятельных устройств системы, и в результате материалы эскизного проектирования отличает большая, по сравнению с аванпроектом, автономность по разделам.

Структурная схема системы БЭСМ-10, описанная в аванпроекте, явилась базой для эскизного проектирования отдельных устройств, каналов связи и определения способов обмена информацией между устройствами. В соответствии с этой структурной схемой БЭСМ-10 представляет собой многопроцессорную вычислительную систему с числом процессоров для типовой структуры равным четырем.

Расширение системы и изменение ее структуры возможно за счет включения дополнительных модулей, объединяемых каналом I-го уровня.

Средняя производительность каждого процессора - 10-15 млн. операций в секунду. Эти процессоры имеют общую оперативную память объемом 512-1024 тыс. слов и общую сверхоперативную память емкостью в 4096 слов. Структура БЭСМ-10 ориентирована в первую очередь на решение больших научных задач методом разбиения их на сильно связанные параллельные процессы. В этом случае структура с общей сверхоперативной памятью оказывается значительно более эффективной, чем вычислительный комплекс с изолированным индивидуальным блоком сверхоперативной памяти при каждом процессоре или многомашинная система, которые обычно используются в слу-

чае ориентации на слабо связанные между собой задачи и процессы. Таким образом, ориентация на определенный класс задач предопределила структуру БЭСМ-10 – многопроцессорную систему, в которой не только оперативная, но и сверхоперативная память является общей и равнодоступной для каждого процессора. В БЭСМ-10 многопроцессорный принцип достиг своего наиболее законченного выражения. В результате на ряде важных задач математической физики производительность каждого процессора БЭСМ-10 достигает 20 млн. операций в сек.

Указанная особенность совершенно не накладывает ограничений на использование БЭСМ-10 в других областях, так как сверхоперативная память полностью скрыта от программ и является лишь своеобразным универсальным ускорителем оперативной памяти. Для повышения эффективности работы системы при решении задач, составленных на языках высокого уровня, в процессорах аппаратно реализованы многие важные функции, например, обращение к процедурам, механизм передачи параметров, защита одним программ и процессов от других и пр. При проектировании таких систем возникает ряд специфических проблем, которые были решены в процессе эскизного проектирования. Были определены способы синхронизации процессоров, разработаны блоки сверхоперативной памяти (СОЗУ) и устройство коммутации, объединяющее между собой модули процессоров ОЗУ и СОЗУ. Принятые решения обеспечивают высокую пропускную способность сверхоперативной памяти, согласованную с тактовой частотой процессоров в расчете на их

работу.

Одновременно было проведено уточнение алгоритмов выполнения команд в арифметическом устройстве процессора БЭСМ-10 и найдены значительно более быстрые способы их выполнения, по сравнению с предполагавшимися ранее. Существенно удалось ускорить обработку команд в устройстве управления. В результате эскизного проектирования отдельных, наиболее важных узлов центрального процессора уточнен объем и состав оборудования, необходимый для построения устройства управления и арифметического устройства, определены конструкция и габариты ячеек, плат и устройств БЭСМ-10 и найдены наиболее эффективные способы взаимного расположения основных модулей многопроцессорной системы.

Разрабатывается система автоматического проектирования для БЭСМ-10 - АПАС-10, являющаяся непосредственным развитием действующей в настоящее время системы АПАС, которая легла в основу технического проектирования и производства системы АС-6.

Достигнутые результаты в проектировании логических устройств базируются на ряде принципиальных особенностей системы элементов БЭСМ-10. Высокие электрические, логические и конструктивные свойства элементов, а также возможность использования опыта разработки и готовых схемных решений БЭСМ-6 и АС-6 значительно ускоряют процесс технического проектирования модулей БЭСМ-10. В этом отношении, как и предполагалось, большую роль играет сохранение схемотех-

нических и логических особенностей систем элементов БЭСМ-6 и программируемость диодных логических матриц, благодаря которой стала возможной простая аппаратная реализация практически всех необходимых логических преобразований.

Уточнение параметров системы элементов на основании результатов, полученных в результате разработки гибридных усилительных схем, показали, что элементы БЭСМ-10 будут обладать значительно более высокими параметрами по сравнению с теми, которые приведены в аванпроекте. Так, например, типовое время задержки логического элемента с усилителем в процессе эскизного проектирования удалось уменьшить почти вдвое по сравнению с приведенным в аванпроекте, а среднюю выделяемую мощность уменьшить примерно в полтора раза. Была разработана экономичная схема усилителя для реализации буферов и регистров, которая находит самое широкое применение в устройствах БЭСМ-10. Одновременно велись работы по освоению системы элементов БЭСМ-10 на предприятиях-изготовителях, которые успешно заканчиваются.

В силу целого ряда существенных особенностей элементной базы внедрение ее на предприятиях МЭП"а не пересекается ни с какими аналогичными работами, и это создает благоприятную обстановку для размещения заказов, необходимых для создания опытного образца БЭСМ-10.

Основным требованием при выборе электрических схем и режимов элементов, конструировании разъемов, ячеек и устройств системы являлась необходимость получения макси-

мального срока службы элементов и максимальной надежности узлов и системы в целом. В связи с этим разработка ориентировалась лишь на освоенные и, главное, хорошо зарекомендовавшие себя с этой точки зрения, изделия. Например, в усилиельных схемах была использована широко освоенная гибридная тонкопленочная технология, которая в настоящее время обеспечивает максимальную надежность микроэлементов при длительной эксплуатации. Электронные схемы элементов БЭСМ-10 обладают повышенной помехоустойчивостью. Помимо этого, учитывая резкое снижение срока службы полупроводниковых приборов при повышении их рабочей температуры, оказалось необходимым принять меры для облегчения электрических и температурных режимов компонентов машины, например, степень интеграции элементов с этой целью была выбрана ниже максимально возможной, предусмотрены простые и эффективные способы охлаждения. Конструкция опытного образца БЭСМ-10 разрабатывается в первую очередь в расчете на скорейшее внедрение его в производство и расчитана по степени механизации производства на уровень, близкий к уровню производства ЭВМ БЭСМ-6. Более того, может быть успешно применен целый ряд конкретных конструктивных элементов и технологических приемов, используемых в настоящее время при производстве БЭСМ-6.

В дальнейшем предполагается существенное повышение уровня механизации производства БЭСМ-10 путем его постепенной рационализации на заводе-изготовителе.

В настоящее время уже изготовлен и частично введен в эксплуатацию структурный и программный прототип БЭСМ-10 - система АС-6, на котором ведутся работы по отладке математического обеспечения БЭСМ-10. В дальнейшем предполагается производить оснащение вычислительных центров устройствами БЭСМ-10 на базе функционирующих систем АС-6 путем их развития и постепенной замены старых устройств. Подобный способ поэтапного ввода в строй системы БЭСМ-10 позволит в несколько раз ускорить ее наладку и корректировку документации. Полная аппаратная и программная совместимость этих систем дает возможность использовать все средства, предоставляемые операционными системами и системами программирования БЭСМ-6 и АС-6 для проведения работ по вводу в действие БЭСМ-10 - систему трансляторов, полный набор действующих внешних запоминающих устройств и устройств ввода-вывода. Окажется возможным создание целого ряда программно-совместимых промежуточных и смешанных вариантов систем БЭСМ-10 - АС-6, например, центральный процессор БЭСМ-10 в комплексе с оперативной памятью АС-6, подключаемой при помощи канала I-го уровня, оперативная и сверхоперативная память БЭСМ-10 с процессором АС-6, типовая многопроцессорная система БЭСМ-10, обслуживаемая комплексом периферийных устройств АС-6 или БЭСМ-6, и пр.

Аналогично, предполагается произвести постепенный ввод и отладку математического обеспечения системы и здесь

очень важным окажется програмная совместимость БЭСМ-10 с БЭСМ-6, так как еще до окончательной отладки математического обеспечения системы БЭСМ-10, она будет широко использоваться для решения программ в кодах БЭСМ-6 с полным сохранением имеющегося для этой ЭВМ программного задела.

При разработке системы математического обеспечения для АС-6, которая ведется в настоящее время, с самого начала учитывалось то, что это математическое обеспечение без изменений должно быть использовано на системе БЭСМ-10. Главным принципом, положенным в основу многофункциональной операционной системы для АС-6 и БЭСМ-10, является принцип независимости организации параллельной обработки от реального числа одновременно работающих физических процессоров, модулей и устройств системы.

В операционной системе функции распределения виртуальных (математических) ресурсов по физическим исполнителям выделены в автономные динамически настраиваемые блоки. Тем самым возможно изменение конфигурации системы в процессе счета без изменений в функционировании всех остальных блоков операционной системы. Иными словами, большинство своих функций операционная система выполняет, управляя виртуальными ресурсами — математической сегментированной памятью, процессами, семафорами, событиями, файлами и т.п. Таким образом, увеличение физических ресурсов БЭСМ-10 по сравнению с АС-6 не потребует изменения операционной системы. Наличие сверхоперативной памяти скрыто от операционной

системы и не отразится на ее функционировании. Подобная организация операционной системы значительно повышает живучесть системы, обеспечивая при этом максимально возможный физический параллелизм в выполнении больших и сложных программ.

Многофункциональность операционной системы, состоящая в совмещении в одной системе режимов пакетной обработки, дистанционной пакетной обработки, коллективного пользования и режима реального времени, достигается за счет различных стратегий назначения процессам приоритетов, ресурсов и квантов времени обслуживания с учетом режима их выполнения.

На этапе эскизного проектирования была выполнена работа по уточнению структуры операционной системы и языков программирования. Разработаны инструментальные языки — язык " " и макроассемблер, разработаны схемы оптимизации распределения памяти и регистров. Решены узловые вопросы реализации рекурсий, обращения к процедурам, передачи параметров, представления структуры загрузочного модуля.

Разработана в деталях структура общей информационной базы системы БЭСМ-10 — архива, или банка данных.

Проработаны вопросы выполнения программ БЭСМ-6 на аппаратуре БЭСМ-10, вопросы преемственности программ на уровне использования возможностей операционных систем. В операционную систему БЭСМ-10 включены средства, позволяющие эффективно интерпретировать языковые конструкции БЭСМ-6 для системы ДИСПАК и НД-70.

К моменту изготовления опытного образца БЭСМ-10 для него будет готово отлаженное математическое обеспечение и накоплен опыт его эксплуатации на действующей системе АС-6.

Данный эскизный проект состоит из двух частей - "Часть I. Аппаратура" и "Часть II. Математическое обеспечение". Кроме того, в состав эскизного проекта в качестве приложений входят следующие материалы, выпущенные предприятием п/я А-ЗИ62:

1. "Система обработки данных АС-6. Центральный процессор. Описание системы команд", 1973г.
2. "Система обработки данных АС-6. Машинно-ориентированные языки", 1973г.
3. "Система обработки данных АС-6. Периферийная машина ПМ-6. Принципы работы и система команд", 1972г.
4. "Вычислительная система БЭСМ-10. Авантпроект", 1973г.
5. "Математическое обеспечение периферийной машины ПМ-6", 1973 г.

## ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР И ОПЕРАТИВНАЯ ПАМЯТЬ

### ВВЕДЕНИЕ

Как уже было отмечено, в структуре системы БЭСМ-10, структуре центрального процессора и его системе команд учтены как требования программирования на языках высокого уровня, так и требования эффективного использования вычислительных ресурсов и ресурсов памяти для решения больших задач.

Реализация многопроцессорной структуры ставит ряд новых проблем, особенно при стремлении достижения высокой производительности на каждом из процессоров системы.

Такими проблемами являются:

проблема распараллеливания решения задачи для эффективной загрузки всех процессоров системы и возникающая при этом проблема обработки общих данных;

проблема эффективной организации системы памяти для обеспечения более полной загрузки каждого процессора системы; проблема повышения производительности каждого процессора в условиях их совместной работы по решению большой задачи.

В многопроцессорных системах остро стоит вопрос обработки общих данных задач, работающих на двух или более процессорах. Данные, которые обрабатываются процедурами операционной системы, в основном общие. Можно предполагать, что с появлением многопроцессорных систем появятся новые методы решения и вес общих данных по отношению к локальным будет увеличиваться. Обработка же общих данных с общим СОЗУ резко упрощается.

Рассмотрим пример. Пусть имеются два процессора, на которых выполняются процедуры  $A_1$  и  $A_2$  соответственно на первом и втором процессоре. Эти процедуры обрабатывают общую переменную В. Разберем каким образом можно организовать доступ к такой переменной в системе с общим СОЗУ, а затем в системе с индивидуальными СОЗУ на процессор.

Для доступа к переменной В процессы синхронизируются с помощью семафора командами закрыть семафор - открыть семафор. Эти команды обрамляют критические секции, где происходит обработка В. Поскольку переменная В находится в общей памяти (СОЗУ), любой из двух процессов всегда будет иметь с самым новым значением переменной В, что и необходимо. Если у каждого процессора есть ассоциативный буфер небольшого размера, то необходимо, чтобы команда открыть семафор очищала ассоциативный буфер. При небольшом размере ассоциативного буфера потери производительности незначительны.

Для обработки переменной В в системе с индивидуальными СОЗУ считывать и записывать значение переменной В необходимо минуя СОЗУ путем явного указания в операциях считывания и записи того факта, что обращение идет к оперативной памяти.

Если объем общих данных велик, то производительность такой системы будет существенно ниже производительности системы с общим СОЗУ.

Дополнительные трудности возникают также и при реализации языков высокого уровня, в которых есть средства распараллеливания вычислений, например, PL-I. В описании данных в PL-I не указывается, обрабатываются ли эти данные только в одном процессе или в нескольких одновременно. Таким об-

разом, либо все процессы задачи нужно выполнять только на одном зафиксированном на все время решения задачи процессоре, либо в параллельно выполняемых процедурах обрабатывать глобальные данные, минуя СОЗУ.

Высокие требования к высокопроизводительной многопроцессорной системе предъявляются и задачами, сводящимися в конечном счете к решению больших систем линейных алгебраических уравнений прямыми (точными) или итеративными методами. К таким задачам относятся многие важные задачи из различных областей физики, управления и оптимального планирования. Поэтому в качестве примеров, наглядно иллюстрирующих возникшие проблемы и пути их решения, целесообразно рассмотреть задачи именно этого типа.

#### Организация памяти и работы процессоров в многопроцессорной системе

На рис. I+3 приведена схема вычисления внутренних точек прямоугольной области при решении задачи Дирихле для уравнения Лапласа. Основной объем вычислительной работы составляет из последовательного вычисления нового значения каждой внутренней точки по значениями соседних точек для большого массива внутренних точек.

Как видно из схемы рис. 2, объем программы внутреннего цикла для обработки одной точки невелик и программа

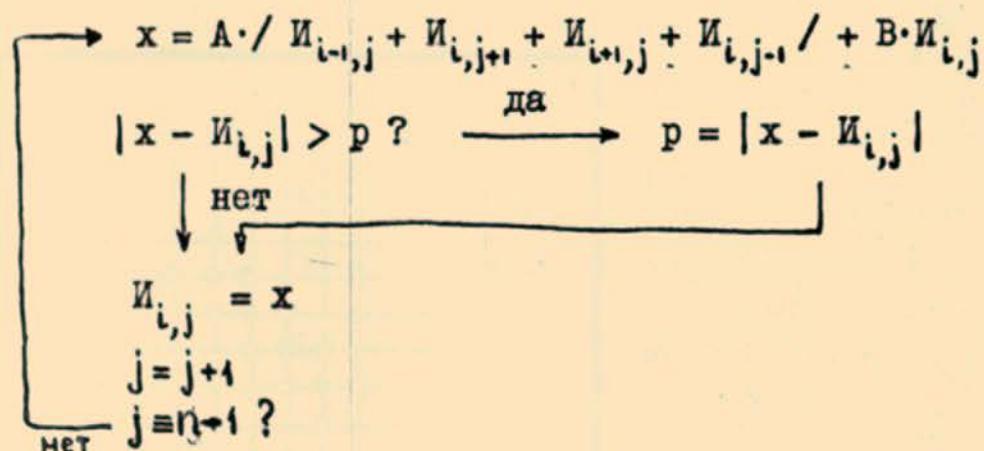


Рис.1. Содержание внутреннего цикла.

# ком	Подготовка цикла по $j$	
I	$\rightarrow$ у, со, до/ио/	2
2	+ , со, д1/ио/	2
3	+ , со, д2/ио/	2
4	+ , со, д3/ио/	2
5	$\times$ , со, с3	2
6	у, с1, д4/ио/	2
7	у, с2, с1	2
8	$\times$ , с1, с6	2
9	+ , с1, со	2
10	- , с2, с1	3
II	-?, с2, с4	2
I2	шу, +к	2
I3	у, с4, с2	2
I4	$\rightarrow$ з, с1, д4/ио/	2
I5	кц, д5/ио/, -м	3
		<u>32 байтн</u>

СО, С1, С2 - промеж. рез-ты

С3 - I/4

С4 - р

С5 - А

С6 - В

ИО - индекс  $j$

ДО - И<sub>i-1,0</sub>

Д1 - И<sub>i,1</sub>

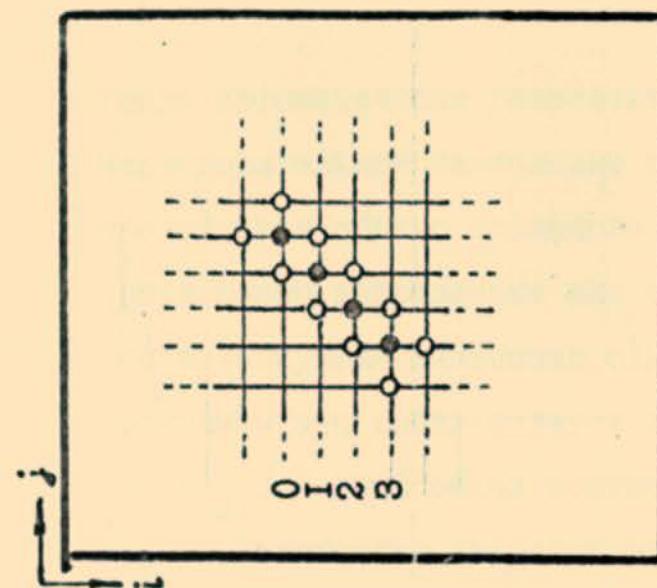
Д2 - И<sub>i+1,0</sub>

Д3 - И<sub>i,-1</sub>

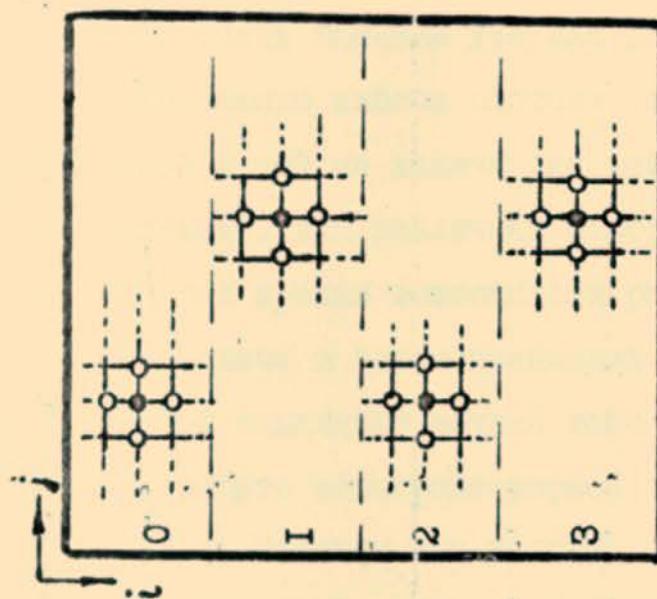
Д4 - И<sub>i,0</sub>

Д5 - дескр. цикла

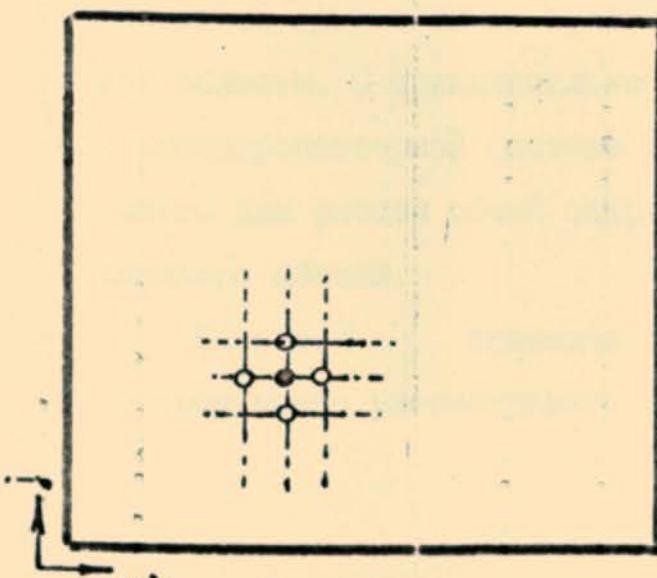
Рис.2. Программа внутреннего цикла в командах АС+6 + БЭСМ-10



В



В



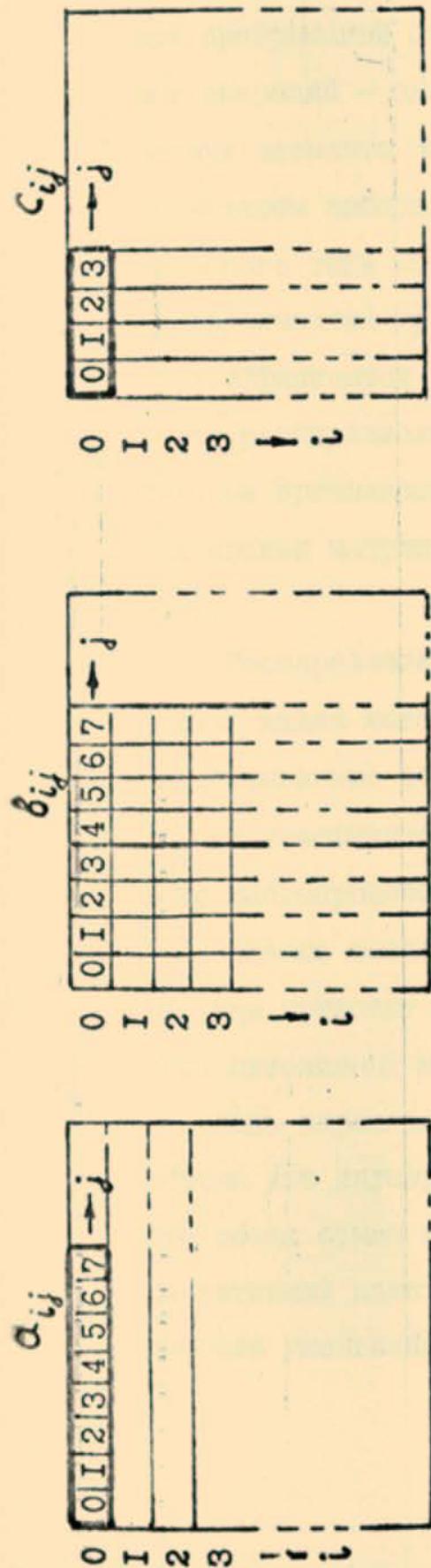
А

Рис. 3. Пример I. Обращение к элементам области:

- А - при работе одного процессора,
- Б - при работе 4-х "слабо связанных" процессоров,
- В - при работе 4-х "сильно связанных" процессоров.

цикла автоматически разместится в буфере команд процессора. Последовательное вычисление внутренних точек области осуществляется обычно последовательным обходом точек по строкам в одном направлении или с чередованием направлений, соответствующим изменению одной из координат области. В программе это соответствует программным циклам по индексам  $i$  и  $j$ . Полный обход соответствует выполнению одной итерации. Количество итераций определяется достижением заданной точности. Объем массива внутренних точек области может быть очень большим уже для двумерных областей. Большая задача такого класса состоит, обычно, из многократного решения одной и той же задачи для различного набора параметров — например, для различных наборов граничных значений. Поэтому, с точки зрения возможности распараллеливания решения большой задачи в таком понимании проблем не возникает. Такой способ выделения ветвей используется в настоящее время при работе однопроцессорной системы в мультипрограммном режиме. Назовем его условно распараллеливанием по циклу параметров. В этом случае каждый процессор системы будет выполнять одну и ту же программу, но для полностью автономной области. С принципиальной точки зрения общность памяти в многопроцессорной системе в этом случае используется только для работы общей операционной системы и каналов внешнего обмена.

На рис. 4 приведен пример умножения матриц, на котором можно рассматривать особенности, характерные для



Математическое описание алгоритма:

$$C_{ij} = \sum_{k=0}^{n-1} a_{ik} \cdot b_{kj}$$

Программа внутреннего цикла в крмандах III АС-6:

Подготовка цикла по  $k$

$\rightarrow y, c_0, d_0(i_0)$	- Выборка $a_{ik}$
$\times, c_0, d_1(i_1)$	- Умножение на $b_{kj}$
$+ , c_1, c_0$	- Накопительное суммирование
$+ , i_1, i_2$	- Изменение индекса для $b$
$\square, d_3(i_0), -\Delta$	- Изменение индекса $i_1$ и повторение

Рис. 4. Пример 2. Распределение памяти и внутренний цикл программы умножения матриц.

прямых (точных) методов решения систем линейных алгебраических уравнений. При использовании этих методов внутренний программный цикл отличается предельно малым количеством операций — одно умножение и одно сложение на два выбираемых элемента матриц  $a_{ij}$  и  $b_{ij}$  и строго регулярным характером выборки коэффициентов матриц. Большая задача подобного типа может состоять в решении систем линейных алгебраических уравнений, отличающихся исходными значениями коэффициентов матрицы. В этом случае естественным способом распараллеливания также как и в предыдущем случае была бы организация параллельной работы каждого процессора над своими матрицами коэффициентов (т.е. цикл по параметрам).

Распараллеливание по циклу параметров для рассматриваемых задач может служить примером разделения задачи на так называемые слабо связанные процессы, которые сравнительно независимо могут выполняться отдельными процессорами многопроцессорной системы.

Однако такая организация работы многопроцессорной системы приводит к выделению в общей оперативной памяти ряда автономных массивов, число которых линейно зависит от числа параллельно работающих процессоров. При большом объеме уже двумерных областей и матриц следует ожидать, что общая сумма автономных массивов превысит емкость оперативной памяти. Это в свою очередь приведет к необходимости уменьшения рабочих массивов, находящихся в общей

оперативной памяти и организации обмена с внешней памятью.

Увеличенная скорость работы процессора по отношению к скорости обмена с внешней памятью и параллельная работа всех процессоров системы ведут к резкому увеличению числа запросов на обмен и делают задачу эффективного обмена с внешней памятью очень сложной. В этих условиях привлекателен вариант использования еще одной ступени оперативной памяти меньшего быстродействия, но большей емкости (mass-MEMORY). Следует отметить, что принятая в БЭСМ-10 система группового обмена очень хорошо подходит для быстрого обмена с такой памятью. К сожалению, вопросы экономической целесообразности дополнительной ступени оперативной памяти и ее реального появления в ближайшие годы остаются открытыми.

Таким образом, естественное и сравнительно легкое по заданию распараллеливание рассматриваемых задач на слабо связанные процессы обостряет и без того острую проблему емкости оперативной памяти и требует в условиях параллельной работы процессоров увеличения емкости оперативной памяти в соответствии с числом процессоров в системе.

Кроме того, одновременная работа всех процессоров системы приводит для рассматриваемого класса задач к большому увеличению числа одновременных запросов к общей оперативной памяти. Произвольный характер обращений со стороны отдельных процессоров при сравнительно большом цикле работы блоков

оперативной памяти может привести к значительному снижению общей производительности многопроцессорной системы. Особен- но большое снижение может произойти при решении задач с очень малым количеством команд, приходящим- ся на одно обращение к памяти, как это имеет место для рассматриваемых примеров. Причем следует подчеркнуть, что указанные трудности имеют принципиальный характер и не мо- гут быть решены за счет увеличения быстродействия памяти. Опыт развития вычислительных машин показывает, что отноше- ние цикла оперативной памяти (требуемой на современном этапе емкости) к такту выполнения команд в процессоре все время увеличивается.

Отмеченные трудности предполагается преодолеть за счет более эффективной структурной организации высокопроизво- дительной многопроцессорной системы БЭСМ-10.

С целью обеспечения эффективной работы многопроцес- сорной высокопроизводительной системы в аванпроекте БЭСМ-10 была предложена иерархическая структура оператив- ной памяти, состоящая из трех уровней: ассоциативной ло- кальной памяти процессора емкостью в несколько десятков слов, максимально приближенной к процессору; ассоциатив- ной сверхоперативной памяти (СОЗУ), емкостью в несколько тысяч слов, служащей промежуточной ступенью между основ- ной оперативной памятью и процессорами системы; основной

оперативной памяти емкостью 0,5 + 1 млн. слов.

Одной из принципиальных особенностей предложенной структуры является обобщение сверхоперативной ступени (СОЗУ) для всех процессоров системы в отличие от известных вариантов организации многопроцессорных систем с локальными СОЗУ при процессорах. При выборе и разработке этого структурного решения преследовались цели эффективной разгрузки основной оперативной памяти, с одной стороны, и обеспечение более полной загрузки всех процессоров системы с другой стороны. При этом также имелось в виду достижение высокой общей производительности всей многопроцессорной системы при решении больших задач, в том числе и задач рассматриваемого типа, которые наиболее тяжелы для универсальных многопроцессорных систем.

Достижение указанных целей основывается на следующих возможностях:

- 1) При решении важных и критичных по своим требованиям к памяти больших задач возможно и целесообразно параллельное выполнение процессов, имеющих общие данные (или так называемых сильно связанных процессов).
- 2) Использование интегральных элементов в СОЗУ позволяет получить большую пропускную способность для обеспечения одновременных больших потоков обращений со стороны всех процессоров системы (включая каналы обмена с внешней системой) и эффективного группового обмена между ступенями ОЗУ и СОЗУ.

Для более детального рассмотрения отмеченных возможностей обратимся к рис. 3 б) и в).

Наличие большого массива элементов области и использование очень малого числа соседних элементов при вычислении нового значения элемента позволяет, вообще говоря, организовать параллельную работу над отдельными довольно большими частями общего массива области, как это схематично изображено на рис. 3 б).

Работа всех процессоров над одним общим массивом области имеет большую выгоду с точки зрения возможности помещения всего массива в оперативную память и исключения или значительного уменьшения обмена с внешней памятью. Параллельная работа над частичными массивами довольно автономна и синхронизация параллельных процессов возникает лишь при окончании обработки частичных массивов, т.е. после выполнения сравнительно длинных последовательностей команд. Отметим, однако, что результат работы каждого процессора над своим частичным массивом используется при работе другого процессора над соседним частичным массивом на границах этих частичных массивов. Отсутствие учета этого обстоятельства приведет к увеличению числа итераций, необходимых для достижения заданной точности, и тем самым к уменьшению выгоды, получаемой за счет параллельной работы.

Доля возможных потерь из-за разбиения общей области на частичные можно уменьшить, если использовать специальные

мери по связи границ, но это может потребовать разработки специальных оценок и дополнительного программирования.

Рассмотрим далее возможный эффект от использования сверхоперативной ступени памяти. Если емкость этой ступени составляет тысячи слов, то целесообразно считываемые из ОЗУ элементы массива области оставлять в СОЗУ. В этом случае при вычислении элементов следующей строки две строки из трех требуемых уже могут находиться в СОЗУ и тем самым число обращений к ОЗУ по считыванию сокращается в три раза (до одного обращения). При каждом выполнении программного цикла выполняется одна запись нового значения элемента. Таким образом, общее число обращений к ОЗУ при использовании СОЗУ в роли ассоциативной буферной ступени сокращается в два раза по сравнению с вариантом без заполнения СОЗУ элементами массива.

Немаловажным обстоятельством является также то, что при обобщенном СОЗУ сама программа содержится в СОЗУ в единственном "экземпляре", а не размножается в каждом локальном СОЗУ, как это получается в структурном варианте с локальными СОЗУ.

Следует отметить, что разгрузка более медленной ступени памяти за счет исключения повторных обращений с помощью ассоциативного буфера использована еще в БЭСМ-6. Однако малая емкость буфера позволяет исключать те повторные обращения, которые производятся на интервале выполнения

сравнительно короткой последовательности команд. Такой эффект обязан наличию в алгоритме вычисления связности по использованию простых переменных и отдельных элементов массива. При емкости СОЗУ в несколько тысяч слов можно использовать уже связность между строками или векторами.

Обобщение памяти на уровне сверхоперативной памяти предоставляет новые возможности для организации еще более эффективной параллельной работы процессоров и более эффективного использования емкости сверхоперативной памяти. На рис. 3 в) приведена схема параллельной работы четырех процессоров в условиях максимально тесной связи между собой. Выгоды такой организации параллельной работы заключаются в следующем. При одной и той же емкости СОЗУ можно эффективно работать с большими длинами строк области. Оперативное использование результата не требует применения специальных мер по связи частичных областей.

Особо выгодной тесная параллельная работа процессоров оказывается при использовании итеративных методов типа Зейделя и прямых методов. При использовании метода Зейделя новое вычисленное значение неизвестного используется при вычислении следующего неизвестного и это может давать для областей сходимости этого метода многократное сокращение времени решения по сравнению с методом простых итераций, который был бы удобнее для процессов с меньшей связью. В этих случаях описываемая тесная работа процессоров системы оказывается очень выгодной, так как позволяет держать

в ОЗУ один общий массив коэффициентов и максимально использовать СОЗУ для уменьшения количества обращений к ОЗУ и согласования высоких скоростей работы каждого процессора с высокой пропускной способностью СОЗУ при их параллельной работе.

Преимущества обобщения СОЗУ и тесной параллельной работы процессоров системы наглядно видны на примере умножения матриц, приведенном на рис. 4. При вычислении одного элемента результирующей матрицы необходимо выполнить суммирование парных произведений элементов строки матрицы  $a_{ij}$  на элементы столбца матрицы  $b_{ij}$ . Таким образом, при вычислении одним из процессоров одного значения  $c_{ij}$  в СОЗУ автоматически окажется (при использовании групповой выборки из ОЗУ по 8 слов) одна полная строка матрицы  $a_{ij}$  (вектор  $a_{ok}$ ) и восемь столбцов матрицы  $b_{ij}$  (по восемь элементов векторов  $b_{kj}$ ). Если остальные три процессора будут вычислять следующие элементы строки результирующей матрицы -  $c_{o1}, c_{o2}, c_{o3}$ , то они будут выбирать элементы строки  $a_{ok}$  и элементы столбцов  $b_{k1} \div b_{k3}$  уже находящиеся в СОЗУ. На этом этапе выполнения будет произведено 2 групповых обращений для всех 4-х процессоров. Однако, если на следующем этапе вычислять следующие четыре элемента столбцов  $c_{oj} \div c_{3j}$  матрицы  $C_{ij}$  (рис. 4), то потребуется вновь вызывать из ОЗУ лишь одну очередную строку матрицы  $b_{ij}$ , общую для всех 4-х процессоров. Вызов элементов столбцов на этом

этапе не потребуется вообще на протяжении всего интервала времени вычисления элементов  $C_{i0} + C_{i7}$ . В результате, число обращений к ОЗУ уменьшится в итоге в восемь раз. Выгода обобщения СОЗУ в рассматриваемом случае проявляется в том, что в обобщенном СОЗУ (суммарной емкости) размер матрицы или клетки матрицы больше чем в случае индивидуальных СОЗУ при одинаковых способах управления СОЗУ.

Необходимо отметить, что свойствами связности по считыванию и результату, использование которых было показано на контрольных примерах, в той или иной мере обладают практически все реализуемые алгоритмы. Поэтому, меры, направленные на эффективное использование этих свойств, а именно - обобщение СОЗУ и возможность организации тесного взаимодействия параллельной работы процессоров являются мерами универсального характера, отличающими универсальные системы широкого назначения от специализированных систем.

При рассмотрении тесного взаимодействия параллельно работающих процессоров предполагалось, что необходимая при этом координация работы выполняется с использованием аппаратных средств при минимуме программных средств.

Эскизная проработка показала, что в качестве программного средства можно использовать модификацию операции синхронизационного считывания, а в качестве аппаратного - специальную схему, автоматически обеспечивающей порядок взаимодействия, задаваемый операциями синхронизационного считывания.

## УПРАВЛЕНИЕ ОПЕРАТИВНОЙ ПАМЯТЬЮ

Структура оперативной памяти (ОП) вычислительной системы БЭСМ-10 основана на использовании трех уровней иерархии, которыми являются:

Ассоциативные буферные регистры процессора;

Сверхоперативное запоминающее устройство (СОЗУ);

Оперативное запоминающее устройство на магнитных сердечниках (МОЗУ).

Ассоциативные буферные регистры, представляющие первый уровень иерархии, размещаются в каждом процессоре, а СОЗУ и МОЗУ образуют общую память процессоров.

Общая оперативная память системы состоит из нескольких независимо работающих модулей, количество и время цикла которых выбрано с учетом требований параллельно работающих процессоров.

СОЗУ представляет собой набор модулей, с каждым из которых связана группа модулей МОЗУ.

Емкость СОЗУ такова, что в нем помещается почти вся информация, необходимая процессорам в данный интервал работы, поэтому одновременные обращения к общим командам и общим данным из нескольких процессоров не вызывают существенного снижения производительности одного процессора.

Типовой для БЭСМ-10 принята четырехпроцессорная структура, основанная на таком способе организации, при котором как оперативная, так и сверхоперативная память

является общей для всех четырех процессоров и равнодоступной для каждого из них. Это позволяет сделать сверхоперативную память полностью скрытой от программиста, для которого она просто не существует.

В структуре БЭСМ-10 сверхоперативная память представляет собой лишь аппаратное средство для уменьшения эффективного цикла оперативной памяти.

На первом уровне иерархии памяти (ассоциативные регистры) используются математические адреса; на втором и третьем уровнях, образующих общую память, используются физические адреса.

Физический адрес, вырабатываемый процессором, идентифицирует все хранящие элементы в устройствах, подключенных к процессору. В СОЗУ ассоциативным способом отображается некоторая часть этой общей памяти. Адреса ячеек, в данный момент находящихся в СОЗУ, хранятся в специальной памяти адресов и при каждом обращении к памяти проверяется, находится ли среди них данный физический адрес.

Память адресов служит центральным узлом объединения процессоров при обращениях к общей памяти и имеет высокую пропускную способность за счет существенного сокращения ее емкости и разделения на независимо работающие модули. Количество модулей и скорость их работы согласована со скоростью работы процессоров. Сокращение емкости памяти адресов достигнуто с помощью использования адресов блоков (а не адресов слов). В данном случае под блоком понимается группа

слов с последовательными адресами.

Выполнение операций обращения к памяти из процессора реализуется по несколько более сложному, чем обычно, алгоритму, который можно свести к следующим правилам:

Операция "запись из процессора в память" сначала выполняется только в СОЗУ с одновременной установкой в регистре, хранящем адрес блока, в котором находится слово, признаков готовности и записи слова.

Операции "считывание в процессор из памяти" либо считывает слово из СОЗУ, либо формирует запрос к МОЗУ на перемещение из МОЗУ в СОЗУ всех тех слов блока, у которых отсутствует признак готовности, если это перемещение требует предварительной переписи слов из СОЗУ в МОЗУ, то переписываются только те слова блока, у которых отсутствует признак записи.

Ниже приведены основные параметры общей памяти. Модули СОЗУ построены на базе системы универсальных элементов БЭСМ-10 и приборов "Импровизация-1". Каждый модуль СОЗУ содержит 512 слов и имеет время цикла, равное 2 тактам. Наличие 8 модулей с временем цикла, равным 2 тактам, обеспечивает, что 4 процессора, имеющие время цикла, равное 2 тактам, в большинстве случаев обращаются к разным модулям.

Модули МОЗУ представляют собой стандартные устройства ферритовой памяти (например, серийно выпускаемые устройства ЕС-3205), имеющие время цикла 1,25 мксек. Выбор стандартного устройства памяти будет уточнен на этапе технического проектирования.

Для получения нужной пропускной способности общей памяти в систему введено 32 модуля МОЗУ, причем они разделены на 8 групп по 4 модуля, каждая группа соединена с соответствующим ей модулем СОЗУ. Такое соединение обеспечивает одновременный обмен между СОЗУ и МОЗУ восемью словами, т.е. одним блоком, при выполнении операции считывания из МОЗУ в СОЗУ могут быть переданы 4 блока с интервалами в 2 такта.

Память, хранящая адреса блоков, находящихся в СОЗУ, построена на тех же элементах, что и СОЗУ, т.е. с использованием приборов "Импровизация-І". Емкость этой памяти для СОЗУ в 4К слов должна быть равна 512 10-разрядных адресов с соответствующими признаками.

Физический адрес слова состоит из 19 разрядов, так как восемь слов образуют блок, то адрес блока - 16 разрядов. При обращении к модулям СОЗУ физический адрес, определяющий блок, преобразуется по функции расстановки в 6 разрядов, а 10 разрядов этого адреса используются для ассоциативного поиска в восьми регистрах.

На этапе технического проектирования вид функции расстановки будет уточнен. Для описываемого случая два разряда из шести разрядов, полученных в результате применения функции расстановки, указывают номер модуля памяти адресов, а другие четыре разряда определяют группу ячеек внутри модуля памяти адресов.

Память адресов блоков, разделенная на 4 независимых модуля, имеет максимальную пропускную способность одного модуля, равную 1 такту. В результате, с учетом разбиения СОЗУ на 8 модулей в лучшем случае возможно выполнение 4 одновременных обращений к СОЗУ за 1 такт.

Далее описаны схемы, временные диаграммы и алгоритмы управления общей памятью.

Прежде всего рассмотрим схему коммутации запросов процессоров, приведенную на рис.5. Эта схема осуществляет передачу кода с 6 выходных регистров процессоров (ВР) на 4 приемные регистра (ПР) модулей памяти адресов блоков, которые в дальнейшем будут называться адресными схемами. Номер приемного регистра адресной схемы указывается двумя разрядами функции расстановки. На рис.6 показаны все поля физического адреса, поступающего на ПР.

При одновременных обращениях нескольких процессоров к адресным схемам максимальная пропускная способность равна одновременной передаче четырех запросов каждый такт с ВР на ПР; если несколько процессоров обращаются к одной адресной схеме, то очередность последовательно проходящих запросов устанавливается схемой таблицы, уточняемой на стадии технического проекта. Каждая адресная схема содержит 16 групп регистров адресов блоков, причем номер группы указывается четырьмя разрядами функции расстановки (см.рис.7 и рис. 6 ).

В каждой группе 8 ре-

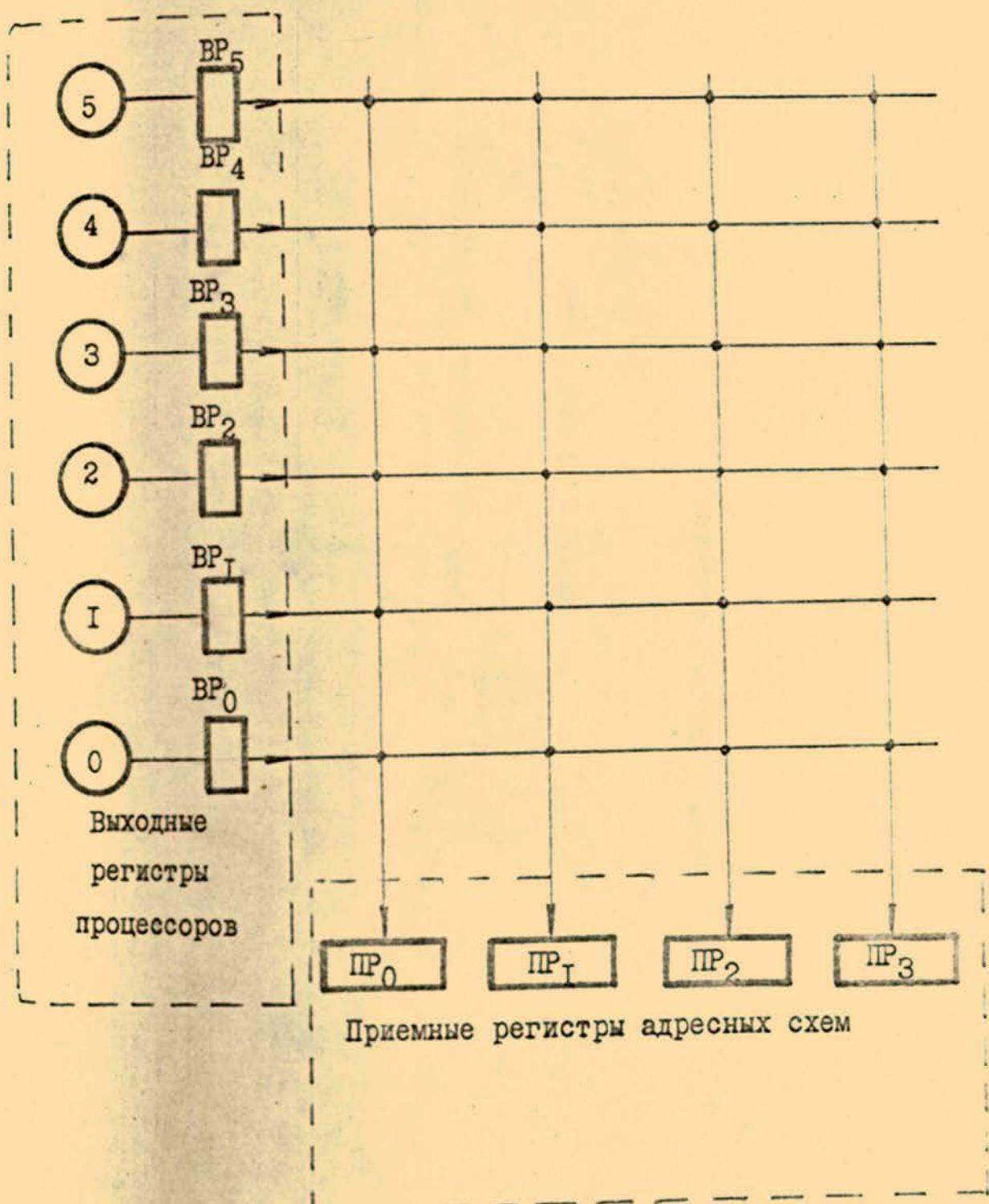


Рис. 5. Схема коммутации запросов процессоров  
к адресным схемам

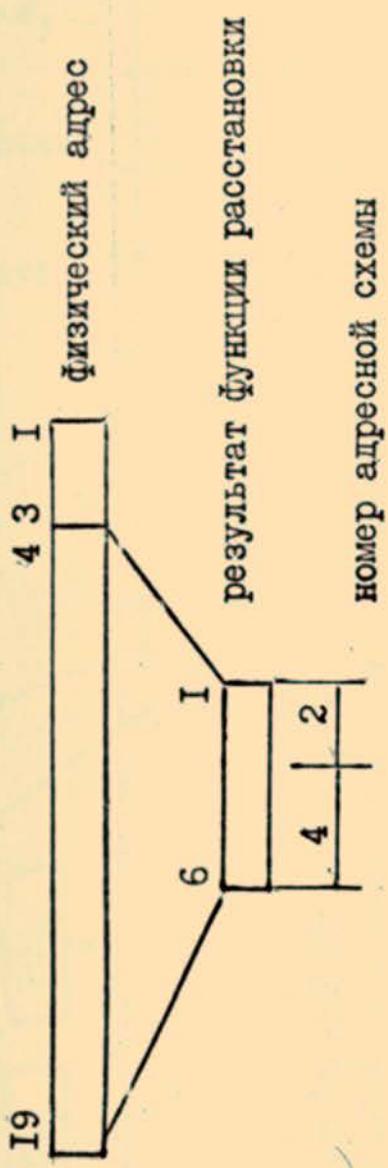


Рис.6. Поля физического адреса.

ПР<sub>i</sub> - приемные регистры  
адресных схем ,

РВ<sub>i</sub> - регистры выборки ,

РНС<sub>i</sub> - регистры

фиксации несовпадений ,

РСП<sub>i</sub> - регистры

фиксации совпадений ,

$i = 0 \div 3$

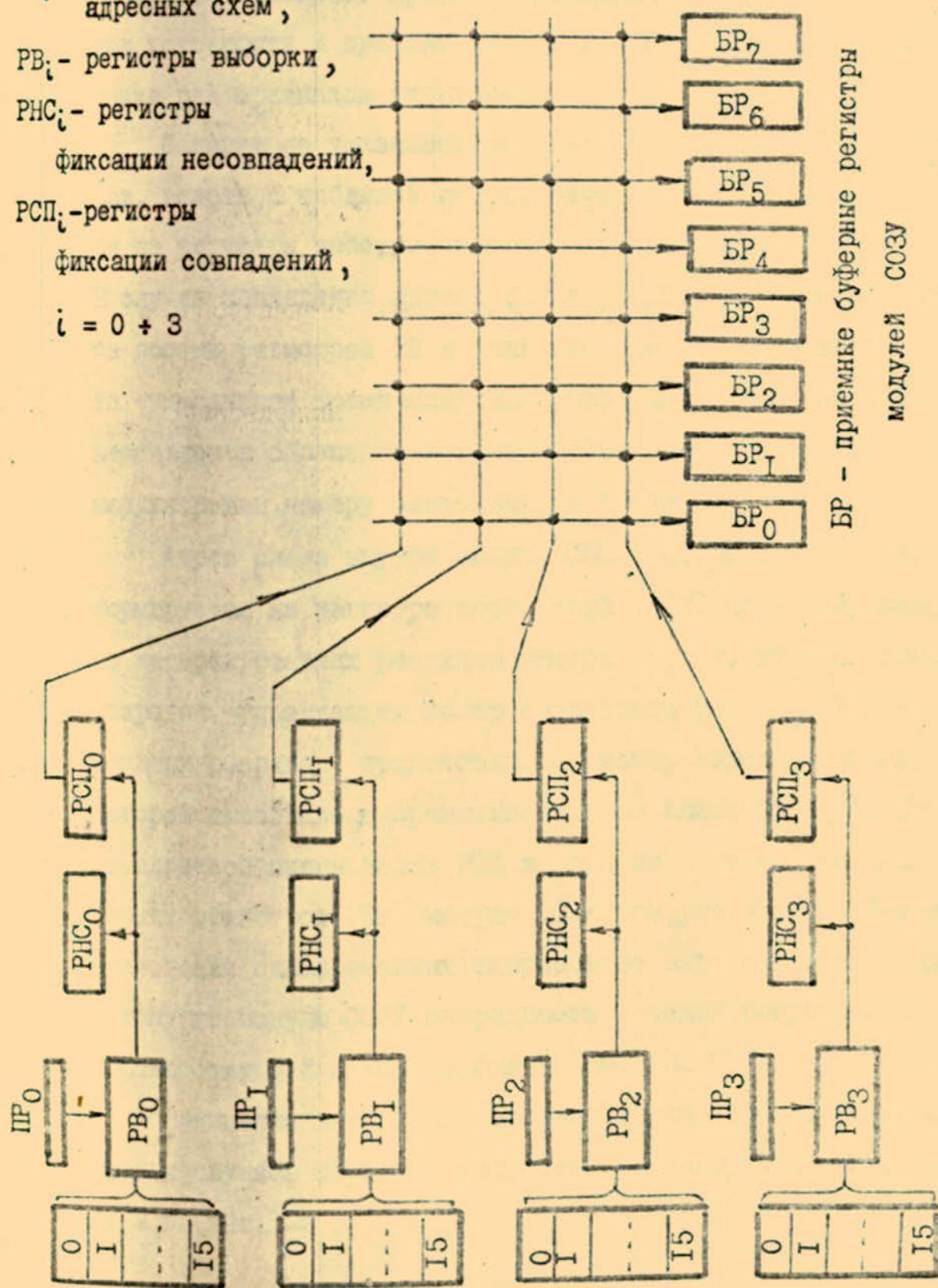


Рис.7. Схема коммутации запросов адресных схем к СОЗУ

$\text{ПР}_i$  - приемные регистры  
адресных схем ,

$\text{PB}_i$  - регистры выборки ,

$\text{PHC}_i$  - регистры  
фиксации несовпадений ,

$\text{PCII}_i$  - регистры  
фиксации совпадений ,

$$i = 0 \div 3$$

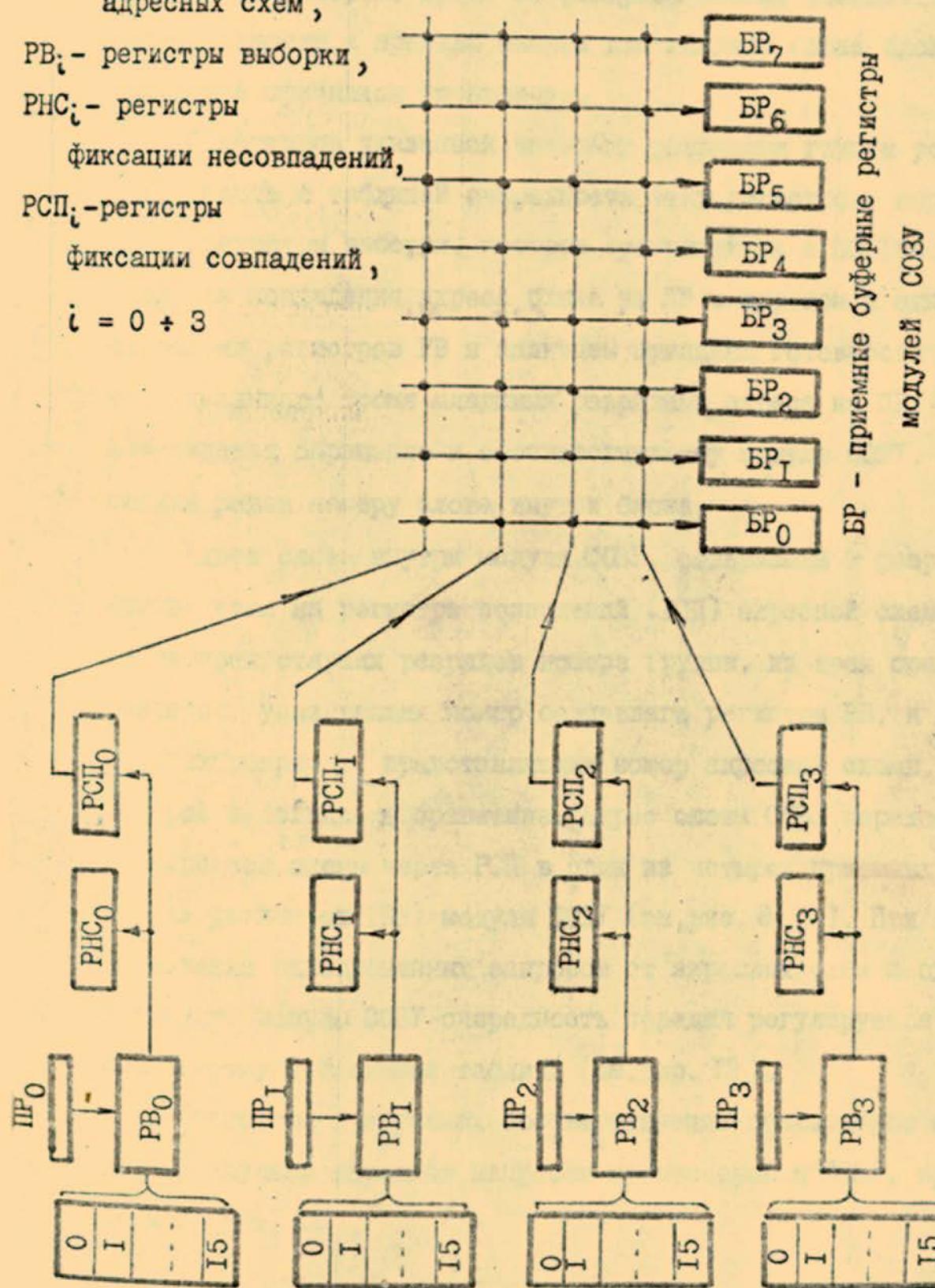


Рис. 7. Схема коммутации запросов адресных схем к СОЗУ

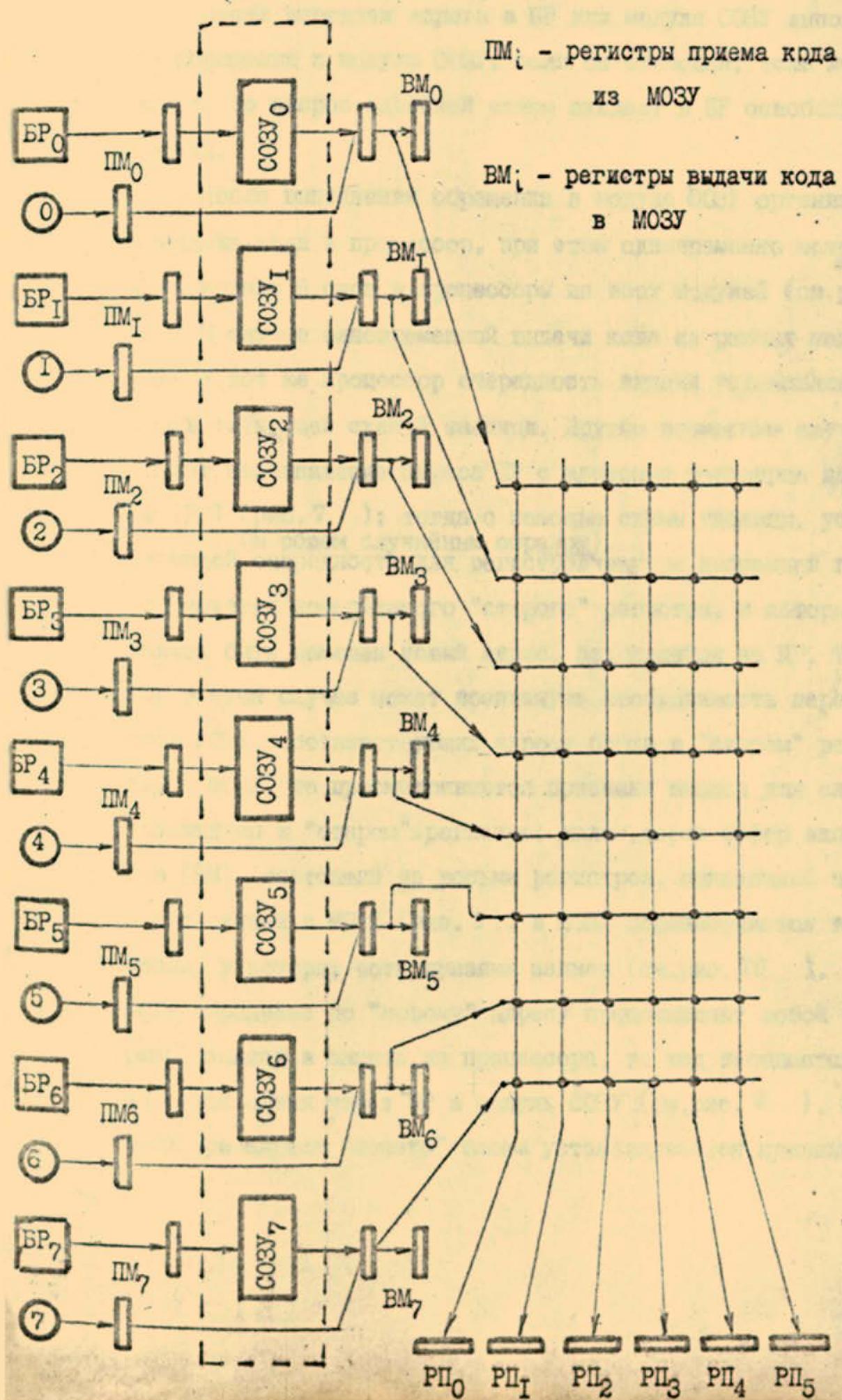
гистров, в которых, кроме 10 разрядов адреса блока, находятся признак готовности и признак записи для каждого слова блока, а также ряд признаков управления.

Содержимое указанной четырьмя разрядами группы регистров, вместе с таблицей очередности этих регистров, передается на регистры выборки, которые сравниваются с ПР (см.рис. 7). В случае совпадения адреса блока на ПР с адресом в одном из восьми регистров РВ и наличием признака готовности у слова, указанного тремя младшими разрядами адреса на ПР, осуществляется обращение к соответствующему модулю СОЗУ. Номер модуля равен номеру слова внутри блока.

Адрес слова внутри модуля СОЗУ, содержащий 9 разрядов, формируется на регистре совпадений (РСП) адресной схемой из четырех старших разрядов номера группы, из трех средних разрядов, указывающих номер совпавшего регистра РВ, и двух младших разрядов, представляющих номер адресной схемы, в которой выполнялось сравнение. Адрес слова СОЗУ передается из адресной схемы через РСП в один из четырех приемных буферных регистров (БР) модуля СОЗУ (см.рис. 8). При возникновении одновременных запросов от адресных схем к одному и тому же модулю СОЗУ очередьность передач регулируется соответствующей схемой таблицы (см.рис. I2).

Временные диаграммы, соответствующие прохождению основных случаев передачи запросов процессоров к СОЗУ, приведены на рис. I2.

Рис. 8. Схема выдачи кода из СОЗУ в процессоры



После передачи адреса в БР для модуля СОЗУ выполняется обращение к модулю СОЗУ, если он свободен; если модуль занят, то запрос адресной схемы ожидает в БР освобождения модуля.

После выполнения обращения в модуле СОЗУ организуется передача кода в процессор, при этом одновременно может быть выдано 6 слов в процессоры из всех модулей (см.рис. 8 )

В случае одновременной выдачи кода из разных модулей в один и тот же процессор очередь выдачи устанавливается соответствующей схемой таблицы. Другим возможным случаем служит несовпадение адреса ПР с адресами регистров выборки (РВ) (рис. 7 ); тогда с помощью схемы таблицы, устанавливающей (в общем случайном образом) очередь для регистров внутри выбранной группы, формируется номер самого "старого" регистра, в который должен быть помещен новый адрес, находящийся на ПР. Так как в этом случае может возникнуть необходимость переписи слов СОЗУ, соответствующих адресу блока в "старом" регистре, в МОЗУ, то просматриваются признаки записи для слов, хранящихся в "старом" регистре; далее, через буфер запросов (БМ), состоящий из восьми регистров, показанный на схеме связей с МОЗУ (рис. 9 ), в МОЗУ переписываются те слова, у которых есть признак записи (см.рис. 10 ).

Если обращение по "новому" адресу представляет собой операцию записи в память из процессора, то код передается из адресной схемы через БР в модуль СОЗУ (см.рис. 9 ), а в регистре адреса "нового" блока устанавливается признак

БР<sub>i</sub> - буферные регистры приема в СОЗУ

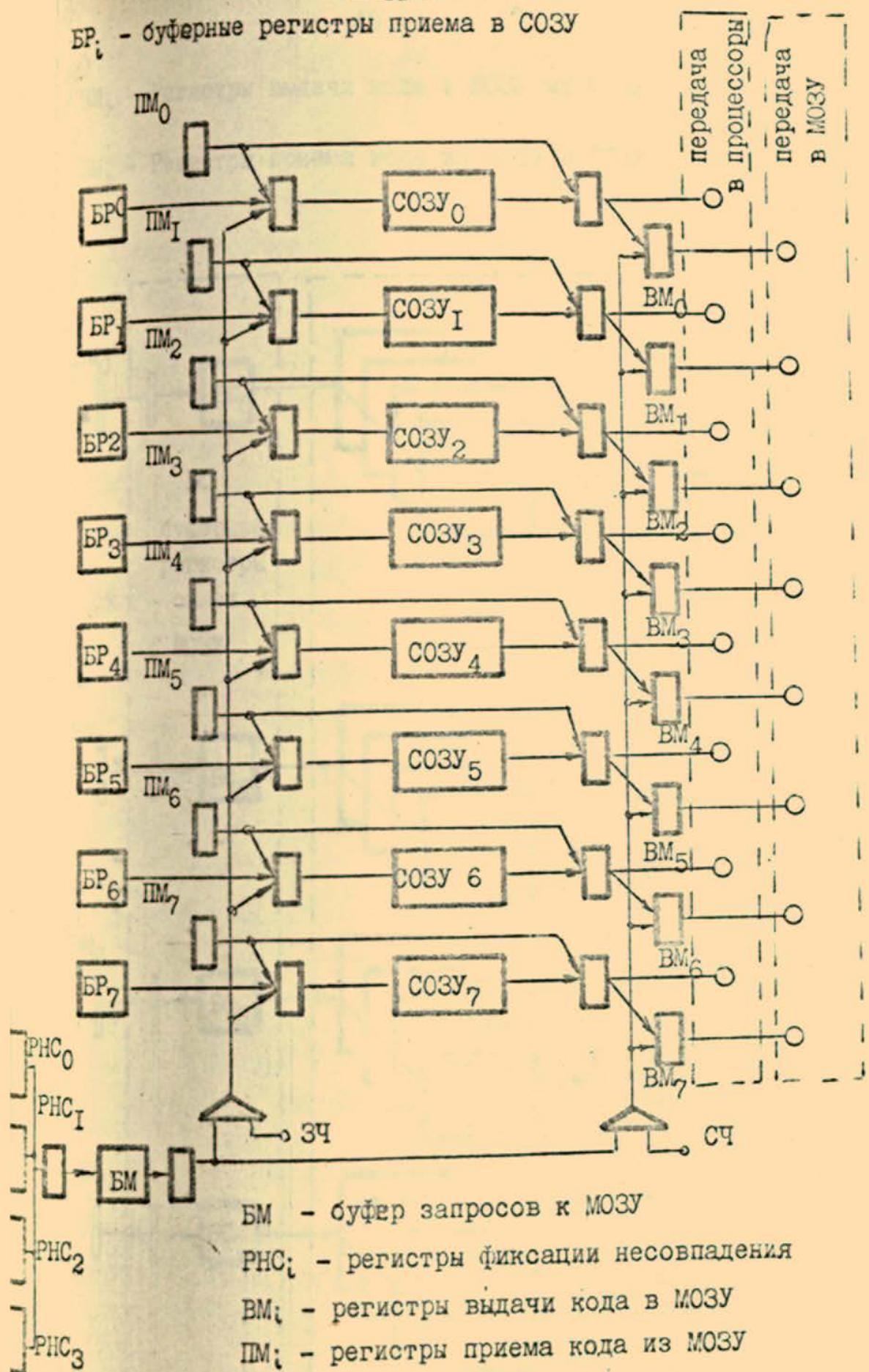


Рис.9. Схема связей с МОЗУ

ВМ<sub>i</sub> - Регистры выдачи кода в МОЗУ из СОЗУ

ПМ<sub>i</sub> - Регистры приема кода из МОЗУ в СОЗУ

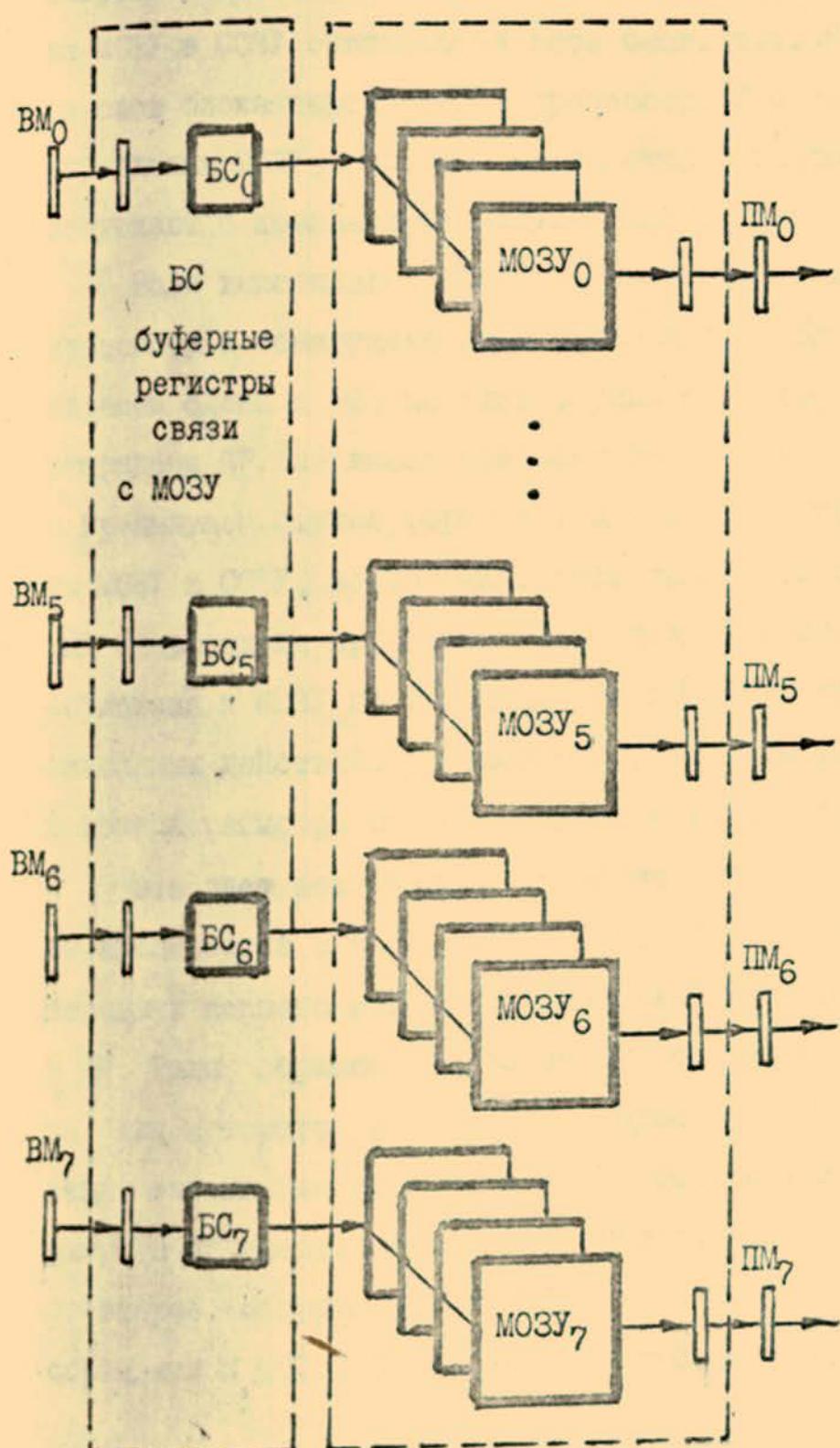


Рис. 10. Схема управления МОЗУ

готовности и признак записи для слова в блоке.

Если обращение по "новому" адресу представляет собой операцию считывания из памяти (см.рис. I3 ) в процессор, то из МОЗУ в ССЗУ считывается весь блок, т.е. 8 слов, и одно из слов блока передается в процессор. С этой целью в схему управления МОЗУ (рис. I0 ) передается 8 запросов, которые запускают 8 независимых модулей МОЗУ.

Если выполняется операция считывания из памяти в процессор и обнаружено совпадение адреса блока на ПР с адресом блока в РВ, но слово, указанное тремя младшими разрядами ПР, не имеет признака готовности, тогда, как и в предыдущем случае, формируется запрос на передачу слов из МОЗУ в ССЗУ, но запрашиваются только те слова, у которых отсутствуют признаки готовности. Выполнение операции обращения к МОЗУ по считыванию сопровождается, кроме уже описанных действий, установкой признака обращения к МОЗУ и номера регистра из БМ в регистре адреса блока.

Это дает возможность построить очереди тех запросов из процессоров к ССЗУ, которые появляются после момента передачи первого запроса по считыванию слов из МОЗУ в ССЗУ в БМ. Таким образом, адрес на ПР, совпавший с адресом блока, для которого уже передан запрос в БМ на выполнение операции считывания из МОЗУ в ССЗУ, передается из адресной схемы в БР вместе с номером регистра БМ, в котором хранится запрос, отправленный в МОЗУ, и с признаком блокировки обращения к ССЗУ. После выдачи слов из МОЗУ в ССЗУ признаки

запроса в МОЗУ, хранящиеся в регистре адреса блока и в регистре БМ, гасятся, а также гасится блокировка обращения к СОЗУ в БР для запросов, накопленных в очередях к тем словам блока, которые считывались из МОЗУ. При выдаче кода из модулей МОЗУ выполняется декодирование по коду Хэмминга, что обеспечивает коррекцию одиночной ошибки и обнаружение двойной ошибки.

Описанные временные соотношения представлены на временных диаграммах на рис. 7,8,9.

На рис. 7 показан период с длительностью 4 такта от выдачи кода с ВР до приема кода на РП. Основные случаи параллельной работы процессоров, в которых осуществляется либо полное совмещение обращений процессоров к общей памяти, либо одновременное обращение к одной адресной схеме, либо, наконец, одновременные обращения к одной и той же адресной схеме и к одному и тому же модулю СОЗУ, показаны на рис. 8.

Основная временная диаграмма обращения к памяти, учитывая наличие различных буферных регистров, показана на рис. 9.

Для пояснения описанного выше алгоритма управления памятью на рис.10-16 приведена блок-схема этого алгоритма.

