



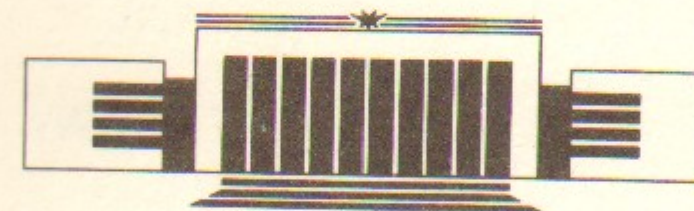
25  
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР

Г.А. Аксенов, А.В. Кислицин, Ю.И. Мерзляков,  
В.Я. Сазанский, И.А. Ткаченко, А.Г. Чертовских

**УНИВЕРСАЛЬНЫЙ АРИФМЕТИЧЕСКИЙ  
ПРОЦЕССОР АП-32**

1. Архитектура, система команд,  
технические характеристики

ПРЕПРИНТ 89-175



НОВОСИБИРСК

Универсальный арифметический  
процессор АП-32

1. Архитектура, система команд,  
технические характеристики

*Г.А. Аксенов, А.В. Кислицин, Ю.И. Мерзляков,  
В.Я. Сазанский, И.А. Ткаченко, А.Г. Чертовских*

Институт ядерной физики  
630090, Новосибирск 90, СССР

АННОТАЦИЯ

Этим препринтом мы начинаем серию публикаций об универсальном быстродействующем арифметическом процессоре АП-32. Он разработан для применения в системах обработки физической информации и обладает производительностью 10 MIPS. В настоящем препринте описаны принципы организации архитектуры процессора, система команд, технические характеристики.

ВВЕДЕНИЕ

Проблема обработки физической информации с детекторов элементарных частиц в физике высоких энергий имеет давнюю историю. В ИЯФ СО АН она стала особенно острой в 1982—1984 г. с введением в эксплуатацию ускорителя ВЭПП-4 и магнитного детектора МД-1 [1]. В рамках программы обработки экспериментов был спроектирован спецпроцессор АП-20 с быстродействием около 10 MIPS [2]. На его основе в 1984 г. была создана высокопроизводительная система для обработки физической информации [3]. Новое поколение детекторов [4—7] с существенно улучшенными параметрами предъявляет более высокие требования и к системам обработки данных. В связи с этим в ИЯФ в 1985 г. была начата и в 1988 г. завершена разработка арифметического процессора АП-32, который превосходит АП-20 по многим весьма важным техническим характеристикам (объему памяти, величине разрядной сетки и т. д.).

АП-32 — универсальный 32-разрядный высокопроизводительный (10 MIPS) арифметический процессор с собственным блоком управления и оригинальной системой команд. Построен на отечественной элементной базе и относится к классу современных супермини — ЭВМ. В АП-32 аппаратно реализованы операции сложения, умножения, деления 32-разрядных операндов в форме чисел с плавающей запятой, а также операции вычисления элементарных функций  $[\ln(x), \text{EXP}(x), \text{SIN}(x), \text{ARCTAN}(x), \text{SQRT}(x)]$ , что

позволяет выгодно использовать его и для научно-технических расчетов.

Хотя АП-32 — автономный процессор, однако в качестве основного канала ввода-вывода используются серийные ЭВМ семейства «Электроника» (Э-60.1, Э-100/25) и СМ. Сопряжение АП-32 с ними (рис. 1) осуществляется с помощью интерфейсов, специально

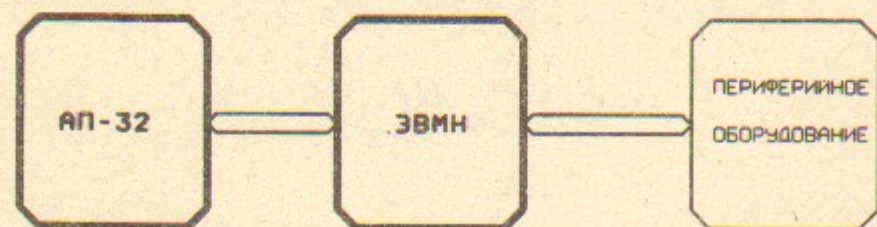


Рис. 1. Вычислительный комплекс АП—ЭВМ.

разработанных для этой цели. Такая конфигурация аппаратных средств позволяет осуществлять ввод в АП-32 обрабатываемой информации и вывод результатов из него путем обмена данными между АП-32 и ЭВМ, играющей роль машины «накачки» (ЭВМН). При этом большая вычислительная мощность АП-32 сочетается с развитым каналом ввода-вывода и удобной операционной системой.

Программа обработки данных в системе команд АП-32 вводится в процессор из ЭВМН, где реализованы программные средства для ее подготовки и отладки. Управление исполнением программы осуществляется самим АП-32 автономно.

В основу принципов протокола обмена, аппаратной и программной поддержки его положен подход, позволяющий рассматривать комплекс АП-ЭВМН как единую быстродействующую универсальную ЭВМ. Основные принципы следующие:

1. Процедуры разработки программ АП-32 принципиально не отличаются от аналогичных процедур на ЭВМН. Для этого используются кросс-средства ЭВМН под ее ОС (RT-11 или TSX-Plus).
2. Для пользователя процесс в АП-32 не отличается от процесса в ЭВМН: из проблемной программы АП-32 обеспечивается доступ к системным средствам и периферийным устройствам ЭВМН.

Представляется, что дальнейшее развитие вычислительного комплекса АП-ЭВМН может осуществляться в следующих направлениях:

- увеличение производительности (до нескольких десятков млн.

опер./с) за счет подключения к ЭВМН дополнительных процессоров АП-32 (многопроцессорный комплекс);

- применение более мощных современных моделей ЭВМ в качестве ЭВМН;
- перевод АП-32 на более совершенную элементную базу нового поколения с обеспечением программной совместимости с прототипом.

## 1. АРХИТЕКТУРА АП-32

### 1.1. Основные принципы

Принципы, положенные в основу архитектуры процессора АП-32, имеют много общего с идеологией ЭВМ с сокращенной системой команд (RISC):

- исполнение команды за один командный цикл;
- отказ от микропрограммного управления, что, безусловно, увеличивает трудности проектирования, но позволяет сократить длительность командного цикла;
- два типа памяти (данных и команд) позволяют совместить доступ к ним обеим в одном командном цикле (133 нс);
- большой набор регистров общего назначения.

В качестве основного схемотехнического метода проектирования принят модульный принцип структурной организации вычислительных систем: каждая арифметико-логическая операция осуществляется системой отдельных модулей (сопроцессоров), логически и электрически связанных между собой и центральным блоком управления шинами данных, команд, флагов и т. д.

Такой метод обладает рядом преимуществ:

- достаточно просто реализуются возможности одновременного исполнения вычислительных операций, поскольку почти каждая из них производится в отдельном модуле с автономным узлом управления;
- при наличии большого набора модулей возможна адаптация (в определенных пределах) вычислительных ресурсов системы к задачам пользователя;
- возможна модернизация модулей, как средство борьбы с моральным старением, в сторону расширения их функциональных возможностей без радикального пересмотра общей структуры процессора в целом.

Недостатком такого метода является то, что в данном случае требуется большее количество интегральных микросхем. Действительно, наряду с упрощением центрального блока управления во всех модулях оказывается неизбежным дублирование многих функций и однотипных узлов. Однако, это увеличение аппаратуры происходит не столь интенсивно и не требует, на наш взгляд, таких усилий, которые были бы необходимы в случае иного подхода.

