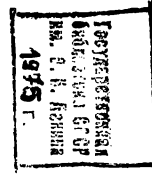


ЕС-2020

ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ

**ТЕХНИЧЕСКОЕ ОПИСАНИЕ
Е13.055.001 ТО**



88048

СОДЕРЖАНИЕ

	Стр.
I. ВВЕДЕНИЕ	3
2. НАЗНАЧЕНИЕ ПРОЦЕССОРА	3
3. ТЕХНИЧЕСКИЕ ДАННЫЕ И ХАРАКТЕРИСТИКИ	4
3.1. Информационные форматы	4
3.2. Адресация	5
3.3. Виды операндов	5
3.3.1. Арифметика с фиксированной запятой	5
3.3.2. Десятичная арифметика	6
3.3.3. Арифметика с плавающей запятой	7
3.3.4. Логические операции	8
3.4. Представление команд	9
3.4.1. Формат команд	9
3.4.2. Формирование адреса	10
3.4.3. Последовательное выполнение команд	11
3.4.4. Переходы	11
3.4.5. Слово состояния программы	12
3.4.6. Прерывания	12
3.5. Оперативная память	16
3.6. Защита памяти	16
3.7. Таймер	17
3.8. Средства прямого управления	17
3.9. Типы и характеристики каналов	17
3.9.1. Мультиплексный канал	17
3.9.2. Селекторный канал	18
3.9.3. Одновременная работа каналов	19
3.9.4. Выполнение операций ввода-вывода	19
3.9.5. Канальная программа	20
3.9.6. Окончание операций ввода-вывода	22
3.9.7. Состояние каналов и внешних устройств	22
3.9.8. Прерывание по вводу-выводу	25
3.10. Пульт управления	26
3.10.1. Приведение ЭВМ в исходное состояние	26
3.10.2. Запоминание и индикация	26
3.10.3. Первоначальная загрузка программы	26
3.11. Устройство питания	27
3.12. Конструкция процессора (ЕС-2020)	28
4. УСТРОЙСТВО И РАБОТА ПРОЦЕССОРА	29
ПРИЛОЖЕНИЕ 1	32
ПРИЛОЖЕНИЕ 2	33
ПРИЛОЖЕНИЕ 3	33

1. ВВЕДЕНИЕ

Материал настоящего описания дает общее представление о функциональных характеристиках, принципе действия и составных частях процессора ЕС-2020, центральной части электронной вычислительной машины ЕС-1020.

Поскольку процессор ЕС-2020 является ядром модели, определяющим ее системные свойства, то для изучения и правильной эксплуатации данного изделия необходимы знания общей структуры ЕС ЭВМ, принципов выполнения операций, организации системного взаимодействия программной и аппаратной логики Единой системы ЭВМ.

Полная информация, относящаяся к составным частям процессора, содержится в ряде документов, список которых приводится ниже:

- ЕС-2020 Вычислительное устройство. Техническое описание EI3.055.001 T01;
- ЕС-2020 Мультиплексный канал. Техническое описание EI3.055.001 T02;
- ЕС-2020 Селекторный канал. Техническое описание EI3.055.001 T03;
- ЕС-2020 Методика выполнения операций. Техническое описание EI3.055.001 T04;
- ЕС-2020 Постоянная память. Техническое описание EI3.055.001 T05;
- ЕС-2020 Блок защиты памяти. Техническое описание EI3.055.001 T06;
- ЕС-2020 Пульт управления. Техническое описание EI3.055.001 T07;
- ЕС-2020 Инструкция по эксплуатации EI3.055.001 ИЭ;
- ЕС-3220 Оперативная память. Техническое описание EI3.061.013 T0;
- ЕС-0820 Техническое описание и инструкция по эксплуатации EI2.087.031 T0;
- БУП-1 Техническое описание и инструкция по эксплуатации EI2.390.019 T0.

Список принятых в настоящем документе сокращений и обозначений приведен в Приложении I.

2. НАЗНАЧЕНИЕ ПРОЦЕССОРА

Процессор электронной вычислительной машины ЕС-1020 предназначен для выполнения процессов обработки информации, которые состоят из восприятия информации с внешних носителей (перфолент, перфокарт, магнитных лент, магнитных дисков), запоминания ее в быстродействующей внутренней памяти, обработки введенной информации и выводе результатов на внешний носитель или итоговый формуляр.

Процессор является ядром систем обработки данных, которые могут быть образованы на базе переменного объема оперативной памяти и номенклатуры внешних устройств ЕС ЭВМ в виде различных вариантов исполнения ЭВМ ЕС-1020.

Процессор может быть частью ЭВМ ЕС-1020 или частью многопроцессорного комплекса, предназначенного для создания системы повышенной надежности.

3. ТЕХНИЧЕСКИЕ ДАННЫЕ И ХАРАКТЕРИСТИКИ

Процессор ЕС-2020 (рис.1) представляет собой комплекс устройств в следующем составе: вычислитель ЕС-2420 EI3.055.00I, состоящий из:

- вычислительного устройства;
- мультиплексного канала;
- селекторного канала I;
- селекторного канала 2;
- пульта управления;

оперативная память ЕС-3220 EI3.06I.0I3;
устройство питания ЕС-0820 EI2.087.03I.

Через шины стандартного интерфейса к процессору подключаются устройства управления внешними устройствами (УВУ), которые соединены с внешними устройствами (ВУ).

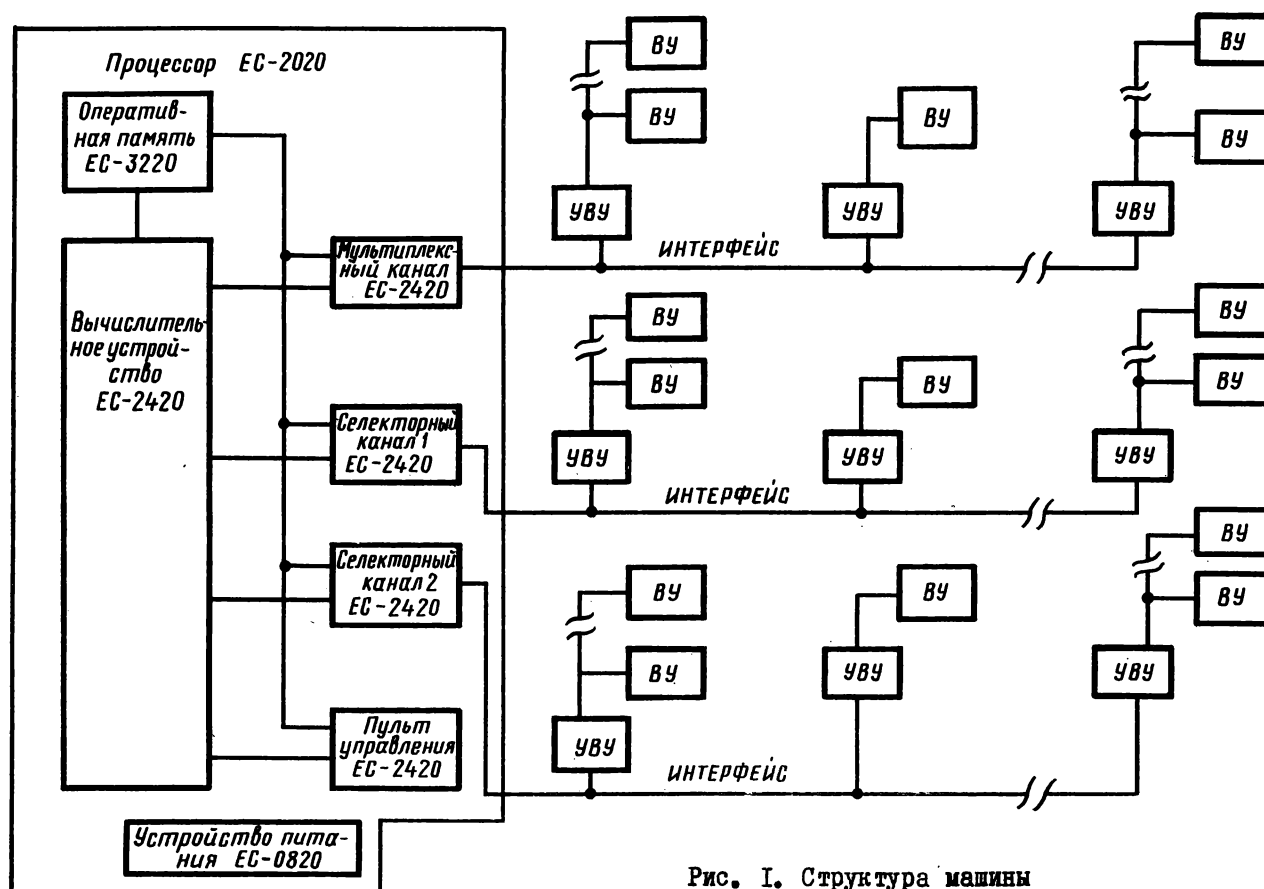


Рис. 1. Структура машины

3.1. Информационные форматы.

При описании структуры информации применяются следующие понятия:

бит - двоичная цифра, которая может принимать значение нуля или единицы;

байт - восьмибитный блок информации. Является основным образующим блоком всех форматов. Каждому байту присвоен девятый бит, называемый битом четности, который дополняет сумму по модулю два содержимого байта до нечетной;

полуслово - состоит из двух последовательно расположенных байтов. Адрес полуслова должен быть кратен 2;

слово - состоит из четырех последовательно расположенных байтов. Адрес слова должен быть кратен 4;

двойное слово - состоит из восьми последовательно расположенных байтов. Адрес двойного слова должен быть кратен 8.

Наименьшей адресуемой единицей информации является байт. Биты, входящие в состав байта, нумеруются последовательно слева направо, начиная с нуля. В качестве внутреннего кода используется двоичный код для обработки информации (ДКОИ) или восьмибитный код обмена информацией (КОИ-8).

В качестве операндов могут использоваться поля фиксированной или переменной длины. Длина поля задается в команде. Операнды переменной длины отличаются по длине на целое число байтов.

3.2. Адресация.

Места байтов в памяти последовательно нумеруются, начиная с нуля. Каждый номер считается адресом соответствующего байта. Группы байтов в памяти адресуются самым левым байтом этой группы. Число байтов в группе определяется форматом информации или задается явно частью команды.

Для адресации памяти используется 18-битный адрес, что позволяет адресовать 262144 ячейки памяти, включая ячейки, резервированные для специальных целей. Обращение к информации, выходящей за пределы памяти, доступной на данной конкретной установке, классифицируется как ошибка и вызывает программное прерывание.

3.3. Виды операндов.

Арифметические и логические операции производятся над операндами, которые представляют собой двоичные числа фиксированной длины (целые и дробные с плавающей запятой), десятичные целые числа переменной длины и логическую информацию фиксированной и переменной длины.

3.3.1. Арифметика с фиксированной запятой.

Основной арифметический операнд представляет собой 32-битное слово с фиксированной запятой. В большинстве операций для увеличения производительности или для улучшения использования памяти могут быть заданы 16-битные операнды длиной в полуслово (рис.2). Для сохранения точности некоторые произведения и все делимые имеют длину 64 бита.

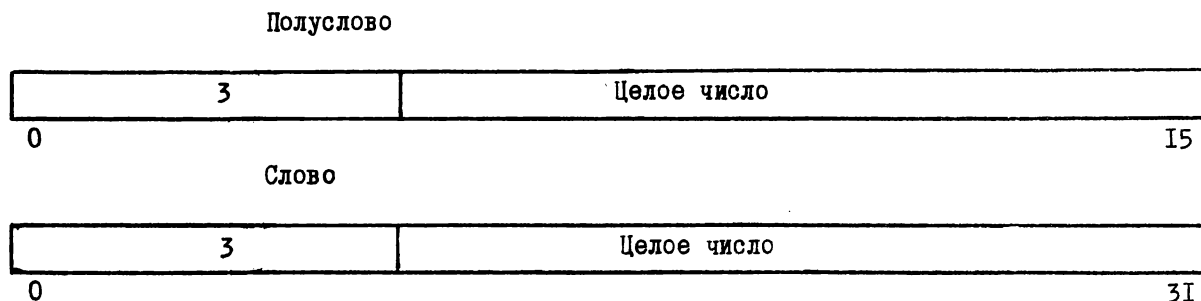


Рис. 2. Форматы чисел с фиксированной запятой

Поскольку 32-битный размер слова позволяет поместить в этом слове 24-битный адрес, операции с фиксированной запятой могут быть использованы не только для арифметических операций над целыми числами, но и для арифметических операций над адресами (адресной арифметики). Такое комбинированное использование дает экономию средств и позволяет употребить при вычислении адреса весь набор команд для операций с фиксированной запятой и некоторых логических операций. Так, например, над адресными компонентами возможны операции умножения, сдвига и логические манипуляции.

Операции сложения, вычитания, умножения, деления и сравнения выполняются над операндами, один из которых находится в регистре, а другой или также находится в регистре, или выбирается из памяти. Операции с повышенной точностью упрощаются благодаря представлению чисел в дополнительном коде и возможности переноса из одного слова в другое. Слово, находящееся в одном регистре, или двойное слово, находящееся в двух смежных регистрах, может быть сдвинуто влево

или вправо. Две команды преобразования: ПРЕОБРАЗОВАТЬ В ДВОИЧНУЮ и ПРЕОБРАЗОВАТЬ В ДЕСЯТИЧНУЮ - позволяют осуществить перевод из десятичной системы счисления в двоичную и наоборот без использования таблиц. Команды групповой загрузки регистров и группового запоминания облегчают переключение подпрограмм.

3.3.2. Десятичная арифметика.

Операции над десятичными числами предусмотрены для тех видов обработки данных, где в промежутке между вводом исходных данных и выводом результатов выполняется небольшой объем вычислений. Такой вид обработки часто встречается при экономических применениях, особенно при использовании проблемно-ориентированных языков. Из-за ограниченного числа арифметических операций, выполняемых над каждым блоком данных, перевод их из десятичной системы в двоичную и затем опять в десятичную не оправдан и использование регистров для промежуточных результатов не дает преимуществ из-за частых обращений к памяти. Поэтому вводятся операции над десятичными числами (десятичная арифметика), в которых оба операнда и результаты располагаются в памяти. Десятичная арифметика включает сложение, вычитание, умножение, деление и сравнение.

Десятичные числа трактуются как целые числа со знаком, имеющие формат переменной длины (от 1 до 16 байтов). Отрицательные числа представляются в прямом коде.