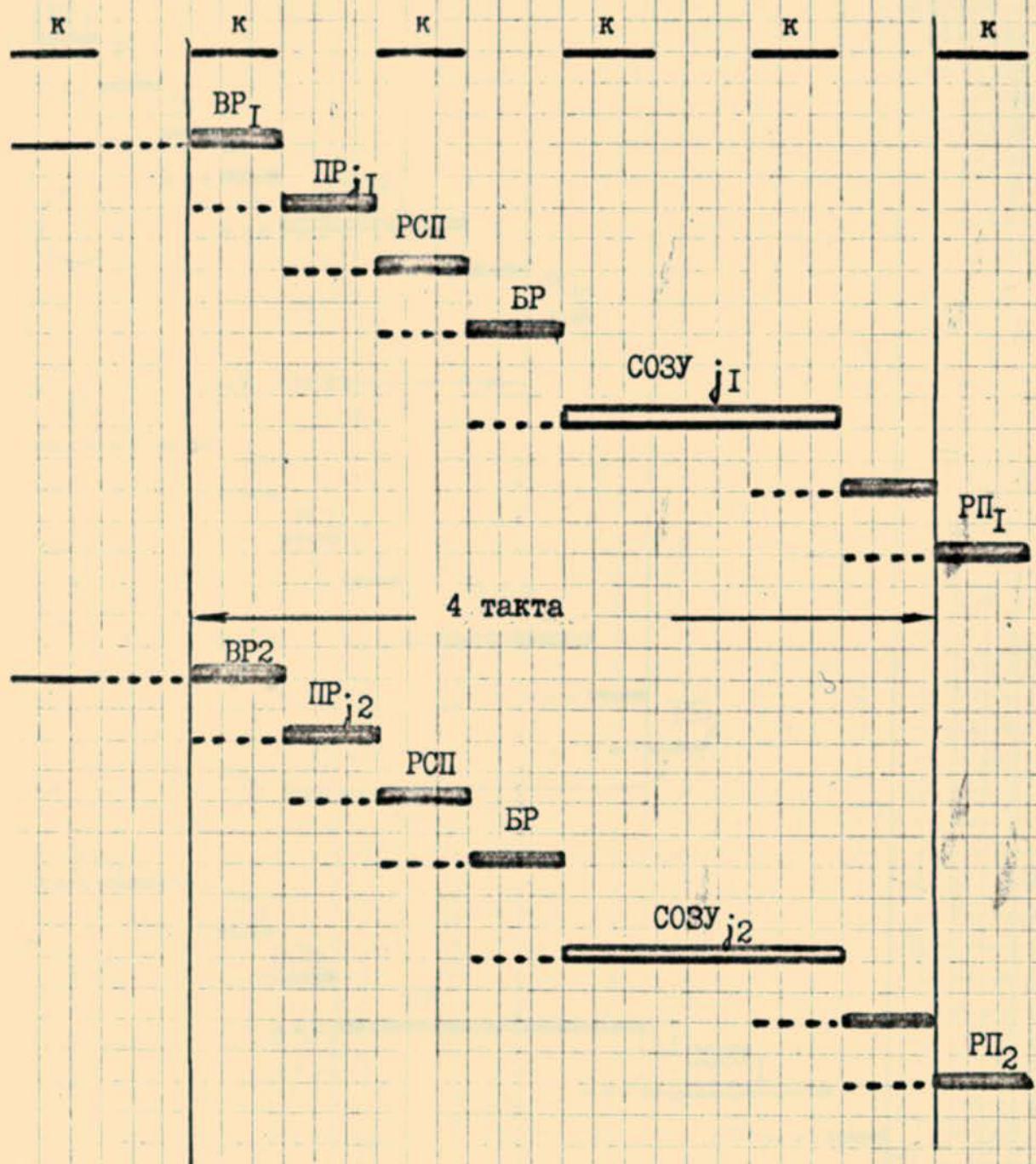
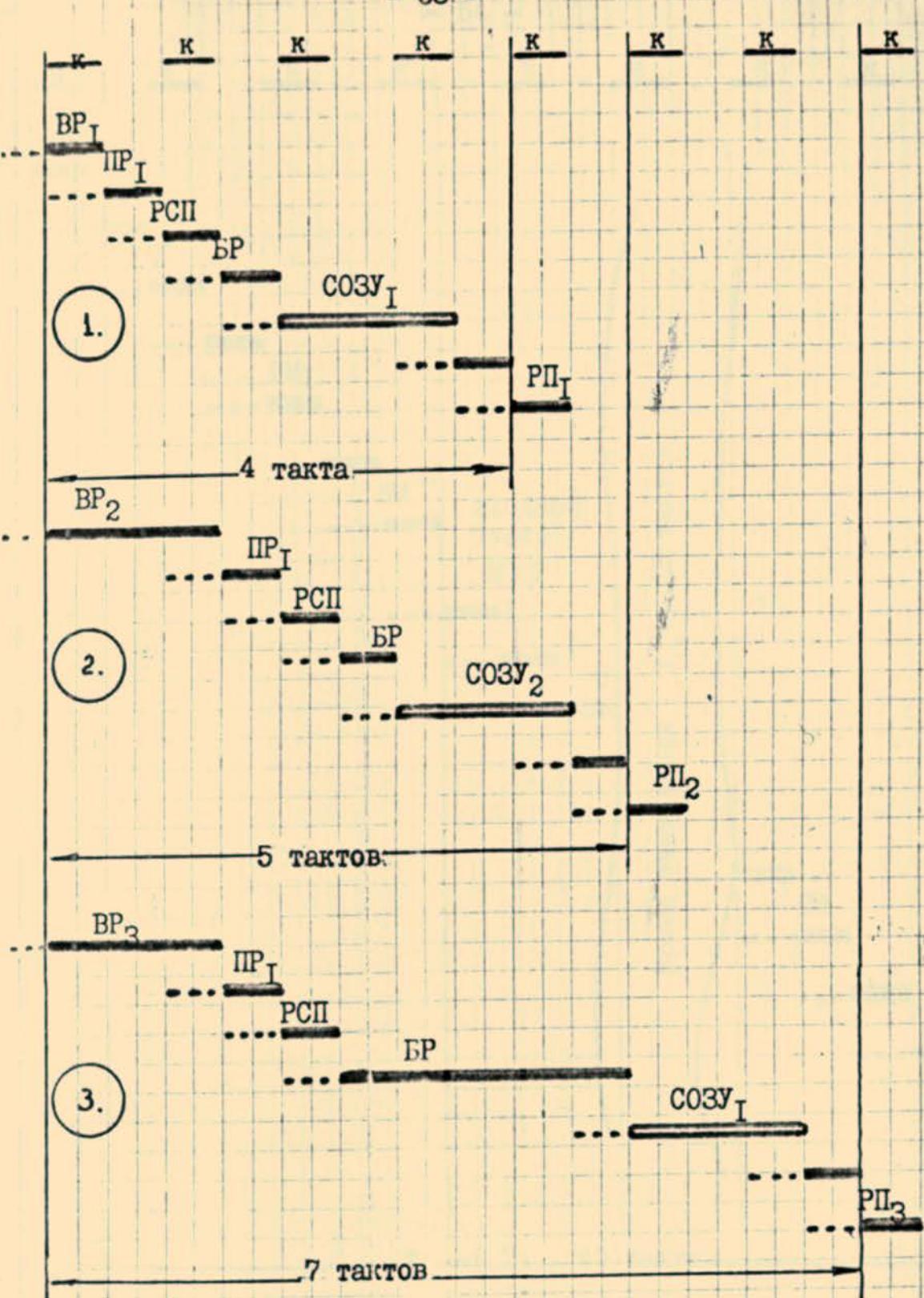


Рис. II. Временная диаграмма параллельной работы двух процессоров



1. Минимальное время обращения к памяти
2. Два процессора используют общую адресную схему
3. Два процессора используют общую адресную схему и общий модуль COZY

Рис. I2. Временные диаграммы обращения процессоров в память, учитывающие их взаимное влияние

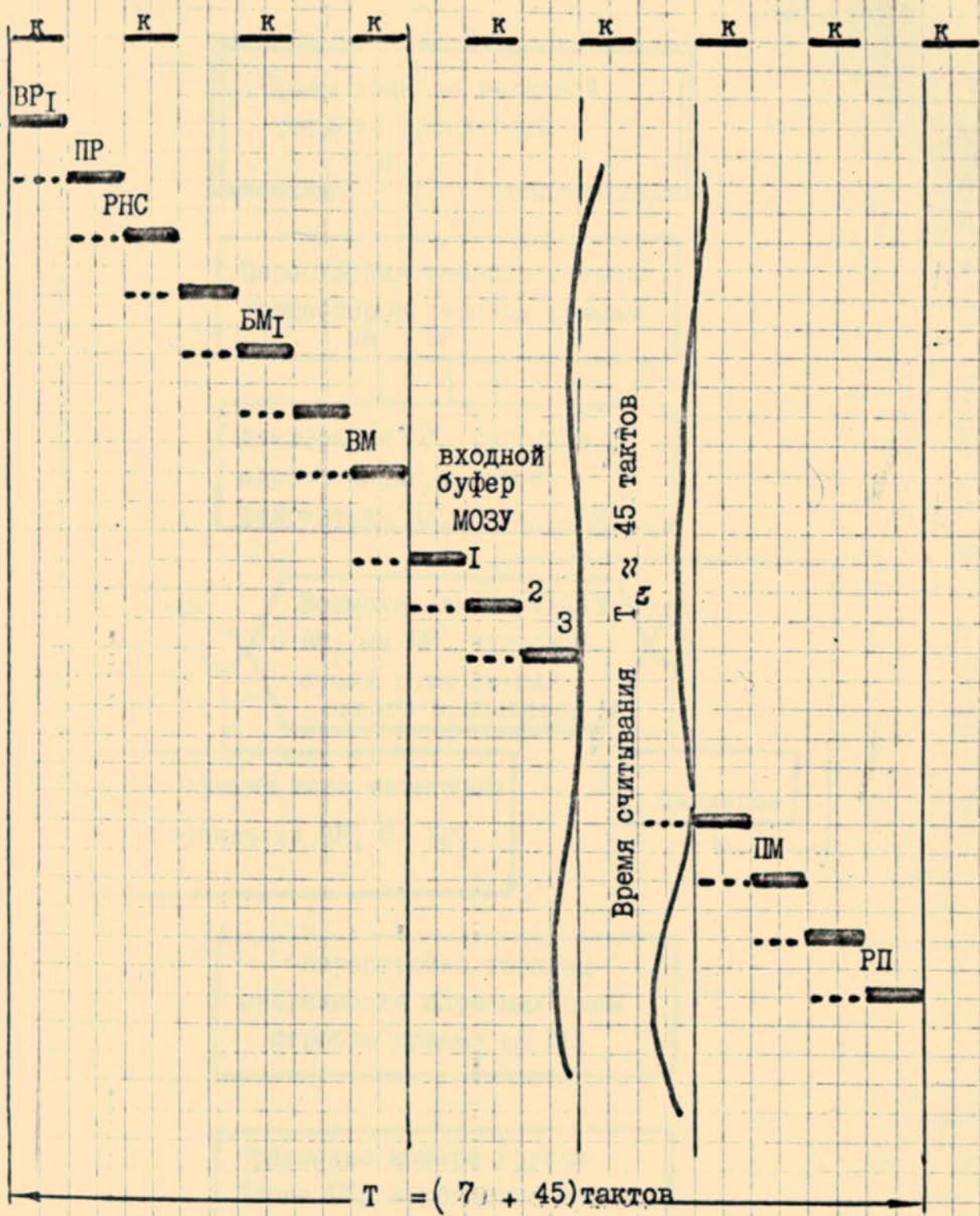


Рис. I3. Временная диаграмма считывания из памяти

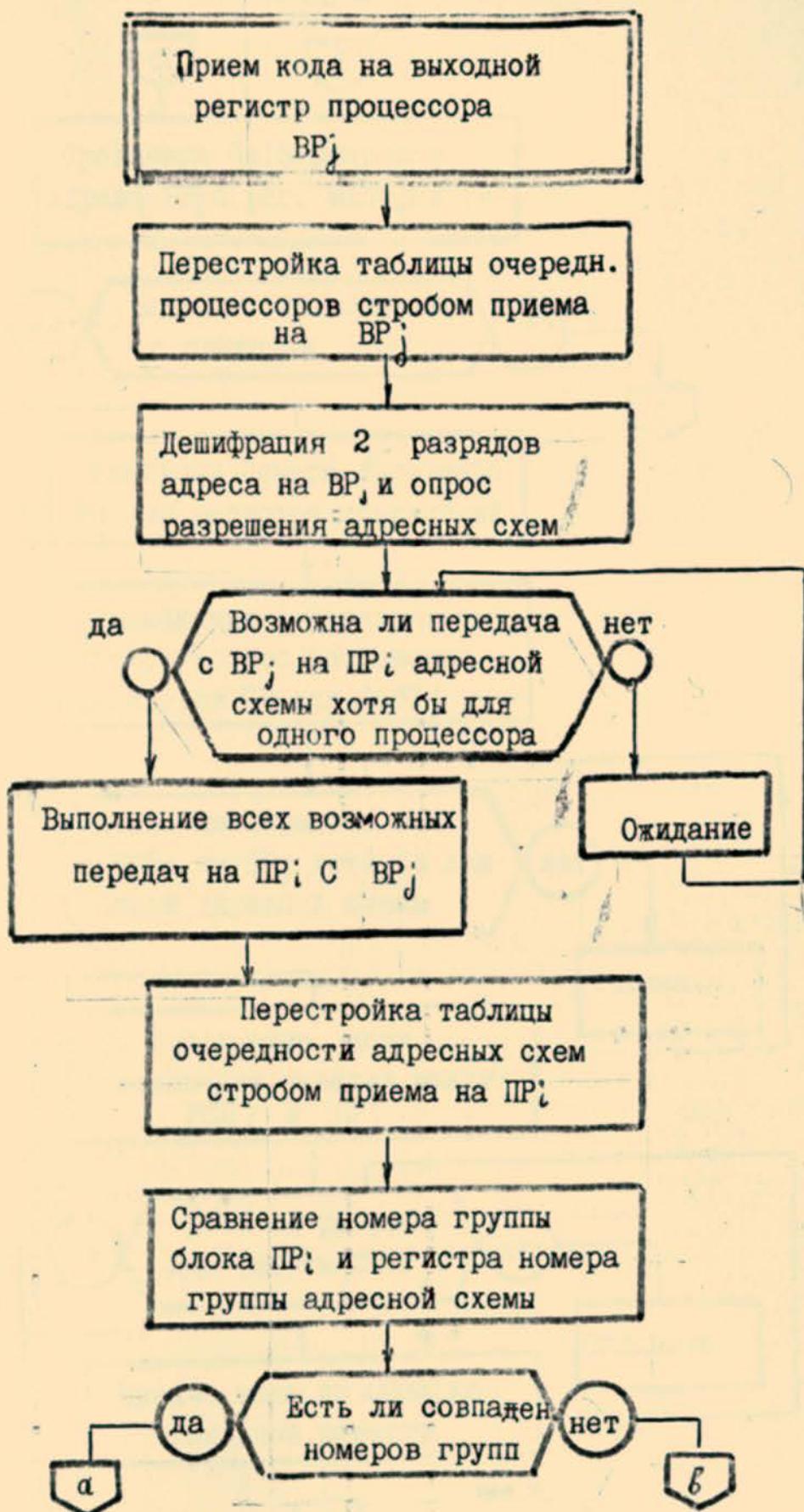


Рис.14. Общая блок-схема алгоритма управления памятью

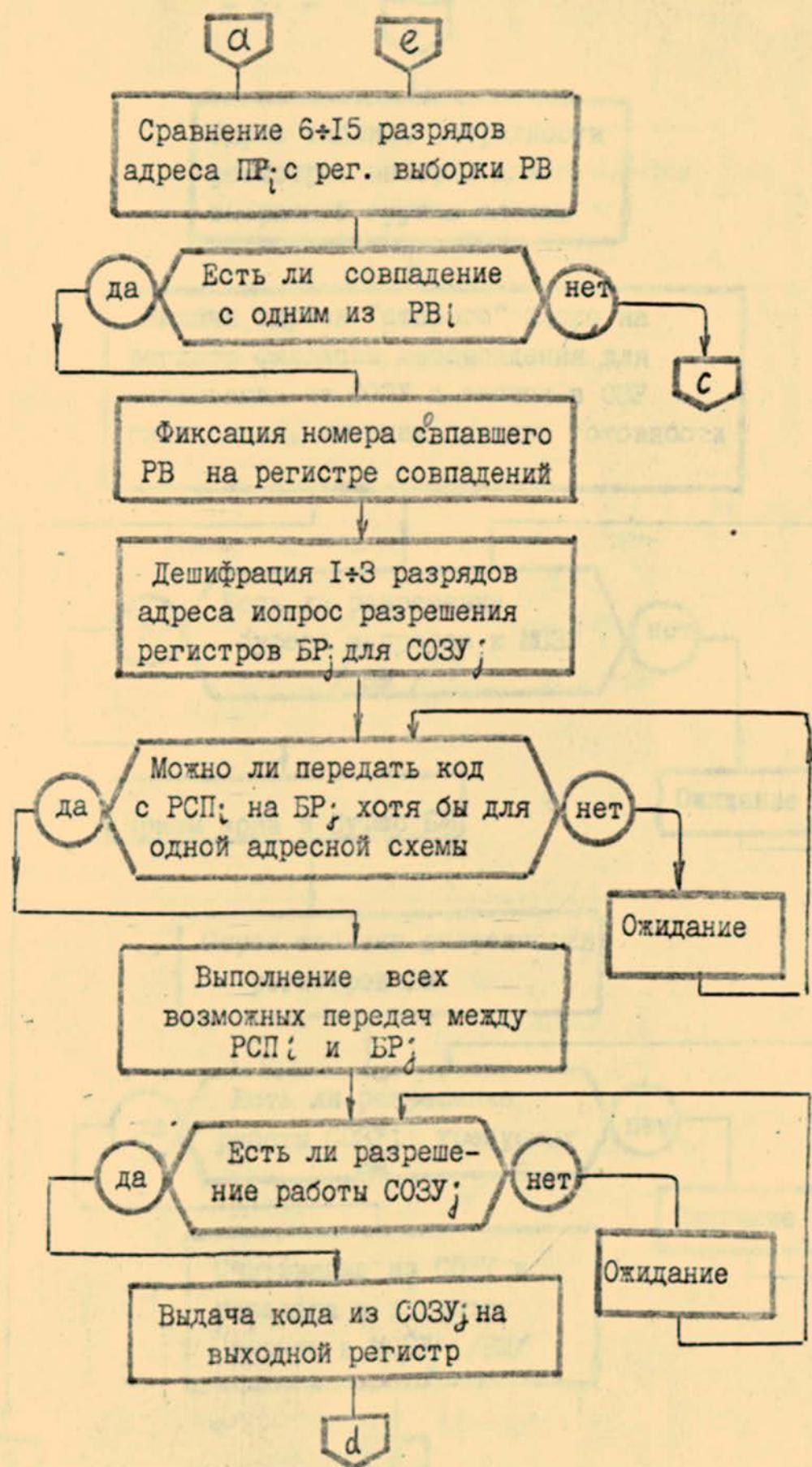


Рис. 15. Общая блок-схема алгоритма  
управления памятью

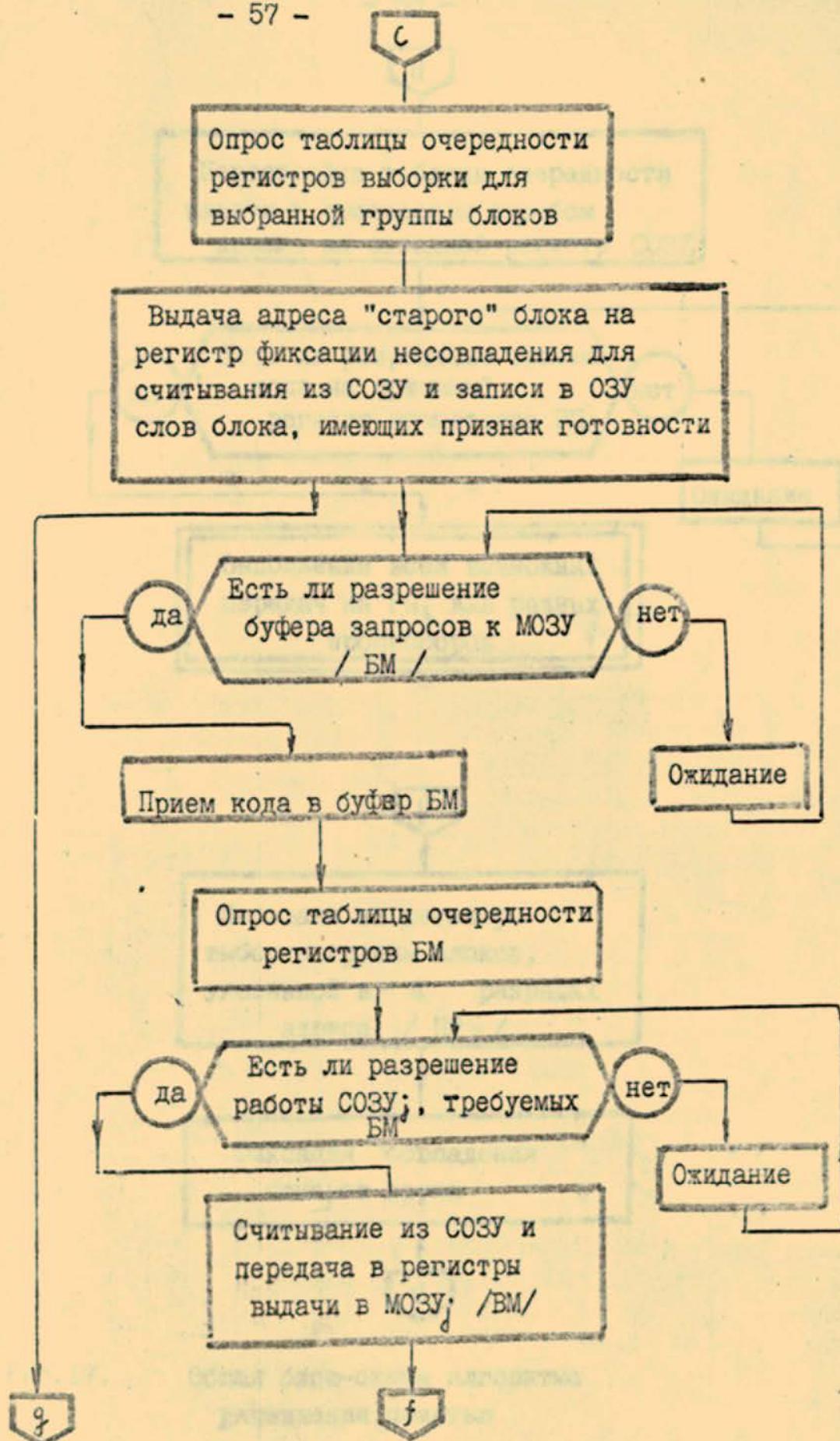


Рис. I6. Общая блок-схема алгоритма управления памятью

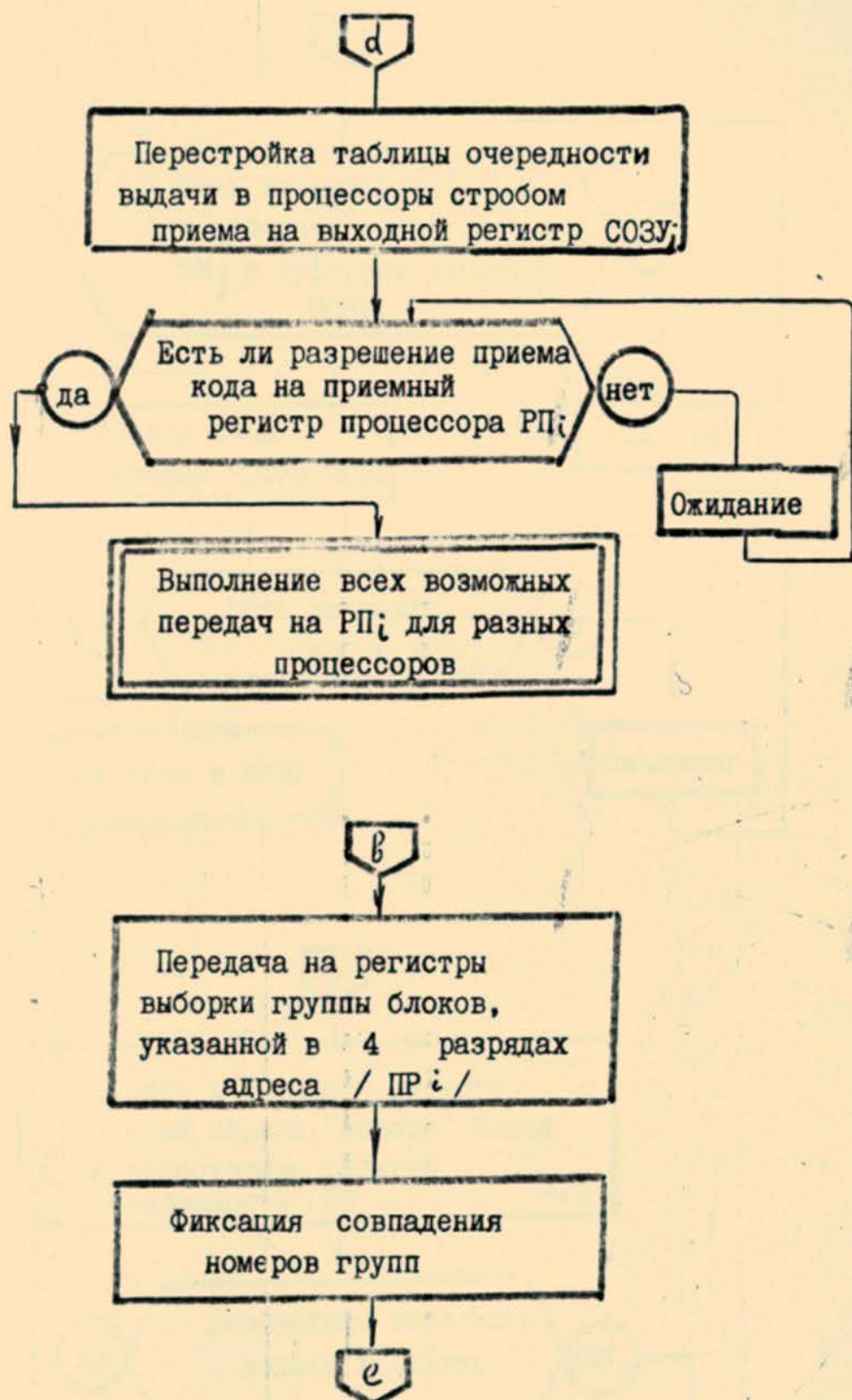


Рис.17. Общая блок-схема алгоритма управления памятью

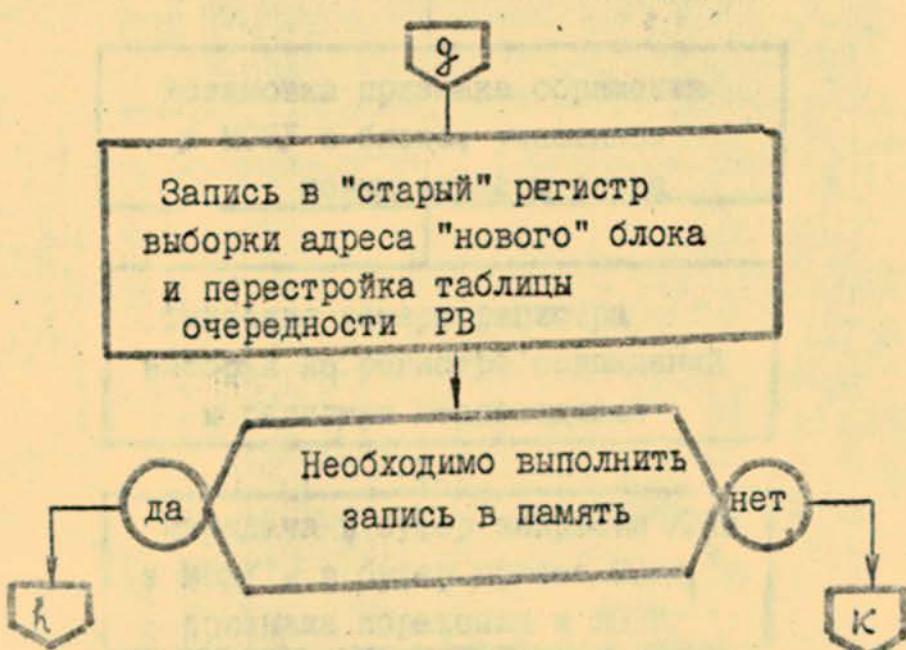
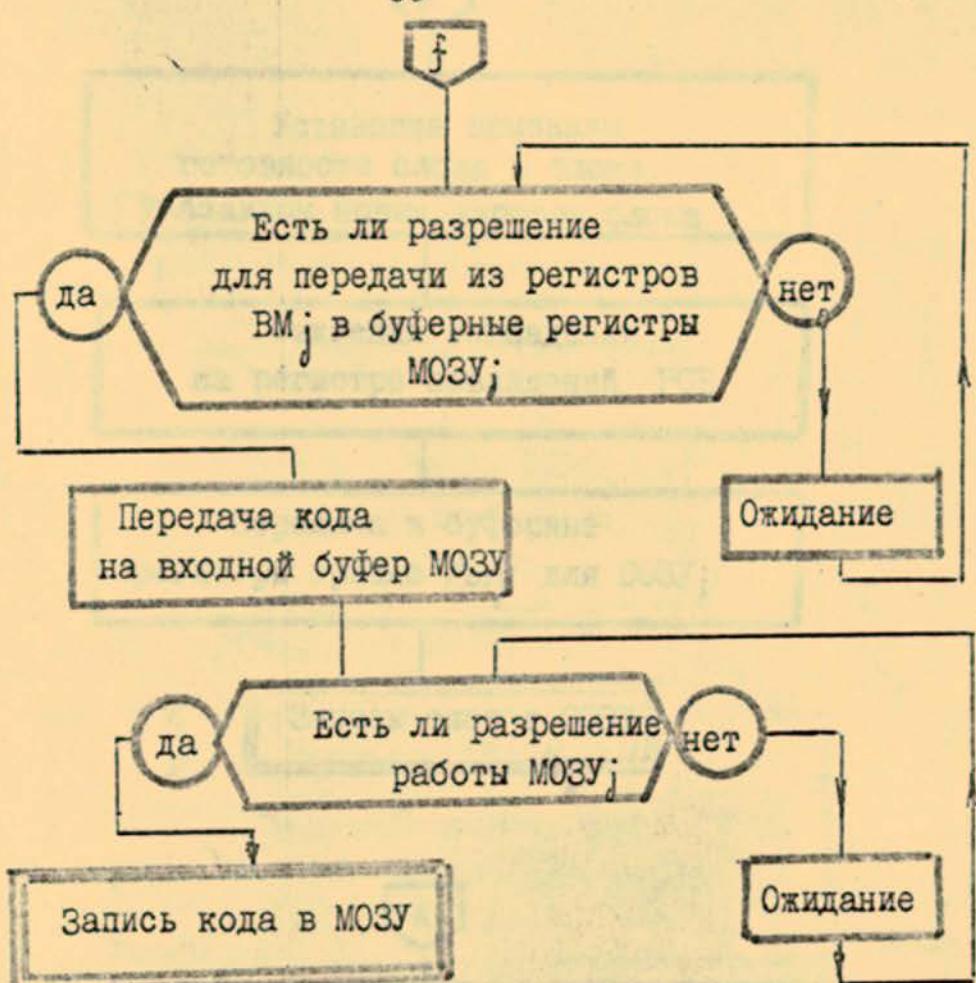


Рис.18. Общая блок-схема алгоритма управления памятью

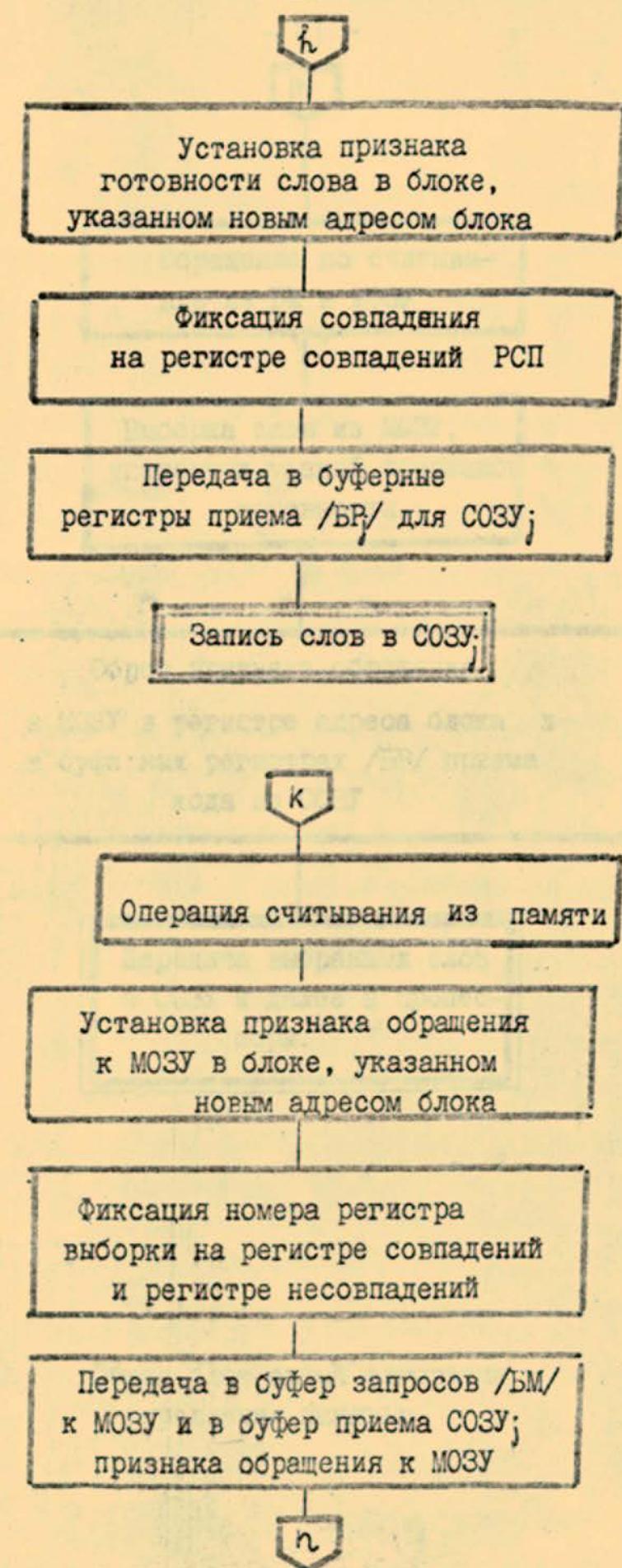


Рис. I9. Общая блок-схема алгоритма управления памятью

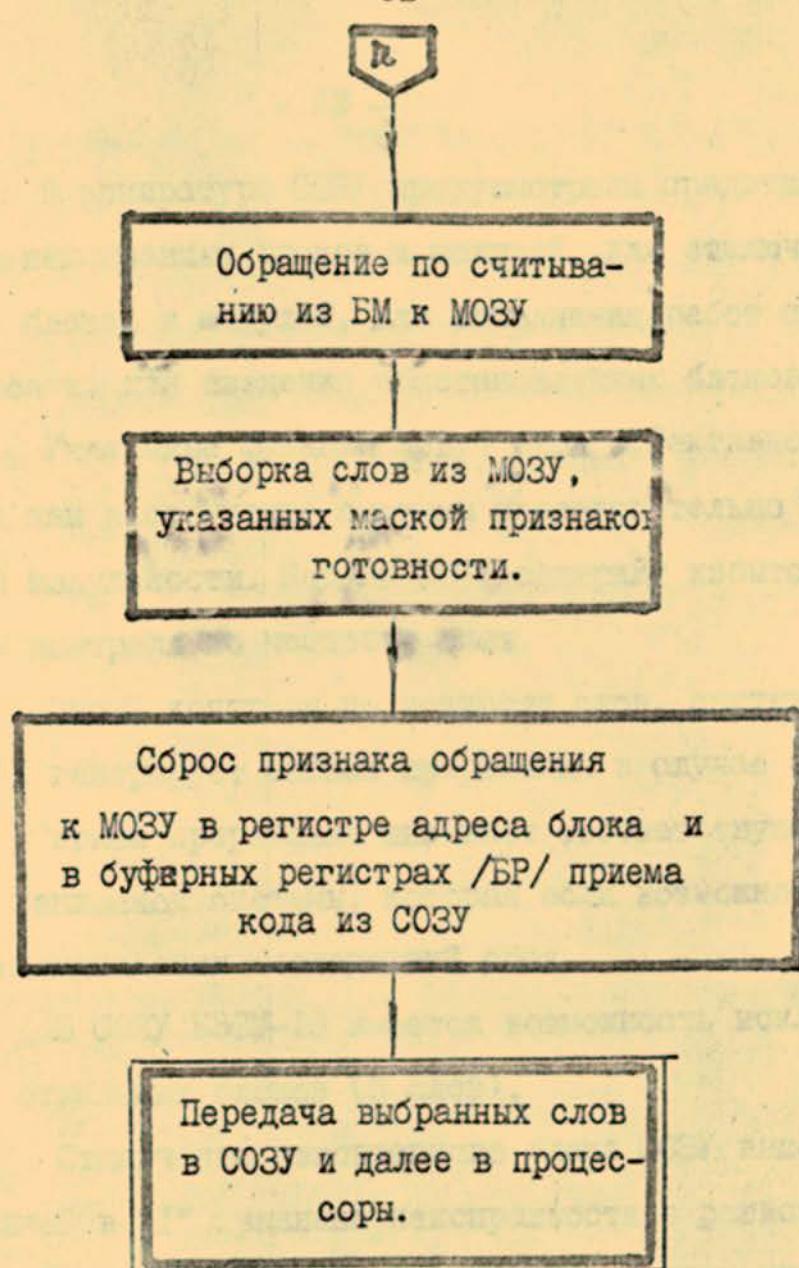


Рис.20. Общая блок-схема алгоритма управления памятью

В аппаратуре СОЗУ предусмотрены средства для обнаружения неисправных блоков и модулей, для отключения неисправных блоков и модулей, для выполнения работ с неполными ресурсами, для введения восстановленных блоков и модулей.

Указанные функции могут быть эффективно реализованы, так как в структуре системы последовательно проводится принцип модульности. Модуль СОЗУ содержит избыточную аппаратуру для контроля по четности слов.

Схема контроля по четности слов, считываемых из модуля СОЗУ, генерирует сигнал прерывания в случае обнаружения ошибки. Сигнал прерывания вызывает соответствующую подпрограмму операционной системы, которая если возможно, примет меры для ликвидации последствий сбоя.

В СОЗУ БЭСМ-10 имеется возможность исключения из работы отдельных блоков (8 слов).

Отключение неисправного блока СОЗУ выполняется установкой в "1" признака неисправности в регистре адреса блока адресной схемы. Емкость СОЗУ при этом уменьшается незначительно (на 8 слов), а пропускная способность сохраняется полностью.

В тех случаях, когда выходит из строя целиком модуль СОЗУ, то отключается 4, 6 или 7 из восьми модулей с соответствующим уменьшением емкости СОЗУ и его пропускной способности.

При обращении к МОЗУ для повышения времени бессбойной работы и увеличения временного интервала между отказами производится коррекция одиночной ошибки и фиксации двойной ошибки при помощи кода Хэмминга. Эта коррекция выполняется

при обмене информацией между ОЗУ и СОЗУ, за счет относительно незначительного замедления эффективного цикла ОЗУ (около 2-3%). Для каждого из 96 разрядов вводится 8 дополнительных контрольных разрядов. При использовании в качестве МОЗУ устройств типа EC 3205 возможно использование в системе 24, 12, 6 и 3 модулей; перестройка конфигурации МОЗУ может производиться программно. Выбор типа устройств памяти будет уточнен на этапе технического проектирования.

## Устройство управления ЦП БЭСМ-10

В структуре центрального процессора и устройства управления БЭСМ-10 нашли дальнейшее развитие те принципы организации выполнения команд, которые имеют место в ЦП АС-6. Основными из этих принципов, которые позволяют добиться высокой производительности центрального процессора, являются:

конвейерная обработка команд;

сокращение длительности полного выполнения отдельной команды;

максимальное распараллеливание выполнения различных команд.

Возросшая по сравнению с ЦП АС-6 производительность БЭСМ-10 связана не только с повышением скорости переключения элементов, но и комплексной реализацией этих принципов.

На рис. 21 приведены блок-схемы, которые дают представление о структуре устройства управления ЦП БЭСМ-10.

На рис. 21 показаны адресуемые индекс-регистры (И) и регистры дескрипторов (РД).

Конвейерная обработка команд осуществляется благодаря последовательному их выполнению в блоке входного регистра команд (РК), в блоке входного регистра (РВ), буфера арифметических команд (БАК); в блоках сумматоров (СА1 и СА2), регистров результатов (РР1 и РР2), проме-

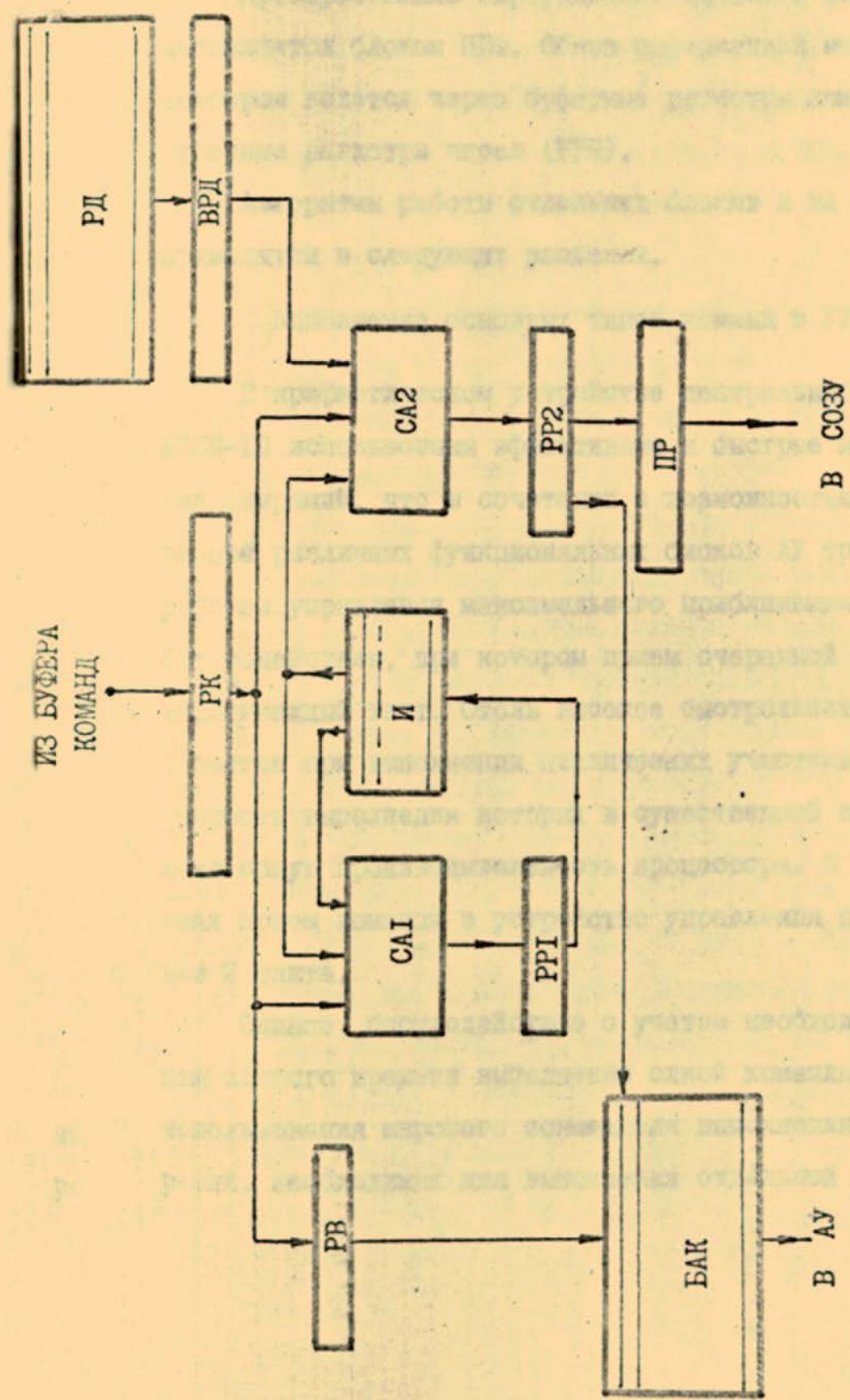


Рис. 21. Блок дешифрации команд и блок сумматоров.

жеточного регистра (ПР).

Преобразование виртуального адреса в физический осуществляется блоком ПВФ. Обмен информацией между СОЗУ и процессором ведется через буферные регистры команд (БРК) и буферные регистры чисел (БРЧ).

Алгоритмы работы отдельных блоков и их взаимодействие приводятся в следующих разделах.

#### Выполнение основных типов команд в УУ БЭСМ-10

В арифметическом устройстве центрального процессора БЭСМ-10 использованы эффективные и быстрые методы выполнения операций, что в сочетании с возможностью параллельной работы различных функциональных блоков АУ требует от устройства управления максимального приближения к данному быстродействию, при котором прием очередной команды происходит каждый такт. Столь высокое быстродействие обеспечивается при выполнении циклических участков программы, скорость выполнения которых в существенной степени определяет общую производительность процессора. В остальных случаях прием команды в устройство управления происходит каждые 2 такта.

Однако, быстродействие с учетом необходимости минимизации полного времени выполнения одной команды потребовало использования широкого совмещения выполнения различных операций, необходимых для выполнения отдельной команды.

Рассмотрим основные узлы устройства управления, алгоритмы их взаимодействия и связи между ними на примере выполнения следующих, наиболее часто используемых типов команд:

- арифметические команды типа регистр-регистр;
- арифметические команды типа регистр-память;
- команды преобразования индексов;
- команды передач управления.

Выполнение арифметической команды типа регистр-регистр в устройстве управления заключается в выяснении типа и приведение ее к стандартному, принятому в буфере арифметических команд, виду. Дешифрация кода команды производится во время нахождения команды на входном регистре команд (РК) и по результату этой дешифрации происходит прием команды на входной регистр (РВ) буфера арифметических команд (БАК).

Перечисленные действия требуют хранения команды на РК в течение 1,5 такта, из которых 1 такт требуется на дешифрацию и 0,5 тайма на организацию передачи команды на РВ.

Выполнение арифметических команд типа регистр-память помимо тех же действий, что и для предыдущего случая, требует вычисления адреса операнда; проверки, если это указано в дескрипторе, нахождения адреса элемента в пределах границ массива; преобразования виртуального адреса в физический и обращения к памяти за операндом. В данном разделе рассмотрим лишь выполнение первых двух действий.

Вычисление исполнительного адреса ведется с помощью трехходового сумматора СА2 (рис.21), на входы которого в этом случае поступают база с выходного регистра дескрипторов (ВРД), индекс из индекс-регистров (И) и адресное смещение, указанное в команде, находящейся на РК. Вычисление исполнительного адреса завершается на регистре результата РР2 через 1,5 такта после начала работы СА2.

Если при вычислении адреса необходима проверка границ, то она производится на сумматоре СА1, на который в этом случае поступают адресное смещение, индекс и значение границы. Одновременное поступление кодов индекса и границы обеспечивается двойной системой выборки из индекс-регистров. Результат проверки границ вырабатывается одновременно с исполнительным адресом операнда.

Двойная система выборки позволяет выполнять за 2 такта и команды преобразования индексов, т.е. освобождение уровня сумматоров происходит через два такта. Результат операции поступает на регистр РР1, с которого происходит запись в соответствующий индекс-регистр. Необходимое для записи кода время не позволяет произвести выборку записываемого индекса непосредственно из регистра для следующей команды в такой момент, который необходим для исключения задержки на РК следующей команды, которой необходим именно этот индекс-регистр. Нежелательная задержка выполнения команд в подобных случаях исключается введением в центральном процессоре БЭСМ-10 цепей, работающих по известному алгоритму общих числовых шин. В

БЭСМ-10 ввиду наличия функционально различных групп адресуемых регистров введены и специализированные общие кодовые шины. В вышеприведенном случае элементом такой цепи являются шины, связывающие регистр РР1 с входными регистрами сумматоров СЛ1 и СЛ2. Благодаря введенной цепи (на рис.21 эта цепь не показана) последовательность связанных по индекс-регистру команд обрабатывается со стандартной скоростью в 3 такта на команду.

Особое внимание уделено повышению скорости выполнения тех участков программы, в которых имеются команды передачи управления. Основная неприятность вызывается тем обстоятельством, что для выборки и подключения команды к РК необходимо дождаться полного выполнения команды передачи управления.

Выполнение команды безусловной передачи управления вызывает максимальную задержку, т.к. дальнейший ход программы целиком определяется адресом перехода. Основной метод ускорения выполнения программы в этом случае - сокращение времени выполнения самой команды передачи управления.

Для условных передач управления допустим (и используется в БЭСМ-10) режим условного выполнения команд, при котором после выполнения на РК передачи управления продолжается прием и частичное выполнение следующих команд. Единственным ограничением является запрещение записи информации в адресуемую память. Если результат проверки условия перехода подтверждает правильность предположенного

направления хода программы, то запрет снимается и условно выполняющиеся команды доварыиваются нормальным образом. В противном случае эти команды гасятся и происходит переход процессора на ветвь, определенную адресом перехода. Гашение условно выполняемых команд или их перевод в тип фактически выполняемых производится тем устройством, условия работы которого проверяются условной передачей управления. Параллельная работа различных функциональных блоков приводит к необходимости наличия в процессоре только одной команды, вырабатывающей опрашиваемые признаки результата. Это достигается тем, что разрешение выработать признаки результата должна иметь только последняя из всех команд, могущих их изменить и находящихся в процессе выполнения в центральном процессоре. Для арифметических команд гашение разрешения выработки признаков результата производится очередной командой, поступающей с РВ в буфер арифметических команд.

Команда "конец цикла" (КЦ) выполняется отличным от других условных передач управления образом ввиду особой важности скорости выполнения циклических участков программы и принципиальной возможности предсказания поведения команды КЦ для следующего ее выполнения. В том случае, если цикл помещается в буферных регистрах команд (БРК), используется дополнительное оборудование, которое позволяет свести выполнение команды КЦ только к времени, необ-

ходимому для изменения индекса (напомним, что благодаря двум сумматорам эта операция совмещается с проверкой границы цикла). Реализация приведенного алгоритма требует хранения информации о местонахождении команды КЦ в БРК, начале команды, на которую происходит передача управления и признака о поведении команды в следующем цикле выполнения. Такая информация позволяет выполнить команду КЦ с необходимым упреждением и, если это оказывается возможным, в такой момент, когда свободен уровень сумматоров. В последнем варианте команда КЦ вообще не тормозит выполнение программы.

Выполнение цикла, помещающегося в БРК, имеет еще одну важную особенность, которая заключается в том, что некоторые типы команд выполняются на регистре команд не за 2 такта, а за 1. Все команды, находящиеся в БРК, при выполнении их в устройстве управления снабжаются признаками, отражающими результат их дешифрации. При повторном приеме этих команд на регистр команд устанавливается режим использования накопленных результатов анализа кода операции, что позволяет полностью исключить время, необходимое для дешифрации на РК команд. В результате, скорость прохождения таких команд на РК возрастает в нашем случае вдвое, и, следовательно, повышается, например, скорость поступления команд в арифметическое устройство. Для системы команд и выбранной структуры центрального процессора БЭСМ-10 объем запоминаемой информации о проведенном анали-

зе кода операции не превышает для наиболее распространенных случаев шести битов на команду.

Выполнение сложных команд в устройстве управления БЭСМ-10 производится по той же методике, какая использована в центральном процессоре АС-6. Напомним, что она основана на разбиении сложных команд на последовательное выполнение базовых подкоманд, каждая из которых возможно максимальным образом использует оборудование центрального процессора. Такой подход позволил добиться высокой эффективности выполнения команд при относительно экономных затратах оборудования.

#### Организация обращения к СОЗУ

Согласование скорости центрального процессора со скоростью вызова информации из СОЗУ заключается в решении проблем вызова команд и обмена операндами.

Командные слова вызываются в буферные регистры команд (БРК) (рис. 22). Минимальное количество этих регистров определяется необходимой глубиной предварительной выборки команд и при тех параметрах СОЗУ и скорости приема команд, которые характерны для БЭСМ-10, может не превышать трех регистров.

Однако, с целью разгрузки канала обращения к СОЗУ и уменьшения влияния задержек по вызову команд в программных циклах объем БРК в БЭСМ-10 будет доведен до 8÷16 регистров с возможностью ассоциативной выборки из них командных слов.

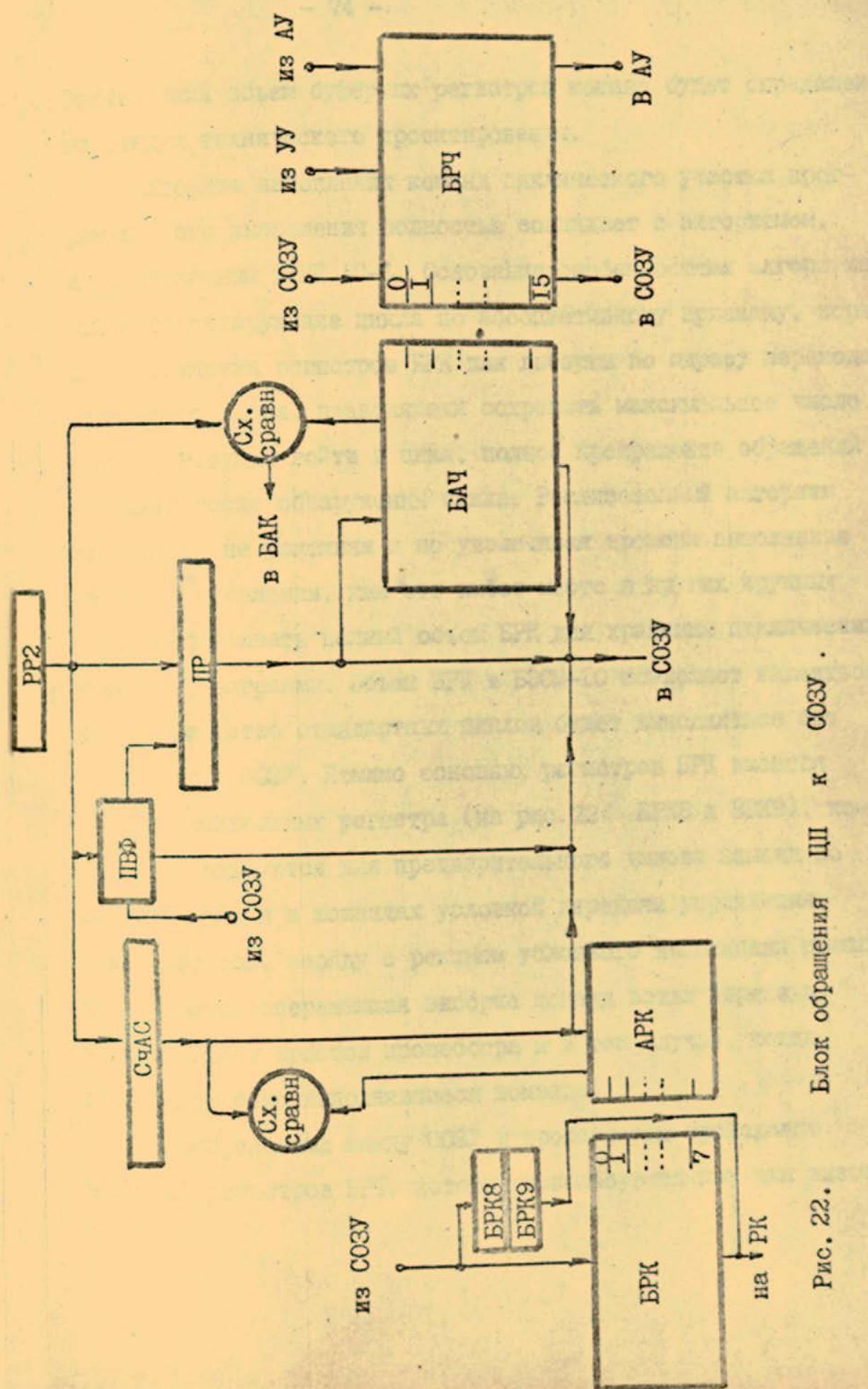


Рис. 22.

Блок обращения III k COZY .

Фактический объем буферных регистров команд будет определен на стадии технического проектирования.

Алгоритм накопления команд циклического участка программы и его выполнения полностью совпадает с алгоритмом, использованным в ЦП АС-6. Основными особенностями алгоритма являются обнаружение цикла по ассоциативному признаку, порядок назначения регистров БРК для выборки по адресу перехода командного слова, позволяющий сохранить максимальное число команд, могущих войти в цикл, полное прекращение обращений к памяти после обнаружения цикла. Реализованный алгоритм позволяет, не усложняя и не увеличивая времени выполнения передач управления, как это имеет место в других крупных ЭВМ, использовать полный объем БРК для хранения циклических участков программы. Объем БРК в БЭСМ-10 позволяет надеяться, что большинство стандартных циклов будет выполняться без обращения к СОЗУ. Помимо основных регистров БРК имеются два дополнительных регистра (на рис.22 БРК8 и БРК9), которые используются для предварительного вызова команд по ветви перехода в командах условной передачи управления. Таким образом, наряду с режимом условного выполнения команд, производится опережающая выборка команд ветви перехода, что сокращает простоя процессора и в том случае, когда гасятся условно выполнявшиеся команды.

Обмен словами между СОЗУ и процессором происходит через 16 регистров БРЧ, которые используются как для вызова

информации из СОЗУ, так и для записи результатов. Каждый из регистров имеет регистр адреса - БАЧ, который хранит полный виртуальный адрес и номер физического листа. Виртуальный адрес является ассоциативным признаком, который позволяет повторное использование информации. Номер физической страницы, который вырабатывает схема преобразования виртуального адреса в физический (ПФ), используется при обращении к СОЗУ.

Обращение к СОЗУ в операциях считывания может происходить непосредственно с регистра ПР, на который поступает физический и виртуальный адрес операнда. Запись этого адреса в соответствующий БАЧ происходит также с регистра ПР. В тех случаях, когда СОЗУ занято, адрес считывания поступает в БАЧ с признаком обращения к памяти. В дальнейшем считывание по требованию БАЧ проходит с приоритетом по отношению к обращениям с ПР.

Запись содержимого БРЧ в память происходит после засыпки кода из АУ в БРЧ. Существуют три регистра записи кода в БРЧ: из СОЗУ, из АУ и из устройства управления. Последний из перечисленных регистров используется для записи в память индексов, дескрипторов и пересылки непосредственно заданных operandов в арифметическое устройство.

Имеется двойная система выборки из БРЧ. Один канал используется для пересылки информации из БРЧ в память в операциях "запись". Выдача кода происходит одновременно с выдачей адреса записи из БАЧ в СОЗУ.

Второй канал выдачи необходим для ускорения пересылки информации из БРЧ в АУ. Выборка осуществляется командой, поступившей из буфера арифметических команд на выходной регистр. Номер БРЧ поступает в БАК либо из схемы, которая определяет порядок заполнения БРЧ, либо из схемы сравнения адресов регистра РР2 с адресами в БАЧ. Работа этих схем синхронизирована с поступлением команды на РР2.

На рис.22 показана условная связь РР2 с БАК, по которой передается номер БРЧ и признак разрешения выполнения команды.

Адрес слова с регистра РР2 поступает в блок преобразования виртуального адреса в физический (ПВФ). В блоке ПВФ имеются регистры, которые предназначены для хранения строк таблицы соответствия виртуальных и физических адресов листов. Виртуальный адрес листа является ассоциативным признаком, по которому производится выборка нужной строки. Ограниченнное количество регистров приводит к проблеме смены содержимого регистров при появлении адреса листа, для которого отсутствует нужная строка таблицы соответствия. В этом случае замене подвергаются самая "старая" по времени последнего использования строка. Вызов нужной строки может потребовать в максимальном случае пять обращений к памяти. Количество регистров, хранящих виртуальные и соответствующие им физические адреса листов, увеличено в БЭСМ-10, по сравнению с ЦП АС-6, до шестнадцати.

Автономная работа ПВФ в том случае, когда не найдена нужная строка таблицы соответствия, позволяет совместить довыполнение накопленных команд с поиском информации.

В схеме управления БРЧ предусмотрены специальные меры, которые позволяют сократить лишние пересылки информации в СОЗУ и организовать параллельную работу функциональных блоков АУ в случае работы с программным стеком. Все записи в программный стек помечаются специальным признаком, который позволяет при считывании из стека этой информации исключить ассоциативный признак в соответствующем регистре БРЧ и отменить запись в СОЗУ. Повторные загрузки в ячейки стека пройдут в другие регистры БРЧ, что позволяет не связывать порядок выполнения команд, работающих с программным стеком, с последовательностью возникновения совпадающих адресов по памяти.

## АРИФМЕТИЧЕСКОЕ УСТРОЙСТВО

Арифметическое устройство /АУ/ центрального процессора БЭСМ-10 предназначено для выполнения арифметических операций и логических операций под словами.

АУ получает команды из устройства управления и выполняет операции, задаваемые этими командами.

Арифметическое устройство центрального процессора БЭСМ-10 предусматривает работу в двух режимах: в системе команд центрального процессора АС-6 и в системе команд ЭВМ БЭСМ-6.

Арифметическое устройство центрального процессора БЭСМ-10 состоит из отдельных независимых функциональных блоков, каждый из которых предназначен для выполнения определенной группы сходных по алгоритмам своего выполнения команд. Такая организация АУ вызывает заметное увеличение оборудования, требуемого для реализации АУ, но это увеличение является достаточно малым на фоне общего увеличения оборудования системы БЭСМ-10, связанного с введением сверхоперативного запоминающего устройства, с резким увеличением объема оперативного запоминающего устройства, с увеличением количества периферийного оборудования.

В то же время введение независимых функциональных блоков позволило значительно повысить производительность АУ.

Повышение производительности АУ обуславливается как одновременным, параллельным выполнением разных команд на разных функциональных блоках, так и уменьшением времени выполнения команды в каждом функциональном блоке.

Специализация функциональных блоков АУ позволила организовать каждый из них наиболее рациональным образом, исходя из конкретных требований выполняемых в нем операций. При этом важно отметить, что специализация функциональных блоков позволяет повысить их быстродействие не только за счет организации наиболее рациональных связей между элементами, но и за счет уменьшения количества связей между элементами (по сравнению с универсальными АУ). Уменьшение количества связей между элементами позволяет расположить элементы так, чтобы длина связей между ними была достаточно мала, что для системы элементов БЭСМ-10, где время прохождения сигналов между элементами сравнимо с задержками внутри элементов, обеспечивает заметное ускорение выполнения операций.

Использование специализированных функциональных блоков позволило уменьшить в АУ центрального процессора БЭСМ-10 (по сравнению с АУ БЭСМ-6 и АС-6) время выполнения основных, наиболее важных команд в заметно большей степени, чем это определяется увеличением быстродействия элементов. Достигнутый выигрыш в производительности вполне оправдывает увеличение оборудования, вызванное отказом от универсального блока выполнения арифметических операций и замену его специализированными блоками.

Детальная разработка схем АУ центрального процессора БЭСМ-10 на стадии эскизного проектирования определила следующие характеристики производительности АУ при выполнении

основных команд:

1. Операции считывания, записи, поразрядные логические операции, операции сдвига, операции над байтовскими и разрядными операндами - 16 млн.команд в секунду.

2. Операции сложения или вычитания чисел с плавающей запятой (при одинарной точности) и с фиксированной запятой и подобные им (вычитание абсолютных значений, обратное вычитание и т.п.) - 16 млн.команд в секунду.

3. Операции умножения чисел с плавающей запятой (при обычной точности) и с фиксированной запятой-

- 16 млн.команд в секунду, если последующая команда не использует результата предыдущей в качестве операнда;
- 6,5 млн.команд в секунду, если последующая команда использует результат предыдущей в качестве операнда.

4. Операция деления чисел с плавающей запятой -

- 1,3 млн.команд в секунду.

5. Операция извлечения квадратного корня-

- 0,6 млн.операции в секунду.

Команды обработки последовательностей байтов -

- 30 млн.байтов в секунду.

Команды обработки последовательностей разрядов -

- 120 млн.разрядов в секунду.

И, наконец, полная суммарная производительность всех функциональных блоков АУ с учетом одновременного параллельного выполнения разных команд в разных функциональных блоках /она определяется пропускной способностью цепей выборки operandов из общих для всех функциональных блоков регистров operandов/ составляет 30 млн. команд в секунду.

### Структура арифметического устройства

Блок-схема арифметического устройства /АУ/ приведена на рис. 23. На блок-схеме показаны:

БАК - буфер арифметических команд

РПО - регистр первого операнда

РВО - регистр второго операнда

РС - регистр слов - адресуемые регистры, указываемые в арифметических командах

БРЧ - буферные регистры чисел

У - блок выполнения операций умножения

Д - блок выполнения операций деления и извлечения квадратного корня

С - блок выполнения операций сложения и вычитания

Л - блок выполнения логических операций

П - блок выполнения операций над последовательностями и над десятичными operandами.

Буфер арифметических команд получает команды из устройства управления и хранит их до тех пор, пока в регистрах РС или БРЧ не появятся operandы, нужные для выполнения

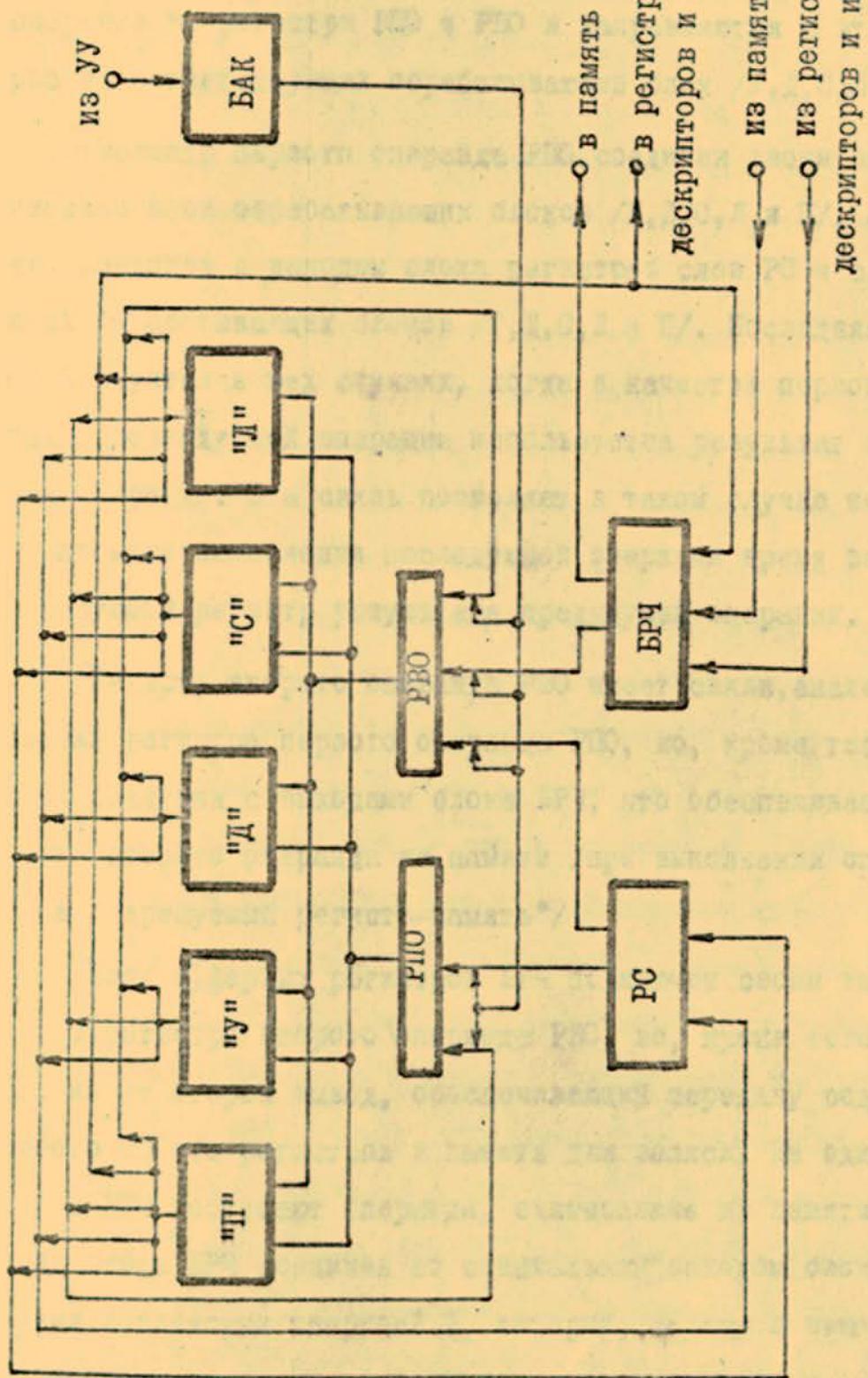


Рис. 23. Блок-схема АУ

очередной команды, после чего одновременно вызываются оба операнда на регистры РЛО и РВО и направляются с этих регистров в соответствующий обрабатывающий блок /У,Д,С,Л или П/.

Регистр первого операнда РЛО соединен своим выходом со входами всех обрабатывающих блоков /У,Д,С,Л и П/, а вход его соединен с выходом блока регистров слов РС и с выходами всех обрабатывающих блоков /У,Д,С,Л и П/. Последняя связь используется в тех случаях, когда в качестве первого операнда последующей операции используется результат предыдущей операции. Эта связь позволяет в таком случае исключить из времени выполнения последующей операции время записи в адресуемый регистр результата предыдущей операции.

Регистр второго операнда РВО имеет связи, аналогичные связям регистра первого операнда РЛО, но, кроме того, его вход соединен с выходами блока БРЧ, что обеспечивает получение второго операнда из памяти /при выполнении операций типа "адресуемый регистр-память"/.

Блок буферных регистров БРЧ подключен своим выходом ко входу регистра второго операнда РВО, но, кроме того, блок БРЧ имеет второй выход, обеспечивающий передачу содержимого любого из его регистров в память для записи. На один вход блока БРЧ поступают операнды, считываемые из памяти, второй вход блока БРЧ соединен со специальным выходом блока выполнения логических операций Л, который, наряду с выполнением логических операций, выполняет и операции записи в память, а на третий вход поступает при необходимости информация из устройства управления /содержимое регистров дескрипторов, индексов и т.п./.

Блок регистров слов РС имеет два выхода - один из них подан на вход регистра первого операнда РПО, а другой - на вход регистра второго операнда РВО. Наличие двух выходов у блока РС обеспечивает одновременную выборку на регистры первого и второго operandов. Количество выходов блока РС также равно двум, что обуславливается относительно низкой пропускной способностью этих входов. Каждый из выходов блока РС выполняет запись в адресуемый регистр за 2 такта и, при одновременном завершении операций во всех пяти обрабатывающих блоках, на запись их результатов в адресуемые регистры при одном входе блока РС потребовалось бы 10 тактов, что вызвало бы задержку вычислений. Введение второго входа уменьшает это время до пяти тактов, обеспечивая заметно лучшее согласование пропускной способности блока РС и обрабатывающих блоков. Выходы каждого из блоков Л, С. и П подключены к обоим входам блока АР, выходы же блоков Д и У подключены каждый только к одному из входов блока РС. Для блока выполнения операций деления Д это использование одного входа обусловлено тем, что операции деления являются сравнительно редкими операциями и незначительное замедление их выполнения не вызывает сколько-нибудь заметного снижения производительности АУ. Для блока выполнения операций умножения У использование только одного входа блока РС обуславливается тем, что этому блоку дан максимальный приоритет по использованию соответствующего входа блока АР /причины этого указаны ниже, при более подробном описании блока У/и, поэтому, подключение его ко второму входу блока РС не нужно.

### Буфер арифметических команд

Буфер арифметических команд получает команды из устройства управления и хранит их до тех пор, пока не будут готовы оба операнда команды, после чего вызывает эти операнды на регистры первого и второго operandов РНО и РВО и направляет их в соответствующее обрабатывающее устройство. Для повышения производительности АУ выход БАК"а обеспечивает возможность обгона: если предыдущая команда не может начать выполняться в следствие отсутствия одного из operandов, а для последующей команды оба операнда уже есть, то последующая команда начинает выполняться раньше предыдущей. В тех же случаях, когда operandы готовы для нескольких команд сразу, эти команды выполняются в том порядке, в каком они поступили в БАК.

Бак содержит:

- а) Восемь регистров хранения команд, именуемых в дальнейшем линейками БАК"а и нумеруемых числами от 0 до 7.
- б) Таблицу очередности линеек БАК"а, фиксирующую очередь заполнения линеек командами, поступающими из устройства управления.
- в) Выходной регистр БАК"а.
- г) Усилители выборки operandов, управляющие вентилями передачи содержимого одного из РС на регистр РНО и одного из РС или БРЧ на регистр РВО.
- д) Вспомогательные усилители, фиксирующие готовность operandов для каждой из линеек БАК"а и обеспечивающие надлежащий порядок работы усилителей выборки operandов.

Каждая линейка БАКа содержит:

- а/ восемь двоичных разрядов кода операции команды;
- б/ один двоичный разряд блокировки записи, наличие единицы, в котором означает, что команда является проверочной, т.е. что после её выполнения устанавливаются лишь признаки результата, а сам результат никуда не записывается;
- в/ адрес первого операнда, состоящий из трех двоичных разрядов номера регистра слова РС, содержащего первый операнд, из трех двоичных разрядов номера байта /при операциях над байтами/ и трех двоичных разрядов номера разряда /при операциях над разрядами/;
- г/ адрес второго операнда, состоящий из пяти двоичных разрядов, задающих нахождение второго операнда в блоке РС или блоке БРЧ и задающих номер регистра в соответствующем блоке, из трех двоичных разрядов номера байта /при операциях над байтами/ и трех двоичных разрядов номера разряда /при операциях над разрядами/;
- д/ признак наличия команды в данной линейке БАКа, устанавливаемый в состояние единицы при приеме в данную линейку команды из устройства управления и сбрасываемый в состояние нуля при вызове operandов этой команды на регистры первого и второго operandов РПО и РВО.

Максимальный темп поступления команд в БАК определяется пропускной способностью устройства управления. Команда из устройства управления поступает в БАК при наличии в БАКе, хотя бы одной свободной линейки.

При наличии в БАКе нескольких свободных линеек, они заполняются командами в порядке возрастания номеров этих линеек. Одновременно с приемом команды в линейку БАКа производится переключение таблицы очередности линеек БАКа. Переключение таблицы производится так, что линейка, в которую производится прием команды, становится самой "молодой" из всех линеек БАКа, а взаимное "старшинство" остальных линеек БАКа сохраняется неизменным.

Каждой линейке БАКа поставлены в соответствие следующие вспомогательные усилители:

а/ усилитель готовности операции, устанавливающийся в состояние единицы, если соответствующий коду операции обрабатывающий блок АУ /блок У,Д,С,Л и П/ будет свободен от выполнения ранее запущенных команд к тому времени, когда он может потребоваться /с учетом задержки в цепях выборки операндов/ для выполнения команды, находящейся в данной линейке БАКа. Возможность судить об освобождении обрабатывающего блока АУ от выполняемой в нём команды с необходимым опережением обеспечивается тем, что все обрабатывающие блоки АУ выполнены так, что время выполнения в них любой команды не зависит от значения операндов этой команды /не зависит, например, от количества нормализаций мантиссы результата влево при выполнении команд над операндами с плавающей запятой/ и может быть, поэтому, предсказано заранее;

б/ усилитель блокировки первого адреса, устанавливающийся в состояние единицы, если в одном из обрабатывающих блоков АУ выполняется команда, результат которой должен

быть записан в тот адресуемый регистр, номер которого указан в первом адресе данной линейки БАКа, причем результат на выходе этого обрабатывающего блока АУ не будет готов к тому, времени, когда он может потребоваться /с учетом задержки в цепях выборки операндов/ для выполнения команды, находящейся в данной линейке БАКа;

в/ усилитель блокировки второго адреса, устанавливающийся в состояние единицы аналогично усилителю блокировки первого адреса в тех случаях, когда второй operand команды, находящийся в данной линейке БАКа, согласно её второму адресу, должен быть взят из одного из адресуемых регистров;

г/ усилитель готовности второго адреса, устанавливающийся в состояние единицы в тех случаях, когда второй operand команды, находящийся в данной линейке БАКа, должен быть взят согласно ее второму адресу из одного из регистров блока БРЧ и когда усилитель готовности этого регистра БРЧ находится в состоянии единицы. Следует отметить, что вследствие задержки в цепях выборки операндов усилители готовности регистров БРЧ устанавливаются в состояние единицы с определенным опережением по отношению к моменту реального появления операнда на соответствующем регистре;

д/ усилитель готовности линейки БАКа, устанавливающийся в состояние единицы, если для данной линейки БАКа усилитель признака наличия команды и усилитель готовности операции находятся в состоянии единицы, усилитель блокировки первого адреса находится в состоянии нуля и если либо усилитель блокировки второго адреса находится в состоянии нуля, либо усилитель готовности второго адреса находится в состоянии единицы.

В каждом такте может сработать усилитель выдачи только одной линейки БАК"а, однако, уже в следующем такте может сработать усилитель выдачи другой линейки БАК"а и, следовательно, максимальный темп выдачи команд из БАК"а равен одной команде за один такт.

Выходной регистр БАК"а имеет ту же разрядность и структуру информации, что и линейки БАК"а. После приема на выходной регистр БАК"а очередной команды в соответствии с ее адресами срабатывают усилители выборки операндов, управляющие вентилями передачи содержимого одного из РС на регистр первого операнда РПО и одного из РС или БРЧ на регистр второго операнда РВО. При этом, если необходимый для выполнения последующей команды результат предыдущей команды не успел еще записаться в соответствующий регистр РС с выхода одного из обрабатывающих блоков У,Д,С,Л и П, то операнд на соответствующий регистр РПО или РВО берется непосредственно с выхода соответствующего обрабатывающего блока. Это обеспечивается наличием специальной схемы сравнения номеров регистров слов и позволяет исключить из времени выполнения операций время записи результата в регистры слов.

Операнды, выбранные на регистры РПО и РВО, поступают затем в тот обрабатывающий блок У,Д,С,Л или П, в котором должна выполняться данная команда. Одновременно в этот же обрабатывающий блок поступит с выходного регистра БАК"а код операции данной команды и та часть адресной информации команды, кото-

рая необходима для ее выполнения и для последующей записи результата в соответствующий регистр слова РС (и при записи в память в соответствующий буферный регистр БРЧ).

Регистры слов РС предназначены для хранения промежуточных результатов вычислений и непосредственно указываются в командах, т.е. предназначены для выполнения тех же функций, что и в центральном процессоре системы АС-6. Вследствие полного совпадения систем команд АС-6 и БЭСМ-10 количество регистров слов РС в БЭСМ-10 равно 8 при нумерации от 0 до 7 и при разрядности каждого регистра 48 информационных разрядов и 6 контрольных разрядов по одному для каждого байта.

В блоке РС предусмотрено два тактированных входных регистра и два комплекта входных вентилей хранящих регистров, слов, что, как указывалось выше, позволяет удвоить пропускную способность блока РС при записи в него результатов с выходов обрабатывающих блоков. Пропускная способность каждого из входных регистров равна одной записи за два такта, следовательно, общая пропускная способность блока РС по записи равна одной записи за такт, что хорошо согласуется с пропускной способностью выходных цепей буфера арифметических команд, равной одной команде за такт.

Выходные вентили хранящих регистров слов в блоке РС также выполнены в двух комплектах. Один комплект предназначается для выборки первого операнда на регистр первого операнда РПО, второй - для выборки второго операнда на регистр второго операнда РВО. Наличие двух комплектов выходных вентилей блока РС обеспечивает пропускную способность этого блока по считыванию операндов так же равную одной команде за такт.

### Блок буферных регистров чисел

Буферные регистры чисел предназначены для буферизации обмена числами между АУ и сверхоперативным запоминающим устройством.

Блок буферных регистров чисел (БРЧ) содержит 16 хранящих регистров, каждый из которых имеет 48 информационных разрядов и 6 контрольных разрядов. Каждый из буферных регистров чисел может быть использован и в качестве буферного регистра записи (при выполнении записи чисел из АУ в СОЗУ) и в качестве буферного регистра чтения (при считывании операндов из СОЗУ). Каждому буферному регистру блока БРЧ в устройстве управления центрального процессора БЭСМ-10 поставлен в соответствие адресный регистр, содержащий адрес по оперативной памяти ОЗУ той ячейки ОЗУ, содержимому которого соответствует содержимое данного регистра БРЧ. Таким образом, БРЧ наряду с выполнением функции буферизации между АУ и СОЗУ играют и роль ассоциативной сверхпамяти и обеспечивают уменьшение количества обменов числами между АУ и СОЗУ. Кроме того, БРЧ используются также в качестве буфера между устройством управления центрального процессора и его АУ в тех случаях, когда в качестве операнда арифметической команды используется содержимое индексного регистра или регистра дескриптора или когда operand для арифметической операции задан непосредственно в коде команды и должен поступать в АУ с регистра команд устройства управления.

В блоке БРЧ предусмотрено три входных тактированных регистра и три комплекта входных вентилей на каждом из хранящих регистров. Один из входных регистров блока БРЧ связан с выходом блока выполнения логических операций Л, выполняющим наряду с логическими операциями, и операции записи информации в память. Через этот входной регистр число, подлежащее записи в память, поступает в соответствующий хранящий буферный регистр БРЧ, выполняющий в данный момент функции буфера записи. К этому же входу блока БРЧ подключен и выход блока П, что обеспечивает выполнение команд, требующих записи в память последовательностей байтов или разрядов и десятичных чисел.

На второй из входных регистров блока БРЧ поступают числа, считываемые из СОЗУ, а на третий — содержимое индексных или дескрипторных регистров УУ или операнды, заданные непосредственно в коде команды. Наличие в блоке БРЧ трех входных регистров определяется не только стремлением к увеличению пропускной способности этого блока, но и тем, что разделение потоков записей в буферные регистры БРЧ заметно упрощает и облегчает согласование управления этими потоками.

В блоке БРЧ предусмотрено два комплекта выходных вентилей хранящих буферных регистров. Один из этих комплектов обеспечивает вызов содержимого буферного регистра на регистр второго операнда РВО, второй — пересылку содержимого буферных регистров в цепи записи СОЗУ. Наличие двух комплектов выходных вентилей исключает влияние отсылок содержимого ре-

гистров БРЧ в СОЗУ на выборку операнда для очередной арифметической команды. Это обеспечивает максимальный темп выборки операндов из блока БРЧ, равный максимальному темпу блоков БАК и РС, т.е. выборку одного операнда за один такт. Кроме того, наличие двух комплектов выходных вентилей максимально развязывает работу управления выборкой операндов для арифметических команд от управления записью содержимого буферных регистров БРЧ в СОЗУ.

Более подробное описание цепей связи блока БРЧ с СОЗУ и УУ, описание управления этими связями и алгоритмов их работы даны в разделе, посвященном устройству управления.

#### Регистры первого и второго operandов

Регистр первого операнда РПО и регистр второго операнда РВО являются, по существу, выходными регистрами блоков РС и БРЧ, в которых и расположена основная масса входных вентилей регистров РПО и РВО. Непосредственно на входах регистров РПО и РВО помещены лишь вентили приема результатов с выхода обрабатывающих блоков У,Д,С,Л и П, предназначенные, как было описано выше, для ускорения выполнения команд, использующих в качестве operandов результаты предыдущих команд.

Оба регистра РПО и РВО выполнены тактированными и, так как operandы каждой очередной команды принимаются на них лишь тогда, когда имеются в наличии оба операнда и соответствующий сбрасывающий блок У,Д,С,Л или П свободен от выполнения предыдущей команды, то operandы каждой очередной команды записывают регистры РПО и РВО только в течение одного такта. Это обеспечивает максимальную пропускную способность регистров РПО и РВО равную одной команде за один такт.

Выходы каждого из регистров РПО и РВО поступают на входы каждого из обрабатывающих блоков У, Д, С, Л и П. Следует отметить, что в большинстве обрабатывающих блоков прием операндов с выхода регистров РПО и РВО совмещен (с целью ускорения получения результата) с некоторой начальной (различной для разных обрабатывающих блоков) обработкой этих операндов.

#### Блок выполнения логических операций

Блок выполнения логических операций Л выполняют команды, задающие следующие операции:

а) поразрядные логические операции над словными и полусловными операндами — логическое сложение ЛСС и ЛСП, логическое умножение ЛУС и ЛУП, логическая нетождественность ("исключающее ИЛИ") ЛНС и ЛНП, логическая проверка ("исключающее ИЛИ" без записи результата, с выработкой лишь признаков результата) ЛПС и ЛПП, установка зеркальная словная УЗС;

б) операции записи и установки для словных и полусловных операндов — запись слова ЗС, запись полуслова ЗП, запись магазинная ЗМ, установка слова УС, установка полуслова УП, установка магазинная ЮМ, запись нуля в слово ЗНС, запись нуля в полуслово ЗНП, запись индексного регистра ЗИ;

в) операции над операндами с плавающей запятой, связанные с изменением лишь знака мантиссы — взятие абсолютного значения числа с плавающей запятой АЗЕ и изменение знака числа с плавающей запятой на противоположный ИЗЕ;

г/ все операции над байтовскими операндами, включая и операции сложения и вычитания байтов;

д/ все операции над разрядными операндами.

Блок-схема блока выполнения логических операций Л дана на рис. 24.

Регистры 1-5 являются входными. На регистр 1 поступает с выходного регистра БАКа код операции и все остальные сигналы, необходимые в блоке Л для выполнения очередной команды. Этот регистр связан с усилителями, управляющими всеми вентильными схемами блока Л так, чтобы обеспечить выполнение заданной команды /на блок-схеме эти усилители и их связи не показаны/.

Регистр 2 имеет 48 разрядов и предназначен для хранения первого операнда, поступающего на него с выхода регистра первого операнда РПО. Регистр 3 также имеет 48 разрядов и предназначен для хранения второго операнда, поступающего на него с выхода регистра второго операнда РВО. При этом левая половина регистра 3 связана только с левой половиной регистра РПО, а правая половина регистра 3 связана и с правой и с левой половиной регистра РВО, что позволяет при выполнении операций с полусловными операндами принимать нужную половину регистра РВО на правую половину регистра 3, оставляя левую половину регистра 3 пустой.

Регистры 4 и 5 имеют по 8 разрядов и предназначены для хранения байтов первого и второго операндов соответственно при выполнении операций над байтовскими и разрядными операндами.

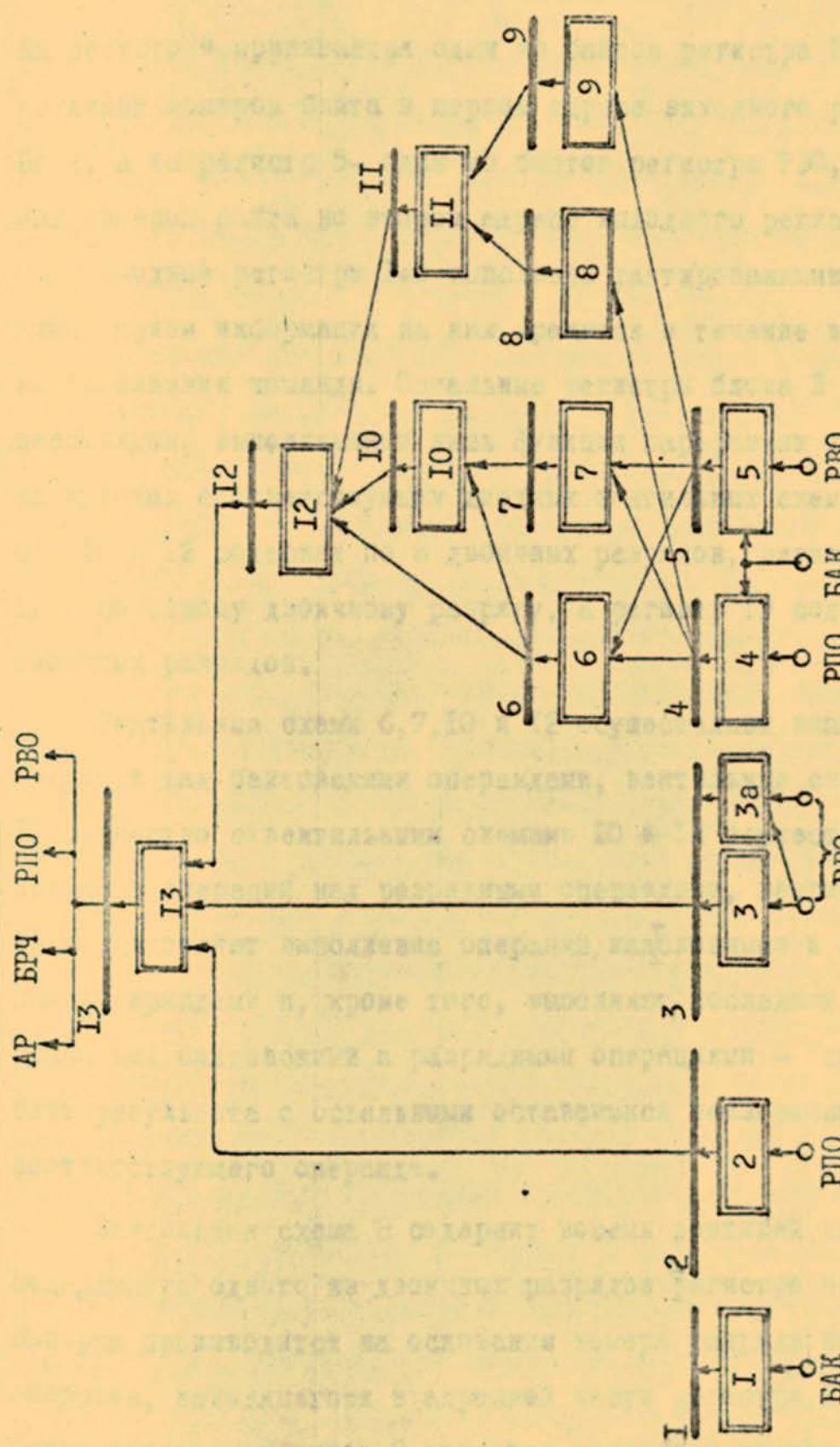


Рис. 24. Блок-схема блока "II".

На регистр 4 принимается один из байтов регистра РНО, определяемый номером байта в первом адресе выходного регистра БАКа, а на регистр 5- один из байтов регистра РВО, определяемый номером байта во втором адресе выходного регистра БАКа.

Входные регистры I+5 выполнены тактированными и хранящими, прием информации на них хранится в течение всего времени выполнения команды. Остальные регистры блока Л являются пассивными, выполняющими лишь функции паразитных усилителей на выходах соответствующих диодных вентильных схем. Регистры 6,7,10 и 12 содержат по 8 двоичных разрядов, регистры 8,9 и II - по одному двоичному разряду, а регистр 13 содержит 48 двоичных разрядов.

Вентильные схемы 6,7,10 и 12 осуществляют выполнение операций над байтовскими operandами, вентильные схемы 8,9 и II совместно с вентильными схемами 10 и 12 осуществляют выполнение операций над разрядными operandами, вентильные схемы 13 осуществляют выполнение операций над словными и полусловными operandами и, кроме того, выполняют последний этап операций над байтовскими и разрядными операциями - "склеивают" байт результата с остальными оставшимися неизменными байтами соответствующего operandса.

Вентильная схема 8 содержит восемь вентилей выборки содержащего одного из двоичных разрядов регистра 4. Выборка производится на основании номера разряда первого operandса, находящегося в адресной части регистра I. Таким образом, регистр 8 является регистром первого operandса при операциях над разрядными operandами.

Регистр 9 является регистром второго операнда при операциях над разрядными operandами. Его входная вентильная схема 9 содержит восемь вентилей выборки содержимого одного из двоичных разрядов регистра 5, управляемых номером разряда второго операнда адресной части регистра I.

Регистр II является регистром разрядного результата при выполнении операций над разрядными operandами. Его входная вентильная схема II содержит 16 вентилей, управляемых разрядами кода операции регистра I, и обеспечивает при выполнении операций над разрядными operandами получение на регистре II булевой функции содержимого регистров 8 и 9, соответствующей выполняемой команде. При выполнении операций над байтовскими, полусловными и словными operandами вентильная схема II обеспечивает нулевое состояние регистра II. После получения разрядного результата для выполнения дальнейших этапов выполнения операции над разрядными operandами используется байтовское и словное оборудование, описанное ниже.

Вентильные схемы 6 содержат в каждом разряде 4 вентиля, обеспечивающие реализацию четырех поразрядных булевых функций /4.5, 4.5, 4.5 и 4.5/. Включение этих вентилей в определенных сочетаниях позволяет получить на регистре 6 результат любой байтовской поразрядной логической операции или поразрядную сумму при выполнении операций сложения или вычитания байтов, а также позволяет передать на регистр 6 содержимое одного из регистров 4 или 5 без какого-либо изменения / что используется при выполнении операций считывания

или записи байтов и при выполнении операций над разрядными операндами/.

Вентильные схемы 7 используются только при выполнении операций сложения и вычитания байтов и содержат в каждом разряде два вентиля: вентиль поразрядного переноса при сложении байтов /4.5/ и вентиль поразрядного переноса при вычитании байтов /4.5/.

Вентильные схемы 10 используются при выполнении операций сложения и вычитания байтов, для чего содержат систему вентилей одношагового распространения переносов. Эта система вентилей, содержащая в каждом разряде число вентилей, равное номеру этого разряда /от одного вентиля в первом разряде до 8 вентилей в восьмом разряде/ устанавливает на регистре 10 код распространенного переноса, соответствующего поразрядным переносам регистра 5 и поразрядным суммам регистра 4. Кроме того в вентильной схеме 10 предусмотрены вентили, обеспечивающие выполнение операций над разрядными операндами, а именно - в каждом разряде имеется вентиль, управляемый соответствующим адресным полем регистра 1 и устанавливающий в состояние единицы тот разряд регистра 10, номер которого соответствует номеру разряда, в который должен быть записан результат операции над разрядными операндами.

Регистр 12 является байтовским регистром результата для операций над байтовскими и разрядными операндами.

Его входная вентильная схема I2 содержит в каждом разряде три вентиля: два вентиля поразрядного суммирования регистров 6 и 10 (вентили 6·10 и 6·10) и один вентиль заноса разрядного результата с регистра II в тот разряд регистра I2, который задан содержимым поля адресной части регистра I, определяющего разряд, в который должен быть записан результат операции над разрядными operandами. При выполнении операций над разрядными operandами из двух вентилей поразрядного суммирования открывается лишь второй (6·10), что обеспечивает передачу с регистра 6 на регистр I2 лишь неизменяемых разрядов байта, а в изменяемый разряд заносится разрядный результат с регистра II.

И, наконец, регистр I3 является словным регистром результата для всего блока выполнения логических операций Л. Его входная вентильная схема содержит в каждом разряде 5 вентилей: три вентиля, реализующие три булевские функции от содержимого соответствующих разрядов регистров 2 и 3 (2·3, 2·3 и 2·3), вентиль зеркального приема содержимого регистра 3 (в I-ый разряд принимается из 48, во 2-ой - из 47-го и т.д.) и вентиль заноса содержимого соответствующего разряда байтовского регистра результата I2. Следует отметить, что управление первыми тремя из этих вентилей выполнено раздельно для левой

и правой половины регистра I3, что обеспечивает выполнение логических поразрядных операций над правой половиной слова при сохранении неизменной его левой половиной. Кроме того, в эти же три вентиля введены и "нулевые" выходы усилителей, управляющих вентилями заноса содержимого байтовского регистра результата I2 в разряды соответствующего байта регистра I3. Это обеспечивает при выполнении операций над байтовскими или разрядными операндами запрет передачи с регистра 2 или регистра 3 старого значения изменяемого байта, новое значение которого находится на регистре I2, разрешая передачу лишь остальных, не изменяемых байтов.

С выхода регистра I3 результат операции поступает в блоки АР, БРЧ, РНО и РВО, где соответствующие схемы управления обеспечивают его надлежащее использование.