## 1.2. Структурная схема АП-32

На структурной схеме (рис. 1.1) показаны основные модули АП-32:

- БУ — блок управления;
- ПЦЧ — процессор целых чисел;
- ОЗУД — память данных;
- ОЗУП — память программ;
- УМН — умножитель чисел в форме с плавающей запятой;
- СУМ — сумматор чисел в форме с плавающей запятой;
- СПЭФ — сопроцессор элементарных функций;
- ИНТ — интерфейс ЭВМ «Электроника».

Со стороны ЭВМН АП-32 можно рассматривать как мощный сопроцессор с собственным блоком управления и памятью. И нао-

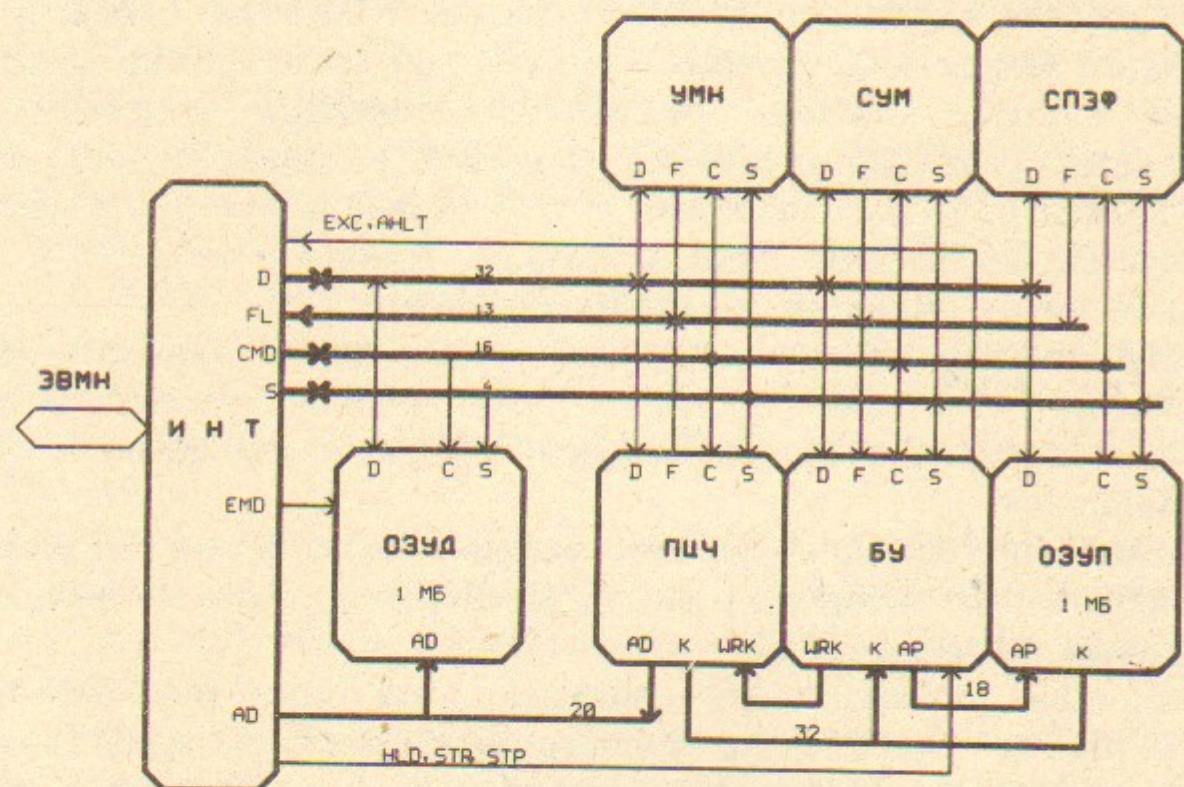


Рис. 1.1. Структурная схема АП-32.

борот, со стороны АП-32 ЭВМН также выглядит как сопроцессор АП-32 (процессор канала ввода-вывода).

Взаимодействие модулей АП-32 друг с другом, а также каждого из них с ОЗУД, обеспечивается с помощью системы шин (табл. 1.1) магистрали АП-32.

Таблица 1.1

№	Назначение	Обозначение	Кол-во	Функция
1	Шина данных	D	32	Передача данных
2	Шина команд ПЦЧ, БУ	K	32	Передача команд для ПЦЧ, БУ
3	Шина адреса памяти данных	AD	20	Передача адреса ОЗУД от ПЦЧ или ИНТ
4	Шина адреса памяти программ	AP	19	Передача от БУ адреса ОЗУП
5	Шина команд сопроцессоров	CMD	16	Передача команд для УМН, СУМ, СПЭФ от БУ или ИНТ
6	Шина флагов	FL	13	Передача признаков результата от ПЦЧ и сопроцессоров БУ и ИНТ
7	Шина стробирующих сигналов	S	4	Синхросигналы командного цикла
8	Шина управления	EMD, EXC, HLD, STR, STP, ANLT	6	Установка интерфейсом режимов работы АП-32

С помощью сигналов управления (HLD, STR, STP) ЭВМН через интерфейс может задать несколько режимов работы АП-32. Основные из них указаны в табл. 1.2.

Таблица 1.2

№	Обозначение	Режим
1	«АВТ»	Пуск программы АП-32 по адресу, содержащемуся в счетчике команд БУ. Управление магистралью и модулями передано БУ, который вырабатывает сигнал WRK=0 («подключен»).
2	«ЗАХВ»	Останов программы АП-32. БУ отключается от магистрали и вырабатывает сигнал WRK=1 («отключен»). Управление магистралью и модулями передано ИНТ.

Установка указанных режимов осуществляется загрузкой ЭВМН соответствующего управляющего слова в регистр состояний и управления интерфейса.

В режиме «АВТ» работа модулей АП-32 полностью определяется содержимым памяти программ. Извлекаемые из ОЗУП команды (32 разряда) по шинам К поступают в БУ и ПЦЧ, где и запоминаются в регистрах команд. При этом старшая часть слова команды через БУ передается на шины СМД (для сопроцессоров УМН, СУМ, СПЭФ). Одновременно с выборкой ОЗУП в БУ инициализируется генератор командного цикла, вырабатывающий синхросигналы на шинах S.

Все модули, имеющие доступ к шинам К или СМД, дешифрируют команды одновременно. Они исполняются только в тех модулях, которым предназначены. Так, например, в БУ исполняются только команды управления программой (JUMP, CALL и т. п.), а в ПЦЧ — арифметико-логические операции над операндами в форме чисел с фиксированной запятой.

Частным случаем указанного режима является режим пошагового исполнения команд («ИНИЦ»). Установка режима и пуск БУ на одну команду производится ЭВМН. В этом случае управление модулями передается БУ на промежуток времени, равный командному циклу. Исполнив текущую команду, БУ «отвечает» интерфейсу сигналом ЕХС.

Режим «ЗАХВ» используется как в профилактических целях (настройка, тестирование модулей), так и при вводе данных и выводе результатов расчета из АП-32 в ЭВМН. В этом режиме БУ блокируется и его роль исполняет интерфейс ЭВМН. Поскольку регистр команд интерфейса подключен только к шине команд СМД, то ЭВМН может управлять теми модулями (или регистрами модулей), которые имеют доступ к этой шине.

Команды обращения к ОЗУД, ОЗУП или другим модулям АП-32 в регистр команд записываются ЭВМН. С помощью регистра состояний и управления ею же инициализируется генератор синхросигналов (S) командного цикла интерфейса.