Таблица 1

Двоичные коды цифр и знаков

Цифра	Код	Знак	Код
0	0000	+	1010
1	0001	-	1011
2	0010	+	1100
3	0011	-	1101
4	0100	+	1110
5	0101	+	1111
6	0110		
7	0111		
8	1000		
9	1001		

Десятичные цифры 0-9 представляются в четырехбитной двоично-кодированной десятичной форме от 0000 до 1001 соответственно (табл. 1). Коды 1010-1111 не могут использоваться в качестве цифр и резервируются для кодов знаков: 1011 и 1101 представляют минус; другие четыре кода интерпретируются как плюс. Какие коды знаков генерируются в десятичной арифметике, зависит от того, какой набор символов предпочитается. При выборе двоичного кода для обработки информации (ДКОИ) этими кодами являются 1100 и 1101. При выборе 8-битного кода обмена информацией (КОИ-8) для знаков используются коды 1010 и 1011. Выбор того или иного кода определяется битом режима в слове состояния программы (ССП).

Десятичные операнды и результаты представляются четырехбитными двоично-кодированными десятичными цифрами, по две в каждом байте. Операнды имеют переменную длину, код знака размещается в четырех правых битах младшего байта. Поля операндов могут быть помещены в памяти, начиная с границы любого байта, и могут иметь длину до 31-й цифры и знак. Операнды, участвующие в операции, могут отличаться по длине. Упаковка цифр в байте (рис. 3) и расположение в памяти полей переменной длины приводит к эффективному использованию памяти, уменьшению времени выполнения арифметических операций и к ускорению процесса обмена информацией между памятью и файлами.

Старший байт

Младший байт

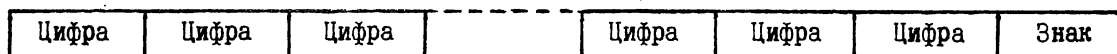


Рис. 3. Формат упакованных десятичных чисел

Десятичные цифры могут быть также представлены в зонированном формате как подмножество набора восьмибитных алфавитно-цифровых символов (рис.4). Такое представление требуется для

Старший байт

Младший байт

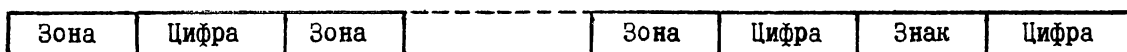


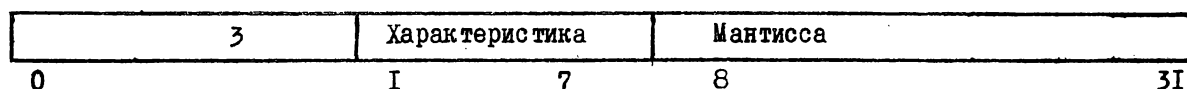
Рис. 4. Формат зонированный десятичных чисел

ВУ, работающих с набором символов. Знак числа, имеющего зонированный формат, размещается в четырех битах младшего байта. Зонированный формат не используется в операциях десятичной арифметики. Имеются команды для перевода десятичных чисел из зонированного формата в упакованный формат.

3.3.3. Арифметика с плавающей запятой.

Числа с плавающей запятой могут быть двух форматов фиксированной длины: короткого и длинного. Эти форматы отличаются только длиной мантиссы (рис.5).

Короткое число с плавающей запятой (слово)



Длинное число с плавающей запятой (двойное слово)

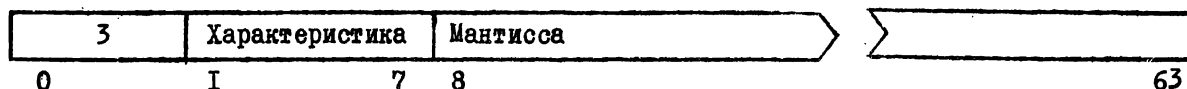


Рис. 5. Форматы короткого и длинного чисел с плавающей запятой

Операнды могут иметь длину 32 или 64 бита. Короткий формат, обеспечивая точность в семь десятичных цифр, позволяет разместить в памяти максимальное число операндов и выполнять операции с наименьшими затратами времени. Длинный формат, используемый для получения более высокой точности, дает точность до 17 десятичных цифр и удовлетворяет большинству требований к арифметическим операциям удвоенной точности.

Длины операндов, будучи степенями двух, позволяют добиться максимальной эффективности при использовании двоичной адресации. В арифметических операциях с плавающей запятой легко осуществляется переход от одного формата к другому.

Мантисса числа с плавающей запятой выражается шестнадцатичными (с основанием 16) цифрами, каждая из которых использует четыре бита и может принимать значения от 0 до 15. В коротком формате мантисса состоит из шести шестнадцатичных цифр, занимая биты с 8-го по 31-й.

В длинном формате мантисса имеет 14 шестнадцатичных цифр, используя биты с 8-го по 63-й.

Предполагается, что запятая находится непосредственно слева от самой старшей значащей цифры мантиссы.

Для получения истинной величины числа с плавающей запятой его мантисса умножается на число 16 в соответствующей степени. Для указания этой степени служит поле характеристики, которое занимает разряды с 1-го по 7-й в обоих форматах. Характеристика отсчитывается относительно числа 64 и представляет значения степени в диапазоне от -64 до +63, что соответствует диапазону десятичных чисел от 10^{-78} до 10^{75} .

Нулевой бит в обоих форматах отводится под знак (3) мантиссы. Мантисса отрицательных чисел представляется в прямом коде.

Для операции с плавающей запятой предусмотрено четыре 64-битных регистра. Арифметические операции выполняются над операндами, один из которых находится в регистре, а другой может находиться в регистре или выбирается из памяти. Результат, образованный в регистре, обычно имеет ту же длину, что и операнды. Наличие нескольких регистров уменьшает общее число обращений к памяти для выборки и запоминания промежуточных результатов.

3.3.4. Логические операции.

Логическая информация может иметь как фиксированную, так и переменную длину. К логическим операциям относятся операции сравнения, трансляции, редактирования, проверки и установки битов.

При использовании операндов фиксированной длины логическая информация может быть представлена одним, четырьмя или восемью байтами и обрабатывается в общих регистрах (рис.6).

Логический операнд фиксированной длины (1,4 или 8 байтов)

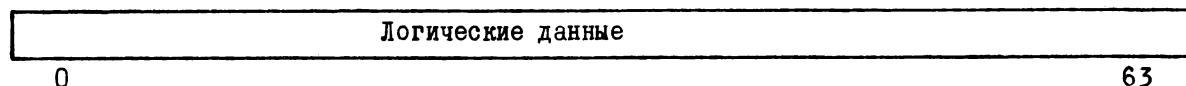


Рис. 6. Логическая информация фиксированной длины

Логический операнд переменной длины (до 256 байтов)

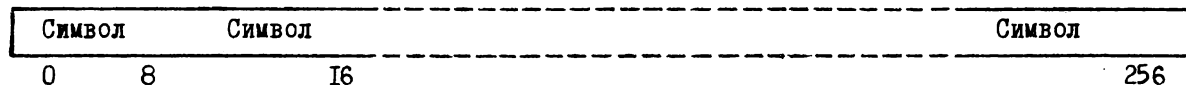


Рис. 7. Логическая информация переменной длины

Большая часть логической информации состоит из графических символов. Эта информация имеет формат переменной длины и может состоять максимально из 256 байтов (рис.7). Обрабатывается информация по принципу "память-память" байт за байтом слева направо.

Вычислительное устройство может обрабатывать любой набор восьмибитных символов, хотя в операциях десятичной арифметики и редактирования имеют место определенные ограничения. Считается, однако, что все ВУ, работающие с символами, используют либо двоичный код для обработки информации (ДКОИ), либо 8-битный код обмена информации (КОИ-8). Нумерация позиций битов символа различна для разных кодов, а именно:

	Позиции битов							
ДКОИ	0	1	2	3	4	5	6	7
КОИ-8	a ₈	a ₇	a ₆	a ₅	a ₄	a ₃	a ₂	a ₁

В этих кодах не все 256 кодов определяют графические символы. Когда желательно представить все возможные комбинации битов, вместо выбранного восьмибитного кода может быть использовано шестнадцатиричное представление, в котором один графический символ соответствует четырехбитному коду. Следовательно, для обозначения восьмибитного байта достаточно двух таких графических символов. Графические символы 0-9 используются для кодов 0000-1001, A-F - для кодов 1010-1111.

3.4. Представление команд.

Программа ВЧУ состоит из команд, а также индексных и управляющих слов, определяющих, какие операции должны быть произведены. Эта информация располагается в основной памяти и может обрабатываться как данные.

3.4.1. Формат команд.

Длина формата команды может быть равна одному, двум или трем полусловам. Это зависит от числа адресов памяти, необходимых для операции. Команды длиной в полуслово не позволяют обращаться к основной памяти. Команды длиной в два полуслова могут содержать один адрес памяти. Команды длиной в три полуслова дают возможность указывать два адреса памяти. Все команды должны быть расположены в памяти на целочисленных границах для полуслов. На рис. 8 показаны пять основных форматов команд.

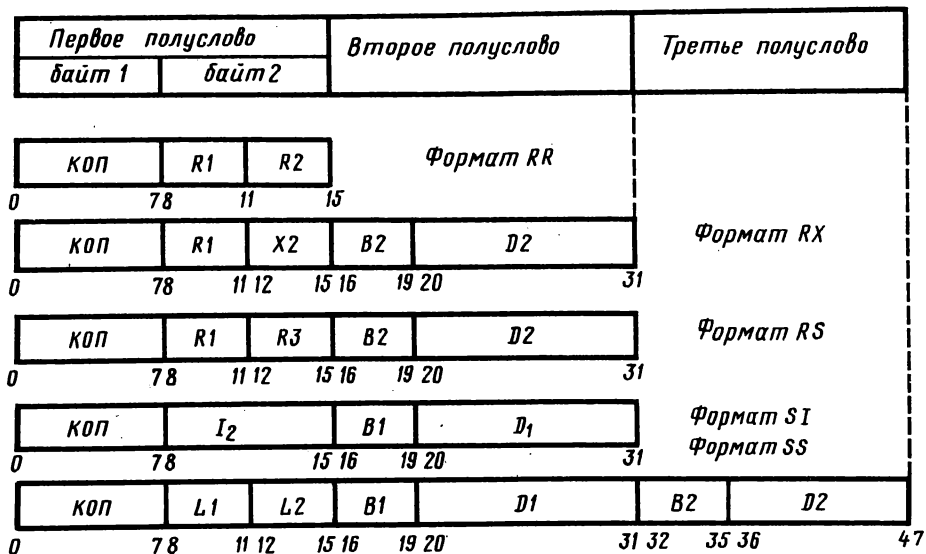


Рис. 8. Основные форматы команд

Пять основных форматов обозначаются форматными кодами: RR, RX, RS, SI, SS. Форматные коды выражают в общих чертах местонахождение операндов, подлежащих использованию в операции.

Форматный код RR обозначает операцию типа регистр-регистр; код RX - операцию типа регистр - индексированная память; код RS - операцию типа регистр-память (без индексации); код SI - операцию типа память-непосредственный операнд, когда один операнд находится в памяти, а другой содержится непосредственно внутри данной команды; код SS - операцию типа память-память.

Для описания выполнения команды операндам присваиваются порядковые номера: первый операнд, второй операнд. В случае команды ПЕРЕХОД ПО ИНДЕКСУ имеется еще третий операнд. Номер операнда определяет способ участия операнда в команде. Операнд, для которого в формате команды указывается поле, обычно обозначается числом, следующим за кодовым наименованием этого поля, например, B1, B2, D1, D2, X2.

В каждом формате первое полуслово команды состоит из двух частей. Первый байт содержит код операции (КОП). Длина и формат команды определяются первыми двумя битами кода операции (табл. 2).

Кодировка длины команды

Позиции битов (0-1)	Длина команды	Формат команды
0 0	Одно полуслово	RR
0 1	Два полуслова	RX
1 0	Два полуслова	RS или SI
1 1	Три полуслова	SS

Второй байт используется либо как два четырехбитных поля, либо как одно восьмибитное. Этот байт может содержать следующую информацию:

четырехбитный номер регистра (R1, R2 или R3);

четырехбитный номер регистра индекса (X2);

четырехбитный код длины операнда (L1 или L2);

восьмибитный код длины операнда (L1);

восьмибитный байт непосредственных данных (I2), т.е. операнд, содержащийся непосредственно внутри данной команды.

В некоторых командах четырехбитное поле или весь второй байт первого полуслова игнорируются.

Второе или третье полуслова имеют всегда один и тот же формат: четырехбитный указатель регистра базы (B1 или B2) и следующее за ним 12-битное смещение (D1 или D2).

3.4.2. Формирование адреса.

По типу адресации операнды можно сгруппировать в три класса: операнды, явно адресуемые в основной памяти, непосредственные операнды, т.е. операнды, содержащиеся непосредственно в данной команде, и операнды, размещенные в общих регистрах и регистрах с плавающей запятой.

Для того, чтобы иметь возможность легко перемещать сегменты программы и обеспечить гибкость задания входных, выходных и рабочих областей основной памяти, все команды, в которых производится обращение к основной памяти, имеют возможность использовать полные адреса.

Адрес обращения к основной памяти формируется из трех двоичных чисел.

Базовый адрес (B) представляет собой 24-битное число в общем регистре, номер которого задан программой в поле B команды. Поле B участвует в формировании всех адресов. Базовый адрес может быть использован как средство для статического перемещения программ и данных. Для массивов он может задавать положение массива, а при обработке записей идентифицировать запись. С помощью базового адреса можно адресовать всю основную память. Базовый адрес может быть использован для индексации.

Индекс (X) представляет собой 24-битное число в общем регистре, который определяется полем X команды. Индекс входит в адрес только для команды формата RX. При употреблении формата RX возможна двойная индексация, т.е. индекс может быть использован для адресации отдельного элемента внутри массива.

Смещение (D) представляет собой 12-битное число, содержащееся в формате команды. Смещение участвует в формировании каждого адреса. Оно позволяет выполнять относительную адресацию 4095 байтов, следующих по порядку за байтом, заданным базовым адресом, или за байтом, адрес которого получается индексацией базового адреса. При вычислениях с массивами смещение может быть использовано для указания одной из нескольких компонент, относящихся к элементу массива. При обработке записей смещение может быть использовано для идентификации блоков внутри записи.

При формировании адреса базовый адрес и индекс трактуются как 24-битные положительные двоичные числа. Аналогично, смещение трактуется как 12-битное положительное двоичное число. Все три компонента складываются как 24-битные двоичные числа. Переполнения игнорируются. Так как при формировании любого адреса участвует базовый адрес, сумма всегда имеет 24 бита. Биты адреса нумеруются числами 8-31 в соответствии с нумерацией битов базового адреса и битов индекса в общем регистре.