Как видно из блок-схемы блока выполнения логических операций Л время выполнения операции в этом блоке определяется операциями над байтовскими и разрядными операндами, при которых информация, принятая на входные регистры I+ 5, достигает выхода выходного регистра I3 после прохождения четырех последовательных вентильно-усилительных уровней. Это определяет пропускную способ-

ность блока Л в два такта на одну команду. Из этих двух тактов один полутакт команды занимает регистры РПО и РВО и три полутакта — входные регистры I+5. Так как задержка каждого пассивного вентильно-усилительного уровня меньше четверти такта, то результат на выходе регистра I3 появляется через два полутакта и следовательно, при двухтактном цикле будет иметь длительность в один полутакт, что более чем достаточно для его передачи по месту назначения.

Операции над словными и полусловными операндами выполняются в блоке Л заметно быстрее, однако, возможность увеличить пропускную способность блока Л при выполнении таких команд не используется, так как это приводит к заметным усложнениям управления записью результатов в блоках АР и БРЧ и управления выборкой очередной команды из БАК"а.

Таким образом, пропускная способность блока Л равна одной команде за два такта для любой из выполняемых в этом блоке команд.

Кроме блоков, показанных на блок-схеме, в состав блока выполнения логических операций Л входит еще добавочное оборудование, связанное с выполнением аппаратного

контроля работы блока Л и с выработкой признаков результата. Оборудование аппаратного контроля состоит из входного оборудования, осуществляющего проверку четности всей информации на входных регистрах I-5, и выходного оборудования, осуществляющего формирование контрольных байтовских разрядов регистра I3. Контрольное оборудование блока Л построено так, что обеспечивает фиксацию любого одиночного сбоя в цепях блока Л.

Выработка признаков результата выполняется в блоке Л в зависимости от режима работы в соответствии с системой команд центрального процессора АС-6 или в соответствии с системой команд ЭВМ БЭСМ-6.

### Блок выполнения операций сложения и вычитания

Блок выполнения операций сложения и вычитания С предназначен в первую очередь для выполнения команд сложения и вычитания чисел с плавающей запятой. Но, кроме того, в нем выполняются и команды сложения и вычитания чисел с фиксированной запятой и целый ряд других команд, из числа тех, которые требуют для своего выполнения того оборудования, которое имеется в блоке С.

В блоке С выполняются следующие команды:

а) Команды для чисел с плавающей запятой с одинарной и двойной точностью из системы команд центрального процессора АС-6 - сложение СЕ и СЕЕ, вычитание ВЕ и ВЕЕ, обратное вычитание ВОЕ и ВОЕЕ, вычитание абсолютных значений ВАЕ и ВАЕЕ.

б) Команды для чисел с фиксированной запятой словного и полусловного формата - сложение СС и СП, вычитание ВС и ВП, сложение кодов СКС и СКП, вычитание кодов ВКС и ВКП, арифметическая проверка АПС и АПП, циклическое сложение ЦСС, взятие абсолютного значения АЗС и АЗП, взятие отрицательного значения ОЗС и ОЗЛ, изменение знака ИЗС и ИЗП, преобразование из полусловного формата в формат с плавающей запятой ЗЕЦ.

в) Команды для чисел с плавающей запятой из системы команд БЭСМ-6 - арифметическое сложение АС, арифметическое вычитание АВ, обратное вычитание ОВ, вычитание модулей МВ, изменение знака ИЗ, сложение порядков СП, вычитание порядков НП, коррекция сложением КС, коррекция вычитанием КВ (последние четыре операции выполняются в блоке С, а не в блоке Л, из-за необходимости нормализации результата),

г) Команды сдвига слов и полуслов(из системы команд центрального процессора АС-6-САЛС, САПС, СЛЛС, СЛПС, САЛП, САПП, СЛЛП, СЛПП.).

д) Команды сдвига(из системы команд БЭСМ-6-СД-СК).

Блок-схема блока выполнения операций сложения и вычитания С дана на рис.25 . Блок-схема дана в упрощенном виде для иллюстрации выполнения только операций сложения и вычитания чисел с плавающей запятой в системе команд центрального процессора АС-6.

Регистры I-4 являются входными. На регистр поступает с выходного регистра БАК+а код операции и вся остальная информация, необходимая в блоке С для выполнения очередной команды. На основании этой информации работают усилители, управляющие вентильными схемами блока С так, чтобы обеспечить выполнение заданной команды (на блок-схеме эти усилители и их связи не показаны).

Регистры 2 и 3 имеют по 48 двоичных разрядов и предназначены для приема с регистров РПО и РВО первого и второго operandов и для хранения этих operandов в течение всего времени выполнения операции. Регистр 4 имеет 8 двоичных разрядов и является вспомогательным. Он подключен через промежуточный пассивный регистр 5 к разрядам порядка (4I+47 разряды) регистров РПО и РВО. Вентильные схемы 5 и 4 обеспечивают получение на регистре 4 разности порядков первого и второго operandов, вычисление которой требуется для выравнивания порядков operandов. Введение пассивного регистра 6 с его вентильными схемами 5 обусловлено тем, что получение разности порядков на одноэтажной схеме потребовало бы введения недопустимо сложной вентильной схемы.

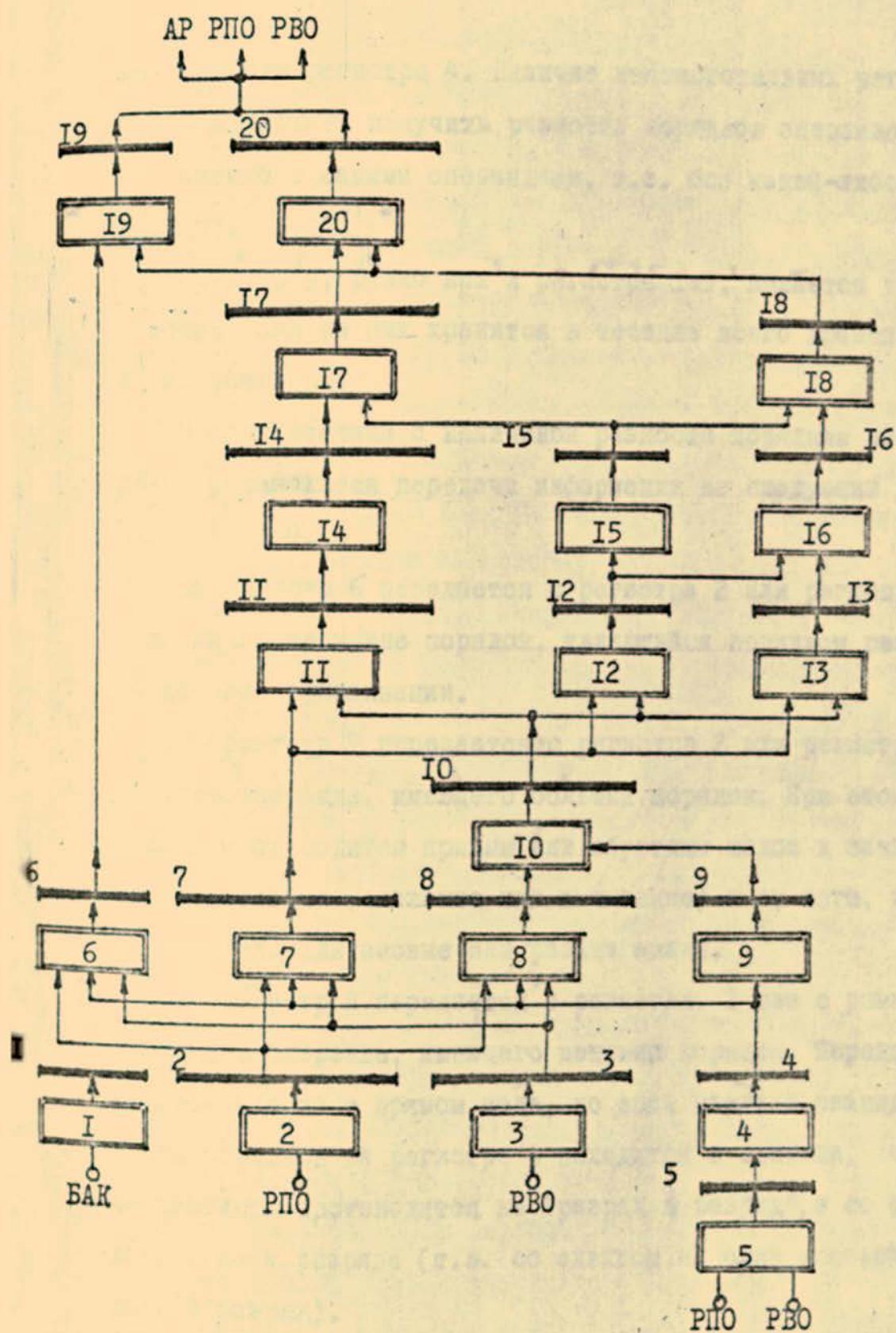


Рис. 25. Блок-схема блока "С"

мы на входе регистра 4. Наличие вспомогательных регистров 4 и 5 позволяет получить разность порядков операндов одновременно с самими операндами, т.е. без какой-либо потери времени.

Регистр 4, равно как и регистры I+3, является тактированным. Код на них хранится в течение всего времени выполнения операции.

В соответствии с величиной разности порядков на регистре 4 производится передача информации на следующий уровень - регистры 6-9.

На регистр 6 передается с регистра 2 или регистра 3 больший по величине порядок, являющийся порядком результата до его нормализации.

На регистр 7 передается с регистра 2 или регистра 3 мантисса операнда, имеющего больший порядок. При этом передача производится прямым или обратным кодом в зависимости от операции (сложение или вычитание) и от того, имеют ли операнды одинаковые или разные знаки.

На регистр 8 передается с регистра 2 или с регистра 3 мантисса операнда, имеющего меньший порядок. Передача ведется всегда в прямом коде, но если младший разряд разности порядков на регистре 4 находится в единице, то передача мантиссы производится не "разряд в разряд", а со сдвигом вправо на 4 разряда (т.е. со сдвигом на один шестнадцатиричный разряд).

Регистр 9 является совокупностью усилителей управления сдвигом выравнивания порядков и содержит 6 усилителей,

управляющих сдвигом мантиссы на 0,8,16,24,32 и 40 двоичных разрядов (на 0,2,4,6,8 и 10 шестнадцатиричных разрядов).

Усилители регистра 9 устанавливаются на основании дешифрации кода разности порядков, находящегося на регистре 4.

Усилители регистра 9 управляют вентилями передачи сдвигаемой мантиссы с регистра 8 на регистр 10 и обеспечивают получение на регистре 10 выравненной мантиссы операнда с меньшим порядком. Следующие ступени производят суммирование мантиссы операнда с большим порядком, находящейся на регистре 7 и сдвинутой мантиссы операнда, имевшего меньший порядок, находящейся на регистре 10. Суммирование ведется в шестнадцатиричном виде. Вентильные схемы II и 24 обеспечивают получение на регистре I4 промежуточного значения суммы, каждая шестнадцатиричная цифра которой равна сумме соответствующих шестнадцатиричных цифр слагаемых (без учета переносов из предыдущих разрядов). Введение двух последовательных ступеней суммирования (II и I4) обусловлено тем, что организация одноступенчатой схемы приводит к недопустимо сложным вентильным схемам.

Одновременно с выработкой промежуточного значения суммы работают схемы формирования и распространения переносов и схемы предварительной выработки информации для управления нормализацией результата.

Вентильная схема I2 вырабатывает и устанавливает на регистре I2 переносы, возникающие в каждом из шестнадцатиричных разрядов суммы при суммировании соответствующих шестнадцатиричных разрядов слагаемых, и, кроме того, вырабатывает

и устанавливает на регистре I2 признаки того, что сумма шестнадцатиричных цифр слагаемых в данном шестнадцатиричном разряде равна 15, т.е. что данный шестнадцатиричный разряд суммы, при переходе в него переноса из более младшего шестнадцатиричного разряда, должен пропустить этот перенос в более старший шестнадцатиричный разряд.

Вентильная схема I5 на основании информации на регистре I2 вырабатывает распространенные переносы, т.е. переносы, приходящие в данный шестнадцатиричный разряд суммы из любого другого шестнадцатиричного разряда. Следует отметить, что поскольку суммирование ведется в обратном коде, то цепи распространения переноса зациклены, т.е. перенос в младшие шестнадцатиричные разряды может прийти и из более старших шестнадцатиричных разрядов.

На основании распространенных переносов, полученных на регистре I5 производится, при пересылке с регистра I4 на регистр I7, формирование окончательной суммы. Каждая шестнадцатиричная цифра передается с регистра I4 на регистр I7 без изменения, если распространенный перенос в данном шестнадцатиричном разряде равен нулю, или с прибавлением единицы, если распространенный перенос в данном шестнадцатиричном разряде равен единице. Одновременно с прибавлением распространенного переноса при передаче с регистра I4 на регистр I7 производится и оборачивание суммы в тех случаях, когда результат суммирования является отрицательным. Для этого в состав вентильной схемы

I7 введен второй комплект вентилей, выполняющий те же функции, что и первый, но устанавливающий все разряды регистра I7 в противоположные значения.

Вентильная схема I3 вырабатывает для каждого шестнадцатиричного разряда дополнительные признаки, обеспечивающие максимально быстрое выполнение нормализации результата после того, как он будет получен. Регистр I3 содержит для каждого шестнадцатиричного разряда суммы 4 признака, каждый из которых фиксирует, что сумма шестнадцатиричных цифр слагаемых в данном разряде равна 0, I4, I6 или 30. На основании этих признаков, с учетом содержимого регистра I2, а затем и регистра I5, вентильные схемы I6 и I8 обеспечивают установку на регистре I8 одного из усилителей сдвига для нормализации. Регистр I8 состоит из I2 усилителей сдвига для нормализации-сдвиг вправо <sup>на</sup> 4 двоичных разряда, передача без сдвига и сдвиг влево на 4, 8, I2, I6, 20, 24, 28, 32, 36 и 40 разрядов.

Под управлением усилителей регистра I8 производится передача (с соответствующим сдвигом) суммы мантисс с регистра I7 на регистр 20 и передача (с соответствующей коррекцией) порядка большего операнда с регистра 6 на регистр I9.

Регистры I9 и 20 являются выходными регистрами блока выполнения операций сложения и вычитания С. С их выхода результат операции поступает в блоки АР, РПО и РВО, где соответствующие схемы управления обеспечивают его надлежащее использование.

Все регистры блока С, за исключением входных регистров

I-4 выполнены пассивными. Это связано не только с желательностью экономии тактирующего оборудования, но и с тем, что только пассивные схемы обеспечивают в данном случае двухтактный цикл работы. При двухтактном цикле работы очередная команда в течение одного полутакта занимает регистры РПО и РВО и в течение трех полутактов находится на входных регистрах блока С (регистры I-4). Так как для возможности передачи в блоки АР, РПО и РВО результат на выходных регистрах 19 и 20 блока С должен иметь площадку не менее четверти такта, задержка на шести вентильно-усилительных уровнях, через которые информация в блоке С переходит от входных регистров до выходных, не должно превышать 2,5 полутактов, что слишком мало для тактированных уровней, но вполне допустимо для пассивных, для которых при расчете прохождения сигналов не требуется учета разброса тактирующих сигналов.

Таким образом, пропускная способность блока С при выполнении команд над операндом с плавающей запятой из системы команд центрального процессора АС-6 составляет одну команду за два такта.

Выполнение в блоке С команд над операндами с фиксированной запятой не вызывает трудностей, так как эти команды выполняются почти на том же оборудовании, лишь с блокировкой выравнивания порядков и нормализации результата. Небольшое количество добавочного оборудования для выполнения этих команд сводится лишь к восьмиразрядному сумматору старших

разрядов при выполнении команд над словными операндами (для суммирования мантисс чисел с плавающей запятой достаточно сумматора длиной в 40 двоичных разрядов) и к добавочным вентилям приема на правую половину регистра 3 содержащего левую половину регистра РВ0 и вентилям пересылки содержащего левую половину регистра 2 на левую половину регистров 19 и 20 (эти дополнительные вентили нужны при выполнении команд над полусловными операндами).

Выполнение в блоке С команд над операндами с плавающей запятой с двойной точностью также не требует практически никакого добавочного оборудования, так как эти команды выполняются, как и в центральном процессоре АС-6, за четыре прохода через блок С, т.е. по частям. За время первого прохода производится определение разности порядков, суммирование старших половин мантисс и получение разрядов старшей половины мантиссы, выдвинутых в процессе выравнивания порядков в младшую половину мантиссы. За время второго прохода производится выравнивание и суммирование младших половин мантисс. За время третьего прохода производится прибавление к старшей половине мантиссы результата переноса из младшей половины, нормализация результата с соответствующей коррекцией порядка и запись старшей половины результата в блок РС. И, наконец, за время четвертого прохода производится запись в блок РС младшей половины результата. Пропускная способность блока С при выполнении операций над операндами с плавающей запятой с двойной точностью составляет, таким образом, одну команду за восемь .

тактов.

Выполнение в блоке С команд сдвига операндов обусловлено наличием в блоке С вентильных схем быстрого сдвига вправо (для выравнивания порядков) и влево (для нормализации результата). Однако, так как эти схемы осуществляют сдвиг с дискретностью в 4 двоичных разряда, для выполнения сдвигов на произвольное количество разрядов в блок Сведен добавочный регистр, не показанный на блок-схеме. Этот регистр получает код с регистра I0 через вентили, обеспечивающие сдвиг содержимого регистра I0 на 1, 2 или 3 разряда вправо или влево и передает свинутый код на вход регистра I7 (пересылка информации через регистры II и I4 при выполнении команд сдвигов не производится). Так как добавочный сдвиговый регистр включен в обход двух последовательных регистров, используемых при обычном прохождении информации, то для него допустима двойная задержка передачи информации. Поэтому он конструктивно может быть вынесен за пределы основной части оборудования блока С (что и приводит к задержке передачи информации от регистра I0 до входных цепей добавочного сдвигового регистра и от его выхода до входных цепей регистра I7) и его наличие не влияет на компактность расположения остального оборудования блока С и, следовательно, на скорость работы основных цепей блока С.

Скорость работы блока С при выполнении команд сдвига та же, что и при выполнении команд над операндами с плавающей или фиксированной запятой, т.е. одна команда за два такта.

Выполнение в блоке С команд для чисел с плавающей запятой из системы команд ЭВМ БЭСМ-6 требует введения добавочного оборудования. Это связано в первую очередь с двоичным сдвигом порядков чисел с плавающей запятой, обуславливающим заметно большее количество сдвигов при выравнивании порядков операндов и нормализации результата, с другим положением в коде числа разрядов порядка и знака мантиссы, с представлением мантиссы числа в дополнительном коде и с рядом других особенностей выполнения операций в системе команд БЭСМ-6.

Команды из системы команд БЭСМ-6 выполняются в блоке С с максимальным использованием оборудования, выполняющего команды из системы команд АС-6, но, кроме того, для их выполнения приходится вводить добавочные регистры с добавочным вентильным оборудованием. Добавочные регистры вводятся аналогично тому, как это было показано для команд сдвига, но в отличие от сдвигового добавочного регистра, заменившего два последовательных регистра, здесь два или даже три регистра заменяют один из основных регистров. Увеличение количества последовательных регистров, через которые проходит информация, вместе с влиянием вынесения их конструктивно на периферийную часть блока С (для предотвращения их влияния на плотность монтажа основных регистров и, следовательно, на скорость работы основного оборудования блока С) приводит к замедлению работы блока С при выполнении команд БЭСМ-6.

Пропускная способность блока С при выполнении команд

над операндами с плавающей запятой из систем команд БЭСМ-6 составляет одну команду за три-четыре такта (уточнение значения будет произведено на стадии технического проектирования).

Помимо вышеописанного, в блоке С предусмотрено добавочное оборудование, связанное с выполнением аппаратного контроля работы блока С и с выработкой признаков результата.

Оборудование аппаратного контроля состоит из входного оборудования, осуществляющего проверку по четности всей информации на входных регистрах I-4 выходного оборудования, осуществляющего формирование контрольных байтовских разрядов на выходных регистрах I9 и 20, и внутреннего оборудования, контролирующего выполнение команды в блоке С. Контрольное оборудование блока С построено так, что обеспечивает фиксацию любого одиночного сбоя в цепях блока С.

Выработка признаков результата выполняется в блоке в зависимости от режима работы в соответствии с системой команд центрального процессора АС-6 или в соответствии с системой команд БЭСМ-6.

### Блок выполнения операций умножения

Блок выполнения операций умножения У предназначен для выполнения следующих команд:

- а) команд умножения чисел с плавающей запятой из системы команд центрального процессора АС-6 - команд МЕ и МДЕ;
- б) команд умножения чисел с фиксированной запятой из системы команд центрального процессора АС-6 для словных и полусловных операндов - команд МДС, МДП, МП;
- в) команды умножения чисел с плавающей запятой, с двойной точностью из системы команд центрального процессора АС-6 - команды МЕЕ;
- г) команды умножения чисел с плавающей запятой из системы команд ЭВМ БЭСМ-6 - команды "арифметическое умножение" - "АУ".

Блок-схема блока выполнения операций умножения У приведена на рис.

Функционально блок У удобно рассматривать как четыре последовательно работающие узла:

- а) входной узел предварительной подготовки умножения - регистры I-3 и 5-7 с их входными диодными вентильными схемами;
- б) узел параллельного умножения - регистры 4,8-19 и их входные диодные вентильные схемы;
- в) узел накапливающего сумматора - регистры 20-22 и их входные вентильные схемы;
- г) узел приведения переносов и нормализации результата - регистры 23-31 и их входные диодные вентильные схемы.

В узле параллельного умножения осуществляется умножение множимого в первом такте умножения на 24 двоичных разряда множителя, а в каждом последующем такте умножения - еще на 18 двоичных разрядов множителя (недоиспользование возможностей

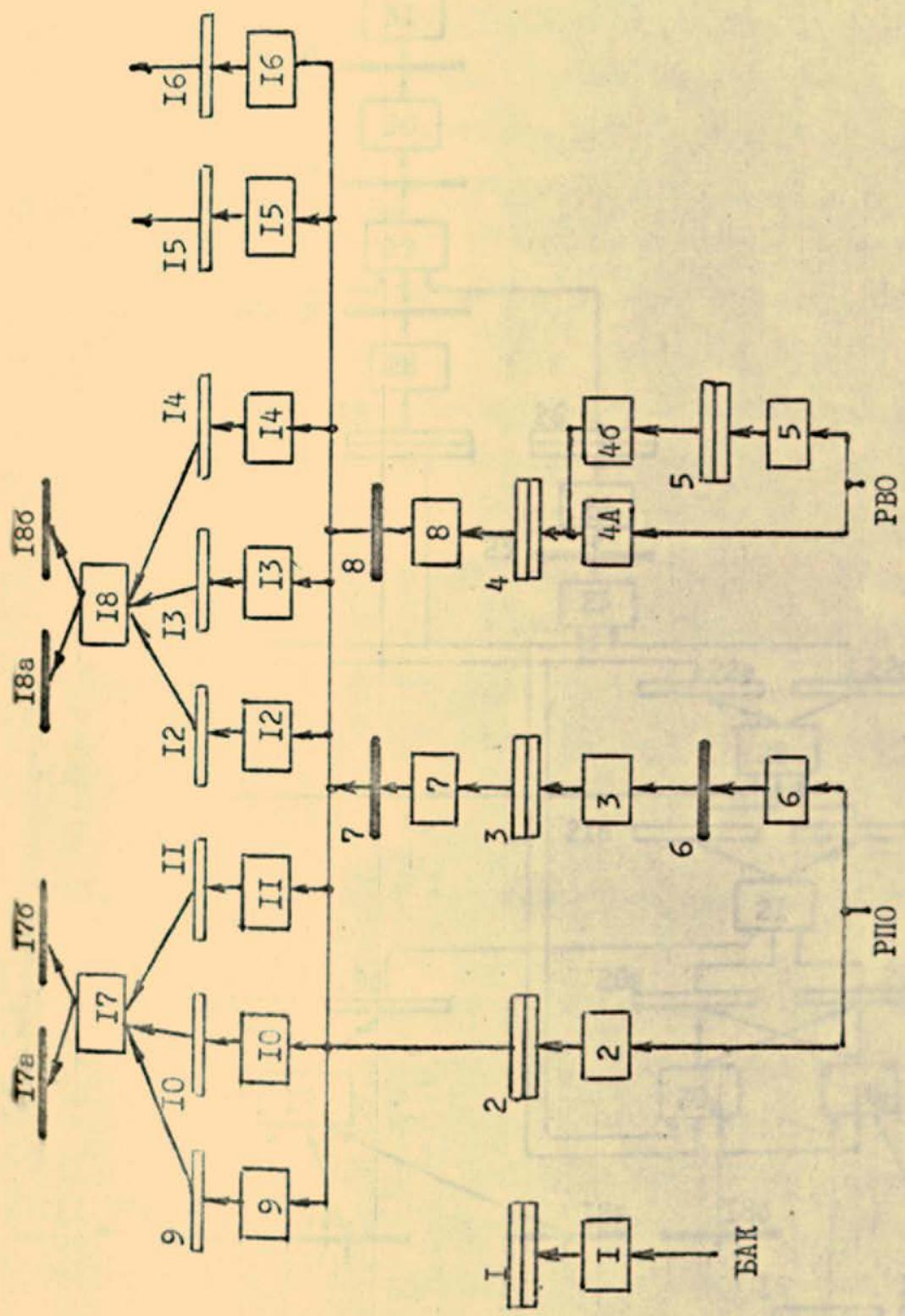


Рис. 26. Блок-схема блока "У"

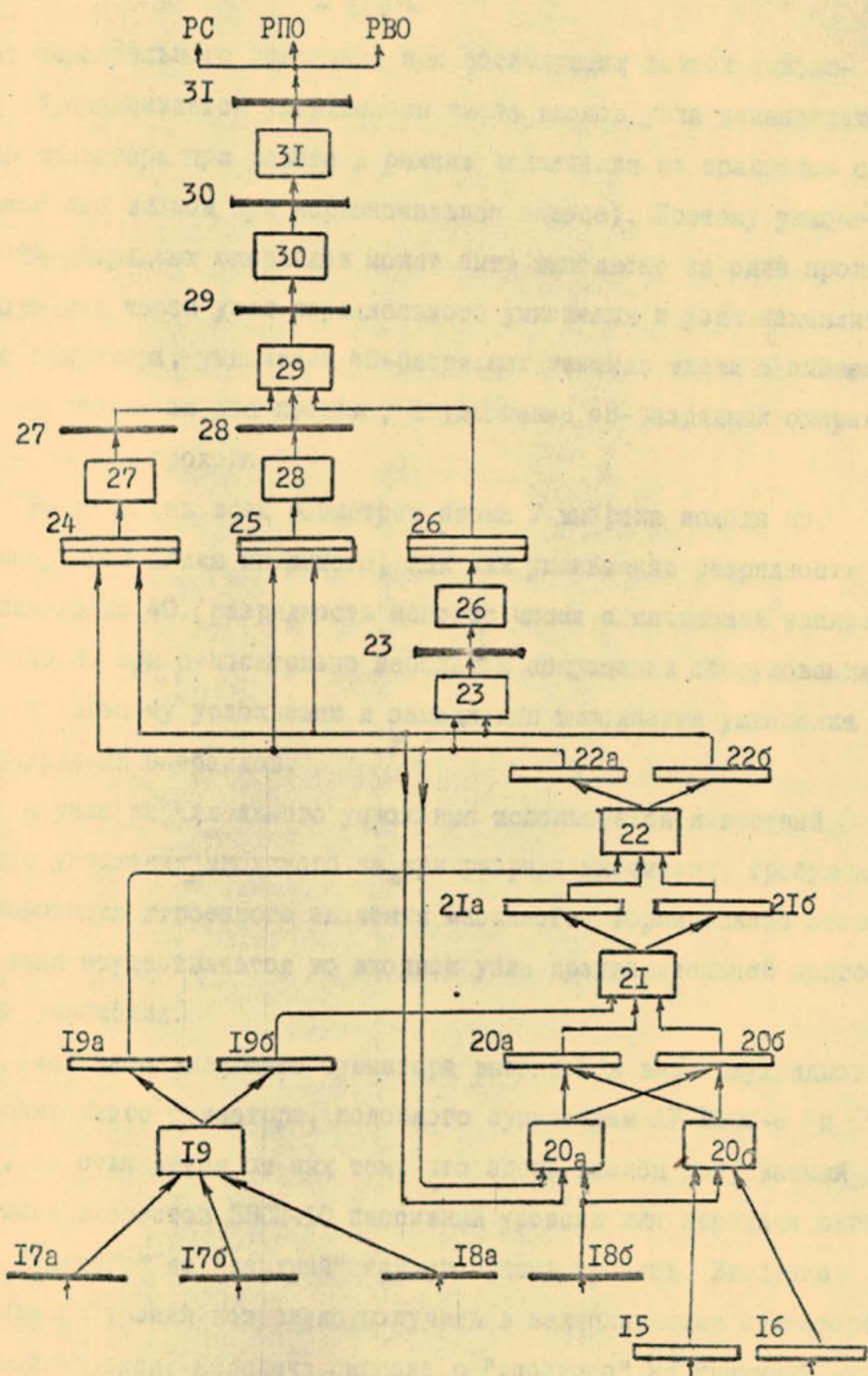


Рис. 27. Блок-схема блока "У" ( продолжение )

узла параллельного умножения при последующих тактах умножения обуславливается уменьшением числа входов узла накапливающего сумматора при работе в режиме накопления по сравнению с числом его входов при первоначальном заносе). Поэтому умножение 24-разрядных операндов может быть выполнено за один проход информации через узел параллельного умножения и узел накапливающего сумматора, умножение 40-разрядных мантисс чисел с плавающей запятой - за два прохода, а умножение 48-разрядных операндов - за три прохода.

Разрядность всех регистров блока У выбрана исходя из 48-разрядной длины множимого, так как уменьшение разрядности множимого до 40 (разрядность мантисс чисел с плавающей запятой) привело бы при относительно небольшом сокращении оборудования к значительному усложнению и замедлению выполнения умножения 48-разрядных операндов.

В узле параллельного умножения используется известный способ умножения множимого на три разряда множителя, требующий формирования утроенного значения множимого. Формирование этого значения осуществляется во входном узле предварительной подготовки умножения.

Узел накапливающего сумматора выполнен в виде двухрядного накапливающего сумматора, подобного сумматорам АУ БЭСМ-6 и АС-6, но отличается от них тем, что здесь введен допускаемый системой элементов БЭСМ-10 пассивный уровень для передачи сигнала с "нижнего" на "верхний" тактированный уровень. Введение пассивного уровня позволило получить в накапливающем сумматоре добавочный вход. Передача сигнала с "верхнего" на "нижний" тактированный уровень выполняется без промежуточного пассивного уровня. Это позволяет выполнять связи между выходами элементов

"верхнего уровня" и входами элементов "нижнего" с повышенной длиной, что заметно упрощает конструкцию сумматора, особенно с учетом того, что передача с "верхнего уровня" на "нижний" производится со сдвигом на 18 двоичных разрядов.

Регистры I-3 и 5 являются входными. Они выполнены тактированными и хранящими. Принимаемая на них информация хранится на них один полутакт при умножении полусловных операндов, три полутакта при умножении чисел с плавающей запятой и пять полутактов при умножении словных операндов.

На регистр I поступает с выходного регистра БАК+а код операции и вся остальная информация, необходимая в блоке У для выполнения очередной команды. На основании этой информации работают усилители, управляющие вентильными схемами блока У так, чтобы обеспечить выполнение заданной команды.

Регистр 2 имеет 48 двоичных разрядов (не считая контрольных) и предназначен для приема с регистра РПО первого операнда, используемого в дальнейшем в качестве множимого.

Регистр 5 также имеет 48 двоичных разрядов (не считая контрольных разрядов) и предназначен для приема с регистра РВО второго операнда, используемого в блоке У в качестве множителя.

Регистр 3 вместе с пассивными регистрами 6 и 7 обеспечивают получение утроенного значения множимого. Утрение производится суммированием множимого с его же удвоенным (полученным сдвигом кода на один разряд влево) значением. На регистре 6 фиксируются переносы, возникающие в каждом шестнадцатиричном разряде этой суммы, фиксируется значение шестнадцатиричного разряда, равное 15 (для управления распространением переносов) и готовится получение значения суммы в каждом шестнадцатиричном разряде. На регистре 3 окончательно формируется

значение суммы в каждом шестнадцатиричном разряде (без учета переноса из более младших шестнадцатиричных разрядов) и одновременно формируется перенос, приходящий в каждый шестнадцатиричный разряд из более младших шестнадцатиричных разрядов. Затем содержимое шестнадцатиричных разрядов регистра 3 передается на регистр 7, причем те шестнадцатиричные цифры, в которые не пришел перенос, передаются на регистр 7 без изменения, а те, в которые перенос пришел, передаются с прибавлением единицы. В результате на регистре 7 получается утроенное значение множимого.

Регистр 4 содержит 24 двоичных разряда и является регистром анализа цифр множителя. Первую (а при полусловных операндах и единственную) часть разрядов множителя (начиная с младших разрядов) он получает непосредственно с регистра второго операнда РВО. Следующие более старшие (и последние при операндах с плавающей запятой) 18 двоичных разрядов множителя поступают на регистр 4 с регистра 5 в следующем такте. При словных операндах с фиксированной запятой в третьем такте на регистр 4 передаются с регистра 5 последние, самые старшие 18 разрядов множителя. Количество двоичных разрядов множителя, принимаемых на регистр 4 в первом такте с регистра РВО зависит от формата операндов. Прием на регистр 4 идет при этом в старшие разряды, а пустыми оставляются его младшие разряды.

Регистр 4 разбит на 8 групп по 3 двоичных разряда в каждой. Каждой такой группе регистра 4 соответствует группа из 8 усилителей на регистре 8 (таким образом регистр 8 содержит 64 усилителя). Работа усилителей каждой группы регистра 8 определяется содержимым каждой трехразрядной группы регистра 4 в соответствии с известной логикой умножения на три разряда множителя.

Усилители регистра 8 управляют вентилями на входах восьми регистров частичных произведений (регистры 9-16), обеспечивая получение на этих регистрах произведений множимого на соответствующие тройки разрядов множителя. Следует отметить, что из 8 регистров частичных произведений два (регистры 15 и 16) используются только в первом такте умножения, а остальные 6 работают и в первом и в последующих тактах.

Вентильные схемы 17-19 являются трехходовыми сумматорами, а их выходные регистры 17а-19а и 17б-19б - регистрами поразрядного переноса и поразрядной суммы на соответствующих выходах этих сумматоров. Трехходовые сумматоры 17-19 уменьшают количество слагаемых, подаваемых в накапливающий сумматор в каждом такте, с шести до трех.

Накапливающий сумматор состоит из "нижнего" уровня, куда входят два тактированных трехходовых сумматора (вентильные схемы 20а и 20б) и два тактированных регистра (регистр переносов 20а и регистр суммы 20б), из промежуточного уровня, в который входят пассивный трехходовой сумматор (вентильная схема 21) и два пассивных регистра (регистр переносов 21а и регистр суммы 21б) и из "верхнего" уровня, включающего тактированный трехходовой сумматор (вентильная схема 22) и два тактированных регистра (регистр переносов 22а и регистр суммы 22б).

Накопление суммы частичных произведений начинается на "нижнем" уровне. В первом такте накопления трехходовой сумматор 20а открыт (что исключает из суммирования содержимое регистров 22а и 22б, не имеющее отношения к данной операции умножения), а открыт трехходовой сумматор 20б (что позволяет получить на регистрах 20а и 20б сумму содержимого трех регист-

ров - I8б, I5 и I6). Затем, пока полученная сумма суммируется на трехходовых сумматорах 2I и 22 с содержимым регистров I9а и I9б, трехходовой сумматор 20б закрывается и открывается трехходовой сумматор 20а. Количество входов в "нижнем" уровне уменьшается с трех до одного, но зато в суммировании принимает участие и ранее полученная сумма, находящаяся на регистрах 22а и 22б.

После выполнения соответствующего числа тактов умножения на регистрах 22а и 22б получается произведение операндов в двухрядном виде. Длина этого произведения доходит до 96 двоичных разрядов (при умножении слов с фиксированной запятой), поэтому преобразование его к однорядному виду - приведение переносов - приходится выполнять в два приема, сначала для младшей половины разрядов, а затем для старшей (с учетом переноса в нее из младшей), так как распространение переносов в одном логическом уровне приводит при таком количестве разрядов к недопустимо сложным вентильным схемам.

Приведение переносов в произведении сводится к сложению содержимого регистров 22а и 22б и организовано по шестнадцатиричной схеме, т.е. каждые 4 двоичные разряда рассматриваются как один шестнадцатиричный разряд.

Пассивный регистр 23 фиксирует для младшей половины разрядов произведения наличие переносов, вырабатываемых в каждом шестнадцатиричном разряде при полном сложении в нем двоичных поразрядных переносов и двоичных поразрядных сумм (без учета прихода переноса из более младшего шестнадцатиричного разряда). Кроме того, регистр 23 фиксирует наличие в каждом из шестнадцатиричных разрядов этой суммы наличие кода I5, что требуется для управления распространением переносов.

Регистр 26 фиксирует результат распространения переносов в младшей половине разрядов произведения, т.е. фиксирует приход в каждый шестнадцатиричный разряд младшей половины разрядов произведения переноса из более младших шестнадцатиричных разрядов. Кроме того, здесь же фиксируется наличие переноса из всей младшей половины произведения в младший шестнадцатиричный разряд старшей половины произведения.

Регистры 25 и 28 обеспечивают распространение переносов в старшей половине разрядов произведения и функционируют аналогично регистрам 23 и 26 младшей половины с той только разницей, что работают с запаздыванием на один логический уровень и что в вентильных схемах регистра 28 учитывается перенос из младшей половины разрядов произведения.

Вентильные схемы 24 и 27 обеспечивают получение на регистре 27 значений шестнадцатиричных разрядов суммы содержимого регистров 22а и 22б (без учета переносов из более младших шестнадцатиричных разрядов). Двухступенчатое образование этой суммы обусловлено тем, что одноступенчатая схема требует недопустимо сложных вентилей.

Вентильные схемы 29 обеспечивают получение на регистре 29 окончательного однорядного значения произведения, передавая с регистра 27 на регистр 29 те шестнадцатиричные разряды, в которые не пришел перенос из более младших разрядов без изменения, а не, в которые пришел перенос - с прибавлением единицы.

Вентильные схемы 30 и 31 обеспечивают нормализацию результата. Наличие двух логических уровней для нормализации результата обуславливается большим количеством сдвигов, которые могут потребоваться при работе с удвоенной точностью в системе команд.

AC-6 и, особенно, при работе в системе команд БЭСМ-6 (где вследствие двоичного представления порядков количество сдвигов при нормализации мантиссы результата может быть от 0 до 79). Вентильные схемы 30 обеспечивают шестнадцатиричный (т.е. кратный четырем двоичным разрядам) сдвиг влево в пределах от 0 до 9 шестнадцатиричных сдвигов, т.е. имеет по 10 вентилей в каждом двоичном разряде. Вентильные схемы 31 обеспечивают сдвиг влево на 10 шестнадцатиричных разрядов (т.е. на 40 двоичных разрядов) и, кроме того, обеспечивают сдвиги влево на 1, 2 или 3 двоичных разряда, необходимые при работе в системе команд БЭСМ-6. Таким образом, вентильные схемы 31 содержат 8 вентилей в каждом двоичном разряде - а именно, вентили сдвига на 0, 1, 2, 3, 40, 41, 42 и 43 двоичных разрядов.

Регистр 31 является выходным регистром блока У. С него результат выполненной в блоке У команды умножения поступает на один из входов блока регистров слов РС для записи в соответствующий регистр слова и на входы регистров РПО и РВО для ускоренного использования, при необходимости, в качестве operandов последующих команд. Регистры 24-26 выполнены тактированными и хранящими. Принятая на них информация хранится на них в течение трех полутактов, что позволяет все последующие уровни блока У выполнять пассивными, обеспечивая определенную экономию тактирующего оборудования. Четыре последовательных пассивных вентильно-усилительных уровня между выходами регистров 24-26 и выходами регистра 31 дают задержку распространения информации, не превышающую двух полутактов. Следовательно, гарантированное время существования результата на выходном регистре 31 составляет не менее полутакта, что вполне достаточно для передачи результата в другие блоки АУ, и даже позволяет считать выход блока У относительно некритичным к конструкции. Выполнение регистров 24-26 в качестве хранящих, приводящее к экономии тактирующего оборудования в последующих уровнях,

в то же время ограничивает пропускную способность блока У одной командой за два такта. Однако это не имеет существенного значения, так как пропускная способность блока У и без того ограничена той же величиной, определяемой пропускной способностью входа записи блока регистров РС, на который поступает для записи результат команды, выполненной в блоке У.

Следует отметить, что управление вентилями нормализации не вызывает в блоке У тех трудностей, которые возникают в блоке С при выполнении сложения или вычитания чисел с плавающей запятой. Это объясняется тем, что при выполнении операции умножения количество нулей в старших разрядах произведения равно или на единицу больше суммы количества нулей в старших разрядах сомножителей. Сумма количества нулей в старших разрядах сомножителей может быть подсчитана в процессе выполнения операции умножения заметно раньше получения произведения и поэтому определение количества сдвигов, необходимых для нормализации, сводится к определению равенства нулю одного заранее известного разряда. Это обуславливает относительную простоту цепей управления нормализацией результата. В состав оборудования блока У входят также цепи аппаратного контроля и цепи выработки признаков результата, аналогичные соответствующим цепям блоков Л и С.

Пропускная способность блока У соответствует как было показано выше, выполнению одной команды за два такта при полусловных операндах или операндах с плавающей за-

пятой и одной команды за три такта при словных операндах с фиксированной запятой.

Время выполнения одной команды умножения, т.е. время, через которое ее результат может быть использован в качестве операнда последующей команды, определяется при умножении полусловных операндов тем, что информация в блоке У проходит от входа до выхода через 14 пассивных и тактированных уровней. Так как время прохождения информации через каждый уровень составляет четвертую часть такта, то, с учетом добавочной четверти такта, которая может быть выделена на распространение информации по проводам от выходных элементов блока У до входных элементов блоков РПО и РВО и с учетом четверти такта прохождения информации через элементы самих регистров РПО и РВО, время выполнения одной команды умножения при полусловных операндах равно 4 тактам.

Умножение чисел с плавающей запятой требует в блоке У на один такт больше и, следовательно, время умножения чисел с плавающей запятой равно 5 тактам.

Умножение словных операндов с фиксированной запятой требует еще одного добавочного такта и следовательно время выполнения умножения таких операндов равно 6 тактам.

Таким образом, блок У обеспечивает следующие характеристики производительности арифметического устройства

центрального процессора БЭСМ-10:

- а) одна команда за два такта при операндах с плавающей запятой и при полусловных операндах с фиксированной запятой и при условии, что результат предыдущей команды умножения не используется в качестве операнда последующих команд умножения;
- б) одна команда за три такта при словных операндах с фиксированной запятой и при условии, что результат предыдущей команды умножения не используется в качестве операнда последующих команд умножения;
- в) одна команда за четыре такта при полусловных операндах с фиксированной запятой и при использовании результата предыдущей команды умножения в качестве операнда следующей команды умножения;
- г) одна команда за пять тактов при операндах с плавающей запятой и при использовании результата предыдущей команды умножения в качестве операнда следующей команды умножения;
- д) одна команда за шесть тактов при словных операндах с фиксированной запятой и при использовании результата предыдущей команды умножения в качестве операнда следующей команды умножения.

### Блок выполнения операций деления

В блоке выполнения операций деления (блок Д) предусмотрено выполнение следующих команд:

- а) Команды деления чисел с плавающей запятой из системы команд центрального процессора АС-6 - команды ДЕ.
- б) Команды деления чисел с плавающей запятой из системы команд БЭСМ-6 - команды "арифметическое деление" АД.
- в) Команды деления чисел с фиксированной запятой из системы команд центрального процессора АС-6 - команды деления полуслова на полуслово ДП и команды деления слова на полуслово ДДП.
- г) Команды образования "циклического кода" из системы команд центрального процессора АС-6 - команды "циклическое деление".
- д) Команды извлечения квадратного корня из числа с плавающей запятой, представленного в форме, принятой в АС-6. В центральном процессоре АС-6 такой команды нет, однако, в систему команд БЭСМ-10 ее решено ввести, так как они часто встречается во внутренних циклах программ определенного круга больших задач, которые предполагается решать на БЭСМ-10.

Все команды, выполняемые в блоке Д встречаются в программах задач относительно редко. Включение в состав АУ центрального процессора БЭСМ-10 специализированного функционального блока для выполнения этих команд обус-

ловлено не столько повышением производительности АУ за счет совмещения во времени выполнения этих команд в блоке Д с выполнением других команд в других блоках АУ, сколько с тем, что для выполнения этих команд требуются специфические связи между элементами. Введение таких связей в какой-либо из остальных блоков АУ привело бы к усложнению конструкции последнего, к увеличению расстояния между его элементами и, как следствие, к снижению его производительности.

В то же время следует, учитывать, что хотя команды, выполняемые в блоке Д, составляют относительно небольшой процент от общего числа команд, выполняемых в АУ, влияние команд, выполняемых в блоке Д, на общую производительность АУ может быть достаточно заметным. Это обуславливается тем, что выполнение таких команд по известным в настоящее время алгоритмам требует времени в несколько раз (а иногда и более, чем на порядок) большего, чем время выполнения других, более часто встречающих команд.

Поэтому при разработке блока Д принят целый ряд новых схемных решений, обеспечивающих повышение его производительности в заметно большей степени, чем это обуславливается повышением тактовой частоты элементов.

Внутренняя структура блока Д определяется в первую очередь требованием максимально быстрого выполнения в нем команд деления. Остальные операции выполняются в блоке

Д так, чтобы максимально использовать оборудование уже имеющиеся для выполнения деления.

Алгоритм выполнения команд деления в блоке Д аналогичен алгоритму выполнения команд деления в центральном процессоре АС-6, т.е. производится также на накапливающем сумматоре, работающем в двухрядном коде, без приведения переносов. Однако в отличие от сумматора АС-6 в сумматоре блока Д образование очередного остатка и его анализ производятся не один раз за такт, а вдвое чаще — в каждом полутакте. Это обеспечивает получение в каждом такте двух разрядов частного, т.е. вдвое повышает скорость собственно деления.

Возможность такого ускорения выполнения деления базируется на том, что система элементов БЭСМ-10 позволяет при определенных условиях, включить между двумя тактированными уровнями промежуточный пассивный уровень. Кроме того, в блоке Д использована новая схема накопления цифр частного, накапливающая их сразу в однорядном виде, а не в двухрядном, как в АС-6 и БЭСМ-6. Это позволяет избежать потери времени на суммирование положительной и отрицательной составляющих частного. И, наконец, для ускорения выполнения деления при операндах с фиксированной запятой в блоке Д использованы более сложные и более быстрые (по сравнению с АС-6) цепи нормализации делителя, цепи приведения переносов при образовании остатка, цепи денормализации остатка.

Вышесказанное, а также и ряд других, менее существенных схемных решений, обеспечили выполнение команд в блоке Д за значительно меньшее число тактов, чем в централь-

ном процессоре АС-6 - например, для выполнения команды деления чисел с плавающей запятой в БЭСМ-10 требуется 25 тактов против 55 в АС-6.

Операция извлечения квадратного корня из числа с плавающей запятой является самой медленной из выполняемых в блоке Д. Она требует для своего выполнения около 50 тактов. Однако, хотя по сравнению с выполнением этой операции по подпрограмме выигрыш по производительности все равно получается очень большой, на стадии технического проектирования предполагается принять меры для ускорения выполнения этой операции.

Наряду с оборудованием, обеспечивающим выполнение заданных команд, в блоке Д предусмотрено, как и в остальных функциональных блоках АУ, оборудование аппаратного контроля. Функции его аналогичны функциям оборудования аппаратурного контроля блоков Л и С, описанным выше.

#### Блок выполнения операций над последовательностями

Блок выполнения операций над последовательностями предназначен для выполнения следующих команд:

- а) Команд обработки последовательностей двоичных разрядов при произвольной, задаваемой в команде длине этих последовательностей - команд для шкал из системы команд центрального процессора АС-6;
- б) Команд обработки последовательностей байтов при

произвольной, задаваемой в команде длине этих последовательностей - команд для строк из системы команд центрального процессора АС-6.

в) Команд для десятичных операндов с переменной, задаваемой в команде разрядностью этих операндов - команд для десятичных чисел из системы команд центрального процессора АС-6.

Структура и схема блока П совпадает со структурой и схемой аналогичного блока, имеющегося в составе центрального процессора АС-6, однако, за счет более быстродействующей системы элементов, скорость выполнения в БЭСМ-10 соответствующих команд значительно выше, чем в АС-6.

Время выполнения команды в блоке П зависит от длины обрабатываемых операндов и может быть охарактеризована следующими цифрами:

- а) Обработка одного байта за такт при выполнении большинства команд обработки последовательностей байтов.
- б) Обработка четырех двоичных разрядов за такт при выполнении большинства команд обработки последовательностей двоичных разрядов.

На стадии технического проектирования предполагается решить вопрос о рациональности увеличения скорости работы блока П еще в два раза за счет определенного увеличения в нем количества оборудования (за счет распараллеливания ряда его цепей).

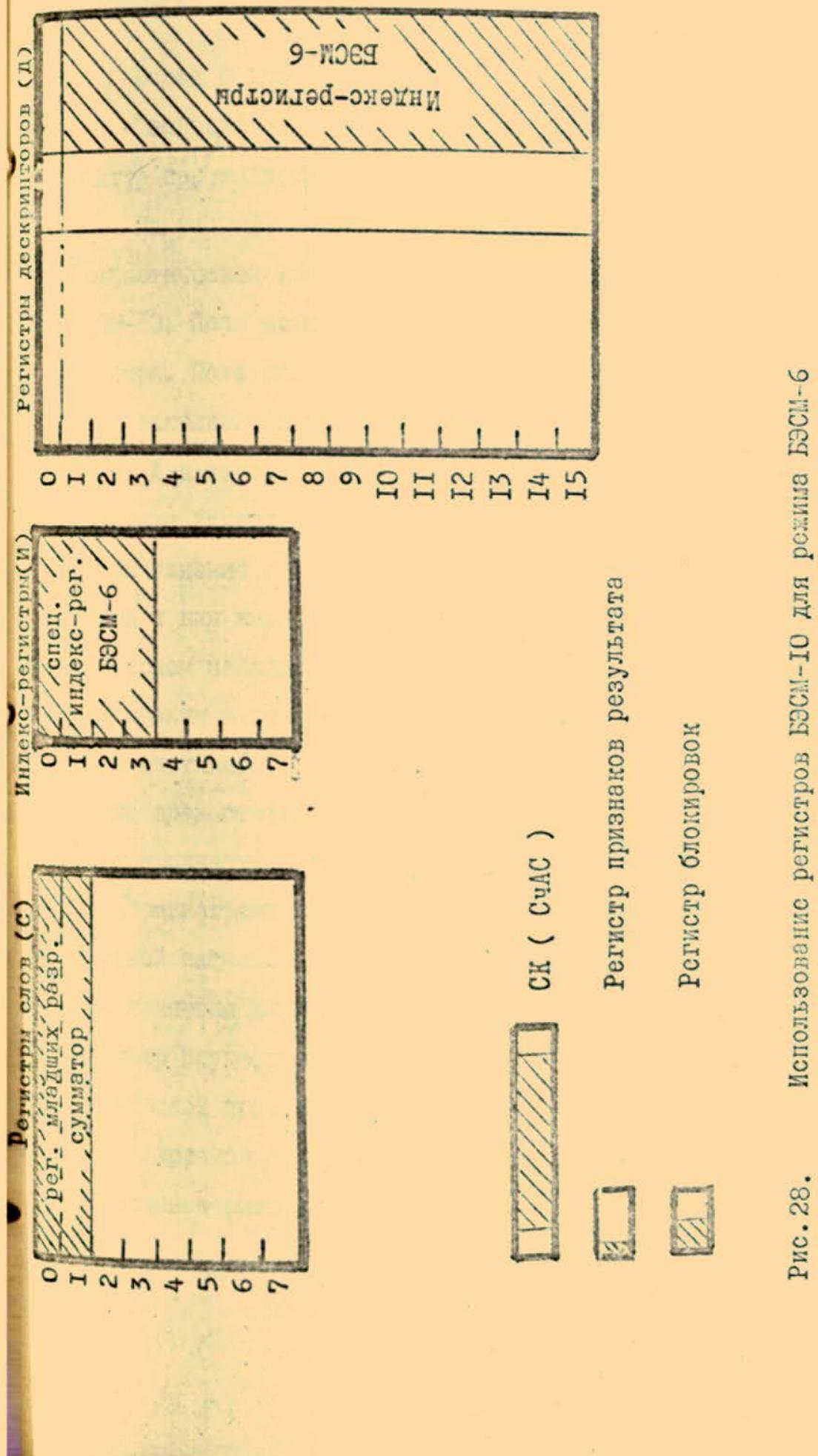
### ОБЕСПЕЧЕНИЕ СОВМЕСТИМОСТИ С БЭСМ-6

Системы команд БЭСМ-10 и БЭСМ-6 близки по своей структуре. Совпадение таких характерных особенностей, как двуместность операций, использование аналогичных функциональных групп прямоадресуемых регистров, вычисление математических исполнительных адресов по памяти, размеров operandов позволяет естественным образом не только обеспечить совместимость обеих ЭВМ, но и в полной мере использовать все достоинства структуры БЭСМ-10 при выполнении программ, написанных в командах БЭСМ-6.

На рис. 28 показано использование регистров БЭСМ-10 для режима выполнения команд БЭСМ-6. Поля базы дескрипторов используются в качестве индекс-регистров БЭСМ-6. Такое решение удобно тем, что значение базы участвует в вычислении исполнительного адреса при выполнении команд БЭСМ-10. Состав операций над дескрипторами полностью обеспечивает выполнение всех команд БЭСМ-6 над индекс-регистрами.

Индекс-регистры БЭСМ-10 используются для хранения таких специиндексов БЭСМ-6, как M 26, M 27, M 16.

Регистры слов БЭСМ-10 используются в качестве сумматора и регистра младших разрядов БЭСМ-6. Имеются некоторые особенности использования регистров сумматоров, которые обеспечивают параллельное выполнение команд в арифметическом устройстве БЭСМ-10.



Более подробно об этом сказано ниже.

Распределение регистров определяет основной объем и характер перекодировки команд БЭСМ-6 в формат команд БЭСМ-10.

На рис. 29 показана принципиальная схема перевода арифметической команды БЭСМ-6 в аналог в системе команд БЭСМ-10. Поле номера индекса принимается в поле номера дескриптора. Поле номера регистра слов в команде-аналоге может быть заполнено номером любого регистра, за исключением нулевого. Для того, чтобы создать условия для параллельного выполнения команд в арифметическом устройстве БЭСМ-10 в процессе выполнения программы выделяются независимые участки программы и для каждого такого участка назначается новый регистр. Критерием независимости участков программы является команда считывания в сумматор. Перебор номеров регистров осуществляется счетчиком, пересчитывающим команды считывания. Такая схема предоставляет возможность параллельной обработки до семи независимых участков программы. Однако параллельная работа регистрами слов должна сопровождаться строго последовательной засыпкой информации в нулевой регистр, предназначенный для хранения младших разрядов. Синхронизация засылок в этот регистр осуществляется схемой, аналогичной схеме, управляемой выработкой признаков результата операций в АУ.

Адресное смещение, указанное в команде БЭСМ-6 заменяется на длинное смещение в команде БЭСМ-10. При приеме команды

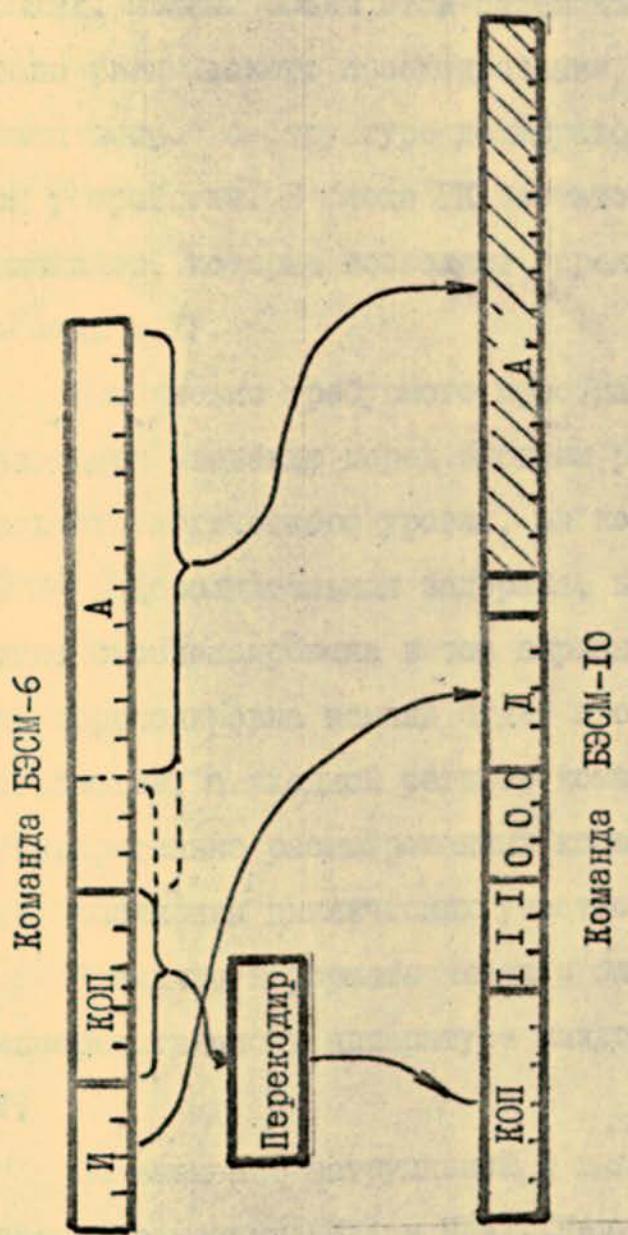


Рис. 29. Преобразование формата команд БЭСМ-6 в формат БЭСМ-10

Рис. 29.

с коротким смещением на входной регистр команд в I3+I6 разрядах адреса происходит распространение значения I9 разряда кода команды БЭСМ-6.

Формирование кода операции осуществляется при приеме команд. Полный объем этой перекодировки будет определен на этапе технического проектирования, когда будет окончательно решен вопрос о структуре дешифратора операций в арифметическом устройстве. В блоке РК достаточно выделение лишь таких признаков, которые позволяют определить характер обработки команды в УУ.

Выполнение требуемого преобразования осуществляется благодаря введению перед входным регистром команд дополнительного логического уровня, на который принимаются команды БЭСМ-6. Дополнительная задержка, вносимая этим уровнем, будет скомпенсирована в том варианте, при котором окончательная перекодировка команд будет происходить в арифметическом устройстве, а входной регистр команд будет работать в режиме предварительно расшифрованных команд, как это имеет место при выполнении циклических участков программы.

Различие в формате чисел с плавающей запятой учитывается непосредственно в аппаратуре каждого из функциональных блоков АУ.

Не вызывает затруднений и выполнение таких операций как изменение команд (ИКЛ и ИКИ). Результат их выполнения засы-

ляется в индекс-регистр, играющий роль модификатора M 16.

Выполнение следующей за этими командами операции сводится к выполнению такой команды БЭСМ-10, в которой вычисление исполнительного адреса производится по трем компонентам: базы, индекса и адресного смещения.

Разработанные способы осуществления совместимости БЭСМ-10 и БЭСМ-6 отличаются простотой и полным использованием всех возможностей, предоставляемых аппаратурой центрального процессора БЭСМ-10. Скорость выполнения программы, написанной в командах БЭСМ-6 уступает скорости выполнения программ в командах БЭСМ-10 лишь в той степени, в которой система команд БЭСМ-10 в целом эффективнее, чем система команд БЭСМ-6. Процессор БЭСМ-10 будет выполнять, примерно, то же число операций в секунду, как для команд БЭСМ-10, так и для команд БЭСМ-6, и это является следствием органической близости их структуры и системы команд.

## ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ

Рассмотренные ранее вопросы организации памяти и параллельной работы процессоров в многопроцессорной системе ставили своей целью обеспечить, по возможности, максимально быструю работу каждого процессорного модуля.

Рассмотрим теперь вопросы, связанные с сокращением времени выполнения программы и, в первую очередь, программы внутренних циклов.

Использование микроэлектронных элементов позволило значительно повысить производительность арифметических устройств как за счет повышения скорости работы самих элементов, так и за счет большего распараллеливания выполнения отдельных арифметических операций внутри арифметического устройства. В результате суммарная максимальная производительность арифметического устройства очень высока и превышает возможности устройства управления, организующего выборку команд программы, их дешифрацию, вычисление адресов и обращение к памяти по этим адресам для выборки операндов операции и записи результата. Нагрузка на устройство управления еще более возрастает при выполнении операций с элементами массивов, где необходимо выполнять индексацию и проверку границ. Эти трудности особо ощущимы в сравнительно небольших внутренних программных циклах.

Рассмотренные ранее конкретные примеры внутренних циклов больших задач весьма показательны в этом отношении.

Таким образом, наиболее критичным с точки зрения уменьшения времени выполнения программы становится устройство управления командами, и вопросы ускорения его работы оказываются тесно связанными со структурой команд.

Одним из принципов, положенных в основу системы команд ЦП АС-6, был принцип повышения "алгоритмического содержания" команд за счет исключения, по возможности, вспомогательных команд, не связанных непосредственно с операциями в исходном алгоритме задачи. Достаточно большое внимание было уделено возможности разгрузки выполняемого тела программного цикла от всякого рода вспомогательных команд, связанных с загрузкой, индексированием и организацией программного цикла во всех тех случаях, когда это действительно не требуется.

Как уже отмечалось, особое значение эти вопросы имеют в сравнительно коротких программных циклах, и поэтому оказалось целесообразным решать их на основе системы команд с прямой адресацией небольшого количества регистров разного функционального назначения: регистров дескрипторов, регистров индексов и регистров слов.

Вместе с тем использование дескрипторов и двух стеков в системе команд ЦП АС-6 позволяет успешно решать и основные вопросы, связанные с реализацией программ на языках высокого уровня.

О высоком алгоритмическом содержании команд ЦП АС-6 можно судить по программам внутренних циклов, которые приведены на рис. 2 и 4 . Как видно из этих примеров, структура и количество операций математической записи весьма близки к структуре и количеству команд программы. Следует отметить и компактность выполняемых программных циклов. На рисунках указан размер каждой команды в байтах и общий объем программного цикла. Для первого примера объем программного цикла составляет 32 байта ( $5\frac{1}{3}$  слова), а для второго - 11 байтов ( $1\frac{5}{6}$  слова), что способствует эффективности буфера команд внутри процессора. Те же самые вычислительные алгоритмы внутреннего цикла, записанные в командах БЭСМ-6, требуют 22 и 7 команд и занимают 11 и 3,5 слов соответственно. Запись тех же циклов в командах машины со стековой организацией требует примерно в три раза больше команд и примерно в два раза больше слов памяти.

Помимо уменьшения числа команд, большое значение для сокращения времени их выполнения играет фиксированность характера их выполнения и отсутствие лишних анализов и динамической интерпретации во время их выполнения, так как все это сильно усложняет логические схемы и, в конечном счете, ведет к лишним задержкам в условиях высокопроизводительного процессора с широким распараллеливанием.

В этих условиях система команд с прямой адресацией регистров позволяет обеспечить быструю дешифрацию и вычисление исполнительных адресов. Проработка узла управления на стадии эскизного проекта показала, что дешифрация команд

типа "регистр-регистр" может быть уменьшена до I такта.

Возможно повышение скорости обработки и команд с адресами по памяти.

В результате, время выделения и обработки устройством управления команд типа "регистр-регистр" сокращено до I такта.

Кроме того, на этапе эскизного проектирования был про-работан улучшенный вариант организации коротких циклов, целиком помещающихся в буфере команд. В результате, удается до-стигнуть практически полной линейной развертки программы цикла, т.е. первая команда цикла выбирается и обрабатывается устройством управления практически без задержки. При некото-ром усложнении, если оно будет признано целесообразным, воз-можно полное исключение и времени выборки самой команды кон-ца цикла (КЦ).

Сокращение времени выборки и обработки указанных команд повышает скорость генерации адресов по памяти и существенно для более полного перекрытия по времени обращений к СОЗУ и выполнения арифметических команд.

На основании проведенных проработок представляется возможным сделать некоторые предварительные оценки эффективнос-ти принятых решений как для отдельного процессора, так и многопроцессорной системы в целом.

По причине сложного характера взаимодействия отдельных узлов высокопроизводительной многопроцессорной системы точная детальная оценка производительности такой системы для широкого

лиапазона задач возможна, по-видимому, лишь на этапе реального функционирования системы.

Арифметическое устройство процессорного модуля БЭСМ-10 обладает очень высокими параметрами. Так время выполнения операции сложения 48-разрядных чисел с плавающей запятой составляет 2 такта, а операции умножения - 5 тактов с возможностью запуска операции умножения над операндами, не являющимися результатом предыдущей операции, каждые 2 такта (пропускное время или такт конвейера).

Суммарное время выполнений операций, поступающих в арифметическое устройство, как правило, значительно меньше, чем суммарное время прохождения команд программы через устройство управления. Кроме того, благодаря эффективной организации конвейерной обработки с использованием циклового режима и режима условного выполнения операций, описанной в разделах по устройству управления, "арифметическое" время может полностью перекрываться временем обработки команд в устройстве управления.

Таким образом, производительность процессора может характеризоваться суммарным временем прохождения команд программы через регистр команд устройства управления. Время такого прохождения (обработки) составляет 2 такта для команд с вычислением адресов по виртуальной памяти и 1 такт для команд с регистровыми адресами.

В качестве верхней практической оценки производительности процессора можно воспользоваться оценкой внутренних цик-

лов, рассматривавшихся примеров, достаточно характерных для задач, сводящихся к большим системам линейных алгебраических уравнений.

Основываясь на методах совмещения, проработанных на этапе эскизного проекта и цикловом режиме выполнения коротких циклов с упреждающей подкачкой по индексу цикла суммарное время выполнения внутренних циклов для Примера -I составляет 22 такта, а для Примера-2 - 7 тактов. За это время выполняется соответственно 15 и 5 команд, составляющих выполняемое тело внутренних программных циклов. Это соответствует среднему времени выполнения одной команды примерно за 1,4 такта. Для выполнения аналогичной по содержанию программы, записанной в командах машины БЭСМ-6, потребуется выполнить соответственно 22 и 7 команд, что соответствует среднему времени выполнения команды за 1 такт.

Такая скорость выполнения команд достигается за счет параллельного выполнения не только отдельных операций в арифметическом устройстве, но и параллельного выполнения операций управления. Так при умножении матриц из пяти операций программного цикла три операции (установка, умножение и сложение) выполняются в параллельных блоках арифметического устройства, а две операции - изменение индекса столбца и операция конца цикла, выполняются в устройстве управления параллельно с арифметическими операциями.

Описанные ранее возможности системы памяти с обобщенным уровнем сверхоперативной памяти для процессоров много-

процессорной системы и организация их параллельной работы позволяют сохранить такую высокую скорость работы при одновременной работе всех 4 процессоров стандартной 4-процессорной структуры БЭСМ-10 над одной задачей. Результирующее среднее время выполнения команды в такой 4-процессорной структуре приближается в этих случаях к 0,35 такта в командах АС-6 — БЭСМ-10 или к 0,25 такта в командах БЭСМ-6.

Принимая длительность такта равной 30 нсек можно считать верхней практической оценкой быстродействия одного процессора 23 млн. команд АС-6 + БЭСМ-10 в сек., что эквивалентно выполнению 33 млн. одноадресных команд БЭСМ-6 в секунду. Верхняя практическая оценка быстродействия 4-х процессорной структуры будет соответствовать в этих случаях примерно 90 млн. команд АС-6 + БЭСМ-10, что эквивалентно примерно 120 млн. одноадресных команд БЭСМ-6 в секунду.

Для рассмотренных примеров задач организация тесной параллельной работы процессоров над общими массивами одной задачи позволяет надеяться на резкое сокращение количества обменов с внешней памятью и, следовательно, сохранение высокой производительности для системы в целом.

Следует подчеркнуть, что такая скорость достигается на гибкой универсальной структуре с использованием лишь самых общих свойств широкого круга реализуемых алгоритмов, в отличие от систем с сильной специализацией структуры системы (ILLIAC IV, STAR-100, AS C-TI).

В процессе проектирования были рассмотрены и оценены примеры, характерные как для различных модификаций программ решения больших систем линейных алгебраических уравнений (работа с матрицами, содержащими малый процент ненулевых коэффициентов, организация массивов для областей с более сложными границами), так и для программ работы со сложными структурами данных. В последнем случае наличие в системе команд индивидуальной адресации байтов и разрядов в сочетании с операциями над последовательностями этих элементов позволяет многократно ускорить задачи анализа и обработки текстов, информации логического характера.

Все это позволяет считать, что производительность системы БЭСМ-10 в пересчете на один процессор для широкого диапазона решаемых задач будет соответствовать 10-15 млн. одноадресных команд БЭСМ-6 в сек. и достигать 20 млн. команд в сек. и выше в ряде важных областей применения.

## КОММУТАТОР КАНАЛА I-ГО УРОВНЯ

Коммутатор канала I-го уровня БЭСМ-10 (К1) является модулем, предназначенным для объединения высокопроизводительных абонентов системы, таких как центральные и периферийные процессоры, модули оперативной памяти, большие ЭВМ и внешние ЗУ с большими потоками информации. Это объединение производится посредством шин унифицированного дуплексного канала I-го уровня, идентичного каналу I-го уровня АС-6. Канал I-го уровня БЭСМ-10 обеспечивает:

- прямую адресацию шестнадцати абонентов системы;
- высокий темп передачи сообщений – 10±15 млн. сообщений в сек.;
- максимально допустимое удаление сопрягаемых абонентов – до 50 м;
- возможность подключения модулей системы АС-6.

Конструктивно коммутатор К1 выполнен в виде одного стандартного шкафа БЭСМ-10.

Блок-схема К1 приведена на рис.30. Как видно из этой схемы, коммутатор содержит семь независимых внешних направлений. Каждое такое направление состоит из блока приемного (входного) и блока выходного сопряжения. Узел коммутации обеспечивает автономные цепи связи каждого входа с каждым выходом. Управление коммутацией производится с учетом динамического приоритета источников запросов (входных блоков).

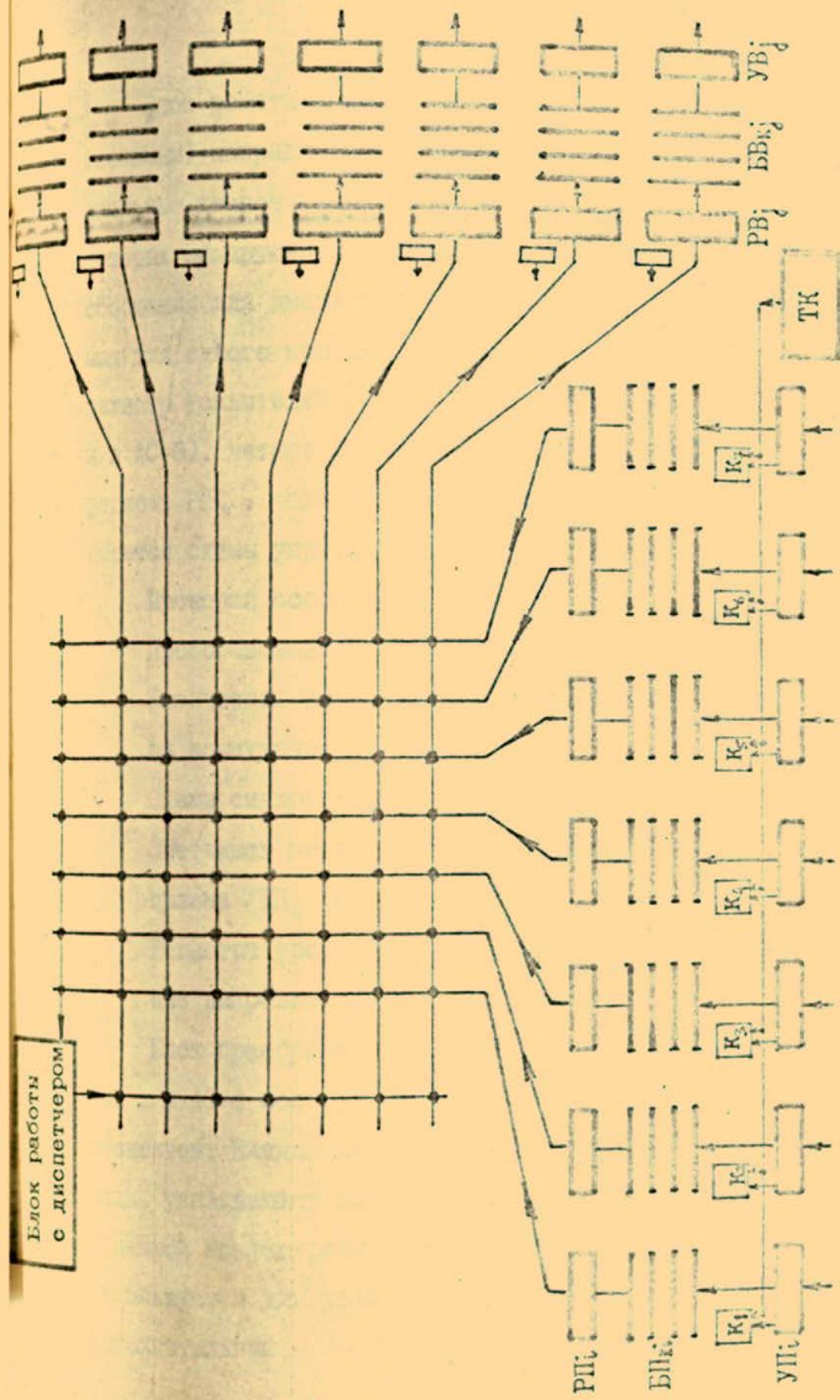


Рис. 30. Блок-схема компьютера панели I-го уровня

Блок работы с диспетчером является внутренним (восьмым) направлением коммутатора. К нему поступают специальные диспетчерские сообщения, а также сообщения, в которых обнаружены ошибки. В свою очередь из него выдаются сообщения для диспетчера системы. Блок приемного сопряжения для одного направления (см.рис. 31 ) содержит регистр усилителей приема сообщения УП<sub>1</sub> (84 разряда, как и в АС-6), четыре буферных регистра приема БП<sub>1</sub> (0÷3) и регистр РП<sub>1</sub>, подключаемый к узлу коммутации. Кроме того имеются схемы управления, которые состоят из блоков:

Проверки сообщения на четность;

Преобразования математического номера абонента в физический номер выхода коммутатора на основе таблицы коммутации (ТК);

Схемы синхронизации;

Счетчиков загрузки УЗП<sub>1</sub> (0÷3) и разгрузки буфера приема УРП<sub>1</sub> (0÷3);

Регистра требования коммутации на различные выходные направления Т<sub>1j</sub>.

Блок преобразования адресов ("схема приписки"), как и в АС-6 состоит из таблицы на 16 строк - по числу абонентов. Каждая строка таблицы содержит 3 двоичных разряда, указывающих номер выходного направления, к которому в данной конфигурации присоединен абонент. Нулевой код используется для указания отсутствия данного абонента. Дополнительный - 4-й разряд используется для контроля

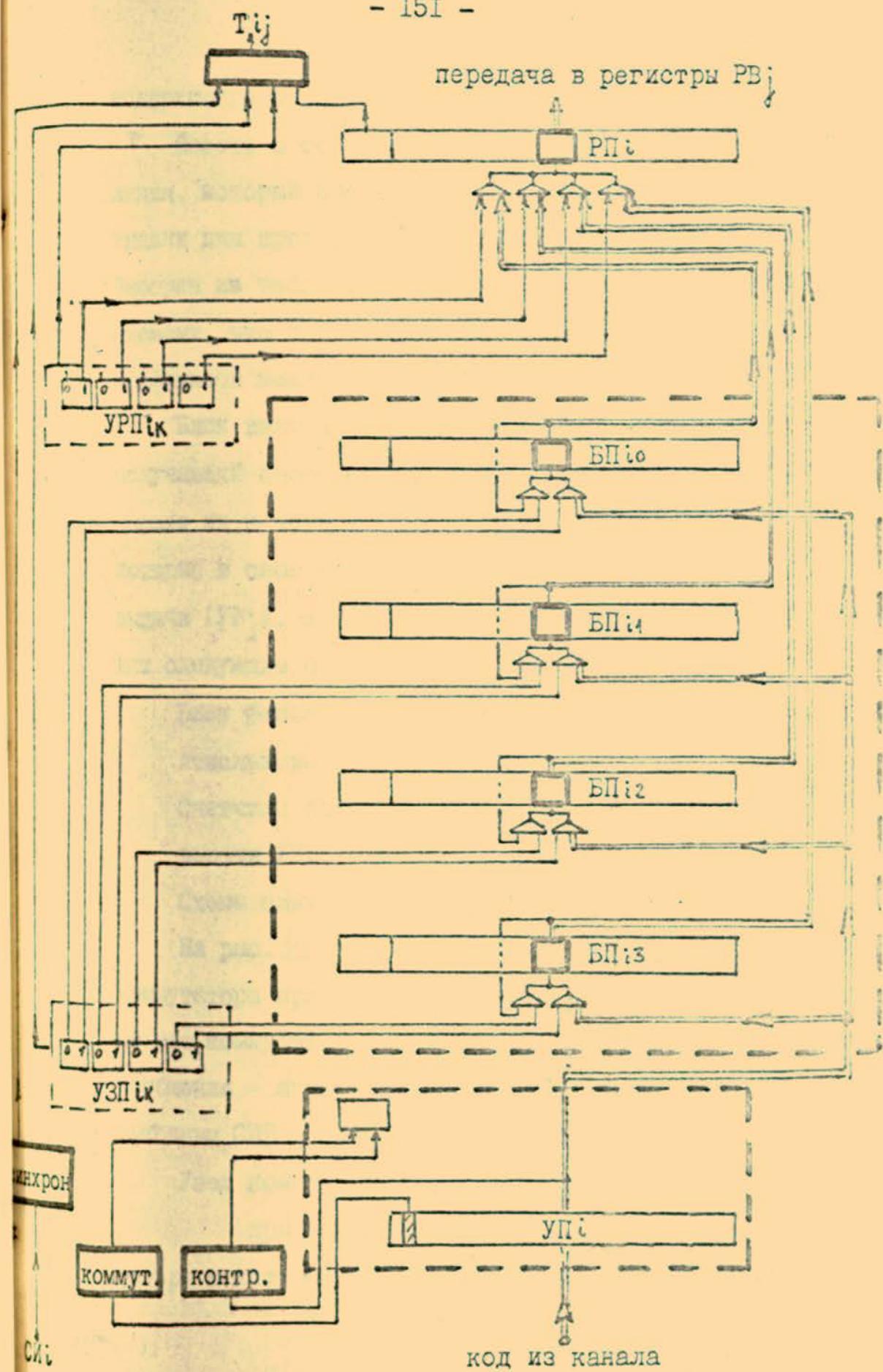


Рис. 31. Блок-схема буфера приема сообщений КИ

содержимого строки на четность.

Вместе с сообщением передается и код номера направления, который сравнивается с фиксированным кодом в блоке выдачи для проверки правильности коммутирующих цепей. Выборка из таблицы производится независимыми адресными схемами, что полностью исключает задержку коммутации из-за конфликтов между разными направлениями.

Блок выходного сопряжения содержит регистр ВР<sub>j</sub>, получающий сообщения из схем приемного сопряжения и подключающие их к четырем буферным регистрам выдачи БВ<sub>j</sub>(0+3), которые в свою очередь передают код на регистр усилителей выдачи (УВ<sub>j</sub>). Кроме того, в схемы управления выдачей входят следующие блоки:

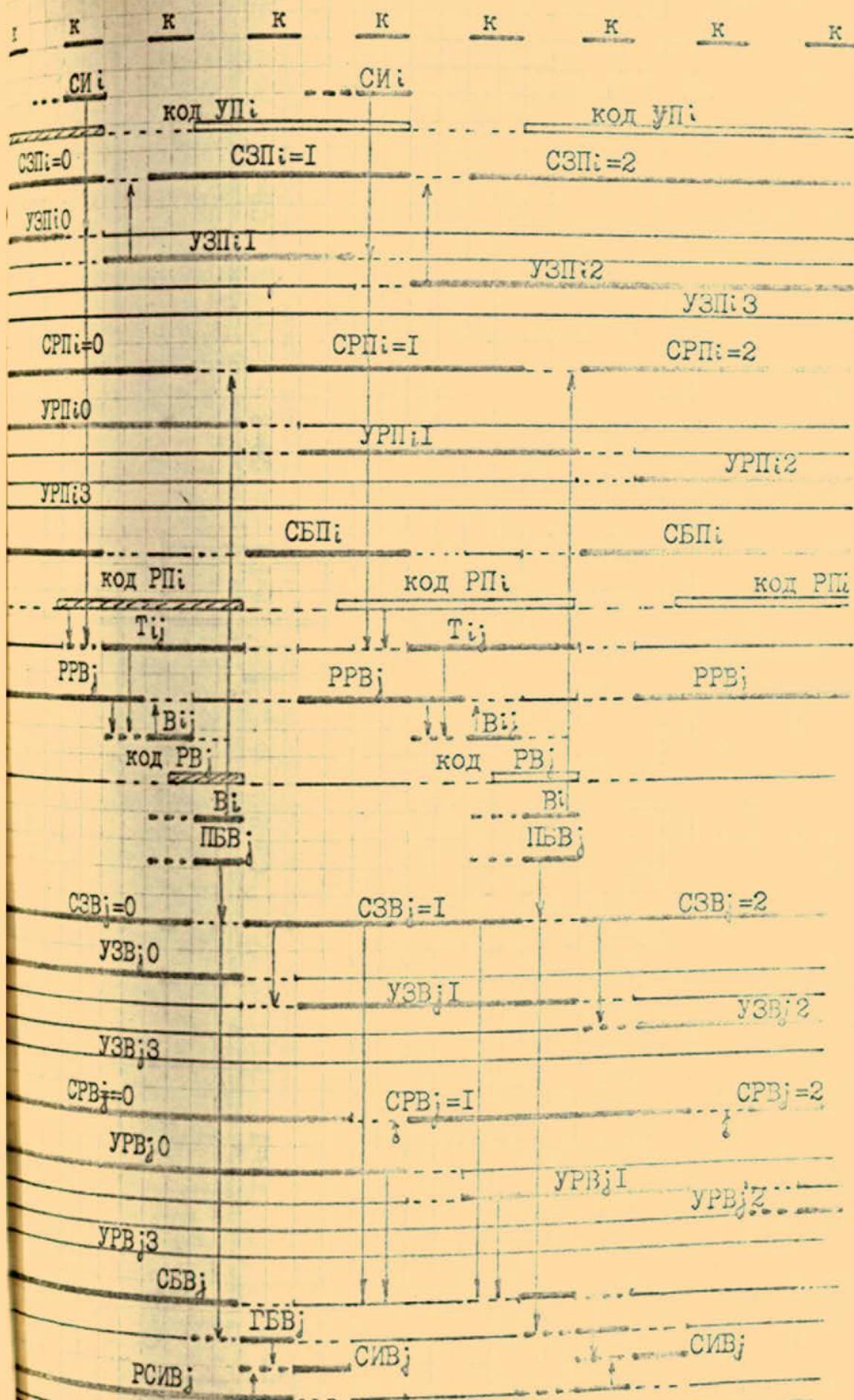
Блок фиксации динамического приоритета, аналогичный использованному в К1 АС-6;

Счетчики загрузки УЗВ<sub>j</sub>(0+3) и разгрузки буфера выдачи УРВ<sub>j</sub>(0+3);

Схемы синхронизации и управления выдачей в канал.

На рис. 32 приведена временная диаграмма работы коммутатора при передаче нескольких сообщений. Как видно из нее коммутатор обеспечивает высокую скорость передачи сообщения — интервал между входным синхроимпульсом СИ<sub>j</sub> и выходным СИВ<sub>j</sub> составляет 2,5 такта.

Узел коммутации имеет пропускную способность 2 такта при передаче на каждый выход. Всего же в коммутаторе может быть до восьми таких одновременных передач.