При обращении к ОЗУД адресная часть команды содержится в адресном регистре интерфейса, куда ее предварительно записывает также ЭВМН. Обмен данными между ИНТ и ОЗУД осуществляется через регистр данных командами типа память-регистр. Этими же командами можно произвести обмен и между другими модулями и ИНТ. Однако в этом случае ОЗУД должна быть отключена от магистрали (сигналом ЕМД).

Обращение ЭВМН к регистру данных интерфейса выполняется как под управлением ЦП ЭВМН (программный канал), так и под управлением самого ИНТ (канал прямого доступа к ОЗУ ЭВМН).

Таким образом, в режиме «ЗАХВ» возможности ЭВМН и аппаратных механизмов, заложенных в интерфейсе, обеспечивают обмен данными и статусной информацией между памятью ЭВМН и модулями АП-32. Это позволяет представлять их на видеотерминале в удобной для пользователя форме в любой, наперед заданной, точке программы АП-32. Поэтому ИНТ, как показано на рис. 1.1, имеет доступ ко всем основным шинам магистрали АП-32.

Используя терминологию обычных ЭВМ, можно сказать, что модули ПЦЧ, БУ и ОЗУП образуют собой центральный процессор АП-32, а УМН, СУМ, СПЭФ — процессор для чисел с плавающей запятой (ППЗ).

В ПЦЧ исполняются все арифметико-логические операции над 16- и 32-разрядными числами с фиксированной запятой (в модифицированном дополнительном коде). Он содержит массив из 32-х регистров общего назначения (РОН), в дальнейшем именуемых I-регистрами, арифметико-логическое устройство, сдвигатель и другие необходимые узлы. В режиме «АВТ» этим же модулем на шине АД формируется исполнительный адрес ОЗУД. В режиме «ЗАХВ» шина АД отключена от ПЦЧ, а адрес ОЗУД формируется ИНТ. Признаки результата операции (флаги N, Z, V, C) по шине FL поступают в БУ и ИНТ. Обмен данными между ОЗУД («АВТ»), ИНТ («ЗАХВ») и I-регистрами производится по шине D магистрали АП-32.

ППЗ представляет собой систему модулей-сoproцессоров, предназначенных для арифметико-логических операций над 32-разрядными данными, хранящимися в ОЗУД. Типы операций определяются самой структурой модуля, а исполнение их осуществляется под управлением собственного узла управления модуля. Число этих модулей, их типы не являются для АП-32 постоянными. **Допустима**, например, установка нескольких одинаковых устройств. Комбинируя модули (из имеющихся), можно создавать конфигурацию, максимально соответствующую реализуемому алгоритму вычисления.

В представленной версии АП-32 все модули (УМН, СУМ, СПЭФ) реализуют операции над числами с плавающей запятой с однократной точностью. Формат представления чисел совместим с форматом, принятым в семействе ЭВМ «Электроника».

Несмотря на то, что практически каждый сопроцессор является

весьма сложным устройством с собственным блоком управления, с точки зрения программиста он представляет собой некоторый набор 32-разрядных регистров. В дальнейшем мы будем называть их F-регистрами. Каждый F-регистр имеет свой адрес и, кроме того, специальное назначение. Именно то, что F-регистры специализированы, и отличает их от I-регистров. Ограничений на число регистров в одном модуле не предусматривается, однако общее количество F-регистров не должно превышать 32. Системой команд обеспечивается обмен данными F-регистров с ОЗУД и между собой. Как в автономном режиме, так и в режиме «ЗАХВ» такой обмен производится командами обращения к ОЗУД. Причем во втором случае («ЗАХВ») роль ОЗУД исполняет регистр данных ИНТ.

Системой команд не предусматривается непосредственная передача содержимого I-регистра F-регистру. Такой обмен возможен только через ОЗУД. Однако обмен данными между однотипными регистрами предусмотрен и реализуется командами типа регистр — регистр (I—I, F—F).

Сопроцессоры могут быть как статического (СУМ, УМН), так и динамического (СПЭФ) типов. В сопроцессорах первого типа отсутствуют элементы запоминания промежуточных результатов, в них используются схемы комбинационного действия, работа которых не тактируется внутренним генератором. С того момента времени, когда в конкретный входной F-регистр будет загружен последний операнд, модуль статического типа начинает исполнять ту арифметическую (логическую) операцию, для которой он предназначен. По истечении определенного отрезка времени в его выходном F-регистре будет находиться результат операции. В сопроцессорах данного типа узел управления как таковой практически отсутствует.

В сопроцессорах динамического типа, наоборот, присутствуют элементы запоминания промежуточных результатов. Наряду с комбинационными в них используются схемы последовательностного действия, работа которых тактируется внутренним генератором. В таких устройствах наличие внутреннего блока управления, синхронизирующего работу всех узлов, является обязательным. Инициализация (запуск) блока управления, как правило, производится специальной командой. Разумеется, этому должна предшествовать процедура загрузки операндами входных F-регистров модуля.

Таким образом, инициализация сопроцессоров может осуществляться либо только загрузкой входных F-регистров (модули стати-

ческого типа), либо загрузкой и последующим запуском специальной командой (модули динамического типа). Причем источниками операндов, как это предусмотрено системой команд АП-32, могут быть либо выходные F-регистры самих сопроцессоров (регистр — регистр), либо ОЗУД (память — регистр).

Команды (все сопроцессоры дешифрируют их одновременно) поступают по шине CMD и представляют собой старшее полуслово (16 разрядов) команды К (32 разряда). При обмене данными между F-регистрами такой команды половинного формата (регистр — регистр) достаточно. В случае же обмена с ОЗУД необходим полный формат команды с адресной частью. Поскольку адрес ОЗУД формируется ПЦЧ, то полный формат команды передается ему, а сопроцессорам только ее старшая половина. В ней кодируются операция обращения к ОЗУД и адрес F-регистра (источника, приемника). В режиме «ЗАХВ», когда БУ АП-32 блокируется (управление передается ЭВМН), источником команд (CMD), адреса ОЗУД (AD) и данных (D) является ИНТ.

Для регистрации признаков результата операции (знак, нуль и т. п.) в сопроцессорах имеются флаговые одноразрядные регистры, выходы которых шиной FL соединяются с БУ и регистром флагов интерфейса.

Флаговые регистры могут быть неадресуемые, а могут наравне с F-регистрами иметь адрес. Неадресуемые флаги используются непосредственно командами условного перехода, по одной команде на конкретный флаг. Состояние адресуемого флага нужно предварительно установить (в соответствии с признаком результата операции), для чего используется специальная команда. Затем уже можно применять команду перехода, условием которого является это состояние.

При чтении выходного регистра модуля, содержимое которого оказалось результатом ошибочной операции, сопроцессором будет установлен флаг FA, что приведет к останову АП-32. В этом случае БУ вырабатывает сигнал АНЛТ, с помощью которого в ИНТ формируется запрос на прерывание ЭВМН с фиксацией в регистре статуса ИНТ информации о причине останова.

Структура памяти АП-32 отличается от классической. Как известно, классическая организация памяти предполагает наличие адресного регистра единственного типа. В современных ЭВМ его роль обычно выполняют универсальные аккумуляторы.

В АП-32 конструктивно и логически память разделена на две области: 1) команд (ОЗУП); 2) данных (ОЗУД). Такая структура

позволяет разделить общий поток чисел на два отдельных потока.

Каждая область имеет свой адресный регистр. Роль адресных регистров ОЗУД играют I-регистры ПЦЧ с достаточно универсальными логическими возможностями формирования адреса. Адресным регистром области команд служит счетчик команд в БУ, функции которого резко ограничены. Это дало возможность при проектировании этого регистра применить быстродействующие микросхемы и добиться малой длительности командного цикла. При обращении к ОЗУД извлечение операнда из этой области и выборка команды из области команд (ОЗУП) производится практически одновременно. Следует отметить, что платы ОЗУП и ОЗУД не взаимозаменяемы и являются отдельными модулями.