В программе могут встречаться нулевые поля базового адреса, индекса и смещения. Нуль указывает на отсутствие соответствующей компоненты адреса. Нулевое содержимое поля базового адреса или поля индекса означает, что при формировании адреса должно быть использовано нулевое значение соответствующей компоненты, а не содержимое общего регистра 0. Трактовка смещения, равного нулю, не имеет никакой специфики. Первоначальное задание, модификация и тестирование базовых адресов и индексов могут быть выполнены при помощи команд с фиксированной запятой или команд ПЕРЕХОД С ВОЗВРАТОМ, ПЕРЕХОД ПО СЧЕТЧИКУ или ПЕРЕХОД ПО ИНДЕКСУ.

Чтобы пояснить логическую структуру формата команды, ниже приводятся примеры двух команд и их форматы (рис.9).

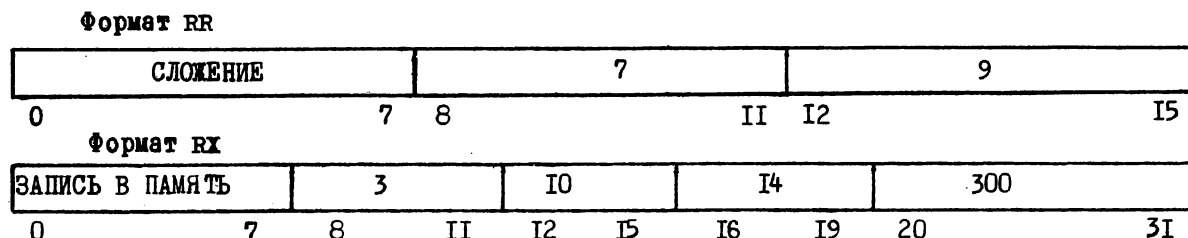


Рис. 9. Логическая структура формата команд

При выполнении команды СЛОЖЕНИЕ содержимое общего регистра с номером 9 складывается с содержимым общего регистра с номером 7, и результат помещается в общий регистр с номером 7.

При выполнении команды ЗАПИСЬ В ПАМЯТЬ содержимое общего регистра с номером 3 заносится в основную память по адресу, который получается при сложении числа 300 с содержимым младших 24 битов общих регистров с номерами I4 и I0.

3.4.3. Последовательное выполнение команд.

Команда выбирается из места памяти, определяемого адресом команды в слове состояния программы. Для получения адреса следующей команды адрес выполненной команды увеличивается на число единиц, равное числу байтов в этой команде. Затем выполняется очередная команда и весь процесс повторяется с новым значением адреса команды.

Можно считать, что все полуслова, из которых состоит команда, выбираются из памяти после завершения выполнения предыдущей операции и до выполнения текущей, даже если размер физического слова памяти и совмещение выполнения команды с обращением к памяти могут повлиять на то, как будет происходить выборка команды в действительности. Поэтому модифицировать команду в памяти можно непосредственно предшествующей командой.

Нарушение обычной последовательности операций может происходить при переходах, переключении состояния, прерывании или при вмешательстве оператора.

3.4.4. Переходы.

Обычная последовательность выполнения команд нарушается при обращениях к подпрограммам, если встречается выбор двух путей, или когда некоторый сегмент программы, например цикл, должен быть выполнен несколько раз. В этих случаях используются команды переходов.

Связь с подпрограммами осуществлена так, что позволяет не только ввести адрес новой команды, но и сохранить адрес возврата и всю необходимую для возврата информацию.

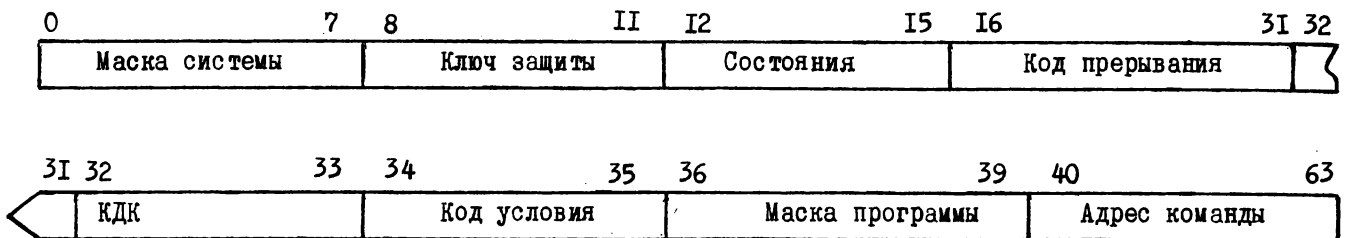
Общий и симметричный характер выбора пути обеспечивается командой УСЛОВНЫЙ ПЕРЕХОД. Эта команда проверяет двухбитный код условия, который отражает результат большинства арифметических и логических операций и операций ВВ. В зависимости от результата в этих операциях код устанавливается в одно из четырех возможных состояний, а команда УСЛОВНЫЙ ПЕРЕХОД может использовать любое из этих состояний в качестве критерия для выполнения перехода. Например, код ус-

ловия отражает такие случаи, как неравенство нулю результата операции, первый операнд больше или равен второму, переполнение, занятость канала и т.д. Однажды установленный код условия не изменяется до тех пор, пока встретится команда, которая установит другой код условия. Двухбитный код условия может иметь четыре значения: 0, 1, 2, 3. Каждое значение кода имеет определенный смысл только для операции, в результате которой оно установлено.

Управление циклом можно осуществить с помощью команды УСЛОВНЫЙ ПЕРЕХОД, анализируя результаты операций над адресами или счета числа повторений. Для некоторых часто встречающихся комбинаций арифметических действий и тестов предусмотрены команды ПЕРЕХОД ПО СЧЕТЧИКУ и ПЕРЕХОД ПО ИНДЕКСУ. Такие переходы (в силу большой специализации) в определенных случаях значительно увеличивают производительность.

3.4.5. Слово состояния программы.

Слово состояния программы (рис. 10) представляет собой двойное слово и содержит информацию, необходимую для выполнения программы. Общее назначение ССП - управление последовательностью выборки команд, а также фиксация и индикация состояния системы по отношению к выполняемой в настоящий момент программе.



КДК - код длины команды; состояния: бит 12 - КОИ-8, бит 13 - маска сбоя машины, бит 14 - состояние "Ожидание", бит 15 - состояние "Задача"

Рис. 10. Слово состояния программы (ССП):

ССП, используемое в данный момент времени, называется текущим. Запоминая текущее ССП во время прерывания, можно сохранить состояние ВЧУ для последующего анализа. Задавая новое ССП или его часть, можно изменить состояние ВЧУ полностью или частично.

3.4.6. Прерывания.

Система прерывания позволяет ВЧУ изменять состояние при возникновении определенных условий вне системы, во внешних устройствах или в самом ВЧУ. Возможны пять классов прерываний: по вводу-выводу, программные, при обращении к супервизору, внешние и по сбою машины.

Каждому классу прерываний соответствуют два ССП, которые называются старым и новым и помещаются в специально отведенных местах основной памяти (табл. 3).

Постоянно распределенная область основной памяти

Таблица 3

Адрес десятичный	Длина	Назначение
0 0000 0000	Двойное слово	ССП первоначальной программы
8 0000 1000	Двойное слово	КСК1 первоначальной загрузки программы
16 0001 0000	Двойное слово	КСК2 первоначальной загрузки программы
24 0001 1000	Двойное слово	Старое ССП для внешних прерываний
32 0010 0000	Двойное слово	Старое ССП для обращения к супервизору
40 0010 1000	Двойное слово	Старое ССП для программных прерываний
48 0011 0000	Двойное слово	Старое ССП для прерываний по сбою машины

Адрес десятичный	Длина	Назначение
56 0011 1000	Двойное слово	Старое ССП для прерываний по вводу-выводу
64 0100 0000	Двойное слово	Слово состояния канала
72 0100 1000	Слово	Адресное слово канала
76 0100 1100	Слово	Не используется
80 0101 0000	Слово	Таймер
84 0101 0100	Слово	Не используется
88 0101 1000	Двойное слово	Новое ССП для внешних прерываний
96 0110 0000	Двойное слово	Новое ССП для обращения к супервизору
104 0110 1000	Двойное слово	Новое ССП для программных прерываний
112 0111 0000	Двойное слово	Новое ССП для прерываний по сбоя машины
120 0111 1000	Двойное слово	Новое ССП для прерываний по вводу-выводу
128 1000 0000	-	Диагностическая область

Для всех классов прерывание состоит в запоминании текущего ССП в месте, предназначенном для старого ССП, и принятии нового ССП в качестве текущего. Старое ССП хранит всю необходимую информацию о состоянии системы в момент прерывания. Если в конце программы обработки прерывания имеется команда, заменяющая текущее ССП на старое, система восстановит состояние, предшествующее прерыванию, и выполнение прерванной программы будет продолжено.

Прерывания осуществляются только в том случае, если ВЧУ не замаскирован по отношению к данному источнику прерывания. Для того, чтобы замаскировать прерывания некоторых типов, используются биты маски системы, маски программы и бит маски сбоя машины, имеющиеся в ССП. В случае, если некоторый тип прерывания замаскирован, запрос на такое прерывание либо ожидает, когда он будет воспринят, либо игнорируется. При помощи маски системы можно замаскировать с ожиданием прерывания по вводу-выводу и внешние прерывания. Маска программы может маскировать 4 из 15 программных прерываний, запросы на которые при этом уничтожаются, а маска сбоя машины может маскировать с ожиданием прерывания по сбоя машины. Другие типы прерываний замаскировать нельзя.

Прерывание производится обычно после того, как выполнение текущей команды закончилось и до начала выполнения следующей. Однако появление прерывания может повлиять на выполнение текущей команды.

Для того, чтобы обеспечить правильную работу ВЧУ после прерывания, производится запоминание причины прерывания и предусмотрена возможность определить расположение последней выполнявшейся до прерывания команды.

Прерывание по вводу-выводу дает возможность ВЧУ реагировать на определенные условия в каналах и внешних устройствах (ВУ).

Прерывание по ВВ может произойти только в случае, если бит маски, связанный с данным каналом, установлен в 1. Адреса канала и ВУ, вызвавшие прерывание, записываются в битах 16-31 старого ССП. Дополнительная информация о работе ВВ хранится в слове состояния канала (ССК), которое запоминается во время прерывания.

Необычные ситуации, встречающиеся в программе, вызывают прерывания, которые называются программными прерываниями. К таким ситуациям относятся появление некорректных операндов или их спецификаций, а также появление особых результатов. Причина прерывания идентифицируется кодом прерывания. Различные причины, которые могут вызвать прерывание, показаны в табл.4.

Прерывание при обращении к супервизору происходит в результате выполнения команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ. Восемь битов формата команды занимают место кода прерывания в старом ССП, позволяя тем самым идентификацию прерывания в пределах данного класса. Основное назначение команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ состоит в том, чтобы произвести переключение ВЧУ из состояния "Задача" в состояние "Супервизор". Кроме того, такое прерывание может быть использовано для других режимов переключения состояний.

Внешние прерывания позволяют ВЧУ отвечать на сигналы, поступающие от кнопки ПРЕРЫВАНИЕ на пульте управления, от таймера, а также отвечать на внешние сигналы средств прямого управления.

Внешние прерывания имеют место только в том случае, если 7-й бит маски системы в ССП равен 1. Причина прерывания идентифицируется в коде прерывания в старом ССП в битах 24-31 (табл.5). Биты 16-23 в коде прерывания устанавливаются в нуль.

Прерывания по сбоям машины. Появление сигнала от схем контроля об ошибке в машине (если прерывания данного класса не замаскированы) прекращает выполнение текущей команды и начинает диагностическую процедуру, после чего происходит прерывание. Сигнал об ошибке в машине не может появиться из-за неправильных данных или команд. Диагностическая информация заносится в диагностическую область, начиная с адреса I28.

Приоритет прерываний

Во время выполнения команды одновременно может поступить несколько запросов на прерывание. Одновременные запросы на прерывания рассматриваются в следующем заранее установленном порядке:

- прерывания по сбою машины;
- программные прерывания или прерывания при обращении к супервизору;
- внешние прерывания;
- прерывания по вводу-выводу.

Программные прерывания и прерывания при обращении к супервизору взаимно исключают друг друга и не могут появиться одновременно.

Таблица 4

Код прерывания для программных прерываний

Код прерывания	Причина прерывания программы
1. 00000001	Код операции
2. 00000010	Привилегированная операция
3. 00000011	Некорректность команды ВЫПОЛНИТЬ
4. 00000100	Защита
5. 00000101	Адресация
6. 00000110	Спецификация
7. 00000111	Десятичные данные
8. 00001000	Переполнение с фиксированной запятой
9. 00001001	Деление с фиксированной запятой
10. 00001010	Десятичное переполнение
11. 00001011	Десятичное деление
12. 00001100	Переполнение порядка
13. 00001101	Исчезновение порядка
14. 00001110	Значимость
15. 00001111	Деление с плавающей запятой

Код прерывания для внешних прерываний

Бит кода	Причина внешнего прерывания	Бит маски
24	Таймер	7
25	Кнопка ПРЕРЫВАНИЕ	7
26	Внешний сигнал 2	7
27	Внешний сигнал 3	7
28	Внешний сигнал 4	7
29	Внешний сигнал 5	7
30	Внешний сигнал 6	7
31	Внешний сигнал 7	7

Когда требуется обслужить более чем один запрос на прерывание, происходит запоминание старого ССП и выборка нового ССП, относящегося к прерыванию, которое рассматривается в первую очередь. Затем это новое ССП запоминается без выполнения какой-либо команды и выбирается следующее ССП. Этот процесс продолжается до тех пор, пока все запросы на прерывание не будут обслужены. После обслуживания последнего запроса возобновляется выполнение команды с использованием ССП, которое было выбрано последним. Поэтому порядок выполнения подпрограммы прерывания является обратным по отношению к порядку, в котором выбирались ССП. Таким образом, фактический порядок обслуживания прерываний (по вводу-выводу, внешние, программные или для обращения к супервизору) соответствует их важности. Прерывания по сбою машины блокируют рассмотрение всех других прерываний.