Временная диаграмма работы коммутатора К1

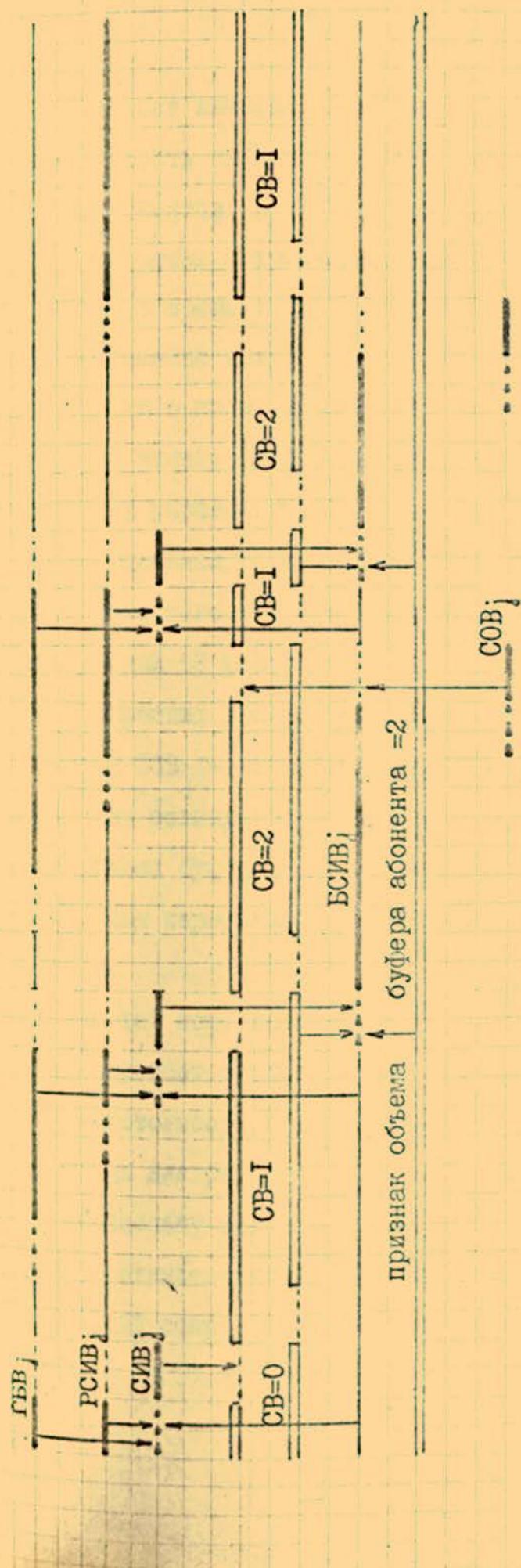


Рис. 33.

Временная диаграмма выдачи из коммутатора канала I-го уровня / пример для удаленного абонента с двумя выходными буферами регистрами /

В блоке выходного сопряжения КI применена схема, позволяющая ускорить передачу пачки сообщений при значительном удалении абонентов. При большом расстоянии между абонентами ( $\sim 50\text{м}$ ) пробег сигналов синхронизации по кабелю требует значительного времени ( $\sim 500$  нсек), что в случае стартстопной работы заметно снижает скорость передачи. Если использовать информацию о наличии буферных регистров у абонента-приемника, то можно значительно повысить эффективность обмена. Для этого в блок управления КI введен регистр объема буфера абонента-приемника (при каждом выходе), загружаемый диспетчером. Кроме того, в блоке управления выдачей СИВ<sub>j</sub> помещен реверсивный (3 разрядный) счетчик выдачи СВ<sub>j</sub>. Любая выдача СИВ производит прибавление единицы, а каждый сигнал ответа абонента СОВ<sub>j</sub> - вычитает единицу. Усилитель БСИВ<sub>j</sub> фиксирует факт совпадения кода счетчика с заданным значением признака объема буфера. Установка БСИВ в единицу означает, что дальнейшая передача приостановлена до получения ответного сигнала.

До тех пор, пока БСИВ находится в нуле выходное сопряжение передает сообщения по собственным условиям (РСИВ<sub>j</sub> и ГБВ<sub>j</sub> - готовность буфера выдачи). На рис. 33 приведена временная диаграмма для случая двух буферных регистров.

Поскольку код объема буфера, равный нулю, означает наличие единственного буфера, то в этом случае происходит простейший стартстопный обмен, как в АС-6.

## ПЕРИФЕРИЙНАЯ СИСТЕМА

### ВВЕДЕНИЕ

Периферийная система БЭСМ-10 полностью совместима с периферийной системой АС-6. Совместимость обеспечена на всех уровнях иерархии программ и аппаратуры, а именно:

1. Совместимость программ центрального процессора, связанных с управлением периферийной системой.
2. Совместимость программ периферийной машины (программ периферийного процессора и каналлера).
3. Совместимость системы команд периферийного процессора и каналлера.
4. Совместимость модулей аппаратуры (периферийной машины, мультиплексоров, адаптеров и т.д.).

5. Совместимость каналов для связи модулей (каналы I-го и 2-го уровня, сопряжение между мультиплексорами и адаптерами, интерфейс ввода-вывода и т.п.).

Эта совместимость вместе с модульным построением аппаратуры и программ обеспечивает взаимозаменяемость аппаратуры, поэтапность развития конкретных установок и упрощает дальнейшее развитие как аппаратуры, так и программного обеспечения системы.

Исключительное значение для потребителей БЭСМ-6 и АС-6 имеет возможность поэтапного развития. Например, на первом этапе модернизации систем может быть введен центральный процессор БЭСМ-10 при сохранении всего периферийного оборудова-

ния системы АС-6. Последующие этапы могут быть связаны сначала с дополнением системы модулями БЭСМ-10 (ПМ-10, К2, МТФ, МТГ, МПС системы БЭСМ-10), а затем и полной заменой модулей АС-6 на новые модули. Число таких этапов и их содержание зависят от потребностей пользователей. Периферийные модули БЭСМ-10 в отличие от аналогичных модулей АС-6 будут обладать более высокой производительностью и надежностью. Например, периферийный процессор БЭСМ-10 проектируется на скорость до 1,3 млн. операций в сек. (Периферийный процессор АС-6 выполняет 200 тыс. операций в сек.). Кроме того, за счет введения собственной памяти повышается производительность всей периферийной машины, сокращается взаимное влияние центральной и периферийной частей системы.

Основные принципы построения периферийной системы, ее состав, особенности периферийной машины ПМ-10, методика подключения абонентов различных типов (внешних устройств, высокоскоростных внешних запоминающих устройств, телефонных и телеграфных каналов связи) достаточно подробно освещены в аванпроекте вычислительной системы БЭСМ-10.

Подробное описание системы команд периферийной машины ПМ-6, полностью совместимой с системой команд ПМ-10, содержится в выпущенном ранее описании "Система обработки данных АС-6. Периферийная машина ПМ-6. Принципы работы и система команд".

В связи с этим дальнейшее описание связано в основном с вопросами аппаратной реализации периферийной машины ПМ-10 и других модулей системы.

## СТРУКТУРА ПЕРИФЕРИЙНОЙ МАШИНЫ ПМ-10

Периферийная машина состоит из нескольких относительно крупных узлов, работающих одновременно.

Основными узлами являются:

Адаптер канала I-го уровня;

МОЗУ, собственная оперативная память;

Периферийный процессор;

Каналлер;

Буфер директив процессора;

Буфер инструкций и собственные регистры;

Блок управления с пульта и система индикации.

Необходимые связи между узлами достаточно эффективно реализованы при помощи мультиплексирования обменов. Роль мультиплексора играет узел обмена. Использование такого общего узла характерно для многих устройств БЭСМ-6 и АС-6 и соответствует принятой системе элементов. Вероятное время занятости узла обмена равно трем тактам.

Предлагаемая структура хорошо обеспечивает следующие возможности:

Имитацию в канале I-го уровня ("заворачивание сообщения")

Автономную наладку с панели управления.

Загрузку собственных регистров - с панели управления, от центрального процессора по каналу I-го уровня или от периферийного процессора.

Выполнение в процессе отладки чтения или записи в ячейку памяти.

Индикацию при обращении к ячейке или регистру (слежение за содержимым ячейки).

Возможность выполнения подирективных обменов, что

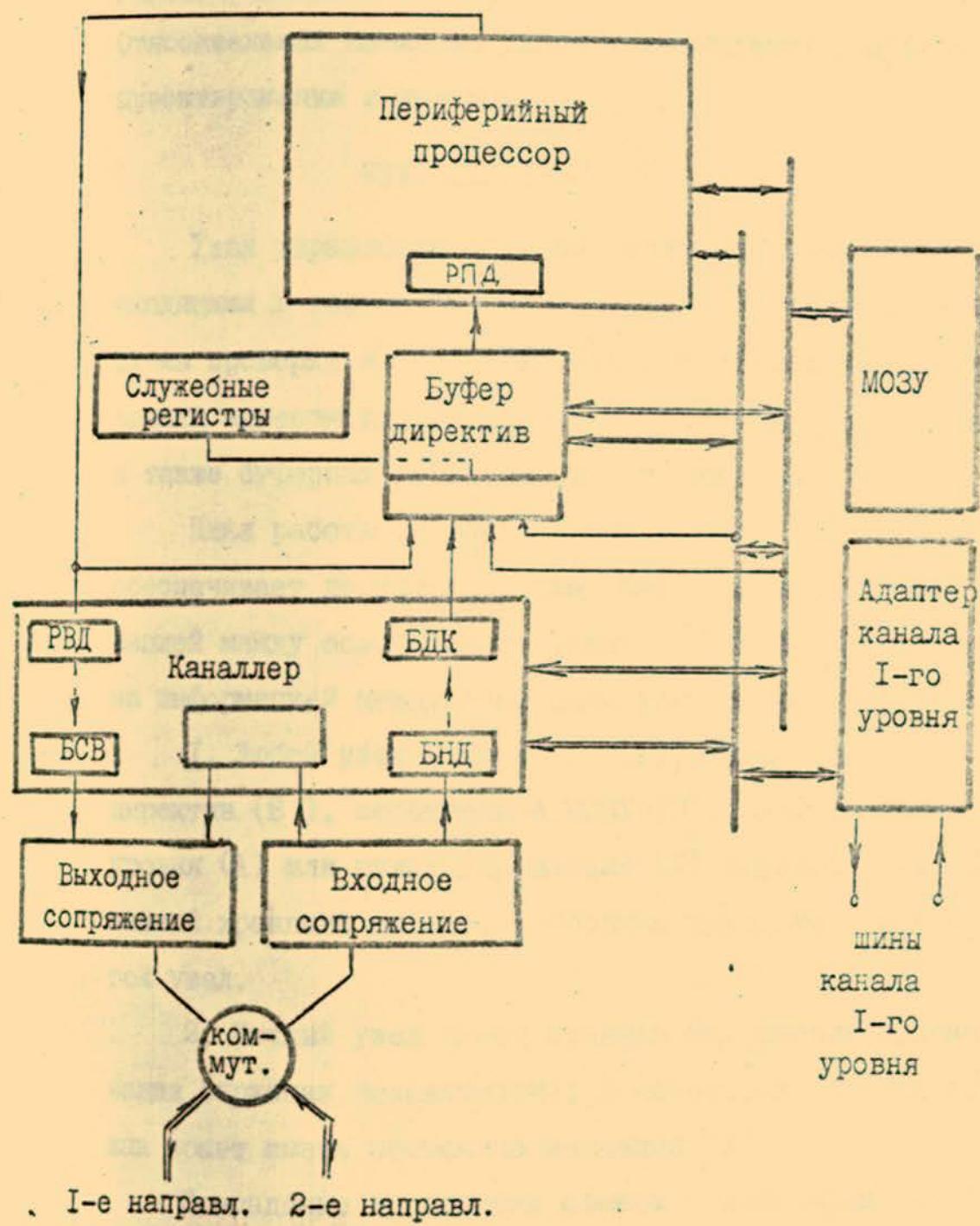


Рис. 34. Блок-схема ПМ-10

упростит связи с ПМ-10.

Относительная изоляция узлов обеспечивает раздельное их проектирование и наладку.

### УЗЕЛ ОБМЕНА

Узел управления обменом между основными устройствами, входящими в состав ПМ-10, состоит из обменного регистра (ОР), схемы проверки на четность, схем совпадения с указанным на панели адресом для останова процессора в режиме отладки, а также буферных регистров индикации БИ1 и БИ2.

Цикл работы ОР равен трем тактам , что обеспечивает достаточную для ПМ-10 скорость обмена информацией между основными ее узлами. Принцип организации обмена информацией между основными узлами состоит в следующем:

1. Любой узел - каналлер (К), процессор (П), буфер директив (Б ), собственное МОЗУ (М), адаптер канала I-го уровня (А) или пульт управления (У) вырабатывает тактируенный хранящий сигнал требования передачи в некоторый другой узел.

2. Каждый узел имеет признак разрешения приема информации (признак незанятости); в некоторых случаях этот признак может иметь постоянно значение "1".

Совпадение требования обмена с некоторым узлом с условием разрешения от этого узла вызывает появление сигнала готовности к обмену.

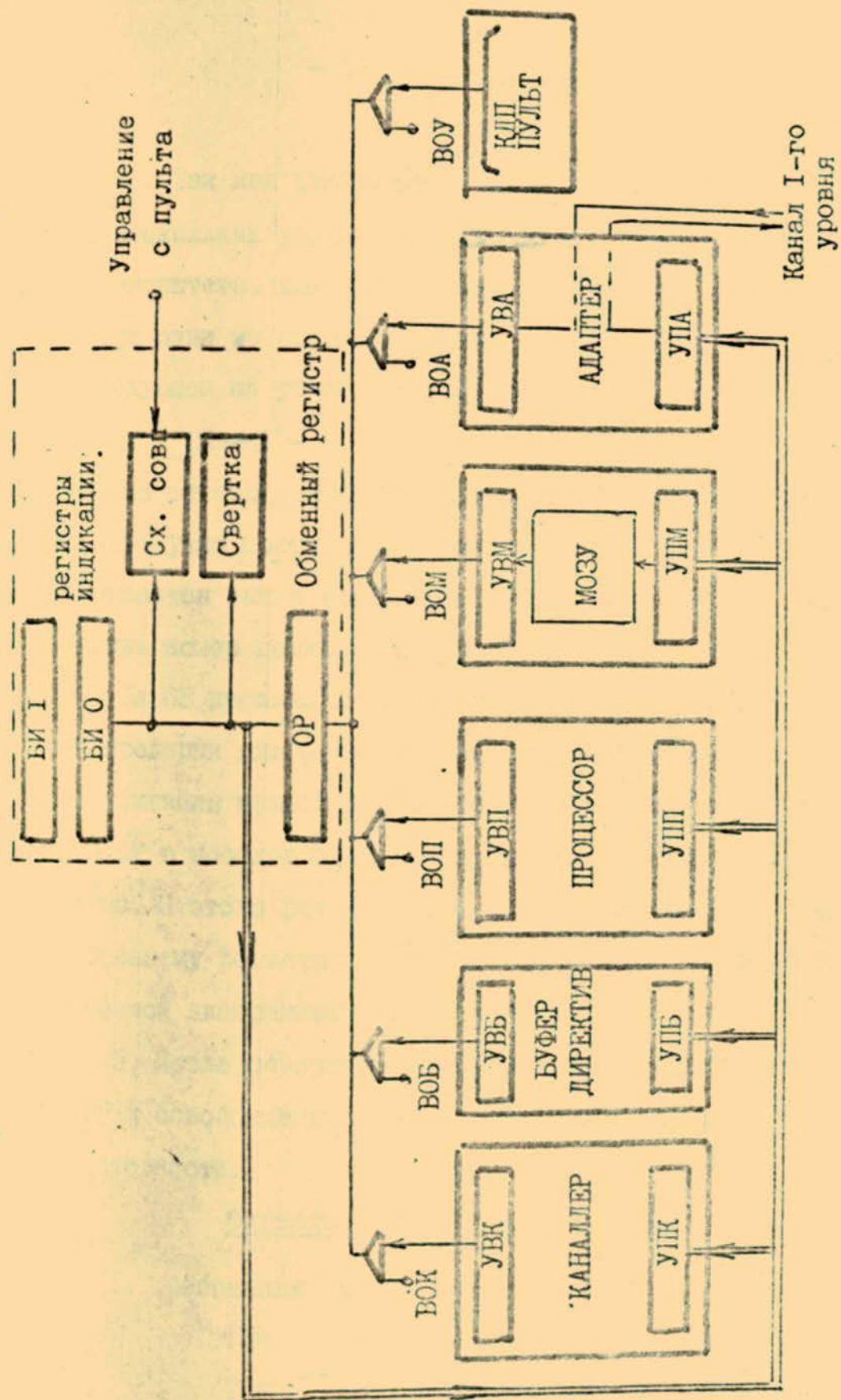


Рис. 35. Узел обмена ПМ-10

3. Так как готовность может возникнуть одновременно для нескольких узлов, то эти сигналы рассматриваются в схеме приоритета, так что в каждые три такта может вырабататься только один из сигналов выборки. Приоритет узлов фиксирован в следующем по убыванию порядке:

M → A → K → B → P → Y

4. По сигналу выборки информация с регистра выдачи выбранного узла поступает в обменный регистр. На этот регистр принимается также номер узла, отправившего эту информацию, а также номер подканала, с которым связан обмен.

На ОР производится контрольная свертка на четность для проверки или формирования признаков контроля сообщения.

Сигналы приема в узел-адресат обеспечивают появление кода ОР в необходимом узле - на соответствующем регистре приема. С этого регистра код может быть передан необходимому внутреннему регистру узла. Момент приема вырабатывается в соответствии с временной диаграммой самого узла-адресата.

5. После передачи информации происходит гашение кода на ОР и освобождение его для обслуживания следующего сигнала готовности.

#### Сигналы требований на обмен:

##### I. Требования от процессора:

ТИМ - к МОЗУ,

ТИА - к адаптеру канала I-го уровня,

ТИКД - к каналлеру для выдачи директивы,

ТИБ - к буферу директив,

2. Требования от каналлера:

ТКМ - к МОЗУ

ТКА - к адаптеру,

ТКБ - к буферу директив,

3. Требования от буфера директив:

ТБМ - к МОЗУ,

ТБА - к адаптеру,

4. Требования от МОЗУ:

ТМА - к адаптеру,

ТМ - к прочим узлам,

5. Требования от адаптера:

ТАМ - к МОЗУ,

ТАК - к каналлеру (прием слова в буфер данных или управляющего слова),

ТАД - к каналлеру для передачи директив в РВД (вспомогательный режим),

ТАБ - к буферу директив,

ТАП - к процессору

Аналогичнорабатываются и требования от пульта управления ИМ-10.

Сигналы готовности к обмену:

ГМ = ТМ + ТМА·РА , где РА - разрешение адаптера

ГА = ТАК + ТАД·РРВД + ТАМ·РМ + ТАП + ТАБ·РБ,

ГК = ТКМ·РМ + ТКА·РА + ТКБ·РБ,

ГБ = ТБМ·РМ + ТБА·РА,

ГП = ТПМ·РМ + ТПА·РА + ТПБ·РБ + ТПКД·РРВД,

ГУ = ТУМ·РМ + ТУА·РА + ТУБ·РБ + ТУКД·РРВД + ТУПД·РРПД,

Эти сигналы воспринимаются обменным регистром при условии его незанятости  $POP = 1$ . Этот сигнал служит для гашения самого обменного регистра. В результате срабатывают сигналы выборки узла с учетом его приоритета:

$$BOM = \overline{POP} \cdot \overline{IM}$$

$$BOA = \overline{POP} \cdot \overline{IM} \cdot \overline{GA}$$

$$BOK = \overline{POP} \cdot \overline{IM} \cdot \overline{GA} \cdot \overline{GK}$$

$$BOB = \overline{POP} \cdot \overline{IM} \cdot \overline{GA} \cdot \overline{GK} \cdot \overline{GB}$$

$$BOP = \overline{POP} \cdot \overline{IM} \cdot \overline{GA} \cdot \overline{GK} \cdot \overline{GB} \cdot \overline{PI}$$

$$BOU = \overline{POP} \cdot \overline{IM} \cdot \overline{GA} \cdot \overline{GK} \cdot \overline{GB} \cdot \overline{PI} \cdot \overline{GU}$$

Срабатывание этих усилителей гасит  $POP$ , обеспечивая хранение кода в ОР. Сигнал цепочки управления ЦОР вновь освобождает этот узел через три такта.

$$POP = \overline{POP} \cdot \overline{BOU} \cdot \overline{BOA} \cdot \overline{BOK} \cdot \overline{BOB} \cdot \overline{BOP} + \text{ЦОР}$$

На рис. 36 приведен пример временной диаграммы обслуживания запросов узлом обмена.

Как видно из диаграммы прием некоторого запроса запускает цепочку однотактных сигналов, причем происходит гашение как обслуженного требования, так и признака разрешения в узле, принявшем информацию. Восстановление этого признака происходит после того, как информация принята какими-то внутренними схемами узла-приемника.

Сигналы подключения ОР к узлу, принимающему информацию, вырабатываются на основе требования и выборки:

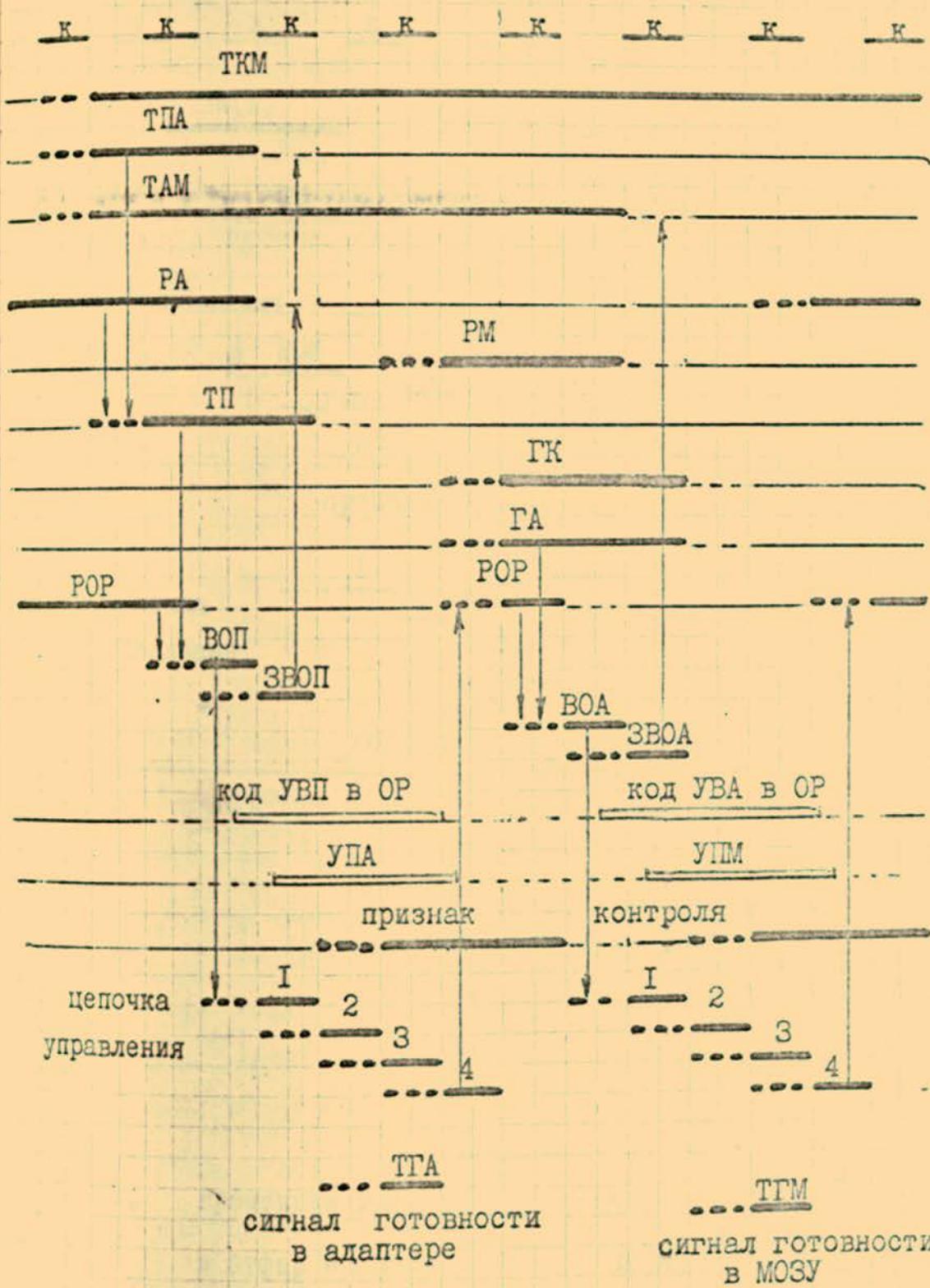


Рис.36. Временная диаграмма обслуживания  
двух запросов в узле обмена ПМ-10

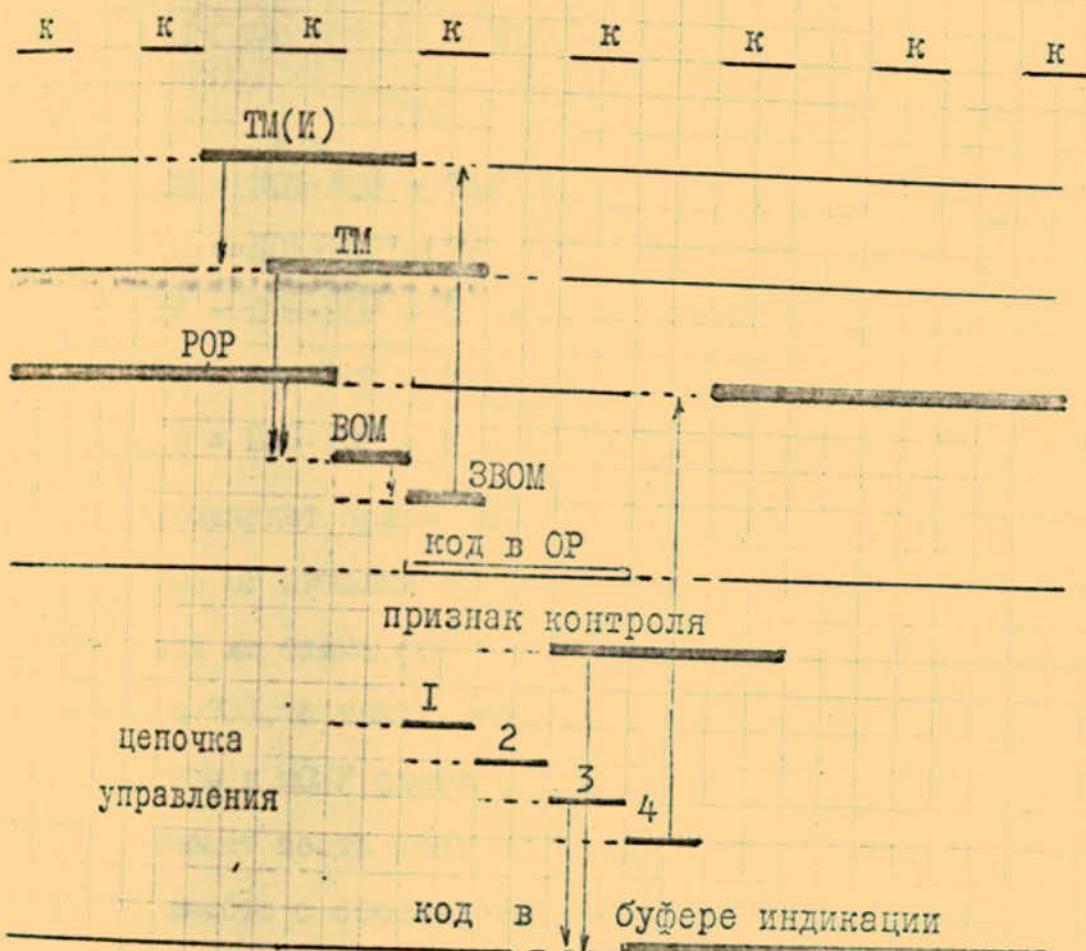


Рис. 37. Временная диаграмма приема информации  
в буфер индикации блока обмена.

ПОМ = ПОМ·POP + ТАМ·ВОА + ТКМ·ВОК + ТБМ·ВОВ + ТИМ·ВОП,  
ПОЛ == ПОА·POP + ТМА·ВОМ + ТКА·ВОК + ТБА·ВОВ + ТИА·ВОП,  
ПОК = ПОК·POP + ТАК·ВОА + ТАД·ВОА + ТМК·ВОМ + ТИК·ВОП,  
ПОП = ПОП·POP + ТАП·ВОА + ТМП·ВОМ + ТУПД·ВОУ,  
ПОВ = ПОВ·POP + ТАБ·ВОА + ТМБ·ВОМ,

Если происходит прием информации из канала I-го уровня, то на уровне ОР выполняется проверка сообщения. При выдаче в канал эти же схемы формируют необходимые контрольные разряды (82 и 73), а также контрольные разряды слова (49 и 50). При записи в МОЗУ схема формирования контрольных разрядов вырабатывает шесть контрольных разрядов байтов слова, передавая их вместе с сообщением в МОЗУ.

В узле управления обменом производится сравнение адреса, попавшего в ОР, с кодом, набранным в специальном регистре на пульте ПМ-IO.

При соответствующем указании оператора блокируется смена кода на обменном регистре (не устанавливается POP) и его содержимое выдается на один из регистров индикации.

Работа после останова может быть продолжена после нажатия пусковой кнопки.

Ввиду невозможности вывода на индикацию всех хранящихся элементов ПМ-IO, в различных узлах устанавливаются дополнительные регистры для индикации наиболее существенных сигналов. Кроме этого, в узле обмена помещается один-два буферных регистра для общих целей индикации. Возможны два режима их выключения. В первом регистр индикации адресуется непосредственно как один из регистров получателей информации. Например,

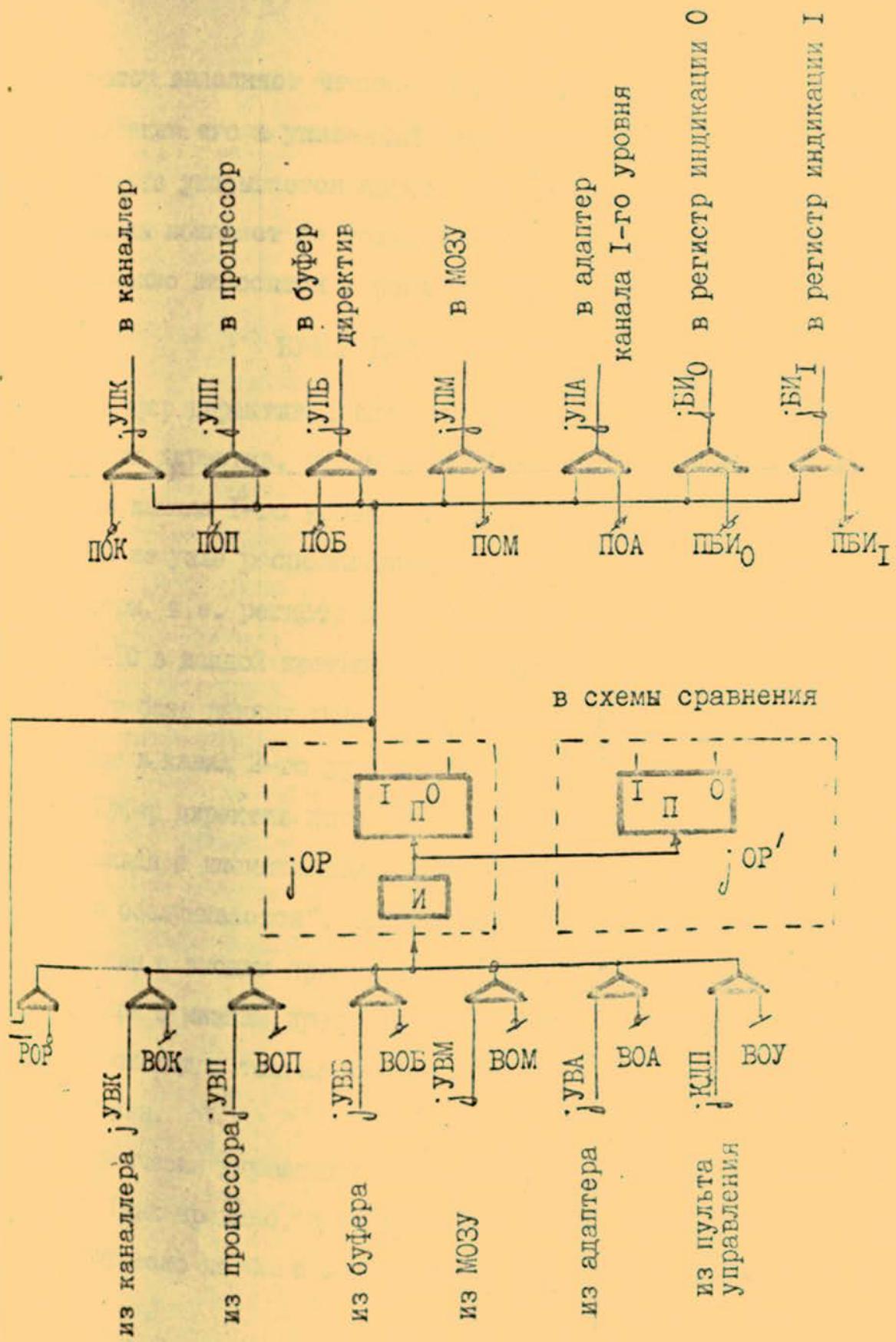


Рис. 38. Массовые цепи обменного регистра ПМ-10

оператор выполняет чтение содержимого некоторой ячейки с помещением его в указанный буфер индикации. В другом режиме на пульте указывается адрес, при совпадении с которым информация попадает не только к узлу-получателю, но и одновременно заносится в регистр индикации.

### БУФЕР ДИРЕКТИВ

Буфер директив процессора предназначен для временного хранения директив, поступающих из различных узлов ПМ-10 или из канала 1-го уровня на вход периферийного процессора. В этом же узле располагаются так называемые "служебные" регистры, т.е. регистр системного номера РСН, присвоенного ПМ-10 в данной конфигурации каналов 1-го и 2-го уровня, регистр базы данных монитора, регистр таблицы коммутации выходов в канал 2-го уровня.

Буфер директив логически представляет собой две очереди, каждая с дисциплиной обслуживания "первым пришел - первым обслуживается". Одна из очередей ("быстрая") хранит директивы с высшим приоритетом, другая ("медленная") - директивы с низким приоритетом. Разгрузка медленной очереди происходит только тогда, когда быстрая полностью разгружена.

Физически директивы помещаются в служебные страницы памяти (как правило, в собственном МОЗУ), положение которых зафиксировано по базе данных монитора (БДМ). Как и в ПМ-6,

массив буфера директив организован циклически, т.е. после заполнения его последней ячейки страницы будет загружена первая.

Положение каждой очереди в массиве определяется значением двух ее границ: нижней, по которой производится загрузка директивы, и верхней, по которой производится разгрузка, т.е. считывание директивы в процессор. Регистры границ подключаются в качестве младших разрядов адреса, по которому производится обращение в память. Старшими разрядами служит префикс - база данных монитора.

Аппаратура узла буфера директив состоит из следующих блоков (рис. 39):

регистра приема УПБ;

регистра выдачи УВБ;

регистров границ очередей;

блока управления записью и чтением директив, а также контроля за границами очередей;

блока служебных регистров.

#### Загрузка информации в служебные регистры ПМ-Ю

При поступлении из канала I-го уровня специальных ("диспетчерских") сообщений, а также при выполнении специальных команд в процессоре вырабатываются требования к буферу ТАБ или ТИБ, причем в адресе содержится признак загрузки и адрес служебного регистра, а в слове - код, который надо загрузить. Следует отметить, что должны быть

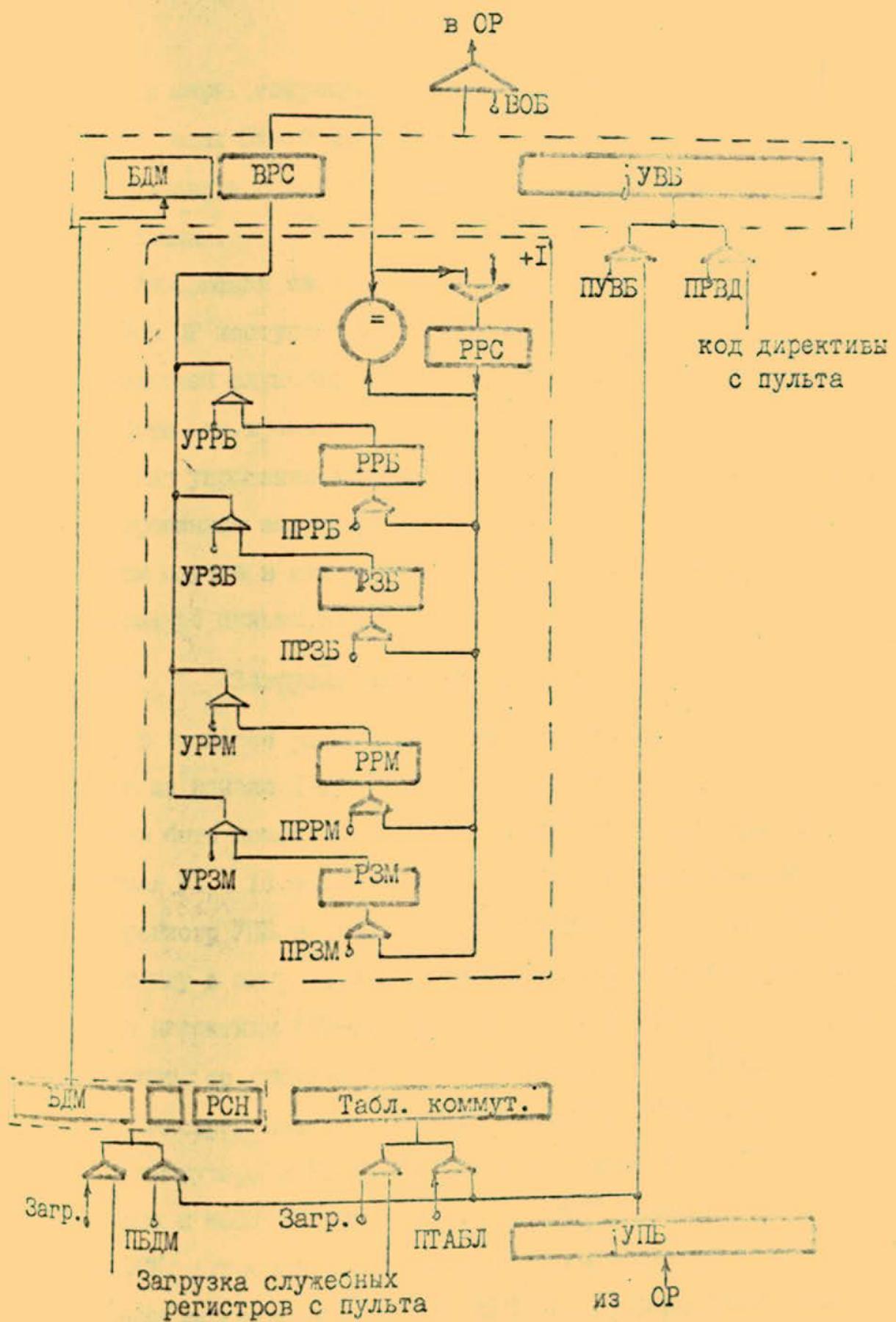


Рис. 39. Блок-схема буфера директив ПМ-10

приняты меры, запрещающие использование этих регистров в других узлах ПМ-IO во время загрузки (например, необходимо приостановить обмен в канале 2-го уровня при смене таблицы коммутации).

Информация из адаптера или процессора через общий регистр ОР поступает на входной регистр буфера УПБ, а с него — в указанный служебный регистр, после чего буфер вновь готов к работе (см. временную диаграмму на рис. 40 ).

Для упрощения управления системой предусматривается фиксированная загрузка служебных регистров по установке нуля. В этом случае в них заносится код, набранный на специальной клавиатуре пульта ПМ-IO.

#### Загрузка директив в буфер

В процессе работы ПМ-IO из каналлера, процессора, а также из канала I-го уровня поступают директивы, которые должны быть помещены в буфер. При разрешении приема (сигнал РБ = 1) эти директивы через ОР принимаются на входной регистр УПБ и устанавливается требование поместить эту директиву в одну из двух очередей. Тип очереди определяется кодом директивы (20-ый разряд). Если выходной регистр УВБ не занят, то производится передача кода с УПБ в УВБ.

Одновременно вырабатывается требование обращения со стороны буфера к МОЗУ или в канал I-го уровня. Признак обращения к МОЗУ — это совпадение старших разрядов БДМ с кодом РСН. Адресная часть регистра УВБ образована кодом БДМ и входным регистром-счетчиком ВРС, к которому подключается

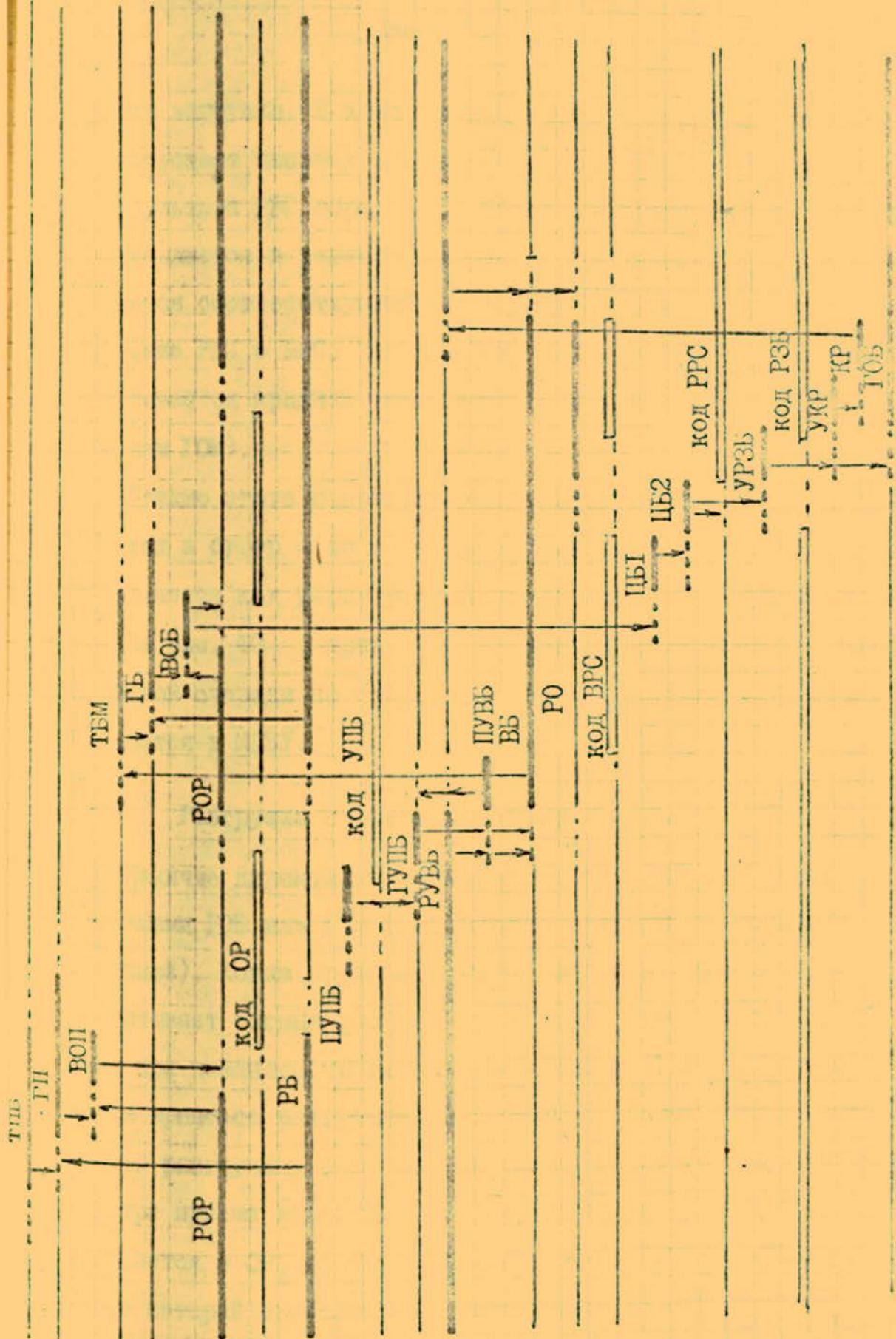


Рис. 40. Временная диаграмма загрузки "быстрой" очереди

регистр загрузки. В ходе приема этого требования сигнал ВОБ запускает цепочку окончания работы. При этом на регистре результата РРС образуется новое значение границы очереди и пересыпается в регистр загрузки. Одновременно к ВРС подключается соответствующий регистр разгрузки и производится сравнение РРС с ВРС. Так как эти коды не совпадают, то устанавливается признак готовности очереди к разгрузке (ГОБ или ГОМ).

Помимо этого основного входа предусмотрен дополнительный вход в буфер — по сигналу с пульта. При этом на УВБ подключается код регистра выдачи директивы с пульта (ПРВД).

На рис. 40 приведена временная диаграмма загрузки медленной очереди со стороны процессора, причем очередь помещается в МОЗУ.

#### Разгрузка очереди директив

Наличие директивы в какой-либо очереди определяется признаками ГОБ или ГОМ (готовность очереди быстрой или медленной). Когда процессор готов принять директиву, он вырабатывает сигнал разрешения РРП. По этим условиям, а также при незанятости выходного регистра УВБ происходит запуск процесса разгрузки. При этом к адресу подключается регистр разгрузки, а в разряды слова сигналом ПАРП (адрес регистра приема директивы) РПД процессора. Когда запрос принимается в ОР, срабатывает цепочка окончания разгрузки в ходе которой прибавляется единица к верхней границе очереди и производится сравнение верхней и нижней границ.

При совпадении происходит гашение признака ГОБ (или ГОМ).

На рис. 41 приведена временная диаграмма разгрузки быстрой очереди.

Директива после считывания из очереди автоматически поступит в процессор, так как в качестве обратного адреса назначен РПД.

### ПЕРИФЕРИЙНЫЙ ПРОЦЕССОР

#### Основные принципы реализации

Выбор оптимальной производительности периферийной машины ПМ-IO и, в частности, периферийного процессора определяется следующими соображениями.

Во-первых, функции обмена и предварительной обработки данных для большого числа внешних абонентов достаточно легко могут быть распараллелены с помощью двух и более периферийных машин.

Во-вторых, достижение максимально возможной производительности периферийного процессора возможно только за счет таких структурных решений, как буферизация при выборке команд и операндов и магистральный принцип выполнения операций. Однако при этом снижается эффективность процессора в условиях относительно частых прерываний.

Поэтому периферийный процессор имеет оптимальную производительность, достижимую при сравнительно простой и эффективной при частых прерываниях структуре и обеспечивающую обслуживание большого числа сравнительно медленных абонентов

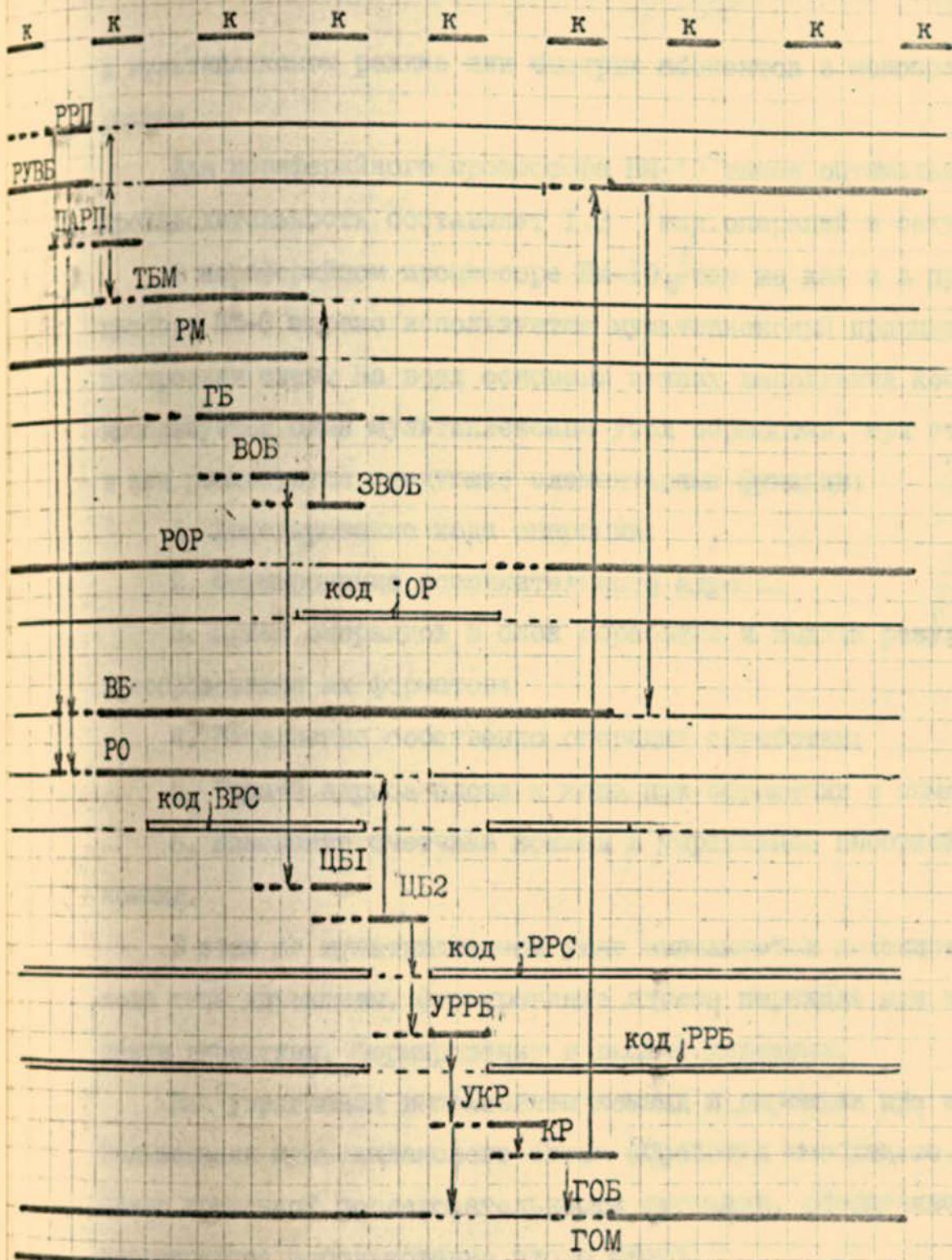


Рис.41. Временная диаграмма разгрузки директивы из "быстрой" очереди

в мультиплексном режиме или быстрых абонентов в монопольном режиме.

Для периферийного процессора ПМ-10 такая оптимальная производительность составляет 1,3 млн. операций в секунду.

В периферийном процессоре ПМ-10, так же как и в процессоре ПМ-6 широко используется мультиплексный принцип построения схем. На всех основных этапах выполнения команд используется один мультиплексный узел обработки, при этом в нем реализуются следующие элементарные функции:

1. Декодирование кода операции;
2. Формирование исполнительного адреса;
3. Прием операндов в блок обработки и выдача результата, преобразование их форматов;
4. Выполнение собственно операции обработки;
5. Выдача адреса слова и кода при обращении в память;
6. Изменение счетчика команд и управление выборкой команд.

В этом же мультиплексном узле выполняется декодирование кода типа директивы, формирование адреса перехода при выполнении директивы, формирование и выдача директив.

Для управления выполнением команд и директив при использовании мультиплексного блока обработки необходимо создание временной последовательности сигналов, обеспечивающей многократное использование этого блока.

Так же как и в ПМ-6, в процессоре ПМ-10 используется принцип многоуровневого управления выполнением команд и директив.

Схемы формирования временной последовательности разбиты на три уровня в соответствии с их назначением и скоростью переключения.

Первый уровень этих схем обеспечивает формирование последовательности однотактных сигналов, при этом общая длительность этой последовательности сигналов соответствует длительности переключения схем второго уровня. В течение времени работы последовательности сигналов первого уровня выполняется некоторая элементарная операция пересылки кода с одного из регистров процессора в блок обработки (например, прием операнда) или обратно (выдача результата). Кроме пересылки при этом производится преобразование форматов кодов, участвующих в пересылке. Такая элементарная операция названа нанооперацией. Каждой нанооперации соответствуют сигналы, сохраняемые в течение всего времени ее выполнения.

Для формирования сигналов нанооперации используется по два запоминающих усилителя, соединенных по схеме регистра сдвига (основной усилитель нанооперации и задержанный).

При переходе с одной нанооперации на другую задержанный усилитель предыдущей нанооперации устанавливает следующую нанооперацию.

Несколько наноопераций объединяются для выполнения законченной, достаточно сложной функции, называемой микроперацией. Такой функцией является, например, формирование

исполнительного адреса в командах обращения в память и передачи управления, обработка операндов, включая прием их с регистров и запись результата, выборка командного слова и т.д. Разбиение последовательности выполнения команд на нанооперации и микрооперации с учетом их функционального назначения обеспечивает модульную структуру алгоритмов выполнения команд. Последнее обстоятельство позволит упростить разработку и наладку процессора.

Особый класс операций, для которых в первую очередь предназначен блок обработки, являются арифметические и логические операции, выполняемые в командах обработки, и операции сложения и вычитания, выполняемые при формировании адресов. По своему месту в последовательности выполнения команды они относятся к нанооперациям, однако, методика создания сигналов первого уровня в этом случае отличается от методики, используемой в нанооперациях. В качестве сигналов первого уровня кроме специальных сигналов запуска операции используются сигналы, управляющие передачами кодов на регистрах.

#### Структура процессора

Основными блоками периферийного процессора (рис.42) являются блок регистров, блок обработки, блок формирования временной последовательности, блок исполнительных усилителей, блок связи с обменным регистром и блок преобразования адресов.

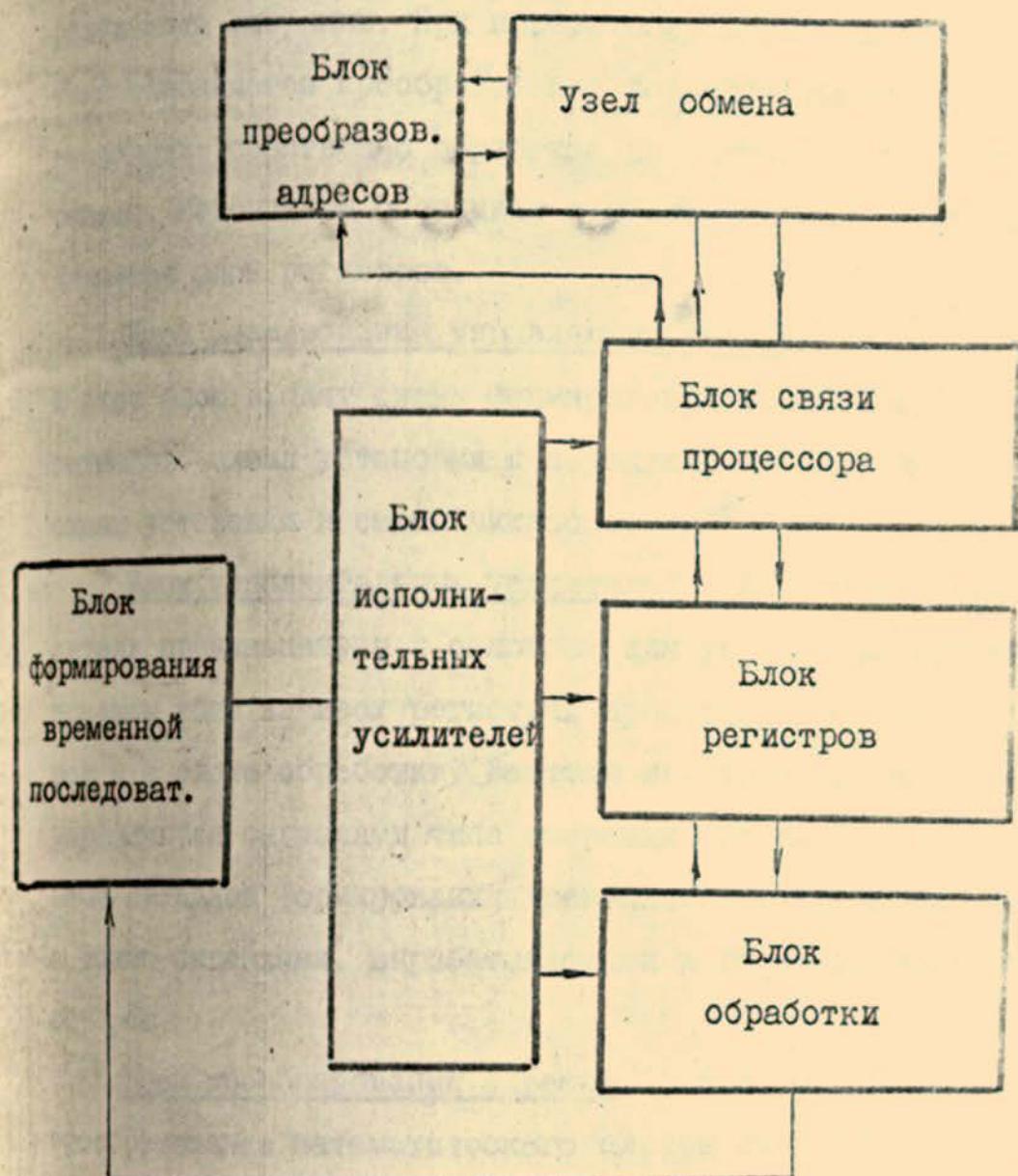


Рис.42. Блок-схема периферийного процессора ПМ-ІО

Блок обработки. В этом блоке производится прием операндов, выполнение собственно операции обработки и выдача результата операции. При приеме операндов и выдаче результата выполняется преобразование форматов таким образом, что выполнение собственно обработки не зависит от формата операндов. Источником операндов и местом записи результата является блок регистров.

Блок формирования управляющих последовательностей. В этот блок входят схемы формирования цепочки однотактных сигналов, схемы установки и переключенияnanoопераций и схемы установки и смены микроопераций.

Блок исполнительных усилителей. Исполнительные усилители предназначены в основном для управления хранением и приемом кода на всех регистрах процессора - в блоке регистров и в блоке обработки. Вентили исполнительных усилителей управляются сигналами типа операции, признаков и тремя уровнями сигналов формирования временной последовательности, а также сигналами, вырабатываемыми в блоке управления сумматором.

Блок преобразования адресов. В этом блоке производится преобразование математического адреса ячейки, относящейся к массиву команд, в физический адрес. Для сокращения числа обращений в память производится сравнение данного математического адреса страницы с математическими адресами страниц, к которым были предыдущие обращения. При их совпадении для преобразования адреса используется физический адрес страницы.

ци, запомненный на одном из двух специальных регистров. При отсутствии совпадения в блоке формируется запрос на считывание ячейки из таблицы преобразования адресов.

Блок связи процессора обеспечивает взаимодействие процессора через узел обменного регистра с другими узлами периферийной машины, в первую очередь, с узлом управления памятью, адаптером канала I-го уровня, каналлером и др.

#### Блок регистров

Оперативные регистры процессора состоят из 16 регистров байтов и буфера сумматора.

Каждая пара регистров байтов образует регистр двойного байта, который подключен к блоку обработки. При выдаче данных в блок обработки и приеме из него передается двойной байт. Выделение байта и запись байта производится в блоке обработки.

Регистры байтов с адресами 0-5 (двойных байтов -0,2,4) объединены в регистр слова Р0, регистры байтов 8-  
(8,А,С - для двойных байтов) - в регистр Р1. Эти регистры имеют непосредственную связь с обменным регистром. Регистры байтов 6,7 (двойной байт 6) и Е, F (двойной байт Е) называются соответственно Д3, Д7 (нумерация двойных байтов в аппаратуре 0-7). Эти регистры, так же как и буфер сумматора (ЕС), связаны только с блоком обработки.

Буфер команд (БК) служит для предварительной выборки команд. Этот регистр связан с адаптером канала I-го уровня (УД). С регистра БК командное слово передается на регистр команд (РК).

В процессе выполнения находится команда, помещенная в старших 24 разрядах РК. После выполнения левой команды на ее место помещается правая.

Поле кода операции (48-41 РК) и поле адресов (40-25РК) могут быть приняты в блок обработки для дешифрирования кода операции, формирования исполнительного адреса и т.п.

Регистр адресов слов (PAC) служит для хранения текущего значения математического адреса командного слова.

Для изменения этого адреса при выполнении последовательности команд и выполнении команд передачи управления код с регистра PAC передается в блок обработки, а новое значение адреса снова записывается на регистре. Кроме того, содержимое регистра PAC принимается в блок обработки в некоторых случаях при формировании исполнительного адреса.

При выполнении команд записи адреса возврата и перехода по адресу возврата код с регистра PAC передается на обменный регистр.

Регистр приема директивы (РПД) служит для хранения текущей директивы, которая вызвала запуск данной программы. Директива принимается из буфера директив.

Содержимое регистра приема директив принимается в блок обработки при выполнении директивы и при выполнении команд управления взаимодействием для данного подканала и команд опроса регистра приема директив.

Регистр выдачи директивы (РВД) служит для промежуточного хранения директивы, выдаваемой из процессора в буфер.

директив, каналлер или сопряжение канала 2-го уровня.

Запись на регистр РВД производится из блока обработки.

Регистр выдачи адреса процессора (РВАП) используется для промежуточного хранения адреса при обращении к памяти.

При обращении в массив данных на регистр РВАП поступает физический адрес, при этом в разряды I-16 поступает код с РВ, в разряды I7-I3 код с одного из регистров базовых адресов (Б0, Б1, Б2). При формировании исполнительного адреса данных разряды I6-9 не изменяются (передаются с регистров базовых адресов), разряды 8-I формируются путем сложения базы, смещения и модификатора.

При обращении в массив команд на регистре РВАП используются младшие 16 разрядов. Адрес, выдаваемый из блока обработки на РВАП, является математическим адресом. Разряды I6-9 передаются в схему преобразования адресов, при этом эта передача производится, если адрес страницы на регистре РВАП не совпадает с адресом страницы, использованной в предыдущих обращениях. Этот адрес хранится на одном из регистров математического листа (РМЛ). Математический адрес после обращения в таблицу страниц преобразуется в физический, при этом физический адрес страницы передается в адаптер канала I-го уровня с соответствующего регистра РФЛ, а адрес внутри страницы с регистра РВАП.

Регистры базовых адресов (Б0, Б1, Б2) служат для хранения базовых адресов, используемых в командах передачи

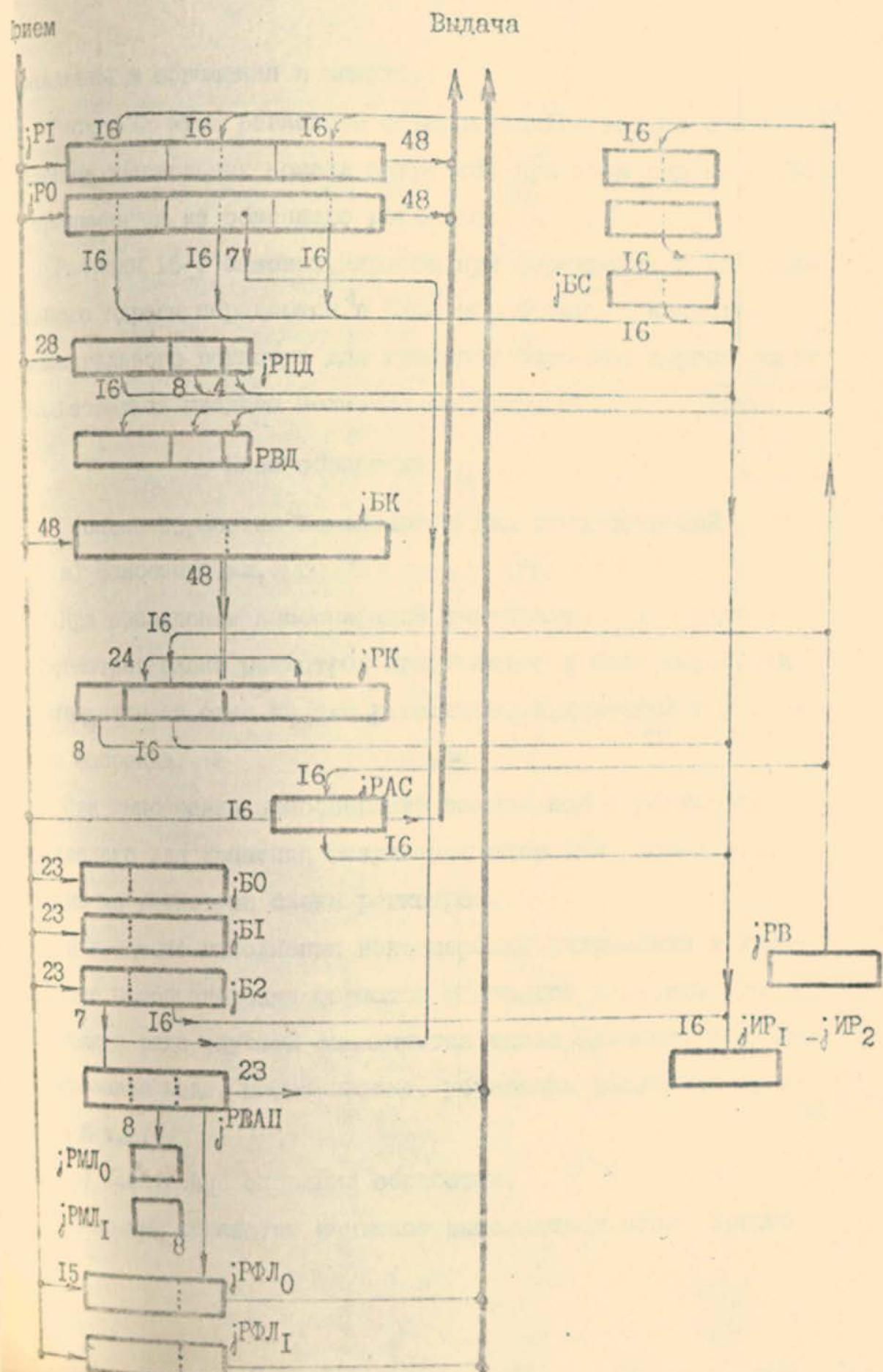


Рис. 43. Блок регистрационно-периферийного устройства

управления и обращения в память.

Загрузка этих регистров осуществляется только при выполнении специальных команд загрузки, при этом код на БО, Б1, Б2 принимается из обменного регистра.

Разряды I6-I базовых адресов при формировании исполнительного адреса передаются в блок обработки. В качестве дополнительного регистра для хранения базового адреса может использоваться младшая половина регистра Р0.

#### Блок обработки

В блоке обработки выполняются два типа операций

##### а) Нанооперации.

При выполнении наноопераций считывания код с одного из регистров блока регистров принимается в блок обработки и помещается на один из его регистров, выделенный для хранения операнда.

При выполнении наноопераций записи код с регистра, выделенного для хранения результата операции, помещается на один из регистров блока регистров.

В процессе выполнения наноопераций считывания и записи кроме преобразования форматов operandов и результата операции в ряде случаев выполняются такие функции, как дешифрование кода, шифрование, установка различных признаков и т.д.

##### б) Собственно операции обработки.

Операции обработки начинают выполняться после приема

на регистры блока обработки одного, двух или трех операндов.

Блок обработки (рис. 44) состоит из 6 16-разрядных регистров.

Регистр РВ является входным регистром блока обработки, на этом регистре производится преобразование форматов операндов. Выдача кода из блока обработки осуществляется через регистр РВ.

Регистр суммы РС в основном используется для образования поразрядной суммы операндов, находящихся на регистрах РВ, РР и РФ. Кроме того, он используется для пересылки операндов и результата операции, для сдвига кодов и т.д.

Регистр результата РР является вторым регистром суммы, на нем образуется поразрядная сумма операндов, находящихся на регистрах РС, РП. На РР помещается один из операндов, формируется результат операции и выполняется ряд других функций.

Регистр переносов РП работает при сложении одновременно с регистром РС. На РП формируются переносы при сложении кодов, находящихся на РВ, РР и РФ. Регистр РП также используется для передачи операндов, для сдвига кодов и др. функций.

Регистр фиксирования переносов РФ работает при сложении одновременно с регистром РР. На РФ формируются переносы при сложении кодов, находящихся на РС и РП. На РФ помещается второй операнд и выполняется ряд других функций.

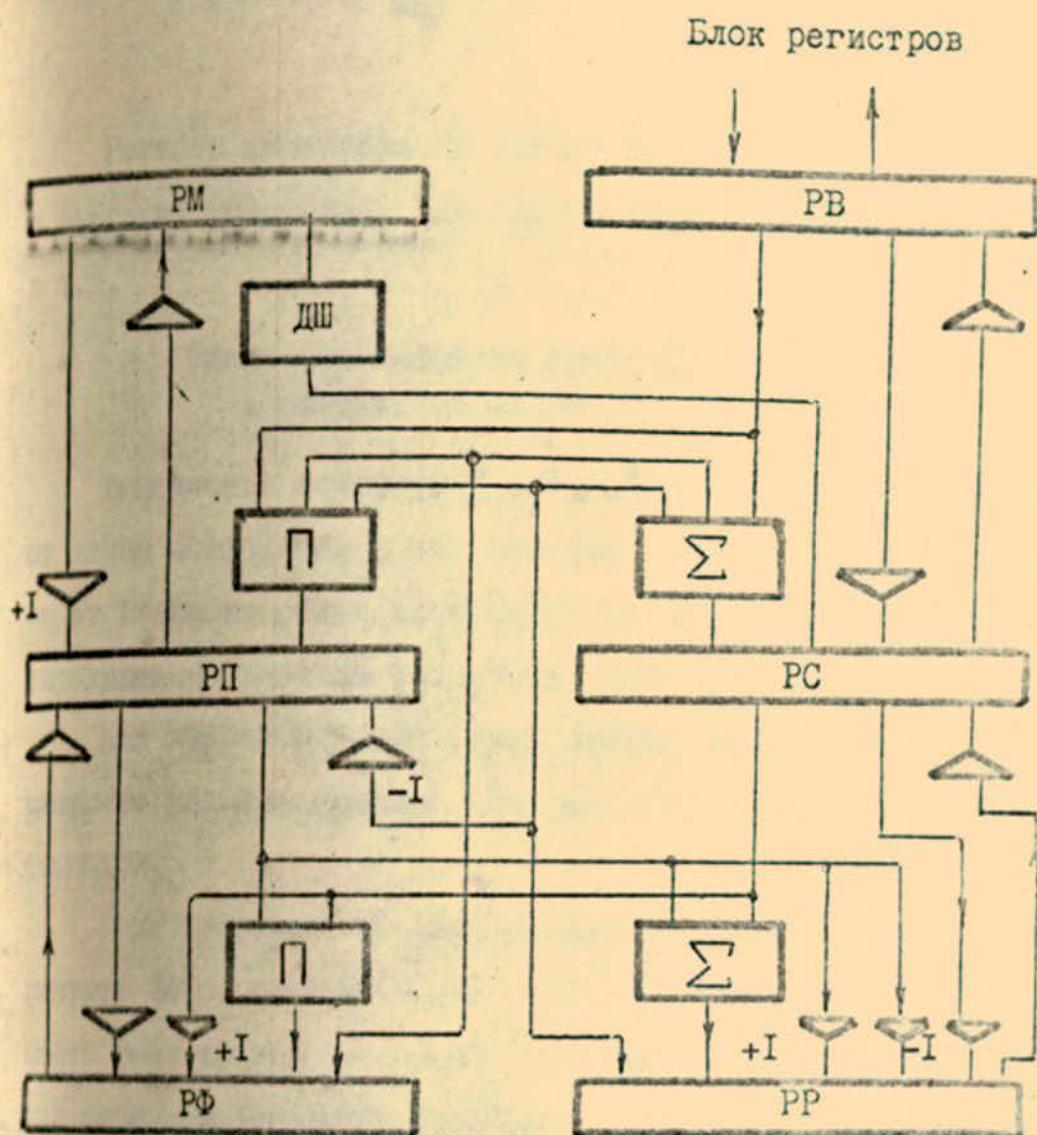


Рис. 44. Блок обработки периферийного процессора

Регистр множителя РМ служит для хранения множителя в операции умножения, для сдвига кодов, дешифрации кода и т.д.

Схема формирования operandов  
и результата операции

Выполнение собственно операций обработки не зависит от длины operandов. В тех случаях, когда operandы отличаются от двойного байта необходимо их выделение и передача в определенные разряды регистров блока обработки.

Для упрощения схем формирования сигнала переноса из старшего разряда принято следующее положение для различных форматов.

Байт занимает старшие разряды I6-9, адресное поле – разряды I6-5, причем адрес слова – разряды I6-9, адрес байта 8-5, тетрада – I6-I2 (Обработка тетрад используется для формирования исполнительного адреса байта при модификации адреса).

При считывании operandса с одного из регистров блока регистров необходимо выделить нужный формат и поместить в соответствующее место.

При записи результат операции, отличающейся от двойного байта, должен быть помещен в соответствующие разряды регистра двойного байта, причем остальные разряды не должны изменяться. Поскольку запись в блок регистров производится двойными байтами, процедура записи из блока обработки

в блок регистров состоит из считывания двойного байта из блока регистров, передачи результата в соответствующие разряды регистра блока обработки (при этом остальные разряды не изменяются) и записи двойного байта в блок регистров.

Для реализации описанных действий используется схема выделения и объединения на регистре РВ и схема сдвига на регистре РС.

Вентили приема на РВ из блока регистров и с регистра РС разделены на 5 групп, что обеспечивает выделение любого формата. В эти группы входят разряды I6-I3, I2-9, 8-6, 5, 4-1. Для управления указанными вентилями приема используются 5 усилителей <sup>поля</sup> выделений, в обозначения которых входят разряды, указывающие границы группы - I6B13, I2B9, 8B6, 5B5, 4B1.

Если все эти усилители установлены в 0, то код на РВ принимается с ИР1 и ИР2, если в 1 - то с РС. Если, например, установлены в 1 усилители I6B13 и I2B9, то в старшую половину РВ принимается код с РС, а в младшую - с ИР1 и ИР2 (прием кода из блока регистров возможен только на ИР1 или ИР2, поэтому при приеме на РВ подключаются оба регистра).

При считывании после приема кода на РВ производится передача на РС, при этом возможна передача без сдвига, либо сдвиг на 8 или 4 разряда.

При записи сдвиг осуществляется до формирования объединенного кода на РВ, однако, для выполнения сдвига используется та же схема, поэтому результат операции с РС передается без изменения на РВ, затем сдвигается и только после этого формируется код на РВ и записывается в блок регистров.

### Нанооперации считывания

Имеется три нанооперации считывания СЧ1, СЧ2, СЧ3. При выполнении нанооперации СЧ1, как правило, принимается первый операнд, помещаемый на регистр РФ, при выполнении нанооперации СЧ2 принимается второй операнд на РР, при выполнении СЧ3 в операциях обработки код третьего операнда принимается на РВ, в операциях обращения в память - на РР. Выполнение операций в начальной части происходит одинаково, а именно код из блока регистров попадает на РВ, где при этом выделяется требуемая часть кода, затем передается на РС (со сдвигом и без сдвига), затем на РР - в нанооперации СЧ2. В нанооперации СЧ1 кроме того производится передача с РР на РП, затем на РФ. В нанооперации СЧ3 код с РС передается на РР или на РВ.

### Нанооперации записи

Имеются две нанооперации записи ЗП1, ЗП2. При выполнении этих наноопераций, как правило, результат операции с регистра РР пересыпается на РС, затем на РВ, затем, если нужно, сдвигается и передается на РС. Одновременно с этим считывается код из блока регистров, затем этот код объединяется с кодом с регистра РС. Результат с регистра РВ переписывается в блок регистров.

### Операции обработки

Операция логического умножения. Для выполнения операции используются вентили переноса на регистре РП. Производится формирование поразрядной функции И. Вентили переноса образуют код, сдвинутый влево на 1 разряд, поэтому выполняется сдвиг вправо при передаче с РП на РР.

Операция сложения по модулю 2. Для выполнения операции используются вентили поразрядного суммирования на регистре РС. Формирование поразрядной суммы производится при приеме кодов с РР и РФ на РС, затем код пересыпается на регистр РР.

Операция логического сложения. Выполнение этой операции состоит в одновременном выполнении операций логического умножения и сложения по модулю 2. Таким образом, на регистре РП формируется сдвинутое поразрядное логическое произведение, на РС - поразрядная сумма по модулю 2. Затем на РР производится логическое сложение сдвинутого кода с РП и кода с РС.

Операция записи по маске. Для выполнения операции используются специальные вентили приема кода на РП при записи по маске. Перед началом операции на РР устанавливается маска, на РВ - операнд, разряды которого, соответствующие 1 в маске, передаются в качестве результата, на РФ - операнд, разряды которого, соответствующие 0 в маске, передаются в качестве результата 1.

Операции сдвига. При выполнении операции операнд находится на регистре РР. Для его сдвига используется прямая передача его на РП и передача со сдвигом вправо или влево на один разряд с РП на РР.

Код числа разрядов, на которое выполняется сдвиг, дешифрируется из двоичного в позиционный и помещается на регистр РМ. На регистре РВ организуется "бегущая" единица. При совпадении "бегущей" единицы с единицей в позиционном коде числа сдвигов фиксируется окончание операции. Сдвиг кода на один разряд сопровождается сдвигом "бегущей" единицы тоже на один разряд.

Операция сборки. При выполнении операции операнд постоянно находится на одном регистре (РФ), маска - на другом (РМ). На третьем регистре (РВ) организуется сдвиг кода, состоящего из одной единицы. Если в разряде маски, соответствующем положению "бегущей" единицы, код равен I, то выполняется сдвиг результата операции (РР). Кроме того, если код в соответствующем разряде операнда равен I, устанавливается в I старший разряд текущего значения результата. Таким образом, результат сдвигается столько раз, сколько единиц в маске, а код в результате зависит от кода операнда в тех разрядах, которые соответствуют разрядам маски, содержащим код I.

Операция разборки. При выполнении операции маска постоянно находится на одном регистре (РВ). Операнд поме-

щается на другой регистр (РФ). На третьем регистре (РМ) организуется сдвиг кода, состоящего из одной единицы. Результат операции (РР) при этом сдвигается на I разряд влево. Если в разряде маски, соответствующем положению "бегущей" единицы код равен 1, то выполняется передача старшего разряда операнда в младший разряд результата операции (РР), при этом операнд сдвигается на один разряд влево. Таким образом, из операнда (РФ) в результат операции (РР) передается столько старших разрядов, сколько единиц в маске, а положение этих разрядов в результате операции определяется положением единиц в маске.

Операция вычисления числа единиц. При выполнении операции операнд постоянно находится на регистре (РМ). На другом регистре (РВ) организуется сдвиг кода, состоящего из одной единицы. При сдвиге фиксируется каждый момент совпадения "бегущей" единицы с единицей в коде операнда. При этом производится сдвиг маркерного кода (одной единицы) на регистре (РР). Число сдвигов маркерной единицы равно числу единиц в коде операнда. После окончания операции позиционный код маркера шифруется в двоичный.

Операция вычисления номера старшей единицы. При выполнении операции операнд находится постоянно на регистре (РВ), на другом регистре (РМ) организуется сдвиг кода, состоящего из одной единицы (т.н. "бегущая" единица). При сдвиге единица переходит из старшего разряда в более младшие. При этом фиксируется первый момент совпадения "бегущей" единицы

с единицей в коде операнда. Затем позиционный код "бегущей" единицы шифруется в двоичный, в результате этого формируется результат операции.

Операция сложения. Операция сложения в отличие от других операций, выполняемых в сумматоре, запускается от специального сигнала, при этом не запускается цепочка блока обработки. Это вызвано тем, что сложение используется, как составная часть других операций, а именно — вычитания и умножения.

Выполнение операции сложения заключается в последовательном формировании поразрядной суммы и переносов. Для хранения двухрядного кода и выполнения операции используются пары регистров РС-РП и РР-РФ. Регистр РВ в качестве регистра для третьего слагаемого, однако, в операции сложения он не участвует (находится в нуле).

Операция вычитания. Операция вычитания выполняется как сложение уменьшаемого с дополнительным кодом вычитаемого. Уменьшаемое помещается на регистр РФ, вычитаемое — на РР.

Для формирования дополнительного кода вычитаемого выполняется инвертирование кода на РВ и прибавление 1 к младшему разряду, для чего устанавливается в 1 усилитель РВ.

Операция умножения. Алгоритм выполнения операции умножения состоит в следующем. Анализируется младший разряд множителя. Если он равен 1, то выполняется прибавление множимого к нулевому коду (начальная частичная сумма) и сдвиг множимого влево на один разряд. Если он равен 0, то выполняется только сдвиг. Затем сдвигается множитель на один разряд вправо, снова анализируется младший разряд (т.е. следующий разряд первоначального значения множителя) и либо прибавляется сдвинутый код множимого к частичной сумме и сдвигается множимое, либо только сдвигается множимое. Эта процедура повторяется 8 раз (по числу разрядов множителя).

Накопление частичной суммы производится в двухрядном коде, поэтому после каждого сложения приведение переносов не делается. Приведение переносов производится в конце операции.

Операция деления на полином. В операции участвуют два операнда — делимое и делитель. При выполнении деления, если старший разряд делимого или его текущего значения равен 1, делитель поразрядно вычитается (складывается по модулю 2) из текущего значения делимого, после чего новое значение делимого сдвигается влево на один разряд. Если старший разряд делимого или его текущего значения равен 0, то выполняется только сдвиг делимого.

Результатом операции является последний остаток после выполнения такого числа циклов операции, которое равно раз-

ности числа разрядов делимого и делителя.

В данном случае делимое содержит 32 разряда, делитель - 16 и остаток - 16 разрядов. Число циклов выполнения операции - 16.

Делимое в начале операции состоит из старого остатка и обрабатываемого отрезка последовательности. При выполнении операции они расположены соответственно на регистрах РР и РВ. С помощью регистров РИ и РС и вентиля сдвига из 16 РВ в ИРП организуется сдвиг 32-разрядного кода делимого.

При вычитании старшие 16 разрядов делимого с регистра РР пересыпаются на регистр РС, а делитель с регистра РФ, где он хранится без изменений, пересыпается на регистр РИ. Поразрядная сумма по модулю 2 кода на регистрах РС и РИ помещается на РР.

Для подсчета числа циклов операции используется сдвиг маркерной единицы на регистре РМ (с помощью регистра РИ).

Операция вычисления контрольной характеристики по коду Хэмминга. При выполнении этой операции производится последовательный анализ разрядов обрабатываемой последовательности. Если текущий анализируемый разряд равен 1, то к контрольной характеристике поразрядно по модулю 2 прибавляется номер этого разряда. Если этот разряд равен 0, то сложение по модулю 2 не выполняется. После анализа разряда производится вычитание 1 из номера разряда и переход к анализу следующего (справа) разряда последовательности.

Перед началом операции на регистре РВ размещается отрезок обрабатываемой последовательности, на регистре РФ - номер старшего разряда последовательности (разряды I6-9) и контрольная характеристика (разряды 8-I), на регистре РР - (разряды I6-I3) - номер разряда последовательности в пределах двойного байта, с которого начинается обработка.

Номер разряда дешифрируется и помещается на регистр РМ, при переходе к анализу следующего разряда маркер сдвигается на РМ вправо (с помощью регистра РМ).

При анализе проверяется разряд последовательности, положение которого соответствует положению маркера.

Для выполнения сложения по модулю 2 код номера разряда и контрольной характеристики, предварительно переписанной с РФ на РР, передаются на регистры РИ и РС, при этом код номера разряда помещается как в левый, так и в правый байт РИ, а в левый байт РС помещается нулевой код, в правый байт - контрольная характеристика. В результате поразрядного сложения по модулю 2 на РР в левый байт помещается номер разряда (складывается с нулевым кодом), в правый - новая контрольная характеристика.

Для вычитания I из номера разряда на РФ передается код I в разрядах 9-I6 и нулевой код в I-8 и производится сложение кода на РР и РФ, при этом контрольная характеристика, находящаяся в разрядах I-8 не изменяется.

В связи с тем, что на РВ при выполнении операции хранится отрезок обрабатываемой последовательности, использование обычного сложения, при котором на РВ должен быть

нулевой код, невозможно. В данном случае сложение выполняется только при передаче двухрядного кода с РП и РС на РФ и РР. При обратной передаче выполняется только перепись кода.

Операция сложения адресных полей. При сложении адресных полей производится сложение по модулю 6 адресов байтов и обычное двоичное сложение адресов слов.

Выполнение операции сводится к обычному сложению за счет формирования сигнала при наличии переноса из поля адресов байтов в поле адресов слов. При этом фактически осуществляется прибавление кода 010 к полю адреса байтов, благодаря чему операция сводится к двоичному сложению, так как прибавление кода 010 эквивалентно замене сложения по модулю 6 сложением по модулю 8, т.е. фактически сложению по модулю 2.

Операция вычитания адресных полей. При вычитании адресных полей производится вычитание по модулю 6 адресов байтов и обычное двоичное вычитание адресов слов.

Выполнение операции сводится к обычному вычитанию за счет формирования сигнала при наличии заема из поля адресов слов. При этом фактически осуществляется прибавление кода  $FF_{16}$  к полю адресов слов, что эквивалентно вычитанию единицы, и прибавление кода 6 к полю адресов байтов, что эквивалентно вычитанию двойки. При этом осуществляется заем из адреса слов и замена вычитания адресов байтов по модулю 6 вычитанием по модулю 8.