### 1.3. Поток информации в АП-32

Своеобразие архитектуры АП-32 состоит в том, что в зависимости от режимов работы направление основных потоков информации (данных, команд, адресов ОЗУД) в нем меняются.

На рис. 1.2 изображена схема потоков информации в режиме «АВТ» («ИНИЦ»). Источником команд являются БУ и ОЗУП.

Команды полного формата (К) поступают из ОЗУП только в ПЦЧ и БУ. На шину CMD БУ выводится только половинный формат (старшее полуслово).

Адрес (AD) ОЗУД формируется I-регистрами в командах обращения (LD ST) к ОЗУД.

Потоки данных распределяются между ОЗУД и I-, F-регистрами. Помимо упомянутых (ОЗУД-I, ОЗУД-F) существуют и два других потока: I—I, F—F.

В режиме «АВТ» функции ИНТ резко ограничены. В этом случае в нем исполняются всего две команды для передачи в ЭВМН статусной информации (через регистр статуса и управления). Эта информация используется для синхронизации процессов в АП-32 с процессами в ЭВМН.

В режиме «ЗАХВ» направление основных потоков информации в АП-32 меняется (рис. 1.3).

Так как БУ в данном режиме блокирован и отключен от магистрали АП-32, то источником команд (CMD) оказывается ИНТ, а не ОЗУП. Поскольку для адресации ОЗУД используется регистр адреса в ИНТ, а для ОЗУП—счетчик команд в БУ, то для обмена данными между памятью, I-, F-регистрами и ИНТ достаточно, как уже упоминалось, команды половинного формата.

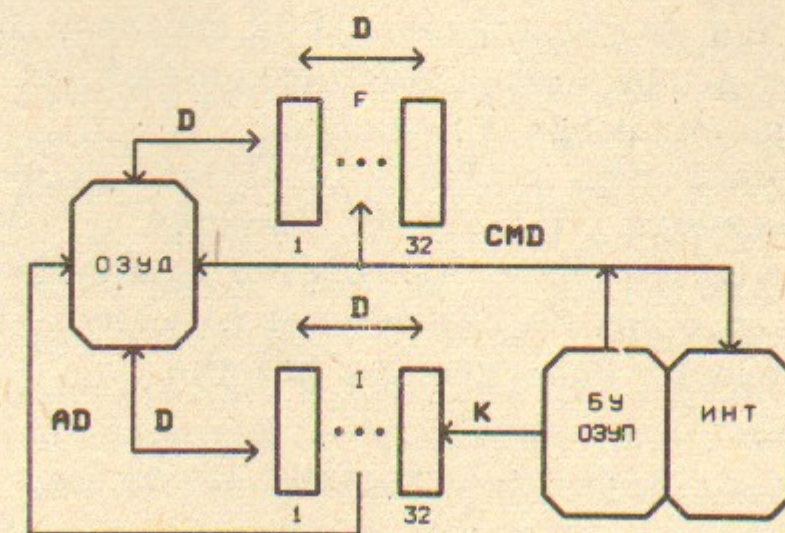


Рис. 1.2. Поток информации в АП-32 в режиме «АВТ».

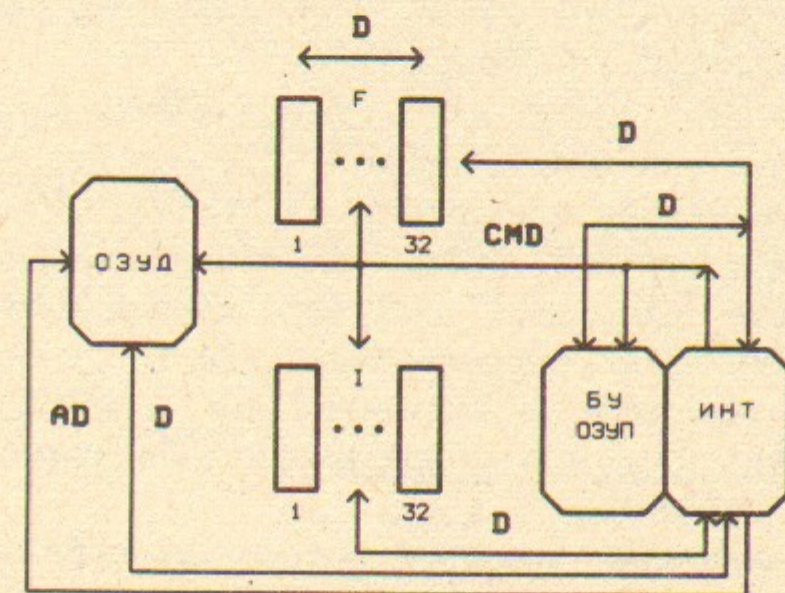


Рис. 1.3. Поток информации в АП-32 в режиме «ЗАХВ».

Когда обмен производится между I-, F-регистрами и ИНТ, то ОЗУД отключается от магистрали АП-32 сигналом EMD. В модулях схемы дешифрации команд идентифицируют тот или иной тип регистра по старшему разряду команды.

В случае обмена ИНТ с ОЗУД ( $EMD=0$ ) источником потока адресов (AD) является ИНТ. Он же является источником (приемником) и потока данных (D). Аналогично производится обмен и с ОЗУП. И в том, и в другом случае регистр адреса ИНТ и счетчик команд БУ работают в инкрементном режиме.

## 2. СИСТЕМА КОМАНД АП-32

### 2.1. Основные принципы

При разработке системы команд АП-32 основная цель состояла в том, чтобы получить удовлетворительный компромисс между простотой и богатством ее функциональных возможностей. Кроме того, необходимо было учитывать, что быстродействие процессора в значительной степени определяется длительностью командного цикла.

Ниже сформулированы основные принципы построения системы команд, заключающиеся в следующем.

1. В системе команд АП-32 нет многотактных команд. Все команды однократные и однословные (32 разряда), что позволило упростить процедуру выборки их из ОЗУП и сократить неизбежные потери времени на задержках электрических схем.
2. Разновидности временных диаграмм командного цикла АП-32 сведены к минимуму — использовано всего два типа. Это упростило схему генераторов состояний БУ и ИНТ, реализующих временные диаграммы, и сократило потери времени на инициализацию их работы.
3. Поле кода операции команды состоит из ограниченного количества разрядов (6). Причем команды кодируются как двоичным (3 разряда), так и позиционным кодом. Такая кодировка позволила сократить фазу дешифрации команды. Малый размер поля кода операции ограничивает и список команд. Однако, система команд АП-32, как показывает опыт, вполне достаточна для реализации языков программирования высокого уровня.
4. Для достижения высокого быстродействия исполнения в неко-

торых командах выделены поля, разряды которых имеют смысл управляющих сигналов. Это означает, что уже в фазе дешифрации команды сигналы этих разрядов подаются непосредственно на соответствующие операционные схемы модулей, а в фазе исполнения они начинают играть роль управляющих.

5. Системой команд все модули, независимо от их функционального назначения, интерпретируются как регистры. Поэтому различные арифметико-логические операции, особенно операции над данными с плавающей запятой, дополнительно к коду операции кодируются еще и адресами соответствующих F-регистров и иницируются командами типа загрузки/выгрузки (LD/ST). Такой подход позволил создать компактную систему команд, пригодную для достаточно широкой номенклатуры модулей.