Программные состояния

Состояние ВЧУ в целом определяется парами программных состояний четырех типов. В каждой паре переход от одной возможной альтернативы к противоположной может делаться независимо, и большинству пар соответствует бит или биты в ССП. Взаимно исключающие программные состояния имеют следующие названия: "Останов/Работа", "Ожидание/Счет", "Прерывание разрешено/Прерывание замаскировано" и "Супервизор/Задача". Эти состояния различаются характером влияния каждого из них на функционирование ВЧУ и организации их индикации и переключения. Функции программных состояний, их индикация и переключение не зависят друг от друга.

Состояние "Останов" или "Работа". Переключение ВЧУ из состояния "Работа" в "Останов" и наоборот производится вручную. Если ВЧУ находится в состоянии "Останов", команды не выполняются, прерывания не принимаются, таймер не обновляется. В состоянии "Работа" ВЧУ может выполнять команды и обрабатывать прерывания.

Состояние "Ожидание" или "Счет". В состоянии "Счет" выборка и выполнение команд происходит обычным образом. Состояние "Ожидание" обычно вводится программой для ожидания прерывания, например, прерывания по вводу-выводу или от вмешательства оператора с пульта управления. В состоянии "Ожидание" команды не выполняются, но таймер продолжает работать. Кроме того, допускаются прерывания от ВВ и внешние прерывания, если они не замаскированы. Вид состояния определяется значением 14-го бита в ССП.

Состояние "Прерывание разрешено" или "Прерывание замаскировано". ВЧУ может быть прерываемо или замаскировано для прерываний по вводу-выводу, внешних, по сбою машины и некоторых программных прерываний. Если ВЧУ находится в состоянии "Прерывание разрешено" для некоторого класса прерываний, эти прерывания могут восприниматься ВЧУ. Если ВЧУ замаскировано, прерывания по вводу-выводу и по сбою машины ожидают обработки в дальнейшем, тогда как программные прерывания игнорируются. Переключение ВЧУ из состояния "Прерывание замаскировано" в состояние "Прерывание разрешено" и наоборот производится путем изменения значения битов маски в ССП.

Состояние "Супервизор" или "Задача". В состоянии "Задача" группа команд является недопустимой. В состоянии "Супервизор" допустимы все команды. Выбор состояния определяется значением 15-го бита в ССП.

3.5. Оперативная память.

Оперативная память процессора является памятью на ферритовых сердечниках, спроектированной по системе 2,5D с циклом 2 мксек и разрядностью два байта.

Оперативная память состоит из трех функциональных памятей:

основная память доступна программе и используется для хранения программ, данных и размещения системной информации;

локальная память - 16 общих регистров, 4 регистра с плавающей запятой и служебная зона памяти, недоступная программам и используемая в качестве рабочих регистров процессора и для целей диагностики;

мультиплексная память - память управляющих слов для выполнения обмена информацией с внешними устройствами через мультиплексный канал.

Объем основной памяти может изменяться от 65536 байтов до 262144 байтов. Нарастивание памяти производится блоками по 65536 байтов.

Объем локальной памяти 256 байтов. Распределение локальной памяти приведено в приложении 2.

Объем мультиплексной памяти зависит от объема основной памяти и равен 768 байтам при объеме основной памяти 65536 байтов и 1536 байтам при объеме основной памяти 131072 и 262144 байта.

Оперативная память имеет разделенный цикл. Это означает, что между тактом чтения информации и тактом записи, которые по длительности равны по 1 мксек, можно выполнять обработку прочитанной информации с последующей записью ее по тому же адресу.

При включении питания содержимое оперативной памяти сохраняется. Выключение питания не влияет на содержимое оперативной памяти, если процессор находится в состоянии "Останов".

Считанная из памяти информация проверяется на соответствие четности информационной части байта значению бита четности. При наличии несовпадения регистрируется сбой оборудования и производится прерывание по сбою машины.

3.6. Защита памяти.

Процессор оборудован средствами защиты памяти, дающими возможность защищать содержимое основной памяти от неправильного использования.

Для целей защиты основная память разделена на блоки по 2048 байтов. С каждым блоком связан пятиразрядный ключ памяти. Ключ памяти при каждом обращении к основной памяти сравнивается с ключом защиты, характеризующим выполняемую программу. Один разряд ключа памяти отражает режим работы средств защиты: защита по записи или защита по чтению и записи.

При работе средств защиты в режиме защиты по записи считается, что произошло нарушение защиты памяти, если ключ памяти блока, в котором должна быть изменена информация, не соответствует ключу защиты.

Если средства защиты работают в режиме защиты по чтению и записи, то нарушение защиты памяти проходит при несоответствии ключей защиты и памяти даже в том случае, когда обращение производилось с целью чтения информации без ее изменения.

В случае, когда обращение к памяти производится по инициативе ВЧУ, для сравнения используется ключ защиты, находящийся в текущем ССП. Если обращение к памяти вызвано операцией в канале, для сравнения используется ключ защиты, поступающий из канала.

Если во время выполнения программы имеет место нарушение защиты памяти, то происходит прерывание программы, а содержимое запрещенной области остается без изменения.

Нарушение защиты при операции ввода-вывода приводит к тому, что передача данных прекращается так, что содержимое защищенной области памяти остается без изменения, а факт нарушения защиты отражается в командном слове состояния канала. (Описание средств защиты см. в Е13.055.001 Т01 и Е13.055.001 Т06).

3.7. Таймер.

В ВЧУ для измерения интервалов времени предусмотрен таймер. С его помощью программным путем можно организовать часы, показывающие время суток. Таймер представляет собой полное слово, хранящееся в основной памяти по адресу 80. Из содержимого этого слова производится вычитание с частотой 50 гц. Слово таймера рассматривается как целое число со знаком и может обрабатываться по правилам арифметики с фиксированной запятой. Когда значение таймера переходит из положительного в отрицательное, выдается запрос на внешнее прерывание. Полный цикл таймера составляет 15,5 часов.

Обновленное значение таймера доступно в конце выполнения каждой команды. Таймер не обновляется, если ВЧУ находится в состоянии "Останов". Значение таймера может быть изменено обращением к 80-й ячейке памяти. В качестве датчика интервалов времени таймер используется для измерения относительно небольших интервалов. Он может быть установлен на любое значение в любой момент времени.

3.8. Средства прямого управления.

К средствам прямого управления относятся команды ПРЯМОЕ ЧТЕНИЕ и ПРЯМАЯ ЗАПИСЬ и шесть линий внешних прерываний. Команды прямого чтения и записи позволяют осуществлять передачу отдельного байта информации между специальным внешним устройством или ВЧУ другой машины и основной памятью системы. Обычно для передачи любого количества информации желательно использовать каналы системы, а средства прямого управления более удобно использовать для передачи управляющей и синхронизирующей информации.

Каждая из шести внешних сигнальных линий, получив сигнал извне, устанавливает запрос на внешнее прерывание.

3.9. Типы и характеристики каналов.

Каналы ввода-вывода предназначены для управления выполнением операций передачи информации между основной памятью процессора ЕС-2020 и УВУ.

Управление работой ВУ стандартизировано. При этом определены структура и набор команд, управляющих сигналами и информационными словами, применимых для управления любым ВУ независимо от его характеристик.

Преобразование стандартного набора информационных и управляющих слов и сигналов к виду, обеспечивающему управление работой конкретного ВУ, производится УВУ. Стандартизация управляющей информации и сигналов позволяет подключать ВУ разных типов через УВУ к каналам.

Подключение УВУ к каналам производится через стандартный интерфейс, под которым понимается система линий и система правил кодирования и передачи информации, обеспечивающая подсоединение ВУ разного типа к одному каналу.

Операции ввода-вывода запускаются по командам управления каналами. Управляющая информация, полученная каналом, преобразуется в последовательность сигналов, поступающих в УВУ. Канал производит запуск УВУ на выполнение команды ввода (считывания) или вывода (записи) информации.

УВУ формирует запросы на передачу информации, которые анализируются и обслуживаются каналом. Если в УВУ возникают сигналы, которые необходимо передать в ВЧУ (например, сигнал об окончании операции ввода-вывода), то канал преобразует эти сигналы в стандартную форму, удобную для дальнейшего использования в ВЧУ.

3.9.1. Мультиплексный канал.

Мультиплексный канал обеспечивает обмен информацией с ВУ с относительно малой и средней скоростью передачи данных (табл.6).

Мультиплексный канал может работать в мультиплексном или монопольном режимах.

В мультиплексном режиме средства канала распределяются во времени для обслуживания нескольких одновременно работающих ВУ, обмен информацией с каждым из которых происходит со скоростью, определяемой быстродействием ВУ. В то время, когда одно ВУ связано с каналом для передачи информации в основную память процессора или из нее, другие ВУ могут выполнять действия, не требующие использования средств канала, например, печать или перфорацию ранее принятой информации, считывание информации с внешнего носителя и т.п.

В мультиплексном режиме выполнение операции ввода-вывода разделяется на короткие интервалы времени, в каждом из которых между ВУ и основной памятью передается один или несколько байтов информации, связанных с выполняемой операцией. Интервалы времени, относящиеся к обслуживанию различных ВУ, чередуются в порядке поступления запросов на обслуживание. При одновременном возникновении запросов в нескольких ВУ их обслуживание производится в порядке приоритета определяемом быстродействием ВУ и его максимально возможным временем ожидания удовлетворения запроса на передачу информации.

Таблица 6

Тип устройства	Идентификатор устройства	Техническая скорость ВУ	Скорость передачи информации
Ввод перфокарт	ЕС-6012	500 карт/мин	670-1340 б/сек
Ввод перфолент	ЕС-6022	1500 стр/сек	1500 б/сек
Устройство печати	ЕС-7030	750-840 стр/мин	1540 б/сек
Вывод перфокарт	ЕС-7010	100 карт/мин	140-280 зн/сек
Вывод перфолент	ЕС-7022	150 стр/сек	150 б/сек
Пишущая машинка	ЕС-7570	10 зн/сек	10 б/сек

Пропускная способность мультиплексного канала в мультиплексном режиме 10-16 кб/сек.

Количество одновременно работающих ВУ в мультиплексном режиме определяется пропускной способностью канала. Суммарная скорость передачи информации всех одновременно работающих в мультиплексном режиме ВУ не должна превышать пропускной способности мультиплексного канала.

Между ВУ и мультиплексным каналом и между каналом и основной памятью одновременно может передаваться только один байт информации.

Если ВУ занимает (монополизирует) оборудование мультиплексного канала до конца выполнения операции ввода-вывода, ВЧУ не может одновременно с передачей информации в монопольном режиме выполнять другую работу и обслуживать запросы на передачу информации других ВУ мультиплексного канала.

Пропускная способность мультиплексного канала в монопольном режиме до 100 кб/сек.

Мультиплексный канал при выполнении операций ввода-вывода разделяет оборудование вычислительного устройства.

Средства канала, необходимые для выполнения определенной операции ввода-вывода, связанной с одним ВУ, называются подканалом. Каждый подканал содержит всю необходимую информацию для управления одной операцией ввода-вывода. Это информация хранится в мультиплексной памяти.

Каждый подканал может управлять операцией ввода-вывода одного ВУ или группы ВУ, имеющих общее УВУ и составленных таким образом, что в каждый данный момент времени операция передачи информации может выполняться в одном из ВУ этой группы. Подканалы, обслуживающие такие группы ВУ, называются разделенными. Первые 8 подканалов являются разделенными и могут управлять работой до 128 ВУ. Остальные подканалы являются неразделенными, каждый из которых может управлять отдельным ВУ.

В табл.7 приведена зависимость количества подканалов и ВУ от варианта и поставки процессора, определяемого объемом оперативной памяти.

3.9.2. Селекторный канал.

Селекторные каналы обеспечивают обмен информацией с ВУ с относительно высокой скоростью передачи информации, таких, например, как запоминающее устройство на магнитной ленте ЕС-55 II.

Каждый селекторный канал содержит всего один подканал. Он всегда работает в монопольном режиме, и только одно ВУ может участвовать в операции передачи данных одновременно.

Селекторные каналы почти независимы от остального оборудования процессора. Для передачи данных они используют только 3 такта основной памяти и, следовательно, вмешательство в работу ВЧУ при обслуживании запросов селекторных каналов на передачу информации минимальное.

Между селекторными каналами и ВУ информация всегда передается по два байта, но в начале или конце операции ввода-вывода может передаваться один байт информации в зависимости от четности адреса и содержимого счетчика байтов.

Максимальная пропускная способность селекторного канала - 300 кб/сек.

К селекторному каналу может быть подключено 256 ВУ.

Таблица 7

Модель процессора	Объем оперативной памяти	Объем мультиплексной памяти (байт)	Количество подканалов	Количество ВУ, подключаемых к каналу	Примечание
ЕС-2020-1	64	768	48	176	K=1024 байта
ЕС-2020-2	128	1792	112	240	
ЕС-2020-3	256	1792	112	240	

3.9.3. Одновременная работа каналов.

При одновременной работе каналов ввода-вывода их запросы на передачу информации обслуживаются в следующем порядке:

- селекторный канал 1;
- селекторный канал 2;
- мультиплексный канал.

Если работают оба селекторных канала одновременно, то пропускная способность каждого канала уменьшается до 200 кб/сек.

Если выполняется обслуживание запроса мультиплексным каналом на передачу информации (или ВЧУ выполняет какую-либо программу) и одновременно работают селекторные каналы, то время обслуживания мультиплексным каналом увеличивается.

3.9.4. Выполнение операций ввода-вывода.

Программа ВЧУ только определяет и начинает операцию ввода-вывода, выполнение которой затем управляется каналной программой.

Операция ввода-вывода начинается по команде НАЧАТЬ ВВОД-ВЫВОД, имеющей формат, представленный на рис. II.

Канал начинает операцию ввода-вывода и определяет адрес ВУ путем сложения содержимого общего регистра, указанного в поле VI (разряды I6-I9), и содержимого поля DI команды (разряды 20-31). Эта сумма имеет формат, показанный на рис. I2.

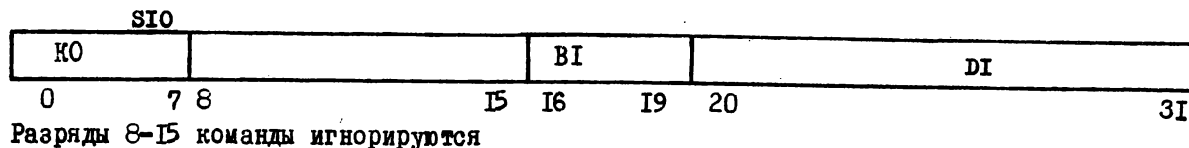


Рис. II. Формат команды НАЧАТЬ ВВОД-ВЫВОД

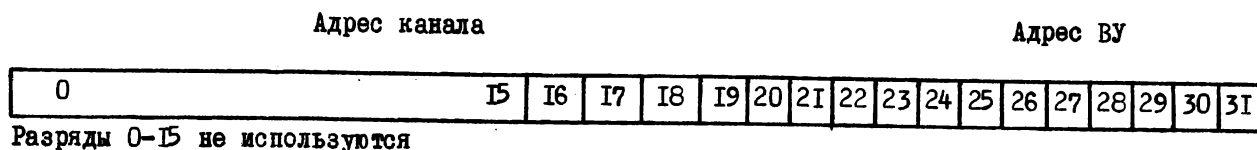


Рис. I2. Формат суммы VI + DI

В байте адреса канала для задания адреса канала используются только разряды 21, 22 и 23 (табл. 8).