Операция формирования адресных полей. При выполнении операции операнд делится на 6. Это деление выполняется за два этапа. Сначала выполняется деление на 2 путем сдвига кода вправо на один разряд. При последующем делении на 3 анализируются по два разряда делимого. Вместе с этими разрядами делимого в анализе участвует старое значение остатка. При этом формируются два разряда частного и новый остаток (два разряда). После этого делимое и частное сдвигаются на два разряда влево, снова формируются два разряда частного и новый остаток. Эта процедура повторяется 8 раз (для деления двойного байта).

Первоначально операнд находится на РФ. После деления на 2 он попадает на РР. Регистр РР служит для хранения как делимого, так и частного. Первоначально все 16 разрядов заняты кодом делимого. Затем после использования двух старших разрядов делимого и формирования двух разрядов частного оставшиеся 14 разрядов делимого сдвигаются в старшие разряды, а 2 разряда частного помещаются в младшие разряды. По мере выполнения операции число разрядов делимого уменьшается, а число разрядов частного увеличивается.

Для хранения остатка используются разряды 14-16 регистра РВ, причем младший разряд остатка определяется при делении на 2 по коду младшего разряда делимого.

### Выполнение команд процессора

Выполнение каждой команды процессора распадается на последовательное выполнение нескольких микроопераций. Эта последовательность в основном зависит от того, к какой группе относится выполняемая операция (рис.45,46 ).

Выполнение директив также состоит из выполнения нескольких микроопераций.

Директива в буфер директив при начальном пуске может быть загружена с пульта или выдана по каналу I-го уровня одним из абонентов, в частности, центральным процессором.

После приема директивы в процессоре запускается микрооперация дешифрации кода типа директивы (ДШД).

Если необходим вызов базового слова, то после выполнения микрооперации ДШД запускается микрооперация дополнительного обращения в память (ДОП). При выполнении этой микрооперации происходит обращение в страницу I массива данных монитора. Адрес ячейки в этой странице определяется номером подканала, указанным в директиве. Считанное базовое слово принимается на регистры Б0 и Б1.

После микрооперации ДОП или после ДШД в случае, если не требуется вызова баз, запускается микрооперация исполнения директивы (ИД). В этой микрооперации формируется адрес начала запускаемой программы и производится вызов из памяти по этому адресу на буфер командного слова.

Вх. I - вход из ИД

## Вх. 2 - выполнение команды в автономном режиме

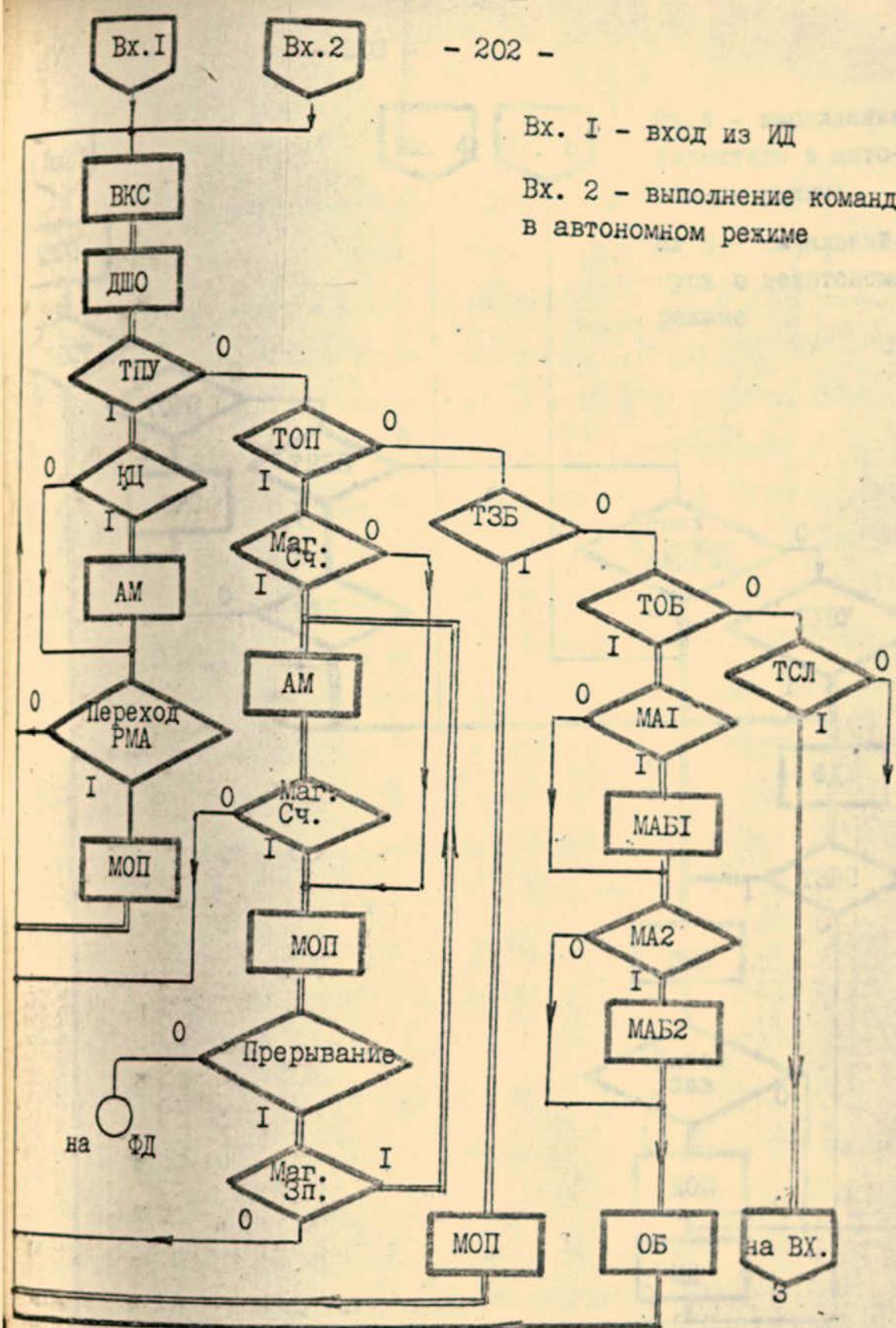


Рис. 45. Общая блок-схема алгоритма выполнения операций в периферийном процессоре

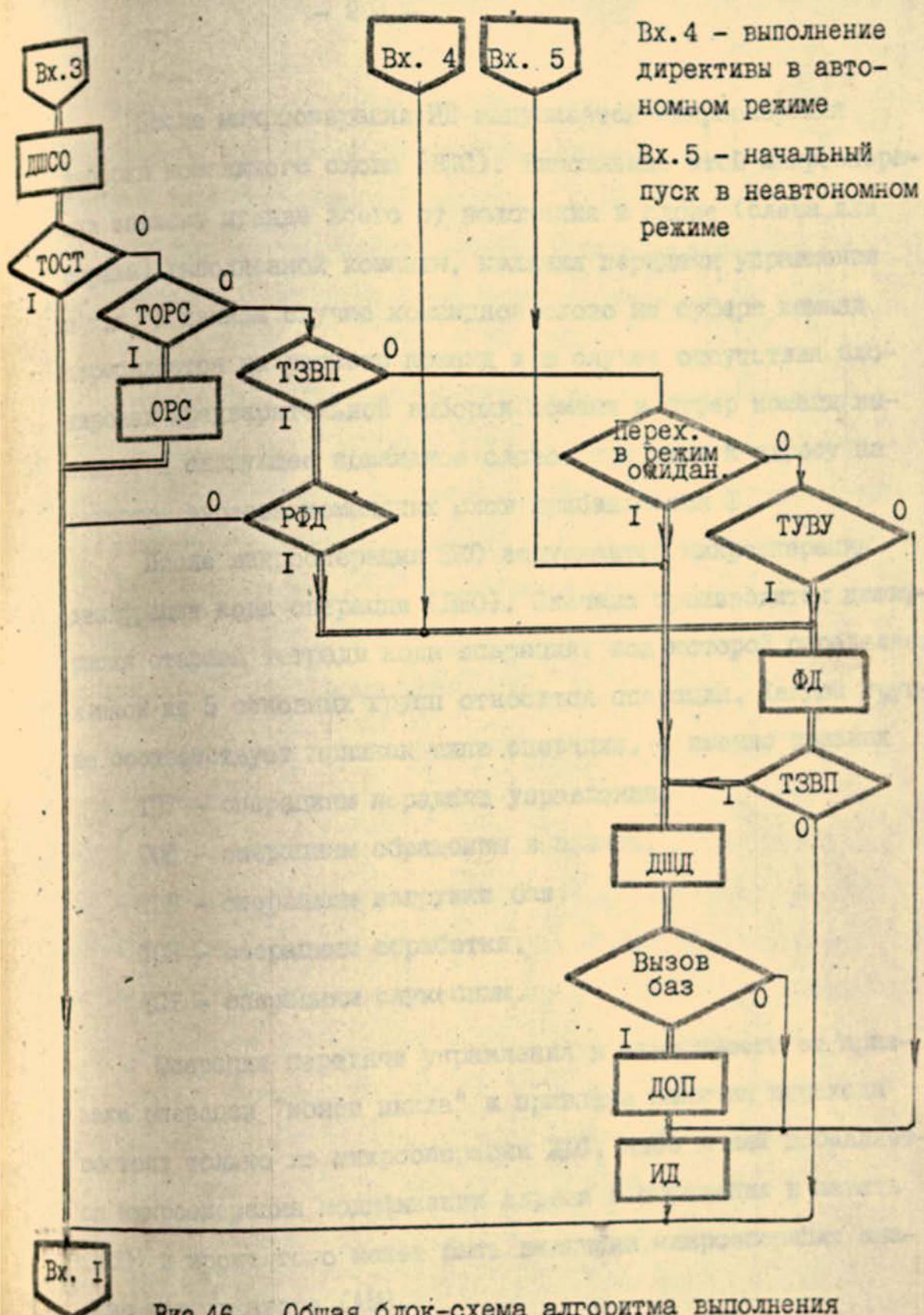


Рис.46. Общая блок-схема алгоритма выполнения операций в периферийном процессоре / окончание /.

После микрооперации ИД запускается микрооперация выборки командного слова (ВКС). Выполнение этой микрооперации зависит прежде всего от положения в слове (слева или справа) выполненной команды, наличия передачи управления и т.д. В данном случае командное слово из буфера команд пересыпается на регистр команд и в случае отсутствия блокировки предварительной выборки команд в буфер команд вызывается следующее командное слово, при этом к адресу на регистре адресов командных слов прибавляется 1.

После микрооперации ВКС запускается микрооперация дешифрации кода операции (ДШО). Сначала производится дешифрация старшей тетрады кода операции, код которой определяет, какой из 5 основных групп относится операция. Каждой группе соответствует признак типа операции, а именно признак

ТПУ - операциям передачи управления,

ТОП - операциям обращения в память,

ТЗБ - операциям загрузки баз,

ТОБ - операциям обработки,

TCI - операциям служебным.

Операции передачи управления в зависимости от признака операции "конец цикла" и признака наличия перехода состоят только из микрооперации ДШО, либо к ней добавляется микрооперация модификации адреса и обращения в память (МОП) и кроме того может быть включена микрооперация анализа модификатора (АМ).

Операции "конец цикла" запускают микрооперацию АМ, выполнение которой состоит в изменении модификатора и проверке его на наличие признака перехода.

В операциях условного перехода проверяется соответствие признаков результата предыдущей операции и условий перехода.

Если имеется признак перехода, то запускается микрооперация МОП. В этой микрооперации формируется новый математический адрес команды, который помещается на регистр адреса командного слова и выдается в блок преобразования адресов, где организуется считывание командного слова.

Если признака перехода нет, то после ДШО или после АМ запускается микрооперация ВКС.

Основой выполнения операций обращения в память также являются микрооперации АМ и МОП, выполняемые, однако, несколько иначе. Микрооперация АМ в операциях обращения в память используется только при выполнении магазинных операций, связанных с изменением модификатора. В операциях магазинного считывания микрооперация АМ выполняется перед микрооперацией МОП, в операциях магазинной записи - после этой микрооперации. В обоих случаях микрооперация АМ служит для изменения значения модификатора (содержимого индекс-регистра).

Остальные операции обращения в память используют только микрооперацию МОП.

Выполнение операции записи с возможностью прерывания (ТЗВП) при наличии директив в буфере сопровождается прекращением текущей программы (после МОП не запускается ВКС). При выполнении этих операций запускается микрооперация формирования директивы (ФД). В данном случае формируется директива возврата из прерывания, после выдачи директивы и записи ее в буфер организуется считывание из буфера очередной директивы и запуск микрооперации ДШД. Дальнейшая работа аналогична начальному пуску программы.

При выполнении операций загрузки базовых регистров используется только микрооперация МОП, после которой запускается ВКС.

Выполнение операции обработки сведено к выполнению микроопераций модификации адреса байта (МАБ1 и МАБ2) и микрооперации обработки (ОБ).

Микрооперации МАБ1 и МАБ2 запускаются в зависимости от наличия в команде признаков модификации первого или второго адреса. В этих микрооперациях выполняется модификация адреса.

Выполнение микрооперации ОБ заключается в приеме одного, двух или трех операндов в блок обработки, выполнении собственно операции обработки и записи результата операции на одном из регистров.

После микрооперации ОБ запускается ВКС.

Служебные операции подразделяются на несколько групп,

которым соответствуют признаки типа операции, а именно, признак ТОСТ соответствует операциям останова и управления разрешением останова, ТОРС - операциям обращения к регистрам, ТУВД - операциям управления взаимодействием для данного подканала, ТУВУ - операциям управления взаимодействием для указанного подканала. Расшифровка кода служебной операции и установка этих признаков производится в микрооперации дешифрации кода служебной операции (ДШСО).

При выполнении операций останова и управления разрешением останова никаких других микроопераций не запускается. При останове процессор прекращает свою работу до следующего пуска. В командах управления разрешением останова производится установка соответствующего усилителя (РОСТ) или его гашение (ГОСТ).

При выполнении операций обращения к служебным регистрам (регистры баз, регистр приема директивы и счетчики загрузки и разгрузки буферов директив) после ДШСО запускается микрооперация ОРС, после которой процессор переходит на микрооперацию ВКС. При выполнении микрооперации ОРС код с одного из служебных регистров принимается в блок обработки, откуда после преобразования формата переписывается на оперативный регистр.

При выполнении служебной операции прерывания мониторной программы последовательность микрооперации зависит от наличия директив в буфере. При отсутствии директив после ДШСО запускается ВКС, т.е. продолжается текущая программа.

При наличии директив запускается микрооперация ФД, при выполнении которой формируется и выдается директива возврата из прерывания. Затем процессор переходит на выполнение микрооперации ДШД.

Команда перехода в режим ожидания связана с прекращением текущей программы, выборкой очередной директивы и запуском ДШД. При отсутствии директив в буфере процессор останавливается до появления очередной директивы.

При выполнении остальных команд управления взаимодействием запускается микрооперация ФД, а затем осуществляется переход на выполнение следующей команды, т.е. на микрооперацию ВКС.

#### КАНАЛЛЕР

Как и в периферийной машине ПМ-6, каналлер предназначен для выполнения групповых операций (передачи) последовательности байтов между ОЗУ системы и абонентами в мультиканальном режиме. Управление обменом задается в виде последовательности управляющих слов, выбираемой из памяти аппаратурой каналлера, либо "подкачиваемой" процессором.

На рис. 34 показано взаимодействие каналлера с прочими узлами ПМ-10.

Пуск обмена в каналлере производится по соответствующей директиве процессора, поступающей на соответствующий регистр. Из блока приемного сопряжения с каналом 2-го уровня поступают данные при операциях приема и запросы при

операциях выдачи. В блок выходного сопряжения поступают данные, вызываемые каналлером из ОЗУ, а также информация из буферного регистра служебных сообщений. Наконец, информация об отработке управляющих слов и информация об ошибочных ситуациях (состояние) через буферный регистр направляется в буфер директив и далее в процессор. Исходя из требования полной программной совместимости, каналлер выполняет в точности те же команды обмена, что и каналлер ПМ-6. При разработке проведены такие структурно-логические изменения, которые в сочетании с возможностями системы элементов позволяют существенно повысить пропускную способность каналлера как на уровне байтов, так и на уровне сообщений.

На рис. 47 изображена блок-схема каналлера, на которой изображены его основные блоки:

Буфер управляющих слов;

Буфер данных;

Буфер промежуточного накопления посылок;

Блок управления выборкой команд и их выполнением.

Ниже будут описаны наиболее существенные особенности этих блоков.

#### I. Буфер накопления посылок.

Учитывая возможности повышения скорости обмена в канале 2-го уровня буфер накопления увеличен до 8÷16 регистров. Кроме того, он включен таким образом, что эффективно используется не только при приеме, но и при выдаче посылок.

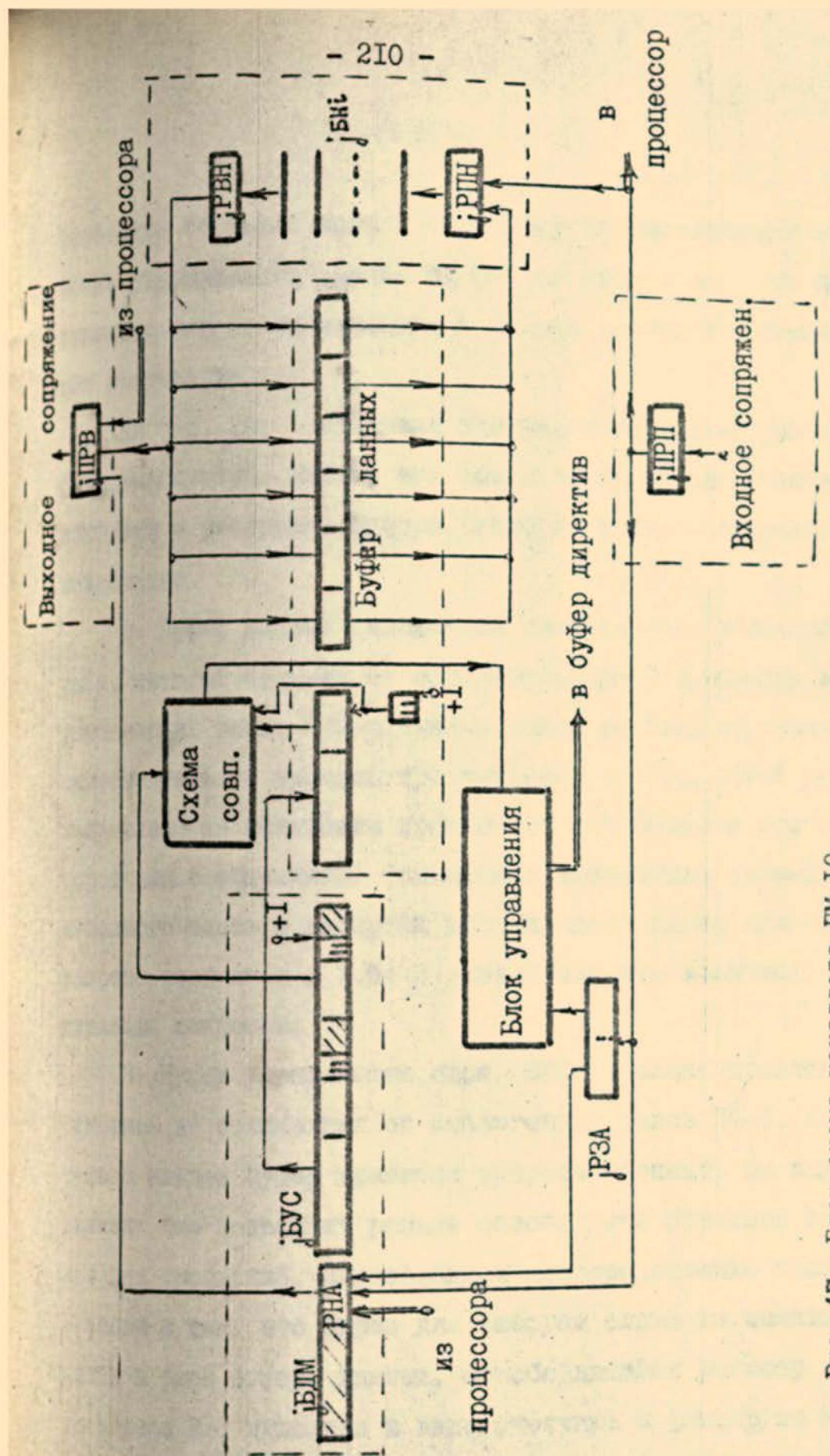


Рис. 47. Блок-схема канадлера ПМ-10

Последнее особенно важно, поскольку обеспечивается возможность упрежденного вызова из ОЗУ до двух-трех слов данных, исключая тем самым перерывы в выдаче сообщения из-за задержек считывания.

На рис. 48 показана временная диаграмма работы буфера накопления. Видно, что обеспечивается двухтактный цикл загрузки и разгрузки буфера, которая может выполняться одновременно.

2. Буфер данных снабжается собственным регистром адреса, который отделен от соответствующего регистра в буфере управляющих слов. Буфер данных таким образом превращен в самостоятельный уровень при запросах к ОЗУ. Такое решение направлено на повышение пропускной способности при обмене короткими сообщениями. Совмещение завершения операции приема неполного слова с выборкой управляющего слова для следующего запроса позволяет в 1,5+ 2 раза увеличить количество обслуживаемых запросов.

3. Буфер управляющих слов (БУС) и блок управления в основном не отличаются от аналогичных узлов ПМ-6. Дополнительно введен буфер хранения запроса абонента на выдачу данных: что позволяет раньше освобождать приемное сопряжение для сообщений, адресованных непосредственно процессору. В связи с тем, что адрес для выборки слова из памяти перенесен в блок буфера данных, освободившийся регистр текущего адреса БУС выполнен в виде счетчика и фиксирует адрес байта фактически прошедшего через сопряжение с каналом 2-го уровня. Такая схема обеспечивает правильность фиксации

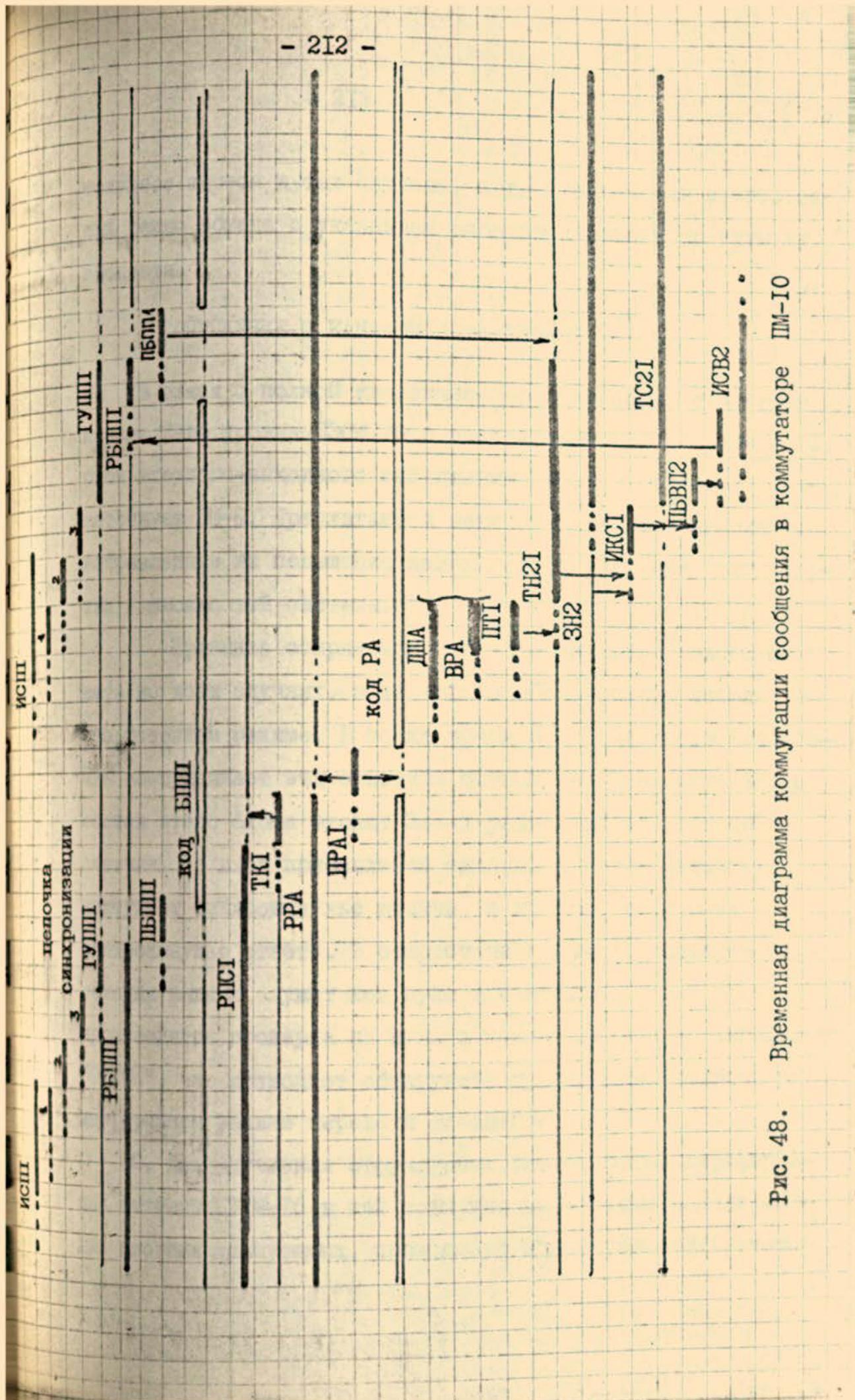


Рис. 48. Временная диаграмма коммутации сообщения в коммутаторе ПМ-10

текущего адреса в тех случаях, когда реализован стартстопный режим обмена с указанием окончания выдачи по сигналу абонента.

### СОПРЯЖЕНИЕ С КАНАЛОМ 2-ГО УРОВНЯ

В связи с полной идентичностью логической структуры канала 2-го уровня БЭСМ-10 с соответствующим каналом АС-6, блок входного-выходного сопряжения ПМ-10 в основном соответствует ПМ-6. Предлагаются некоторые добавочные схемы, направленные на повышение надежности и улучшение структурных возможностей системы.

1. Приемное сопряжение отвечает синхроимпульсом ответа во всех случаях, а не только после приема посылки в стартстопном режиме. В то же время блок выходного сопряжения воспринимает эти ответные сигналы во всех случаях. В состав этого блока входит 3+4-х разрядный реверсивный счетчик. На вход прибавления единицы к этому счетчику поступает синхроимпульс выдачи, а на вход вычитания — синхроимпульс ответа. В стартстопном режиме условием разрешения выдачи служит код нуля в счетчике. Кроме того производится проверка на нуль в момент получения сигнала "ОТБОЙ", что позволяет обнаружить пропадание посылок при непрерывном режиме передачи сообщения.

2. Для улучшения структурных возможностей периферийной системы БЭСМ-10 в ней можно использовать модификацию системы коммутации, приводящую к виртуальной (матема-

тической) адресации абонентов. При этом необходимо использовать коммутатор канала 2-го уровня, описанный ниже.

Такая коммутация требует передачи дополнительной посылки, содержащей собственный номер абонента. В блоке сопряжения с каналом 2-го уровня ПМ-10 установлены схемы, позволяющие принимать и выдавать такую посылку.

Если ПМ-10 выходит в канал с виртуальной адресацией абонентов, то в блоке сопряжения используется так называемый "встроенный коммутатор", обеспечивающий дублирование выхода в канал 2-го уровня. Управление коммутацией осуществляется с помощью таблицы, загружаемой процессором. Работа блока коммутатора происходит следующим образом.

Внешние выходы в канал пронумерованы как 1-ый и 2-ой; собственные блоки приема и выдачи считаются нулевым направлением для коммутатора. Появление коммутирующей посылки на любом входе вызывает выборку номера направления из таблицы, определяя физический номер направления, куда требуется направить сообщение. Если это направление свободно, то сообщение коммутируется обычным способом. При этом вслед за коммутирующей посылкой блок выходного сопряжения выдает посылку с собственным номером ПМ-10. В режиме приема ПМ-10 производит проверку коммутирующей посылки, сравнивая ее с регистром собственного номера. Подробное описание коммутации приведено в разделе, посвященном описанию коммутатора.

Следует отметить, что в вышеописанной схеме естественно реализуется замыкание выхода на вход для тестовой про-

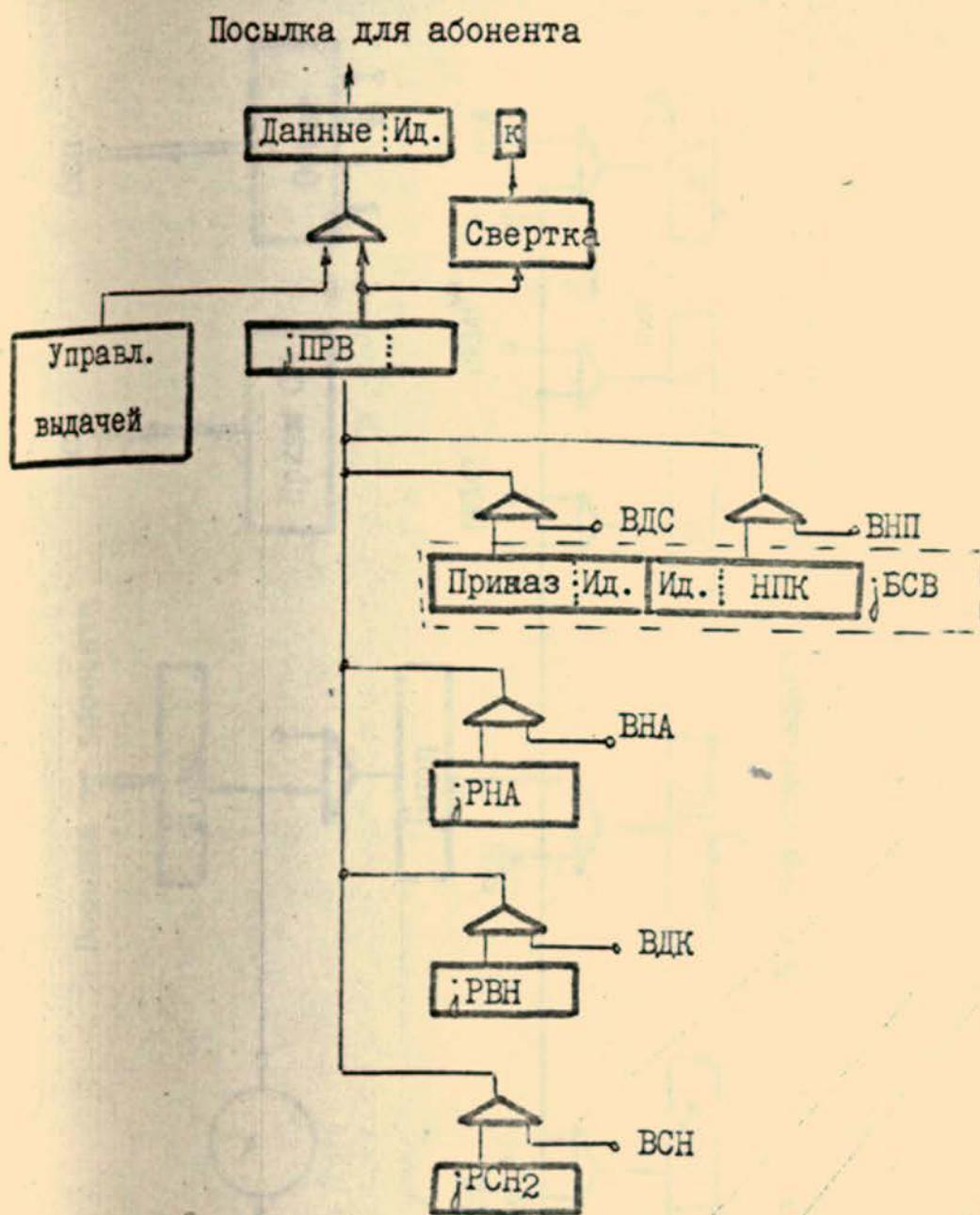


Рис.49. Цепи выдачи сообщения в канал 2-го уровня

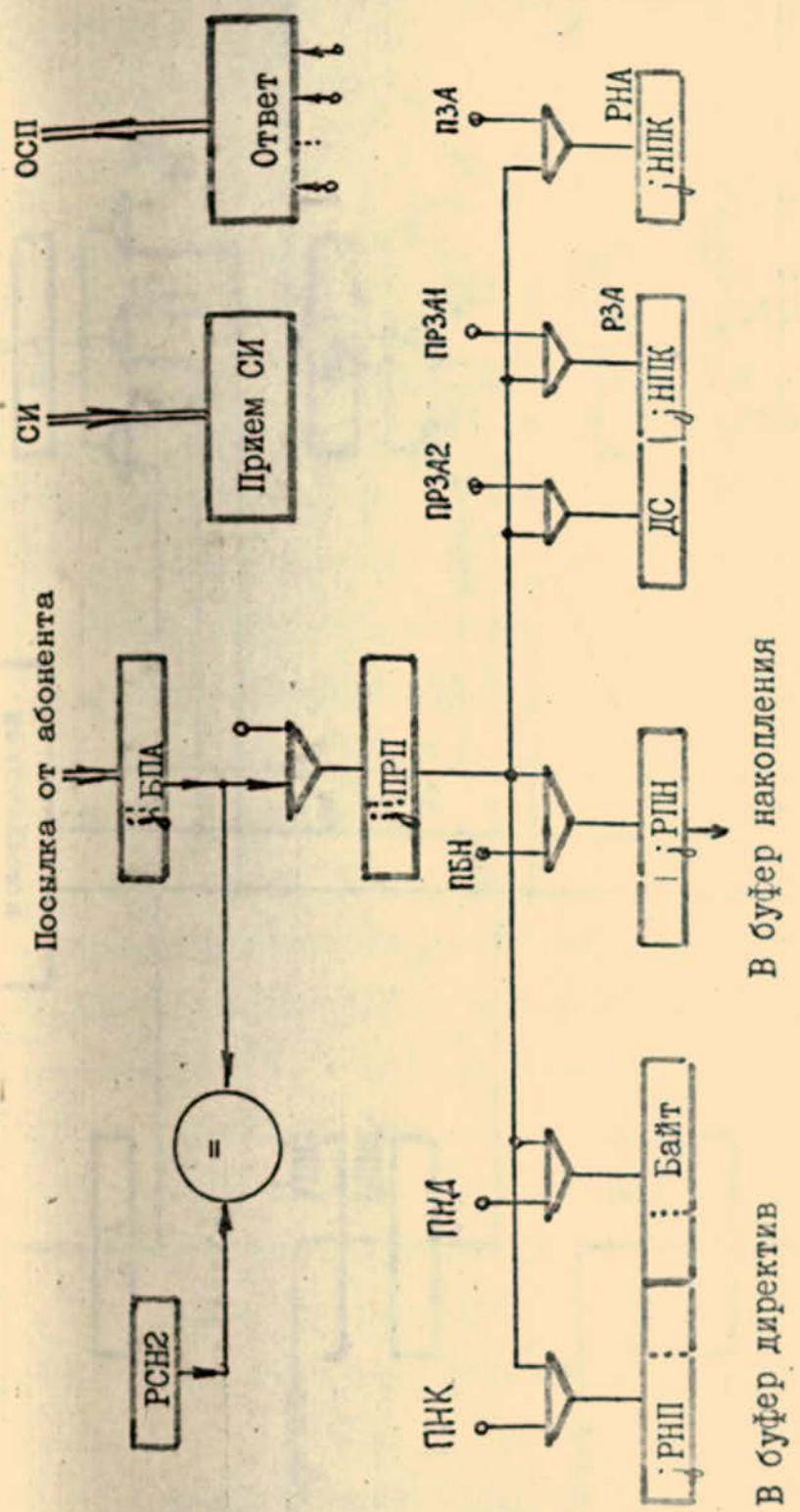


Рис. 50. Цепи приема сообщения от абонента

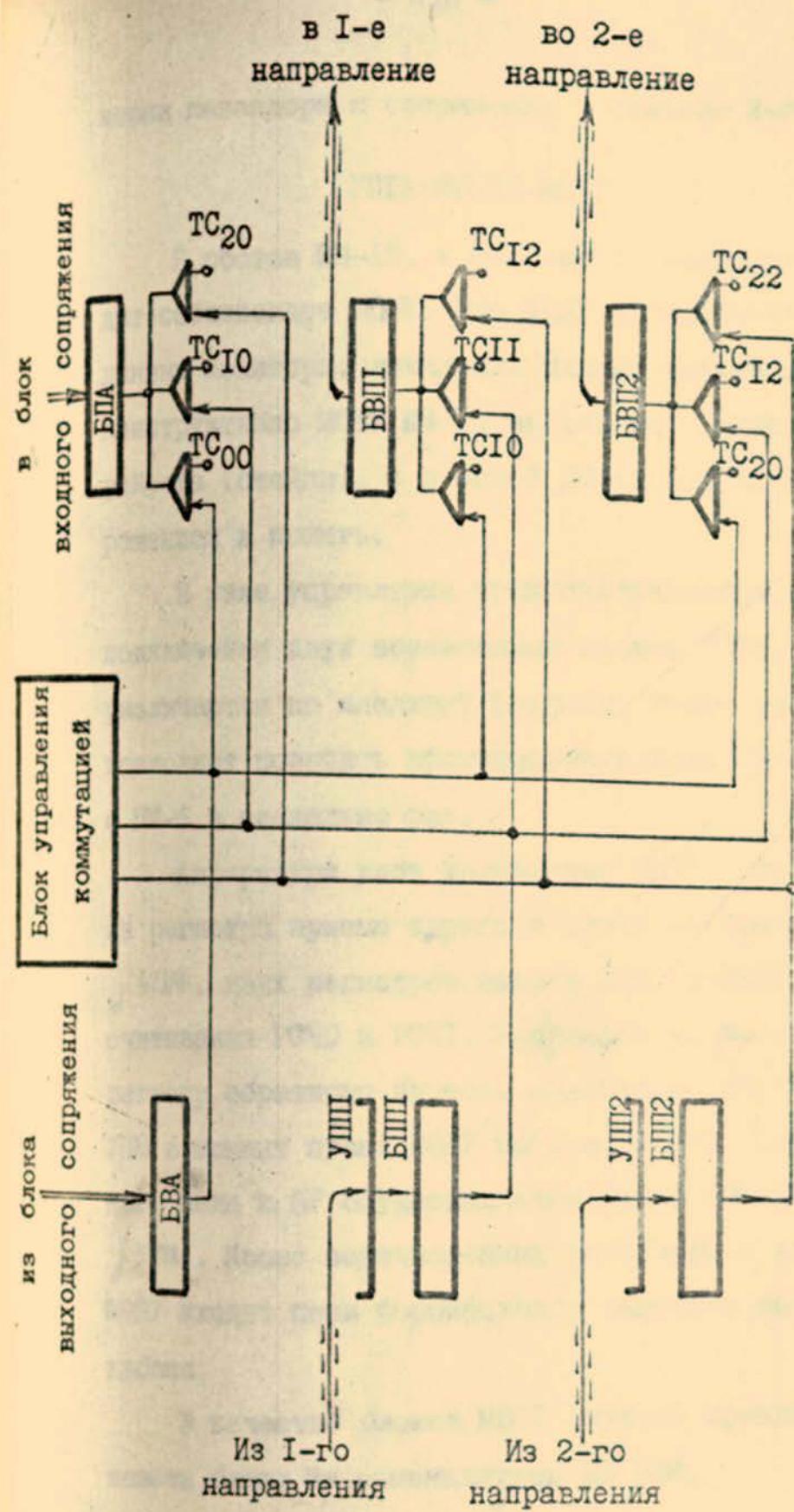


Рис. 51. Блок-схема входного коммутатора ПМ-10

верки каналлера и сопряжения с каналом 2-го уровня.

### УПРАВЛЕНИЕ МОЗУ

В состав ПМ-10, в отличие от аналогичной ПМ-6, входит собственное МОЗУ. Это МОЗУ предназначено для хранения данных монитора, локальных данных подканалов и программ. Конструктивно МОЗУ ПМ-10 выполнено в виде отдельного устройства (стойки), а в самой ПМ-10 размещено управление обращением в память.

В узле управления предусматривается возможность подключения двух независимых блоков МОЗУ, адреса которых различаются по младшему разряду. Такое расслоение памяти позволяет повысить производительность ПМ-10 по сравнению с ПМ-6 в несколько раз.

Аппаратура узла управления МОЗУ (рис. 52) состоит из регистра приема адреса и слова из обменного регистра УПМ, двух регистров записи РЗПО и РЗП1, двух регистров считывания РСЧ0 и РСЧ1. При каждом из последних имеется регистр обратного адреса, на который пересыпается код из УПМ в момент пуска МОЗУ по считыванию. Подключение считанного кода к ОР осуществляется через общий регистр выдачи УВМ. Кроме перечисленных регистров в узел управления МОЗУ входят цепи формирования сигналов взаимодействия с кубами.

В качестве блоков МОЗУ (кубов) предполагается использовать блоки из номенклатуры ЕС ЭВМ.

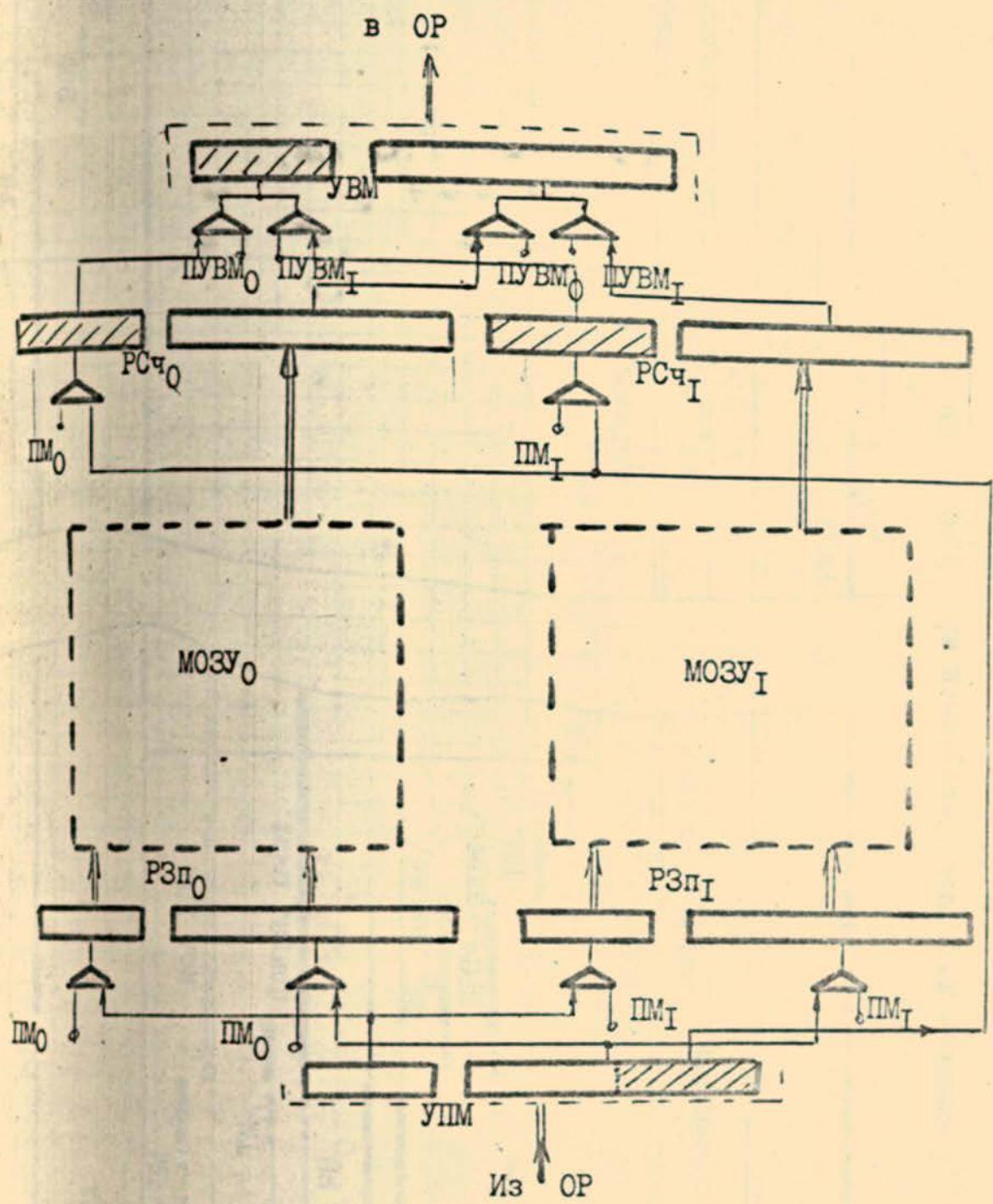


Рис. 52. Блок-схема управления МОЗУ ПМ-10

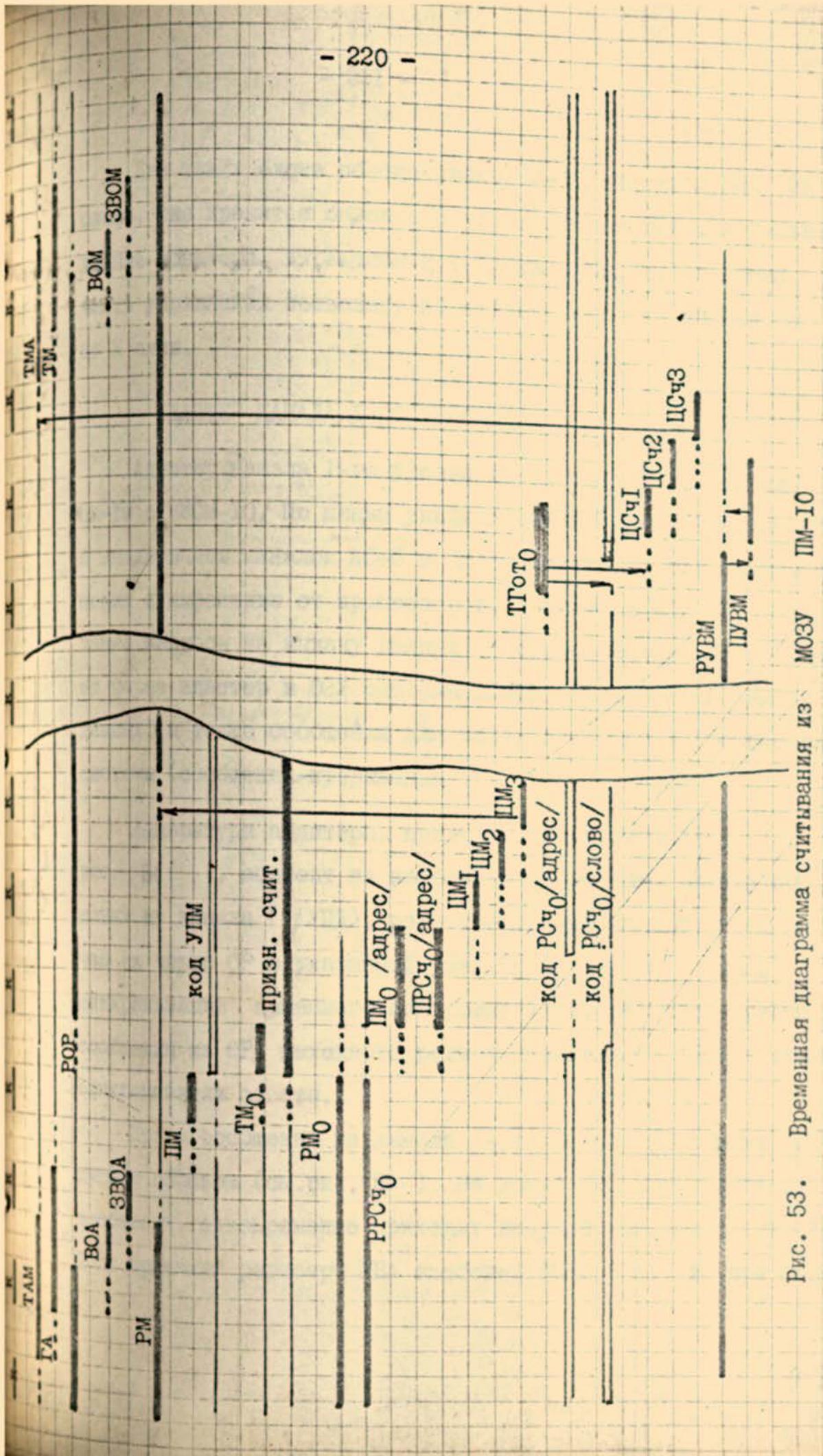


Рис. 53. Временная диаграмма считываия из МОЗУ ПМ-10

Два таких блока обеспечивают ПМ-10 массивом БДМ (2К слов), где хранятся базовые и управляющие слова, локальные данные монитора, директивы и таблицы адресов. Адресная часть управления позволяет наращивать емкость памяти до 64 К слов.

#### АДАПТЕР КАНАЛА I-ГО УРОВНЯ

Адаптер канала I-го уровня предназначен для связи ПМ-10 с БЭСМ-10. По шинам унифициированного и совместимого с аналогичным каналом АС-6 в ПМ-10 поступают служебные сообщения и директивы от процессоров, данные из ОЗУ БЭСМ-10, а также запросы на выдачу данных. В свою очередь, ПМ-10 выдает через адаптер в ОЗУ системы данные и запросы на выдачу данных, а также сообщения для взаимодействия с процессорами системы (сообщения-прерывания).

Аппаратура адаптера, приведенная на блок-схеме рис. 54 , состоит из кабельных усилителей приема сообщений из канала (  $j_{УПД}$  ), регистра приема данных перед выдачей их через ОР в указанный узел ПМ-10 (  $j_{УВА}$  ), схемы синхронизации приемного синхроимпульса, регистра приема сообщения из ОР, выходного регистра канала (  $j_{УВД}$  ) и схемы синхронизации ответа.

Прием сообщения из канала I-го уровня происходит следующим образом (см.рис. 55 ). На шинах УПД появляется код сообщения, одновременно приходит импульс синхронизации СИП. Если приемный регистр УВА свободен ( $СПР = 1$ ), то при фикса-

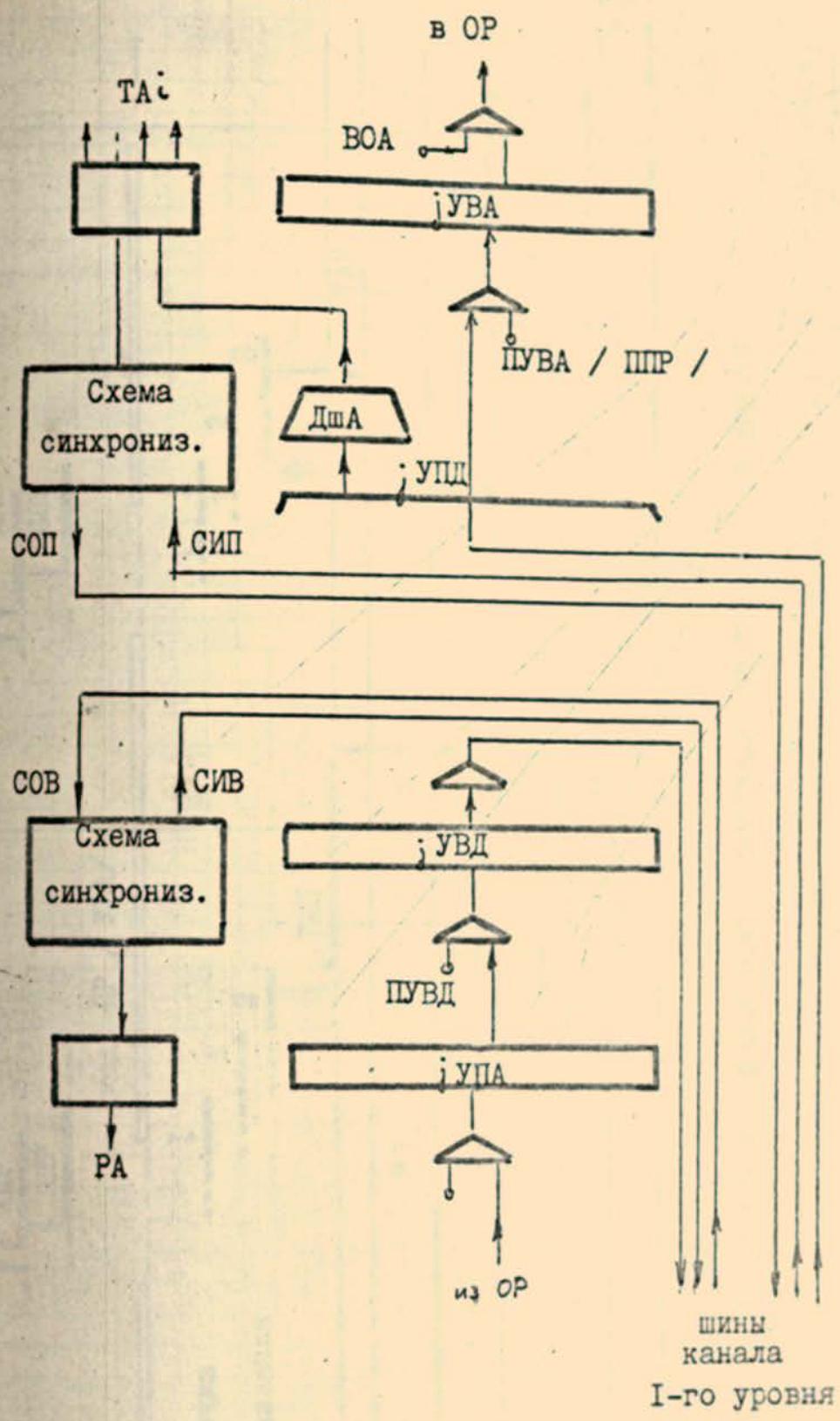


Рис. 54. Блок-схема адаптера канала I-го уровня ПМ-10

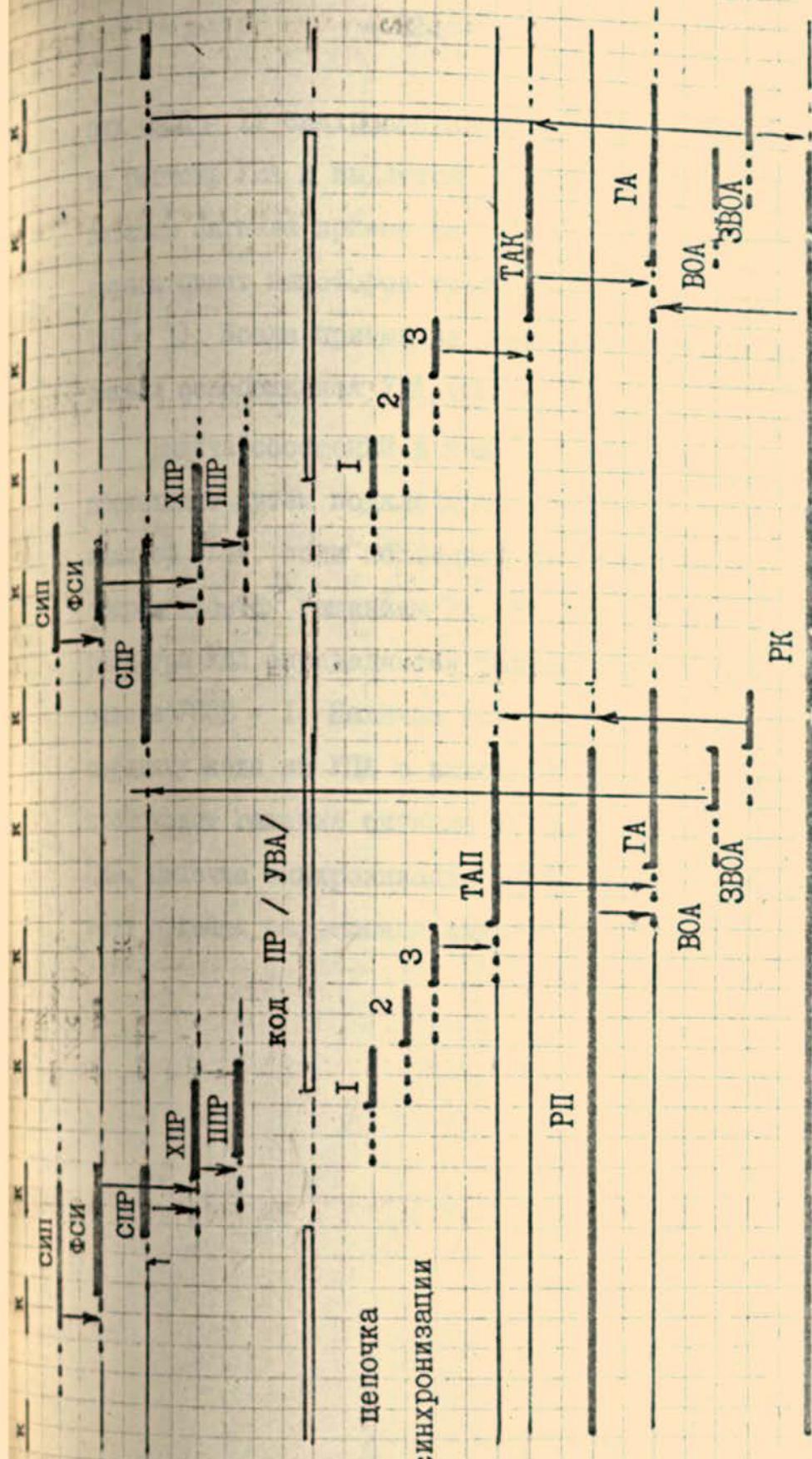


Рис. 55. Временная диаграмма приема сообщения из канала I-го уровня

ции ( $\Phi\text{СП} = I$ ) синхроимпульса производится прием сообщения на регистр УВА и выдается сигнал ответа в канал СОП. Синхронизатор сигнала приема на УВА вместе с дешифратором адреса вырабатывает некоторое требование (в приведенном примере  $\text{TAP} = I$ ). После приема на обменный регистр вырабатывается сигнал освобождения УВА ( $\text{СПР} = I$ ).

Выдача сообщений в канал I-го уровня (см.рис. 56) происходит путем подключения кода ОР к регистру приема в адаптер УПА, если последний не занят  $\text{PA} = I$ . Прием на УПА сопровождается гашением  $\text{PA}$ . Условие незанятости выходного регистра УВД определяется запомненным в адаптере сигнале ответа  $\text{УСОВ} = I$ . Наличие кода в УПА и это условие вызывает передачу кода из УПА в выходной регистр УВД; одновременно происходит гашение сигнала  $\text{УСОВ}$  и выдача синхроимпульса СИВ. Цепочка синхронизации этой передачи вновь устанавливает признак разрешения выдачи  $\text{PA} = I$ .

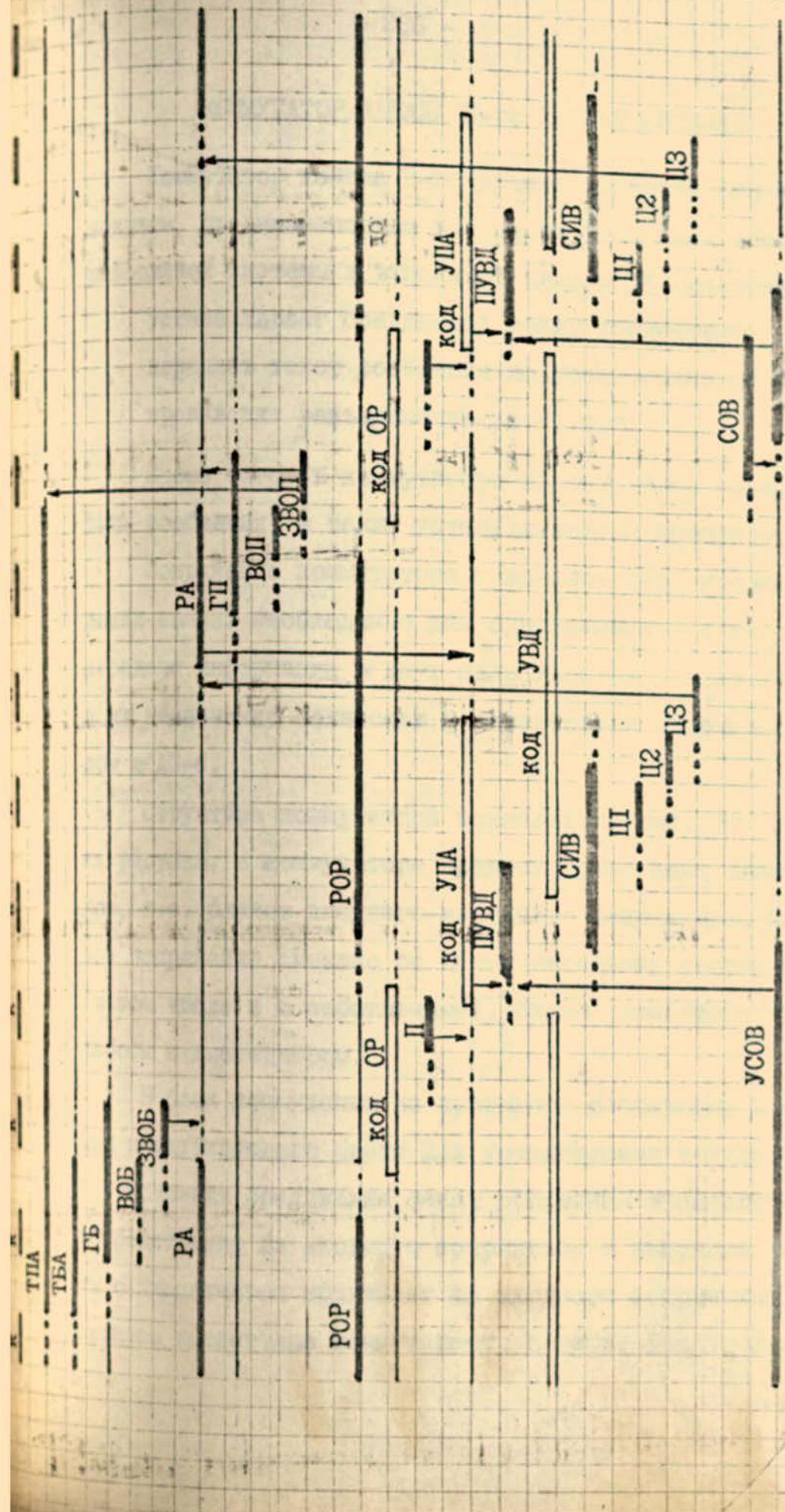


Рис. 56. Временная диаграмма выдачи в канал I-го уровня

### КОММУТАТОР КАНАЛА 2-ГО УРОВНЯ БЭСМ-10

Коммутатор канала 2-го уровня является функциональным модулем, предназначенным для построения сети абонентов периферийной системы и выполняет следующие основные функции:

устанавливает соединение между абонентами;

передает текст сообщения по установленному пути;

производит разъединение абонентов.

Кроме того, в коммутаторе производится контроль передачи сообщения, а также устанавливается приоритет абонентов.

Сопряжение коммутатора с каналом 2-го уровня содержит линии связи, необходимые для одновременной и независимой передачи информации в двух направлениях. Число этих линий и их назначение полностью соответствует сопряжению с каналом в АС-6.

Структура коммутатора показана на рис. 57. Как видно из рисунка, в коммутаторе предусмотрено семь внешних выходов, т.е. блоков входного-выходного сопряжения. Каждое входное сопряжение связано со всеми выходными сопряжениями (в том числе и с собственным), блоком коммутации и блоком работы с диспетчером.

В блок коммутации из приемного сопряжения поступает код коммутирующего байта для установления соединения. После установления соединения текст сообщения транслируется непосредственно из входного сопряжения в выходное. Блок работы с диспетчером принимает из входного сопряжения текст приказа диспетчера коммутатору, а также выдает ответные

Блоки приемного сопряжения

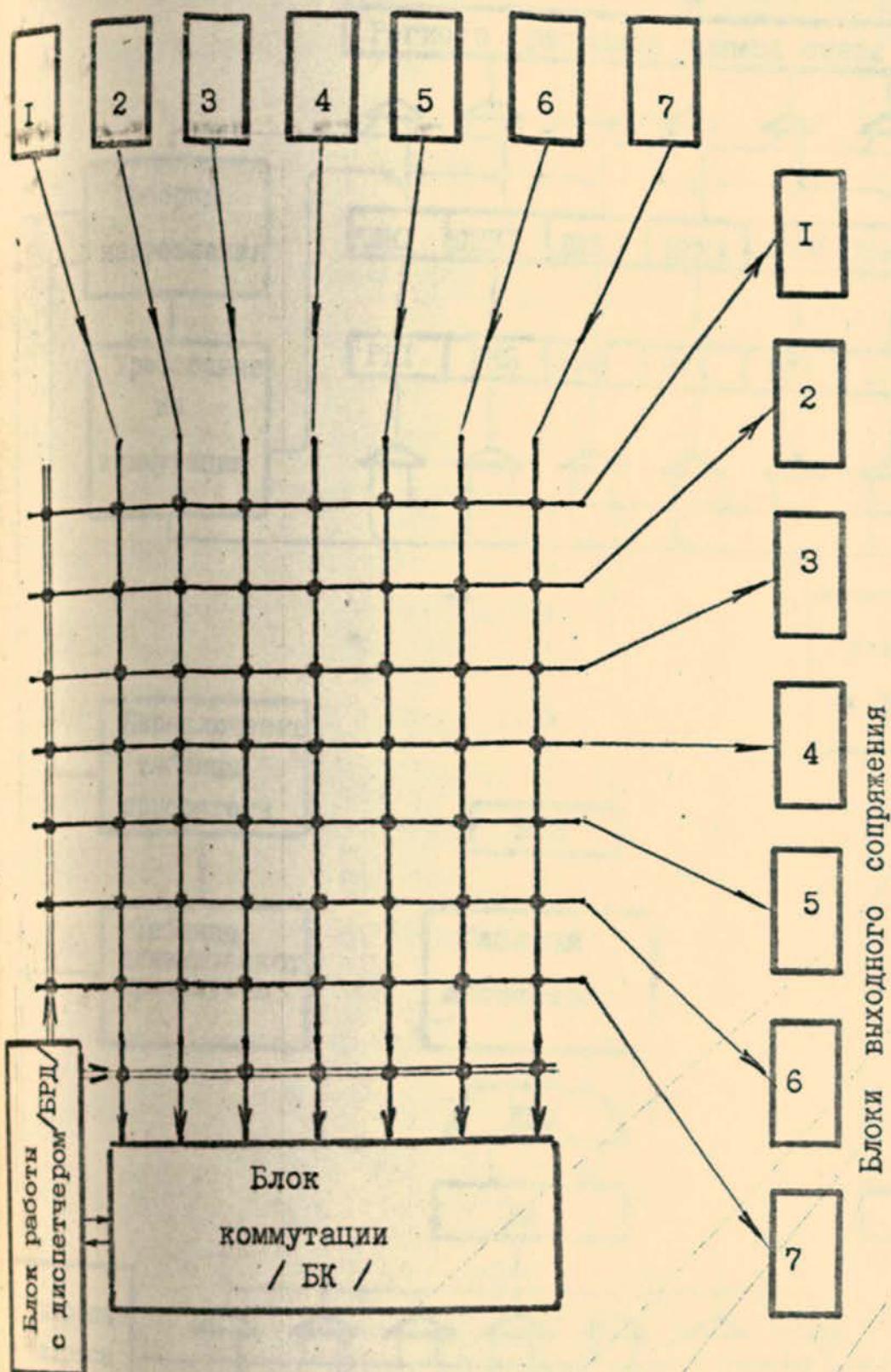


Рис. 57. Блок-схема коммутатора канала 2-го уровня

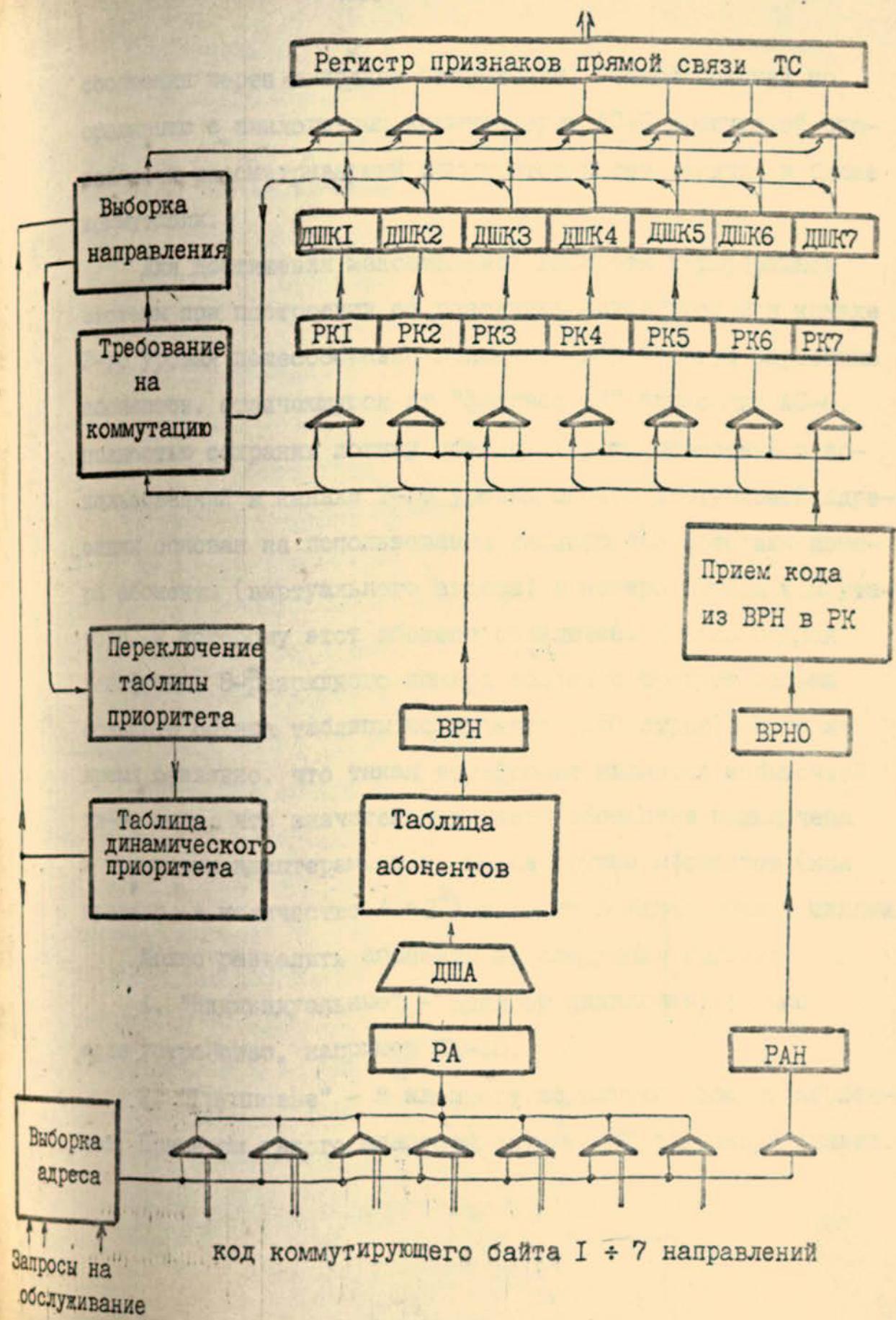


Рис.58. Блок-схема узла управления коммутацией /БК/

сообщения через выходное сопряжение. Помимо большей по сравнению с аналогичным коммутатором АС-6 пропускной способности рассматриваемый коммутатор имеет отличия в блоке коммутации.

Для достижения максимальной гибкости периферийной системы при построении ее различных конфигураций в канале 2-го уровня целесообразно применить виртуальную адресацию абонентов, отличающуюся от "Физической" адресации АС-6, полностью сохранив логику обмена. Хорошо известный и использованный в канале I-го уровня способ виртуальной адресации основан на использовании таблицы соответствия номера абонента (виртуального адреса) и номера выхода коммутатора, к которому этот абонент подключен. Однако полная дешифрация 8-разрядного номера абонента требует весьма большого объема таблицы коммутации (256 строк). В то же время очевидно, что такая дешифрация является избыточной из-за того, что значительная часть абонентов подключена к групповым адаптерам, т.е. целая группа абонентов (как правило, в количестве  $N \approx 2^K$ ) выходит в канал общими шинами.

Можно разделить абоненты на следующие группы:

- I. "Индивидуальные" - адаптер подключает только одно устройство, например ПМ-10.
2. "Групповые" - к адаптеру подключено восемь абонентов. Примером такого адаптера служит МПС выносного пульта.

3. "Массовые" - адаптер подключает 32 устройства.

Таковы адаптеры МТг, МТф или МПС, управляющий магнитными лентами.

Анализируя ряд конкретных конфигураций системы БЭСМ-10, можно допустить, что в периферийную систему, непосредственно взаимодействуя друг с другом, входят 8 индивидуальных, 7 групповых и 6 массовых абонентов (возможно также и разбиение 8-15-4). При таком разбиении в таблице коммутации требуется всего 21 строка (или 27 строк), т.е. объем вполне приемлемый.

Особый вид индивидуальных абонентов составляют сами коммутаторы. Поскольку обращение к ним производится с помощью специальных служебных сообщений, они не входят в число основных 256 абонентов. Направление для передачи сообщений коммутаторам задается дополнительной таблицей, рассчитанной на  $8 \div 16$  номеров.

Коммутация канала в коммутаторе происходит следующим образом. При поступлении коммутирующей посылки на буфер приемного сопряжения на  $i$ -м входе  $j_{БДП}^i$  производится проверка посылки на четность и устанавливается сигнал СИ $i$ . Этот сигнал направляется в блок коммутации в качестве запроса на обслуживание. Из всех таких запросов выбирается один, имеющий в данный момент наивысший приоритет. Этот приоритет фиксируется таблицей динамического приоритета, состоящей из  $\frac{п.(п-1)}{2}$  парафазных усилителей (где  $п = 7$  - количество выходов в коммутаторе). Усилитель таблицы

$TPI_j^i = I$  фиксирует, что  $i^{\text{е}}$  направление имеет приоритет перед  $j^{\text{е}}$ . Прием запроса выполняется в два этапа. На первом определяется отсутствие обслуживания из-за низкого приоритета:

$$OBA_i^j = СИ2 \cdot TPI2 + СИ3 \cdot TPI3 + \dots + СИ7 \cdot TPI7,$$

На втором этапе при условии незанятости входного регистра блока коммутации  $PPA=I$  выполняется прием кода коммутирующей посылки на этот регистр по сигналу:

$$VA_i^j = СИi \cdot \overline{OBA}_i^j \cdot PPA \cdot RTAB,$$

Сигнал  $RTAB = I$  означает, что в данный момент состояние в таблице фиксации приоритета не изменяется.

Код входного регистра:

$$j_{PA} = [j_{PA}] \cdot \overline{OPA} + \sum_{i=1}^7 j_{BDPI} \cdot VA_i^j,$$

служит адресом для выборки строки из таблицы коммутации.

В соответствии с принятой классификацией абонентов таблица состоит из четырех групп регистров (строк):

8 регистров индивидуальных абонентов;

7 регистров групповых абонентов;

6 регистров массовых абонентов,

а также 8 (16) регистров для обращения к коммутаторам. На рис. 59 приведено распределение видов абонентов в соответствии с кодом виртуального адреса:

Построенный на регистре  $j$  РА дешифратор ДША выбирает заданную строку на выходной регистр  $j$  ВРН. Стока содержит четыре разряда, причем три используются для задания направления коммутации, а четвертый является контрольным. Нулевой код строки означает, что требуемый адрес в системе отсутствует или заблокирован.

Если код ВРН правильный и нет блокировки, то он пересыпается в регистр коммутации обслуживаемого направления, при этом вырабатывается сигнал требования коммутации  $T_i$ , а запрос СИ $i$  гасится. После произведенного таким образом определения номера требуемого выхода коммутатора происходит в точности такая же работа, как и в коммутаторе АС-6. Задержка, вызванная обращением к таблице, составляет всего 4 такта и практически не влияет на время передачи сообщения. Временная диаграмма коммутации приведена на рис. 60

Сообщения, предназначенные для самого коммутатора, направляются в блок работы с диспетчером и попадают в регистры:

номера коммутатора РНК;

номера абонента-отправителя РНО;

Тип адаптера		Адрес
Индивидуальный	№ 0	<u>00000000</u>
Индивидуальный	№ I	<u>0000000I</u>
.....	...	<u>00000xxx</u>
Индивидуальный	№ 7	<u>00000III</u>
Групповой	№ I	<u>0000IxXX</u>
Групповой	№ 2	<u>00010xxx</u>
.....	...	.....
Групповой	№ 7	00IIIxxx
Массовый	№ I	0I0xxxxx
Массовый	№ 2	0IIxxxxx
.....	...	.....
Массовый	№ 6	IIIxxxxx

Рис. 59. Таблица соответствия виртуальных  
адресов типам абонентов

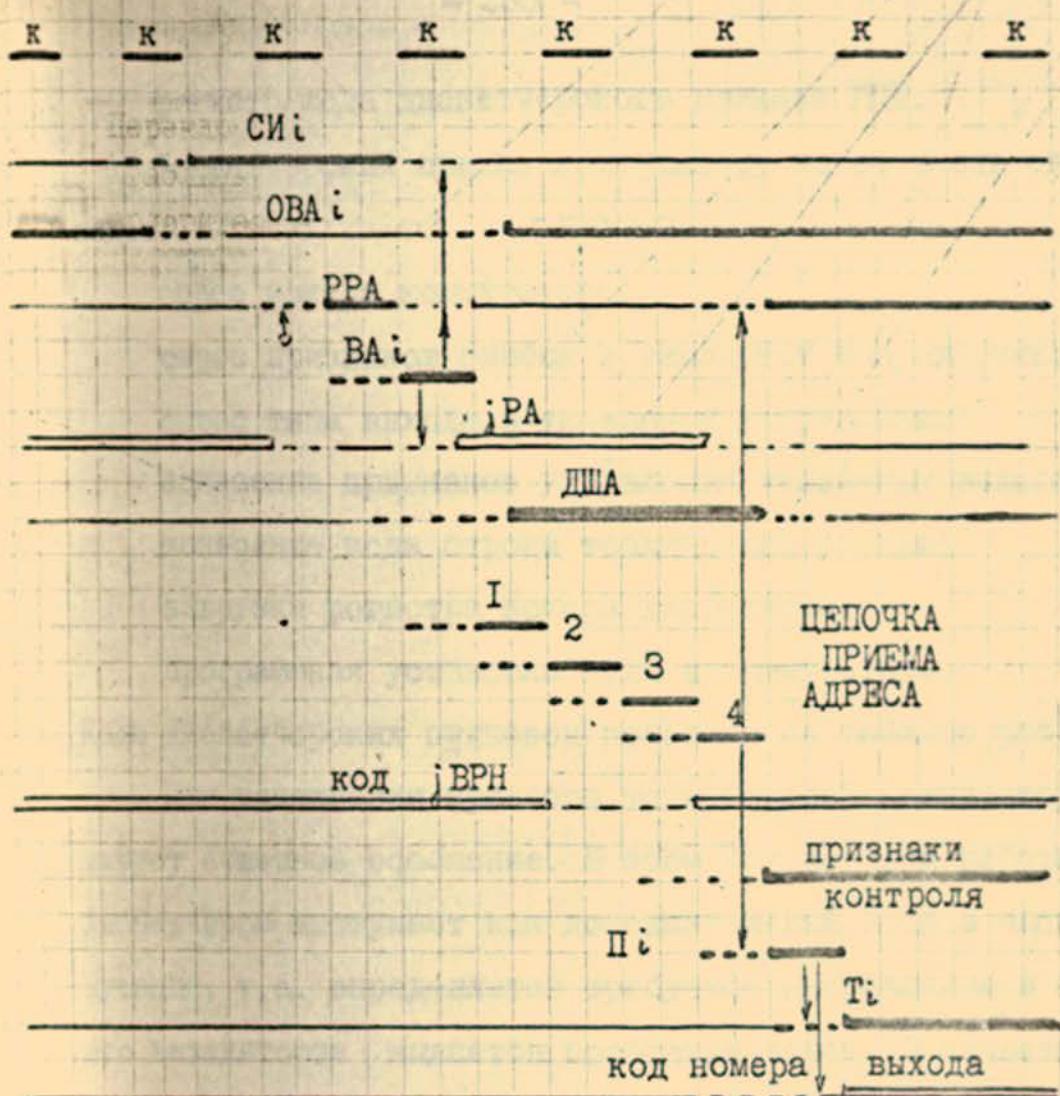


Рис. 60. Временная диаграмма работы блока коммутации

регистр. кода диспетчерского приказа РЦД.

Диспетчерский приказ коммутатору может иметь один из видов:

опрос номера коммутатора;

опрос признаков ошибок в указанном направлении;

опрос типа выхода в указанном направлении;

занесение признаков управления указанным выходом;

занесение кода строки таблицы коммутации;

загрузка регистра номера диспетчера;

программная установка нуля в коммутаторе.

Коды диспетчерских приказов показаны на таблице рис.61-63.

При исполнении приказов типа "опрос" коммутатор формирует ответное сообщение. В этом случае блок работы с диспетчером выступает как дополнительный вход в блок коммутации, т.е. определяется требуемое направление и при его занятости выдается сообщение-ответ. В качестве коммутирующего байта в канал выдается код номера абонента- отправителя РНО вместе со специальным идентификатором ответа. На рис.64,65 приводится таблица кодов ответа коммутатора. После выдачи ответа в коммутаторе снимается признак занятости и разрешается новое обращение к коммутатору.

0	I	2	3	4	5	6	7
Тип регистра	Номер регистра				Код, загружаемый в регистр		

Рис. 61. Распределение разрядов в байте диспетчерского приказа коммутатору. Приказ "Загрузка"

Таблица

Байт приказа			Назначение приказа
1 0 0 0 0 0 0 0			Установка нуля
1 0 0 0 0 0 0 1			Загрузка регистра номера диспетчера
0 0 I			
0 I 0			
0 I I			Загрузка
0 0			управляющих регистров
I 0 0			
I 0 I			I + 7 направлений
I I 0			
I I I			

Рис. 62. Таблица приказов коммутатору канала 2-го уровня

Таблица /продолжение/

Тип	Номер	Назначение приказа	Адрес абонента
01	000		00000000
	001	Загрузка строки таблицы коммутации для абонентов с индивидуальными адаптерами.	00000001
	010		00000010
	011		00000011
	100		00000100
	101		00000101
	110		00000110
	III		00000111
10	001		000001XX
	010	Загрузка строки таблицы коммутации для абонентов с "групповыми" / по 8 / адаптерами.	000101XX
	011		000111XX
	100		001001XX
	101		001011XX
	110		001101XX
	III		001111XX
	11	010	
011		Загрузка строки таблицы коммутации для абонентов с "массовыми"/32/ адаптерами.	011XXXXX
100			100XXXXX
101			101XXXXX
110			110XXXXX
III			111XXXXX

Рис.63. Таблица приказов коммутатору канала 2-го уровня

0	I	2	3	4	5	6	7
Код "00000" -				Номер			
признак типа				опрашиваемого регистра			
приказа							

Рис.64. Распределение разрядов в байте диспетчерского приказа коммутатору. Приказ "Опрос".

Номер регистра	Назначение приказа
0 0 0	Опрос номера коммутатора
0 0 I	
0 I 0	Опрос признаков состояния -
0 I I	тип выхода, флаги ошибок - для
I 0 0	I + 7 направлений
I 0 I	
I I 0	
I I I	

Рис.65. Таблица приказов коммутатору канала 2-го уровня

### АДАПТЕР ПРЕОБРАЗОВАНИЯ СОПРЯЖЕНИЯ

В состав периферийной системы БЭСМ-10 входит адаптер для подключения внешних устройств ЕС ЭВМ. Этот адаптер (АПС) производит главным образом преобразование сопряжения канала 2-го уровня БЭСМ-10-ЛС-6 к сопряжению стандартного канала ввода-вывода ЕС ЭВМ.

Периферийная машина ПМ-10 совместно с модулем АПС обеспечивает все режимы, необходимые для работы с устройствами ЕС ЭВМ: мультиплексный, селекторный и блочно-мультиплексный.

В целом АПС действует так же, как и аналогичный ему модуль МПС системы ЛС-6. Вместе с тем в нем предусмотрены меры для повышения производительности при обслуживании внешних устройств. Для этого управление магистралью интерфейса расширено таким образом, что обеспечивается работа с данными двухбайтового формата (модифицированный интерфейс).

В АПС предусматривается размещение двух блоков выхода на магистрали канала блоков сопряжения с интерфейсами ЕСИ1 и ЕСИ2, способных работать независимо и одновременно. Кроме того предусмотрен дополнительный блок-адаптер "канал-канал" (АКК). Этот блок предназначен для непосредственного подключения каналов ЕС-ЭВМ, а также для организации проверки работы интерфейса (магистралей канала МК1 и МК2) с помощью тестов.