### 2.2. Командный цикл

Генератор состояний БУ АП-32 вырабатывает два вида временных диаграмм командного цикла. На рис. 2.1 представлена диаграмма, соответствующая командам обращения к памяти данных (LD/ST).

Командный цикл состоит из 4 фаз однократной длительности  $T_1$ , практически полностью зависящей от быстродействия памяти. Управляющие сигналы S1, S2, S3, S4 вырабатываются либо БУ («АВТ»), либо ИНТ («ЗАХВ») и передаются на сквозные полосковые линии (S) магистрали АП-32.

Диаграмма всех остальных команд по форме аналогична предыдущей и отличается от нее только тем, что короче на одну фазу  $T_1$ .

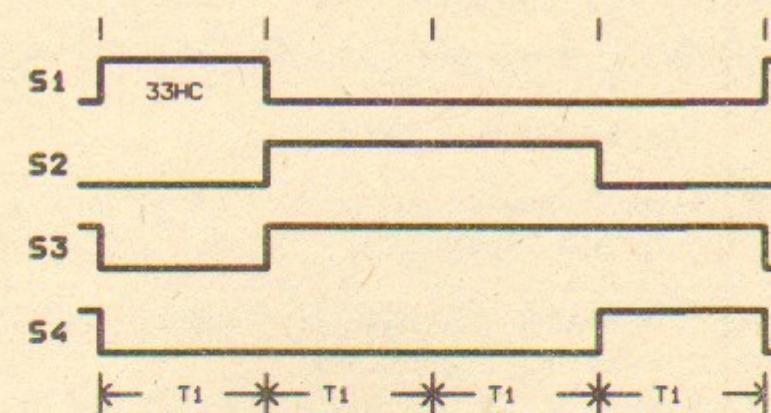


Рис. 2.1. Командный цикл АП-32 типа LD/ST.



### 2.3. Форматы команд

Основная информация о команде содержится в 6-разрядном коде операции. Это поле подвергается дешифрации всеми модулями уже в фазе выборки команды из памяти программ. Оно представлено на рис. 2.2.

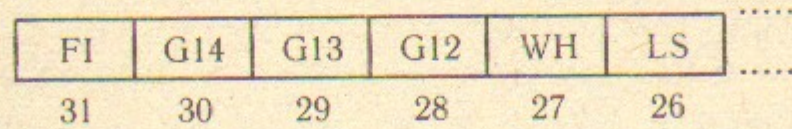


Рис. 2.2. Код операции команды.

Все команды делятся на 8 групп. Номер группы кодируется двоичными разрядами G14, G13, G12 (табл. 2.1). Разрядом FI обозначен тип регистра (FI=0—F-регистр, FI=1—I-регистр). Разряд WH кодирует тип данных (WH=0—слово, WH=1—полуслово). Разряд LS указывает направление обмена данными:

LS	Направление
0	ОЗУД → регистр
1	регистр → ОЗУД

Таблица 2.1

G14	G13	G12	Команды
0	0	0	Управление программой, условные переходы.
0	0	1	Управление программой.
0	1	0	Загрузка I-регистров непосредств.
0	1	1	Управление флагами ПЦЧ.
1	0	0	Обмен данными между I-, F-регистрами и ОЗУД с прямой адресацией.
1	0	1	Обмен данными между I-, F-регистрами и ОЗУД с индексной адресацией.
1	1	0	Арифметико-логические операции.
1	1	1	Команды действия.

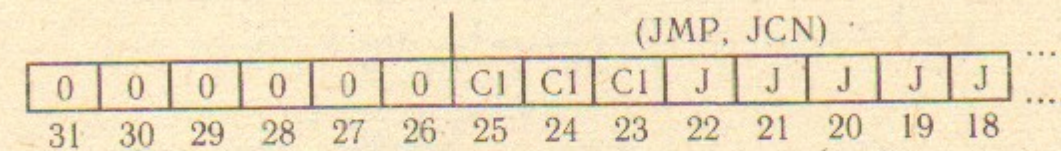
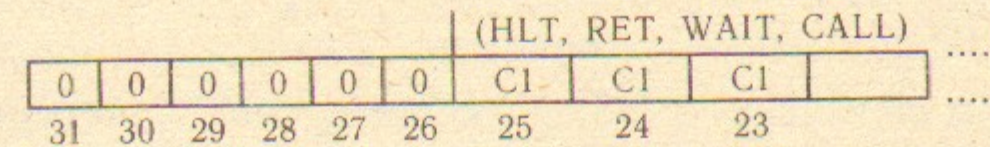
Каждая группа состоит из различного количества команд. Так, например, в группе 010 всего одна команда, в то время как в 110 количество команд весьма велико. В табл. 2.2 показано, в каких модулях АП-32 исполняются указанные группы команд.

Таблица 2.2

Группа	В каких модулях исполняется
000	БУ
001	БУ
010	ПЦЧ
011	ПЦЧ
100	ОЗУД, ПЦЧ, ППЗ
101	ОЗУД, ПЦЧ, ППЗ
110	ПЦЧ, ППЗ
111	ППЗ

#### 1. Группа 000.

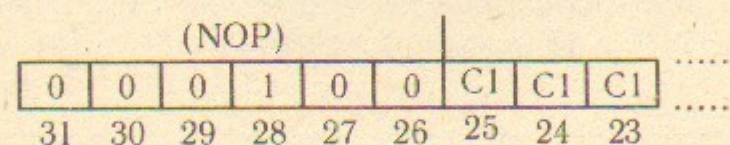
К этой группе относятся команды управления программой (HLT, RET, WAIT, CALL) и команды безусловного и условного переходов (JMP, JCN). Дополнительно к полю кода операций, который для всех этих команд одинаков, используются и другие поля (C1, J), а именно:



Разрядами поля C1 кодируется конкретный тип команды, а полем J—условия перехода.

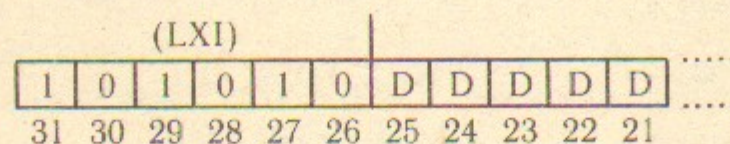
## 2. Группа 001.

В этой группе всего одна команда NOP (нет операций).



## 3. Группа 010.

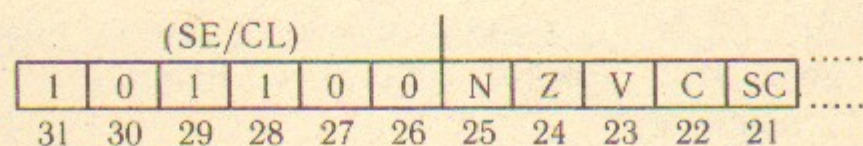
Она тоже состоит из одной команды LXI. Производит загрузку I-регистра непосредственно 19-разрядным беззнаковым операндом (операнд в поле команды).



Поле DD — код адреса приемника.

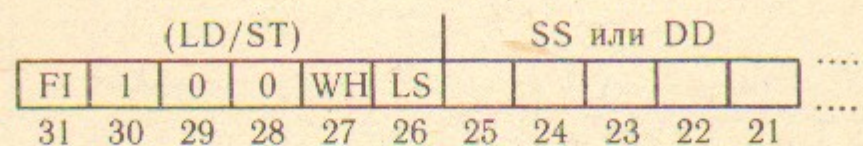
## 4. Группа 011.