Таблица 8

Разряды			Канал
21	22	23	
0	0	0	Мультиплексный канал
0	0	1	Первый селекторный канал
0	1	0	Второй селекторный канал
0	1	1	Недействительные адреса каналов в процессоре
1	1	1	ЕС-2020

В разрядах 24-31 указывается адрес ВУ.

3.9.5. Канальная программа.

Канальная программа представляет собой совокупность командных слов канала (КСК) и размещена в основной памяти.

Адрес первого КСК канальной программы всегда выбирается из фиксированной ячейки 72 постоянно распределенной области в основной памяти, где расположено адресное слово канала (АСК).

В АСК (рис. 13) указан ключ защиты области основной памяти, предназначенной для операции ввода-вывода, а также адрес первого КСК.

0	Ключ защиты	3	4	7	8	Адрес первого КСК	31
---	-------------	---	---	---	---	-------------------	----

Рис. 13. Формат АСК

КСК - это двойное слово, которое может храниться в любой части основной памяти и содержит всю необходимую информацию для задания одной команды ввода-вывода.

Канальная программа может содержать одно или несколько КСК, каждое из которых определяет:

- операцию, которую необходимо выполнить;
 - область основной памяти, которая должна быть использована;
 - действия, которые должны произойти после завершения операции ввода-вывода.
- Формат КСК показан на рис. 14.

Разряды 0-7 определяют одну из канальных команд, приведенных в табл. 9.

Код команды		Адрес данных		Флажки		000		Счетчик байтов			
0	7	8	31	32	36	37	39	40	47	48	64

Рис. 14. Формат КСК

Таблица 9

Команда	Разряд кода команды							Выполняемые действия	
	0	1	2	3	4	5	6		7
НЕДЕЙСТВИТЕЛЬНАЯ	X	X	X	X	0	0	0	0	В основную память передается информация о состоянии ВУ, относящаяся к необычным условиям, обнаруженным в течение последней операции ввода-вывода и детализирующих состояние ВУ
УТОЧНИТЬ СОСТОЯНИЕ	M	M	M	M	0	1	0	0	
ПЕРЕСЛАТЬ В КАНАЛ	X	X	X	X	1	0	0	0	Указывается адрес следующего КСК канальной программы в основной памяти
СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ	M	M	M	M	1	1	0	0	Информация считывается при обратном движении внешнего носителя и записывается в основную память в порядке убывания адресов
ЗАПИСАТЬ	M	M	M	M	M	M	0	1	Информация из основной памяти передается на внешний носитель
СЧИТАТЬ	M	M	M	M	M	M	1	0	Информация из ВУ передается в основную память в порядке возрастания адресов
УПРАВЛЕНИЕ	M	M	M	M	M	M	1	1	Из основной памяти передается управляющая информация ВУ для выполнения таких операций, как протяжка бумаги на печатающем устройстве и т.п.

Примечание. X - значение разряда игнорируется; M - разряд модификатора

Разряды модификатора кода команды определяют ВУ условия, при которых должна быть выполнена операция ввода-вывода (например, плотность записи на магнитную ленту). Значение разрядов модификатора зависит от ВУ.

Разряды 8-31 КСК определяют адрес расположения первого байта информации в основной памяти.

В разрядах 32-36 указываются признаки (флажки), которые модифицируют основную команду КСК (табл. 10).

Разряды 40-47 не используются.

В разрядах 48-63 КСК указывается количество байтов, которое должно быть передано под управление этого КСК.

Счетчик байтов вместе с адресом данных полностью определяют область основной памяти, используемую текущим КСК. Их содержимое модифицируется при передаче каждого байта данных.

Каналы передают данные под управлением канальных программ, при этом ВЧУ работает параллельно, выполняя какую-либо программу. ВЧУ становится свободен для продолжения своей программы после начала операции ввода-вывода, исключая работу мультиплексного канала в монопольном режиме.

Каналы сохраняют следующую управляющую информацию для каждого ВУ, выбранного для операции ввода-вывода:

- ключ защиты;
- адрес данных;
- идентификатор операции ввода-вывода (код команды);
- признаки КСК;
- счетчик байтов;
- состояние канала;
- адрес следующего КСК.

Разряд КСК	Название признака	Идентификатор	Выполняемые действия
32	Цепочка данных	ЦД	После того, как счетчик байтов текущего КСК станет равным 0, канал должен использовать из следующего КСК новый адрес данных, счетчик байтов и признак. Код команды игнорируется и продолжается выполнение той же команды
33	Цепочка команд	ЦК	После завершения ВУ текущей операции ввода-вывода, канал должен считать следующее КСК и выполнить новую команду
34	Подавление индикации неверной длины	ПИД	Канал игнорирует индикацию неверной длины, когда ВУ сигнализирует о конце операции раньше, чем количество байтов, определенное счетчиком байтов, будет передано, или наоборот
35	Блокировка записи	БЗП	Канал выполняет команду ввода (СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ или УТОЧНИТЬ СОСТОЯНИЕ) без записи информации в основную память
36	Программно-управляемое прерывание	ПУП	Канал вызывает прерывание основной программы в течение выполнения канальной программы при считывании из основной памяти КСК, содержащего признак ПУП

Каждый селекторный канал имеет только один набор регистров для перечисленной выше информации, поскольку он работает только с одним ВУ одновременно.

В мультиплексном канале управляющая информация должна сохраняться для каждого работающего подканала в мультиплексной памяти. Когда отдельное ВУ, выбранное по команде НАЧАТЬ ВВОД-ВЫВОД, приступает к выполнению операции, в соответствующий подканал мультиплексной памяти записывается необходимая управляющая информация.

Канал обращается к мультиплексной памяти для того, чтобы связать ВУ с основной памятью. После окончания обслуживания запроса ВУ на передачу информации, модифицированная управляющая информация записывается в соответствующий подканал мультиплексной памяти, и канал становится свободным для выполнения других операций ввода-вывода.

3.9.6. Окончание операций ввода-вывода.

Операция, заданная командой НАЧАТЬ ВВОД-ВЫВОД, заканчивается, если будет выполнена вся цепочка КСК. Обычно операция ввода-вывода заканчивается, когда вырабатывается сигнал КАНАЛ КОНЧИЛ. Этот сигнал свидетельствует о том, что область основной памяти, определенная счетчиком байтов в КСК, исчерпана, и канал передал весь объем информации, указанный в программе.

Для принудительного прекращения операции ввода-вывода используется команда ОСТАНОВИТЬ ВВОД-ВЫВОД. Необходимость в прекращении операции может возникнуть, например, в случае, когда селекторный канал должен выполнить операцию более срочную по сравнению с ранее начатой операцией. В команде ОСТАНОВИТЬ ВВОД-ВЫВОД указывается адрес канала и ВУ. По этой команде ВУ дается указание отключиться от канала и прекратить выполняемую операцию ввода-вывода.

Операция ввода-вывода может быть также прекращена при возникновении ошибок в аппаратуре ВУ или каналов, или из-за таких ситуаций, как обрыв бумаги в печатающем устройстве и конец колоды перфокарт в перфораторе.

3.9.7. Состояние каналов и внешних устройств.

В течение выполнения программы ввода-вывода непрерывно изменяется состояние канала и ВУ. С момента начала операции ввода-вывода канал продвигается по цепочке КСК со скоростью работы ВУ, т.е. асинхронно по отношению к программе. В некоторый данный момент канал может

быть занят выполнением ранее начатой операции или находиться в свободном состоянии, позволяющем начать следующую операцию ввода-вывода. При выполнении операций в канале и ВУ могут возникнуть особые ситуации, которые должны учитываться при организации ввода-вывода.

Для анализа состояний каналов и ВУ имеются команды, позволяющие определить состояние канала в процессе выполнения текущей операции ввода-вывода и получить информацию о состоянии ВУ.

Состояние канала и ВУ записывается в специальное слово состояния канала (ССК) (рис. 15).

Ключ защиты				Адрес КСК	Состояние	Счетчик байтов			
0	3	4	7	8	31	32	47	48	63

Рис. 15. Формат ССК

В поле 0-3 указывается ключ защиты;

8-31 адрес последнего использованного КСК+8;

48-63 остаточный счетчик байтов;

32-47 записываются байты состояния ВУ и канала, значение разрядов которых приведено в табл. II и I3 соответственно.

Таблица II

Разряд ССК	Идентификатор	Название	Условия установки
32	U_0	Внимание	Оператор нажал специальную кнопку на пульте управления УВУ
33	U_1	Модификатор	Устанавливается совместно с U_3 и U_5 для модификации основного значения состояния
34	U_2	УВУ кончилось	УВУ закончило операцию ввода-вывода. Вырабатывается обычно только в разделенных УВУ
35	U_3	Занято	ВУ занято выполнением предыдущей операции ввода-вывода. Если одновременно установлен U_1 и U_3 , занято УВУ
36	U_4	Канал кончил	Подканал закончил свою часть операции передачи данных. ВУ может к этому моменту и не закончить операции ввода-вывода
37	U_5	ВУ кончилось	1. ВУ закончило операцию ввода-вывода и готово для выполнения других операций. 2. Предварительно остановленное ВУ начало работать (ВУ перешло из состояния "не готово" в состояние "готово")
38	U_5	Контроль ВУ	В ВУ один или несколько разрядов уточненного байта состояния. Программа должна проверить уточненный байт состояния для того, чтобы определить, что произошло в ВУ
39	U_6	Исключительный случай в ВУ	В ВУ произошел исключительный случай, например, конец страницы на печатающем ВУ, пустой карман в устройстве ввода карт

Более подробная информация о состоянии ВУ приводится в уточненных байтах состояния. Количество уточненных байтов состояния зависит от ВУ. Однако первые 6 разрядов первого уточненного байта состояния одинаковы для всех ВУ (см. табл. I2).

Таблица I2

Разряд в байте	Значение разряда в состоянии "I"	Условия установки
0	Команда отвергнута	ВУ не может выполнить команду, которую ему выдал канал
1	Требуется вмешательство	Замятие карт, нет бумаги в устройстве и т.д.
2	Контроль ШИН-К	ВУ получило байт на ШИН-К с неправильным контрольным разрядом
3	Контроль оборудования	Неисправность некоторой части оборудования в ВУ
4	Контроль данных	Обнаружены неверные данные (несуществующий код)
5	Переполнение	Канал не может ответить на требование ВУ в отведенные ему промежутки времени, т.е. суммарная скорость потока данных в канале превышает пропускную способность канала

Таблица I3

Разряд ССК	Идентификатор	Название	Условия установки
40	S ₀	Программно управляемое прерывание (ПУП)	В КСК установлен признак ПУП
41	S ₁	Неверная длина (НД)	Поле счетчика КСК не равно длине блока информации на внешнем носителе и в КСК не установлен признак ПИД
42	S ₂	Контроль программы (КП)	Неверно АСК или КСК (например, разряды 4-7 АСК не равны 0, или указан несуществующий код канала)
43	S ₃	Контроль защиты (КЗ)	Ключ защиты, указанный в АСК, не сравнился с ключом основной памяти
44	S ₄	Контроль данных канала (КДК)	В канале обнаружен байт с неверной четностью. Канал исправляет контрольный разряд при записи информации в основную память
45	S ₅	Контроль управления канала (КУК)	Неверный сигнал управления обнаружен в канале. Подканал или ВУ неисправны,
46	S ₆	Контроль управления интерфейса КРИ (Контроль работы интерфейса)	В ВУ или УВУ, подключенном к подканалу, произошла неисправность
47	S ₇	Контроль зацепления	При выполнении операций (СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ и УТОЧНИТЬ СОСТОЯНИЕ) первый байт данных после зацепления представлен ВУ раньше, чем канал может его обработать. ВУ передает данные быстрее, чем канал может принимать их

3.9.8. Прерывание по вводу-выводу.

При помощи прерывания по вводу-выводу канал сообщает ВЧУ, что он просит внимания. Запрос на прерывание по вводу-выводу формируется, если:

- канал закончил канальную программу;
- ВУ закончило свою операцию ввода-вывода;
- в ВУ произошло необычное условие;
- в КСК указан признак ПУП.

Когда возникает запрос на прерывание по вводу-выводу, в ячейку 56 постоянно-распределенной области основной памяти записывается в качестве старого ССП ввода-вывода текущее ССП выполняемой программы с кодом прерывания, содержащим адрес канала и внешнего устройства, затем ВЧУ считывает из ячейки I20 новое ССП ввода-вывода и начинает обслуживать прерывание. При этом также запоминается ССК в ячейке 64.

Для управления прерываниями в ССП используются разряды маски системы. Разряды 0, 1, 2 ССП соответствуют каналам КМ, КС1 и КС2. Если разряд маски, соответствующий каналу, установлен в "0", то прерывание от данного канала замаскировано, а если установлен в "1", то каналу разрешено вызвать прерывание. Если канал замаскирован, он не может вызвать прерывания и запрос на прерывание остается "висеть" в канале. Когда ВЧУ принимает запрос на прерывание по вводу-выводу, этот запрос в соответствующем канале гасится.

Так как прерывание по вводу-выводу запоминает ССК, необходимо маскировать все каналы, когда выполняется команда НАЧАТЬ ВВОД-ВЫВОД или ОСТАНОВИТЬ ВВОД-ВЫВОД, из-за того, что при их выполнении может быть запомнена часть ССК. Если при этом будет разрешено прерывание по вводу-выводу, то начальное значение байтов состояния, запомненное при выполнении команды, может быть потеряно. Когда происходит обработка прерывания по вводу-выводу, все каналы должны быть также замаскированы до тех пор, пока программа не обработает содержимое ССК.