АКК имеет два дополнительных выхода - магистрали абонента МА1 и МА2, включаемых дополнительно к обычным абонентам на магистрали канала.

Блок-схема АПС приведена на рис. 66. Он состоит из следующих узлов:

Адаптера канала 2-го уровня (АК2);

Блока управления с буфером (БУ), хранящим управляемую информацию (приказы, состояния и т.п.) для 32 одновременно работающих подканалов;

Буфера данных (БД), обеспечивающего промежуточное хранение данных в объеме до 64÷128 байтов. Это позволяет устройствам, работающим монопольно, выходить в канал 2-го уровня в мультиплексном режиме;

Блоков сопряжения с магистралями интерфейса (БСИ1 и БСИ2).

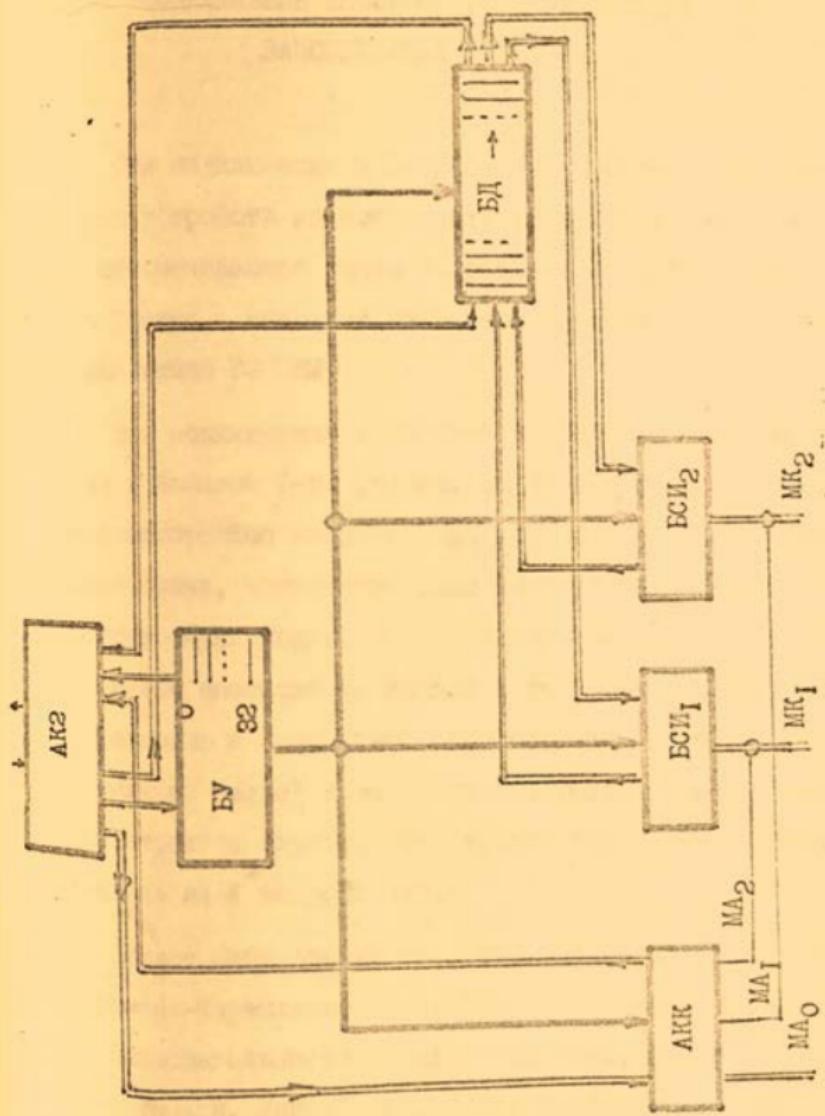


Рис. 66. Блок-схема адаптера канала ввода-вывода ЕС ЭВМ (ЛПС)

## ПОДКЛЮЧЕНИЕ ВЫСОКОСКОРОСТНЫХ ВНЕШНИХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

Для подключения к БЭСМ-10 высокоскоростных внешних запоминающих устройств из номенклатуры ЕС ЭВМ используется устройство, обеспечивающее связь оперативной памяти системы через канал I-го уровня с внешними накопителями через стандартный интерфейс ввода-вывода ЕС ЭВМ.

Для подключения к системе устройство имеет адаптер для связи с каналом I-го уровня. По отношению к подключенными с помощью интерфейса ввода-вывода устройствам управления внешними накопителями, устройство выполняет функции канала, работающего по блочно-мультиплексному принципу. Благодаря более высокой пропускной способности канала I-го уровня (60нсек на слово) по сравнению с пропускной способностью интерфейса ввода-вывода (2мксек на слово) и высокому быстродействию элементов, на которых строится адаптер, это устройство может мультиплексно обслуживать до 4 направлений.

Каждое направление представляет собой специализированное программно-управляемое устройство, программа которого состоит из последовательности управляющих слов, заготовленных в оперативной памяти. Управляющее слово сохраняет, в основном, структуру управляющего слова, принятого в ЕС ЭВМ, однако, оно помещено в одно 48-разрядное слово. Благодаря этому цепочки управляющих слов занимают меньше места в ОЗУ, а также уменьшается вдвое число обращений по их выборке. Ниже приведен формат управляющего слова.

АДРЕС ОЗУ	НПБ	ПРИЗНАКИ	СЧЕТЧИК СЛОВ	КОМАНДА
48	26 25 24 22 21		17 16	9 8

- Разряды I-8      - команда, выполняемая внешним ЗУ
- Разряды 9-16      - счетчик слов, задающий количество слов, передаваемых из или в ОЗУ
- Разряды I7-21      - признаки, задающие режим обмена и связывающие последовательно выполнение управляемые слова
- Разряды 22-24      - номер последнего байта в последнем передаваемом слове, позволяющий заканчивать обмен на заданном байте последнего слова
- Разряд 25      - признак, показывающий, что начало передаваемого массива совпадает с началом правого полуслова. Отсутствие этого признака означает, что начало передаваемого массива совпадает с началом слова.
- Разряды 26-48      - начальный адрес области ОЗУ, с которой ведется обмен.

Таким образом, выбранная структура управляющего слова позволяет пересыпать массивы переменной длины, не превышающие по объему страницу (256 слов), причем начало массива должно совпадать с началом полуслова, а конец задается с точностью до байта. Возможность передачи массивов, превышающих по объему страницу, достигается наличием механизма цепочки данных (один из признаков).

Все направления, подключенные к одному адаптеру канала I-го уровня, идентичны и состоят из следующих основных блоков: блока управления, блока буферных регистров и блока связи с интерфейсом ввода-вывода (см. рис. 67 ).

канал I-го уровня

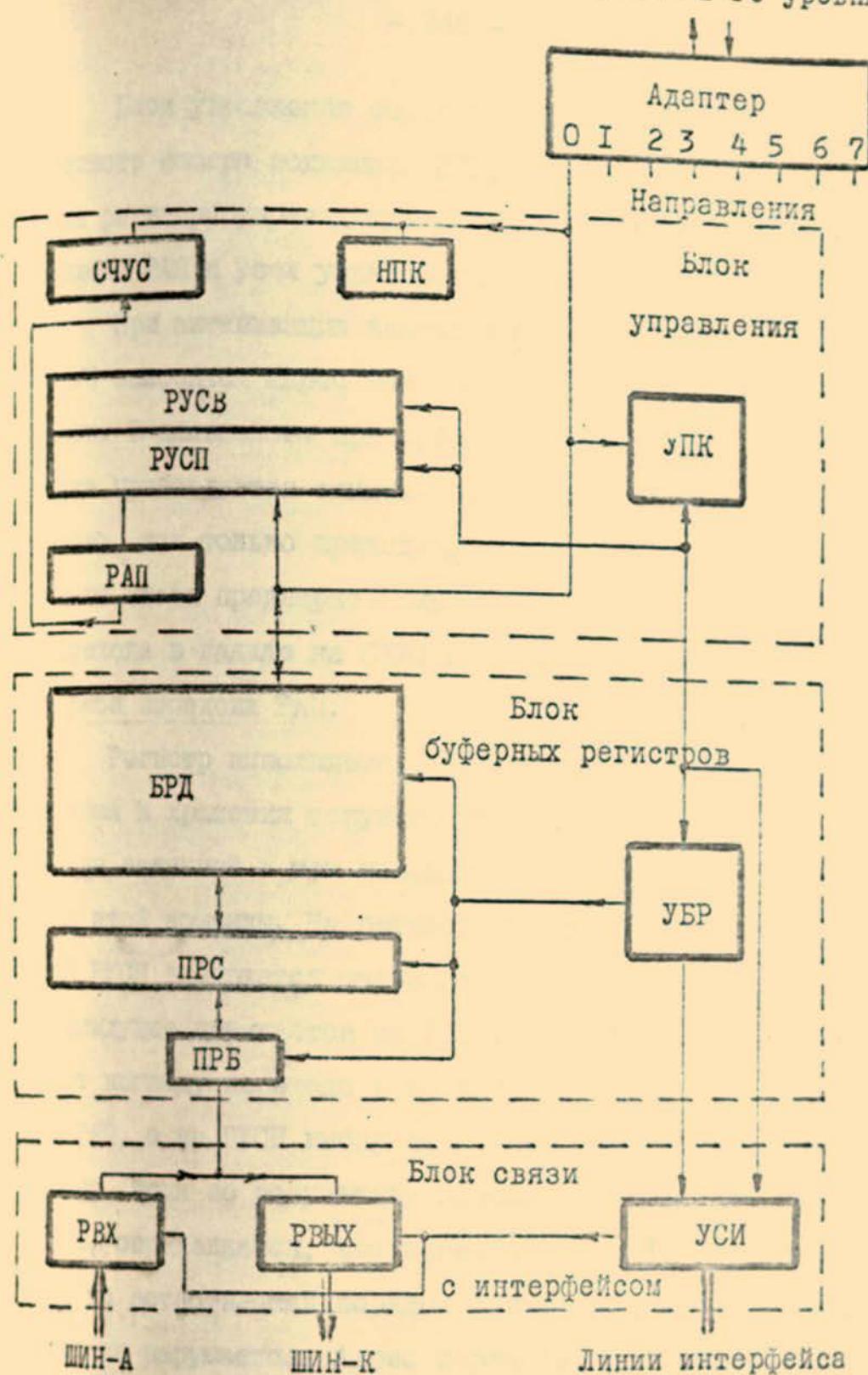


Рис. 67.

Блок управления содержит счетчик управляющих слов СЧУС, регистр номера подканала НПК, предварительный и исполнительный регистры управляющих слов РУСП и РУСВ, регистр адреса перехода РАП и узел управления подканалом УПК.

При активизации направления на счетчик управляющих слов СЧУС заносится адрес начала последовательности управляющих слов. В дальнейшем при работе направления к содержимому счетчика прибавляется единица для выборки очередного управляющего слова, как только предыдущее будет принято на регистр управляющего слова предварительный РУСП. В случае выполнения команды перехода в канале на СЧУС заносится адрес перехода с регистра адреса перехода РАП.

Регистр выполняемого управляющего слова (РУСВ) служит для приема и хранения текущего управляющего слова на время выполнения заданной в нем команды и для управления ходом выполнения этой команды. На регистр управляющего слова предварительный РУСП выбирается очередное управляющее слово, как только предыдущее передается на РУСВ. Если это управляющее слово заставляет команду перехода в канале, то адрес перехода записывается на РАП, а на РУСП выбирается следующее по порядку управляющее слово. Если по результату выполнения команды, находящейся на РУСВ, определяется, что переход в канале не должен выполняться, то естественный порядок выполнения и выборки управляющих слов не нарушается. Адрес перехода, находящийся на РАП, считается недействительным, а РАП — свободным. Если же определяется, что необходимо выполнить переход, адрес перехода с РАП пересыпается на СЧУС, и далее управляющие слова выбираются, начи-

чиная с этого адреса, причем выбранное ранее на РУСП управляющее слово аннулируется. Механизм использования регистров РУСИ, РУСП и РАП позволяет реализовать в промежутках между информационными областями на носителе достаточно быструю смену команд даже тогда, когда эти команды разделены командой перехода в канале.

Блок буферных регистров состоит из буферных регистров данных (БРД), промежуточного регистра слова ПРС, промежуточного регистра байта ПРБ и узла управления буферными регистрами УБР.

Имеется 16 буферных регистров данных, которые предназначены для сглаживания неравномерности поступления информации по каналу I-го уровня. При выполнении команд типа "запись" устройством вырабатываются сообщения-запросы данных из ОЗУ, отправляемых по каналу I-го уровня. Эти запросы могут быть двух типов: одиночные и групповые. Одиночный запрос - это запрос одного слова данных, групповой - это запрос группы, состоящей из 8 последовательно расположенных в ОЗУ слов. Тип запроса определяется двумя факторами - количеством слов, которое осталось еще запросить в данный момент из ОЗУ для завершения операции и наличием свободного места в БРД. Групповые запросы существенно снижают вероятность возникновения ситуации "переполнение в канале" при выполнении команд типа "запись". В случае работы по групповым запросам БРД представляет собой по сути дела два полукармана, каждый из которых состоит из 8 регистров.

Промежуточный регистр слова ПРС позволяет упростить связь между буферными регистрами данных и промежуточным регистром байта ПРБ. ПРБ служит для выдачи и приема информации всех ви-

лов, проходящей по информационным шинам интерфейса в обоих направлениях. На промежуточном регистре байта осуществляется побайтовый контроль передаваемой информации.

Блок связи с интерфейсом и его узел управления сигналами интерфейса УСИ обеспечивает выработку всех последовательностей сигналов, необходимых для работы интерфейса.

## СРЕДСТВА СОПРЯЖЕНИЯ С КАНАЛАМИ СВЯЗИ

Для использования БЭСМ-10 в системах реального времени, автоматизированных системах управления, системах коллективного доступа необходимо обеспечить непосредственное подключение (без промежуточных носителей) каналов связи. По этим каналам организуются тракты передачи данных для обмена с удаленными корреспондентскими пунктами. В зависимости от класса корреспондента (от измерительного пункта наземного контрольно-измерительного комплекса до пишущей машинки программиста в системе коллективного доступа) необходимо обеспечивать обмен с широким диапазоном скоростей  $50 \div 48000$  бод, дальности ( $1 \div 10.000$  км) и достоверности ( $10^{-3} \div 10^{-9}$ ). Это разнообразие определяет необходимость программного управления в средствах сопряжения.

Анализируя виды каналов связи, которые могут быть предоставлены для передачи, можно выделить три типа каналов: телефонный, телеграфный и широкополосный. Для каждого из этих каналов характерна некоторая совокупность функций, необходимых в сопряжении. Эта совокупность определена физической реализацией канала и практически не зависит от цепей передачи данных примененного алгоритма защиты, алгоритма управления корреспондентским пунктом. Поэтому для средств сопряжения БЭСМ-10, так же как и для АС-6 предлагаются аппаратно-программный метод решения задач сопряжения.

Разделение задач между специализированной аппаратурой и программами следующее. Специализированная аппаратура решает "стабильные" функции сопряжения, которые не зависят от типа корреспондентского пункта, а определены лишь свойствами используемого вида канала связи. Программное оборудование средств связи решает "динамичные" функции сопряжения, состав и содержание которых существенно зависит от типа корреспондентского пункта и цепей обмена. Специализированная аппаратура представлена в БЭСМ-10 устройствами следующих типов:

Телеграфный мультиплексор МТГ-10, рассчитанный на подключение телеграфных каналов связи всех видов - выделенных, коммутируемых и физических пар.

Телефонный мультиплексор МТФ-10 и адаптеры, рассчитанные на подключение модемов (устройства преобразования сигналов-УПС). Модемы, в свою очередь, подключаются к стандартным каналам тональной частоты или к первичным 12-канальным группам с шириной полосы пропускания 3,1  $K_{Hz}$  и 48  $K_{Hz}$  соответственно.

Консервативность принципов работы каналов позволяет ввести аппаратную реализацию части функций. С другой стороны, прогнозы нагрузки на средства сопряжения в условиях существующего дефицита каналов связи вызывают требование снижения затрат на специализированное оборудование, которое не может быть производительно использовано в те проме-

жутки времени, когда центр обработки не имеет связи с корреспондентами и работает в режиме традиционного ВЦ.

Динамическая часть задач сопряжения решается программами, которые выполняются на периферийной машине ПМ-10, так же как в АС-6-на ПМ-6. В частности, к этим задачам относятся индивидуализированные для данного алгоритма защитные функции - способ организации обратной связи с корреспондентом, определение длины линейного блока, глубины буферной области переспроса, способа помехоустойчивого кодирования и т.д. Результатами работы программ периферийной машины являются массивы данных, полученных от корреспондентов, достоверность и время доставки которых соответствует требованиям к данному типу корреспондента. Все разнообразие алгоритмов этих задач оказывается скрытым от пользователя и даже от операционной системы центрального процессора.

Выбранное для БЭСМ-10 разделение задач сопряжения между программным и аппаратным оборудованием проверено в системе АС-6. Так для первой очереди АС-6 было разработано и введено в эксплуатацию только для обслуживания обменов на ТГ каналах 7 алгоритмов дальних связей, программы протоколирования работы средств сопряжения, ведения астрономического времени (календаря) с точностью до секунд, программы диалога операторов и средств сопряжения.

Программы обслуживания дальних связей включают ветви анализа отказов канала, восстановления после отказов или

кратковременных перерывов связи, сбор статистической информации для инженерной оценки качества сеанса связи, протоколирования процесса обмена с удаленным пунктом. Подробнее задачи такого рода программы рассмотрены ниже, в разделе "ориентированное математическое обеспечение".

Учитывая совместимость системы команд ПМ-Ю и ПМ-6 и систем приказов специализированной аппаратуры-мультимплексоров, можно считать, что программы обслуживания дальних связей, которые разработаны для АС-6, будут использованы и в БЭСМ-Ю.

### ТЕЛЕГРАФНЫЙ МУЛЬТИПЛЕКСОР

#### Функции мультиплексора

Телеграфные каналы, о которых идет речь при постановке задачи сопряжения с ЭВМ - это каналы телеграфирования постоянным током. К таким каналам на окончных пунктах подключаются буквопечатающие телеграфные аппараты со стартстопным принципом работы. Рассматривая вопросы сопряжения с телеграфными каналами можно ограничиться рассмотрением только таких каналов, оставляя в стороне вопросы образования телеграфных каналов с помощью уплотнения телефонных каналов - телеграфирование переменным током. В настоящее время основная масса телеграфных каналов образована способом уплотнения, но окончания этих каналов подготовлены для включения телетайпов. Возможность и в то же

время необходимость ограничения сопряжения с ТГ-каналами - как каналами телеграфирования постоянным током определяется огромным масштабом телеграфных сетей и эксплуатируемой на них аппаратурой и, в силу этого, большой консервативностью принципа работы таких сетей. Консервативность принципов работы каналов, которую можно оценить в 15-20 лет, позволяет многие функции сопряжения реализовать в аппаратуре с жестким способом работы. Однако следует подчеркнуть, что эти функции относятся только к непосредственному сопряжению с каналом, но не относятся к алгоритмам передачи данных.

Для системы АС-6 разработано и введено в эксплуатацию специализированное устройство сопряжения с телеграфными каналами - телеграфный мультиплексор МТГ, которое является прототипом для мультиплексора МТГ-10 системы БЭСМ-10. В МТГ-10 предполагается повторить структуру, принцип работы, набор программных приказов, реализованные в МТГ системы АС-6, с некоторым расширением возможностей. Это расширение касается работы с 7-ми элементным знаком и программного управления скоростью обмена.

Основной набор функций мультиплексора, разработанный и отработанный в АС-6 содержит:

преобразование байтов данных в стартстопные комбинации, передаваемые в канал;

прием стартстопных комбинаций и выделение информационных элементов данных;

преобразование сигналов по электрическим параметрам.

Эти процедуры характерны как для коммутируемых (выделенных) каналов, так и для сетей абонентского телеграфа (АТА) и прямых соединений (ПС). Кроме таких функций для телеграфного мультиплексора введена аппаратная реализация отдельных операций по взаимодействию с коммутационной станцией в процессе установления соединения:

вызов станции и анализ ответа станции "приглашение к набору";

выдача цифры номера абонента;

выдача интервала между цифрами с анализом возможной занятости приборов станции;

анализ результатов набора "занято" или "соединение установлено".

Все перечисленные функции являются стабильными, так как обусловлены основными свойствами каналов и сетей.

Преобразование байта данных в стартстопные комбинации и обратно делается для того, чтобы свести сигнал, исходящий из ЭВМ к виду сигнала, исходящего из телетайпа. Такое преобразование требует разбиения последовательности данных на 5-разрядные отрезки, хотя данные, которыми обменивается ЭВМ с корреспондентами, не имеют структуру 5-разрядных знаков. Стартстопные комбинации с 5-элементными отрезками данных базируются на 5-элементном первичном коде МТК-2,

принятом в качестве стандартного в 1932 году. Впоследствии, в 1962 г. начата разработка (в рамках МККТТ) нового семиэлементного кода МТК-5, который вскоре был утвержден, а в 1967 году был утвержден код ГОСТ 13052-67, разработанный на базе МТК-5. В настоящее время в рамках проектов СЭВ ведется разработка телетайпов, использующих семиэлементные коды.

За 40 лет, которые прошли с начала внедрения МТК-2, произошли большие изменения и в аппаратуре образования каналов, которая стала свободной от требования передачи абонентами только стартстопных 7,5-элементных комбинаций (аппаратура ТТ-17, "Десна" и т.д.).

Однако в настоящее время в предоставляемых каналах связи возможна аппаратура, ограничивающая вид сигнала абонента только 7,5-элементной стартстопной комбинацией (рекомендательные трансляции).

В связи с этим предложенный для МТГ системы АС-6 набор функций должен стать основой для набора функций МТГ-10.

Расширение набора содержит:

преобразование байта в стартстопную комбинацию, содержащую 7 информационных элементов (и обратно);  
программное задание длины стоповой коррекционной посылки (1,0; 1,5; 2,0 То, где То - длина информационной посылки);

возможность программного задания скорости передачи (50,100,200 бод).

Набор операций для установления соединения следует оставить без изменений, ибо в настоящее время не наблюдается тенденций к какому-либо изменению процедур коммутации. Этот набор выбран в основном из соображений удобства программирования процедуры. Алгоритм, по которому строится программа, аналогичен алгоритму поведения человека, проводящего соединение по городской телефонной сети.

### Принцип построения

Алгоритм. Алгоритм, по которому ведется прием знаков данных в телеграфном мультиплексоре, состоит в том, что проводится периодическое сканирование согласующих элементов-приемников. Состояние (значение сигнала в приемниках) канала анализируется в каждой точке сканирования, пока не обнаружится стартстопный переход. Затем отсчитываются интервалы времени и опрашивается состояние канала в предполагаемых серединах посылок, которое и принимается в качестве значений информационных посылок.

Требуемая частота сканирования получается из расчета желаемой исправляющей способности реализуемого приема. Теоретическая исправляющая способность приемника, работающего по такому алгоритму, и частота сканирования связаны следующим образом:

$$M = 0,5 - \frac{1}{2n} \quad (\text{при } n \text{ -нечетном})$$

$$M = 0,5 - \frac{1}{n} \quad (\text{при } n \text{ -четном})$$

где  $n$  - кратность частоты сканирования  $f$  и скорости манипуляции  $V$ .

$$f = n \cdot V$$

Конкретные значения зависимости  $M$  от целых  $n$ :

$n$	16	31	32	63	64
$M$	44	49	47	49,6	49,2 (в процентах)

В МТГ было принято  $N = 32$ . Результаты эксплуатации на реальных каналах связи показали достаточность такого соотношения скорости и частоты. Это соотношение было выдержано для всех скоростей манипуляции, что обеспечило одинаково высокую исправляющую способность мультиплексора.

В качестве задающих генераторов должны быть применены генераторы с высокой ( $10^{-5} \div 10^{-6}$ ) стабильностью частоты и не требующие первоначальной настройки. В МТГ применена камертонная стабилизация. Вид стабилизации (кварц, камертон) в МТГ-10 будет уточнен на этапе технического проектирования.

Применение высокоскоростной системы элементов позволяет свести к минимуму индивидуальное оборудование канала, которое содержит только согласующие оконечные элементы и ячейку буферной памяти. Все необходимые действия по обработке сигналов канала выполняются общим оборудованием. Тогда это общее оборудование (блок обработки) должен обслуживать все  $N$  каналов за интервал  $t_0$  между двумя метками времени сканирования канала. Для максимальной скорости телеграфирования  $V = 200$  бод и соотношения  $\phi = 32V$  на обслуживание одного канала отводится 4,8 или 2,4 мксек, если число каналов  $N = 32$  и 64 соответственно. В МТГ число каналов равно 32. В МТГ-10 число каналов увеличено до 64, учитывая возросшее быстродействие системы элементов.

Объем памяти. Следующий фактор, ограничивающий число каналов, является объем буферной памяти. Так в МТГ на обслуживание одного полудуплексного обмена требовалось 18 бит памяти. Учитывая расширение программно-управляемых функций мультиплексора МТГ, потребуется увеличение объема памяти, отводимого на обслуживание одного канала.

На рис. 68 приведены распределение полей ячейки буферной памяти МТГ АС-6 и предполагаемое распределение в МТГ-10. Рассмотрим режим приема, наиболее емкий в смысле требуемого объема буферной памяти. Наименования полей:

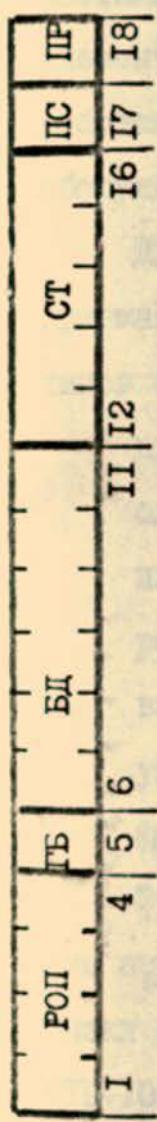
- РОП - режим и операция (4 бита);
- ГБ - готовность буфера к обмену (1 бит);
- БД - буфер данных (5 и 7 бит);
- СТ - счетчик времени (5 бит);
- ПС - признак поиска старта (1 бит);
- ПР - состояние прерывания (1 бит)

и вновь вводимых полей:

- Ф - формат (1 бит) 5 элементов или 7 элементов;
- НС - номер скорости (2 бита);
- КД - контактное деление (2 бита).

Как видно из сравнительного рисунка, объем памяти расширен до 25 бит на один полудуплексный канал.

Согласующие элементы. Верхняя граница числа обслуживаемых каналов определяется еще одним фактором - физическим объемом согласующих элементов. Согласующие элементы,



Распределение полей ячейки в МТр АС-6



Распределение полей ячейки в МТр-10

Рис. 68. Распределение полей ячейки буферной памяти

так же как и память, являются индивидуальным оборудованием канала и объем их линейно зависит от числа каналов. Так в МТГ согласующие элементы составили около 10% общего объема устройства при числе каналов 32, в то время как память составила 30%. Уменьшение объема согласующих элементов достижимо труднее, чем уменьшение объема памяти, так как элементы содержат мощные оконечные транзисторы, которые работают непосредственно на электромеханические элементы оборудования канала связи - реле и электромагниты.

Для МТГ разработаны универсальные элементы сопряжения - приемники и передатчики, допускающие любой из существующих типов окончания телеграфного канала:

двух- и четырехпроводное;

одно- и двухполюсную работу;

питание линейных цепей от станции или от мультиплексора;

выделенные каналы и абонентские линии коммутационных узлов;

физические пары.

Эти универсальные элементы, использующие промежуточную амплитудную и фазовую модуляцию для гальванической развязки с линейными цепями являются прототипом для элементов МТГ-Ю.

Ожидается, что физический объем этих элементов позволит увеличить число каналов до 64-128 в одном мультиплексоре.

### Система приказов

Основной вопрос разработки системы программных приказов - совместимость по отношению к системе приказов МТГ АС-6. Система приказов МТГ АС-6 полностью вложена в систему приказов МТГ-IO. Как видно из предыдущего изложения состав функций МТГ-IO расширен по отношению к функциям МТГ АС-6, что, естественно, требует расширения системы приказов. Расширение будет выполнено за счет резервных кодов приказа.

Приказ имеет формат байта и состоит из двух полей:

тип приказа (ТП);

код приказа (КП);

каждое из которых имеет формат 4 разряда.

ТП	КП
0	3 4

В МТГ АС-6 был введен следующий набор приказов:

ТП КП

0 0 - прекратить обмен,

0 У - опросить группу признаков с номером У,

8 У - установить режим У,

9 У - начать сканирование группы признаков У,

А У - установить режим У и пустить имитатор  
канала связи.

Для МТГ-Ю этот набор будет расширен:

7У - установить режим У для работы с 7-ми элементным кодом;

6У - уточнить описание канала кодом У.

Приказ 8У из предыдущей системы будет интерпретироваться как "установить режим У для работы с 5-элементным кодом", что соответствует воспроизведению функций МТГ АС-6. Между значением У поля КП в приказах установки режима и собственно режимом имеется следующее соответствие:

У = 0 - выключить

У = 1 - опознать входящий вызов

У = 2 - начать исходящий вызов

У = 3 - выполнить ОТБОЙ по коммутируемой сети

У = 4 - начать передачу

У = 5 - восстановить канал

У = 6 - начать передачу с контролем линейного исходящего тока

У = 7 - начать имитационную передачу

У = 8 - начать прием

У = 9 - начать прием с проверкой ВК

У = А - начать имитационный прием

У = В - начать имитационный прием с проверкой ВК

У = С - начать прием с генерацией меток времени

У = Д - начать прием с проверкой ВК и генерацией меток времени

$Y = E$  - начать имитационный прием с генерацией меток времени

$Y = F$  - начать имитационный прием с проверкой ВК и генерацией меток времени

Режимы приема с проверкой ВК введены для упрощения программирования диалога. В этих режимах аппаратура проверяет каждый принятый символ на соответствие с фиксированным:

00010 ("возврат каретки") при 5-элементном коде

00000II ("конец текста") при 7-элементном коде

Обнаруженный байт оформляется в системном канале так, чтобы при приеме в ОЗУ через ПМ-10 вызвать завершение исполнения обменной команды приема данных. Это дает возможность работать с переменной длиной строки вводимого текста.

Режимы приема с генерацией меток времени введены для упрощения программирования ветвей временного контроля процессов обмена в данном канале. Временный контроль повсеместно используется при обмене с удаленными корреспондентами. Так, ни один из 7 алгоритмов обмена, реализованных в АС-6 в 1973 году, не обходился без временного контроля, особенно на этапах вхождения в связь. Приказы "уточнить описание канала связи" вводятся для программного управления скоростью и контактным делением в стартстопной комбинации:

$Y = 0$  скорость фиксированная, контактное деление (КД) равно  $n + 1,5$ , где  $n$  - формат знака (= 5 или 7)

$Y = 1$  скорость 50 бод, КД =  $n + 1,5$

$Y = 2$  скорость 100 бод, КД =  $n + 1,5$

$Y = 3$  скорость 200 бод, КД =  $n + 1,5$

$Y = 4$  скорость фиксированная, КД =  $n + 1,0$

$Y = 5$  скорость 50 бод, КД =  $n + 1,0$

$Y = 6$  скорость 100 бод, КД =  $n + 1,0$

$Y = 7$  скорость 200 бод, КД =  $n + 1,0$

$Y = 8$  скорость фиксированная, КД =  $n + 1,5$

$Y = 9$  скорость 50 бод, КД =  $n + 1,5$

$Y = A$  скорость 100 бод, КД =  $n + 1,5$

$Y = B$  скорость 200 бод, КД =  $n + 1,5$

$Y = C$  скорость фиксированная, КД =  $n + 2,0$

$Y = D$  скорость 50 бод, КД =  $n + 2,0$

$Y = E$  скорость 100 бод, КД =  $n + 2,0$

$Y = F$  скорость 200 бод, КД =  $n + 2,0$

При фиксированной скорости ее значение определяется с помощью ДЗУ, в котором задана характеристика канала ( $Y = 45, 50, 75, 100$ ) – так же как и в МТГ АС-6. Если программа не уточнит описания канала приказом, то параметры канала будут выбраны из ДЗУ.

Для проведения процедур исходящего вызова мультиплексор использует "данные управления", которые должны

выдаваться из ПМ-10 в ответ на запрос из МТГ-10. Кодировка байта данных управления сохраняется такой же, как принятая для МТГ АС-6, где она составлена с учетом рекомендации МККТТ V24. Байт имеет вид ОУ, где соответствие параметра У и цифр номера или служебных функций:

У = 1 - цифра номера "1"

У = 2 - "2"

У = 3 - "3"

У = 4 - "4"

У = 5 - "5"

У = 6 - "6"

У = 7 - "7"

У = 8 - "8"

У = 9 - "9"

У = 0 - "0"

У = А, В - резерв

У = С - конец набора (EON)

У = Д - разделитель (Sep)

У = Е, F - резерв

Служебная функция "конец набора" означает "перейти" к анализу результатов набора" (занято или соединение установлено).

Служебная функция "разделитель" означает выполнение интервала между цифрами и анализ сигналов станции с целью опознавания возможного состояния занятости приборов станции.

### Структура мультиплексора

Основными узлами мультиплексора сопряжения с телеграфными каналами являются:

общий для всех каналов обрабатывающий блок (блок обработки);

блок оперативной буферной памяти, каждая ячейка которого отведена для обслуживания отдельного канала;

блок согласующих элементов.

Кроме этого мультиплексор должен содержать дополнительные блоки, предназначенные для оформления связей устройства с системой и организации внутренних циклов обслуживания многих каналов.

Устройство (рис. 69) содержит блок сопряжения с системным каналом. Этот блок выполняет оформление обменов мультиплексора с остальной частью системы в виде стандартных процедур, принятых для унифицированного системного канала.

Генераторы меток времени – блок, вырабатывающий метки для сканирования состояния канала и выполнения основного алгоритма обслуживания.

Блок управления обеспечивает взаимодействие всех блоков устройства и, главное, разделение времени блока обработки для обслуживания многих каналов.

Ячейка памяти, отводимая для обслуживания одного канала, разбивается на несколько полей, причем это разбиение

каналы связи

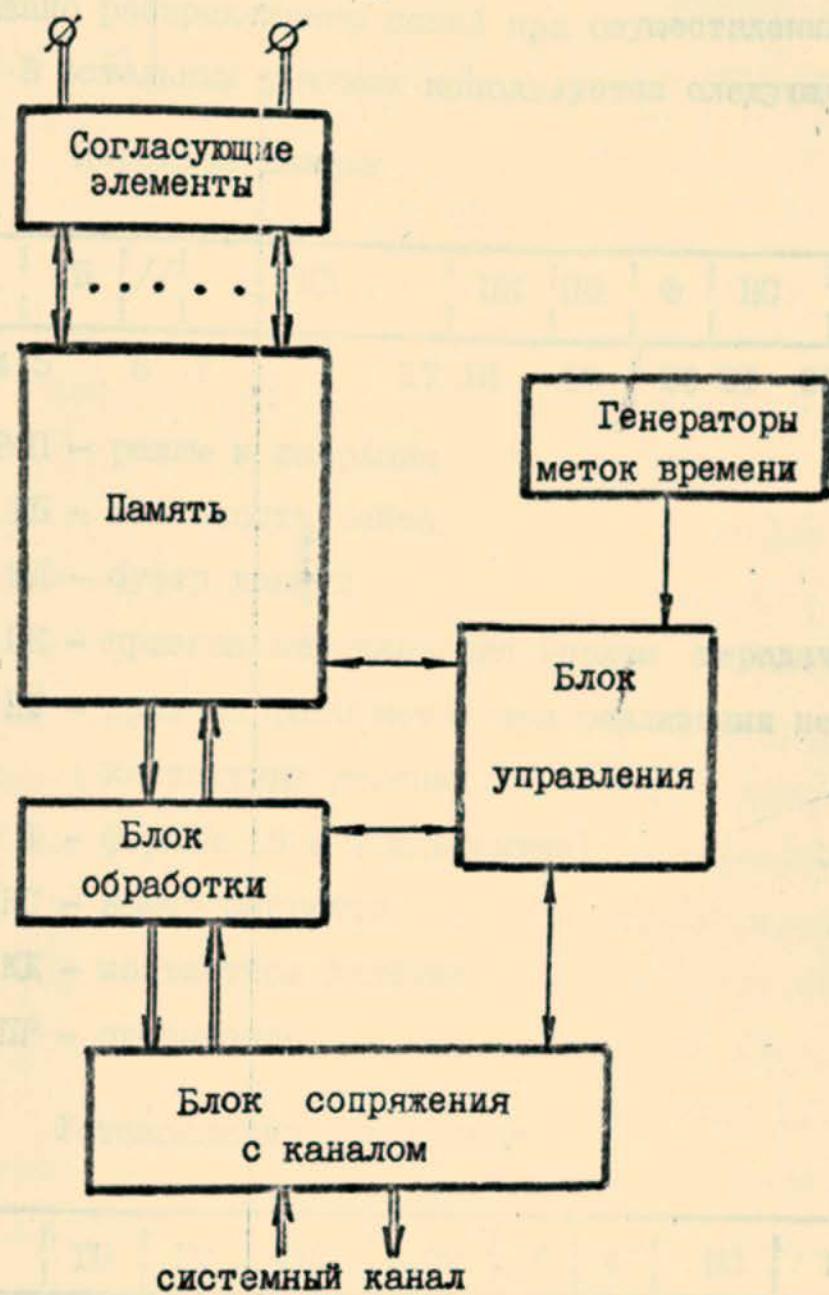


Рис. 69. Блок-схема телеграфного мультиплексора

различно для реализации разных режимов. Выше (рис. 68) показано распределение полей при осуществлении приема данных. В остальных режимах используется следующее разбиение:

Передача данных

! РОП ! ГБ ! // !	БД	! ПК ! ПФ ! Ф ! НС !	КД !	ПР
I 4 5 6 7	I7 I8	I9 / 20 21	22 23	24 25

где РОП - режим и операция

ГБ - готовность байта

БД - буфер данных

ПК - признак переключения (прием передача)

ПФ - признак фазы меток при реализации нецелых контактных делений

Ф - формат (5 и 7 элементов)

НС - номер скорости

КД - контактное деление

ПР - прерывание

Установление соединения

! РОП ! ГБ ! УТ ! СС ! СТ ! // ! Ф ! НС ! КД ! ПР	
I 4 5 6 9 10 13 И 18 19 20 21 22 23 24 25	

где УТ - указатели амплитуды тока

СС - состояние соединения

СТ - счетчик времени

Указанные поля позволяют хранить описание процесса обслуживания канала вплоть до процесса опознавания стартовых/переходов, элементов данных, сигналов станции при установлении соединения.

### ТЕЛЕФОННЫЙ МУЛЬТИПЛЕКСОР

#### Функции мультиплексора

Телефонные каналы – самое распространенное средство для электросвязи, предназначенное для передачи речевой информации. Основное отличие телефонных каналов от телеграфных каналов с точки зрения задач сопряжения – отличие в спектре передаваемого сигнала. Полоса ТФ-каналов  $0,3\pm 3,4$  кГц, а полоса ТГ-каналов  $0\pm 0,3$  кГц. В ТФ-каналах отсутствует постоянная составляющая сигнала и ширина полосы на порядок превосходит ширину полосы ТГ-каналов. Широкополосные каналы, по которым возможна высокоскоростная передача данных до 40 Кбод, представляют собой первичные группы аппаратуры уплотнения линий связи с полосой  $60\pm 108$  кГц. Обычно такие каналы уплотняются 12 телефонными каналами (12-канальная первичная группа), но могут предоставляться и целиком – без разделения на узкополосные стандартные каналы. Такие группы образуются в настоящее время сериейной аппаратурой К-60, К1920, которые работают по кабельным линиям связи, что обеспечивает высокое качество каналов, пригодных для высокоскоростной передачи данных. Широкополосный канал также

не содержит нулевой компоненты сигнала и превосходит телефонные каналы по ширине спектра на порядок. Однако общность широкополосных и узкополосных телефонных каналов в необходимости модуляции и демодуляции сигнала данных при передаче по каналу связи. Поэтому главной функцией специализированной аппаратуры сопряжения является преобразование сигнала данных с учетом полосы ТФ-канала.

На современном уровне развития техники обработки сигнала эта функция может быть решена аналоговыми средствами, индивидуальными для каждого канала. В термин "преобразование" вкладывается значительно больший состав действий, чем только модуляция или демодуляция. Сюда входит коррекция характеристик ТФ-каналов, фильтрация сигнала и синхронизация по элементам. Модемы, которые предполагается использовать на современных отечественных сетях связи, должны обладать одним специальным качеством — по команде от аппаратуры центра должен устанавливаться режим "не данные". В этом режиме в телефонный канал, подключенный к модему, не выдаются никакие сигналы. Режим необходим в том случае, когда в алгоритме обмена не предусмотрено автоматическое (без участия оператора) решение всех конфликтных ситуаций. Тогда по тому же каналу связи можно организовать переговоры операторов центра и корреспондентского пункта.

Современные системы сопряжения рассматривают не телефонные каналы (как непрерывные каналы связи), а дискретные каналы, построенные на базе телефонных каналов.

Дискретный канал включает в себя непрерывный канал связи (НКС) и устройство преобразования сигнала (УПС) или модем.

Для того чтобы обеспечить гибкость аппаратных средств сопряжения по отношению к типам модемов, в АС-6 был введен универсальный стык между "стабильной" частью аппаратуры и адаптерами - "динамической" частью аппаратуры, существенно зависящей от типа примененного модема. Этот же стык предлагаются применить и в аппаратуре сопряжения в БЭСМ-10.

"Стабильная" часть аппаратуры - мультиплексор выполняет такие задачи, которые являются общими для многих алгоритмов повышения достоверности:

преобразование последовательно-параллельных кодов и форматов данных;

поиск фазирующих последовательностей в потоке принимаемых элементов.

Кроме этого мультиплексор осуществляет:

обработку команд, полученных из периферийной машины для модемов в сигналы универсального стыка;

преобразование сигналов, входящих из адаптеров по универсальному стыку, в унифицированную форму сообщений, передаваемых по системному каналу.

"Динамическая" часть аппаратуры - адаптеры, существенно учитывают различие специализированных стыков между модемом и системой. Состав задач адаптеров следующий:

декодирование команд для модемов, поступающих по универсальному стыку, в последовательность сигналов

на специализированном стыке с модемом;

преобразование сигналов управления, поступающих из модема по цепям специализированного стыка, в обобщенную форму сигналов универсального стыка.

На рис. 70 приведена блок-схема аппаратуры сопряжения с телефонными каналами, на которой показаны категории используемых стыков.

#### Принцип реализации

Синхронизация. Основное предположение, определившее способ обработки данных в телефонном мультиплексоре, состоит в том, что данные, поступающие из дискретного канала (через адаптер), представлены в виде пары сигналов – элемента данных и синхронизирующего сигнала. При осуществлении передачи в дискретный канал мультиплексор обеспечивается синхронизирующими сигналами. Источник синхронизации находится в модеме (например, модемы 153387, 5127). Для того, чтобы аппаратура сопряжения обеспечивала возможность работы с модемами, не имеющими источника синхронизирующего сигнала, при разработке АС-6 предусмотрена установка в адаптере общего задающего генератора с частотами 600, 1200, 2400 и 4800 Гц, сигналы которого могут быть введены в любой индивидуальный стык.

Мультиплексирование. Любая обработка элемента данных, т.е. обслуживание канала, начинается по требованию, сформированному при поступлении синхросигнала и требует времени  $T_k$ .

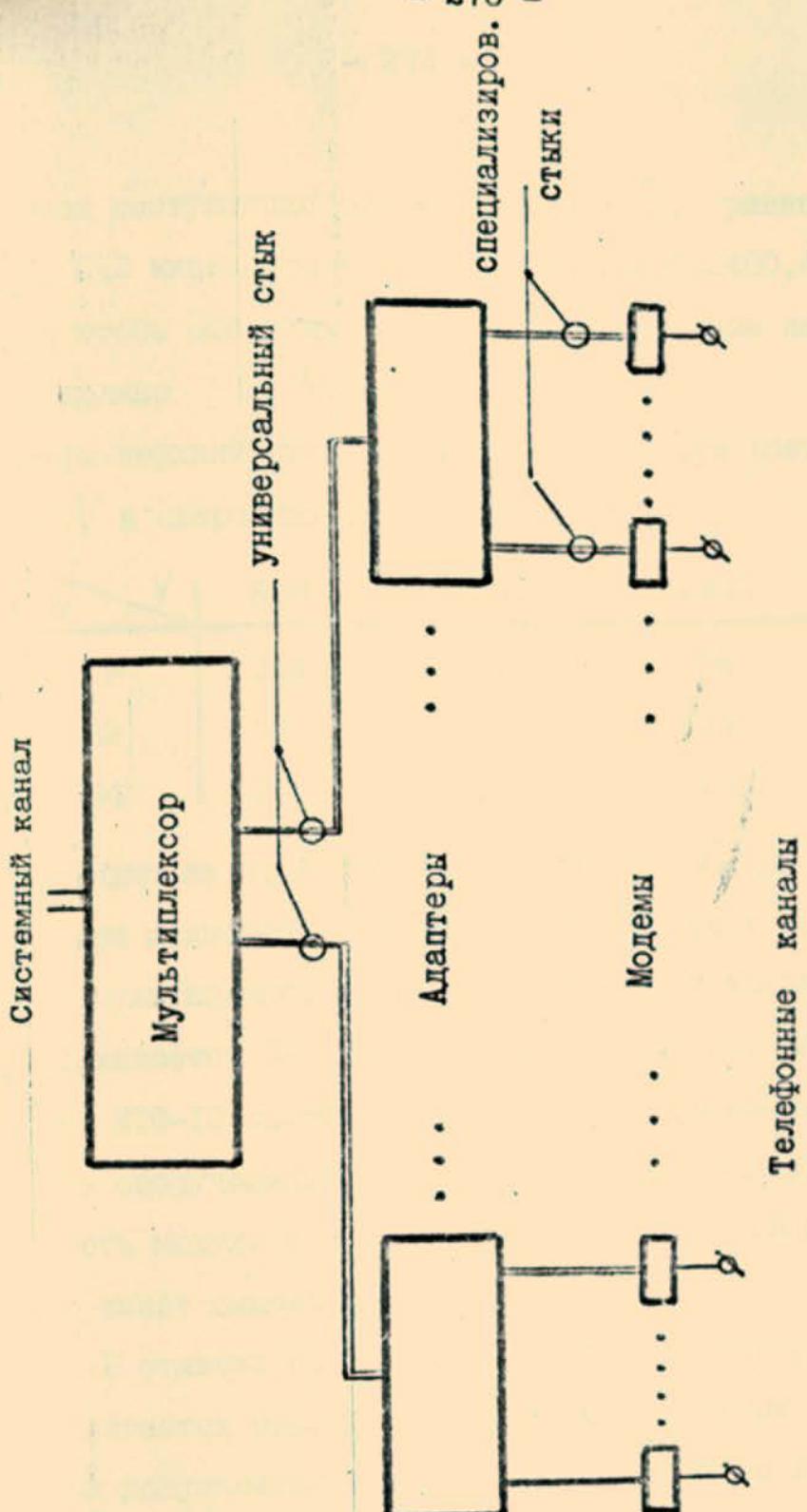


Рис. 70. Блок-схема специализированной аппаратуры сопряжения с телефонными каналами.

Период поступления синхросигналов  $T_{СИ}$  равен  $\sim 1660, 830, 415, 212$  мксек при скоростях 600, 1200, 2400, 4800 бод. Для того чтобы обслужить все  $N$  каналов одним обрабатывающим узлом должно  $N \cdot T_k < T_{СИ}$ .

Отсюда верхний предел  $T_k$  (мксек) при разном числе каналов  $N$  и скоростях модуляции  $V$ :

$N \backslash V$	600	1200	2400	4800
16	104	52	26	13
32	52	26	13	6,5
64	26	13	6,5	3,2

Как видно из этой таблицы, величины  $T_k$  достаточно большие, что при применении высокоскоростной системы элементов допускает мультиплексное обслуживание всех каналов. В МТФ АС-6 обслуживается 32 канала, и это число предлагается сохранить и для МТФ-10 системы БЭСМ-10. Если рассмотреть всю совокупность оборудования — мультиплексор, адаптеры, модемы, то емкость модуля в 32 канала представлется вполне достаточной и не имеет смысла ее увеличивать.

В отличие от МТФ АС-6 в мультиплексоре системы БЭСМ-10 предлагается ввести 2+4 привилегированных подканала, по которым допускается обмен до 48 Кбод, в то время как по остальным — до 4800 бод. Период следования синхросигналов при скорости 48 Кбод составляет  $\sim 2,1$  мксек. Обслуживание таких каналов станет возможным при увеличении быстродействия блока обработки на новой элементной базе. В МТФ АС-6 минималь-

ный цикл обслуживания канала - 0,8 мксек. За этот цикл могут выполняться: обращение в память (0,3 мксек), две арифметические операции ( $2 \times 0,2$  мксек), две операции сравнения при поиске фазы ( $2 \times 0,2$  мксек). Сумматор и компаратор работают параллельно. При трехкратном увеличении скорости цикл можно сократить до  $0,25 + 0,3$  мксек, тогда станет возможным обслуживание высокоскоростных каналов. Привилегированность подканалов, обслуживающих высокоскоростной обмен - в более высоком приоритете их доступа в блок обработки, чем подканалов, работающих со скоростью до 4800 бод.

Скоростные соотношения дают возможность реализовать мультиплексирование на уровне обработки одного элемента данных.

Объем памяти. Объем памяти, необходимый для обслуживания одного дискретного подканала определяется, в первую очередь, составом действий по обработке.

Преобразование "параллельно-последовательно" и обратно требуют введения буферной области, размер которой  $(n+1)$ бит, где  $n$  - формат системного канала. При байтовом канале ( $n = 8$ ) минимальный объем памяти - 9 разрядов.

Увеличение буферной области происходит из-за мультиплексной обработки каналов в периферийной машине, ибо данные должны храниться в мультиплексоре, пока не будут востребованы в машину. С точки зрения увеличения возможного резервного времени, в течение которого данные находятся в мультиплексоре, надо как можно больше увеличивать буферную

область. Однако увеличение буферной области памяти ограничено сверху требованием реактивности программ, или, что то же, ограничением задержки информации в буферной области. Ограничение связано с использованием алгоритмов повышения достоверности, в которых применяется:

обратная связь (решающая РОС или информационная ИОС);  
работа передатчиков без ожидания решения.

Увеличение и использование буферной области при таких алгоритмах снижает дальность, на которую рассчитан алгоритм. В таблице приведены численные значения снижения дальности (в тыс. километров) в зависимости от скорости  $V$  и объема буферной резервной области  $n_b$ .

Расчет сделан для случая межцентровой связи.

$n_b \backslash V$	600	1200	2400	4800
8	4,4	2,2	1,1	0,55
10	5,6	2,8	1,4	0,7
16	8,8	4,4	2,2	1,1

Отсюда был сделан выбор для МТФ АС-6 и МТФ-10, а именно  $n_b = 8$ .

Рассмотрение задержки в буферной области определило необходимость введения еще одной функции — преобразование формата. Поскольку большинство алгоритмов ориентированы на блочные коды и обработка ведется на границе, необходимо обеспечить:

минимальную задержку последнего элемента блока в памяти мультиплексора;

выдачу сигналов запуска программы в момент выдачи в системный канал последнего элемента блока;

одинаковую компоновку элементов блока в памяти программы для всех блоков одного формата, принятых из одного канала.

Преобразование формата заключается в выдаче (приеме) в системный канал любого заданного количества элементов К, причем

$$1 \leq K \leq 8$$

Задание формата также требует увеличения объема памяти.

В МТФ системы АС-6 объем буферной памяти достиг 33 элементов на канал. Ячейка памяти распределяется так:

БД	СБ	ФК
I	16 17	20 21
KЛ	СЧ	ГБ ПМ РЗУ РПР РОУ КВ ПР
25	26	27 28 29 30 31 32 33

где:

БД - бффер данных;

СБ - счетчик бит в буферной области;

ФК - формат канала;

КЛ - код из линии;

СИ - синхроимпульс элемента данных;  
ГБ - готовности буфера к обмену с ПМ-6;  
ПМ - признак прием/передача;  
РЗУ - маска универсального стыка (ЗУ);  
РПР - -"- -"- -" (ПР);  
РОУ - -"- -"- -" (ОУ);  
КВ - признак разрешения обмена с ПМ-6;  
ПР - признак состояния прерывания.

Для БЭСМ-10 введено расширение:

Поля ПМ до нескольких разрядов, позволяющих указать более детально состав обработки - формат фазирующей комбинации, вид кодзащиты и т.д.;

Поля ПР для указания категории причины, вызвавшей состояние прерывания (в АС-6 для этого используется часть поля БД);

Поля БД в режиме поиска фазирующих комбинаций для возможности оперирования с длинными последовательностями и для упрежденной подкачки в режиме передачи с длинными (240, 500, 960 бит) блоками.

Блок-схема. Блок-схема представлена на рис. 71  
Кроме обсуждавшихся блоков памяти и обработки реальное устройство содержит:

Блок сопряжения с системным каналом;  
Блок управления;  
Блок групповой части универсального стыка;

К адаптерам

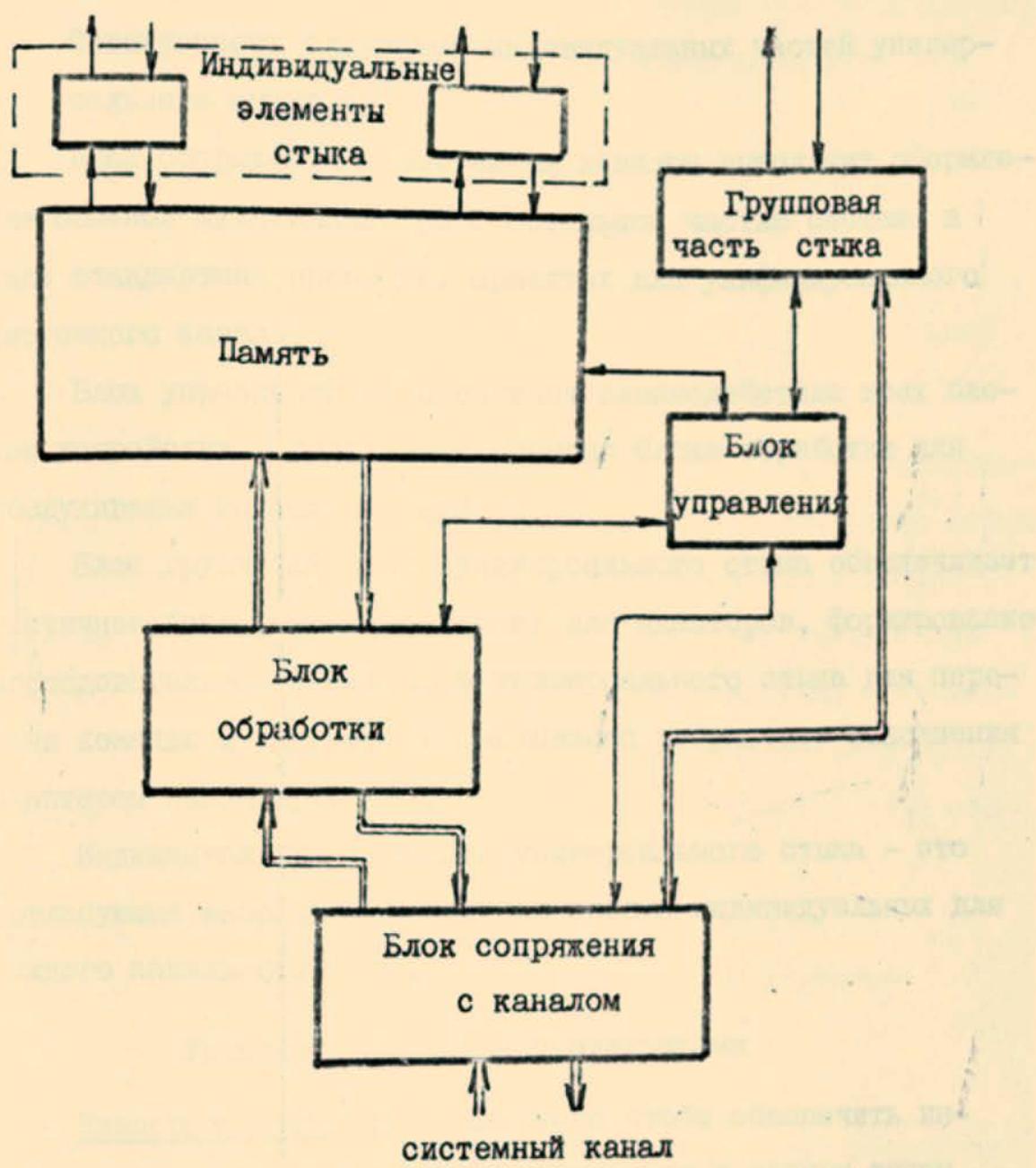


Рис. 71. Блок-схема телефонного мультиплексора

Совокупность элементов индивидуальных частей универсального стыка.

Блок сопряжения с системным каналом выполняет оформление обменов мультиплексора с остальной частью системы в виде стандартных процедур, принятых для унифицированного системного канала.

Блок управления обеспечивает взаимодействие всех блоков устройства и разделение времени блока обработки для обслуживания многих каналов.

Блок групповой части универсального стыка обеспечивает частичное дешифрирование команд для адаптеров, формирование последовательности сигналов универсального стыка для передачи команды в адаптер и считывания результата выполнения адаптером команды опроса.

Индивидуальные элементы универсального стыка - это согласующие кабельные усилители приема индивидуальных для каждого канала сигналов.

#### Универсальный стык с адаптерами

Классификация цепей. Для того чтобы обеспечить инвариатность мультиплексора по отношению к разным типам разработанных и будущих адаптеров проведена общая классификация цепей стыков и на основе этой классификации введена обработка обобщенных сигналов, к которым могут быть сведены сигналы любых современных модемов.

Сигналы (или цепи на стыках) можно разделить на:  
данные и синхронизация данных;  
команды управления в модем;  
сигналы исполнения команды модемом;  
внеочередные сигналы состояния из модема;  
сигналы аварийного состояния из модема.

Цепи для данных и синхронизации обсуждены выше.

Команды управления в модем. Выдача команд управления в модем проявляется в изменении состояния одной (чаще всего) или нескольких цепей на специализированном стыке модем-адаптер, т.е. это команды записи в некоторую абстрактную память. Но в целом для управления необходимы еще и команды опроса состояния некоторой абстрактной памяти (регистровых признаков, цепей стыка). Поэтому в универсальный стык сведены:

- 8 цепей передачи байта команды в адаптер;
- 8 цепей приема состояния при опросе адаптера;
- 2 цепи идентификации команды;
- 5 цепей номера канала.

Сигналы исполнения команды модемом могут поступать через различные интервалы ( $10^{-3} \div 10^1$  секунд) времени. Эти сигналы сводятся в один "запрос управления" (ЗУ) независимо от того, какая команда исполнена модемом. Так сделано в расчете на то, что алгоритмы управления модемом -- последовательные, в которых команда следует после исполнения

предыдущей. Предположение подтвердилось в процессе разработки АС-6 при рассмотрении специальных модемов 153387, 5Ц27, всей группы модемов ЕС ЭВМ и модемов, рекомендованных МККТТ (V. 21, V.23, V.24).

Внеочередные сигналы состояния. К этой группе сигналов отнесены те, которые по характеру обработки не нарушают основного обмена данными, и такая обработка представляется как параллельный процесс. Примером могут служить сигналы меток времени, или сигналы результата косвенной оценки качества приема, или данные, поступающие по медленному обратному каналу в модемах V.23. Все эти сигналы сводятся к одному - "особое условие", при обработке которого в мультиплексоре производится выдача сообщения - состояния для запуска параллельной ветви в программе обмена.

#### Сигналы аварийного состояния.

Название группы уже определяет назначение и характер обработки сигналов. Все сигналы сводятся в один - прерывание, по которому прекращается текущий обмен в мультиплексоре (по данному каналу) и организуется запуск программной ветви обработки сигналов аварийного состояния. Состояние уточняется с помощью команд опроса, рассмотренных выше. Примером сигналов рассматриваемой группы является "детектор несущей" стыка С2 и V.24 (цепь I09), означающий, что на входе модема зафиксировано снижение уровня несущей ниже допустимого предела (пропадание канала) и продолжение обмена невозможно.

Подробная классификация цепей "стандартного" стыка С2 и V.24 выполнена при разработке адаптера АТФ2 системы АС-6.

Индивидуальные цепи стыка. Рассмотренные группы сигналов образуют индивидуальную для каждого (симплексного) канала часть универсального стыка:

- КД - код данных (из адаптера);
- СИ - синхронизация данных (из адаптера);
- ДВ - передаваемые данные (в адаптер);
- ПМ - признак прием/передача (в адаптер);
- ЗУ - запрос управления (из адаптера);
- ОУ - особое условие (из адаптера);
- ПР - прерывание (из адаптера).

### Телефонные адаптеры

Телефонные адаптеры в БЭСМ-10 - это устройства, выполняющие преобразование сигналов специализированного стыка модема в сигналы универсального стыка "адаптер-мультиплексор". Для таких адаптеров предложен единый подход к структуре и составу основного оборудования. Этот подход позволяет выполнить устройства-адаптеры адаптируемыми, в свою очередь, к любому модему из некоторого класса модемов. Дело в том, что даже модемы, имеющие специализированный стык, удовлетворяющий требованиям гостированного стыка С2 или, что то же, рекомендации V.24 МККТГ могут обладать разнообразным набором цепей в стыке и различными процедурами управления. Требования

ГОСТ"а и МККТТ ограничивают лишь максимальную номенклатуру цепейстыка, оставляя разработчику свободу определения конкретного набора. Такой подход удобен для разработчиков модемов, но создает трудности разработки адаптера. Отсюда возникает требование к адаптеру, которое было уже выполнено в адаптерах АТФ1 и АТФ2 системы АС-6, заключающееся в том, что аппаратура адаптера должна быть рассчитана на любой набор цепей в пределах номенклатуры стандарта или нормали и адаптироваться к любому модему с помощью программ управления. Выполнение этого требования обеспечит "стабильность" аппаратуры адаптера при переходе от одного типа модема к другому в процессе эксплуатации системы.

Однако гибкость адаптера не должна требовать существенных изменений в программах управления. Естественно, что в таком случае требование гибкости может быть выполнено за счет увеличения объема аппаратуры. Конкретно увеличение касается объема памяти, необходимой в адаптере для каждого канала.

Способ построения адаптеров состоит в том, что в аппаратуре выделяется по назначению два класса памяти — память масок и память сигналов выдачи (рис.72). С помощью памяти масок производится настройка классификатора цепейстыка, который распределяет сигналы с каждой из цепей специализированногостыка в одну из обобщенных цепей универсального стыка. Распределение выполняется с точностью

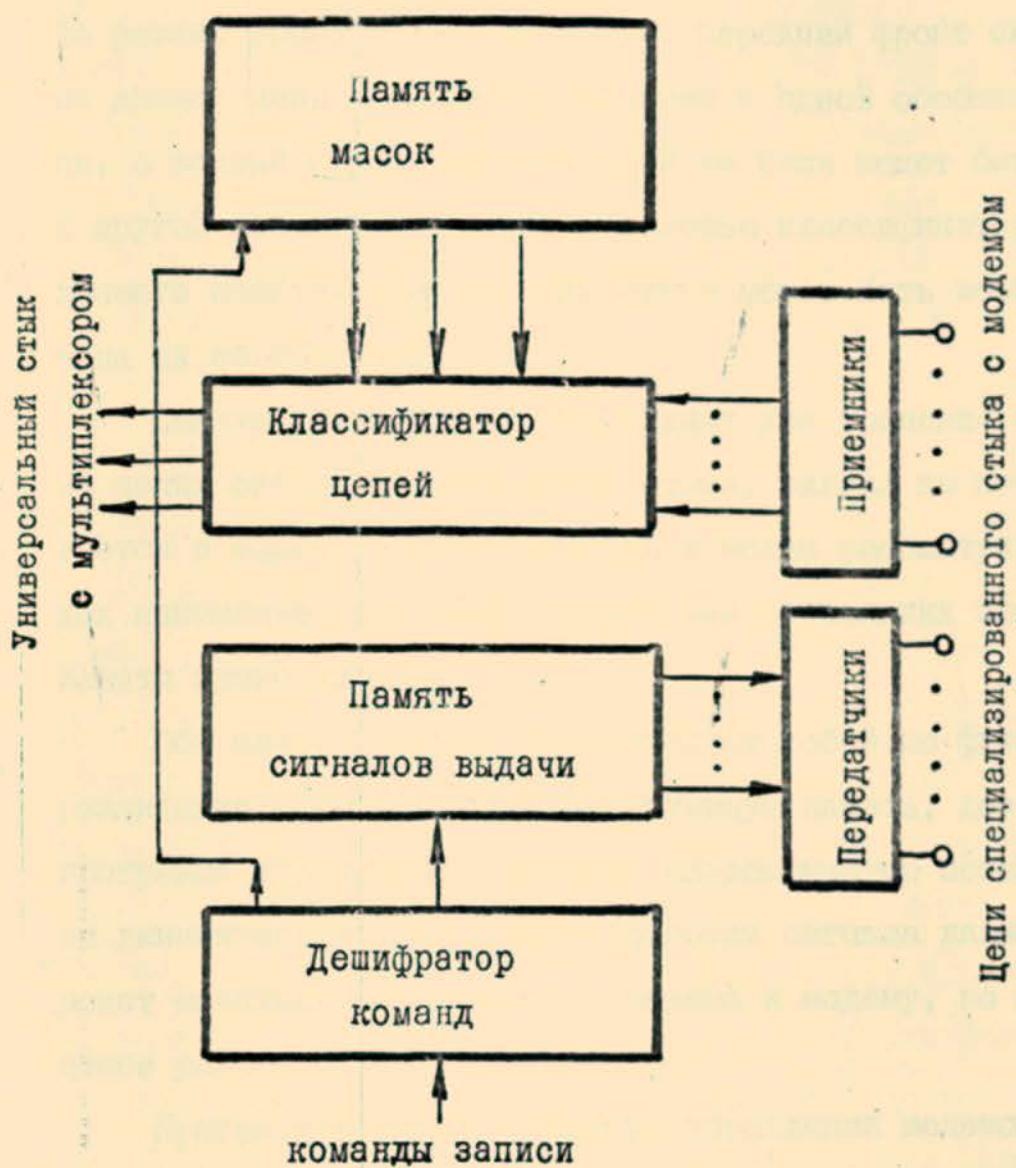


Рис. 72. Блок-схема типового оборудования телефонного адаптера

до разных фронтов сигнала, т.е. передний фронт сигнала на данной цепи может быть отнесен к одной обобщенной цепи, а задний фронт сигнала этой же цепи может быть отнесен к другой обобщенной цепи. С помощью классификатора, управляемого памятью масок, цепь стыка может быть вообще исключена из обработки.

Память сигналов выдачи служит для хранения состояния на цепях специализированного стыка, сигнал по которым выдается в модем. Выдача команды в модем рассматривается как изменение состояния одного или нескольких элементов памяти этого класса.

Оба класса памяти представляют собой по физической реализации регистровую оперативную память, доступную программе управления, поэтому классификатор цепей является динамическим, т.е. вид обработки сигнала данной цепи может меняться не только от модема к модему, но и в процессе работы одного модема.

Программирование процедур управления модемом с помощью такого адаптера заключается в составлении таблиц состояния памяти масок и памяти сигналов выдачи для каждого шага управления. Длина таблиц может быть постоянной и может не зависеть от модема, но содержимое таблицы полностью определяется реализуемой процедурой и модемом. Таблицы не зависят от номера канала и могут быть повторно входными. Собственно программа управления состоит из

команд, задающих выдачу состояний таблиц в память адаптера и, поскольку длина таблиц постоянная, то команды одинаковы для любого модема. Смена состояний памяти адаптера называется кадром и управление сводится к чередованию кадра маски и кадра сигналов выдачи.

Действительно, после пары кадров (масок и сигналов выдачи) адаптер оказывается настроенным на получение реакции модема на определенной цепи после выдачи в этот модем какой-то команды. Благодаря наличию классификатора реакция ожидается по определенной цепи или группе цепей.

Таким образом, телефонные адаптеры БЭСМ-10 так же как и телефонные адаптеры АТФ1 и АТФ2 системы АС-6 по отношению к программам управления являются памятью двух классов и набором универсальных реакций: запрос управления, особое условие, прерывание, причем программа может прогнозировать тип реакции и круг цепей, по которым будет осуществляться реакция модема.

Рассмотренный метод построения адаптеров, проверенный в АС-6 позволил разработать не только адаптер для модемов, удовлетворяющих С2, но и два адаптера для специальных модемов, определенных заданиями заказчиков.

## ЭЛЕМЕНТАЯ И КОНСТРУКТИВНАЯ БАЗА БЭСМ-10

### ВВЕДЕНИЕ

В соответствии с принципами, изложенными в аванпроекте БЭСМ-10, при построении процессоров и высокоскоростных каналов системы будет использована диодно-транзисторная система элементов на основе интегральных диодных КНС-матриц и транзисторных усилительных элементов (ДЛТУ-схемы).

На этапе эскизного проектирования системы БЭСМ-10 была проведена разработка базовых элементов и определены основные микросхемы комплекса.

Как следует из аванпроекта БЭСМ-10, основными структурными единицами системы элементов являются вентиль, содержащий схему совпадения и сборочный диод, и два усилительных элемента - инвертор и переключатель, обеспечивающие усиление и формирование логического сигнала в системе (см.аванпроект вычислительной системы БЭСМ-10, стр. I09+I60).

### ПРОГРАММИРУЕМАЯ ИНТЕГРАЛЬНАЯ ДИОДНАЯ МАТРИЦА

Основным логическим элементом сложных высокопроизводительных узлов процессора типа устройств сложения, мультиплексоров и т.п. будут 2 типа программируемой (настраиваемой) диодной матрицы, разработанной на основе микросхем серии I39.

Электрическая схема программируемой диодной матрицы (ПДМ-1) приведена на рис.72. Микросхема содержит собственно КНС-матрицу и 4 резистора и обеспечивает реализацию до 4 вентилей.

На рис.73 приведена схема матрицы ПДМ-2. Микросхема типа ПДМ-2 рассчитана на использование корпуса с 20 выводами.

Программирование микросхем ПДМ-1 и ПДМ-2 можно проводить непосредственно перед их установкой при изготовлении или ремонте ячеек.

Микросхемы ПДМ-1 и ПДМ-2 имеют следующие характеристики:

$U_o$  - напряжение на р-п переходе при токе 10 мка;

$$U_o = 550 \text{ мв}$$

$\Delta U_o$  - разброс напряжения на р-п переходе при токе 10мка в пределах матрицы;

$$\Delta U_o = \pm 15 \text{ мв}$$

$m\varphi_T$  - для КНС-структуры вдвое больше, чем для обычного р-п перехода;

$$m\varphi_T = 55 \text{ мв}$$

$\Delta m\varphi_T$  - разброс  $m\varphi_T$  в пределах матрицы;

$$\Delta m\varphi_T = \pm 4,0 \text{ мв}$$

$R_\delta$  - активное сопротивление базы р-п перехода;

$$R_\delta = 15 \text{ ом}$$

$R_p$  - активное сопротивление пережигаемой перемычки;

$$R_p = 10 \text{ ом}$$

$R_{sh}$  - активное сопротивление соединительных шин;

$$R_{sh} = 10 \text{ ом}$$

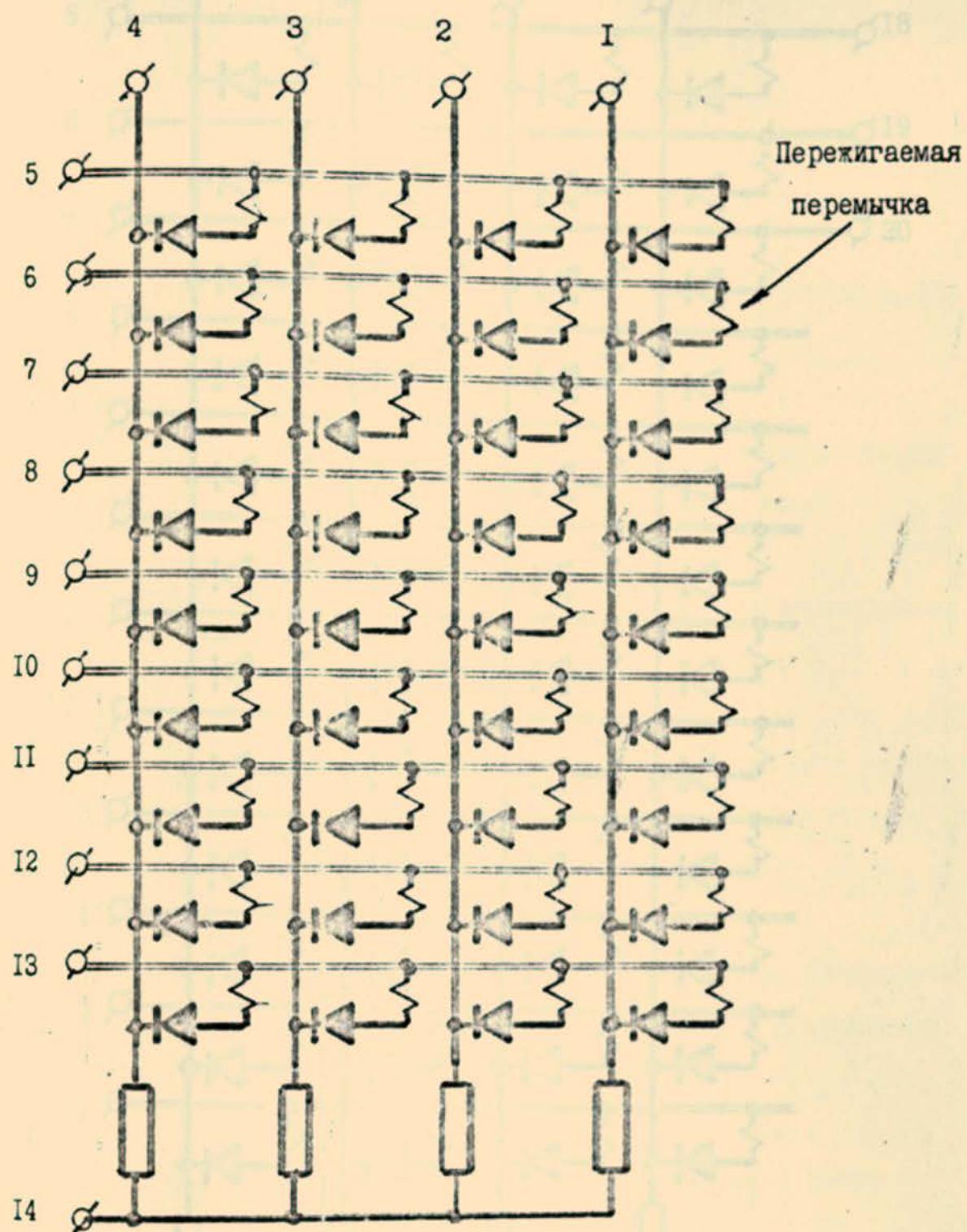


Рис. 72. Программируемая диодная матрица ПДМ-1

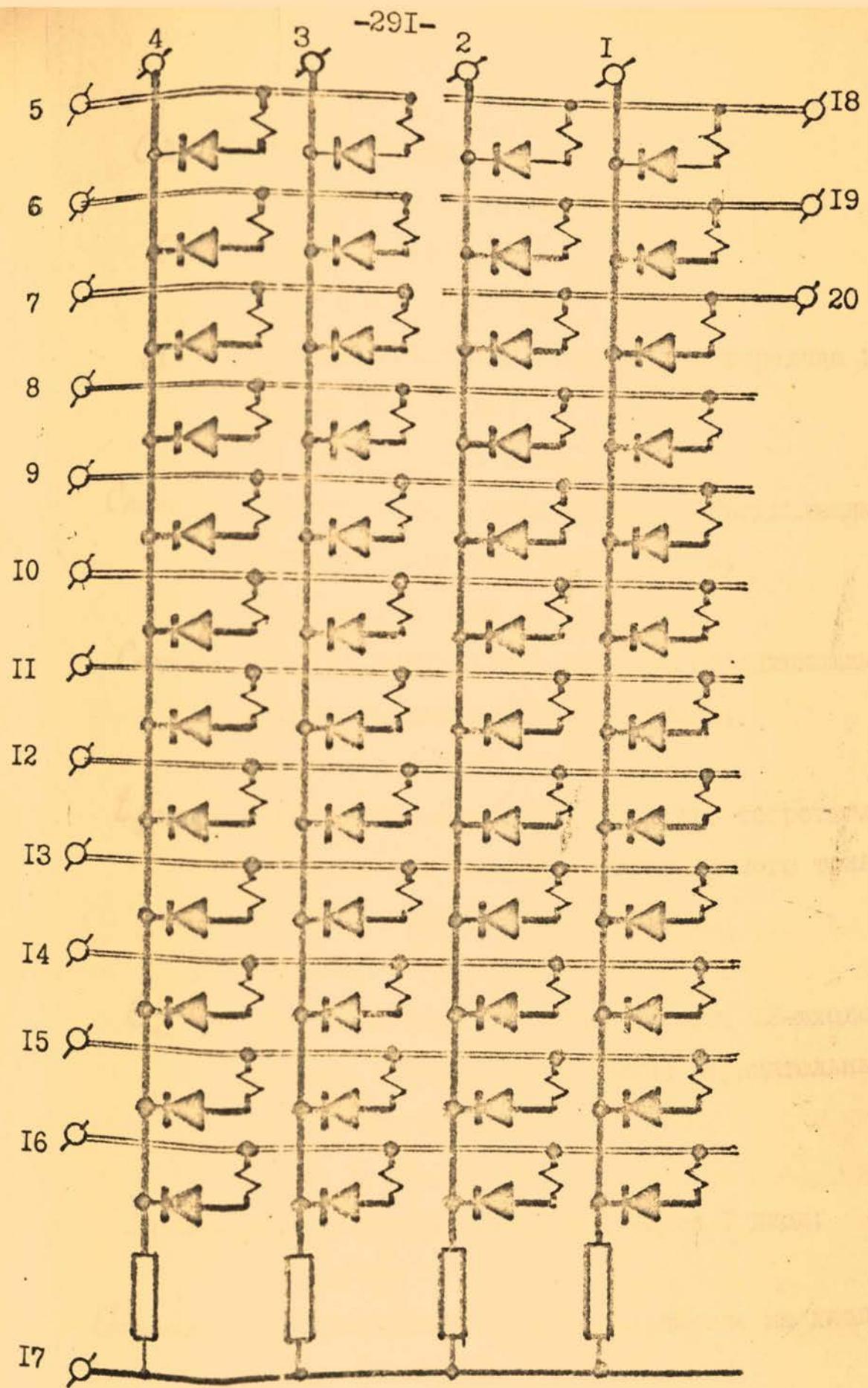


Рис. 73. Программируемая диодная матрица ПДМ-2

$C_d$  - емкость диода;

$$C_d = 0,05 \text{ пФ}$$

$C_p$  - емкость перекрестия;

$$C_p = 0,09 \text{ пФ}$$

$Q$  - заряд переключения одного р-п перехода при прямом токе  $I$  ма;

$$Q \leq 0,5 \text{ пкул.}$$

$C_{par1}$  - емкость между двумя шинами, объединяющими положительные электроды диодов;

$$C_{par1} \leq 0,05 \text{ пФ}$$

$C_{par2}$  - емкость между двумя шинами, объединяющими отрицательные электроды диодов;

$$C_{par2} \leq 0,15 \text{ пФ}$$

$t_{ восст}$  - время восстановления обратного сопротивления перехода при переключении с прямого тока  $I_{ma}$  до уровня обратного тока  $I_{ma}$ ;

$$t_{ восст} \leq 0,5 \text{ нсек}$$

$C_{\Sigma}$  - суммарная переключаемая емкость I2-входового вентиля в точке объединения отрицательных электродов диодов;

$$C_{\Sigma} \leq 1,5 \text{ пФ}$$

$I_{прmax}$  - максимальный прямой ток через  $I$  диод;

$$I_{прmax} = 5 \text{ ма}$$

$U_{обрmax}$  - максимальное обратное напряжение на диоде;

$$U_{обрmax} = 6 \text{ в}$$

$P_{max}$  - максимальная мощность, выделяемая диодно-резистивной матрицей;

$$P_{max} = 50 \text{ мвт}$$

$U_{прог}$  - напряжение пережигания перемычки;

$$U_{прог} = 4,0 \text{ в}$$

$\delta$  - точность изготовления резисторов;

$$\delta = \pm 5\%$$

$\Delta U_{g\max}$  - максимальный технологический разброс напряжения на диодах при прямом токе  $I_{ma}$  (включая разброс между разными матрицами);

$$\Delta U_{g\max} = \pm 60 \text{ мв}$$

### ДИОДНАЯ МАТРИЦА С ПРЕДВАРИТЕЛЬНОЙ НАСТРОЙКОЙ

Значительный объем оборудования в устройствах системы БЭСМ-10 состоит из типовых многократно повторяющихся схем, как правило, логически достаточно простых. Сюда можно отнести регистры, буферные памяти и т.п. В этих случаях для реализации диодной логики целесообразно использовать ненастраиваемые диодные матрицы, точнее, матрицы, настройка которых тем или иным способом осуществляется на заводе-изготовителе микросхем. Использование матриц такого типа позволяет в некоторых случаях в 1,5 раза увеличить плотность упаковки вентилей, так как в этом случае отпадает необходимость использовать выводы корпуса матрицы для ее программирования. При серийном производстве матрицы такого типа будут изготавливаться с помощью соответствующих фототаблонов.

При проектировании устройств с использованием этих матриц предполагается ориентироваться на кристаллы, размеры которых (т.е. число строк и столбцов) совпадают с размерами описанных выше программируемых матриц. Это позволит на этапе изготовления и отладки опытного образца системы БЭСМ-10 широко использовать кристаллы матриц ПДМ-типа и, таким образом, быстро изготавливать новые типы приборов, не ожидая изготовления соответствующих новых фотоматриц и кристаллов. Настройка кристаллов ПДМ-матриц в этом случае будет осуществляться с помощью вспомогательных щупов перед герметизацией корпуса микросхемы.

Параметры ненастраиваемых матриц аналогичны параметрам программируемых матриц. Эти приборы представляют собой развитие микросхем серии I39.

На рис. 74 приведен пример ненастраиваемой диодной матрицы, обеспечивающей построение 4 разрядов буферного регистра, и логическая функция, которую осуществляется.