Команды установки/сброса флагов ПЦЧ. Дополнительно к полю кода операции используется еще 5-разрядное поле N, Z, V, C, SC.



## 5. Группа 100.

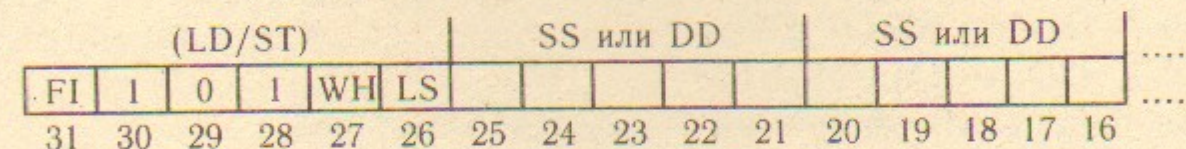
Команды обращения к ОЗУД (метод прямой адресации).



Осуществляют загрузку/выгрузку ОЗУД содержимым I-, F-регистров. Адрес памяти содержится в адресном поле команды. Тип регистров, тип данных, направление обмена кодируются разрядами FI, WH, LS. Поле SS (DD) — код регистра-источника (приемника).

## 6. Группа 101.

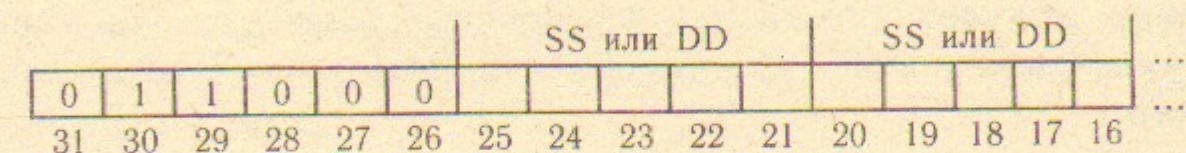
Содержание команд аналогично командам предыдущей группы, однако метод адресации ОЗУД — индексный.



Добавлено поле адреса регистра, содержащего базовый адрес ОЗУД.

## 7. Группа 110.

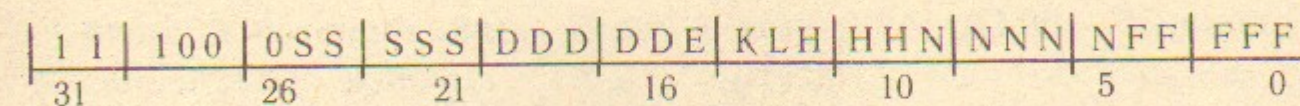
Команды типа регистр — регистр для выполнения арифметико-логических операций над данными обоих форматов.



Представленный формат команды соответствует случаю, когда FI=0, WH=0, LS=0 (команды типа MVF).

Таковыми командами осуществляется обмен данными между F-регистрами, доступными для записи и чтения. Поскольку вид операции над содержимым F-регистров кодируется их адресами, то для выполнения предполагаемых арифметико-логических действий достаточно команды 16-разрядного формата. Эти команды исполняются как в режиме «АВТ», так и в режиме «ЗАХВ».

Операции над данными I-регистров осуществляются командами 32-разрядного формата (команды типа FUNC). В этом случае FI=1, а вид конкретной операции закодирован в младшей части слова команды. Эти команды исполняются только в режиме «АВТ».



Здесь:

FFF — код арифметико-логической операции;

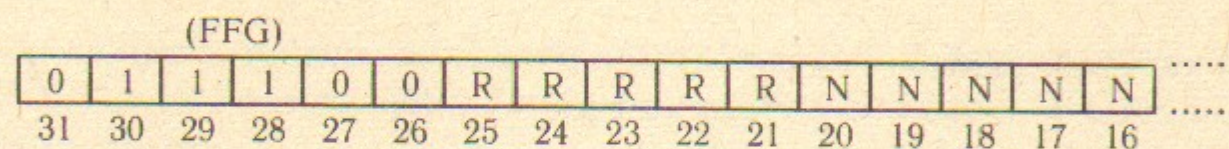
HNN — код операции сдвига;

NNN — код числа сдвигов;

- L — источник кода числа сдвигов  
 (L=0 — число сдвигов задано полем NNN,  
 L=1 — число сдвигов задано в регистре-источнике);
- K — код вида операции  
 (K=0, DD ← SS.OP.DD;  
 K=1, DD ← DD.OP.SS);
- E — код результата операции  
 (E=0 — результат записывается в приемник,  
 E=1 — результат не записывается);
- SS — код адреса I-регистра — источника;  
 DD — код адреса I-регистра — приемника.

### 8. Группа 111.

Команды этой группы, называемые командами действия, используются, как правило, для получения информации о текущем состоянии модулей (установка адресуемых флагов), для инициации (или блокировки) узлов управления и других аналогичных действий.



Здесь поле RR — адрес псевдо-регистра модуля, NN — поле «расширения» кода адреса псевдо-регистра. Предусмотрено в качестве резерва.

Полный список команд АП-32 дан в Приложении.

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ АП-32

Магистраль АП-32 состоит из трех двуслойных печатных плат, с помощью которых контакты трех разъемов модуля соединяются с сигнальными линиями. Все шины выполнены печатными проводниками (0,5 мм) в виде полосковых линий с волновым сопротивлением около 50—60 Ом. Оба конца линий подключаются к согласующим сопротивлениям (100 Ом, —2 В), которые обеспечивают уровень «логического нуля» (—1,65 В). Общая длина магистрали составляет 382,5 мм. В нее вмонтировано три ряда разъемов по 18 штук в каждом, что обеспечивает возможность подключения 18 печатных плат.

Процессор АП-32 собран на 14 четырехслойных печатных пла-

тах (из них 8 заняты под ОЗУ) размером 350×400 мм. Используются микросхемы отечественного производства серий 1800, 1500, 500. Уровни электрических сигналов соответствуют стандарту ЭСЛ.

Память собрана на БИС ОЗУ статического типа серии К132РУ6А (16К×1).

### Технические параметры АП-32

Производительность	около 10 млн. коротких операций в сек
Разрядность слова данных и команды	32
Среднее время выполнения операций над 32-разрядными операндами:	
— сложение с фиксированной запятой	0,10 мкс
— сложение с плавающей запятой (однократ. точн.)	0,10 мкс
— умножение с плавающей запятой (однократ. точн.)	0,20 мкс
— короткие операции	0,10 мкс
— деление с плавающей запятой	4,0 мкс
— вычисление элементарных функций	4,0 — 13,0 мкс
Оперативная память:	
— объем	2 Мбайт
— время доступа	0,10 мкс
Потребляемая мощность	300 Вт
Размер корпуса (без источника питания)	500×500×600 мм

### Сравнительная оценка быстродействия АП-32

Ниже приведены характеристики быстродействия АП-32 и некоторых ЭВМ на основе тестов «решето Эратосфена» (табл. 3.1) и Ветстоуна (табл. 3.2, параметр ТЕСТА N=500, однократная точность).