Для того, чтобы предотвратить прерывание по вводу-выводу, нужно замаскировать канал, но присутствие висящего прерывания препятствует выполнению дальнейших команд НАЧАТЬ ВВОД-ВЫВОД. Можно использовать команду ПРОВЕРИТЬ ВВОД-ВЫВОД для того, чтобы сбросить запрос на прерывание в канале. Если при выполнении команды ПРОВЕРИТЬ ВВОД-ВЫВОД запоминается полное ССК, устанавливается в ССП код условия ОI и запрос на прерывание в канале сбрасывается.

Обычная операция ввода-вывода выполняется следующим образом:

программа ВЧУ выдает команду НАЧАТЬ ВВОД-ВЫВОД, которая устанавливает код условия ОО; маскируется прерывание по вводу-выводу на время выполнения команды НАЧАТЬ ВВОД-ВЫВОД; ВЧУ продолжает работу одновременно с выполнением канальной программы;

в конце выполнения канальной программы канал формирует запрос на прерывание по вводу-выводу. Программа, обслуживающая прерывание, определяет, как окончена операция. Если при ее выполнении произошла ошибка (сбой) - программа принимает соответствующие меры и устанавливает соответствующие признаки;

когда программа ВЧУ достигает такой точки, начиная с которой она не может "двигаться" дальше без считывания затребованной с внешнего устройства информации, ВЧУ проверяет, закончена ли операция ввода. Если операция закончена, ВЧУ использует введенные данные, или, если требуется, начинает новую операцию ввода-вывода;

если операция ввода-вывода еще не закончена, ВЧУ переводится в состояние "Ожидание".

Рабочая программа всегда выполняется в состоянии "Задача" и не может выполнять собственных команд управления каналами. Имеется специальная системная программа, которая выполняет эти функции. Связь между рабочей и системной программами осуществляется при помощи команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ.

ССК записывается в ячейку 64 постоянно распределенной области основной памяти при выполнении операций ввода-вывода. Некоторые операции могут изменять только часть ССК (байты состояния канала и ВУ), сохраняя значение остальных полей.

Операция ввода-вывода начинается при условии, что адресуемое ВУ и подканал находятся в доступном состоянии. Если операция не может быть принята подканалом ВУ, то состояние адресуемого ВУ запоминается в ССК.

Для анализа состояния канала, подканала и ВУ используются команды ПРОВЕРИТЬ ВВОД-ВЫВОД и ПРОВЕРИТЬ КАНАЛ, имеющие формат SI.

При выполнении команд управления каналами производится опрос состояний канала, подканала и ВУ и формируется код условия для записи его в ССП.

В табл. I4 приведены значения кода условия для всех 4 команд управления каналами.

Таблица I4

Команды управления каналами	Код условия			
	00	01	10	11
Начать ввод-вывод	Доступно	Запомнено ССК	Занято	Неработоспособно
Проверить ввод-вывод	Не работает	ССК готово	Работает	"
Остановить ввод-вывод	" ~ "	Остановлено	Стоит	"
Проверить канал	Доступен	-	-	"

3.10. Пульт управления.

Пульт управления предназначен для управления работой ЭВМ ЕС-1020. Пульт управления обеспечивает выполнение следующих основных функций: приведение ЭВМ в исходное состояние (сброс), занесение информации в оперативную память и регистры, индикацию информации, находящейся в оперативной памяти, регистрах и ССП, и первоначальную загрузку программы.

3.10.1. Приведение ЭВМ в исходное состояние.

Приведение ЭВМ в исходное состояние заключается в приведении в исходное состояние ВЧУ, каналов, неавтономных неразделенных УВУ и ВУ.

При сбросе ВЧУ переводится в состояние "Останов", а все ждущие запросы на прерывание уничтожаются. Исправляется четность всей информации, содержащейся в оперативной памяти. Все индикаторы ошибок устанавливаются в нуль.

Автономные УВУ не сбрасываются. В разделенном УВУ или разделенном ВУ сигнал сброса из ВЧУ устанавливает в исходное состояние только те функциональные части, которые относятся к данному ВЧУ. Сброс производится также, когда иницируется первоначальная загрузка программы, или когда осуществляется последовательность действий по включению питания.

3.10.2. Запоминание и индикация.

Функция запоминания и индикации позволяет вручную вмешиваться в процесс выполнения программы. Используя органы управления пульта, можно перевести ВЧУ в состояние "Останов", а затем занести или индигировать информацию в оперативной памяти и некоторых внутренних регистрах ВЧУ и каналов. Состояние "Останов" достигается в том случае, если задана работа ВЧУ по команде, при достижении предварительно заданного адреса или при нажатии кнопки ОСТАНОВ. После того, как нужные действия произведены, можно снова пустить ВЧУ. Останов и пуск ВЧУ, исключая фактор времени, сами по себе никаких изменений в выполнение программы не вносят.

Сбои оборудования ЭВМ, возникшие во время действий по запоминанию и индикации, не вызывают немедленного прерывания, но вызывают появление запроса на прерывание, ожидающего обработки. Прерывание, если оно не замаскировано, будет обработано, если ВЧУ окажется в состоянии "Работа".

3.10.3. Первоначальная загрузка программы.

Первоначальная загрузка программы (ПЗП) производится для того, чтобы иницировать работу в том случае, когда содержимое памяти или ССП оказываются неподходящими для дальнейшей работы.

Для того, чтобы иницировать ПЗП вручную, с помощью переключателей указывается устройство ввода и затем нажимается кнопка ЗАГРУЗКА. При нажатии этой кнопки ЭВМ приводится в исходное состояние, загорается лампочка ЗАГРУЗКА и затем производится операция чтения в выбранном устройстве ввода.

Если ввод завершается успешно, то оказывается занесенным новое ССП, ВЧУ запускается, а лампочка ЗАГРУЗКА гасится.

3. II. Устройство питания.

Устройство питания процессора (УПП) (рис. I6) обеспечивает процессор ЕС-2020 необходимыми постоянными и переменными напряжениями и осуществляет управление электропитанием устройств не ниже уровня УВУ, входящих в состав ЕС-1020. УПП осуществляет дистанционное последовательное включение электропитания подключенных к процессору устройств не ниже уровня УВУ и дистанционное одновременное отключение (в том числе и аварийное) электропитания всех устройств, включая ВУ.

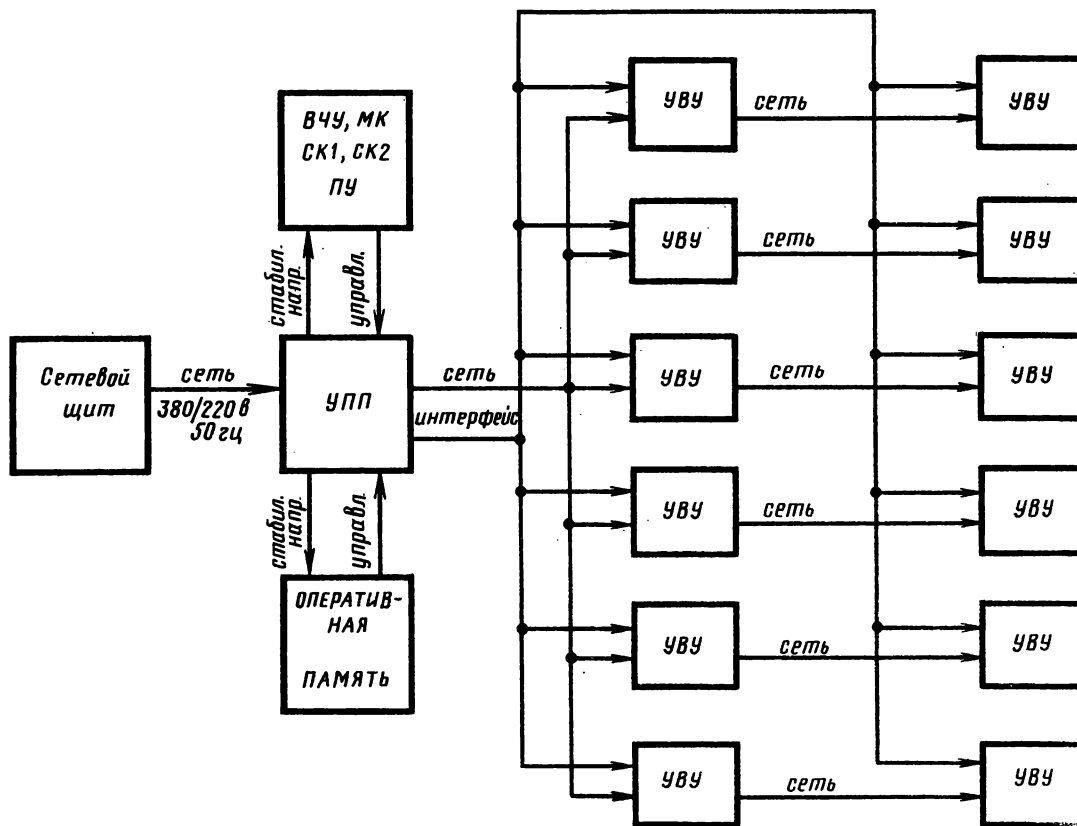


Рис. I6. Структурная схема электропитания

Питание УПП осуществляется от трехфазной сети переменного тока напряжением 380/220 в с допустимыми отклонениями от +10 до -15% и частотой 50 ± 1 гц.

Параметры блоков УПП, вырабатывающих постоянные стабилизированные напряжения, приведены в табл. I5.

УПП вырабатывает переменные напряжения, равные 6 в и 380/220 в, используемые соответственно для питания счетчика времени в ВЧУ и вентиляторов.

УПП имеет защиту входной сети от коротких замыканий и перегрузок, которая осуществляется с помощью автомата АК50К - ЗМГ.

УПП обеспечивает выдачу в ВЧУ сигнала, предупреждающего аварийное отключение стабилизированных источников питания в случае понижения напряжения на их выходе не более чем на 5% от номинальной величины из-за аварийного состояния сети питания УПП. По этому сигналу ВЧУ предпринимает действия, обеспечивающие сохранение информации в ОП при аварийном отключении питания.

Мощность, потребляемая процессором, не превышает 7 кВа.

Рабочее заземление внутри процессора выполнено с помощью проводов суммарным сечением 250 мм^2 , а между процессором и УВУ - экраном кабелей сигнальных связей. Защитное заземление выполнено путем соединения корпусных клемм всех устройств. Рабочая и защитная "земли" объединяются в одной точке в УПП.

Таблица 15

Шифр блока	Стабилизированное напряжение, В	Потребитель	Максимальный ток нагрузки	Пределы срабатывания защиты по превышению напряжения
ЕС-0901/0011	+5	ВЧУ, МК, СК1, СК2, ПП, блок защиты	18	$5,8 \pm 0,2$
ЕС-0901/0002	-5	оперативная память	3	$5,8 \pm 0,2$
	+5	оперативная память		
ЕС-0901/0022	-5	ВЧУ, ПУ, ПП	5	15+1
	+12,6	ПП		
ЕС-0901/0012	-12,6	оперативная память	14	$8 \pm 0,5$
	+6,3	ПУ		
ЕС-0901/0008	+20	оперативная память	10	24+1

Примечание. Источники питания имеют защиту от понижения выходного напряжения до уровня 20-40% от номинальной величины и защиту от коротких замыканий и перегрузок.

3.12. Конструкция процессора (ЕС-2020).

Устройство ЕС-2020 (рис. 17) состоит из трех устройств:

- вычислителя (ЕС-2420) 1;
- оперативной памяти (ЕС-3220) 3;
- устройства питания (ЕС-0820) 2.

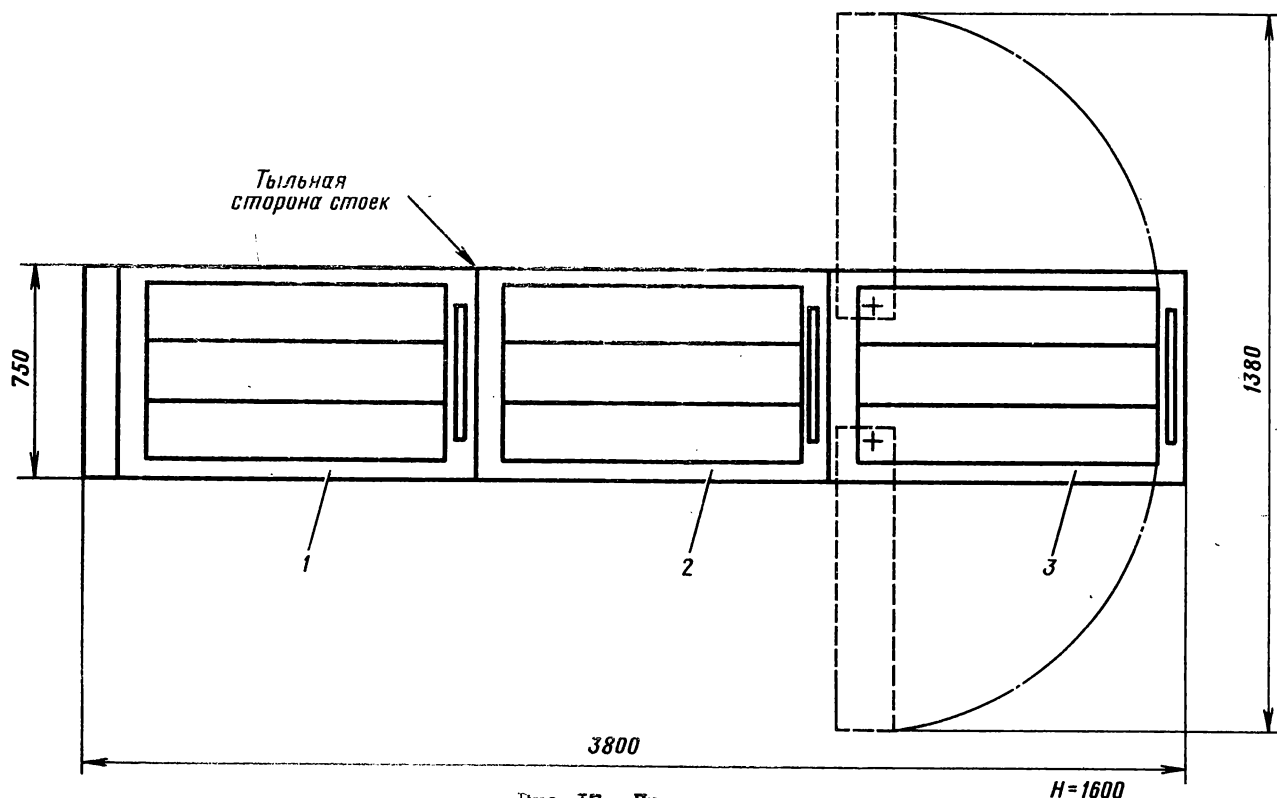


Рис. 17. Процессор

Устройство ЕС-0820, расположенное между двумя другими устройствами, подключено к ним через кабели и шины земли.