#### ЭЛЕКТРИЧЕСКАЯ СХЕМА ОСНОВНОГО УСИЛИТЕЛЬНОГО ЭЛЕМЕНТА

Функции усиления и формирования логического сигнала осуществляются в ДЛТУ-схемах транзисторным усилителем. Электрическая схема базового усилительного элемента разработана применительно к использованию серийно освоенной гибридной технологии типа "Посол" с использованием в качестве основного активного элемента бескорпусного транзистора

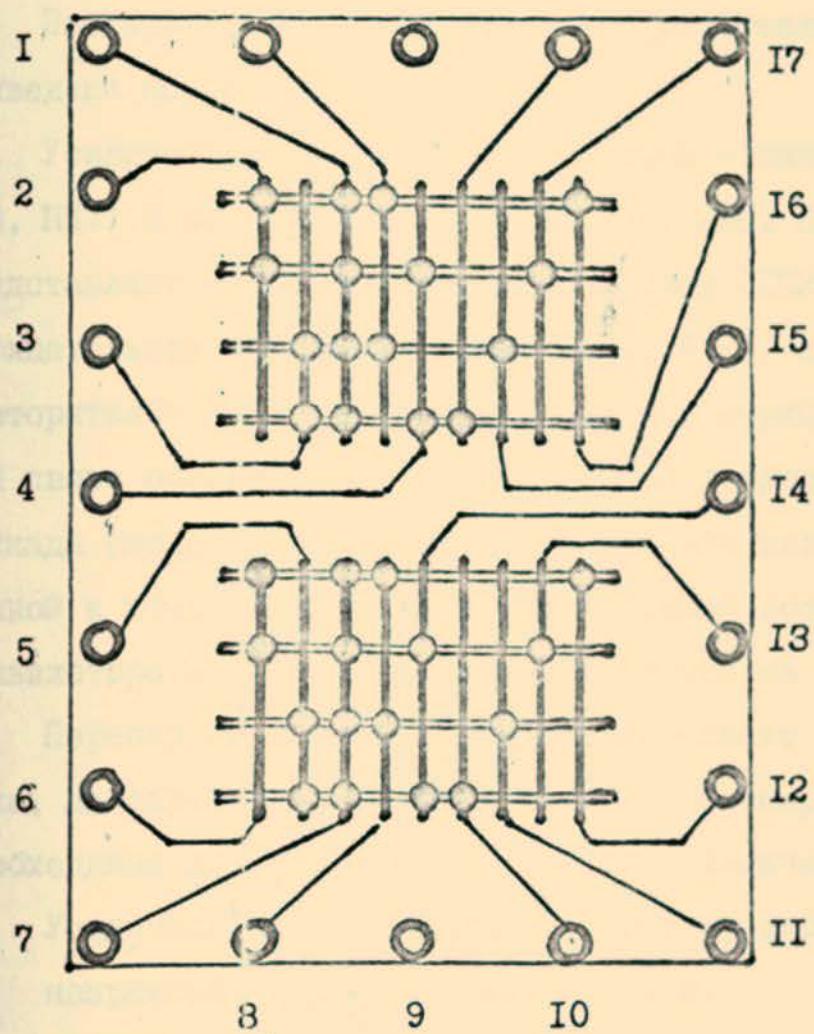
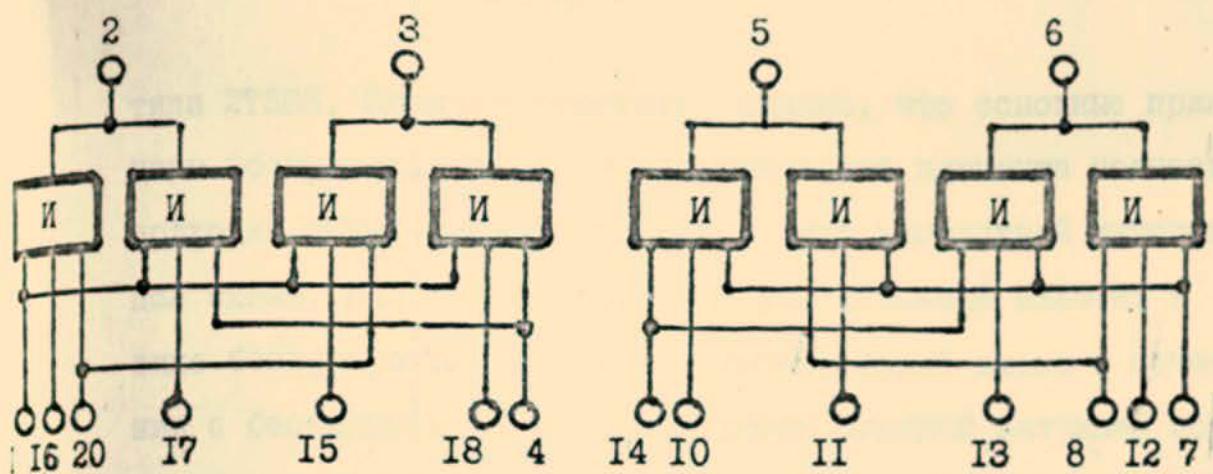


Рис. 74. Диодная матрица с предварительной настройкой.

типа 2Т366. Следует отметить, однако, что основные принципы построения базового усилительного элемента полностью подходят и для реализации его в виде монолитной интегральной схемы. Более того, базовый усилительный элемент в виде бескорпусной монолитной интегральной схемы в сочетании с бескорпусной программируемой диодной матрицей позволит в дальнейшем перейти к так называемой многокристальной конструкции и наиболее полно использовать все преимущества ДЛТУ-схем.

Электрическая схема основного усилительного элемента приведена на рис. 75 .

Усилитель состоит из двух частей - инвертора (ПП1, ПП2, ПП3) и переключателя (ПП4, ПП5, ПП6, ПП7). Инвертор представляет собой усилительный каскад (ПП2), охваченный отрицательной обратной связью (ПП1, R2), с эмиттерным повторителем (ПП3) на выходе. Наличие отрицательной обратной связи обеспечивает весьма низкое входное сопротивление каскада (менее 10 ом). При переключении инвертора подключенной к нему диодной логической схемой потенциал базы транзистора ПП2 изменяется не более чем на 10-15 мв.

Перепад на выходе инвертора управляет переключателем тока, на парафазных выходах которого формируются сигналы, необходимые для переключения диодных логических схем.

Усилитель имеет следующие основные параметры:  
напряжение питания - минус 5,2 в;  
напряжение управления - пороговым уровнем - минус 2,2 в;

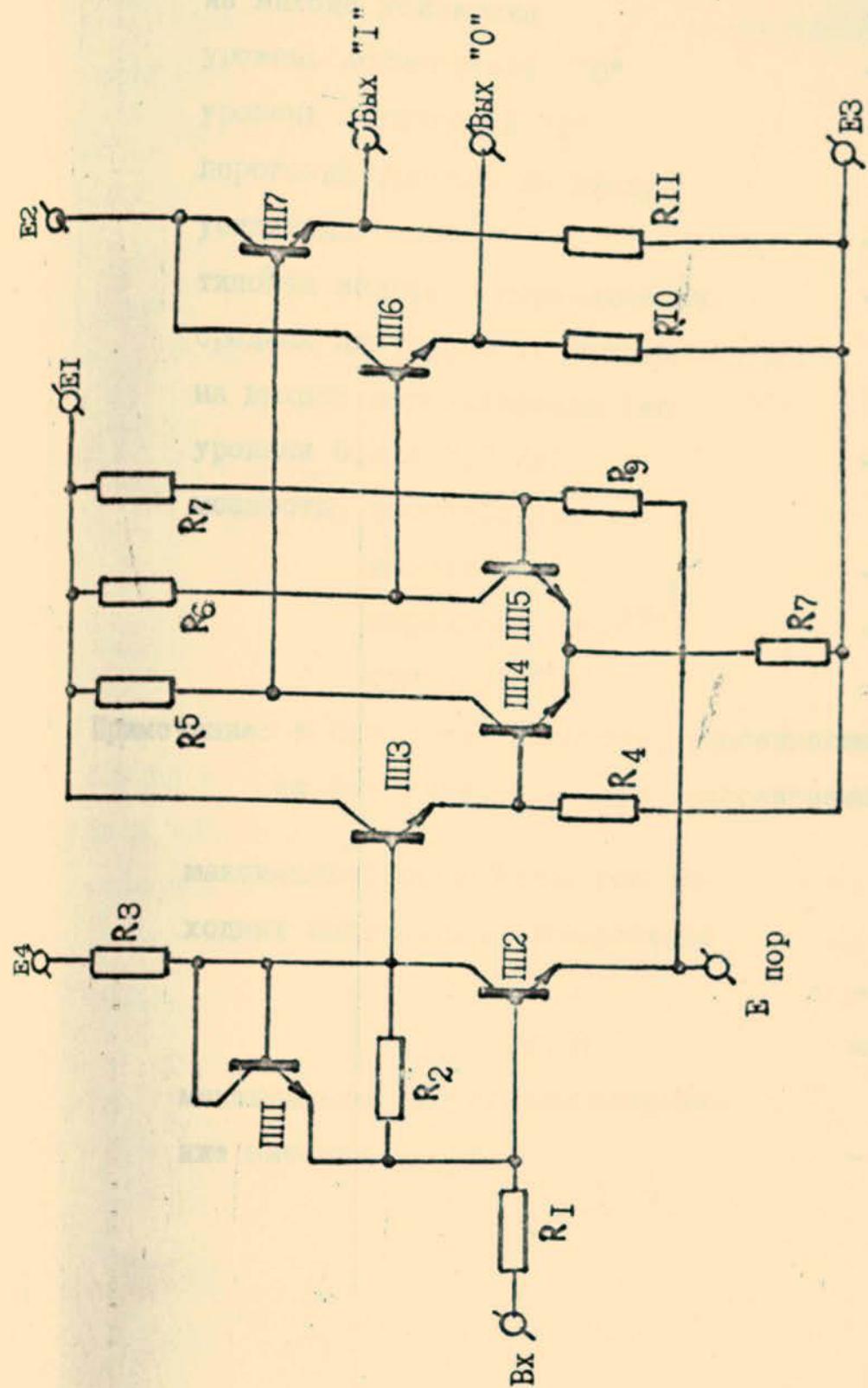


Рис. 75. Электрическая схема основного усиительного элемента

номинальный логический перепад	
на выходе усилителя	- 1300 мв;
уровень логического "0"	- 860 мв;
уровень логической "1"	- 2160 мв;
пороговый уровень по входу	
усилителя	- 1510 мв;
типовая задержка переключения	- 2,6 нсек;
средняя длительность фронта	
на выходе переключателя (по	
уровням 0,2 + 0,8 Ил)	- 2,0 нсек;
мощность, рассеиваемая	
инвертором *	- 12 мвт;
переключателем **	- 30 мвт;
суммарная ***	- 50 мвт;

Примечание: \* без учета мощности, рассеиваемой  $R_4$

\*\* без учета мощности, рассеиваемой  $R_{10}$  и  $R_{11}$ .

максимально допустимый ток вы-  
ходных эмиттерных повторителей

тип Б	- 20 ма,
тип В	- 45 ма,

максимально допустимое напряже-  
ние питания

- 10 в.

Функциональные возможности усилительного элемента  
следующие:

1. Количество объединений вентилей на  
входе усилителя (инвертора) - до 8
2. Количество объединений по выходу  
инвертора - до 3
3. Количество переключателей, подклю-  
чаемых к выходу инвертора - до 12
4. Максимальное количество вентилей  
на выходе усилителя:

подключенных	- 20
переключаемых	- 6

В ходе эскизного проектирования ДЛТУ-схем была разра-  
ботана в гибридном варианте микросхема усилительного элемен-  
та и изготовлена партия микросхем (600 шт.). Анализ харак-  
теристик полученных образцов при окружающей температуре  
 $+25^{\circ}\text{C}$ , номинальных питающих напряжениях и нагрузке переклю-  
чателя 100 ом дал следующие основные результаты:

$\Delta U_n^0$  - разброс уровня логического "0" переключате-  
ля;

$$\Delta U_n^0 = 150 \text{ мв}$$

$\Delta U_{\text{л}}^{\prime}$  - разброс уровня логической "1" переключателя;

$$\Delta U_{\text{л}}^{\prime} = 250 \text{ мв}$$

$\Delta U_{\text{лmin}}$  - минимальный логический перепад переключателя;

$$\Delta U_{\text{лmin}} = U_{\text{лmin}}^{\circ} - U_{\text{лmax}}^{\prime} = 1100 \text{ мв}$$

$\Delta U_{\text{лmax}}$  - максимальный логический перепад переключателя;

$$\Delta U_{\text{лmax}} = U_{\text{лmax}}^{\circ} - U_{\text{лmin}}^{\prime} = 1500 \text{ мв}$$

Зависимость логических уровней от окружающей температуры отражается следующими температурными коэффициентами:

Для переключателя

$$\frac{\Delta U_{\text{лmax}}^{\circ}}{\Delta T} = +1,3 \text{ мв/}^{\circ}\text{C}$$

$$\frac{\Delta U_{\text{лmin}}^{\prime}}{\Delta T} = +0,6 \text{ мв/}^{\circ}\text{C}$$

$$\frac{\Delta U_{\text{л}}^{\circ}}{\Delta T} = -\frac{\Delta U_{\text{л}}^{\prime}}{\Delta T} \approx 0,10 \text{ мв/}^{\circ}\text{C}$$

для инвертора\*

$$\frac{\Delta U_{\text{быхи}}^{\circ}}{\Delta T} = -0,6 \text{ мв/}^{\circ}\text{C}$$

$$\frac{\Delta U_{\text{быхи}}^{\prime}}{\Delta T} = -2,0 \text{ мв/}^{\circ}\text{C}$$

$$\frac{\Delta U_{\text{быхи}}^{\circ}}{\Delta T} = \frac{\Delta U_{\text{быхи}}^{\prime}}{\Delta T} = -1 \text{ мв/}^{\circ}\text{C}$$

Примечание: \* для выхода инвертора уровню логического "0" соответствует нижний уровень, а уровню логической "1" - верхний.

## УПРАВЛЯЮЩИЕ ЦЕПИ В ДЛТУ-СХЕМАХ

Основные принципы управления большим числом разнесенных нагрузок типа цепей управления многоразрядными регистрами были описаны в аванпроекте БЭСМ-10.

В ДЛТУ-схемах основным элементом, обеспечивающим передачу сигнала по согласованной линии связи является инвертор. Электрическая схема инвертора практически тождественная соответствующей части основного усилительного элемента (рис.76).

Инвертор имеет следующие основные характеристики:

напряжение питания	- минус 5,2 в;
напряжение управления	
пороговым уровнем	- минус 2,2 в;
номинальный логический	
перепад на выходе	- 740 мв;
уровень логического "0"	- 2200 мв;
уровень логической "1"	- 1460 мв;
пороговый уровень по	
входу инвертора	- 1510 мв;
типовая задержка переклю-	
чения	- 1,3 нсек;
средняя длительность фрон-	
та на выходе переключателя	
(по уровням 0,2± 0,8 Ил)	- 1,5 нсек.

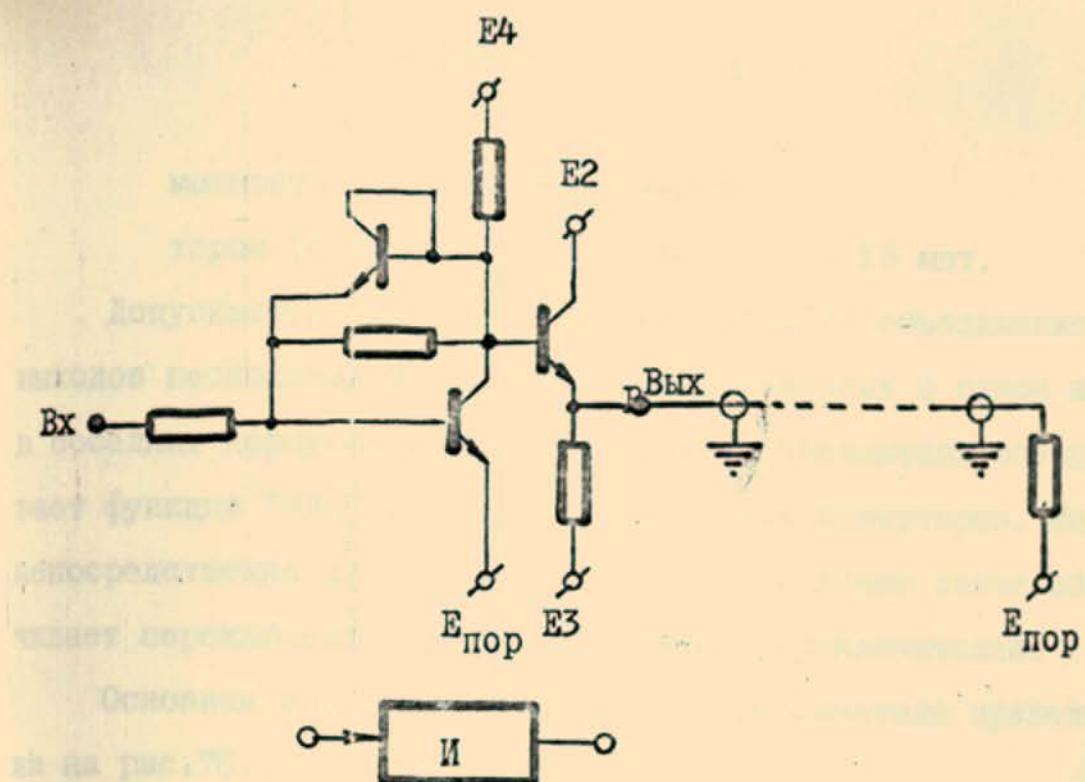


Рис. 76. Электрическая схема инвертора и ее условное изображение

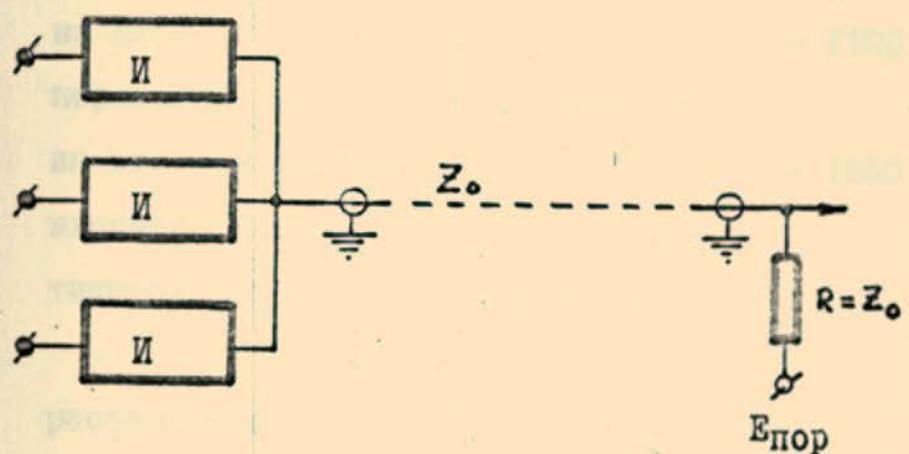


Рис. 77. Объединение инверторов по выходу

мощность, рассеиваемая инвертором (с открытым выходом) - 15 мвт.

Допускается непосредственное (рис.77) объединение выходов нескольких инверторов, расположенных в одном или в соседних корпусах микросхем. Такое объединение обеспечивает функцию "ИЛИ" для входных сигналов инверторов. Инвертор непосредственно или через согласованную линию связи обеспечивает переключение до 12 парафазных переключателей.

Основная электрическая схема переключателя приведена на рис.78.

Характеристики переключателя идентичны характеристикам соответствующей части основного усилительного элемента.

пороговый уровень логического "0"

на входе переключателя - 1900 мв;

пороговый уровень логической "1"

на входе переключателя - 1560 мв;

входная емкость  $C_{bx}$ . З пФ:

типовая задержка переключения

$$t_3 = 1,3 \text{ нсек};$$

рассеиваемая мощность (с открытыми выходами) - 30 мвт;

К одному выходу инвертора (к одной согласованной линии связи) может быть подключено до 12 переключателей (рис.79).

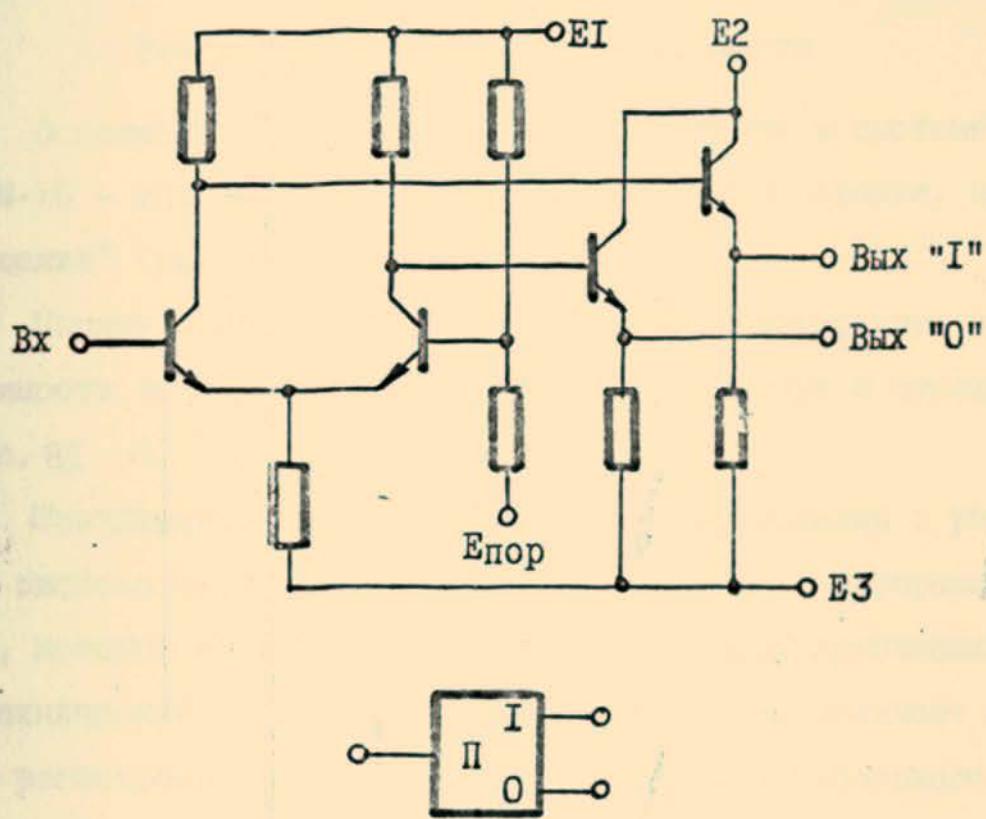


Рис. 78. Электрическая схема переключателя и его условное изображение

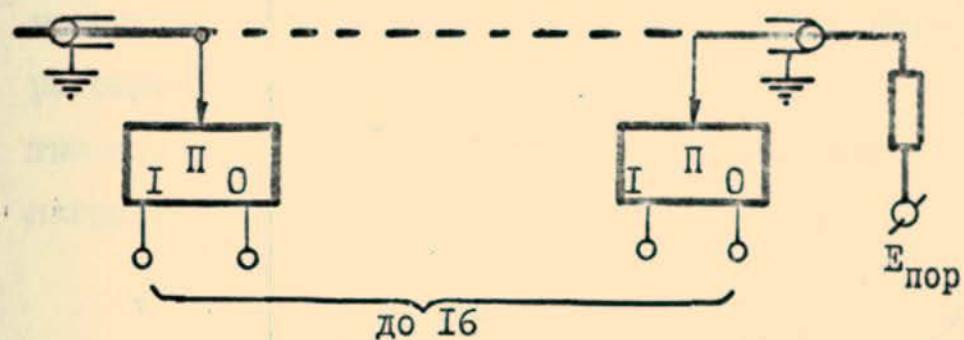


Рис. 79. Объединение переключателей по выходу

### Регистровые запоминающие элементы

Основная схема запоминающего элемента в системе БЭСМ-10 — усилитель с управляемой обратной связью, или "зашелка" (рис. 80).

Наряду с этой схемой могут использоваться при необходимости ее модификации на основе инвертора и переключателя (рис. 81).

Значительный процент (до 50%) оборудования в устройствах системы составляют регулярные цепи типа буферных регистров, которые могут иметь довольно скромные логические и схемотехнические возможности. Этим требованиям отвечает специальный регистровый усилительный элемент типа кабельного усилителя, схема которого приведена на рис. 82. Схема представляет собой аналог соответствующего элемента БЭСМ-6 и полностью совместима с основными элементами ДЛТУ-схем. В сочетании с диодными матрицами она будет использоваться для построения запоминающих элементов типа "зашелка". Основные параметры регистрового запоминающего элемента:

время задержки переключения — 4 нсек;

рассеиваемая мощность (с открытым выходом)

— 4 мвт;

нагрузочная способность:

подключенных — 8

срабатывающих — 3

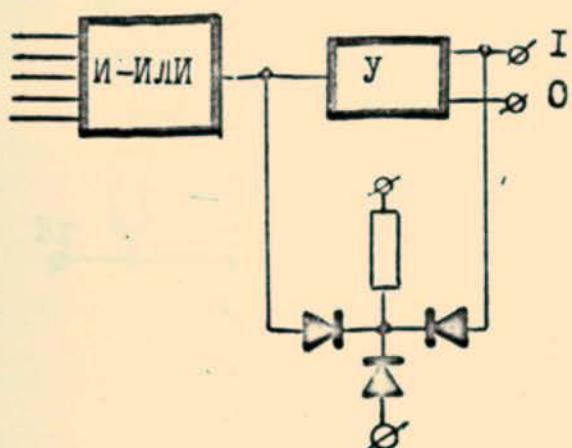


Рис. 80. Защелка на основе стандартного усилителя

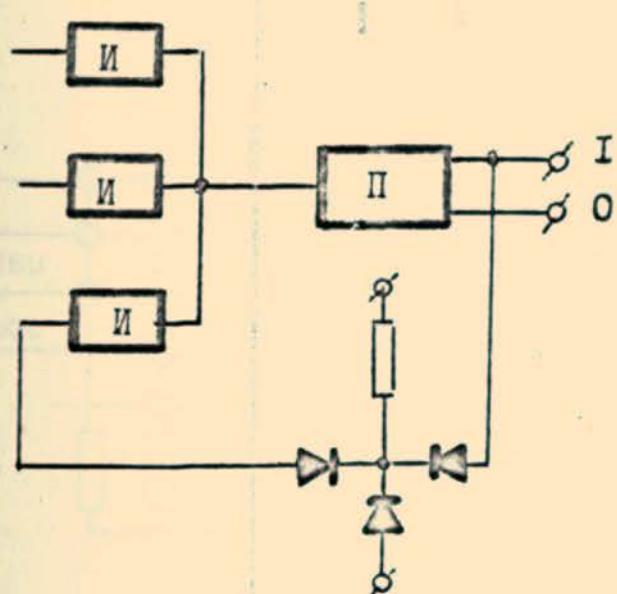


Рис. 81. Защелка с объединением инверторов

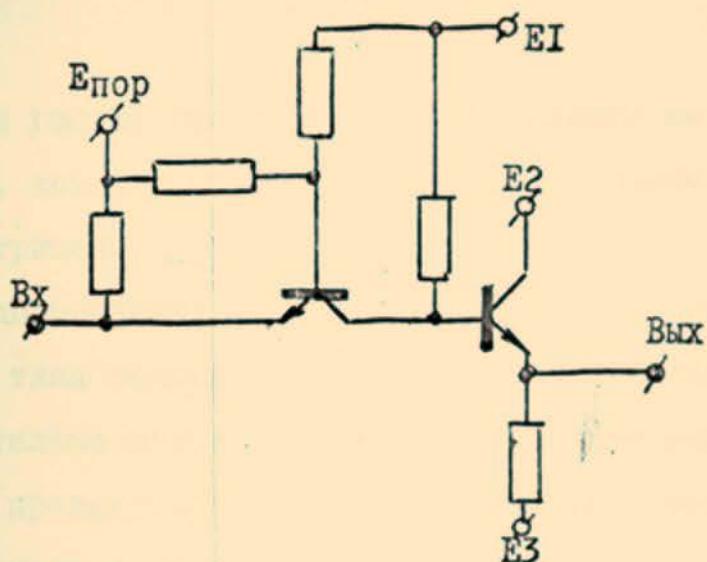


Рис. 82. Регистровый усилительный элемент  
типа кабельного усилителя

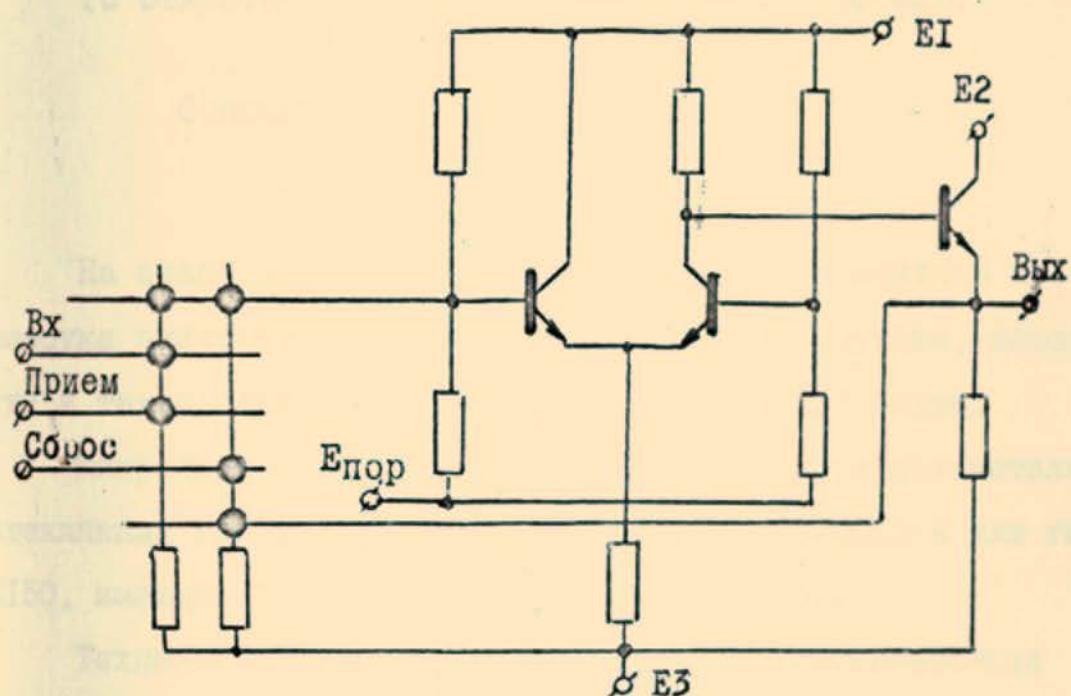


Рис. 83. Запоминающий регистровый элемент  
(диодная матрица показана условно, точка  
соответствует наличию диода в пересечении шин)

На рис.83 приведена схема защелки на основе переключателя, которая будет реализована в одном корпусе с диодной матрицей.

Запоминающий регистровый элемент - усилитель (УЗР) такого типа полностью совместим с ДЛТУ-схемами, выделяет относительно малую мощность и обладает повышенными относительно предыдущей схемы нагрузочными возможностями.

Основные характеристики схемы:

время задержки переключения - 6 нсек;

рассеиваемая мощность с учетом  
мощности, выделяемой вентилями

(с открытым выходом) - 30 мвт.

#### Основные микросхемы серии

#### ДЛТУ-схем

На этапе эскизного проектирования проводилась проработка электрических характеристик, конструкции, топологии и технологии изготовления микросхем ДЛТУ-схем.

Микросхемы будут изготавливаться в серийных металло-стеклянных гостированных корпусах типа 252МС15-2 или типа К150, имеющих 14 или 20 сигнальных выводов.

Технология изготовления микросхем тонкопленочная гибридная, аналогична технологии изготовления микросхем серии "Посол".

В серийном производстве будет использована технологическая линия "Титан", в настоящее время предназначенная для выпуска микросхем серии "Посол".

На этапе опытно-конструкторской разработки серии ДЛТУ-схем присвоен шифр "Канал".

1. "Канал"-1 - два основных усилительных элемента (рис. 84);
2. "Канал"-2 - три инвертора (рис. 86);
3. "Канал"-3 - три переключателя (рис. 85);
4. "Канал"-4 - четыре регистровых усилительных элемента (рис. 87);
5. "Канал"-5 - четыре регистровых запоминающих элемента (рис. 88);
6. "Канал"-6 - диодная матрица ПДМ-1 (рис. 89);
7. "Канал"-7 - диодная матрица ПДМ-2 (рис. 90);
8. "Канал"-8 - диодная матрица с двумя кристаллами типа ПДМ-1 (рис. 91).

#### ОСНОВНЫЕ ТИПЫ СВЯЗЕЙ МЕЖДУ ЭЛЕМЕНТАМИ

В ДЛТУ-схемах используются три типа связей между элементами:

связь между выходом вентиля и входом инвертора,  
или связь ВИ-типа;

связь между выходом инвертора и входом переключателя,  
или связь ИП-типа;

связь между выходом переключателя и входом вентиля,  
или связь ПВ-типа.

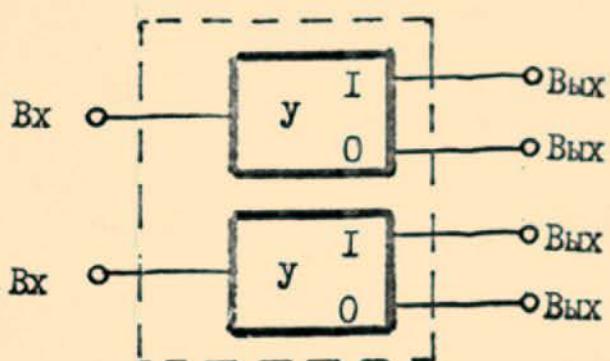


Рис. 84. Микросхема типа "Канал"-I

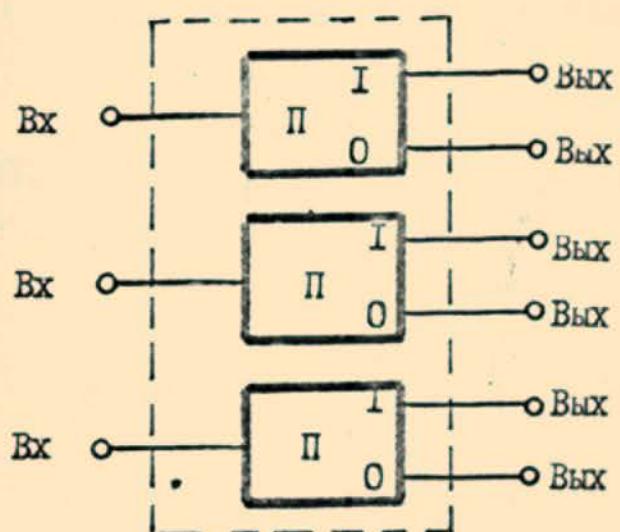


Рис. 85. Микросхема типа "Канал"-3

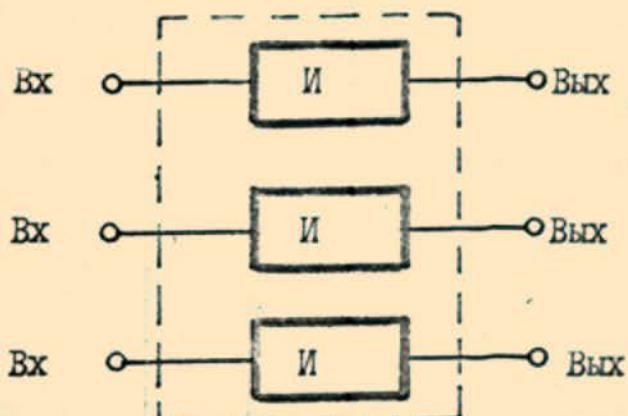


Рис. 86. Микросхема типа "Канал"-2

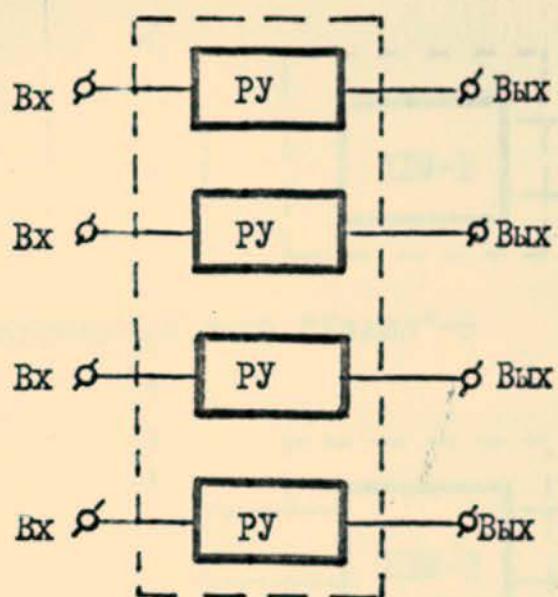


Рис.87. Микросхема типа "Канал"-4

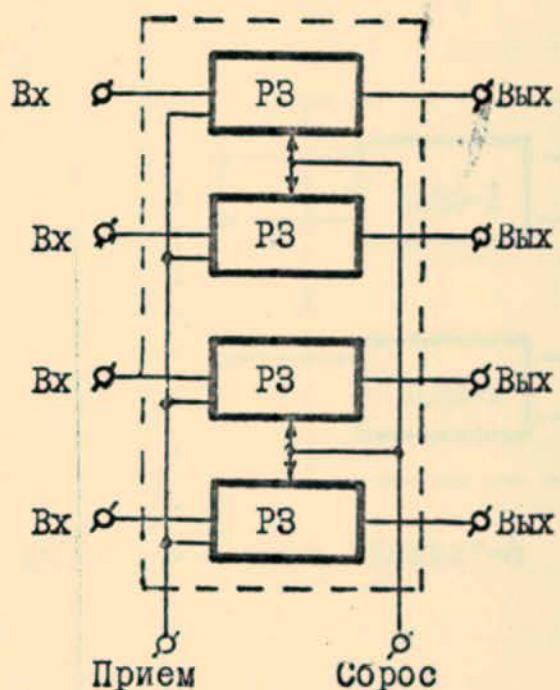


Рис.88. Микросхема типа "Канал"-5

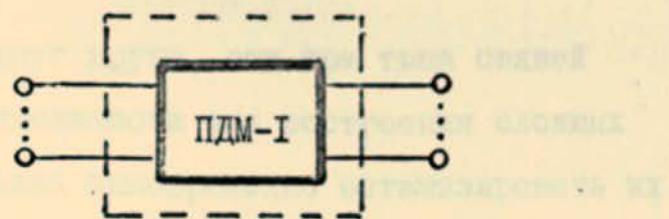


Рис. 89. Микросхема типа "Канал"-6

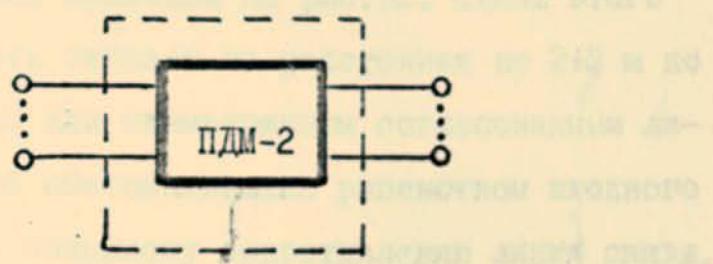


Рис. 90. Микросхема типа "Канал"-7

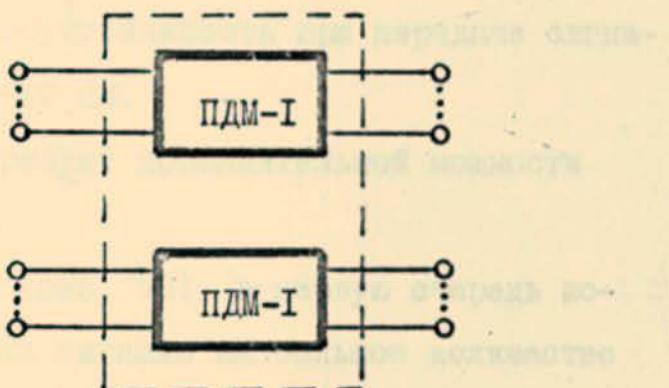


Рис. 91. Микросхема типа "Канал"-8

Взаимно дополняя друг друга, эти три типа связей обеспечивают широкие возможности для построения сложных логических узлов, позволяя одновременно оптимизировать их конструкцию и минимизировать мощность, затрачиваемую на передачу сигналов.

Пример связи ВИ-типа приведен на рис.92. Связи этого типа позволяют передавать сигналы на расстояния до 2-3 м по коаксиальным, полосковым или симметричным согласованным линиям. Согласование линий обеспечивается равенством входного сопротивления инвертора волновому сопротивлению линии связи.

Специфический режим работы соединений ВИ-типа, состоящий в том, что источник сигнала представляет собой источник тока, приемник согласован с линией связи, обеспечивает их повышенную помехоустойчивость при передаче сигналов на расстояние до 15-20 см.

Связи ВИ-типа не требуют дополнительной мощности для передачи сигналов.

Соединения ИП-типа (рис. 93) в первую очередь используются для трансляции сигнала на большое количество разнесенных нагрузок. Требования к связям этого типа аналогичны требованиям к связям в EC<sub>L</sub>-схемах.

Соединения ПВ-типа рассчитаны в основном на передачу сигналов на расстояние не более 15-20 см. Допустимая длина связи зависит от количества и характера нагрузки (т.е. от

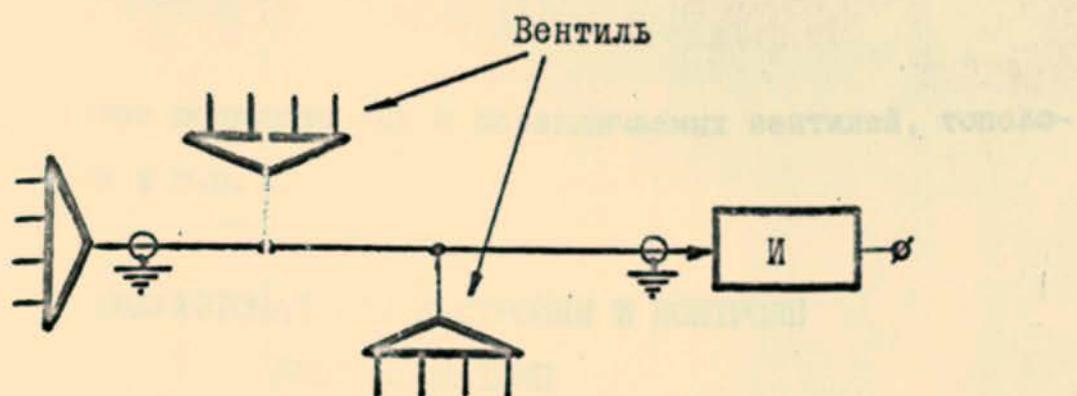


Рис. 92. Связь ВИ-типа

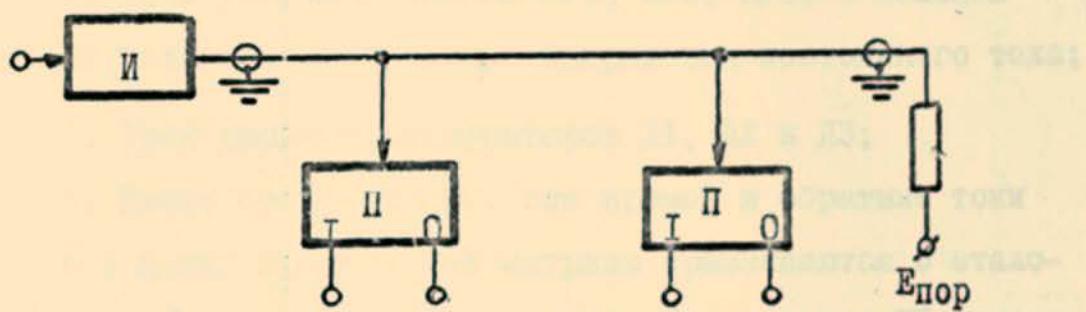


Рис. 93. Связь ИП-типа

количества подключенных и переключаемых вентилей, топологии цепи и т.п.).

### ПОЛУАВТОМАТ ДЛЯ НАСТРОЙКИ И КОНТРОЛЯ ДИОДНЫХ МАТРИЦ

Полуавтомат предназначается для настройки диодных матриц и последующего сравнения с эталонами оставшихся диодов по прямому и обратному току.

В качестве примера рассматривается матрица с количеством диодов  $4 \times 15$ .

Блок-схема (см. рис. 94) состоит из следующих частей:

- I. Разъема (контактного приспособления) для диодной матрицы;
2. Трех релейных блоков БР1, БР2, БР3, с помощью которых подаются необходимые напряжения постоянного тока;
3. Трех диодных дешифраторов Д1, Д2 и Д3;
4. Блока сравнения БС, где прямой и обратные токи диодов в цепях проверяемой матрицы сравниваются с эталонами. В случае несовпадения вырабатывается для БУ сигнал "стоп";

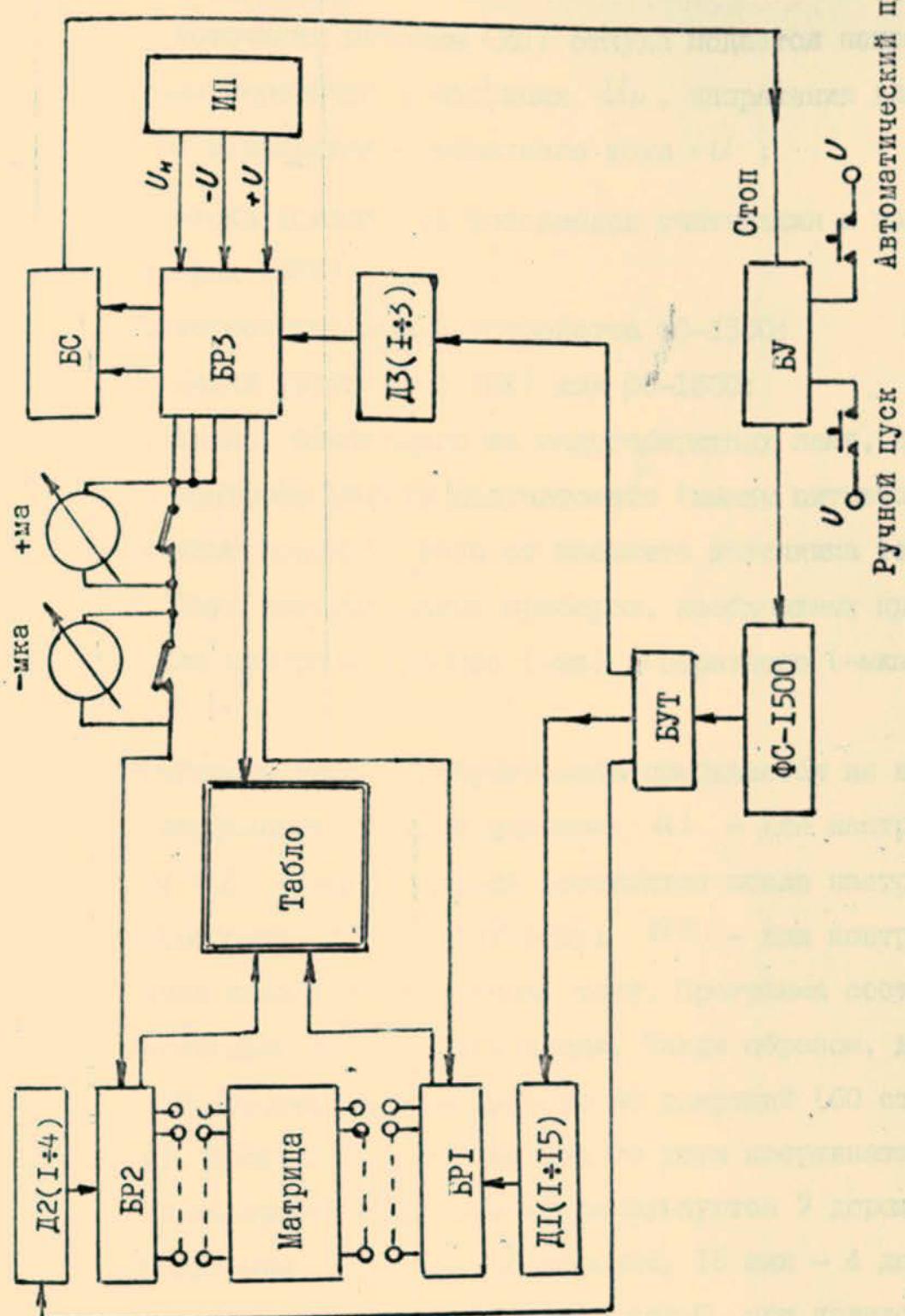


Рис. 94. БЛОК-СХЕМА ПОЛУАВТОМАТА ДЛЯ ИНСТРОКИ И КОНТРОЛЯ ДИОДНЫХ МАТРИЦ

Ручной пуск  
Автоматический пуск

5. Источника питания (ИП) откуда подается напряжение разрушения перемычек в матрицах  $U_H$ , напряжение для прямого тока  $+U$  и напряжение обратного тока  $-U$ ;
6. Блока усилителей фотодиодов считывания и памяти на триггерах (БУТ);
7. Фотосчитывающего устройства ФС-1500;
8. Блока управления (БУ) для ФС-1500;
9. Табло, состоящего из малогабаритных ламп, для визуального контроля работы полуавтомата (лампы питаются через параллельные контакты реле от внешнего источника тока);
10. Двух измерительных приборов, необходимых при ручной работе для контроля прямого (+ма) и обратного (-мка) токов диодов.

Программа работы полуавтомата пробивается на восьмидорожечной перфоленте по трем адресам:  $a_1$  - для настройки матрицы;  $a_2$  - для контроля оставшихся после настройки диодов (рабочих) по прямому току;  $a_3$  - для контроля тех же рабочих диодов по обратному току. Программа составляется последовательно для каждого диода. Таким образом, для матрицы 4x15 необходимо максимально 60 операций (60 строк перфоленты). Программа составляется по двум координатам матрицы.

Для кодирования координат используются 7 дорожек - для направления из 4 шин - 3 дорожки, 15 шин - 4 дорожки. Восьмая дорожка при этом содержит код 0, при кодировании рода работы на этой дорожке - код I, код рода работы  $a_1$ ,  $a_2, a_3$  пробивается на двух других дорожках.

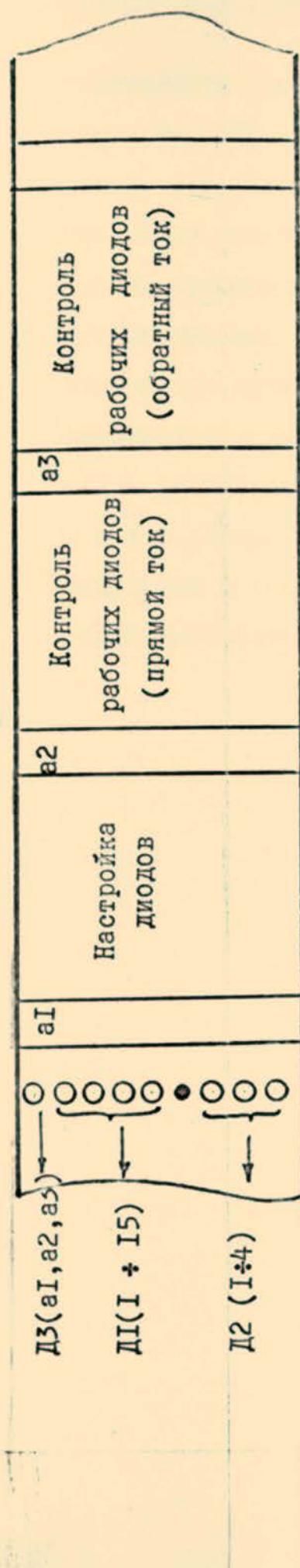


Рис. 95. Перфолента для полуавтомата настройки и контроля диодных матриц

Составление программы предполагается производить с использованием ЭВМ, при этом пробивка перфоленты будет выполняться с помощью ленточного перфоратора. Схема в промежутках между пагами перфоленты возвращается в "0".

Контроль за правильностью настройки идет одновременно с разрушением диодов. Скорость работы полуавтомата в режиме автоматического пуска отрабатывается экспериментально и устанавливается в БУ специальным генератором. Она может меняться в зависимости от частоты смены программ, качества матриц и т.п. Если считать, что частота тактовых импульсов 5-10 импульсов в сек., то при 180 операциях на матрицу может быть настроено и проверено 1+ 2 матрицы в минуту.

## СИСТЕМА ЭЛЕКТРОПИТАНИЯ

В процессе подготовки эскизного проекта проработке подверглись следующие вопросы:

- а) Повышение помехозащищенности и обеспечение бесперебойности подачи энергии систем электропитания;
- б) Использование ЭВМ как органа управления для программного контроля и регулирования вторичных источников питания (ВИП);
- в) Создание ВИП с хорошими электрическими характеристиками, способы распределения энергии и заземления устройств машины.

Основным способом решения первого вопроса является развязка ВИП от силовой сети, что достигается использованием фильтров высоких частот, электромашинных или статических преобразователей частоты. Входные фильтры хорошо защищают ЭВМ от высокочастотных помех силовой сети и поэтому часто устанавливаются в системах электропитания. Однако для такой вычислительной системы как БЭСМ-10 необходимо не только подавлять высокочастотные помехи, но и обеспечить непрерывность работы при авариях силовой сети.

С этой точки зрения наиболее перспективной системой электропитания является система с использованием статических стабилизированных преобразователей частоты.

В таблице 2 приведены сравнительные данные механических преобразователей типа ПСЧ и статических преобразователей типа СИГС. Из таблицы видно, что система электропитания с использованием статических преобразователей значительно надежней, экономичней и имеет лучшие динамические характеристики. Кроме того, такая система не требует генераторного зала, специальной коммутационной и защитной аппаратуры, а также отдельного обслуживающего персонала.

Среднее время между отказами для мотор-генераторов около 1000 часов. Для статического преобразователя эта цифра равна 5000 часам. Для больших вычислительных систем надежность системы электропитания должна быть значительно выше. Повышение надежности приходится осуществлять с помощью резервирования аппаратуры. На рис. 96 приведена система бесперебойного питания с двумя статическими преобразователями (один резервный), питаемыми от разных силовых сетей. В случае аварии одной из сетей ЭВМ питается от оставшейся. Надежность системы с двойным резервированием при надежности одного преобразователя, равной 0,95 на 100 часов работы, равна 0,9975. Эта цифра надежности системы высока, однако такую систему не всегда возможно осуществить. В том случае, если отсутствует вторая сеть и невозможно двойное резервирование, можно применить другую систему. Такая система может состоять, в зависимости от мощности инверторов, из трех (один избыточен) или из четырех (один избыточен). В первом случае, при надежности одного преобразователя 0,95 на 100 ча-

Таблица 2

Параметр Тип	КПД (%)	Предпола- гаемая наработка на отказ (час)	Срок служ- бы (час)	Непрерыв- ность ра- боты без ремонта (час)	Необхо- димость дополни- тельного оборудо- вания	Необходи- мость спец.пер- сонала
ПСЧ-15	70	1000	5000	3000	Коммути- ционные и защит- ные	Да
СПСЧ-20	86	5000	10000	5000	Нет	Нет

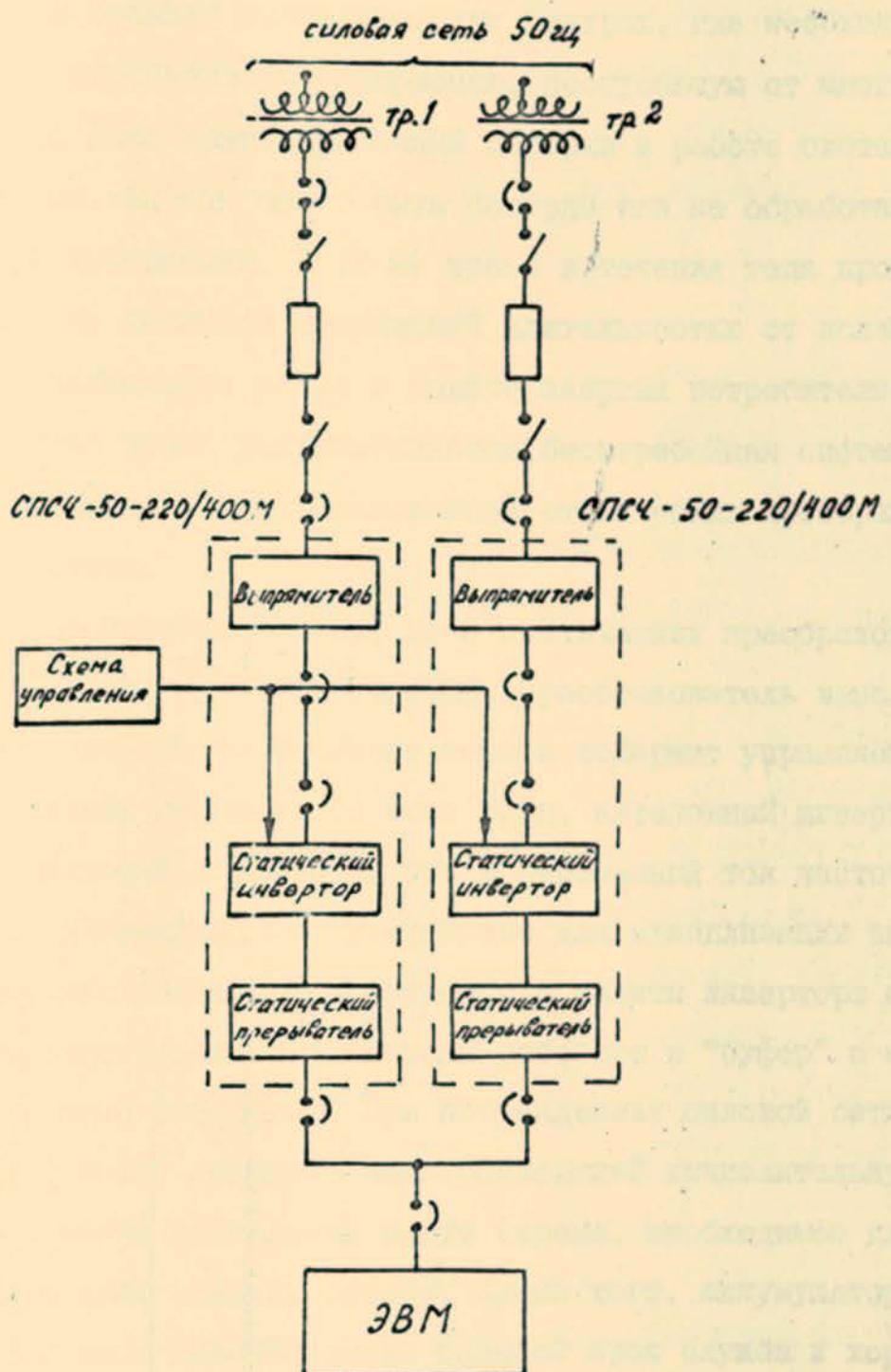


Рис. 96.

сов, надежность системы будет равна 0,99, а во втором случае - 0,986.

Вычислительная система БЭСМ-10 предназначена для установки в больших вычислительных центрах, где необходимо непрерывно обрабатывать информацию, поступающую от многих абонентов. Даже кратковременный перерыв в работе системы недопустим, так как может быть потерян или не обработан важный массив информации. В то же время в течение года происходит несколько десятков прерываний длительностью от долей секунды до нескольких минут в подаче энергии потребителю. В настоящее время разрабатывается бесперебойная система электропитания с использованием статических преобразователей частоты.

Эта система состоит из 3 статических преобразователей частоты типа СПЧС-20-220/400М. Преобразователь выполнен по схеме с звеном постоянного тока и содержит управляемый выпрямитель переменного тока 50Гц, автономный инвертор, преобразующий постоянный ток в переменный ток частотой 400Гц, компенсирующее устройство для стабилизации выходного напряжения инвертора и устройство защиты инвертора от токовых перегрузок. Выпрямители работают в "буфер" с аккумуляторными батареями. При повреждениях силовой сети такая система может снабжать электроэнергией вычислительную машину в течение нескольких минут (время, необходимое для переключения силовых сетей). Кроме того, аккумуляторные батареи типа ВКН-350 имеют большой срок службы и хорошо

фильтруют как пульсации напряжения выпрямителя, так и помехи из силовой сети. В качестве выпрямителя используется 6-фазный тиристорный мост с фазовым управлением и со схемой ограничения тока. Тиристорные инверторы собраны по 6-фазной схеме. Общая схема управления инверторов дает возможность им работать на общую нагрузку. Стабильность выходного напряжения инвертора  $\pm 2\%$ .

Существующая в настоящее время система контроля и регулирования выходных напряжений источников питания оператором не удовлетворяет возрастающим требованиям к надежности и эффективности работы больших вычислительных систем. Решение этой проблемы целесообразно искать в использовании ЭВМ, как органа управления для автоматического контроля и регулирования ЭП.

Введение в состав системы устройства, выполняющего функции программного контроля и управления источниками питания уменьшает время профилактической проверки, повышает надежность, дает возможность автоматизировать выбор последовательности проверки, выбор наилучших режимов работы, нахождения оптимальных величин напряжения и поддержания их в процессе работы, набора статистики по выходу из строя схем системы электропитания, что значительно повышает эффективность эксплуатации ЭВМ.

В связи с важностью поставленных задач в ЭВМ БЭСМ-10 вводится система автоматического контроля и управления питанием. Она состоит из следующих элементов:

I. Программно-управляемых стабилизированных вторичных источников питания, используемых для питания элементов ЭВМ.

2. Устройства контроля и управления питанием (УКП), с помощью которого по команде из центрального процессора (ЦП) можно контролировать и изменять питающие напряжения на всех устройствах вычислительной системы.

3. Каналов обмена, по которым поступают команды управления и выдаются результаты измерений, а также цепей измерения, сигнализации и управления.

4. Набора программ, которые задают алгоритмы работы для всей системы.

Устройство контроля и управления электропитанием представляет собой законченное функциональное устройство, связанное с периферийной машиной ПМ-10 по каналу 2-го уровня. Машина получает информацию, обрабатывает ее и выдает команды на выполнение той или иной программы, при этом должны выполняться следующие основные функции:

- а) электрическое и логическое сопряжение с каналами обмена;
- б) накопление принимаемой информации и выдача информации;
- в) обработка информации.

В разрабатываемой системе используется аппаратно-программный метод накопления, выдачи и обработки данных, который состоит в выполнении следующих задач:

1. Контроль источников питания;
2. Установка напряжений источников питания в поле допуска;
3. Определение оптимальных величин напряжений питания;
4. Изменение питающих напряжений при профилактической проверке ЭВМ;
5. Контроль температурных режимов устройств системы.

С целью увеличения гибкости системы автоматического изменения напряжений ВИП при профилактической проверке машины и нахождения их оптимальных значений предлагается использовать две периферийные машины ПМ-10, каждая из которых работает только по своему каналу при проверке определенных устройств и параллельной ПМ-10.

Устройство контроля и управления электропитанием, блок-схема которого приведена на рис. 97, представляет собой законченное функциональное устройство, состоящее из блока сопряжения с каналом 2-го уровня, обеспечивающего прием данных и команд из канала, их предварительную расшифровку, проверку правильности поступившей информации и передачу данных из буферной памяти в канал, блока управляющих регистров, предназначенных для хранения принятой информации до окончания процесса регулирования ВИП и выдачи величины установленного напряжения в цифровом коде в буфер данных. Блок дешифраторов дешифрирует полученную

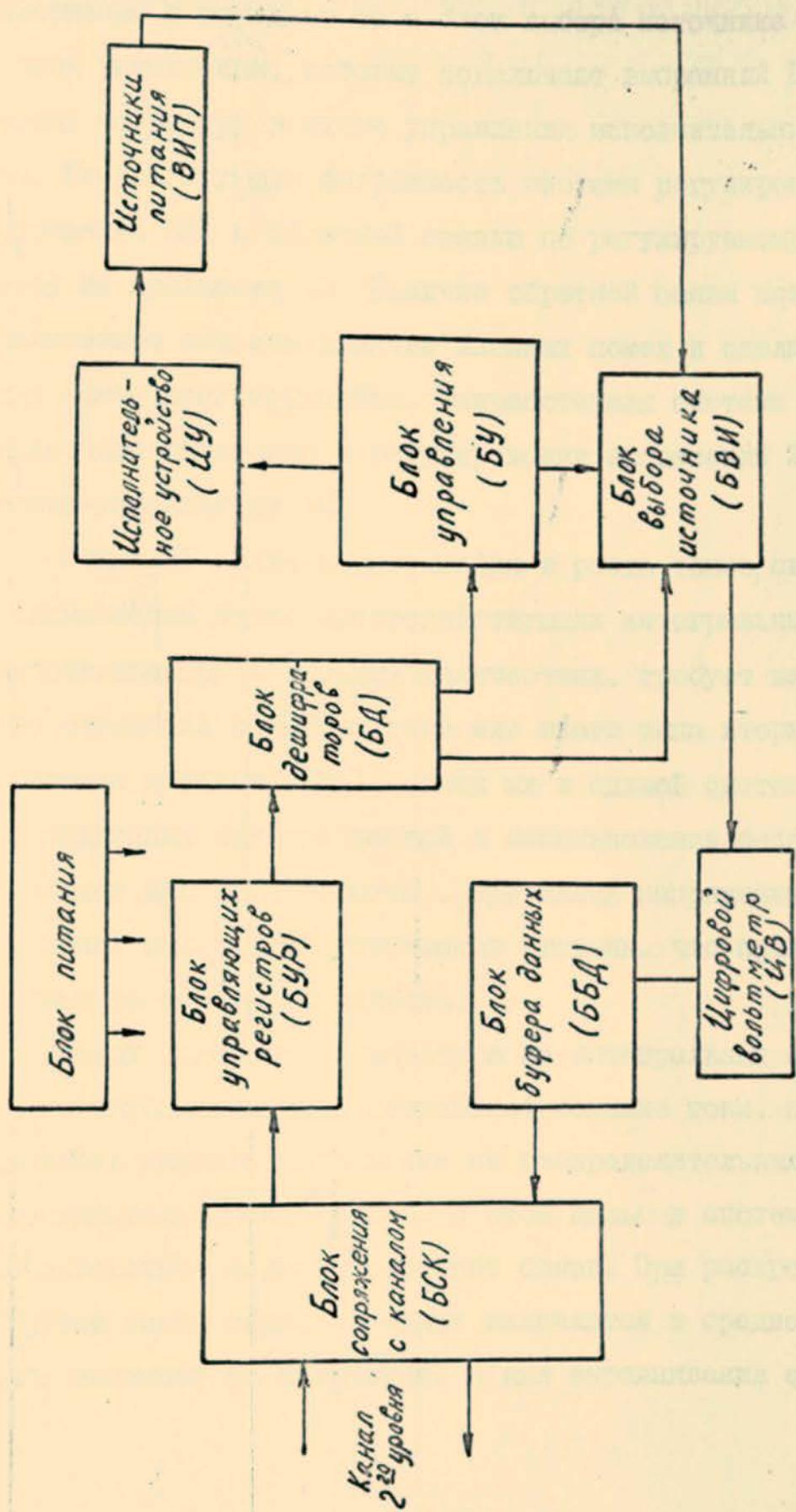


Рис. 97. Блок-схема УКУП

информацию и передает ее в блок выбора источника питания и блок управления, которые подключают выбранный ВИП к цифровому вольметру и схеме управления исполнительным устройством. Результирующая погрешность системы регулирования напряжения ВИП с обратной связью по регулируемому параметру не превышает 1%. Наличие обратной связи позволило значительно снизить влияние внешних помех и сделать систему самокорректирующейся. Разработанная система предназначена для измерения и регулирования напряжений 256 программно-управляемых ВИП.

Снижение величин напряжения и роста токов, связанное с применением более быстродействующих интегральных схем, устанавливаемых в больших количествах, требует внимательного отношения к выбору того или иного типа вторичного источника питания (ВИП), связи их в единой системе, методов индикации неисправностей и использования бесконтактных элементов для целей защиты. При малых напряжениях и больших токах падает КПД источников питания, что приводит к усложнению систем охлаждения.

Ввиду того, что аппаратура на интегральных схемах при низких напряжениях потребляет большие токи, необходимо учитывать падение напряжение на распределительных проводах и соединительных контактах. С этой целью в системе должны использоваться внешние обратные связи. При распределенной нагрузке концы обратной связи включаются в средней точке поля, занимаемого нагрузкой, а для выравнивания омических

сопротивлений варьируется длина и сечение распределительных проводов. Длина линий обратной связи должны быть ограничена 5±10 метрами, в противном случае из-за наличия распределенной индуктивности проводов может возникнуть неустойчивая работа системы. Каждая нагрузка должна быть включена парой проводов (прямой и обратный провод). Это исключит влияние одной нагрузки на другую. Для того, чтобы уменьшить возможность возникновения помех в схеме заземления машины необходимо иметь только одну точку заземления в системе электропитания.

Из-за требования малой суммарной нестабильности ( $1\pm2\%$ ) источников питания центральных процессоров и оперативных запоминающих устройств целесообразно использовать линейные стабилизаторы напряжения. В связи с этим были разработаны линейные транзисторные стабилизаторы напряжения на 2 и 5 в на токи нагрузки 50а, с суммарной нестабильностью 0,5-1%, с возможностью параллельной работы, наличием защиты от перегрузок, внешней обратной связью и возможностью программного управления.

Для питания коммутаторов, мультиплексоров и адаптеров предполагается использовать высокоэффективные бестрансформаторные ключевые стабилизаторы напряжения, удельная мощность которых достигает 40-50вт/л.

Для питания системы сигнализации и индикации разработаны электромагнитные стабилизаторы напряжения. В связи

с тем, что в разрабатываемой машине используются логические элементы с постоянным потреблением тока, значительно облегчилась разводка линий питания. С целью улучшения температурных режимов, повышения КПД всей системы, улучшения обслуживания и эксплуатации для центральной машины применена централизованная система питания, т.е. все источники питания, блоки защиты и управления установлены в стойках питания, которые отделены от устройств центрального процессора. Для охлаждения источников питания применяется принудительная вентиляция кондиционированным воздухом.

На рис.98 представлена блок-схема электропитания БЭСМ-10.

В данной системе для улучшения помехозащищенности и осуществления бесперебойной работы применены мощные статические преобразователи частоты, которые работают в режиме непрерывного подзаряда с аккумуляторной батареей. Для повышения надежности используется избыточность аппаратуры на входе системы. Для осуществления программного контроля и управления вторичными источниками питания (ВИП) разработана автоматическая система, состоящая из специального периферийного устройства УКП и программно-управляемых ВИП; для повышения электрических характеристик применены ВИП с улучшенными динамическими параметрами.

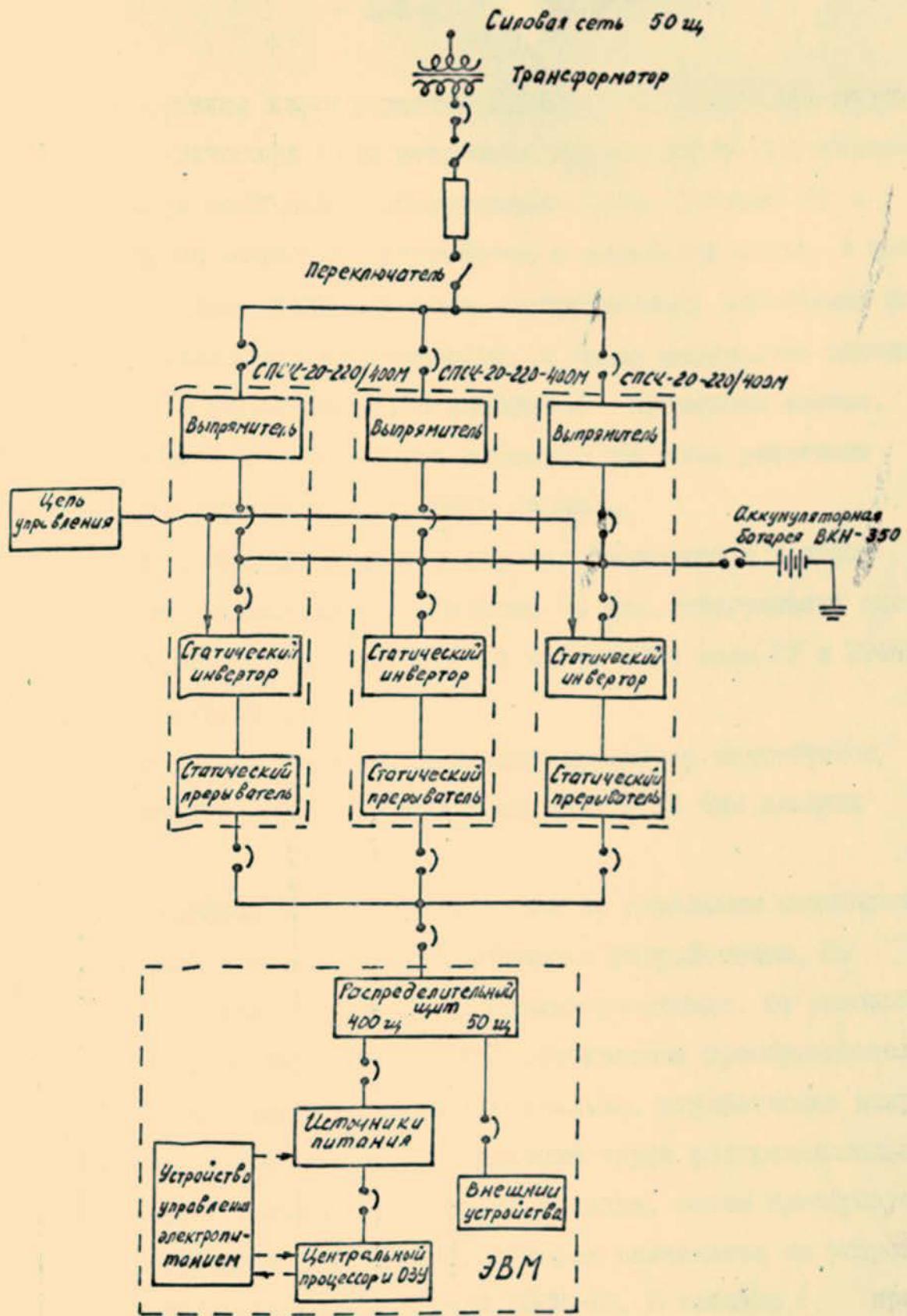


Рис. 98

Уточнение характеристик ДЛТУ-схем и проработка структуры и логических схем устройств системы БЭСМ-10 позволили определить количество оборудования, потребляемый ток и мощность на отдельные устройства и систему в целом. В таблице 3 даны величина тока, потребляемого различными схемами и выделяемая ими мощность, а также количество триодов и матриц в одной схеме. В таблице 4 приведены данные, характеризующие количество интегральных схем различных типов для отдельных устройств системы.

Для создания опытного образца вычислителя в однопроцессорном варианте необходимо 56 тыс. интегральных схем, в том числе 20480 усилительных схем, 6420 схем ЗУ и 29460 диодных матриц.

При расширении системы на каждый процессор потребуется дополнительно 7400 усилительных схем и 16 тыс. диодных матриц.

В таблице 5 приведены токи по отдельным номиналам напряжений и мощности, потребляемые устройствами. На рис.99 показана схема цепей электропитания. От силовой сети 380 в 50 гц питаются три статических преобразователя частоты, которые, работая параллельно, вырабатывают напряжение 220 в 400 гц. Это напряжение через распределительный щит питания подается на стойки питания, затем преобразуется в постоянное напряжение, которое разводится по устройствам центрального вычислителя БЭСМ-10. В таблице 6 приведены данные о количестве устройств, входящих в центральный вычислитель, а также суммарная мощность, потребляемая

Таблица 3

Схема потребляет номинал МВт	У	И	II	УК	ДМ	ПС
I. -5,2 В	73	21	42	16	44	102
2. -2,2В	II	II	4	4	-	-
3. +1,2	6	6	-	-	-	-
4. Мощность на элемент	90	38	46	20	44	102
5. Общая мощность на I корпус	180	II4	I36	80	44	204

Таблица 4

Устр-ва схемы	AY (шт.)	YY (шт.)	УУП (шт.)	KI (шт.)	COSY (шт.)	УКУП (шт.)
У	1300	1500	3500	1000	-	200
И	700	300	500	-	200	-
П	1100	700	1000	2000	400	50
УК	1100	1200	1400	1600	-	80
ДМ	10000	6000	6000	5500	300	400
ПС	-	-	-	-	2000	-
ИКУ	-	-	-	-	6400	-

Таблица 5

Устр-во	AU		УУ		УУП		COЗУ		КИ		УКУП	
	а	Квт	а	Квт	а	Квт	а	Квт	а	Квт	а	Квт
-5,2В	170	0,87	130	0,67	195	1,03	1,82	2,5	180	0,94	12	0,06
-2,2В	37	0,08	30	0,07	55	0,12	10	0,02	32	0,07	2	0,004
+1,2В	24	0,03	30	0,04	40	0,05	3	0,003	10	0,01	2	0,002
Суммарная мощность	-	0,98	-	0,78	-	1,20	-	2,53	-	1,02	-	0,066
Суммарная мощность с учетом дополнительных уте- хений ог-	-	1,28	-	0,97	-	1,63	-	2,7	-	1,33	-	0,07

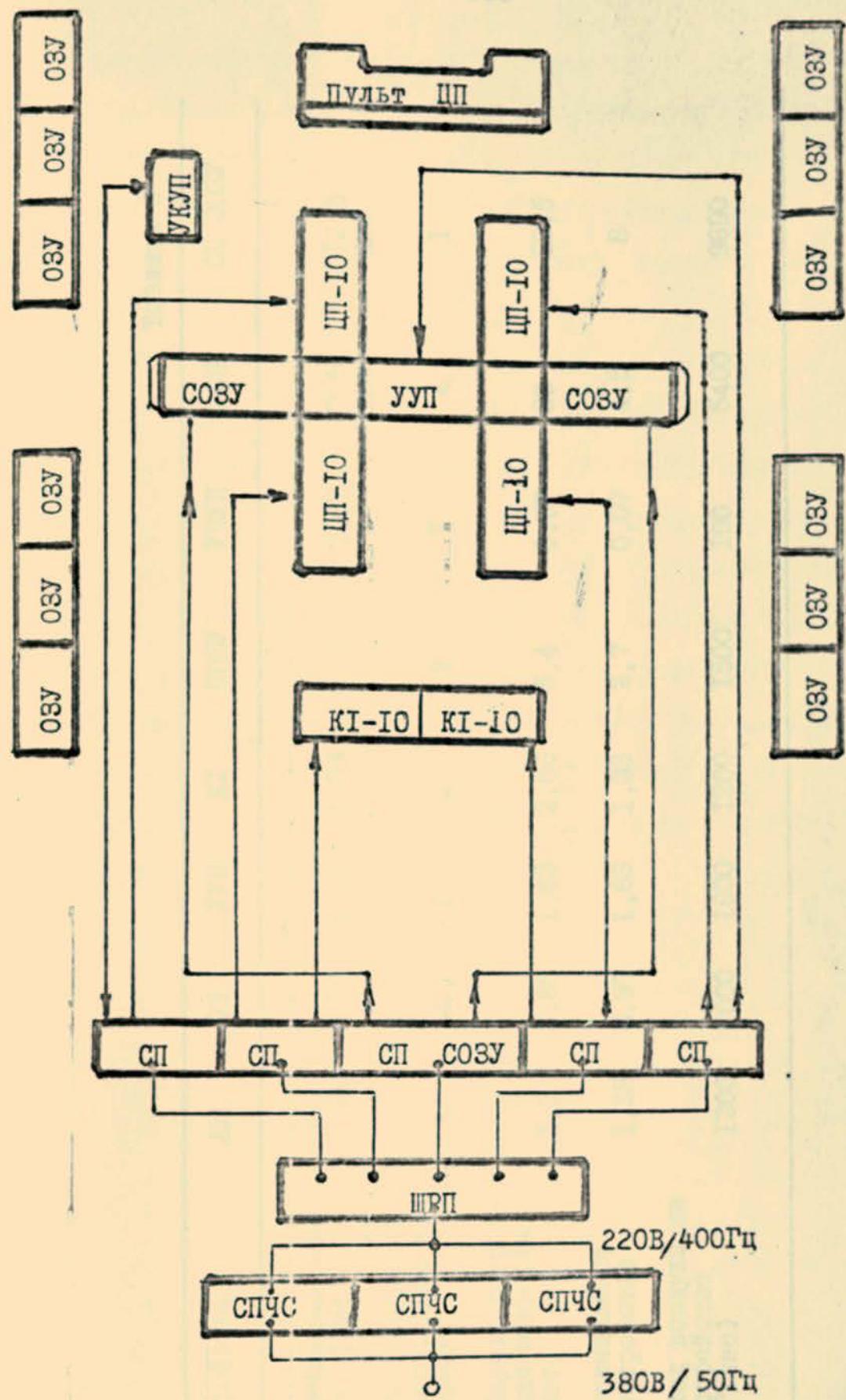


Рис.99. Схема цепей питания БЭСМ-10

Таблица 6

Устройства	АУ	УУ	УП	К1	СОЗУ	УКП	СII	СII СОЗУ
Потребляемая мощность (кВт)	1,28	0,97	1,63	1,33	2,7	0,07	7,5 кВа	12,5 кВа
Количество устройств в центральном вычислителе	4	4	1	2	2	1	4	1
Суммарная потребляемая мощность (кВт)	5,1	3,88	1,63	2,66	5,4	0,07	30	12,5
Тепловыделение на устройство	1,28	0,97	1,63	1,33	2,7	0,07	4,5	8
Расход воздуха на 1 устройство (м <sup>3</sup> /час)	1200	1000	1600	1200	1500	100	5400	9600

этими устройствами, и количество воздуха, необходимое для охлаждения этих устройств. Потребляемая центральным вычислителем мощность по сети 50 гц равна 43 ква. Расход воздуха системы вентиляции БЭСМ-10 вычисляется из условия, что для отвода 100 ватт рассеиваемой мощности необходимо  $120 \text{ м}^3/\text{час}$  воздуха при температуре  $+15^\circ \pm 2^\circ\text{C}$ . Для охлаждения центрального вычислителя БЭСМ-10 необходима подача  $50.000 \text{ м}^3/\text{час}$  воздуха при относительной влажности  $65 \pm 15\%$  с возможностью регулировки расхода и регулировки температуры от  $+15^\circ\text{C}$  до  $+18^\circ\text{C}$  с точностью  $\pm 2^\circ\text{C}$ . Воздух подается под герметичный фальшпол, на котором устанавливаются устройства системы. Вентиляционные отверстия в фальшполе герметично соединены с вентиляционными коробами шкафов. Система вентиляции для центрального вычислителя БЭСМ-10 по своим параметрам совпадает с системой вентиляции АС-6 и поэтому может быть использована вентиляция существующих вычислительных центров.

## КОНСТРУКЦИИ

В БЭСМ-10 выделяются следующие конструктивные уровни: корпус (микросхема), ячейка, плата и шкаф.

Относительно высокая плотность упаковки элементов, достигаемая применением интегральных матриц, и хорошие возможности для связи элементов между собой, обеспечиваемые ЛПУ - схемами (диодная логика - транзисторные усилители), позволяет при разработке конструкции БЭСМ-10 ограничиться наиболее простыми решениями, традиционными для ЭВМ третьего поколения, и не использовать в логических устройствах гибридных или многокристальных больших интегральных схем. Это дает возможность уменьшить число типов различных микросхем и увеличить коэффициент их применяемости.

Ячейка, показанная на рис. 100, состоит из: многослойной печатной платы (МПП), содержащей девять слоев, 90-контактного гнезда ГРПМ7-901 и корпусов (до 48 микросхем) типа 151 (см. рис. 101).

Многослойная печатная плата будет рассчитана на технологию, освоенную производством, при изготовлении ЭВМ третьего поколения серии ЕС и некоторых других специализированных изделий. Это существенно сократит сроки разработки технической документации и внедрения в производство многослойных печатных плат новых ячеек, трудоемкость изготовления которых, как правило, составляет большую часть общей трудоемкости изделия.

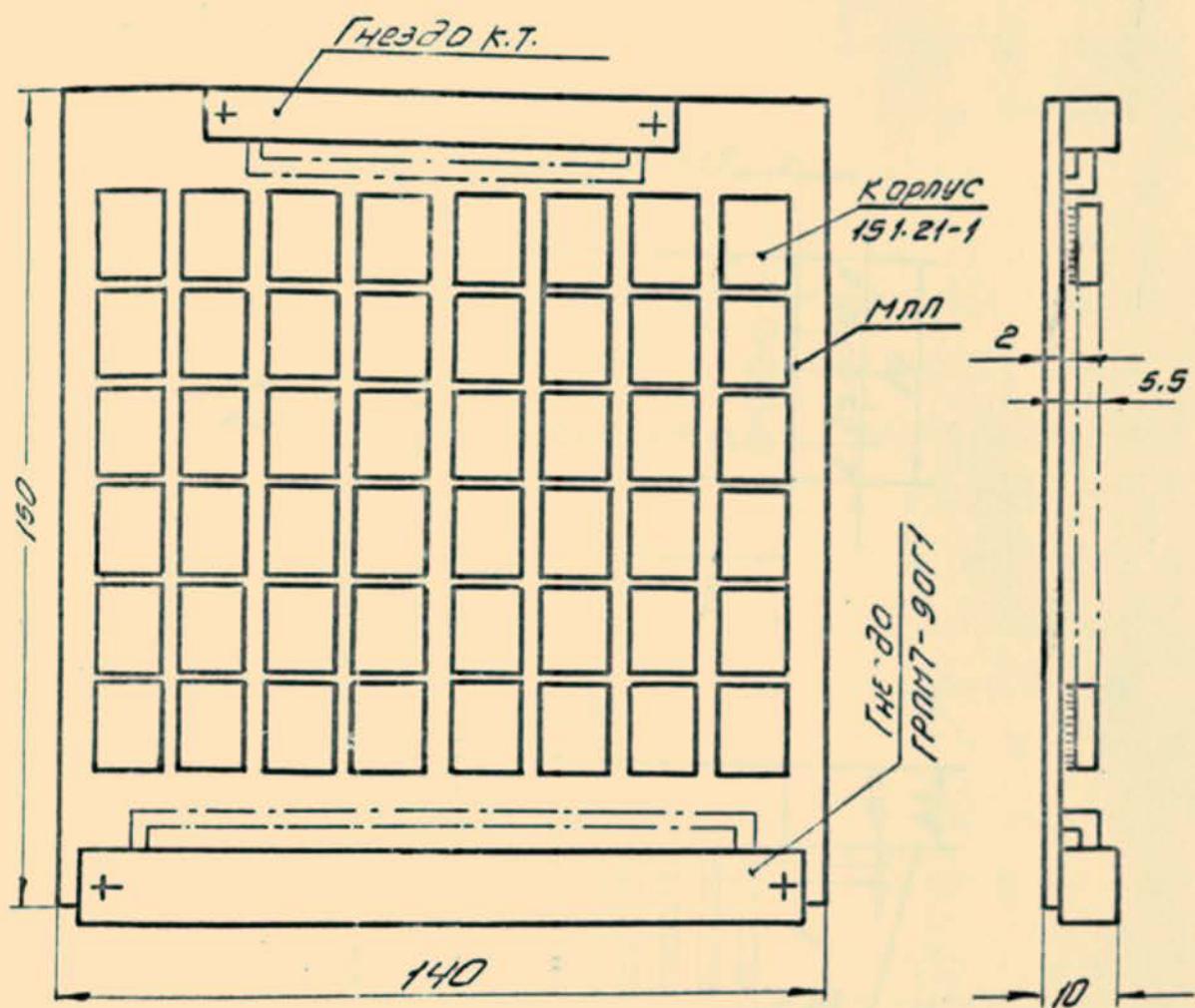


Рис. 100. Ачевка

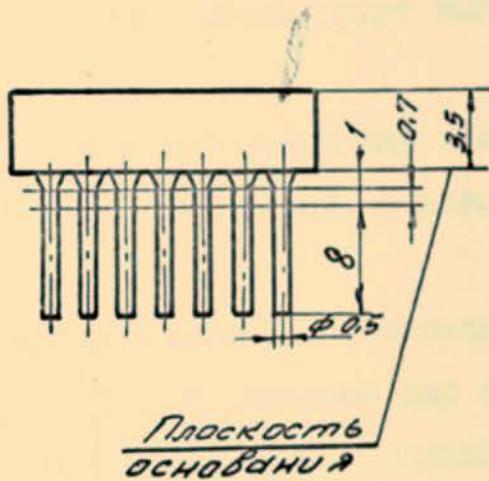
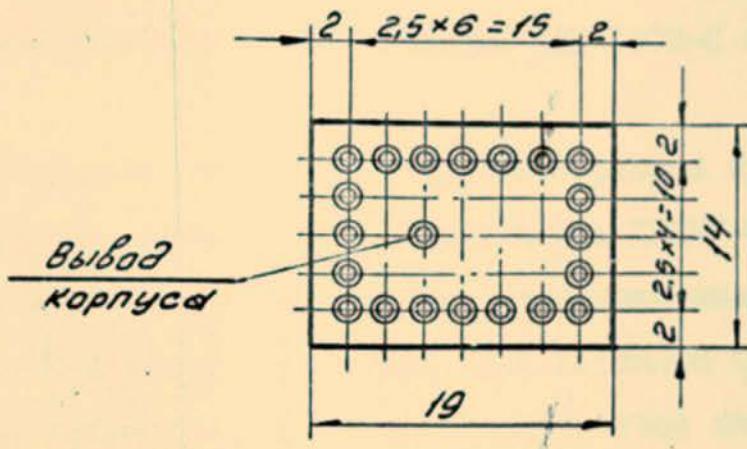


Рис. 101. Корпус 151.21-1

Плата, показанная на рис. I02 , состоит из двух плат второго уровня, собранных вместе, что позволяет производить двухсторонний монтаж. Такой двухсторонний монтаж плат позволяет минимизировать длины связей между ячейками и оправдал себя как конструктивная особенность ЭВМ БЭСМ-6 и системы АС-6.