Таблица 3.1

ЭВМ	Время, с	Язык
АП-32	0,50	F-4
ЕС-1061	1,77	F-4
VAX-11/780	2,3	F-77
БЭСМ-6	2,6	F-4
ЭЛЕКТРОНИКА-79	3,3	F-4

Таблица 3.2

ЭВМ	Время, с	Язык
VAX 8800	10,04	F-77
АП-32	20,0	F-4
ЕС-1061	53,0	F-4
ЕС-1061	33,0	F-77, OPT(3)
ЭЛЕКТРОНИКА-79	78,6	F-77

## ЛИТЕРАТУРА

1. Бару С.Е. и др. Детектор МД-1.—Препринт ИЯФ СО АН СССР 83-39. г. Новосибирск, 1983.
2. Аксенов Г.А. и др. Универсальный арифметический процессор АП-20 для быстрой обработки данных в физическом эксперименте.—В кн.: Труды III Международной конференции по методике экспериментов на встречных пучках, изд. ИЯФ СО АН СССР. Новосибирск, 1984, с.173—178.
3. Аксенов Г.А. и др. Высокопроизводительная система для обработки физической информации.—В кн.: XII Международный симпозиум по ядерной электронике. Дубна, 1985, с.324—328.
4. Аксенов Г.А. и др. Препринт ИЯФ 85-125. Новосибирск, 1985.
5. Аксенов Г.А. и др. Препринт ИЯФ 86-11. Новосибирск, 1986.
6. Аксенов Г.А. и др. Препринт ИЯФ 85-118. Новосибирск, 1985.
7. Аульченко В.М. и др. Препринт ИЯФ 87-36. Новосибирск, 1987.

## Приложение

## СИСТЕМА КОМАНД АП-32

## 1. Условные обозначения в операторах языка ассемблера и пояснениях действия команды

FSR	F-регистр — источник;
FDR	F-регистр — приемник;
ISR	I-регистр — источник;
IDR	I-регистр — приемник;
TMP	внутренний регистр для временного хранения;
LR	регистр связи;
PC	программный счетчик;
OPR	непосредственный операнд;
ADDR	адрес памяти;
N	флаг «знак» (ф.з.);

Z	флаг «нуль» (ф.з.);
V	флаг «переполнение» (ф.з.);
C	флаг (и бит) «перенос» (ф.з.);
WT	флаг «конец»;
FG	флаг общего назначения;
FZ	флаг «нуль» (п.з.);
FN	флаг «знак» (п.з.);
FA	флаг «ошибка»;
FLG	адрес флага для команды FFG;
(...)	содержимое регистра или памяти;
[ID]	необязательное поле константы-идентификатора;
'...'	вектор сдвига в команде сдвига;
n.	нуль.

## 2. Состояние флага после исполнения команды

*	устанавливается по результату операции;
0	устанавливается в 0;
1	устанавливается в 1;
—	не изменяется;
?	неопределенное состояние.

## 3. Имена регистров АП-32

Все регистры в поле операнда на языке ассемблера можно обозначать как #XX, где XX—восьмеричный адрес для F-регистров, и как %XX, где XX—восьмеричный номер I-регистра. Кроме того, введены символические имена.

## 3.1. Регистры БУ

Имя	Код	Название
PC	00	программный счетчик
LR	01	регистр связи

## 3.2. Регистры ПЦЧ (аккумуляторы)

Имя	Код	Название
R0	0	аккумулятор 0
...		
R8	10	аккумулятор 8
...		
R31	37	аккумулятор 31

### 3.3. Регистры ППЗ

Имя	Код	Название	Флаги			
			FZ	FN	WT	FA
AD	20	Слагаемое сумматора	*	*	—	—
AR	21	Слагаемое сумматора	*	*	—	—
AF	20	Результат сумматора	*	*	—	*
AM	21	Результат сумматора по модулю	*	*	—	*
SB	22	Уменьшаемое вычитателя	*	*	—	—
SR	23	Вычитаемое вычитателя	*	*	—	—
SF	20	Результат вычитателя	*	*	—	*
SM	21	Результат вычитателя по модулю	*	*	—	*
MP	30	Множимое умножителя	—	—	—	—
MR	31	Множитель умножителя	—	—	—	—
MF	30	Результат умножителя	—	—	—	*
VX	05	Регистр 0 СПЭФ	—	—	*	*
VY	06	Регистр 1 СПЭФ	—	—	*	*
VZ	07	Регистр 2 СПЭФ	—	—	*	*

### 3.4. Псевдо-регистры интерфейса

Имя	Код	Название
—	37	Прерывание в ЭВМН (по команде «FFG #37»)
—	36	«Конец процесса» (по команде «FFG #36»)

### 4. Команды передачи управления

Оператор ассемблера	Описание команды
HALT [ID]	Программный останов процессора
NOP [ID]	Ничего не делать в течение цикла команд
RETURN [ID]	$PC \leftarrow (LR)$
WAIT [ID]	Ничего не делать до появления флага WT
CALL ADDR	$LR \leftarrow (PC+1), PC \leftarrow ADDR$

Оператор ассемблера	Описание	Условие перехода
JUMP ADDR	$PC \leftarrow ADDR$	Безусловно
JCN FMI,ADDR	$PC \leftarrow ADDR$	меньше н. FN=1
JCN FPL,ADDR	$PC \leftarrow ADDR$	больше н. FN=0
JCN FEQ,ADDR	$PC \leftarrow ADDR$	равно н. FZ=1
JCN FNE,ADDR	$PC \leftarrow ADDR$	не равно н. FZ=0

JCN FGT,ADDR	$PC \leftarrow ADDR$	больше	FZ=0 и FN=0
JCN FLE,ADDR	$PC \leftarrow ADDR$	меньше, равно н.	FZ=1 или FN=1
JCN FA,ADDR	$PC \leftarrow ADDR$		FA=1
JCN WT,ADDR	$PC \leftarrow ADDR$		WT=1
JCN FG,ADDR	$PC \leftarrow ADDR$		FG=1
JCN NE,ADDR	$PC \leftarrow ADDR$	не равно н.	Z=0
JCN EQ,ADDR	$PC \leftarrow ADDR$	равно н.	Z=1
JCN PL,ADDR	$PC \leftarrow ADDR$	больше н.	N=0
JCN MI,ADDR	$PC \leftarrow ADDR$	меньше н.	N=1
JCN VC,ADDR	$PC \leftarrow ADDR$		V=0
JCN VS,ADDR	$PC \leftarrow ADDR$		V=1
JCN CC,ADDR	$PC \leftarrow ADDR$		C=0
JCN CS,ADDR	$PC \leftarrow ADDR$		C=1
JCN GE,ADDR	$PC \leftarrow ADDR$	больше, равно н.	$(N.XOR.V) = 0$
JCN LT,ADDR	$PC \leftarrow ADDR$	меньше н.	$(N.XOR.V) = 1$
JCN GT,ADDR	$PC \leftarrow ADDR$	больше н.	$(Z.OR.(N.XOR.V)) = 0$
JCN LE,ADDR	$PC \leftarrow ADDR$	меньше, равно н.	$(Z.OR.(N.XOR.V)) = 1$
JCN HI,ADDR	$PC \leftarrow ADDR$	больше	$(C.OR.Z) = 0$
JCN LOS,ADDR	$PC \leftarrow ADDR$	меньше, равно	$(C.OR.Z) = 1$
JCN HIS,ADDR	$PC \leftarrow ADDR$	больше, равно	C=0
JCN LO,ADDR	$PC \leftarrow ADDR$	меньше	C=1

### 5. Команды обращения к памяти данных

Оператор ассемблера	Описание команды
LXI IDR,OPR	$IDR \leftarrow OPR$
LD FDR,ADDR	$FDR \leftarrow (ADDR)$
LD IDR,ADDR	$IDR \leftarrow (ADDR)$
LDH IDR,ADDR	$IDR \leftarrow 1/2(ADDR)$ , «расширение» знака
LD FDR,OFFSET(ISR)	$TMP \leftarrow (ISR) + OFFSET,$ $FDR \leftarrow (TMP)$
LD FDR,(ISR)	$TMP \leftarrow (ISR),$ $FDR \leftarrow (TMP)$
LD IDR,OFFSET(ISR)	$TMP \leftarrow (ISR) + OFFSET,$ $IDR \leftarrow (TMP)$
LD IDR,(ISR)	$TMP \leftarrow (ISR),$ $IDR \leftarrow (TMP)$
LDH IDR,OFFSET(ISR)	$TMP \leftarrow (ISR) + OFFSET,$