Устройства ЕС-2420 и ЕС-3220 также соединены между собой кабелями.

Конструктивно процессор выполнен в виде трех приборных стоек с габаритными размерами 1600x1200x750 мм.

На подвижной боковой обшивке стойки ЕС-2420 укреплен инженерный пульт ЕС-2420/НО15.

Внутри стойки расположены три рамы, две из которых подвижные и одна стационарная. Подвижные рамы в закрытом положении запираются замками.

Подвижные рамы установлены так, что монтаж их расположен у внешних сторон стойки.

Доступ к ТЭЗ, как на подвижных рамах, так и на неподвижной раме обеспечивается после открывания подвижных рам.

Для обеспечения работы процессора в заданном интервале температур в рамках процессора предусмотрены места для установки блоков вентиляторов (2 шт. на раму).

В ЕС-3220 на подвижных рамах предусмотрена возможность установки магнитных блоков ЕС-3220/КО01 в количестве до 4 шт.

4. УСТРОЙСТВО И РАБОТА ПРОЦЕССОРА

Процессор состоит из вычислительного устройства, одного мультиплексного и двух селекторных каналов, пульта управления, оперативной памяти и устройства питания. Вычислительное устройство включает в себя устройство управления, блок регистров (БР) и арифметическо-логический блок (БА). Перечисленные выше функциональные части процессора (рис. 18) соединены общими шинами, по которым осуществляется передача информации между ними.

В устройстве управления используется микропрограммный принцип работы. Микропрограммное устройство управления предназначено для управления процессами обработки информации в процессоре ЕС-2020.

Основной частью микропрограммного устройства управления является блок постоянной памяти (ПП), который используется для хранения управляющей информации, представляющей собой совокупность микропрограмм. (Описание ПП см. в Е13.055.001 Т05). Управляющие сигналы, под воздействием которых выполняются необходимые преобразования информационных сигналов, вырабатываются путем считывания из ПП микрокоманд, составляющих микропрограммы, и последующей их дешифрации. Управляющие сигналы распределяются по электронным блокам вычислительного устройства, оперативной памяти, каналов и связанных с ними ВУ. Адрес читаемой из ПП микрокоманды задается блоком управления (БУ) микропрограммного устройства управления.

В процессе выполнения микропрограмм порождается различного рода информация, отражающая состояние выполняемой программы и состояние отдельных блоков процессора. Для хранения этой информации используются адресные и служебные регистры, а также регистры общего назначения, составляющие блок регистров (БР). Занесение информации в регистры и извлечение ее из регистров осуществляется в основном арифметическо-логическим блоком БА под управлением микропрограммного устройства управления.

БА обладает широким набором элементарных операций и обрабатывает адреса, команды и операнды, а также осуществляет пересылку информации между регистрами процессора. БА обрабатывает информацию по одному байту. БА связан с остальными блоками процессора системой трех девятиразрядных шин: двух входных А и В и выходной шины С (Описание БУ, БА и БР см. в Е13.055.001 Т01). В процессоре имеются также кодовая шина адреса КША и кодовая шина чтения КШЧ, по которым осуществлена связь с оперативной памятью.

Оперативная память включает в себя три типа памяти - основную, локальную и мультиплексную. Для всех трех типов памяти используется один адресный и один информационный регистр.

Основная память доступна программам и используется для хранения программ и данных. Локальная память состоит из 16 общих регистров и 4 регистров с плавающей запятой, доступных программам. Остальная часть локальной памяти используется для хранения различной служебной и диагностической информации и не может быть изменена с помощью программ. Мультиплексная память также

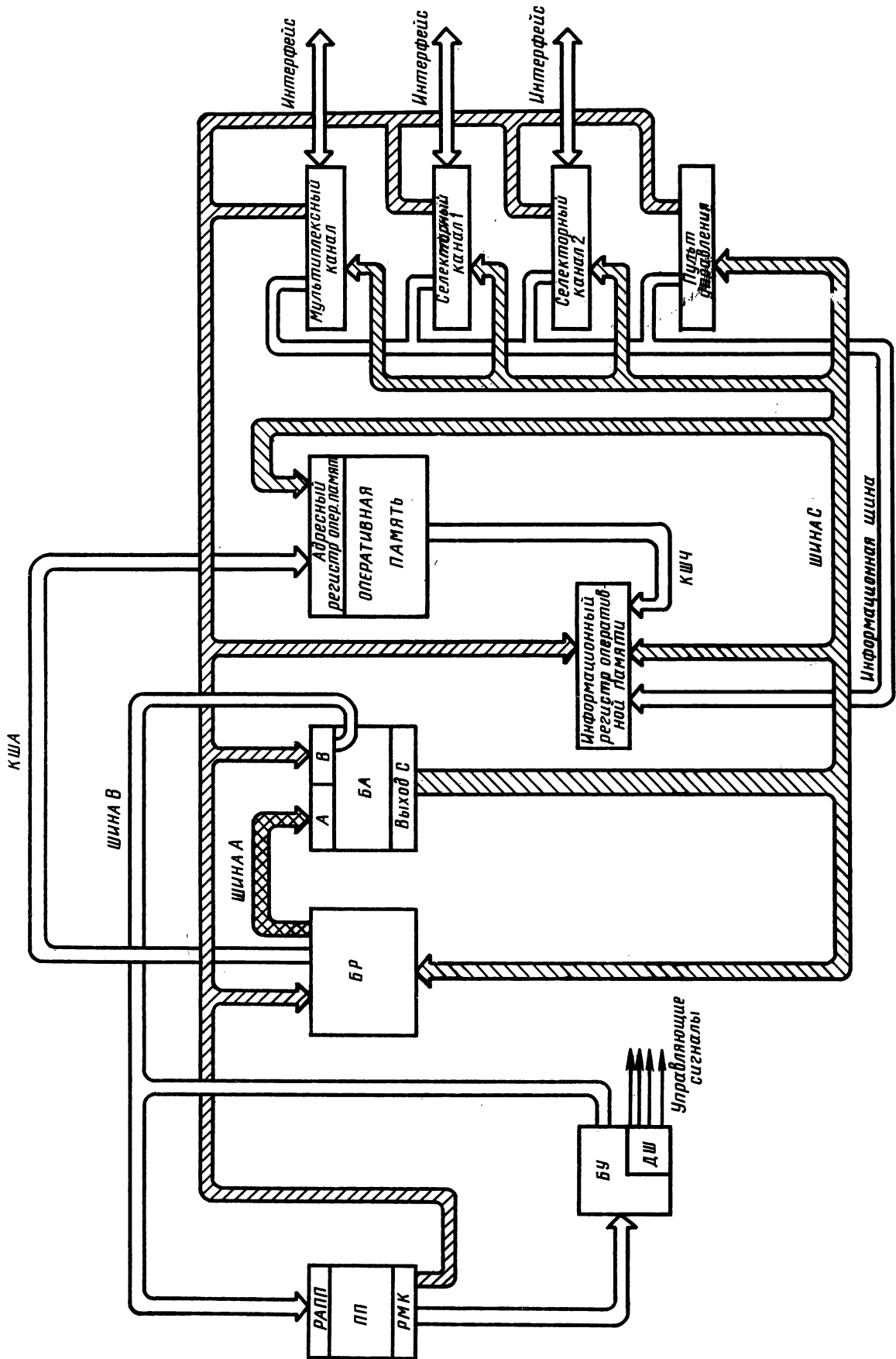


Рис. 18. Структура процессора

недоступна программам и хранит информацию, управляющую операциями обмена с внешними устройствами через мультиплексный канал. (Описание оперативной памяти см. в Е13.061.013 ТО, Информационный регистр оперативной памяти описан в Е13.055.001 ТО1).

Мультиплексный канал позволяет организовать обмен информацией между основной памятью и внешними устройствами с относительно малой и средней скоростью передач данных. Подканалы мультиплексного канала реализованы в виде мультиплексной памяти. Мультиплексный канал может работать в мультиплексном или монопольном режиме. Для обслуживания вводов-выводов операций в мультиплексном канале используются оборудования других блоков процессора (Описание мультиплексного канала см. в Е13.055.001 ТО2).

Селекторные каналы обеспечивают сопряжение с основной памятью ВУ с относительно высокой скоростью передачи информации, таких, например, как накопители на магнитной ленте, дисках, барабанах. Каждый селекторный канал может работать только в монопольном режиме. Селекторные каналы почти независимы от остального оборудования процессора (Описание селекторных каналов см. в Е13.055.001 ТО3).

Каналы и ВЧУ работают параллельно. На выборку и занесение данных в основную память ВЧУ не влияет одновременная передача данных по какому-либо каналу. При одновременной работе ВЧУ и каналов в первую очередь обслуживаются запросы первого селекторного канала, затем, в порядке понижения приоритетов, запросы второго селекторного канала, мультиплексного канала и ВЧУ.

Каналы связаны с остальными блоками процессора через шины В и С, а также через информационную шину, соединяющую каналы и информационный регистр оперативной памяти. Таким же образом связан с блоками процессора и пульт управления (ПУ).

ПУ предназначен для управления работой ЭВМ ЕС-1020. Он обеспечивает выполнение следующих основных функций: приведение ЭВМ в исходное состояние; занесение информации в основную, локальную и мультиплексную памяти, в регистры блока регистров и каналов; индикацию информации; первоначальную загрузку программы (описание ПУ см. в Е13.055.001 ТО7).

Питание всех блоков процессора осуществляется УПП. На рис. 16 изображена структурная схема разводки электропитания.

Напряжение трехфазной сети от сетевого щита поступает на УПП. Выработанные УПП напряжения поступают в оперативную память, ВЧУ, каналы и пульт управления.

Для разводки первичной сети по УВУ УПП имеет 6 разъемов с напряжением 380/220 В. К каждому разъему подключается по одному УВУ. Подключение дополнительных УВУ (сверх 6) осуществляется посредством выводных разъемов, с напряжением 380/220 В каждого из УВУ.

УПП имеет 24 разъема с сигналами интерфейса управления электропитанием устройств не ниже уровня УВУ.

Интерфейс управления электропитанием осуществляет:

дистанционное последовательное включение электропитания всех УВУ, причем очередность включения данного УВУ определяется номером разъема интерфейса на УПП, к которому подключено данное УВУ;

дистанционное одновременное отключение (в том числе аварийное) всех ВУ.

В шкафу УПП имеется тумблерный набор, позволяющий централизованно задать состав включаемых УВУ, а также индивидуально отключать устройства в случае необходимости.

Централизованное включение и отключение питания всех устройств производится при нажатии кнопок ПИТАНИЕ ВКЛ. и ПИТАНИЕ ВЫКЛ. на пульте управления. (Описание УПП см. в Е12.087.031 ТО).

Эффективное выполнение функций процессора и ЭВМ в целом возможно только при решении задач под управлением операционной системы. Операционная система обеспечивает подготовку программ и обслуживание процесса их выполнения. Программы операционной системы позволяют эффективно управлять всеми ресурсами вычислительной системы, планируя работу ЭВМ и обрабатывая сигналы прерывания, возникающие в процессе выполнения программ.

В связи с применением в процессоре микропрограммного принципа управления обработкой информации, часть алгоритмов работы процессора реализуется микропрограммно, т.е. с участием

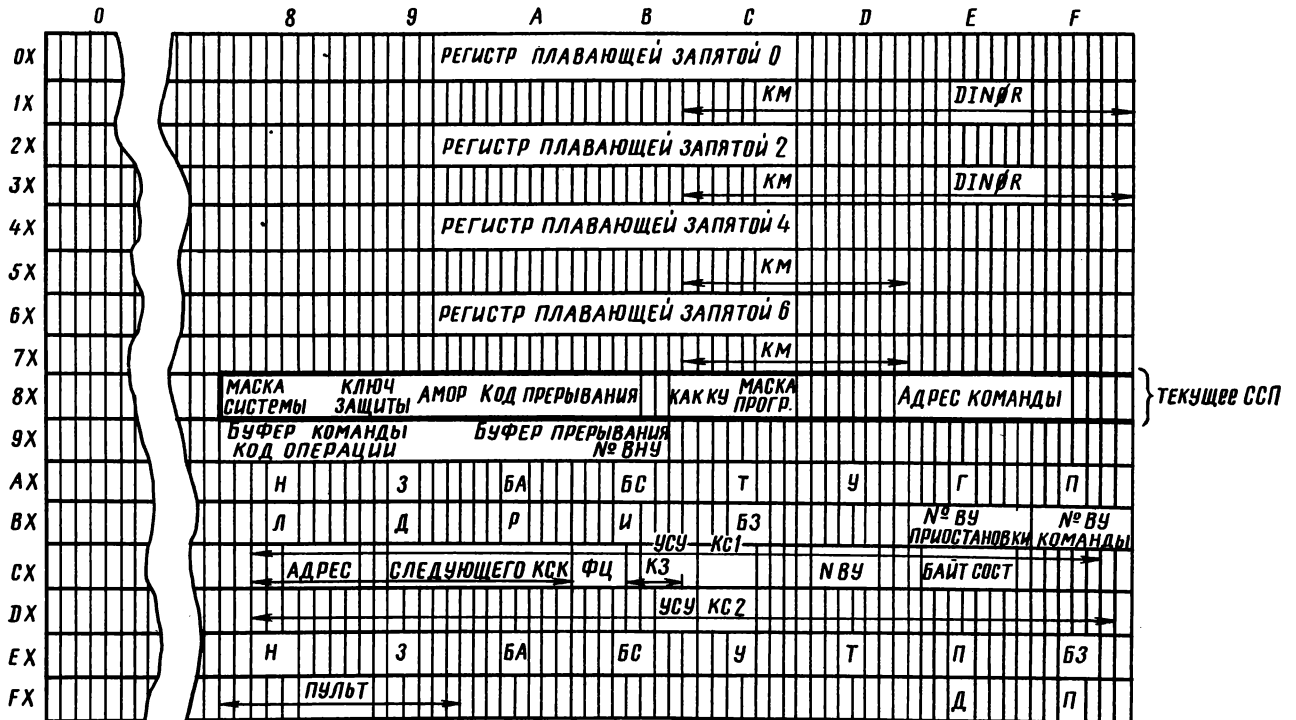
микропрограммного устройства управления, а часть аппаратно, т.е. без участия микропрограммного устройства управления. Микропрограммно реализованы все команды системы команд машины. Описание этих микропрограмм приведено в Е13.055.001 Т05. Времена выполнения команд даны в приложении 3. Примененные в процессоре способы микропрограммной реализации процедур обработки прерываний, обслуживания вводов-выводных операций в мультиплексном канале и пультовых процедур изложены соответственно в Е13.055.001 Т01, Е13.055.001 Т02, Е13.055.001 Т07.