Платы второго уровня на опытных образцах будут выполняться объемным монтажом, для чего вилка ГРПМ7-90Ш (см. рис. I03 ) имеет выводы для паск, расположенные в шахматном порядке, на боковые поверхности. При серийном производстве монтаж плат второго уровня может выполняться на многослойных печатных платах. Платы второго уровня будут двух типов: на 61 и 76 разъемов, соответственно, платы будут иметь по 122 и 152 разъема каждая.

Типовой шкаф будет состоять из шести пар плат второго уровня (панели), сборного каркаса, дверей, обшивок и других типовых деталей общих для устройств.

Устройство, показанное на рис. I04 , состоит из панели с монтажом, шкафа, ячеек и панелей разъемов для связи с другими устройствами, панели управления и автономной наладки (в случае необходимости). Компоновка каждого отдельно стоящего устройства и устройств центральной части многопроцессорной системы будет выбрана с учетом требований эргономики и технической эстетики.

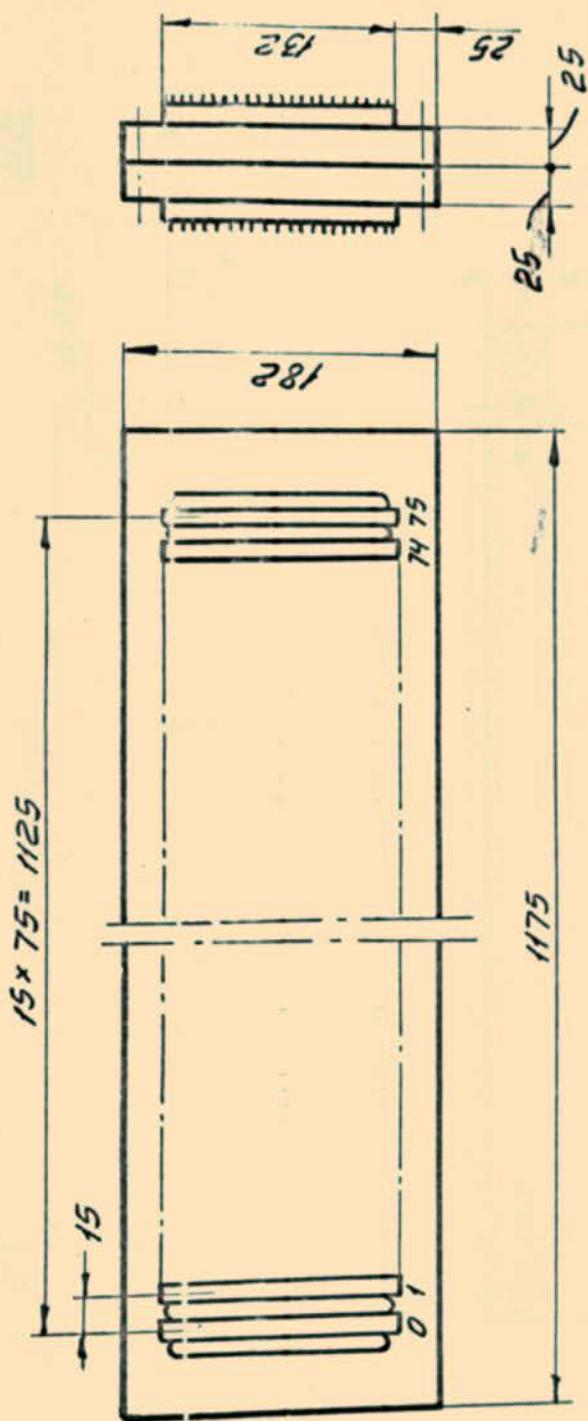


Fig. 102. Norma

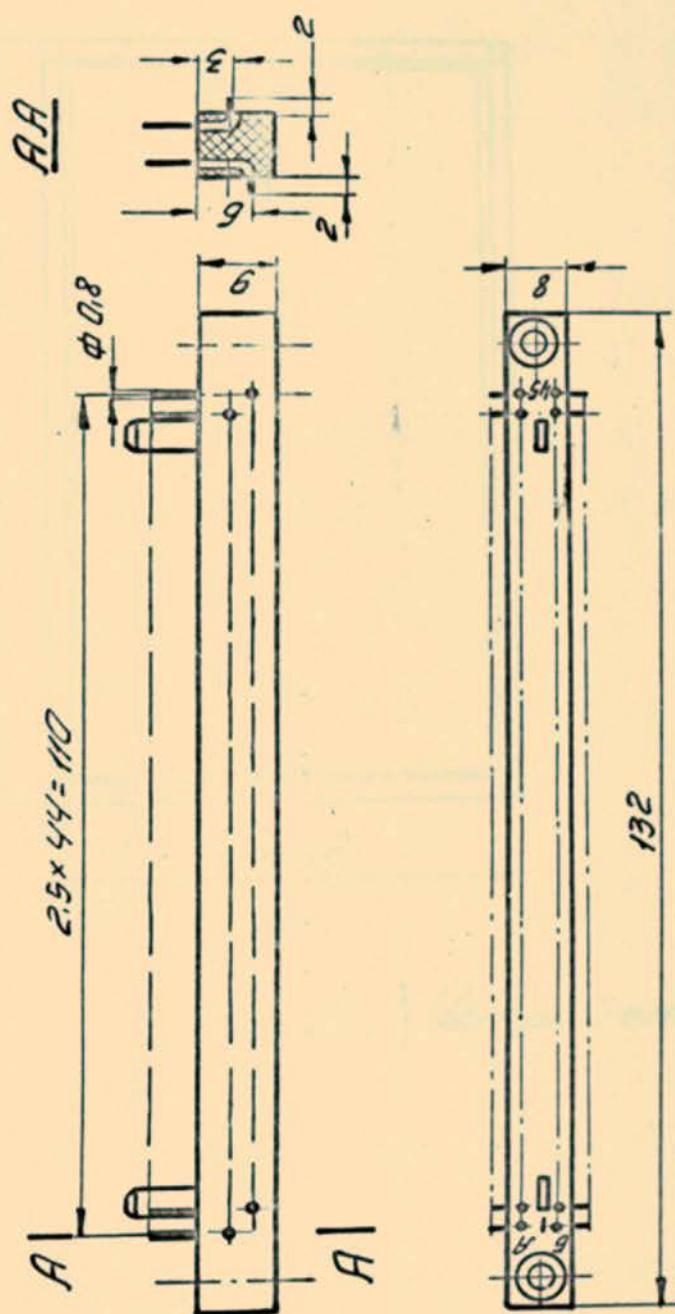


Рис. 103. Бункер РПН 7-90У

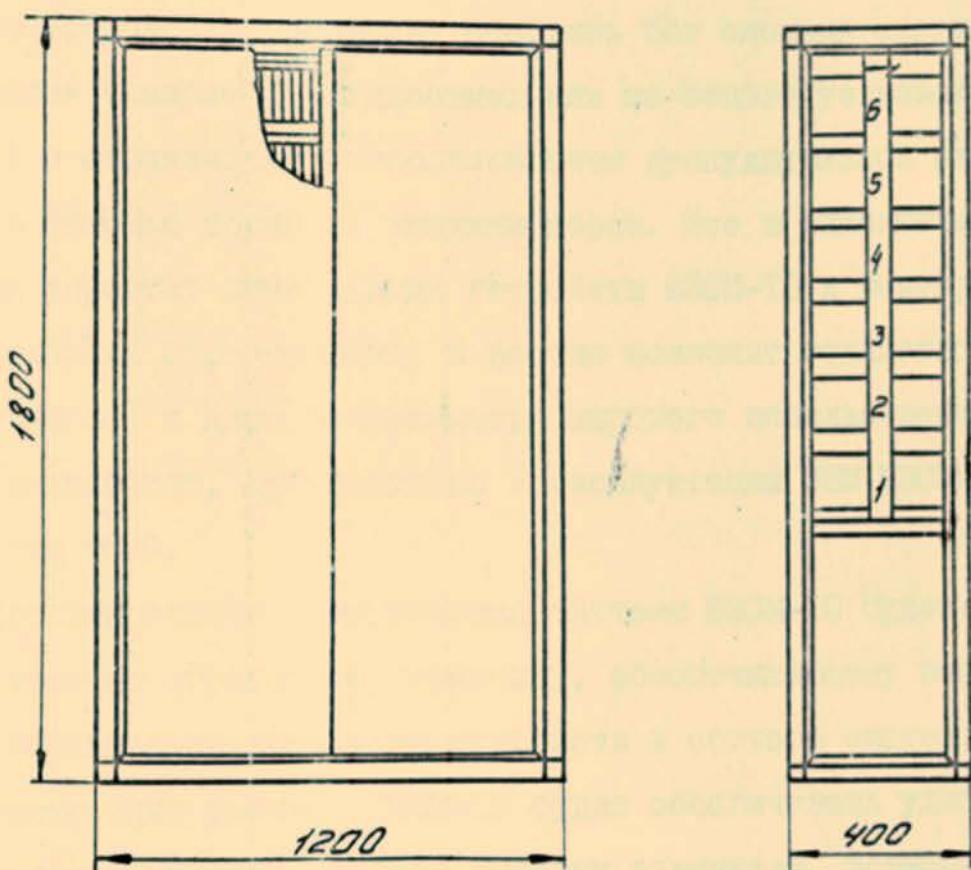


Рис. 104. Ямпou сambô

Сравнительно низкая мощность, выделяемая элементами каждого устройства, позволит обойтись без сложных систем охлаждения (например, жидкостных или на основе тепловых трубок) и ограничиться использованием принудительной вентиляции с помощью серийных кондиционеров. Это в значительной степени упростит конструкцию устройств БЭСМ-10 и конструкцию фальшпола (а, возможно, и вообще позволит отказаться от фальшпола) и дает возможность широкого использования опыта разработки, производства и эксплуатации ЭВМ БЭСМ-6 и системы АС-6.

Проектирование и компоновка системы БЭСМ-10 будет производиться по агрегатному принципу, обеспечивающему возможность наращивания комплекта устройств в составе системы.

Конструкция системы БЭСМ-10 будет обеспечивать удобство эксплуатации, доступ ко всем сменным элементам, возможность ремонта.

Конструкция и схемы системы обеспечат высокую надежность и большой срок службы (не менее 10 лет) при работе в помещениях с постоянным кондиционированием воздуха.

Конструкция системы обеспечит безопасность обслуживающего персонала при эксплуатации в соответствии с требованиями техники безопасности.

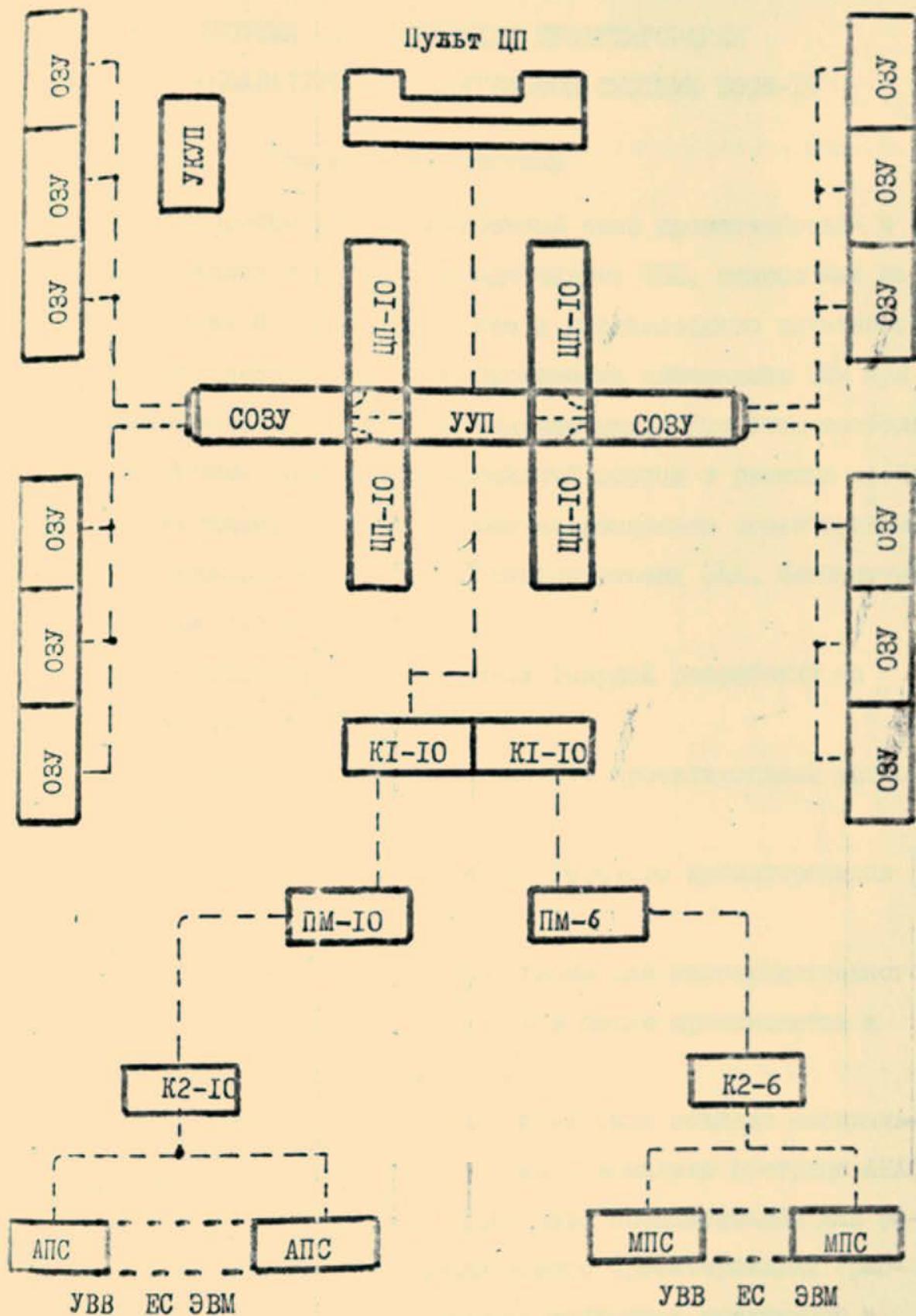
В установленном для транспортирования виде система позволит перевозить ее на место эксплуатации любыми видами

транспорта и на любые расстояния.

На рис. I05 показана схема размещения устройств центральной части системы БЭСМ-10. Четыре центральных процессора (ЦП-10), устройство управления памятью (УУП) и два сверхоперативных запоминающих устройства (СОЗУ) расположены друг относительно друга таким образом, чтобы были обеспечены наикратчайшие связи между ними. Эти связи (на рис. I05 показаны пунктиром) осуществляются с крайних разъемов плат специальными короткими кабелями. Вблизи каждого СОЗУ могут быть установлены до 12 шкафов ОЗУ ЕС ЭВМ (на схеме показаны по 6 шкафов ОЗУ). Вблизи процессоров устанавливается пульт центрального процессора и устройство управления и контроля питанием (УКУП).

Далее располагаются коммутаторы, периферийные машины и мультиплексоры.

Вблизи центрального процессора могут быть при необходимости установлены внешние устройства, необходимые для наладки и проверки работы системы (например, пишущая машинка, устройство ввода-вывода алфавитно-цифровой информации на ЭЛТ и др.). Остальные устройства системы располагаются в соседних помещениях с учетом расстояний, максимально допустимых для связей этих устройств между собой и с устройствами центральной части системы.



105 Схема размещения устройств системы БЭСМ-10

## СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ АППАРАТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ БЭСМ-10

### Назначение системы

Отечественный и зарубежный опыт проектирования и изготовления высокопроизводительных ЭВМ, основанных на применении интегральных схем и многослойного печатного монтажа, подтверждает необходимость применения ЭВМ при проектировании и выпуске документации. При этом наиболее эффективным является комплексный подход к решению проблемы - создание системы автоматизированного проектирования с использованием высокопроизводительных ЭВМ. Основными задачами системы являются:

1. Ведение архива данных текущей разработки на внешних носителях ЭВМ.
2. Решение задач логического проектирования устройств и узлов.
3. Решение задач конструкторского проектирования ячеек, блоков, панелей.
4. Выдача с ЭВМ документации для непосредственного использования в технологическом цикле производства и эксплуатации новой ЭВМ.

При разработке системы АС-6 было создано специальное математическое обеспечение - комплекс программ АПАС для ЭВМ БЭСМ-6. Система АПАС была предназначена для решения ряда задач конструкторского проектирования (раскладка диодных блоков, расчет монтажных соединений в стойках) и для выдачи технической документации на АС-6.

В основу документов по АС-6 были заложены принципы и формы, которые хорошо зарекомендовали себя на этапе создания БЭСМ-6.

В 1971-1973 годах с использованием системы АПАС была выпущена документация на АС-6. По этой документации завод САМ произвел изготовление 10 устройств. Высокая эффективность системы АПАС была доказана на практике.

Система АПАС-10 является развитием системы АПАС и в ней основное внимание уделяется этапам конструкторского проектирования - размещению и трассировке с использованием интегральных элементов, работающих в наносекундном диапазоне.

Развитие общего математического обеспечения БЭСМ-6 позволило выбрать для системы АПАС-10 эффективную систему программирования "Мониторная система Дубна" с использованием языка ФОРТРАН и ассемблера МАДЛЕН.

#### ИСПОЛЬЗОВАНИЕ ФОРМУЛ

Опыт разработки и многолетней эксплуатации машины БЭСМ-6 и системы АС-6 показал большую эффективность применения формул практически на всех этапах проектирования: разработка логики, наладка опытного образца и эксплуатация. Широкое применение формул в системе АПАС-10 позволило эффективно решить многие вопросы контроля работы системы автоматизации проектирования и задачу мелкоблочного моделирования.

Работа с системой начинается с ввода в нее формул, описывающих логику работы устройства. После каждого этапа проектирования (составление функциональной схемы, трассировка многослойных плат, составление монтажных таблиц) в системе предусмотрено автоматическое составление формул по результатам данного этапа проектирования и сравнение этих формул с теми, которые были введены разработчиком. Это позволяет, во-первых, осуществлять детальный контроль любой информации на всех этапах проектирования и, во-вторых, легко отслеживать все изменения в логике, вносимые в процессе наладки и эксплуатации.

Использование формул позволяет эффективно реализовать мелкоблочное моделирование любого устройства. Моделирование одного такта работы устройства в 10 000 пятиходовых вентилей занимает около одной секунды. Максимальный объем моделируемого узла зависит только от величины памяти на магнитных барабанах. Использование 100 трактов МБ позволяет моделировать около 60 000 пятиходовых вентилей.

#### БАЗОВАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ

Система программирования, используемая для написания программ автоматизации проектирования должна отвечать следующим требованиям:

Возможность использования языков высокого уровня одновременно с машинно-ориентированными языками;

Наличие удобной библиотеки модулей загрузки;

Возможность динамической перезагрузки программ.

Всем этим требованиям в полной мере удовлетворяет мониторная система "ДУБНА". В ней предусмотрен выход подпрограмм на ФОРТРАН и автокоде "МАДЛЕН" на одинаковые модули загрузки, которые могут храниться в нескольких библиотеках разного уровня. В рамках мониторной системы накоплено множество сервисных подпрограмм редактирования и хранения текстов во внешней памяти, подпрограмм работы с графопостроителем и т.д. Все это делает мониторную систему "ДУБНА" весьма подходящей для использования в крупной системе автоматизации проектирования.

#### ОРГАНИЗАЦИЯ ВНЕШНЕЙ ПАМЯТИ

Современная система автоматизации проектирования имеет много общего с информационно-поисковой системой. Поэтому одним из центральных вопросов является вопрос о хранении информации на внешних носителях.

Данные, с которыми имеет дело система автоматизации проектирования, имеют некоторые специфические особенности. К ним можно отнести довольно четко выраженную иерархическую многоуровневую структуру и чрезвычайное разнообразие в форматах. Более того, как в процессе проектирования, так и в процессе эксплуатации системы автоматизации проектирования эти форматы довольно часто меняются.

Для работы с информацией, возникающей в процессе проектирования, была разработана система программ, облегчающая поиск и хранение самых разнообразных данных. Эта система программ обеспечивает:

1. Присвоение имени любому массиву данных;
2. Обмен между различными внешними запоминающими устройствами, а также между внешними устройствами и оперативной памятью;
3. Автоматическое распределение памяти внешних запоминающих устройств.

Возможность использования во всей системе автоматизации проектирования единой системы массивов данных существенно облегчаетстыковку между программами, написанными разными программистами.

В качестве внешней памяти в системе используются, архивная магнитная лента, буферная магнитная лента и магнитные барабаны.

При наличии<sup>и</sup> дисков архив и буфер легко могут быть переведены на дисковую память.

В архиве системы хранится и накапливается вся информация, связанная с проектированием устройства. Каждый запуск системы начинается с переписи из архивной ленты на барабаны массивов, которые необходимы для оперативной работы. Как правило, объем памяти на барабанах оказывается недостаточным, и поэтому часть массивов, в которых произошли изменения, приходится хранить на буферной ленте. Эти

массивы можно было бы записывать сразу в архив, минуя буферную ленту, однако, это не сделано из соображений надежности. Система построена так, что запись на архивную ленту производится только один раз, в самом конце работы. Это позволяет не включать запись на накопителе с архивной лентой вплоть до самого конца работы, что существенно повышает надежность всей системы в целом.

#### АРХИВ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ

Перед этапом конструкторского проектирования устройство, по-возможности, разбивается на функционально законченные узлы. Объем каждого узла не должен превышать нескольких тысяч вентилей. Далее проектирование различных узлов идет параллельно и независимо друг от друга вплоть до этапа составления монтажных таблиц.

На каждый узел в системе заводится:

Словарь названий сигналов, который обеспечивает компактное и удобное хранение всей остальной информации;

Архив формул, описывающий логику работы узла;

Архив функциональных элементов, в котором накапливаются все результаты конструкторского проектирования, кроме результата работы программы трассировки.

В архиве функциональных элементов содержатся описания всех микросхем и плат, входящих в данный узел. Кроме того, в нем могут храниться и элементы функциональной схемы, еще не привязанные к какой-либо физической реализации. В описании каждой микросхемы (элемента) присут-

ствует тип микросхемы (тип элемента функциональной схемы), порядковый номер в узле, физические координаты по устройству и перечисление сигналов на всех выводах (ножках). Физические координаты микросхемы — ориентация и координаты на плате, адрес разъема — формируются по мере прохождения различных этапов проектирования.

Описание 1000 микросхем, каждая из которых имеет 20 выводов, занимает 8 трактов магнитного барабана.

#### АРХИВ ФОРМУЛ

Широкое применение архива формул в системе автоматизации проектирования ставит задачу нахождения компактного и удобного для работы способа записи формул. В процессе проектирования часто возникает необходимость одновременного хранения нескольких вариантов формул. Описываемый архив позволяет иметь практически неограниченное количество вариантов, причем совпадающие части различных вариантов хранятся в одном экземпляре.

Логика работы большинства разрядов регистров массовых цепей практически совпадает. Для экономного описания логики работы массовых цепей в архиве предусмотрена возможность хранения лишь одной формулы на регистр (или группу регистров) с указанием к каким разрядам регистра эта формула относится.

Архив формул на узел из 5000 5-входовых вентилей с нерегулярной логикой занимает около 10 трактов на магнитном барабане.

## СЛОВАРЬ НАЗВАНИИ СИГНАЛОВ

Во всех частях системы автоматизации проектирования встает задача хранения и обработки большого количества названий сигналов. Каждое название обычно по несколько раз входит в различные таблицы системы. Чтобы съэкономить на довольно громоздких кодах названий (обычно от 5 до 10 байтов), целесообразно сами названия хранить только в одном месте, в системной таблице, которую можно назвать словарем названий сигналов. Словарь названий сигналов каждому названию ставит в соответствие его номер, что позволяет во всех внутренних таблицах системы хранить только номера, а не сами названия. Под номер названия в системе отводится 16 разрядов.

Использование номеров названий сигналов вместо их непосредственно текстового представления приводит к сокращению многих таблиц системы в 3-5 раз.

Словарь названий сигналов состоит из двух основных таблиц: таблицы входов и таблицы названий.

Все названия в словаре условно разбиты на 1024 класса. Названия, относящиеся к одному классу, связанны в кольцевой список. Кольцевые списки, соответствующие непустым классам образуют таблицу названий.

Таблица входов, состоящая из 1024 16-разрядных адресов, служит для указания начальных элементов списков названий сигналов.

Объем словаря названий сигналов можно значительно

уменьшить, используя некоторые особенности названий сигналов. Многие названия имеют общую основу и отличаются лишь несколькими символами в самом начале или конце названия. Это позволяет так организовать списки, что на каждое название приходится в среднем I.2 - I.3 ячейки.

При общем количестве сигналов 20-30 тыс. в каждом списке будет, как правило, около 20-30 названий. Отсюда следует, что в случае хранения таблицы названий сигналов на магнитном барабане, для вызова в ОЗУ всего списка в большинстве случаев потребуется считывание только одного тракта, и трансляция тысячи названий в номера (соответственно восстановление тысячи названий по номерам) займет около 20 секунд.

#### РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ

Известны следующие критерии, применяемые при размещении:

- 1) минимум суммарной длины проводников;
- 2) минимум максимального по длине соединения;
- 3) минимальное число пересечений соединений схемы;
- 4) комбинированные критерии указанных выше типов;
- 5) максимальная связность между элементами.

В связи с использованием элементов с фронтами сигналов порядка 2-3 нсек возникают специфические ограничения на постановку задачи размещения элементов на плате. Эти ограничения требуют выбора других критериев размещения. Действительно, для схем с элементами наносекундного диапазона, весьма существенна длина соединения максималь-

но удаленного элемента-приемника от элемента-источника сигнала. Существенна также компактность размещения элементов-приемников около элемента-источника. Поэтому возникает необходимость использования критерия нового типа — критерия компактности размещения приемников вокруг источника с ограничением на длину соединения максимально-удаленного приемника. (Критерий компактности).

В настоящее время, наиболее эффективным приемом считается использование связки двух алгоритмов размещения: алгоритма последовательного размещения и алгоритма итеративных перестановок, использующего результат работы первого как начальное размещение. Именно такая система из двух алгоритмов и будет описана.

Наиболее эффективным критерием алгоритмов последовательного размещения является критерий максимальной связности между элементами. Специфические ограничения наносекундного диапазона учитываются дополнительным ограничением на максимально-возможную длину соединения между приемником и источником сигнала. Широко используемые алгоритмы размещения элементов, такие как алгоритм Штейнберга, Гильмора, Важенина, используют матрицу фиксированных соединений. Для ее получения необходимо до размещения определить структуру дерева соединений логического звена, что значительно ухудшает качество размещения.

Из практики использования различных алгоритмов размещения известно, что алгоритмы последовательного размещения, использующие матрицу "элементы - логические звенья" более точны, чем итерационные алгоритмы размещения, оперирующие с матрицей фиксированных соединений. Отметим, что использование матрицы "элементы - логические звенья" для вычисления коэффициента связности затрудняется необходимостью учета циклов при формировании соединительных деревьев. Существующие алгоритмы последовательного типа не всегда имеют эффективные процедуры учета циклов.

Предлагаемый подход позволяет точно вычислять коэффициент связности по полным графикам звеньев цепей с учетом циклов и позволяет этап оценки и выбора очередной позиции свести к тривиальному решению.

Алгоритм существенно использует понятие  $\varepsilon$ -окрестности фиксированной позиции. Под окрестностью  $\varepsilon_1$  будем понимать совокупность соседних позиций, находящихся рядом на одной горизонтали, вертикали или диагонали. Можно аналогичным образом ввести понятие окрестностей  $\varepsilon_2, \varepsilon_3, \dots$  пока какая-то окрестность  $\varepsilon_K$  не включит в себя все позиции платы.

Наилучшим претендентом на размещение в фиксированной позиции считается элемент, имеющий наибольшее число связей с  $\varepsilon_J$  окрестностью, где  $J = 1, 2, \dots, K$ .

Значение индекса  $\varepsilon$ -окрестности может задаваться как параметр режима и меняется автоматически в процессе работы алгоритма. Основным режимом работы считается

использование  $\varepsilon_1$  -  $\varepsilon_3$  окрестностей.

Рассмотрим идею точного вычисления коэффициента связности с учетом образования циклов. В процессе размещения элементов формируется дерево соединений. Оно может быть связным или состоять из нескольких компонент связности. В несвязном графе типа дерева разность между числом вершин и ребер больше единицы. К связному дереву - графу можно подсоединить вершину только одним ребром без образования цикла. К несвязному дереву - графу можно подсоединить вершину без образования циклов таким количеством ребер, сколько имеется связных компонент.

Реализация этого утверждения для  $\varepsilon_1$  - окрестности может быть такова. Пусть звено цепи содержит в  $\varepsilon_1$  - окрестности хотя бы два элемента  $e_1$  и  $e_2$  и ему принадлежит претендент  $e^*$ . Если ранее расположенные элементы логического звена имеют связное дерево соединений по  $\varepsilon_1$  - окрестностям, то  $e^*$  подсоединяется к одному элементу  $\varepsilon_1$  - окрестности. Если же дерево соединений не связно, то считаем связь  $e^*$  с обоими элементами  $e_1$  и  $e_2$  звена цепи. Аналогичные рассуждения для любого индекса  $\varepsilon$  - окрестности позволяют вычислять коэффициент связности без учета циклов.

Ниже представлены три алгоритма последовательного размещения, каждый из которых по-своему реализует описанный выше подход к вычислению коэффициента связности.

Первый алгоритм А1 предназначен для размещения большого числа элементов до  $5 \cdot 10^3$  на плате, если элементы имеют около 12 сигнальных выводов, или для размещения

элементов с большим числом выходных контактов, например, плат I уровня в кассетной конструкции или на плате II уровня и т.д.

Поэтому алгоритм использует компактное представление входной информации, имеет специальные процедуры общения с барабаном. Процедуры вычисления коэффициента связности и выбора позиции максимально упрощены.

Алгоритмы А2 и А3 предназначены для размещения элементов непосредственно на плате. Максимально возможное число элементов в пересчете на 12-сигнальные элементы несколько превышает 500. Алгоритм А2 использует другую, чем в А3 форму представления входной информации и, соответственно, другую процедуру вычисления коэффициента связности.

Оба алгоритма учитывают директивно расположенные элементы и наличие запрещенных мест.

Алгоритм А2 более универсален, имеет несколько режимов счета, зависящих от стратегии выбора позиций, от способа учета весовых коэффициентов по направлению, по глубине  $\mathcal{E}$  - окрестности и т.д.

В алгоритме А3 реализовано ограничение на длину соединения между источником сигнала и приемником, более точное вычисление коэффициента связности.

#### АЛГОРИТМ А1

Предлагаемый последовательный алгоритм размещения функциональных элементов, предназначен для получения предварительной информации для работы алгоритма разбиения

функционального узла . Он также может быть использован автономно для размещения микросхем на плате.

Алгоритм АІ использует исходную информацию в виде таблицы звеньев цепи, таблицы элементов и справочника элементов. Таблица звеньев цепей состоит из названий звеньев цепей и перечисления элементов, входящих в соответствующие звенья. Таблица элементов содержит перечисление адресов тех цепей, в которые входит элемент. В справочнике элементов хранится адрес начала перечисления каждого элемента. В качестве критерия размещения выбран коэффициент связности с элементами, занимающими соседние позиции.

В алгоритме АІ, как и в других последовательных алгоритмах, различают два принципиально разных этапа: выбор позиции для претендента и выбор претендента на эту позицию. Отличительной чертой алгоритма АІ является то, что сначала происходит выбор позиции, а затем происходит назначение претендента для нее.

Существуют несколько режимов выбора позиций:

1. На плате, обычно в центре, ставится элемент. Следующие позиции выбираются последовательно по периметрам, окружающим этот элемент. При этом начало и направление обхода периметра являются параметрами программы.

Этот режим применяется для предварительного размещения микросхем с последующим разрезанием функциональной схемы на конструктивные элементы.

2. В этом режиме заполнение платы начинается с максимального периметра. Возможны все варианты обхода

периметра, перечисленные в предыдущем пункте.

Этот режим может быть применен для автономной работы алгоритма.

3. Нефиксированный выбор позиции. В отличие от предыдущих режимов, где выбор позиции осуществлялся последовательным фиксированным образом, этот режим дает более качественное размещение. На незанятые позиции вводится некоторая оценочная функция аналогичная коэффициенту связности элемента, и следующей выбирается позиция, имеющая максимальное значение этой функции. Нефиксированный выбор позиции возможен, как от центра, так и от внешнего периметра.

После выбора позиции осуществляется определение элемента, который в ней необходимо разместить. Для этого определяется подмножество претендентов  $E_i$  для назначения в позиции  $P_i$ . Для этого определяем подмножество уже размещенных элементов в  $\mathcal{E}I$ -окрестности позиции  $P_i$ . По таблице элементов определяем подмножество звеньев  $S_{\mathcal{E}}$ , в которое входят элементы  $\mathcal{E}I$ -окрестности. Затем по таблице звеньев цепей находим множество претендентов  $E_i$ . Для каждого претендента  $e_j \in E_i$  определяется коэффициент связности с элементами  $\mathcal{E}I$ -окрестности  $K_{ej}$  по формуле:

$$K_{ej} = \sum_{q=1}^L K_{ej}^q \beta^q \gamma^q$$

где  $K_{ej}^q$  - число связей между  $e_j$  и элементами  $\mathcal{E}I$ -окрестности по  $q$ -му звену;  $L$  - число общих между  $e_j$  и элементами  $\mathcal{E}I$  окрестности;  $\gamma^q$  - весовой коэффициент по

направлению связей, который вычисляется особым способом;  $\beta^q$  - весовой коэффициент  $q$ -того звена.

В свою очередь  $K_{\varepsilon j}^q$  вычисляется по формуле:

$$K_{\varepsilon j}^q = \begin{cases} 1, & \text{если } n_p^q - m_p^q = 1, n_{\varepsilon} = 1 \\ t, & \text{если } n_{\varepsilon} > 1, n_p^q - m_p^q > 1 \end{cases}$$

где  $n_p^q$  - число размещенных элементов  $q$ -го звена;

$m_p^q$  - число размещенных соединений  $q$ -го звена по  $\varepsilon I$ -окрестности;  $n_{\varepsilon}$  - число размещенных элементов  $q$ -го звена в  $\varepsilon I$ -окрестности;

$t$  - число связных компонент дерева соединений, имеющих хотя бы один элемент в  $\varepsilon I$ -окрестности.

Далее определяется претендент  $e_d$  по значению индекса максимального коэффициента связности

$$K_d = \max_{j \in E_i} K_{\varepsilon j}$$

Если несколько элементов имеют одинаковый коэффициент связности, то выбирается любой из них. В случае, если коэффициенты связности неразмещенных элементов равны нулю, предусмотрено несколько режимов пропуска этой позиции. Возврат происходит в том случае, если для очередного претендента не хватает позиции.

Для лучшего размещения используется аппарат весовых коэффициентов на цепи и на направления связей в  $\varepsilon I$ -окрестности.

Последовательный алгоритм размещения реализован на БЭСМ-6 в операторах автокода "МАДЛЕН". Общий объем команд составил  $2,5 \cdot 10^3$ .

Время счета по данному алгоритму, при размещении 72 элементов с общим количеством цепей около 1300, составляет 40 секунд.

### Алгоритм А2

Исходная информация задается в виде матрицы  $(N+1) \times M$  ( $N$  - число элементов,  $M$  - число цепей).

$A = \{a_{ij}^j\}$ , где  $j$  - номер элемента,  $i$  - номер звена цепей. Элемент  $a_{ij}^j = 1$ , если  $i$  - звено проходит через элемент  $j$ , и 0 в противном случае. Разъем фигурирует как  $N+1$  элемент. Матрица  $A$  в памяти машины хранится по столбцам, причем каждый элемент столбца занимает один разряд машинного слова.

Коэффициент связности элемента  $\alpha$  с элементами  $\mathcal{E}$  - окрестности вычисляется по формуле

$$K_{\alpha}^{\mathcal{E}} = \sum_{j=1}^n \sum_{i=1}^M a_{ij}^{\alpha} \wedge a_i^j \bigwedge_{e=0}^{j-1} \bar{a}_i^e \quad (1)$$

где  $n$  - число элементов  $\mathcal{E}$  - окрестности.  $M$  - число звеньев,  $a_i^0$  - нульвектор.

В алгоритме предусмотрены различные варианты выбора очередного элемента  $\mathcal{E}$  - окрестности, охватывающие возможные случаи. Кроме того, для каждого направления вводятся весовые коэффициенты, значения которых можно варьировать. На каждом шаге алгоритма в качестве претендентов на размещение рассматриваются все неразмещенные элементы. Из них определяется элемент с наибольшим полным коэффициентом связности  $K_{\alpha}$ , т.е.

$$K = \max_{\alpha=1,2,\dots,n} K_{\alpha}^{\mathcal{E}} \quad (2)$$

где  $N_2$  - число неразмешенных элементов. Полный коэффициент связности элемента  $\alpha$   $K_{\alpha}^P$  вычисляется при этом по формуле

$$K_{\alpha}^P = \beta_1 \cdot K_{\alpha}^{\ell} + \beta_2 K_{\alpha}^{\varepsilon} + \beta_{\sigma\tau\rho} K_{\alpha}^{\sigma\tau\rho} \quad (3)$$

где  $K_{\alpha}^{\ell}$  - коэффициент связности с разъемом,  $K_{\alpha}^{\varepsilon}$  - коэффициент связности с  $\varepsilon$  - окрестностью,  $K_{\alpha}^{\sigma\tau\rho}$  - коэффициент связности с противоположной стороной платы,  $\beta_1, \beta_2, \beta_{\sigma\tau\rho}$  - соответствующие весовые коэффициенты. Процедура выбора очередной позиции, а также вычисления  $K_{\alpha}^{\sigma\tau\rho}$  зависит от расположения внешних контактов.

Алгоритм может быть использован в режиме одноразового или многоразового счета (с возвратом на просчет с другими режимами работы), а также с возвратом на позиции с неоднозначным выбором элементов. Таким образом, возможно итеративное использование алгоритма.

Предусмотрены следующие режимы выбора весовых коэффициентов:

- а) постоянные ( $=1$ ) весовые коэффициенты;
- б) выбор весовых коэффициентов из таблиц (по направлениям или при наличии "опасных" цепей);
- с) вычисление коэффициентов по формулам:

$$\beta_{\varepsilon} = \frac{4}{\varepsilon} \quad \beta_{\sigma\tau\rho} = \zeta$$

где  $\zeta$  - расстояние до "противоположной" стороны.

Кроме того, предусмотрено 2 режима обхода  $\varepsilon$  - окрестности, а при  $\varepsilon_{max} > L$  предусмотрено:

- а) увеличение  $\varepsilon$  (до  $\varepsilon_{max}$ ) лишь при  $K_1 = K_2$  ;

б) вычисление коэффициента связности при  $1 \leq \varepsilon \leq \varepsilon_{\max}$  для всех претендентов.

Программа размещения реализована на БЭСМ-6 на автокоде "МАДЛЕН", общий объем программы составил  $3 \cdot 10^3$  операторов автокода. Максимальное число элементов, предусмотренное в программе, равно 500 при числе цепей 1500. Скорость работы программы оценивалась в режиме двух разъемов. Время счета для 84 элементов (1500 цепей) не превосходит минуты, для 200 элементов составляет 5 минут и для 400 элементов - 20 минут (последнее - в простейшем режиме работы).

#### Алгоритм А3

Для второго варианта последовательного алгоритма размещения исходная информация задается в виде таблиц звеньев цепей, таблицы элементов и справочника элементов. Таблица звеньев цепей состоит из названий звеньев и совокупности номеров элементов, составляющих эти цепи. Различно маркируются элементы-источники и элементы-приемники. Каждой цепи может быть присвоен весовой коэффициент  $\omega_i$ . Все цепи разбиты на несколько типов в основном отражающих их связь с разъемом.

Таблица элементов состоит из перечисления адресов тех цепей, в которые входит элемент. В справочнике элементов хранится адрес начала перечисления каждого элемента.

На каждом шаге работы алгоритма производятся:  
I. Выбор позиции для размещения;

2. Оценка позиции для размещения;
3. Размещение элемента в выбранную позицию.

#### ВЫБОР ПОЗИЦИИ ДЛЯ РАЗМЕЩЕНИЯ

На данном этапе для каждой незанятой позиции  $J$  оценивается взвешенное число  $L$ , недоразмещенных цепей  $\{k\}$  элементы которых принадлежат  $E_i$  — окрестности позиции  $J$ . 
$$L = \sum_{\{k\}} L_{Wk}$$

Выбирается позиция с максимальным значением  $L$ . При наличии неразмещенных элементов, связанных с разъемом, выбирается позиция, находящаяся в непосредственной близости от разъема.

#### ОЦЕНКА РАСПОЛОЖЕНИЯ ЦЕПЕЙ

В алгоритме А3 учтены некоторые требования схем наносекундного диапазона:

1) расстояние по линиям связи между двумя любыми элементами-источниками сигналов одного логического звена не должно превышать значения  $\ell_0$ ;

2) расстояние между источником и элементами-приемниками одного логического звена не должно превышать  $R_0$ , где  $R_0 = R(n^q)$

Считаем, что все источники одного логического звена, обычно так называемого "проводного ИЛИ", расположены не более чем в четырех корпусах интегральных схем. Если корпус интегральной схемы включает в себя источники двух логических звеньев с "проводными ИЛИ", то одно из них

может заключаться в четырех корпусах, а другое только в трех. В алгоритме размещения при постановке в очередную позицию интегральной схемы с "проводным ИЛИ", которое заключается более чем в одном корпусе, предусмотрено остальные корпуса "проводного ИЛИ" размещать в одну из соседних позиций.

Второе ограничение реализуется с помощью функции дефекта  $F(D)$ . Если размещен элемент-источник логического звена, то от него определяется расстояние до размещаемых элементов-приемников. Если расстояние от источника до элемента приемника меньше  $R_o$ , то  $F(D)=I$ , если равно  $R_o$ , то  $F(D)=C_1$ , если больше  $R_o$ , то  $F(D)=C_2$ . Причем значения  $C_1$  и  $C_2$  зависят от числа и конфигурации плохо расположенных элементов-приемников. Если первым в звене размещается элемент-приемник, то от него оценивается расстояние до размещаемого элемента-источника.

#### РАЗМЕЩЕНИЕ ЭЛЕМЕНТА В ВЫБРАННУЮ ПОЗИЦИЮ

Используя таблицу цепей и элементов, выделяется  $E_i$  – множество неразмещенных элементов, которые связаны с элементами  $\mathcal{E}$  – окрестности заполняемой  $i$ -ой позиции. Находится пересечение  $P$  множества звеньев претендента  $e_j$ , которое считывается из таблицы элементов, и множества звеньев, содержащих элементы  $\mathcal{E}$  – окрестности. Затем определяется число связей претендента  $e_j$  с элементами  $\mathcal{E}$  – окрестности по каждому звену. Из развитого ранее принципа вычисления коэффициента связности по звену следует, что число связей претендента  $e_j$  равно 1, если

дерево соединений связно, и равно числу несвязных компонент, если дерево-граф несвязно. С использованием связей только по  $\mathcal{E}$ -окрестностям в случае несвязного графа-дерева, число связей претендента  $e_j$  с элементами  $\mathcal{E}$ -окружности равно числу несвязных компонент, хотя бы один элемент которых принадлежит  $\mathcal{E}$ -окрестности.

Введем следующие обозначения:

$n_p^q$  - число размещенных элементов  $q$ -ого звена;

$m_p^q$  - число размещенных соединений  $q$ -ого звена по  $\mathcal{E}$ -окрестности;

$n_{\mathcal{E}}$  - число элементов  $\mathcal{E}$ -окрестности  $q$ -ого звена.

Тогда коэффициент связности по  $q$ -ому звену претендента  $e_j$  с элементами  $\mathcal{E}$ -окрестности вычисляется по формуле:

$$K_{e_j}^q = n_p^q - m_p^q - \gamma^q$$

где  $\gamma^q$  - коэффициент коррекции.

Значение  $\gamma^q$  есть число несвязных компонент, лежащих вне  $\mathcal{E}$ -окрестности. Оно вычисляется по специальной процедуре в случае, если  $n_p^q - m_p^q > 1$  и  $n_{\mathcal{E}} > 1$ . В случае вычисления  $K_{e_j}^q$  по  $\mathcal{E}$ -окрестности при  $n_p^q \leq 6$  удается вычислить значение коррекции, используя только  $n_p^q, m_p^q, n_{\mathcal{E}}$  и в редких случаях просмотр на принадлежность  $q$ -ому звену одного элемента окрестности. Отметим, что звенья с числом элементов менее семи обычно составляют (75-80%) от общего числа звеньев цепей.

Значение коэффициента связности претендента  $e_j$  с элементами  $\mathcal{E}$ -окрестности вычисляется по формуле:

$$K_{\varepsilon j} = \sum_{q=1}^K \beta_q \cdot K_{\varepsilon j}^q$$

где  $\beta_q$  - весовая функция звена. В простейшем случае  $\beta_q = \beta(n_p^q, n^q)$ , где  $n^q$  - число элементов  $q$ -ого звена. Затем вычисляется максимальный коэффициент связности из всего множества претендентов  $E_i$ , т.е.

$$K_\alpha = \max_{j \in E_i} K_{\varepsilon j}$$

Претендент располагается в исследуемой  $i$ -ой позиции.

В качестве основного режима работы алгоритма выбрано вычисление коэффициента связности по  $\varepsilon_1$ -окрестности с возможностями вычисления коэффициента связности по  $\varepsilon_2$  и  $\varepsilon_3$ -окрестности, а также по всем ранее размещенным элементам.

Алгоритм А3 реализован на БЭСМ-6 на языке "ФОРТРАН". Общий объем программы составляет около  $2 \cdot 10^3$  операторов. Время счета программы с использованием ограничения на длину соединения источника с приемником для задачи с 72 элементами и общим числом цепей около 1300 не превышает 8 минут.

#### Итеративный алгоритм (А4)

Будем считать, что логическое звено расположено компактно, если все элементы-приемники расположены на расстоянии от источника, меньшем или равном  $R = R(n, T)$ . Где  $R$  - есть расстояние от источника до приемника, допускаемое для определенного типа цепи  $T$  и числа приемников  $N$ . Логические звенья дифференцируются по типам.

К первому типу относят звено, приемники и источник которого являются контактами микросхем, ко второму ти-

цу отнесем звено, источником которого служит контакт разъема, а следующим типом назовем звено, имеющее контакт разъема в качестве приемника.

Дифференциация по типам полезна для построения специализированных процедур, максимально эффективных для каждого типа.

Сначала определяется множество звеньев, не удовлетворяющих критерию компактности, т.е. лежащих за пределами  $R = R(n, T)$ . Далее в зависимости от типа звена, числа приемников, выходящих за пределы окружности радиуса  $R$ , строится процедура определения подмножества позиций, куда можно переставлять опасные приемники. Просматриваются возможные перестановки, вычисляется функция дефекта  $F(\lambda)$ . Если при какой-то перестановке  $F(\lambda) = 0$ , то она фиксируется, если для всех возможных перестановок  $F(\lambda) > 0$ , то увеличивается значение  $R$  и итерации повторяются. При этом уменьшается максимально-допустимая величина отвода от приемника, используемая в дальнейшем при трассировке.

Величина отвода характеризует близость топологии связи к соединениям типа звезда (величина отвода не ограничивается) или же к последовательному обходу приемников (величина отвода равна нулю).

Чем больше становится значение  $R$ , тем сильнее уменьшается величина отвода, пока она не становится равной нулю. Тогда получаем случай последовательного обхода приемников звена. В результате работы алго-

ритма можно получить или бездефектное размещение, или массивы звеньев с величинами допускаемых отводов.

На эффективность процедуры размещения по критерию компактности сильное влияние оказывает зависимость

$R=R(n, T)$ . При правильном задании стратегии выбора значения  $R$  алгоритм дает размещение, учитывающее значительно более тонкие нюансы, чем перечисленные ранее критерии.

Алгоритм А4 реализован на БЭСМ-6 на языке "ФОРТРАН". Объем программы составляет  $2,5 \cdot 10^3$  операторов. Время счета зависит от величины  $R$ . Например, при размещении 72 элементов с общим числом цепей 1300 при величине  $R = 150 - 10(n - 2)$  время счета составило 20 секунд. В результате определилось некомпактно расположенных 8 цепей при начальном числе некомпактно расположенных цепей равном 16. Программа предусматривает наличие запрещенных мест, фиксированных элементов и свободных позиций.

#### РАЗНЕСЕНИЕ СОЕДИНЕНИЙ ПО СЛОЯМ МПП

При проектировании многослойных печатных плат с попарным секционированием встают задачи:

1. Разнесения связей по парам слоев для более равномерного заполнения поля платы;
2. Разнесение уже оттрассированных ортогональных проводников по различным слоям.

Первая задача решается перед трассировкой с целью уменьшения количества неоттрассированных связей. Для ее решения используется алгоритм последовательного

заполнения пар слоев платы. Исходной информацией является список координат контактов связей всей платы.

Покажем принцип работы алгоритма на примере четырехслойной платы.

Каждой связи ставится в соответствие описывающий ее прямоугольник  $S^j$  — минимальный прямоугольник, в котором лежат все контакты связи. Выбираются контакты первой связи и заносятся в список контактов первой пары слоев, контакты второй связи — в список контактов для второй пары слоев. На полях первой и второй пары строятся соответствующие описывающие прямоугольники  $S_1$  и  $S_2$ . Предположим, что  $(i-1)$  связь уже отнесена к какой-либо паре слоев. Далее выбираются контакты следующей  $i$ -той связи и строится описывающих их прямоугольник  $S_i$ . Вычисляются значения дискретных функций  $\theta_i^1(x), \theta_i^2(x), \theta_i^1(y), \theta_i^2(y)$ .

$$\theta_i^j(x) = \begin{cases} \theta_{i-1}^j(x)+1 & \text{при } m_1 \leq x \leq m_2 \text{ где } m_1 \text{ и } m_2 \\ & \text{крайние точки проекции } S_i \\ & \text{на ось } X \\ \theta_{i-1}^j(x) & \text{при } 1 \leq x < m_1, m_2 < x \leq n, \\ & \text{где } n - \text{граница платы по } X \end{cases}$$

$$\theta_i^j(y) = \begin{cases} \theta_{i-1}^j(y)+1 & \text{при } k_1 \leq y \leq k_2 \text{ где } k_1 \text{ и } k_2 \\ & \text{крайние точки проекции } S_i \\ & \text{на ось } Y \\ \theta_{i-1}^j(y) & \text{при } 1 \leq y < k_1, k_2 < y \leq \ell, \\ & \text{где } \ell - \text{граница платы по } Y \end{cases}$$

Прямоугольник  $S_i$  заносится на поле той пары слоев (соответственно, контакты связи заносятся в список координат контактов той пары слоев), для которой достигается

$$\min_{j=1,2} \left\{ \lambda_1 \max_x \theta_i^j(x) + \lambda_2 \max_y \theta_i^j(y) \right\}$$

где  $\lambda_1$  и  $\lambda_2$  некоторые коэффициенты.

Если же эти минимумы равны для каждой пары слоев, то исследуется

$$\min_{j=1,2} \left\{ M_1 \int \theta_i^j(x) dx + M_2 \int \theta_i^j(y) dy \right\}$$

где  $M_1$  и  $M_2$  – некоторые коэффициенты, и т.д. до исчерпания всех цепей.

После этого проводится трассировка соединений сразу пары слоев, в результате которой выдается список горизонтальных и вертикальных проводников, соединяющих контакты цепей.

Вторая часть работы – размещение уже оттрасированных соединений по двум слоям таким образом, чтобы на каждом из слоев не было пересекающихся соединений и было бы минимальное число переходов со слоя на слой.

Реализовано 2 алгоритма решения этой задачи. Первый алгоритм реализует построение множеств, взаимно пересекающиеся проводники, которых заранее должны лежать на разных слоях. Множества эти строятся следующим образом:

1. Из списка проводников уже оттрасированных соединений выбирается первый пересеченный проводник, заносится в множество I, соответствующее первому слою;

2. Отыскиваются все проводники, пересекающие первый выбранный и заносятся в множество II, соответствующее второму слою.

Далее отыскиваются все проводники, отличные от уже занесенных в множество I и пересекающие проводники из множества II.

После построения таких множеств подсоединяются непересекающиеся проводники. Они заносятся на тот слой, на который уже занесен первый пересеченный проводник в цепи.

Во втором алгоритме предлагается последовательное разнесение проводников по слоям следующим образом: из списка проводников, соединяющих контакты связей последовательно выбираются проводники одной связи. Пересеченные вертикальные помещаются на один слой, пересеченные горизонтальные — на другой. Далее отыскиваются еще не разнесенные непересеченные проводники этой же связи, подсоединяются к первому найденному уже разнесенному проводнику и заносятся на тот же слой, на который занесен этот проводник. Если же в связи нет пересеченных проводников, то она целиком заносится на слой вертикальных проводников, если в ней вертикальный проводник имеет самую большую длину и на слой горизонтальных пересеченных проводников в противном случае.

#### ПРОГРАММА ТРАССИРОВКИ

Наиболее универсальными программами с точки зрения учета различных электрических и технологических ограничений являются программы основанные на волновом алгоритме. Ниже описывается программа, основанная на волновом алгоритме и ортогональном размещении проводников по парам слоев печатной платы. Основное внимание уделялось на

реализацию возможности настройки программы на широкий набор конструкций плат.

Входной информацией для программы трассировки соединений на плате является массив связей функциональной схемы и описание конструкции платы.

В массиве связей каждая связь описывается следующим образом.

Дескриптор связи содержит:

Номер связи;

Количество контактов связи, которые должны быть соединены между собой;

Номер, который показывает, какой ширины должны быть проведены проводники этой связи;

Характеристика, задающая характер топологии связи.

После дескриптора перечисляются контакты элементов, которые должны быть соединены между собой (задается номер элемента и номер контакта).

Заданный массив связей должен быть оттрасирован на паре слоев с переходами в любом свободном месте.

На плате задается дискретная решетка, то есть координаты на плате могут быть только целочисленными. После преобразования входного массива связей - перехода к координатам платы, получаем непересекающиеся множества точек с целочисленными координатами, которые нужно соединить внутри каждого множества.

На трассировку связей налагаются следующие требования:

1. Некоторые проводники должны иметь большую ширину, чем остальные (2-3 дискета);
2. Для того, чтобы искажения в линиях связи не превышали допустимой величины, должно быть предусмотрено последовательное соединение контактов связи;
3. Процент неоттрастированных соединений должен быть минимальным;
4. При выполнении предыдущих условий длина дерева, соединяющего контакты каждой связи, должна быть минимальной.

Ни один из известных алгоритмов не позволяет до конца решить таким образом поставленную задачу. Обычно задача трассировки разбивается на ряд подзадач, которые решаются последовательно. При решении данной задачи был выбран следующий путь.

Все связи разделяются на  $N$  массивов в соответствии с требованиями, предъявляемыми разработчиком к порядку трассирования связей. Вначале обрабатывается полностью первый массив, затем второй и т.д. Рассмотрим обработку одного из массивов. В каждой связи вначале проводится по одному проводнику. Затем список связей массива просматривается вторично. Если встречается связь, в которой содержится более двух контактов, то к уже построенному фрагменту подсоединяется третий контакт. Таким образом, после второго прохода по массиву связей полностью построены проводники связей, состоящих из двух и трех контактов. В

связях, содержащих более трех контактов, соединены 3 контакта, остальные контакты еще не подсоединенны. Если есть такие связи, то идем на новый проход по массиву связей и т.д. Максимальное число проходов по массиву связей равно ( $K-1$ ),

где  $K$  — максимальное количество контактов в связи. Ранее построенные проводники служат преградой для последующих. Поэтому часть контактов может оказаться неподсоединенными к другим контактам своей связи.

Для соединения двух контактов или подсоединения  $i$ -того контакта к ранее построенному фрагменту был выбран волновой алгоритм *Ли*, позволяющий строить кратчайшие соединения и учитывать некоторые дополнительные условия.

В оперативной памяти ЭВМ находится массив, отражающий конструкцию платы, расположение на ней контактов, проведенные проводники и т.д. В данной задаче на этом поле разрешается проведение взаимно ортогональных проводников.

Максимальный размер поля платы — 73 500 дискретов (300 x 245).

Были просчитаны платы:

1) 144 x 125 дискретов;

100 связей, каждая связь содержит в среднем 4 контакта;

время счета  $\approx$  3 мин.;

2) 122 x 11 дискрет,

39 связей, каждая связь содержит в среднем 4 контакта;  
время счета  $\approx 1$  мин.

## АНАЛИЗ ПАТЕНТНОЙ СИТУАЦИИ

Для сравнения принципов, положенных в основу вычислительной системы БЭСМ-10, с основными тенденциями развития многопроцессорных систем и многомашинных комплексов был проведен анализ патентной ситуации в области развития вычислительных средств.

Основное внимание было уделено патентной деятельности ведущих зарубежных фирм: IBM, Burroughs, CDC, GECO преимущественно в области создания вычислительных систем. Повышенный интерес к созданию вычислительных систем начинает проявляться с середины шестидесятых годов.

К этому времени ряд фирм, в том числе и ведущих в вычислительной технике, начинают патентовать материалы по вычислительным системам.

Имеются патенты, относящиеся и к более раннему времени, например, патент США № 3419849 фирмы Burroughs Corp. от 30.II.62, в котором была запатентована вычислительная система, построенная по модульному принципу. Набор модулей включал вычислительные модули, модули памяти, модули управления вводом-выводом. Работой системы управляла операционная система, хранящаяся в запоминающем устройстве, доступном для всей вычислительной системы.

Патенты по вычислительным системам фирмы Burroughs Corp., относящиеся к середине шестидесятых годов, свидетельствуют об уже достаточно глубокой проработке системных

вопросов.

Например, патент № 3401376 от 26.II.65г., в котором каждое устройство системы функционирует самостоятельно по специальным программам, а их совместная работа осуществляется с помощью центрального устройства управления, или патент № 34III39 от 26.II.65г., в котором патентуется мультипроцессорная модульная система, количество модулей памяти которой может быть доведено до 16 и она может включать до 16 вычислительных модулей и модулей управления вводом-выводом. Специальная система связи должна обеспечивать взаимную коммутацию любых модулей, входящих в систему. Существенно, что для каждого типа модулей, объединяемых в систему, характерна высокая степень внутреннего параллелизма: в вычислительных модулях применено глубокое совмещение выполнения операций, запоминающие модули допускают одновременное обращение к четырем последовательным ячейкам памяти. Концепция модульности распространяется и на структуру операционной системы, которая представляет собой набор программ, подразделяющихся по функциональному назначению на 4 группы. Динамическая взаимосвязь программ определяется нагрузкой на вычислительные средства.

Патентуют вычислительные системы и другие фирмы, причем это связано либо с необходимостью работы в реальном масштабе времени при решении конкретных задач, например, патент США № 36II300 фирмы Honeywell Information Systems Inc., заявленный 25.02.66, в котором описывается система, приме-

няемая при решении задачи управления посадкой самолетов на авианосец, либо для увеличения вычислительной мощности при мультипрограммной работе, как это имеет место в пат.США № 3566357 (RCA Corp.) от 5.07.66г.

Управление работой таких систем осуществляется либо главной ЦВМ, либо специальным устройством управления, которые осуществляют распределение приоритетом, распределение задач и ресурсов, выполняемых программно-аппаратным способом, а также анализируют причины прерывания и обеспечивают переключение вычислительных средств на выполнение другой задачи.

Наибольшее количество патентований наблюдается в это время у Фирмы General Electric Co, например патенты США № 3475730, 3479647, 3504344, 3510844, заявленные в мае, июне 1966г. В этих патентах, наряду с внутренней структурой многопроцессорной системы, управляемой централизованно от центрального устройства управления, предлагаются дополнительные устройства межсистемной связи, обеспечивающие обмен информацией с другой вычислительной системой.

Кроме того, предлагаются различные варианты внутренней организации вычислительной системы: рассматриваются вопросы управления памятью, обеспечение совмещенного ввода-вывода, а также обеспечение возможности подключения с помощью каналов связи внешних абонентов, причем каждому каналу связи поставлен в соответствие определенный массив памяти системы.

Для обеспечения работы в реальном масштабе времени предусмотрены специальные команды установления прямой (исключительной) связи, выдаваемые тем или иным процессором системы, по которым на время, указанное в команде, данный процессор соединяется с указанными в этой же команде блоками памяти (см.пат.США № 3505651 от 28.2.67).

К 1967 г. в области патентования материалов по вычислительным системам повышается активность фирмы IBM.

Патентуется многопроцессорная вычислительная система, состоящая из множества отдельных процессоров с автономными устройствами управления и общим центральным запоминающим устройством, доступ к которому обеспечивается для всех процессоров системы. Каждое автономное устройство управления процессором имеет дополнительную аппаратуру для выбора заданий процессору из множества очередей заданий, имеющихся в центральном запоминающем устройстве (см.пат.США № 3496551 от 13.07.67г.).

Предусмотрена возможность работы такого ЗУ с несколькими разнотипными устройствами обработки и источниками информации за счет параллельной работы в ряде случаев на несколько внешних устройств. Это обеспечивается тем, что обобщенный адрес выдаваемого ЗУ или поступающего в ЗУ кода может относиться одновременно к нескольким устройствам, воспринимающим различные зоны адреса и соответствующие участки выходного информационного кода. Адресные регистры работают в комплексе

с общими узлами группирования информационных слов, максимальная емкость которых соответствует формату выходного слова ЗУ (см.пат. США № 3500337 от 27.09.67).

Для обеспечения автоматического выбора свободного устройства системы, наилучшим образом отвечающего требованиям выполняемой задачи в пат.США № 3593300 (IBM) от 13.II.67 патентуется "узел выбора".

Возможности устройства характеризуются двоичным вектором, каждая компонента которого соответствует определенному свойству, например, для процессора такими свойствами являются наличие арифметических операций с плавающей запятой, операций десятичной арифметики и т.д. "Узел выбора" придан каждому устройству. Для выбора устройства вектор, характеризующий потребности задачи, сравнивается с векторами возможностей устройств в "узлах выбора" этих устройств. Для выполнения задачи выбирается устройство, имеющее все требуемые свойства; если таких устройств несколько, то среди них выбирается устройство с наименьшим индексом мощности, под которым понимается сумма весов свойств устройства, указанных в векторе его возможностей.

Патенты фирмы Burroughs Corp. 1967г. продолжают традиции развития быстродействующих модульных вычислительных систем обработки данных. Вычислительные модули могут работать параллельно, обмениваясь информацией с модулями памяти. (см.пат.США № 3492654 от 29.05.67). В системе применяются быстродействующие модули памяти. Чтобы полностью использовать

это быстродействие входящие в систему вычислительные модули и модули управления вводом-выводом разделены на блоки, способные работать одновременно.

В системы включают устройства, выполняющие функции сопряжения вычислительной системы с линиями связи. (См. пат. США № 3514758 от 27.03.67). Эти устройства сопряжения рассчитаны на обслуживание большого количества устройств ввода-вывода, которые могут различаться режимом обмена и форматом информации.

Патентуется также система для параллельной обработки данных (см. пат. США № 3537074 от 20.12.67), состоящая из множества блоков управления, каждый из которых управляет набором обрабатывающих элементов, выполняющих арифметические и логические операции над данными. С каждым обрабатывающим элементом связана память, которая одновременно входит в состав основной памяти блока управления. Система связана с внешним миром посредством управляющей ЦВМ, которая сама является крупной системой обработки данных. Система включает также подсистему ввода-вывода, связывающую все ее элементы.

Другие фирмы используют вычислительные системы при работе в режиме, близком к режиму реального времени, когда необходимо обеспечить достаточно высокую надежность работы. (См. пат. США № 3651480 от 24.11.67 фирмы Bell Telephone Labs, Inc. "Система для управления телефонной сетью").

Дальнейшим развитием этой системы, по-видимому является пат.США № 3587058 от 4.06.69 той же фирмы, в котором патентуется система переключения потока информационных массивов для хранения или взаимных пересылок в системе обработки данных, имеющей в своем составе вспомогательный буферный процессор, подключенный к центральному процессору. Этот буферный процессор используется для соединения с каналами связи, имеющими различные скорости передачи данных и различные форматы сообщений, а также для соединения с накопителями данных на магнитных дисках и магнитных лентах. Каналы связи подключены к автономно работающим канальными буферным разделителям, осуществляющим преобразование кодов при вводе и выводе данных, под управлением буферного процессора. Буферные разделители осуществляют также выработку сигналов состояния канала, а также идентификации данных.

Эта же фирма в пат.США № 3641505 от 25.06.69 предусматривает возможность формировать в системе независимо действующие подсистемы при помощи тумблеров и программно вырабатываемых сигналов. Подсистема может содержать произвольное число процессоров, ЗУ и т.д.

Продолжает патентование материалов по вычислительным системам и фирма Honeywell Inc.- патенты США № 3648252 от 3.II.69 и № 3631405 от 12.II.69.

Еще более повышается активность фирмы IBM. в области развития мультипроцессорных систем. Патентуются различные способы организации работы таких систем. Основное внимание

по-прежнему уделяется созданию систем, состоящих из вычислительных машин и центральной памяти, которая управляет обменами в такой системе. Предлагается производить такие обмены по специальной постоянной программе с применением централизованной синхронизации (см.пат.США № 3551892 от 15.01.69).

Наряду с обычной мультипроцессорной обработкой информации, когда каждый процессор выполняет свою последовательность команд, в системах предусмотрено оборудование для работы в так называемом "векторном" режиме, когда все или часть процессоров работают синхронно, выполняя в каждый момент времени одну и ту же команду с разными данными (см.пат. США № 3560934 от 10.06.69). Программа работы процессоров, вовлеченных в "векторный" режим, задается ведущим процессором: ведущий процессор извлекает команды из памяти и передает их в устройства управления работающих совместно с ним процессоров, в этих процессорах блокируется работа механизма выборки команд, работа же механизмов индексации и выборки данных из памяти протекает обычным образом. В системе может быть образовано несколько групп процессоров, работающих в векторном режиме.

Уделяется внимание и созданию универсальных вычислительных систем, не ориентированных на решение какого-то определенного класса задач и не накладывающих ограничений на используемые алгоритмические языки (см.пат.США № 3634830 от 13.06.69). Такая система объединяет ряд взаимозаменяемых

ЦВМ, каждая из которых содержит автономную память и процессор и обрабатывает информацию, поступающую с дистанционных пультов пользователей. Одна ЦВМ служит управляющим процессором-супервизором сбора, распределения и обмена информацией между ЦВМ системы и абонентами, причем каждому пульту пользователя в системе жестко отводится свой блок внешней памяти на дисках. Система работает в режиме распределения времени между абонентами.

На основании вышеизложенного может быть построена гистограмма активности ведущих фирм США в области патентования материалов по вычислительным системам за период 1965-1970 год, из которой можно сделать заключение о гегемонии в этой области к концу данного периода двух фирм: IBM и Burroughs Corp. (рис. I06).

Имеются патенты и других фирм, но их удельный вес значительно меньше.

Наряду с патентами на вычислительные системы за указанный период получено много патентов по структуре вычислительных устройств, например, патенты США № 3477063, 3490005, 3508038, 3566098, 3584206, 3614404, 3648246 по вопросам, связанным с управлением памятью, с организацией виртуальной памяти, например, патенты США № 3611315, 3588839, 3541528, 3533075, 3475730, 3609700, 3596257, по вопросам, связанным с организацией сверхоперативной буферной памяти и буферных запоминающих устройств, например,

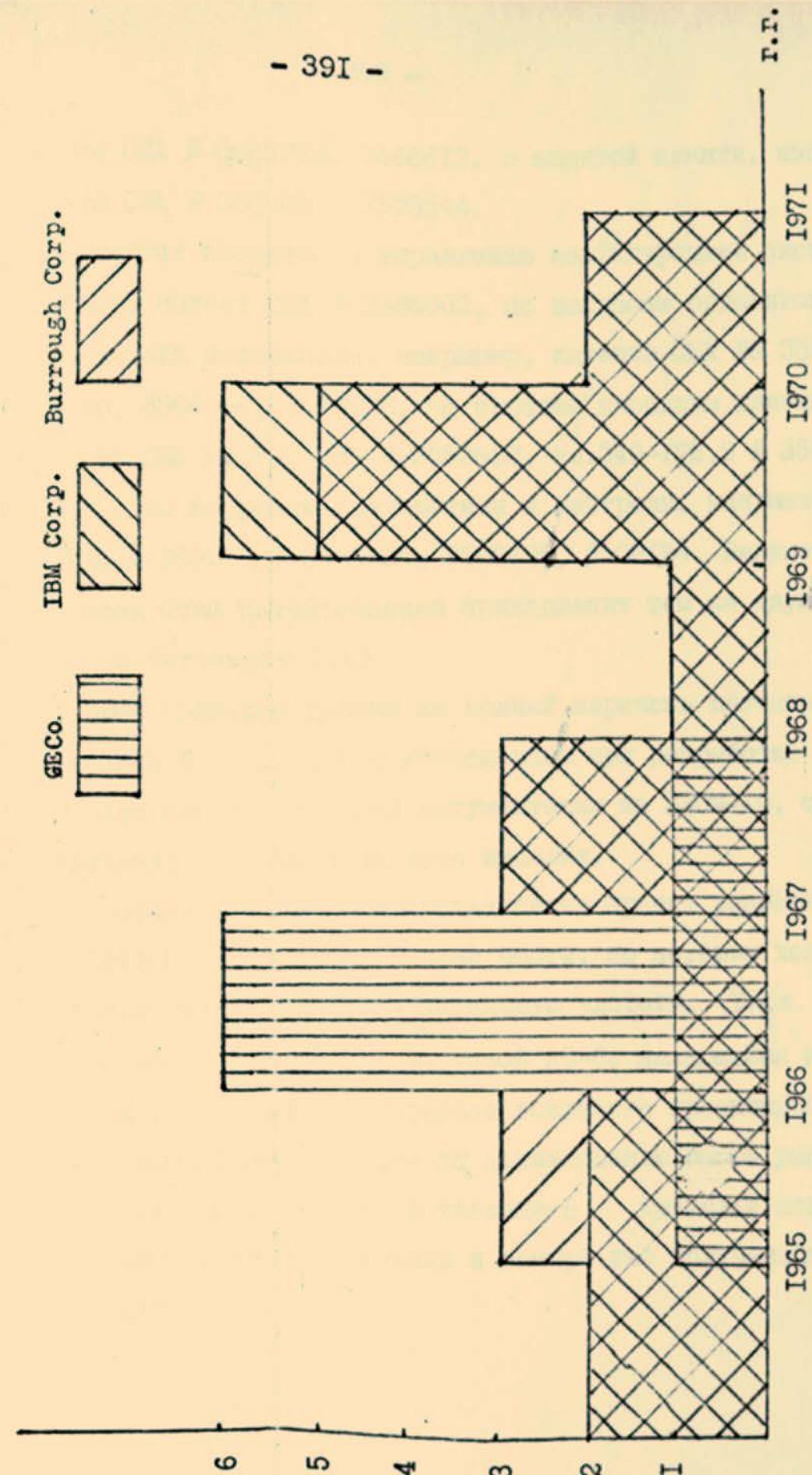


Рис.106. Гистограмма патентной активности ведущих зарубежных фирм в области патентования вычислительных систем

патенты США № 3601812, 3466613, с защитой памяти, например, патенты США № 3609697, 3576544.

Имеются патенты по управлению конфигурацией системы, например, патент США № 3386082, по вопросам организации ввода-вывода информации, например, патенты США №№ 3514758, 3406380, 3564509, 3475729, по системе передачи данных, например, пат.США кл.307-259 № 3582682, кл.340-172.5 № 3553655, 3573853, по вопросам диагностики и контроля, например, пат.США № 3609692, 3560924, 3508195, 3508194. Ведущее место и по всем этим патентованиям принадлежит тем же двум фирмам - IBM и Burroughs Corp.

Выше приведен далеко не полный перечень патентов, которые должны быть детально исследованы при разработке устройств и выпуске конструкторской документации на БЭСМ-10, с целью обеспечения патентной чистоты изделия.

В связи с этим ниже предлагается проект разбиения устройств БЭСМ-10 на составные части, по которым должно быть проведено исследование на патентную чистоту. (Табл. 7).

Во время работы над системой АС-6, являющейся прототипом БЭСМ-10, а также в процессе эскизного проектирования БЭСМ-10 проводилась работа по патентованию вновь разрабатываемых устройств и схем. В таблице 8 приведен список поданных заявок на изобретения и номера уже полученных авторских свидетельств.

Таблица 7

№	Наименование составных частей объекта или его элементов	Классификационные индексы			
		МКИ	Советско-германская классификация	США	
I	2	3	4	5	
1.	Вычислительная система БЭСМ-10	G 06 f 15/16	42 m <sup>3</sup> 1516	340-I72.5	
	Арифметическое устройство универсаль- ных операций				
	2. Устройство умножения	G 06 f 7/39, 52	42 m <sup>3</sup> 739, 752	235-I56, I60	
	3. Устройство деления	G 06 f 7/39, 52	42 m <sup>3</sup> 739, 752	235-I60, I64	
	4. Устройство сложения, вычитания с плавающей запятой	G 06 f 7/38	42 m <sup>3</sup> 738	235-I68	
	5. Устройство выполнения прочих операций	G 06 f 5/02	42 m <sup>3</sup> 502	235-I55	
	6. Прямодрессемые и буферные регистры	G 06 f 7/00	42 m <sup>3</sup> 700	340-I72.5	
	7. Буфер арифметических команд	G 06 f 7/00	42 m <sup>3</sup> 700	340-I72.5	
	8. Система связи (обменные регистры и кодовые шины)	G 06 f 13/00	42 m <sup>3</sup> 1300	340-I72.5	
		G 11c 9/00	21 a' 900	235-I75	

1	2	3	4	5
<b>Арифметическое устройство специальных операций</b>				
<b>9.</b> Блок вызова operandов	G 06 ♂	7/38	42 м <sup>3</sup> 738	340-І72.5
<b>10.</b> Буфер команд	G 06 ♂	7/00	42 м <sup>3</sup> 700	340-І72.5
<b>11. Входные и выходные регистры</b>	G 06 ♂	7/00	42 м <sup>3</sup> 700	340-І72.5
<b>12. Процессор операций над последовательностями (байтов, разрядов)</b>	G 06 ♂	5/00	42 м <sup>3</sup> 500	340-І72.5
<b>13. Процессор десятичных операций</b>	G 06 ♂	7/28, 38	42 м <sup>3</sup> 728, 738	235-І70
	G 06 ♂	7/38, 50	42 м <sup>3</sup> 738, 750	235-І70
<b>Устройство управления центрального процессора</b>				
<b>14. Блок обработки команд</b>	G 06 ♂	7/00	42 м <sup>3</sup> 700	340-І72.5
<b>15. Сумматор адреса со сквозным переносом</b>	G 06 ♂	7/38, 50	42 м <sup>3</sup> 738, 750	235-І70
<b>16. Схема деления на фиксированное число</b>	G 06 ♂	7/52	42 м <sup>3</sup> 752	235-І60
<b>17. Блок преобразования виртуального адреса в физический</b>	G 06 ♂	9/00, 10, 20	42 м <sup>3</sup> 900, 10, 20	340-І72.5
<b>18. Таблицы ассоциативных регистров</b>	G 06 ♂	13/06	42 м <sup>3</sup> 13 06	340-І72.5



	1	2	3	4	5
29. Адаптер канала 1-го уровня	6 06 ♂	3/00	42 м³	3'00	340-I72.5
30. Адаптер канала 2-го уровня	6 06 ♂	7/00	42 м³	7'00	340-I72.5
31. Буфер директив					

Коммутатор канала I-го уровня

- |                               |                                |                |                          |           |
|-------------------------------|--------------------------------|----------------|--------------------------|-----------|
| 32.                           | Блок приема сообщения          | 6 06 ₽ 5/00    | 42 м <sup>3</sup> 500    | 340-І72.5 |
| 33.                           | Блок выдачи сообщения          | 6 06 ₽ 13/06   | 42 м <sup>3</sup> 1300   |           |
| 34.                           | Блок управления коммутацией    | 6 06 ₽ 9/00,20 | 42 м <sup>3</sup> 900,20 | 340-І72.5 |
| 35.                           | Блок работы с диспетчером      | 6 06 ₽ 11/00   | 42 м <sup>3</sup> 1100   | 340-І46.1 |
| 36.                           | Блок контроля передачи         | 6 06 ₽ 11/00   | 42 м <sup>3</sup> 1100   | 340-І46.1 |
| Коммутатор канала 2-го уровня |                                |                |                          |           |
| 37.                           | Блок приема сообщений          | 6 06 ₽ 5/00    | 42 м <sup>3</sup> 500    | 340-І72.5 |
| 38.                           | Блок выдачи сообщений          | 6 06 ₽ 9/00,20 | 42 м <sup>3</sup> 900,20 | 340-І72.5 |
| 39.                           | Блок управления коммутацией    | 6 06 ₽ 11/00   | 42 м <sup>3</sup> 1100   | 340-І46.1 |
| 40.                           | Блок диспетчерского управления | 6 06 ₽ 11/00   | 42 м <sup>3</sup> 1100   | 340-І46.1 |
| 41.                           | Блок контроля передачи         | 6 06 ₽ 11/00   | 42 м <sup>3</sup> 1100   | 340-І46.1 |

1 2 3 4 5

Мультиплексор преобразования  
сопряжения

42.	Интерфейс ввода-вывода	6 06 ₽	3/00	42 м <sup>3</sup> 3'00	340-І72.5
43.	Адаптер канала 2-го уровня	6 06 ₽	9/00	42 м <sup>3</sup> 9'00	
44.	Блок накопления данных	6 06 ₽	3/00	42 м <sup>3</sup> 3'00	340-І72.5
45.	Блок управляющей памяти	6 06 ₽	5/00	42 м <sup>3</sup> 5'00	340-І72.5
		6 06 ₽	9/16	42 м <sup>3</sup> 9'16	340-І72.5

Заявки по АС-6 и БЭСМ-10

Таблица 8

№ пп	Ф.И.О.	Наименование изобретения	Дата сдачи заявки	Дата приоритета заявки и №	Выдачное решение или авторское свидетельство
1.	Жироев В.Ф., Смирнов В.И. Соколов А.А.	Устройство декодирования кода Хэмминга	27.Х.70 г. Соколов А.А.	№ 1486619/18-24 а.с. 383050	398-
2.	Жуковский В.А., Зак Л.А., Знаменский Ю.Н., Ли В.Л., Митропольский Ю.И., Соколов А.А.	Устройство для управления схемой фиксации динамиче- ского приоритета информа- ции	31.ХII.70 г. Соколов А.А.	№ 1621874/18-24 а.с. 359653	4.І.71 г.
3.	Зак Л.А., Знаменский Ю.Н., Новиццев В.С., Жуковский В.А., Соколов А.А.	Устройство для фиксации динамического приоритета	31.ХII.70 г. В.А., Соколов А.А.	№ 1614581/18-24 а.с. 386396	12.І.71 г.

№	Ф.И.О.	Наименование изобретения	Дата сдачи заявки	Дата приоритета заявки и №	Выдачное решение или авт. свидетельство
4.	Новиценцев В.С., Жуковский В.А., Соколов А.А., Митропольский Ю.И.	Устройство формирования синхросигнала	I, III, 7 I г.	12, III, 7 I г. № 1628136/26-9	з.с. 572725
5.	Мельников В.А., Соколов А.А., Смирнов В.И., Тяпкин М.В., Зак Н.А., Митропольский Ю.И.	Информационно-вычислительная система	18. VI. 7 I г.	28.VI.71 г. № 1674090/18-24	з.с. 402871
	Знаменский Ю.Н., Жуковский В.А., Жирков В.Ф., Ли В.Л., Новизенко В.С., Королев Л.Н., Иванников В.П., Томилин А.Н., Степанов Г.В., Журавлев А.И., Авзев А.В.				

№	Ф.И.О.	Наименование изобретения	Дата сдачи заявки	Дата приоритета заявки и №	Выдачное решение или евт. свид.
6.	Абазов Н.Ф., Барбашкин В.М., Жуковский В.А., Соколов А.А., Ульянов В.В.	Формирователь прямогоуголь- ных сигналов	16. IX. 71 г.	21. IX. 71 г. № 1698946/26-9	выдачное решение от 5. III. 73 г.
7.	Андреев С.С., Жирох В.Ф., Командровский В.Г., Ли В.Л., Смирнов В.И., Шнитман В.З.	Буферное ЗУ с ассоциативной выборкой	30. XII. 71 г.	3. I. 72 г. № 1732460/18-24	400- выдачное решение от 20. III. 71
8.	Азазев А.В., Ли В.Л., Тяпкин М.В.	Арифметическое устройство вычислительной машины		15. V. 72 г. № 1784042/18-24	выдачное решение от 20. III. 71
9.	Азазев А.В., Ли В.Л., Тяпкин М.В.	Устройство управления вычислительной машины	3. XI. 72 г.	9. XI. 72 г. № 1845312/18-24	выдачное решение от II. X. 73

№	Ф.И.О.	Наименование изобретения	Дата сдачи заявки	Дата приоритета заявки и № решения	Выдачное или авт. св.
10.	Аваев А.В., Ли В.Л., Тяпкин М.В.	Устройство управления вычислительной машиной	3.XI.72 г.	9. XI.73 г. № 1845316/18-24	Выдачное решение от 24/XII-73г.
11.	Щербаков О.К., Гавриш Ю.С.	Устройство контроля и управления электропитанием ЭВМ	-	30. III.73 г. № 1902775/18-24	401 -
12.	Марычев Л.Г., Сердюкова О.Н., Шильяков Е.И., Жирох В.Ф.	Устройство организации буферных очередей	-	-	31.IX.73 г. № 1956678/18-24
13.	Мельников В.А., Соколов А.А., Тяпкин М.В., Жуковский В.А., Смирнов В.И., Ульянов В.В., Цибульников М.Б.	Универсальный логический элемент	12.II.73 г.	15.II.73 г.	выдачное решение № 1882651 от 25.XI.73 г.