LDH	IDR, (ISR)	IDR ← 1/2(TMP), «расширение» знака TMP ← (ISR), IDR ← 1/2(TMP), «расширение» знака
ST	FSR, ADDR	(ADDR) ← FSR
ST	ISR, ADDR	(ADDR) ← ISR
STH	ISR, ADDR	(ADDR) ← 1/2 ISR
ST	FSR, OFFSET(IDR)	TMP ← (IDR) + OFFSET, (TMP) ← FSR
ST	FSR, (IDR)	TMP ← (IDR), (TMP) ← FSR
ST	ISR, OFFSET(IDR)	TMP ← (IDR) + OFFSET, (TMP) ← ISR
ST	ISR, (IDR)	TMP ← (IDR), (TMP) ← ISR
STH	ISR, OFFSET(IDR)	TMP ← (IDR) + OFFSET, (TMP) ← 1/2 ISR
STH	ISR, (IDR)	TMP ← (IDR), (TMP) ← 1/2 ISR

#### 6. Команды операций над числами с плавающей запятой

Оператор ассемблера	Описание команды
MVF FSR, FDR	FDR ← FSR
FFG FLG	FG ← (FLG)

#### 7. Арифметико-логические операции над числами с фиксированной запятой

Оператор ассемблера	Описание команды	N	Z	V	C	
MOV	ISR, IDR	IDR ← ISR	*	*	0	—
ADD	ISR, IDR	IDR ← ISR + IDR	*	*	*	*
SUB	ISR, IDR	IDR ← IDR - ISR	*	*	*	*
CMP	ISR, ISD	ISR - ISD	*	*	*	*
AND	ISR, IDR	IDR ← ISR.AND.IDR	*	*	0	—
BIT	ISR, IDR	ISR.AND.IDR	*	*	0	—
BIC	ISR, IDR	IDR ← (.NOT.ISR).AND.IDR	*	*	0	—
OR	ISR, IDR	IDR ← ISR.OR.IDR	*	*	0	—
BIS	ISR, IDR	IDR ← ISR.OR.IDR	*	*	0	—
XOR	ISR, IDR	IDR ← ISR.XOR.IDR	*	*	0	—
CLR	IDR	IDR ← 0	0	1	0	—
ONE	IDR	IDR ← -1	1	0	0	—

MIN	IDR	IDR ← -1	1	0	0	0
COM	IDR	IDR ← .NOT.IDR	*	*	0	—
CMV	ISR, IDR	IDR ← .NOT.ISR	*	*	0	—
INC	IDR	IDR ← IDR + 1	*	*	*	*
DEC	IDR	IDR ← IDR - 1	*	*	*	*
TST	IDR	IDR ← IDR	*	*	0	0
ADC	IDR	IDR ← IDR + C	*	*	*	*

#### 7.1. Команды сдвигов

Оператор ассемблера	Описание команды	N	Z	V	C		
ROR	OPR, IDR	IDR ← > IDR >	'OPR'	*	*	0	?
ROR	ISR, IDR	IDR ← > IDR >	'ISR'	*	*	0	?
LSR	OPR, IDR	IDR ← 0 > IDR	'OPR'	*	*	0	—
LSR	ISR, IDR	IDR ← 0 > IDR	'ISR'	*	*	0	—
ASR	OPR, IDR	IDR ← N > IDR	'OPR'	*	*	0	—
ASR	ISR, IDR	IDR ← N > IDR	'ISR'	*	*	0	—
ASL	OPR, IDR	IDR ← IDR < 0	'OPR'	*	*	0	—
LSL	OPR, IDR	IDR ← IDR < 0	'OPR'	*	*	0	—
ASL	ISR, IDR	IDR ← IDR < 0	'ISR'	*	*	0	—
LSL	ISR, IDR	IDR ← IDR < 0	'ISR'	*	*	0	—
ARC	OPR, IDR	IDR ← N > IDR > C	'OPR'	*	*	*	*
ARC	ISR, IDR	IDR ← N > IDR > C	'ISR'	*	*	*	*
ALC	OPR, IDR	IDR ← C < IDR < 0	'OPR'	*	*	*	*
ALC	ISR, IDR	IDR ← C < IDR < 0	'ISR'	*	*	*	*
RRC	OPR, IDR	IDR ← C > IDR > C	'OPR'	*	*	*	*
RRC	ISR, IDR	IDR ← C > IDR > C	'ISR'	*	*	*	*
RLC	OPR, IDR	IDR ← C < IDR < C	'OPR'	*	*	*	*
RLC	ISR, IDR	IDR ← C < IDR < C	'ISR'	*	*	*	*

#### 8. Команды управления флагами

Оператор ассемблера	Описание команды	N	Z	V	C	
CLF	N, Z, V, C	Очистить флаги, в поле которых 0	*	*	*	*
SEF	N, Z, V, C	Установить флаги, в поле которых 1	*	*	*	*
CLC			—	—	—	0
CLV			—	—	0	—
CLZ			—	0	—	—

CLN		0	—	—	—
CCC		0	0	0	0
SEC		—	—	—	1
SEV		—	—	1	—
SEZ		—	1	—	—
SEN		1	—	—	—
SCC		1	1	1	1
DFA	Запрет установки FA по ошибке				
EFA	Разрешение установки FA по ошибке				

### 9. Команда типа микрокоманды

Оператор ассемблера:		N	Z	V	C
MIX	ISR, IDR, E, K, L, HHH, NNN, CCC	*	*	*	*
SSS	регистр-источник				
DDD	регистр-приемник				
E=0	результат записывается в приемник				
1	результат не записывается				
K=0	IDR ← ISR.Op.IDR				
1	IDR ← IDR.Op.ISR				
L=0	число сдвигов в поле NNN				
1	число сдвигов в регистре SSS				
HHH	код операции сдвига				
NNN	число сдвигов непосредственно				
CCC	код операции АЛУ				

#### Описание команды:

Над приемником выполняется операция сдвига, описанная полем HHH, число сдвигов, в зависимости от бита L, извлекается из поля NNN или регистра SSS, затем над результатом сдвига и источником SSS выполняется арифметическая или логическая операция, описанная полем CCC, результат записывается в приемник (E=0) или не записывается (K=1).

Все поля операндов, кроме ISR и IDR, — численные константы.

*Г.А. Аксенов, А.В. Кислицин, Ю.И. Мерзляков,  
В.Я. Сазанский, И.А. Ткаченко, А.Г. Чертовских*

### УНИВЕРСАЛЬНЫЙ АРИФМЕТИЧЕСКИЙ ПРОЦЕССОР АП-32

#### 1. Архитектура, система команд, технические характеристики

Ответственный за выпуск С.Г. Попов

Работа поступила 14.12 1989 г.  
Подписано в печать 28.12 1989 г. МН 10616  
Формат бумаги 60×90 1/16 Объем 2,3 печ.л., 1,9 уч.-изд.л.  
Тираж 150 экз. Бесплатно. Заказ № 175

*Набрано в автоматизированной системе на базе фото-  
наборного автомата ФА1000 и ЭВМ «Электроника» и  
отпечатано на ротапринтере Института ядерной физики  
СО АН СССР,  
Новосибирск, 630090, пр. академика Лаврентьева, 11.*