ПРИЛОЖЕНИЕ I

СПИСОК СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

АСК - адресное слово канала
БА - блок арифметическо-логический
БР - блок регистров
БУП - блок управления питанием
БУ - блок управления
В - база
ВВ - ввод-вывод
ВУ - внешнее устройство
ВЧУ - вычислительное устройство
D - смещение
ДКОИ - двоичный код для обработки информации
ДШ - дешифратор
КО - код операции
КОИ-8 - 8-битный код обмена информации
КСК - командное слово канала
КША - кодовая шина адреса
КШЧ - кодовая шина чтения
L - длина
ЛП - локальная память
МП - мультиплексная память
ОП - основная память
ПП - постоянная память
ПЗП - первоначальная загрузка программы
ПУ - пульт управления
R - регистр
RR - форматный код команды типа регистр-регистр
RS - форматный код команды типа регистр-память без индексации
RX - форматный код команды типа регистр-память с индексацией
S - память
SI - форматный код команды типа память - непосредственный операнд
SS - форматный код команды типа память-память
СП - слово состояния программы
СК - слово состояния канала
УВУ - устройство управления внешними устройствами
УПП - устройство питания процессора
X - индекс
РАПП - регистр адреса ПП
РМК - информационный регистр ПП

Распределение локальной памяти



ПРИЛОЖЕНИЕ 3

ВРЕМЕНА ВЫПОЛНЕНИЯ КОМАНД И ИНСТРУКЦИЙ

Обозначения

- В - общее количество обработанных байтов первого операнда
- Н - количество значащих шестнадцатиричных цифр (исключая впереди стоящие нули) в двоичном операнде
- N - общее количество байтов первого операнда для тех инструкций, в которых оба операнда имеют одинаковую длину
- N_1 - общее количество байтов первого операнда
- N_2 - общее количество байтов второго операнда
- D - количество значащих десятичных цифр
- N_{\min} - наименьшее из N_1 и N_2
- N_{abc} - $(N_1 - N_2)$
- S - число знаков в редактируемом поле
- Z - общее число символов начала значимости и выбора цифры в шаблоне редактирования
- R - общее число разделителей полей в шаблоне редактирования
- OTM - число раз занесения адреса в регистр I
- A - равно 0, если регистр I не изменяется;
равно I, если в регистр I заносится адрес
- V - количество регистров, загружаемых или записываемых в память.

Примечание. В табл. 16-20 времена выполнения приводятся с учетом базирования адресов и без учета индексирования (для команд формата RX). Время базирования одного адреса - 4 мксек. Время индексирования адреса - 4 мксек.

Стандартная система команд

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ВЫПОЛНИТЬ	EX	RX	33 + время выполнения подчиненной команды
ВЫЧИТАНИЕ	SR	RR	20
ВЫЧИТАНИЕ	S	RX	33
ВЫЧИТАНИЕ КОДОВ	SLR	RR	22
ВЫЧИТАНИЕ КОДОВ	SL	RX	35
ВЫЧИТАНИЕ ПОЛУСЛОВА	SH	RX	29
ДЕЛЕНИЕ	DR	RR	390
ДЕЛЕНИЕ	D	RX	398
ЗАГРУЗКА	LR	RR	18
ЗАГРУЗКА	L	RX	27
ЗАГРУЗКА АДРЕСА	LA	RX	24
ЗАГРУЗКА ГРУППОВАЯ	LM	RS	18+8 V
ЗАГРУЗКА ДОПОЛНЕНИЯ	LCR	RR	21
ЗАГРУЗКА И ПРОВЕРКА	LTR	RR	21
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ	LNK	RR	23
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ	LPR	RR	23
ЗАГРУЗКА ПОЛУСЛОВА	LH	RX	28
ЗАПИСЬ В ПАМЯТЬ	ST	RX	27
ЗАПИСЬ В ПАМЯТЬ ГРУППОВАЯ	STM	RS	18+10 V
ЗАПИСЬ В ПАМЯТЬ ПОЛУСЛОВА	STH	RX	22
ЗАПИСЬ В ПАМЯТЬ СИМВОЛА	STC	RX	23
И	NR	RR	20
И	N	RX	30
И	NI	SI	19
И	NC	SS	36+3N, если оба адреса четные или нечетные, 28+5N в остальных случаях
ИЛИ	OR	RR	20
ИЛИ	O	RX	30
ИЛИ	OI	SI	19
ИЛИ	OC	SS	36+3N, если оба адреса четные или нечетные, 28+ N в остальных случаях
ИСКЛЮЧАЮЩЕЕ ИЛИ	XR	RR	20
ИСКЛЮЧАЮЩЕЕ ИЛИ	X	RX	30
ИСКЛЮЧАЮЩЕЕ ИЛИ	XI	SI	19
ИСКЛЮЧАЮЩЕЕ ИЛИ	XC	SS	36+3N, если оба адреса четные или нечетные, 28+5N в остальных случаях

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ОБРАЩЕНИЕ К СУПЕРВИЗОРУ	SVC	RR	59
ПЕРЕКОДИРОВАТЬ	TR	SS	34+10N
ПЕРЕКОДИРОВАТЬ И ПРОВЕРИТЬ	TRT	SS	57+10N, если все байт- функции нули, 44+9N - есть хотя бы одна ненулевая байт-функция
ПЕРЕСЫЛКА	MVI	SI	18
ПЕРЕСЫЛКА	MVC	SS	37+3N, если оба адреса четные или нечетные, 30+5N в остальных случаях
ПЕРЕСЫЛКА ЗОН	MVZ	SS	43,5+3,5N, если оба адреса четные или нечетные, 39,5N в остальных случаях
ПЕРЕСЫЛКА СО СДВИГОМ	MVO	SS	27+9N ₁ , если N ₁ ≤ N ₂ 32+6N ₂ + 3N ₁ , если N ₁ > N ₂
ПЕРЕСЫЛКА ЦИФР	MVN	SS	43,5+3,5N, если оба адреса четные или нечетные, 39,5N в остальных случаях
ПЕРЕХОД ПО СЧЕТЧИКУ	BCTR	RR	24
ПЕРЕХОД ПО СЧЕТЧИКУ	BCT	RX	39
ПЕРЕХОД ПО ИНДЕКСУ БОЛЬШЕ	BXH	RS	42
ПЕРЕХОД ПО ИНДЕКСУ МЕНЬШЕ ИЛИ РАВНО	BXLE	RS	42
ПЕРЕХОД С ВОЗВРАТОМ	BALR	RR	27
ПЕРЕХОД С ВОЗВРАТОМ	BAL	RX	33
ПРЕОБРАЗОВАНИЕ В ДВОИЧНУЮ	CVB	RX	50+26D
ПРЕОБРАЗОВАНИЕ В ДЕСЯТИЧНУЮ	CVD	RX	54+23N+4N ²
ПРОВЕРИТЬ ПО МАСКЕ	TM	SI	19
ПРОВЕРИТЬ И УСТАНОВИТЬ	TS	SI	21
ПРОЧИТАТЬ СИМВОЛ	IC	RX	22
РАСПАКОВАТЬ	UNPK	SS	4I+4N ₁ , при N ₁ > 1; 34 при N ₁ = 1
СДВИГ ВЛЕВО	SLA	RS	62
СДВИГ ВЛЕВО ДВОЙНОЙ	SLDA	RS	93
СДВИГ ВЛЕВО ДВОЙНОЙ КОДОВ	SLDL	RS	80
СДВИГ ВЛЕВО КОДОВ	SLL	RS	55
СДВИГ ВПРАВО	SRA	RS	54
СДВИГ ВПРАВО ДВОЙНОЙ	SRDA	RS	83
СДВИГ ВПРАВО ДВОЙНОЙ КОДОВ	SRDL	RS	80
СДВИГ ВПРАВО КОДОВ	SRL	RS	52
СЛОЖЕНИЕ КОДОВ	ALR	RR	21
СЛОЖЕНИЕ КОДОВ	ASL	RX	34
СЛОЖЕНИЕ ПОЛУСЛОВА	AH	RX	29
СЛОЖЕНИЕ	AR	RR	20

Продолжение

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
СЛОЖЕНИЕ	A	RX	33
СРАВНЕНИЕ	CR	RR	20
СРАВНЕНИЕ	C	RX	31
СРАВНЕНИЕ КОДОВ	CLR	RR	13+3B
СРАВНЕНИЕ КОДОВ	CL	RX	25, если B=I 25+2B - в остальных случаях
СРАВНЕНИЕ КОДОВ	CLI	SI	19
СРАВНЕНИЕ КОДОВ	CLC	SS	67+5B
СРАВНЕНИЕ ПОЛУСЛОВА	CH	RX	29
УМНОЖЕНИЕ	MR	RR	338
УМНОЖЕНИЕ	M	RX	348
УМНОЖЕНИЕ ПОЛУСЛОВА	MH	RX	218
УПАКОВАТЬ	PACK	SS	$4I+4N_1$, при $N_1 > 1$; 34 при $N_1 = 1$
УСЛОВНЫЙ ПЕРЕХОД	BCR	RR	16
УСЛОВНЫЙ ПЕРЕХОД	BC	RX	29
УСТАНОВИТЬ МАСКУ ПРОГРАММЫ	SPM	RR	12
УСТАНОВИТЬ МАСКУ СИСТЕМЫ	SSM	SI	23

Таблица 17

Команды для операций с плавающей запятой

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	SWR	RR	73
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	SW	RX	54
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	SVR	RR	57
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	SV	RX	70
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	SDR	RR	98
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	SD	RX	107
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	SEB	RR	67
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	SE	RX	53
ДЕЛЕНИЕ (ДЛИННОЕ)	DDR	RR	2059
ДЕЛЕНИЕ (ДЛИННОЕ)	DD	RX	2069

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ДЕЛЕНИЕ (КОРОТКОЕ)	DER	RR	389
ДЕЛЕНИЕ (КОРОТКОЕ)	DE	RX	399
ЗАГРУЗКА (ДЛИННАЯ)	LDR	RR	31
ЗАГРУЗКА (КОРОТКАЯ)	LER	RR	19
ЗАГРУЗКА (ДЛИННАЯ)	LD	RX	40
ЗАГРУЗКА (КОРОТКАЯ)	LE	RX	27
ЗАГРУЗКА ДОПОЛНЕНИЯ (ДЛИННАЯ)	LCDR	RR	30
ЗАГРУЗКА ДОПОЛНЕНИЯ (КОРОТКАЯ)	LCER	RR	19
ЗАГРУЗКА И ПРОВЕРКА (ДЛИННАЯ)	LTDR	RR	31
ЗАГРУЗКА И ПРОВЕРКА (КОРОТКАЯ)	LTER	RR	19
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (ДЛИННАЯ)	LNDR	RR	31
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (КОРОТКАЯ)	LNER	RR	19
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (ДЛИННАЯ)	LPDR	RR	31
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (КОРОТКАЯ)	LPER	RR	19
ЗАПИСЬ В ПАМЯТЬ (ДЛИННАЯ)	STD	RX	38
ЗАПИСЬ В ПАМЯТЬ (КОРОТКАЯ)	STE	RX	29
ПОПОЛАМ (ДЛИННОЕ)	HDR	RR	75
ПОПОЛАМ (КОРОТКОЕ)	HER	RR	45
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	AWR	RR	79
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	AW	RX	82
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	AUR	RR	55
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	AU	RX	65
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	ADR	RR	93
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	AD	RX	107
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	AER	RR	62
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	AE	RX	70
СРАВНЕНИЕ (ДЛИННОЕ)	CDR	RR	58
СРАВНЕНИЕ (ДЛИННОЕ)	CD	RX	72
СРАВНЕНИЕ (КОРОТКОЕ)	CEB	RR	44
СРАВНЕНИЕ (КОРОТКОЕ)	CE	RX	52
УМНОЖЕНИЕ (ДЛИННОЕ)	MDR	RR	1239
УМНОЖЕНИЕ (ДЛИННОЕ)	MD	RX	1248
УМНОЖЕНИЕ (КОРОТКОЕ)	MEB	RR	489
УМНОЖЕНИЕ (КОРОТКОЕ)	ME	RX	498

Таблица 18

Команды десятичной арифметики

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ВЫЧИТАНИЕ ДЕСЯТИЧНОЕ	SP	SS	$70+3,2N_{\min}+2,2N_{\text{abs}}+0,2N_1$
ДЕЛЕНИЕ ДЕСЯТИЧНОЕ	DP	SS	$2(N_1-N_2)(100+19N_2)$
ОТРЕДАКТИРОВАТЬ	ED	SS	$46+7N+2,5Z+R+5N_2-0,5S$
ОТРЕДАКТИРОВАТЬ И ОТМЕТИТЬ	EDMK	SS	$42+7N+2,5Z+R+5N_2-0,5S+$ $+50TM+3A$
СЛОЖЕНИЕ ДЕСЯТИЧНОЕ	AP	SS	$74+3,2N_{\min}+2,2N_{\text{abs}}+0,2N_1$
СЛОЖЕНИЕ С ОЧИСТКОЙ	ZAP	SS	$65+3N_{\min}+2,5N_{\text{abs}}$, если оба адреса четные или нечетные; $70+4,5N_{\min}+2,5N_{\text{abs}}$ - в остальных случаях
СРАВНЕНИЕ ДЕСЯТИЧНОЕ	CP	SS	$51+2N_1+2N_2$
УМНОЖЕНИЕ ДЕСЯТИЧНОЕ	MP	SS	$109+9N_1+9N_2+(2N_2-I)$ $(27+6N_1-3N_2)$

Таблица 19

Команды управления каналами

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
НАЧАТЬ ВВОД-ВЫВОД	SIO	SI	I66 - мультиплексный канал I40 - селекторный канал
ОСТАНОВИТЬ ВВОД-ВЫВОД	HIO	SI	66 - мультиплексный канал 50 - селекторный канал
ПРОВЕРИТЬ ВВОД-ВЫВОД	TIO	SI	I10 - мультиплексный канал I00 - селекторный канал
ПРОВЕРИТЬ КАНАЛ	TCH	SI	24

Таблица 20

Команды прямого управления

Команды			Время выполнения, мксек
Название	Мнем.	Формат	
ПРЯМАЯ ЗАПИСЬ	WED	SI	20
ПРЯМОЕ ЧТЕНИЕ	RDD	SI	21

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	изъятых					